DDoS Schutz

Abwehr von DDoS Attacken

Tobias Swolany aka „keksa“

DDoS@keksa.de

http://keksa.de

09/27/2010

Prolog

Zunächst ein Wort der Warnung. Ich habe leider bisher selber noch keinerlei praktische Erfahrung in diesem Themengebiet, weder auf Angriffsseite (was aber nicht besonders spektakulär sein dürfte) noch auf Abwehrseite.

Aber das ist absolut kein Grund sich nicht damit auseinanderzusetzen. DDoS Angriffe werden immer mehr zu einem riesen Stichwort in der Computersicherheit. Nicht nur, dass jeder von uns irgendwo irgendeinen kennt, der ein Botnetz besitzt und jederzeit auf Knopfdruck wild um sich rumballern kann; nein, auch in der Politik werden große DDoS Angriffe immer mehr zum Totschlagargument, um irgendwelche Gesetze in der Computersicherheit durchzubringen. Wer mein Twitter-Feed verfolgt weiß wovon ich rede; so hätte z.B. Obama sich vor paar Monaten fast schon die Möglichkeit eingesackt, das ganze Internet ausschalten zu können, um solch einen Angriff auf die US-Infrastruktur "abzuwehren" [1].

Oder es entstehen politische Spannungen weil irgendwelche Länder behaupten andere Länder würden sie per DDoS auf Regierungsrechner in Schacht halten wollen [2]. Ich meine auch gelesen zu haben, dass sich Amerika das Recht eingeräumt hat ein anderes Land anzugreifen (mit Panzern und so), wenn sie Opfer einer gezielten DDoS Attacke aus einem Land wird :D Sind die da drüben eigentlich total bekloppt? :D

Naja, das zeigt uns jedenfalls, dass in Zukunft noch viel spannendere Sachen in dieser Ecke passieren werden, denn die politische Aufmerksamkeit wächst in diesem Gebiet. Aber auch in kleineren Kreisen wird es immer mehr zum Gespräch, man kann nämlich mal eben per DDoS die Konkurrenz ausschalten; schließlich kriegt man ein DDoS aus der Cloud von Amazon ja schon für $6 [3]. Es wurde auch

schon mal dafür gesorgt, dass Ubisoft-Kunden deren Spiele nicht zocken konnten, weil die DRM-Server geplättet wurden [4]. Oder es wird mal eben ein ganzer Internet-Provider lahmgelegt... [5] Ja, sowas ist sehr geschäftsschädigend :D Meine Lieblingsmeldung ist aber immernoch von Eugene Kaspersky, Experte der Computersicherheit und Gründer des Unternehmens Kaspersky Lab (Antivirensoftware und andere Sicherheitssoftware): er hat geäußert, man könne per DDoS ganze Flugzeuge vom Himmel abschießen... [6] Bwahaha, wie geil.

DDoS, eine Unterart von DoS

Aber das reicht an Motivation zum Thema :) Schmeißen wir uns auf das Technische! Zuallererst: Was ist ein DDoS überhaupt?

DDoS ist eine Unterart von DoS. Zunächst ein paar Worte zu DoS. DoS steht für Denial of Service [7], was soviel heißt wie Dienstverweigerung. Dabei wird einem Programm (die Dienstleistung) ein Code oder Daten untergejubelt, die ein Fehler auslösen, sodass dieses sich aufhängt (Dienstverweigerung). Es gibt im Internet viele DoS-Exploits (Programme oder Dateien, die ein DoS verursachen) die z.B. den Browser aufhängen lassen. Es wird also ein Dienst (der Browser) zur Dienstverweigerung (er hängt sich auf) gebracht. Also hat ein Denial of Service stattgefunden.

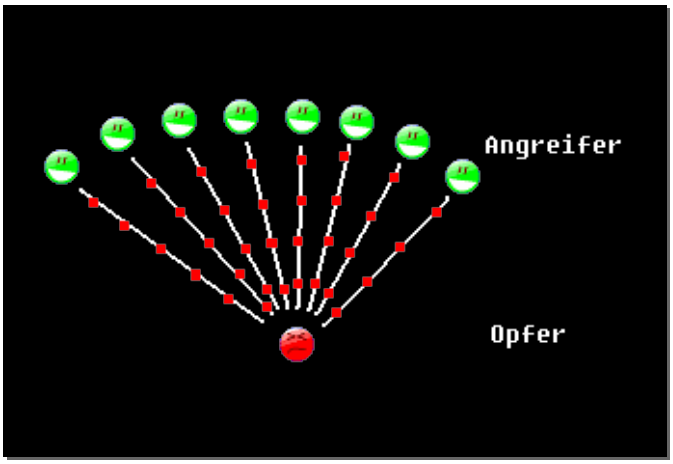
Da aber das Aufhängen eines Browsers ziemlich langweilig ist (das ist vielleicht die ersten 3 Male interessant, wenn man jemanden ärgern will), versucht man auch bei anderen Dienstleistungen ein DoS zu verursachen. Wie wäre es zum Beispiel, wenn wir einen Fehler in einer Webserver-Software finden? Wäre ja ganz lustig, man löst beim Webserver ein Fehler aus, sodass dieser dann den "Geist" aufgibt und nicht mehr erreichbar ist; schwups, ist die Internetseite nicht mehr verfügbar. Das wäre dann ein DoS beim Webserver.

