White Paper

The DDoS Threat Spectrum

Bolstered by favorable economics, today's global botnets are using distributed denial-of-service (DDoS) attacks to target firewalls, web services, and applications, often simultaneously. This DDoS threat spectrum includes conventional network attacks, HTTP and SSL floods, and an emerging wave of low-bandwidth threats, plus the new threat vectors likely to target emerging service platforms.

by David Holmes
Senior Technical Marketing Manager

Contents

Introduction

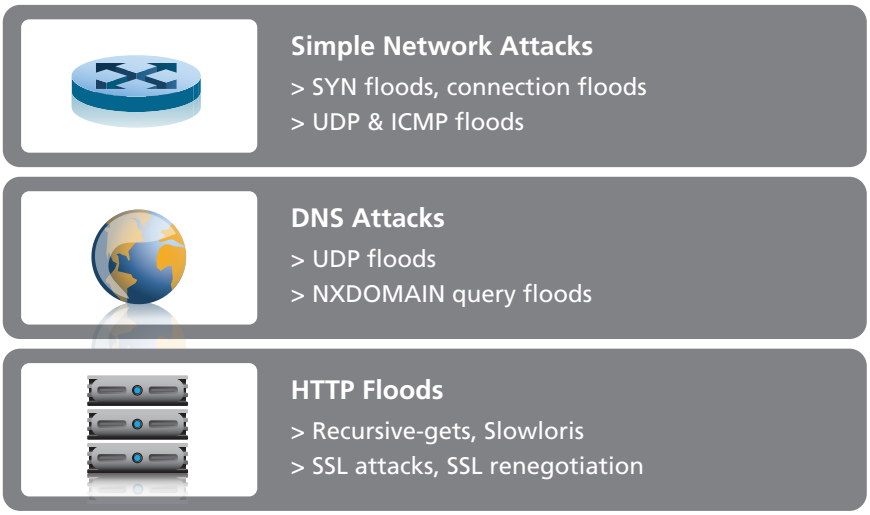
The world is becoming increasingly connected electronically, expanding markets and reducing the inefficiencies of doing business across borders. Services can be hosted anywhere and customers can be served from anywhere as the Third World catches up to the First World's broadband penetration. Emerging market territories often lack proper client control, however, and malware infection rates are high. When these malware clients are directed by centralized command-and-control servers, they become "botnets." The sheer number of client machines involved in botnets provides enormous load-generation capacity that can be rented cheaply by any party with an interest in disrupting the service of a competitor or political target. Today's global botnets are using distributed denial-of-service (DDoS) attacks to target firewalls, web services, and applications, often all at the same time.

Though DDoS attacks have been with us for decades, the scope, nature, and magnitude of the DDoS threat spectrum have evolved significantly over time.

The Evolution of DDoS Attack Targets

Early DDoS attacks used a limited group of computers (often a single network) to attack a single host or other small target. When commercial interests gained entry to the Internet in the 1990s, they presented a target-rich environment for any group with an axe to grind against a competitor or perceived commercial monopoly; Microsoft and the Recording Industry Association of America (RIAA) were frequent targets. Thus, DDoS attacks were perceived as being a problem primarily for "big players" or, in fact, for the Internet itself. In 2002 and 2007, coordinated DDoS attacks were launched against the 13 DNS root servers in an attempt to attack the Internet at its most vulnerable infrastructure. The 2002 attack was largely successful, but the 2007 attack failed (11 of 13 root servers stayed online), thanks to lessons learned from the 2002 attack. Commercial DDoS defense services were developed for deployment at the service provider level.

Today, smaller services and organizations are being targeted. The motivations behind the attacks are often commercial, political, or more often, simple extortion.



Simple Network Attack Effectiveness

Simple network attacks still work against undefended hosts. For example, a single Linux host running the world's most popular web server software, the Apache 2 server, fails under these simple attacks at very low packet rates.

Attack	Metric	Result
SYN flood	1500 syns per second	Denial-of-service
Conn flood	800 connections	Denial-of-service

Figure 1: Terminal metrics for a single Linux host with Apache 2 server

DDoS Attack Profiles

Early DDoS attack types were strictly low-level protocol attacks against Layers 3 and 4. Today, DDoS attacks come in three major categories, climbing the network stack from layer 3 to layer 7.

