Gruppe „Denial of Service“

Philipp Oldenburg, Patrick Zumsteg, Simon Wallny

Frühjahrssemester 2015

Inhaltsverzeichnis

1 Vorwort

Das Projekt „Denial of Service“, entstand im Rahmen der Vorlesung „Internet-Technologien“ an der Universität Basel im Frühjahrssemester 2015.
Die bearbeitende Gruppe besteht aus Philipp Oldenburg, Patrick Zumsteg und Simon Wallny.

2 Definition

„Denial of Service“ (engl. *Verweigerung des Dienstes*) beschreibt eine Art eines Angriffs auf einen öffentlich verfügbaren Dienst im Internet. Ziel dieses Angriffes ist es, dass der Dienst des Ziels, etwa ein Online-Shop, zeitweise nicht mehr zur Verfügung steht. Dies wird auf unterschiedliche Weise bewerkstelligt, aber die zugrundeliegende Idee ist in jeden Fall dieselbe: Die Ressourcen des Dienstes, Prozessorauslastung, Speicher oder Bandbreite, die immerhin über das Internet zur Verfügung gestellt werden, werden in unverhältnismäßigem Maße beansprucht, sodass tatsächlichen Nutzern des Dienstes keine Kapazitäten mehr zur Verfügung stehen.
Erfolgt ein solcher Angriff über mehrere Rechner an verschiedenen Orten, spricht man von einer „Distributed Denial of Service“-Attacke (engl. *Verteilte Verweigerung des Dienstes*), kurz DDoS.
Das Ausführen von Angriffen dieser Art in öffentlichen Netzen sind in der Schweiz gemäß Artikel 144 des Strafgesetzbuches strafbar.

3 Angriffs- und Verteidigungs-Strategien anhand einiger Beispiele

DoS-Angriffe können über verschiedene Wege gefahren werden, und auf unterschiedlichen Ebenen unterschiedliche Ressourcen blockieren. Im folgenden sind einige Beispiele aufgeführt.

3.1 Applikationsebene - Slow Headers

Angriff

Ein „Slow Headers“-Angriff ist ein auf Webserver spezialisierter, besonders eleganter Angriff, der dem Angreifer kaum Ressourcen abverlangt. Hierbei wird dem Ziel ein Strom von unfertigen Headern von HTTP-Get Anfragen geschickt. Der Server muss, damit der Angriff effektiv ist, konfiguriert sein die Verbindung offen zu halten und auf den Rest der Anfrage zu warten. Diese Verbindungen lassen sich nun offen halten, indem bevor der timeout erfolgt, der mehrere Minuten lang sein kann, ein weiterer Teil der Anfrage geschickt wir, wobei der Header allerdings immer noch nicht geschlossen wird. Der Webserver, der nur eine endliche Menge Verbindungen halten kann, wird so für normale Nutzer unzugänglich.

Verteidigung

Obwohl Slow-Header-Angriffe extrem mächtig sind und einen Server mühelos in die Knie zwingt, solange die Angreiferseite mehr Ports offen halten kann als der verteidigende Anbieter, ist es leider (oder zum Glück) sehr leicht, sich gegen diese Attacken zu immunisieren. Da ein Slow-Headers-Angriff alle Verbindungen besetzt und offen hält, die der Server bereitstellt, muss dieser nichts weiter tun, als die Anzahl an simultanen Verbindungen, die ein Client (sprich: eine IP-Adresse, was nicht unproblematisch ist) offen halten kann, zu begrenzen. Hierfür gibt es beispielsweise das Apache-Modul *mod_qos*.
Allerdings ist diese Verteidigung nicht narrensicher; einerseits kann es sein, dass ein Serviceanbieter damit rechnet, mit vielen Clients gleichzeitig zu arbeiten, die sich hinter einem NAT befinden, also aus Sicht des Servers von der gleichen IP aus kommunizieren. In diesem Fall müsste er eine große Anzahl an simultanen Verbindungen zulassen, was seine Verteidigung natürlich stark schwächt. Außerdem ist man gegen einen verteilten Angriff immer noch nur mittelmäßig geschützt: Sei x die Anzahl Verbindungen, die pro IP simultan geöffnet sein dürfen, und M die Anzahl Verbindungen, zu der der Server insgesamt fähig ist. Dann braucht es höchstens $\frac{M}{x}$ Angreifer-IPs, um den Host lahmzulegen. Die maximale Anzahl an Verbindungen, die der Server eingehen kann, ist zwar frei wählbar, allerdings müssen hinter dieser Zahl natürlich entsprechende Systemressourcen stehen, und das Problem wird nur auf CPU/RAM verschoben.