Es gibt viele Leute, die das ungemein witzig finden. Oder, wie gesagt, es auch zum eigenen Nutzen verwenden, um z.B. die Konkurrenz aus dem Weg zu schaffen, oder es zu kommerzialisieren und heftigen Profit draus zu schlagen. Da es aber total umständlich ist zu jeder Webserver-Software einen Fehler zu finden, der diese zum Aufhängen bringt, versucht man es sich irgendwie einfacher zu machen. Und so kommen wir zum DDoS :)

Wie funktioniert ein DDoS?

DDoS steht für Distributed Denial of Service [8] was soviel heißt wie Verteilte Dienstverweigerung. Der Unterschied zu DoS besteht darin, dass man keinen Aufwand tätigen muss um einen Fehler zu finden. Es wird zwar ein Dienst zum Stillstand gebracht, aber nicht durch gezielte Provokation eines Fehlers, sondern durch simple Überforderung des Dienstes.

Nehmen wir als Beispiel wieder einen Webserver. Anstatt eine gezielte Anfrage auf den Webserver zu starten, die diesen dann aus dem Konzept bringt, überfluten wir diesen Webserver einfach mit ganz normalen legitimen Anfragen. Wir schicken aber derart viele Anfragen an den Webserver, dass er einfach nicht mehr hinterherkommt. Jedoch wird man alleine mit einem Computer und einer Internetverbindung nicht so viele Anfragen schicken können um einen Webserver zum Absturz zu bringen. Dazu braucht man schon mehrere Sympathisanten, oder meist hat man ein sogenanntes Zombie-Netzwerk, oder Botnet, mit dem man hunderte, oder besser tausende, von fremden Computern diese Anfragenüberflutung starten kann (natürlich ohne, dass die Besitzer der Computer im Zombie-Netzwerk jemals merken würden, dass von ihrem Rechner aus ein Angriff gestartet wurde oder sie überhaupt Teil eines Zombie-Netzwerkes sind).



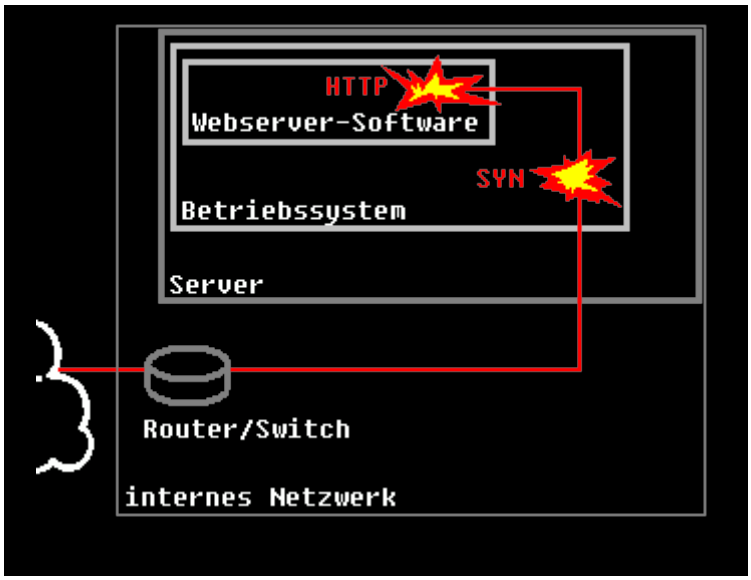
Der Angreifer muss also vorher ganz viele Computer mit einem Programm infizieren, das im Hintergrund auf Befehle wartet, um dann ganz viele Anfragen auf einen Server zu starten. So entsteht eine ganze Armee von Zombies, die dann gemeinsam auf einen einzigen Rechner hämmern, bis er in die Knie geht. Deswegen auch Distributed Denial of Service, Verteilte Dienstverweigerung.

Ein gewöhnlicher Webserver kann paar hundert Anfragen pro Sekunde verkraften, bevor die Performance darunter leidet. Jedoch kollabieren die meisten Webserver sofort bei 5000 oder 6000 Anfragen pro Sekunde, gnadenlos. Was geschieht in dem Moment beim Webserver? Die meisten Leute denken am Anfang, dass einfach die Bandbreite der Internetverbindung vollkommen aufgefressen wurde und der Server deswegen nicht mehr erreichbar ist. Dies ist heute aber seltener der Fall. Was üblicherweise bei einem DDoS den Server zum Sturz bringt ist die Systemauslastung, sei es die CPU durch rechenintensive Anfragen (vor kurzem haben paar dunkle Gestalten eine Art SSL-DDoS ausprobiert [9]; nette Idee, da durch Kryptographie sehr rechneintensiv) oder einfach Überlastung des Arbeitsspeichers; schließlich besetzt jede aufgebaute Verbindung auch ein Stückchen Speicher auf dem Server.

Also ist der resultierende Kollaps ein Effekt, der im Betriebssystem stattfindet. Und zwar im Code des Betriebssystems, durch Funktionen des Betriebssystems

(Sockets, oder die Umlagerung von Arbeitsspeicher auf die Festplatte, das bremst den Rechner extrem...). Das ist eine sehr wichtige Information, wenn man sich ein Bild davon machen will, wie man einen solchen Angriff abwehren kann.