Simple Network Attacks

The most basic attacks in the DDoS threat spectrum are simple network attacks against the weakest link in the network chain. These attacks, called floods, harness a multitude of clients to send an overwhelming amount of network traffic at the desired target. Sometimes the target succumbs and sometimes a device in front of the target (such as a firewall) succumbs, but the effect is the same—legitimate traffic is denied service. By using multiple clients, the attacker can amplify the volume of the attack and also make it much more difficult to block, since client traffic can appear to come from all over the globe. The SYN flood and connection

flood (conn flood) typify these simplest distributed attacks, which are designed either to tie up stateful connection mechanisms of devices, such as hosts, that terminate layer 4, or to fill up flow tables for stateful devices that monitor connections, such as stateful firewalls or intrusion prevention systems (IPS).

Modern network attacks rarely fill or exceed the throughput capacity of the ingress pipes of the targets because they don't need to; stateful devices within the target data center typically fail long before the throughput limit is exceeded[1].

Attack	Target Vector	Description
SYN flood	Stateful flow tables	Fake TCP connection setup overflows tables in stateful devices
Conn flood	Stateful flow tables	Real, but empty, connection setup overflows tables in stateful devices
UDP flood	CPU, bandwidth	Floods server with UDP packets, can consume bandwidth and CPU, can also target DNS servers and VOIP servers
Ping flood	CPU	Floods of these control messages can overwhelm stateful devices
ICMP fragments	CPU, memory	Hosts allocate memory to hold fragments for reassembly and then run out of memory
Smurf attack	Bandwidth	Exploits misconfigured routers to amplify an ICMP flood by getting every device in the network to respond with an ICMP broadcast
Christmas tree	CPU	Packets with all flags set except SYN (to avoid SYN flood mitigation) consume more CPU than normal packets
SYN/ACK, ACK, & ACK/PUSH floods	CPU	SYN-ACK, ACK, or ACK/PUSH without first SYN cause host CPUs to spin, checking the flow tables for connections that aren't there
LAND	CPU	Identical source and target address IPs consume host CPU as they process these invalid addresses
Fake TCP	Stateful flow tables	TCP sessions that look real, but are only recordings of previous TCP sessions; enough can consume flow tables and avoid SYN flood detection
Teardrop	CPU	Sends a stream of IP fragments; meant to exploit an overlapping fragment problem present in some systems

Figure 2: Simple network attacks that can nonetheless be very effective

1 Network Infrastructure Security Report VI, Arbor Networks

These simple network attacks are still in use today, often in concert with the more advanced techniques of DNS attacks and HTTP floods.

DNS Attacks

The Domain Name System (DNS) translates name queries (e.g., www.example.com) into numerical addresses (e.g., 192.168.204.201). Nearly all clients rely on DNS queries to reach their intended services, making DNS the most critical—and public—of all services. When DNS is disrupted, all external data center services (not just a single application) are affected. This single point of total failure, along with the historically under-provisioned DNS infrastructure, especially within Internet and enterprise data centers, makes DNS a very tempting target for attackers. Even when attackers are not specifically targeting DNS, they often inadvertently do; if the attack clients are all querying for the IP of the target host before launching their floods, the result is an indirect attack against the DNS server that can often bring it down.

Because of the relatively simple, UDP-based DNS protocol, a DNS attack has two main characteristics:

- DNS attacks are easy to generate.
- DNS attacks are difficult to defend against.

Three specific classes of DNS attacks

UDP floods: The DNS packet protocol is based on UDP, and UDP floods are extremely easy for attackers to generate. When under attack from a UDP flood, the DNS server must spend CPU cycles to validate each UDP packet until it runs out of connection contexts or CPU, at which point the services either reboot or drop packets. The most common response is to reboot (often causing a reboot cycle until the attack ends). The second option, dropping packets, is little better, as many legitimate queries will be dropped.

Legitimate queries (NSQUERY): The hierarchical nature of the Domain Name System can require a DNS server to contact multiple other DNS servers to fully resolve a name; thus a single request from a client can result in four or five additional requests by the target server. This asymmetry between the client and the server is exploited during an NSQUERY DDoS attack, whereby clients can overload servers with these requests. A variation on this attack is a reflection/amplification DNS attack, whereby a series of misconfigured servers can be fooled into amplifying a flood of queries by sending their responses to a target victim's IP address. Such

exploitation accounted for one of the largest attacks in recorded history (over 100 Gb/s) in 2010[1].