3.2 Transportebene - Syn Flood.

Angriff

Syn Flood (engl. *Syn Flut*) ist ein simpler Angriff, bei dem von dem normalen TCP-Handshake der erste Schritt von Seitens des Angreifers so schnell wie möglich wiederholt wird. Das Ziel ertrinkt nun in der Flut von Verbindungsanfragen und kann nicht mehr auf alle antworten, auch nicht auf die vereinzelten echten Anfragen seiner Nutzer.

Verteidigung

Momentan sind mehrere Wege bekannt, sich gegen eine SYN-Flut zu wehren. Ein naiver Ansatz wäre natürlich, die älteste der halboffenen Verbindungen (die also noch kein ACK vom Client erhalten haben) einfach zu recyclen und für ein neu ankommendes SYN zu verwenden. Hierbei hat natürlich der Client, dem die Verbindung ursprünglich gehörte, das Nachsehen, und im Falle einer heftigen SYN-Flut ist das Backlog des Servers immer noch (fast) nur

mit ,,böswilligen“ Verbindungen befüllt.
Gegen SYN-Floods gibt es aber schon seit langer Zeit eine "Wunderwaffe", bekannt als SYN-Cookies. Die Idee hierbei ist, dass der Server die Sequenznummer seines SYN-ACK-Pakets anhand eines üblicherweise im Minutentakt inkrementierten Zeitstempels, seiner eigenen und der Adresse des Clients berechnet. Ein Client, der wirklich eine Verbindung will, kann würde die Sequenznummer dieses Pakets um eins erhöhen und als ACK-Nummer in seinem ACK-Paket zurückschicken. Der Server kennt nun ja sowohl den aktuellen(oder letzten) Zeitstempel, als auch Quell-und Zieladressen, kann also nachrechnen, ob er diesem Client vor kurzem ein SYN-ACK geschickt haben kann.

3.3 Netzwerkebene - ICMP Flood

Angriff

ICMP Flood (engl. *ICMP Flut*) Bei einem ICMP Smurf-Angriff kann ein schlecht konfiguriertes Netzwerk ausgenutzt werden, um ein Ping-Packet das so aussieht, als komme es vom Ziel, an alle Rechner des Netzes zu schicken, die allesamt antworten und das Ziel damit stark belasten.