Unterschiedliche Angriffsarten des DDoS

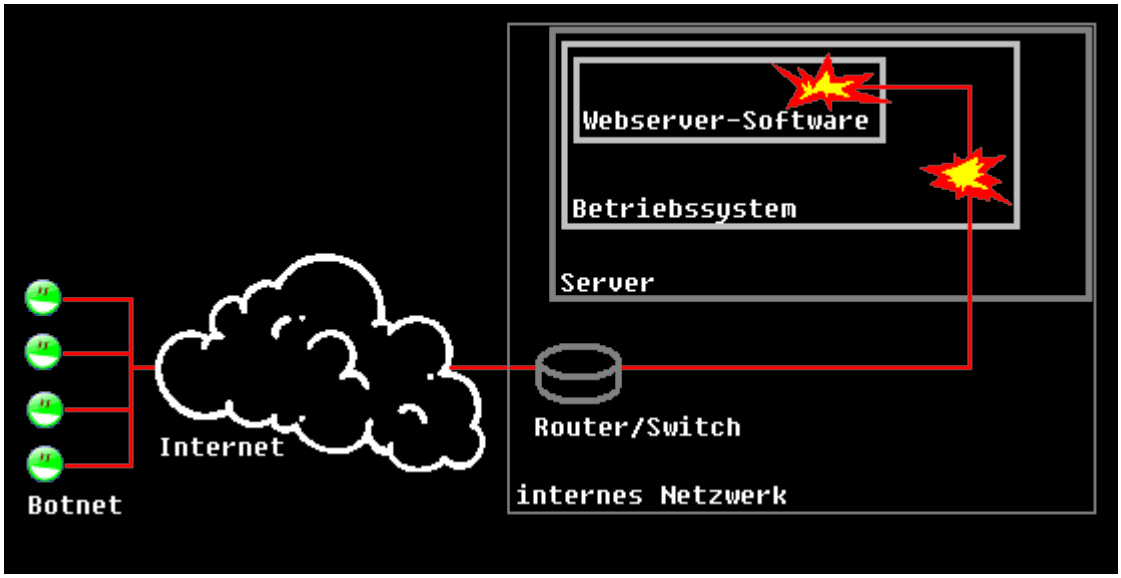


Es gibt aber auch unterschiedliche Arten des DDoS. Ich selber möchte hier nicht besonders drauf eingehen, da ich es persönlich total langweilig finde alle zu erklären und man dadurch nicht wirklich eine Erleuchtung erfährt. Ich gehe nur mal kurz auf 2 Sorten ein um vorzuführen, worin sich die Arten des DDoS unterscheiden; zum einen ein HTTP-DDoS, bei dem ganz viele Verbindungen mit einem Opfer-Server aufgebaut werden und dann so viele HTTP-Anfragen (Webseiten-Dateien anfordern) wie möglich gesendet werden um die Webserver-Software zu belasten (geht auch mit jedem anderen Protokoll); zum anderen ist der SYN-DDoS erwähnenswert, bei dem einfach nur sehr viele Verbindungen mit dem Server aufgebaut werden, bis er die ganzen Verbindungen nicht mehr managen kann. Interessant beim SYN-DDoS ist, dass die Webserver-Software davon garnichts mitbekommt und alle dort implementierten Sicherheitsmechanismen

vollkommen nutzlos sind. Es werden also andere Ebenen im OSI-Modell angegriffen. Wer den Satz verstanden hat weiß wovon ich rede, alle anderen müssen es nicht unbedingt verstehen; es wäre jetzt zuviel um es komplett auszubreiten :D

Abwehr von DDoS-Attacken

Wenn man sich per Google über die Abwehr von DDoS-Angriffen informiert, findet man meist in Foren den Tipp, die Angreifer IP's zu blocken. Jede IP baut gewöhnlich mehrere Verbindungen auf und schließt diese nicht mehr, so kann man sehen (z.B. per "netstat" unter Windows und Linux), wer Angreifer ist und wer nicht. Dann trägt man diese einfach unter der Deny-Regel in die root-.htaccess des Webverzeichnisses ein, und so weiter, bla bla :) Nur leider ist dieser Rat nur für die Leute interessant, die es bereits versäumt haben und unter einen nervigen Beschuss leiden. Der effektivste Schutz vor DDoS-Attacken beginnt noch zeitlich vor der DDoS-Attacke selber. Ist man unvorbereitet unter Beschuss, kann man nur eines tun, und zwar den Server vom Traffic komplett trennen. Entweder indem man den Stecker zieht (vorausgesetzt man hat physikalischen Zugriff) oder durch einen Router, der noch vor dem Server ist, mit dem man den gesamten Traffic an einen nicht existierenden Server schickt, also in ein Null-Interface, auch Blackholing [10] genannt. Dies kann man z.B. erreichen, indem man dem Server einfach eine neue IP verpasst und den DNS-Server drüber informiert. Die angreifenden Rechner sind auf die alte IP fixiert, während neue Besucher durch den DNS-Server die neue IP bekommen :) Dann leidet der Server nicht mehr unter dem Beschuss und wir können uns in Ruhe auf den nächsten Angriff vorbereiten.