Legitimate queries against non-existent hosts (NXDOMAIN): This is the most advanced form of attack against DNS services. Distributed attack clients send apparently legitimate queries to a DNS service, but each query is for a different host that does not exist anywhere, for example, urxeifl93829.com. The DNS service must then spend critical resources looking in its cache and zone database for the nonexistent host. Not finding a record, some DNS services will pass the attack request onward to the next service and wait for a response, tying up further resources. Even if the service survives the attack, it will ultimately replace all its valid cache entries with these invalid entries, further impacting performance for legitimate queries. These NXDOMAIN attacks are extremely difficult to defend against. In November 2011, a similar attack disabled many DNS servers around the world. The vulnerability remained until the Internet Systems Consortium could release a patch for BIND, the software on which many DNS servers are based.

The historic under-provisioning of DNS machinery is being corrected, but only slowly, and often in response to a DDoS attack. Until the system as a whole is strengthened, DNS attacks on this vulnerable target will continue to be a tempting method for attackers.

HTTP Attacks

Floods

Over 80 percent of modern DDoS attacks are HTTP floods[1]. Unlike most simple network attacks, which overwhelm computing resources with invalid packets, HTTP flood attacks look like real HTTP web requests. To conventional firewall technology, these requests are indistinguishable from normal traffic, so they are simply passed through to the web servers inside the data center. The thousands or millions of attacking clients overwhelm the web servers with a massive number of requests.

The two main variations of the HTTP flood attack differ in the requested content. The most common, basic attack merely repeats the same request over and over again. Clients that use this attack often do not bother to parse the response; they simply consume it and then resend the same request. Because they are the equivalent of a finger pressing a doorbell, these “dumb” attack clients are easy to program, but also easy to detect and filter.

The more advanced version of the HTTP flood attack is a recursive-get denial-of-service. Clients using this attack request the main application page, parse the response, and then recursively request every object at the site. These attacks are very difficult to detect and filter on a per-connection basis because they are requesting different, yet legitimate, objects. Recursive-get attack clients are quite intelligent and getting more sophisticated over time, resulting in an arms race between security services and recursive-get attackers.

Low-bandwidth HTTP denial of service attacks

An undefended modern web server is a surprisingly vulnerable target for very simple HTTP attacks such as the Slowloris script. Slowloris works by opening connections to a web server and then sending just enough data in an HTTP header (typically 5 bytes or so) every 299 seconds to keep the connections open, eventually filling up the web server's connection table. Because of its slow approach, it can be a devious attack, remaining under the radar of many traffic-spike attack detection mechanisms. Against a single, typical web server running Apache 2, Slowloris achieves denial-of-service with just 394 open connections[2].

Like Slowloris, the Slowpost attack client uses a slow, low-bandwidth approach. Instead of sending an HTTP header, it begins an HTTP POST command and then feeds in the payload of the POST data very, very slowly. Because the attack is so simple, it could infect an online Java-based game, for instance, with millions of users then becoming unwitting participants in an effective, difficult-to-trace, low-bandwidth DDoS attack.

A third low-bandwidth attack is the HashDos attack. In 2011, this extremely powerful DoS technique was shown to be effective against all major web server platforms, including ASP.NET, Apache, Ruby, PHP, Java, and Python[3]. The attack works by computing form variable names that will hash to the same value and then posting a request containing thousands of the colliding names. The web server's hash table becomes overwhelmed, and its CPU spends all its time managing the collisions. The security professionals exploring this attack demonstrated that a single client with a 30 Kbps connection (which literally could be a handset) could tie up an Intel i7 core for an hour. They extrapolated that a group of attackers with only a 1 Gbps connection could tie up 10,000 i7 cores indefinitely.

If a web server is terminating SSL connections, it can be vulnerable to the SSL renegotiation attack invented and popularized by the French group "The Hacker's

2 F5 testing of slowloris.py vs. HP ProLiant DL165 G6 with Apache 2.2.3

3 Denial of Service through Hash Table Multi-Collisions

Choice." This attack capitalizes on the SSL protocol's asymmetry between the client and server. Since the server must do an order of magnitude more cryptographic computation than the client to establish the session, a single SSL client can attack and overwhelm a web server with a CPU of the same class.