Verteidigung

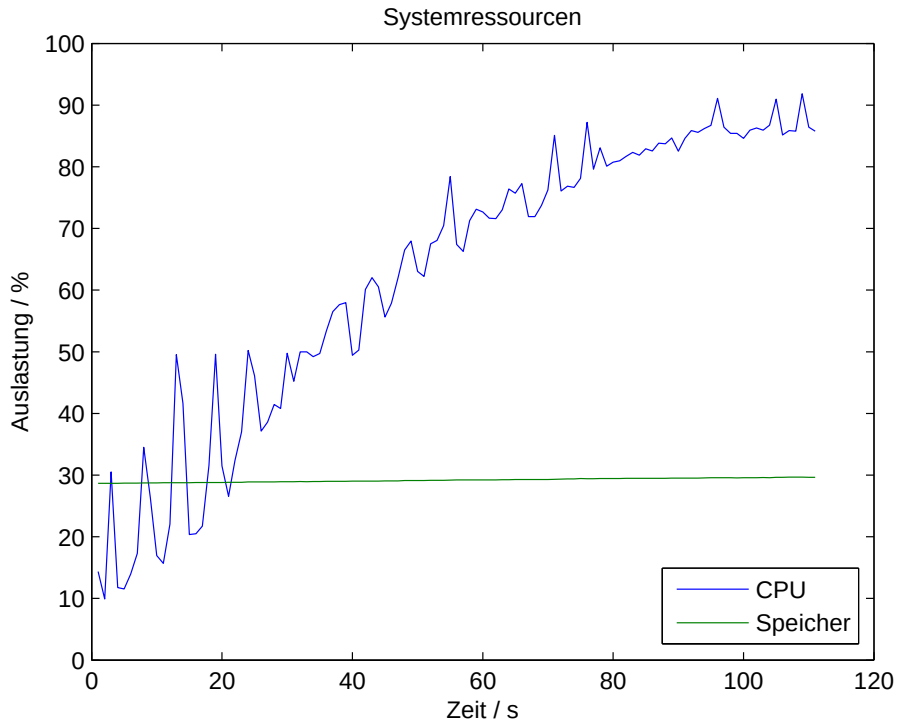
Ein häufig genutzter Service um gegen solche Angriffe vorzugehen ist die Smurf Amplifier Registry. Dieser Service wird von Internet-Providern genutzt. Es werden nun Netzwerke auf ihre Fähigkeit geprüft einen solchen Angriff zu ermöglichen. Falls sich ein solches gefährliches Netzwerk findet wird es in einer Blacklist-Datenbank eingetragen. Diese Datenbank ermöglicht es nun fortwährend Pakete aus den gefährlichen Netzwerken zu filtern.
Ein Client selbst kann natürlich auch einfach jegliche ICMP-Pakete ignorieren, sofern er sie nicht benötigt. Falls dieser aber doch den Ping Dienst anbieten möchte, bietet es sich an den ICMP transmission rate zu drosseln z.B. auf max 512Kb/s, sodass jeder darüberliegende traffic ignoriert werden kann.

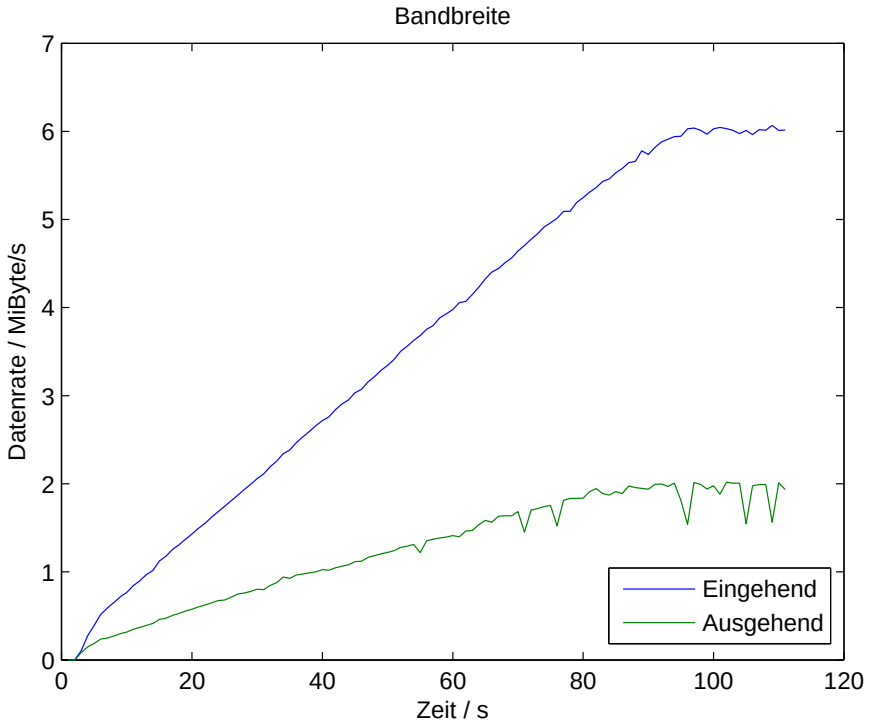
4 Bekannte Anwendungen

4.1 Low Orbit Ion Cannon

Die LOIC ist ursprünglich eine open-source Software für Belastungstests. Unsere Tests haben wir allerdings mit einer abgewandelten Version der Anwen-