Okay :) Zunächst einmal müssen wir uns ein Bild davon machen, was eine gute und was eine schlechte Abwehr ist. Hmm. Eine gute Abwehr ist effizient. :) Ein DDoS wird durch Überlastung hervorgerufen, also, je mehr unser Server und unser Netzwerk eine bösartige Anfrage bearbeiten, desto ineffizienter ist unsere Abwehr. Das heißt: Je früher ein Angriff erkannt und abgewehrt wird, desto weniger wird unser System durch diesen belastet und desto besser ist unsere Abwehr. Gut, dann gehen wir mal alle Abwehrmöglichkeiten durch und schauen welche gut ist und welche nicht :)

1) PHP

Nehmen wir mal an, wir haben uns eine Homepage in PHP programmiert. Und in dieser PHP-Datei haben wir eine Funktion implementiert, die überprüft wie oft pro Sekunde eine IP eine Anfrage startet. Wenn es zu oft geschieht, dann wird diese Anfrage einfach verworfen (per Funktion "die()"; einfach bääm -> weg!). Wie effizient ist diese Abwehr? :) Also, der Angreifer schickt den Befehl zum Angriff und es kommen über's Internet paar hundert Anfragen pro Sekunde mit Lichtgeschwindigkeit auf den Server zugeflogen. Die Anfragen betreten das interne

Netzwerk, in dem der Server steht und werden durch die Router in Richtung Server geleitet. Das Betriebssystem des Servers nimmt die Anfragen an, antwortet auf jede einzelne Anfrage und baut jeweils eine Verbindung auf. Dann wird der Webserver-Software vom Betriebssystem pro Anfrage bescheidgesagt, dass da jemand irgendwas will und die Webserver-Software bearbeitet nochmals jede einzelne Anfrage. Für jede einzelne Verbindung ruft die Webserver-Software jetzt die PHP-Datei auf, führt den Code aus und erst dann greift die Schutzfunktion, die merkt, dass da irgendwer wie ein Behinderter auf den Server einhämmert. Erst dann, wenn bei jeder einzelnen Anfrage die PHP-Datei bis zur Schutzfunktion ausgeführt wird, wird die Arbeit abgebrochen, der Speicher wird wieder freigegeben und die Verbindung wird verworfen.

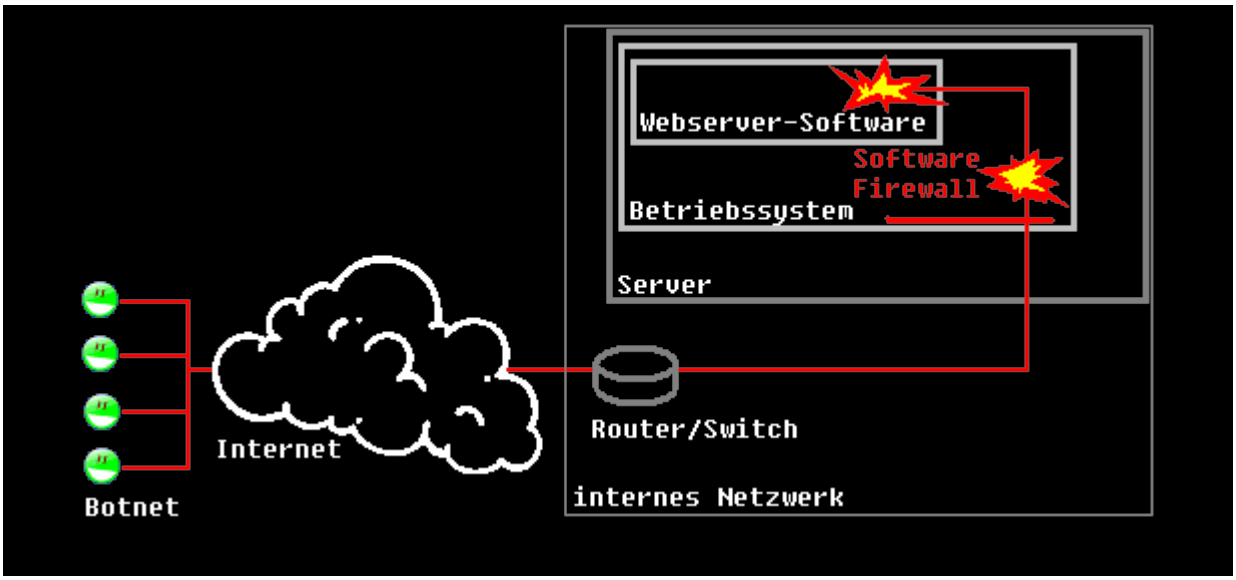
Ist das effizient? Also man muss zunächst vermerken, dass es immerhin ein funktionierender Schutz gegen HTTP-DDoS ist. Der Angriff wird erfolgreich bemerkt und die Ressourcen werden wieder freigegeben, womit man dem DDoS-Effekt entgegenwirkt. Nur leider würde ich die Abwehr als nicht besonders effizient bezeichnen, da tausende Schritte unternommen werden, bevor die Abwehr greift. Besonderer Knackpunkt ist das Betriebssystem; der Angriff wird erst erkannt, nachdem das Betriebssystem sich damit beschäftigt hat. Also ein motivierter Angriff von einem Kerlchen, der ganz genau weiß was er da tut, wird den Server auf jeden Fall in einem Aschenbecher hinterlassen...

Es gibt auch Module für Apache (eine beliebte Webserver-Software), die, wenn man sich damit auskennt, auch enorm mehr bringen als garkein Schutz. Solche Plugins wären "mod_dosevasive", speziell gegen HTTP-DDoS und Brute-Force-Attacken, und "mod_security", nützlich um Anfragen nach Kriterien zu filtern, bevor sie großartig verarbeitet werden. Für Normalsterbliche reichen diese dann auch meistens schon, da sie eher unwahrscheinlicher riesigen Angriffen ausgesetzt werden, sondern eher kleinen Gruschel-Angriffen :) Also das wäre schonmal die erste gute Lösung, wenn man den Server unter eigener Kontrolle hat, um die

Module zu installieren. Aber dennoch ist schon eine Menge Energie aufgefressen, bis die Module eingreifen. Und man muss bemerken, dass damit nur HTTP-DDoS-Angriffe (zielt auf die Webserver-Software) abgewehrt werden können ;) Ein SYN-DDoS (zielt auf das Betriebssystem, also noch vor der Webserver-Software) bleibt trotz dieser Sicherheitsvorkehrungen ungehindert. Große Firmen geben sich damit nicht zufrieden. Wir wollen es effizienter, also früher...

2) Software-Firewalls

Wie wäre es denn mit Software Firewalls? Also, Software-Firewalls hängen sich ins Betriebssystem ein, damit sind sie schonmal genau auf der Ebene, auf der das Problem besteht. Und Software-Firewalls sind meistens dazu konzipiert, solche Probleme zu lösen. In der Praxis zeigen sie sich auch garnicht mal so schlecht. Viele Leute schwören darauf, zum Teil, weil ihnen keine andere Wahl bleibt, da sie keinen physikalischen Zugriff auf die Hardware haben; zum anderen Teil, weil es simply die beste technische Gratis-Option ist :)



Also, was machen Software-Firewalls? Grundsätzlich kann jede Firewall Verbindungen blockieren, oder gestatten. Das größte Problem von Firewalls bei DDoS-Angriffen ist, dass sie "bösen" Traffic von "guten" Traffic unterscheiden müssen um dann zu wissen, welche Verbindungsversuche blockiert und welche zugelassen werden sollen. Bleiben wir bei dem Beispiel des Webservers. Ein Computer sendet eine Verbindungsanfrage auf den Webserver (SYN auf Port 80); macht er das nun, weil er einfach nur eine Webseite aufrufen will, oder weil er Teil einer DDoS-Attacke ist? Woran soll man das erkennen? Dabei zeigen sich viele Hürden, z.B. gibt es zum einen die sogenannten persistenten HTTP-Verbindungen [11] (auch bekannt unter Keep-Alive oder Pipelining), bei denen nur eine Verbindung aufgebaut wird worüber dann alle Anfragen durchgeführt werden (eine Webseite beinhaltet ja mehrere Elemente wie Bilder, Stylesheet-Dateien usw. die einzeln angefragt werden müssen). Es besteht also nur eine Verbindung. Dann gibt es aber noch eine weiterhin oft verwendete Vorgängerversion, die nicht-persistente HTTP-Verbindung, bei der für jedes einzelne Element eine eigene Verbindung aufgebaut wird. Da bestehen also mehrere (bei manchen Webseiten sind es hunderte... Web 2.0 ftw -.-) Verbindungen, was den meisten DDoS-Attacken vom technischem Aufbau ähnelt, jedoch keine ist.

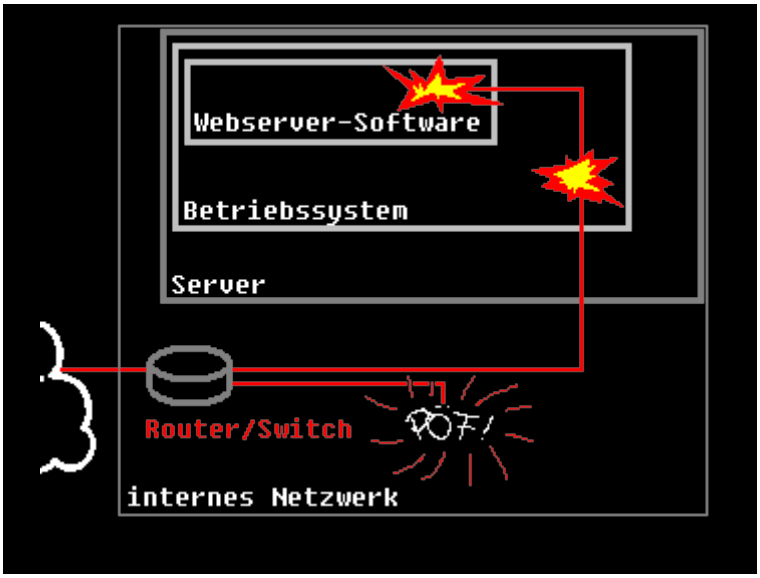
Viele sehen das als das aktuelle Hauptptoblem bei der Abwehr von DDoS-Attacken. Man kann halt nicht die Intention eines Pakets sehen. Wegen diesem Problem hat ein Forscher mal als Aprilscherz ein RFC geschrieben, bei dem jeder Hacker bitte ein "Evil-Bit" in seinen bösgemeinten Paketen setzen soll, damit diese sofort erkannt und abgewehrt werden können :D Aber ist klar, dass sich keiner daran hält. Es gibt einzelne Heuristiken, wie man guten und schlechten Traffic voneinander trennt, aber diese sind eben nur Heuristiken, also nicht perfekt. So werden z.B. bei einer DDoS-Attacke die Verbindungen vom Angreifer nicht mehr geschlossen, da diese ja den Server belasten sollen. Das wäre schonmal ein