dung durchgeführt, einem von einer sich zum freien Internetkollektiv Anonymous bekennender Gruppe Hacktivisten entwickeltes DoS-Angriffstool. Anders als bei den meisten frei verfügbaren Tools war ihre Bestimmung niemals Belastungstests, sondern das anrichten von Schaden an politisch motivierten Zielen. Während diese Ion Cannon in erster Linie für verteilte Angriffe, DDoS, gedacht ist, und sich zwecks dessen in einen „Hive Mode“ schalten lässt, bei dem ein Angriff zentral synchronisiert von vielen Teilnehmern gleichzeitig gestartet wird, lassen sich auch mit einer einzigen Anwendung messbare Ergebnisse erzielen.
Wir haben den uns zur Verfügung gestellten Rechner über eine lokale Verbindung mit der LOIC angegriffen, und folgende Messwerte erhalten.

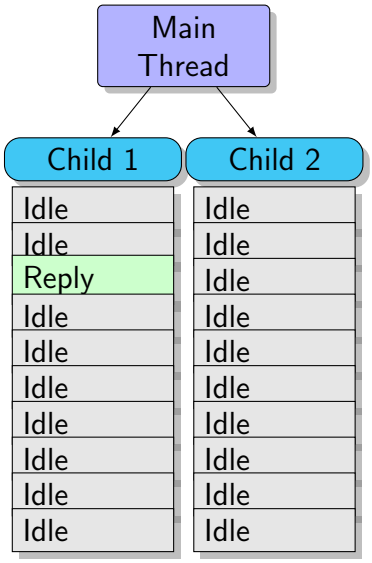




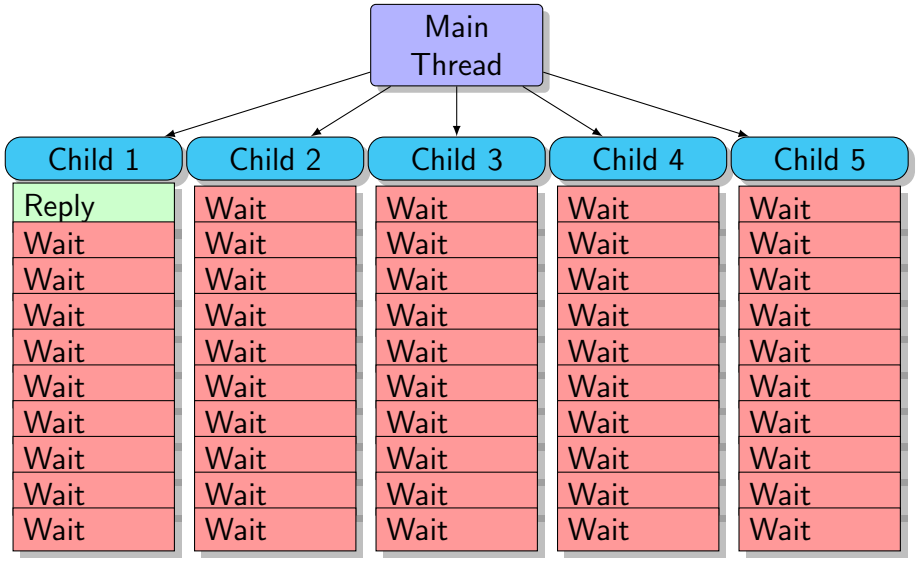
Dieser vereinzelte Angriff war nicht genug, damit der Server den Dienst vollständig quittiert, allerdings hat er schon bei diesen harmlosen Dimensionen merklich langsamer geantwortet als sonst.