Hinweis. Oder wenn von einem Rechner viel mehr Verbindungen gefordert werden, als es gewöhnlich der Fall ist. Wie gesagt sind Heuristiken nicht perfekt und können umgangen werden, indem der Angreifer ein wenig Aufmerksamkeit und Feinschliff in den Angriff steckt. Mal wieder, wie immer in Cybersecurity, befinden wir uns hier in einem Wettrüsten :)

Aber hey, immerhin bewegen wir uns jetzt schon auf hohem Niveau; eine spezialisierte Software-Firewall ist also schon eine sehr fortschrittliche Abwehr, wie sie auch von vielen Unternehmen verwendet wird. In den letzten Wochen wurden sogar schon ganze Firmen aufgekauft, die sich mit DDoS-Abwehr auf Firewall-Ebene beschäftigen, also ein Markt mit viel Potential.

3) Hardware

Gehen wir weiter :) Jetzt kommt die Hardware :) Dabei geht es nicht um die Hardware am Server, sondern die Hardware vor dem Server ;) Bevor die Pakete des Angriffs den Server erreichen, durchlaufen sie Knotenpunkte im Internet, sowie im lokalen Netzwerk. Dabei handelt es sich um Switches und Router. Manchmal kommt es vor, dass auch solch ein Knotenpunkt zusammenbricht, besonders oft, wenn es sich dabei um ein Billigrouter beim Heimanwender handelt. Aber ob der Router zusammenbricht oder der Server, in beiden Fällen kann man professionelle Router einsetzen, die DDoS-Angriffe effektiv abwehren können. Vorteil ist, dass es eine der besten technischen Lösungen ist, die es zur Zeit gibt. Nachteil ist, dass es was kostet :) Und außerdem hat nicht jeder die Macht um solch ein schönes Stück vor dem Server installieren zu können, weil man dazu ja das Netzwerk auseinandernehmen muss. Aber wenn man das macht, dann gehört man wohl zu den international bestgeschützten Computerfreaks :D



Wie schützt ein solcher Router/Switch? Essentiell hier ist, dass die Pakete bearbeitet werden, noch bevor der Server irgendwas mitbekommt. Hier kann man zum Einen natürlich auch die selbe Schiene fahren wie bei der Softwarelösung auf dem Server: wir können eine Firewall installieren bzw. die verwenden, die bereits drauf ist. Oder etwas ähnliches ACL-mäßiges (ACL steht für Access control list [12]). So fällt die Rechenzeit am Server vollkommen weg, da der Router dies ja jetzt übernimmt. Wer denkt, dass das die einzige Erhöhung der Messlatte ist, der täuscht sich. Denn diese verfluchten Dinger sprühen vor Magie :) Zum Einen besteht immernoch das Problem des DDoS-Erkennens, wenn man aber weiterschaut, dann bemerkt man, dass man einen gewaltigen Vorteil durch die vorgesetzte Hardware ziehen kann.

Fangen wir mit einfachen Kunststücken an, die wir mit einem Router durchführen können :) Die meisten Router/Switches sind befähigt die Rate zu limitieren, wir bedienen uns hier also ganz einfachen Routing-Mitteln, um den Server zu entlasten. Das sogenannte Rate limiting [13] ist dazu da um einzelne Pakete zu managen. So kann es z.B. exzessive Pakete verwerfen, Pakete in eine Warteschleife stecken und verschiedene Arten der Staukontrolle ausführen (auf Feedback des internen Netzwerks reagieren, intelligente Fairness-Steuerung und

so'n cooles Zeugs).

Auch interessant ist das Traffic-Shaping [14], insbesondere dabei die Überlauf-Kondition mit dem bekannten Random early detection-Algorithmus [15]. Wobei dieser Algorithmus in unserem Szenario wohl eher zu den Primitivlingen gehören dürfte, da er nicht besonders "schön" ist :) Noch ein interessanter Teil der Überlauf-Kondition wäre bei uns selbstreduzierende Quellen, bei denen bestimmte Dateien nicht mit voller Geschwindigkeit übetragen werden, sondern nur mit einer ausreichenden Geschwindigkeit. Wie bei Youtube z.B., man kann an der Videoleiste sehen, dass man die Videos nie mit volle Pulle saugt, sondern meistens nur ein bissl schneller als die Videos abgespielt werden :)

Aber eine der wirkungsvollsten Methoden bei Switches und Routern ist das sogenannte Delayed binding [16].