4.2 Open Web Application Security Project

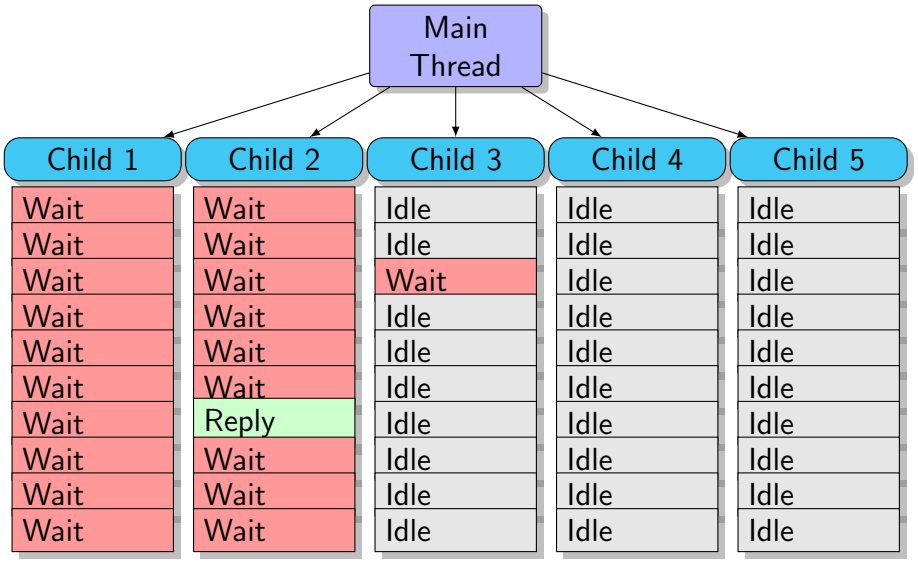
SwitchBlade, lizenziert von ProactiveRISK, ist eine Anwendung in Rahmen des OWASP um Belastungstests für eigene Webserver auszuführen. Dazu führt es einen ,,Slow Headers“-Angriff aus, und legte in unseren Tests einen frisch aufgesetzten, ungeschützten Apache-Server binnen Sekunden gründlich lahm. Die Threadkonfiguration des frisch gestarteten Servers bestand aus zwei Kindprozessen mit jeweils 25 Threads, hier auf 10 gekürzt.



Nachdem der Angriff gestartet wurde, sah die Threadkonfiguration dagegen wie folgt aus, und der Server war über den Browser nicht mehr zu erreichen.



Wenn allerdings ein Modul zur Verteidigung geladen wird, in unserem Test *mod_qos*, ist der Effekt sehr viel weniger drastisch:



Es findet zwar eine gewisse Belastung statt, aber der Server kann den Betrieb aufrechterhalten.

5 Anonymisierung

Ob der illegalen Natur von DoS-Angriffen liegt es grundsätzlich im Interesse des Angreifers seine Identität nicht preiszugeben. Proxys sind dabei keine Hilfe, da diese den Datenstrom zu stark drosseln um nachwievor effektiv zu sein. Die LOIC beispielsweise betreibt keinen Versuch die Identität der Angreifer zu verbergen. Bestimmte Angriffe allerdings erfordern nicht, dass der Angreifer seine eigene IP-Adresse preisgibt, insbesondere, da oft eine Antwort des Angegriffenen nicht erforderlich und somit ohnehin nicht erwünscht ist. So ist in gewissen Fällen **IP-Spoofing** möglich. Das bedeutet, in den geschickten Packeten eine falsche IP-Adresse als Absender anzugeben, und es ist bei gewissen Angriffen sogar ein integraler Bestandteil des Vorgehens, sich als jemand anderes, etwa das Angriffsziel auszugeben. Es kann dabei allerdings passieren, dass ein Zwischenknoten auf dem Weg zum Ziel bemerkt, dass die Absenderadresse aus einer Richtung kommt, aus der sie nicht hätte kommen dürfen, und das Paket subsequent verwirft, denn das kann nur bedeuten, dass beim Routing etwas grundlegend schiefgelaufen ist, oder jemand das System missbraucht.