> "Some application switches and routers delay binding the client session to the server until the proper handshakes are complete so as to prevent Denial of Service attacks."
>
> *-Wikipedia*

Dabei handelt es sich um die Verschiebung der Verbindung zwischen dem Client und dem Server, um ausreichende Informationen zu erhalten mit denen dann eine Routing-Entscheidung getroffen wird. Total genial. Also der Client sendet eine Anfrage (SYN), die Anfrage wird vom Router nicht an den Server weitergeleitet, sondern es wird gewartet, bis der Client auf die Antwort des Routers (SYN ACK) eine Bestätigung (ACK) sendet. Macht der Client währenddessen Quatsch, reagiert er z.B. nach der ersten Anfrage (SYN) nicht, dann wird die Verbindung komplett

verworfen. Verhält der Client sich "gut", so wird die Verbindung mit dem Server "verschmolzen"; dann erst besteht also die echte Client-Server-Verbindung. Und der Server kriegt von dem ganzen Spektakel nicht mal was mit :D SYN-DDoS adieu :)

4) Verteiltes Hosting

Okay, jetzt haben wir das komplette interne Netzwerk abgeklappert. Die meisten denken jetzt, dass man am Ende der Möglichkeiten angekommen ist, aber das stimmt nicht. Es geht noch viel weiter... Was ist denn mit der kompletten Infrastruktur der Dienste, kann man da nicht etwas machen? Wenn man ein spezialisiertes Unternehmen nach DDoS-Rat fragt, dann empfehlen sie einem unbedingt die Dienste zu verteilen. Das ist ein total einfacher Trick, den jeder anwenden kann und selbst mich schon einmal gerettet hat. Wenn wir z.B. einen Online-Shop eröffnen, dann können wir ein Hoster für die Webseite mieten und einen komplett anderen Hoster für den E-Mail-Verkehr :) Das kostet nicht viel und man verliert wenigstens nicht komplett den wichtigen Kontakt zu den Kunden. Das geht auch mit anderen Diensten; grundlegend verteilt man Webserver (HTTP), E-Mail-Server (POP3, SMTP) und Download-Server (FTP, vllt. auch HTTP).

Wie gesagt, mir hat der Trick schon einige graue Haare gerettet, da ich auf einem Gratishoster eine Webseite hatte und auf einem ganz anderen Gratishoster die Dateien, die man von meiner Webseite aus downloaden konnte. Und zu der Zeit lief irgend ein Spinner herum, der bei Gratishostern einen Mega-Traffic durch Downloaddateien verursachte, dem eine Löschung durch den Gratishoster folgte, weil die natürlich keinen Bock auf zu viel Traffic hatten. Aber dank der Verteilung waren es bei mir nur die Dateien, die paar Tage fehlten, bis der Hoster sich eingekriegt hatte; die Homepage war immer lückenlos verfügbar :) Ein anderer Kumpel hatte ein eigenes Forum mit Membern verloren und war so verzweifelt,

dass... nun ja... hmmh... dass ich seitdem eigentlich nie wieder was von ihm gehört hab :D

Es gibt aber auch Hoster, die so einen ähnlichen Aufbau von selbst zur Verfügung stellen. Die hartnäckigsten Geschöpfe, die ich gesehen hab, waren sogenannte Bulletproof-Hoster, die in mehreren Dritte-Welt-Ländern ihre Server stehen hatten. Und zwar in Dritte-Welt-Ländern, in denen es noch nichtmal eine gesetzliche Grundlage gibt, diese Server vom Netz nehmen zu können... Deren technische Lösung zum Schutz vor DDoS-Attacken waren mehrere komplett voneinander getrennte Serverzentren; ist ein Server unter Beschuss, dann wechselt man einfach zum nächsten Server in einem ganz anderen Zentrum, in einem ganz anderen Land. Natürlich auch mit ganz anderem DNS-Server. Der Angreifer läuft der Homepage dann in der ganzen Welt hinterher. Aber solche Hoster kosten dann auch unverschämt viel ;)

5) Gesetze

Können wir noch früher was gegen DDoS-Angriffe unternehmen? Immer doch, aber dann kommen wir an einem Punkt an, an dem wir unsere technischen Mittel kurzzeitig verlassen und in die Struktur des Angreifers eindringen. Wir kommen zum Botnet des Angreifers selber; in Zukunft gibt es Hoffnung, dass da etwas gemacht werden kann. Und zwar auf gesetzlicher Ebene. Ein weiser und alter Mann hat mir mal gesagt, dass Gesetze nur für die Probleme geschrieben werden, die technisch nicht lösbar sind. Deswegen würde ich darauf (besonders im Cyberraum) nicht unbedingt schwören, aber dennoch ist es erwähnenswert, dass Politiker unterschiedlicher Länder sich bemühen Gesetze durchzubringen, die den "good guys" mehr technische Mittel zur Verfügung stellen. So war ein Richter kürzlich dazu befähigt, Microsoft 276 IP's zu schenken, damit die dann ein Botnet aus dem Verkehr ziehen können [17]. Es ist fraglich, inwiefern diese Gesetze

missbraucht werden könnten, oder wie es sich in Zukunft weiterentwickelt, aber dieses Beispiel ist schonmal ziemlich positiv.

6) Teams/Organisationen

Desweiteren gibt es auch noch Teams und Organisationen, die sich mit dem Thema Botnet und DDoS auseinandersetzen. So gut wie keiner kennt sie, aber es gibt sie wirklich :) Tendenz steigend; wie gesagt, dieses Thema hat viel Potenzial. Darunter fallen zum einen Gruppen, die DDoS-Angriffe in Art, Stärke, Ursprung und weiterem dokumentieren; Gruppen, die jeden Tag Botnets detailliert erfassen und dokumentieren [18]; und natürlich auch Gruppen (meistens AntiVirus-Firmen o.ä., weil sie dadurch Werbung machen...), die die Botnets über gehackte CnC-Server (Command'n'Control-Server, damit werden Befehle an das Botnet verteilt) und anderen Mitteln schwächen, zerschlagen, oder komplett die Kontrolle übernehmen (Beispiel Microsoft). Aber nicht immer ist man da so erfolgreich, vor paar Wochen ist ein Botnet aufgetaucht, das eine vernünftige asymmetrische Verschlüsselung mit Signatur und allem drum und dran verwendet, da sind dann viele "good guys" mit dem Latein ziemlich schnell am Ende :) Ein Wettrüsten, mal wieder.

7) Motivation des Angreifers

Wie kann man ein DDoS NOCH FRÜHER verhindern? Ganz einfach :) Irgendwo sitzt doch ein Kerlchen am Rechner und sendet ein Angriffsbefehl :) Wenn man es grob nimmt, dann ist diese Person der wirkliche Ursprung des Angriffs. Was ist denn mit dem? Was geht in seinem Kopf vor? Wieso macht er das? Genau da können wir ansetzen. Die Antwort, warum jemand irgendwas macht, ist immer die selbe: Motivation. Der Typ hat irgendeine Motivation gefunden deinen Webserver anzugreifen. Und die müssen wir ihm wegnehmen.

Zu allererst gibt es den Grund, warum er es macht. Vielleicht ist er neidisch, weil du ein super Forum aufgebaut hast oder er will einfach dein Ego brechen, weil du eingebildet bist; du hast ihm was weggenommen oder Schaden angerichtet; du hast ihn verletzt, bist rassistisch oder redest schlecht über andere; oder vielleicht will er mit dem Angriff einfach nur das ersetzen, was er in der Hose nicht hat :) Woran es liegt, musst du am Besten wissen. Viele Angreifer hinterlassen auch eine Nachricht, sei es in Form einer tatsächlichen Nachricht, oder indem sie auf ein bestimmtes Dokument oder auf eine bestimmte Art und Weise einhämmern. Sei dir des Grundes immer bewusst.

Danach überlegt jeder Angreifer, ob er es tatsächlich machen soll, oder nicht. Das ist auch ein sehr wichtiger Punkt, an dem man ein Angriff verhindern kann. Er sucht gezielt nach Argumenten, um es nicht zu tun und wägt diese dann mit dem Grund es zu tun ab :) Also, wie kann man einen Angreifer schlagfeste Gegenargumente bringen? Wovor hat er denn am meisten Angst, falls er es tut? Dass die Polizei an seiner Haustür klopft und er im Knast landet, wegen einem lächerlichen DDoS! :) Also, wie wäre es mit einem LKA-Banner im Impressum, oder ein Hinweis über eine Kooperation mit dem BSI in der deutschen Anti Botnet Initiative [19].

Epilog

Der Fantasie sind keine Grenzen gesetzt. Und es ist ein Wettrüsten. Wie gesagt, so ist es überall in der Cybersecurity. Irgendwann werden auch die heftigsten Schutzmechanismen, die wir heute haben, als nutzlos dastehen, weil die Angreifer neue Techniken entwickelt haben. Das ist das Schlimme, aber auch das Geniale an Cybersecurity :) Jeder, der programmieren kann und ein wenig Ahnung von Betriebssystemen hat kann jetzt seine eigene technische Lösung zu dem Problem entwickeln. Der Fantasie sind keine Grenzen gesetzt... Und es ist ein Wettrüsten... Also lass dir was einfallen! ;)

Quellen:

[1] http://twitter.com/keksaDE/status/10998412266
[2] http://twitter.com/keksaDE/status/17939002002
[3] http://twitter.com/keksaDE/status/20459417163
[4] http://twitter.com/keksaDE/status/10236455530
[5] http://twitter.com/keksaDE/status/7527671126
[6] http://twitter.com/keksaDE/status/16793461416
[7] http://de.wikipedia.org/wiki/Denial_of_Service
[8] http://en.wikipedia.org/wiki/Denial-of-service_attack#Distributed_attack
[9] http://twitter.com/keksaDE/status/8390846382
[10] http://en.wikipedia.org/wiki/Black_hole_(networking)
[11] http://en.wikipedia.org/wiki/HTTP_persistent_connection
[12] http://en.wikipedia.org/wiki/Access_control_list
[13] http://en.wikipedia.org/wiki/Rate_limiting
[14] http://de.wikipedia.org/wiki/Traffic-Shaping
[15] http://en.wikipedia.org/wiki/Random_early_detection
[16] http://en.wikipedia.org/wiki/Delayed_binding
[17] http://twitter.com/keksaDE/status/24022415303
[18] http://twitter.com/keksaDE/status/24468389191
[19] http://twitter.com/keksaDE/status/23941892258