Attack	Target Vector	Description
Slowloris	Connection table	Slowly feeds HTTP headers to keep connections open
Slowpost	Connection table	Slowly POSTs data to keep connections open
HashDos	CPU	Overwhelms hash tables in back-end platforms
SSL renegotiation	CPU	Exploits asymmetry of cryptographic operations

Figure 3: Low bandwidth HTTP attacks

Rounding out the category of low-bandwidth attacks are simple HTTP requests that retrieve expensive URLs. For example, an attacker can use automated reconnaissance to retrieve metrics on download times and determine which URLs take the most time to fetch. These URLs can be then be distributed to a small number of attacking clients. Such attacks are very difficult to detect and mitigate, turning any weak points in an application into a new attack vector.

Political and Commercial Targets

Attacks are so ubiquitous today that many sites are constantly under some form of traffic attack, 24 hours a day, 365 days a year. These large sites defend and provision their resources accordingly. Smaller companies, which may not have the resources to routinely over-provision, must defend against attacks on a case by case basis.

The reasons for DDoS attacks vary, but currently the primary motivation is either political or financial. Since the mid-2000's, a counterculture movement has arisen to use DDoS attacks as a form of online protest. The most infamous of these groups calls itself Anonymous. Anonymous sprang from the counterculture image board 4chan, and its main rallying point is freedom of speech. Early Anonymous targets were The Church of Scientology (for its presumed efforts at stifling information about its practices), YouTube (for censorship), and the Australian government (again

for censorship). In recent years, Anonymous has made headlines by attacking governments and firms that it perceived were against the Wikileaks document-leaking service. Subsequently, they were involved in more high-profile attacks during the Arab Spring of 2011 and the Occupy Wall Street movement, with which they became closely associated. Occupy Wall Street participants and Anonymous share the same mask: the Warner Brothers version of Guy Fawkes, the seditionist who attempted to blow up the English Parliament in 1605.

Attacks against commercial properties are motivated by financial gain for the attackers and/or loss for the targets. Gaming sites and auction sites have time-specific windows where DDoS can prevent game actions or final bids, and this occurs frequently enough that some auction sites automatically cancel any auction where a DDoS has occurred.

Botnet Capacity Trends

The average botnet size peaked in 2004 at over 100,000 client machines. Now, however, there are many more botnets in a wide range of sizes. The average botnet size in 2011 shrank to 20,000, in part because more efficient, smaller botnets are still effective at DDoS but better at evading detection, analysis, and mitigation. Nonetheless, enormous botnets with 10 to 15 million client machines still exist. The massive Rustock and Cutwail botnets, for instance, were taken down in a simultaneous, globally coordinated campaign conducted by Interpol, Microsoft, and the University of Washington, with the help of other key sovereign enforcement agencies.

Though there have been some truly enormous DDoS attacks in recent years (including a documented 40 Gbps attack in 2010 and a 127 Gbps attack in 2009 [4]), most DDoS attacks are much smaller. Approximately 80 percent involve less than 1 Gbps[1] and short durations (typically hours, not days).

The Economics of Botnets

Modern botnet clients can be easily constructed by purchasing a software development kit (SDK), available on the Internet for $1,500 to $3,000, and then tuning the software to perform the malicious activity desired, whether spam, DDoS

4 The State of the Internet Report, Q3 2011, Akamai

5 An Inside Look at Botnets, Paul Barford and Vinod Yegneswaran, University of Wisconsin—Madison, 2006

6 The Evolution of Click Fraud, Erick Schonfeld, TechCrunch, October 2009

7 The Economics of Botnets, Yury Namestnikov, SecureList, July 2009

attacks, click fraud, adware, hostage, bank authentication credential theft, and so on. These clients can attack other, nearby computers, infect them and increase the size of the botnet. The individual who put together the botnet is known as the bot-herder. Once the botnet has achieved the desired size, the bot-herder can either use it to threaten or attack targets or simply rent it to interested parties wishing to threaten or launch an attack.

Aspect	Cost per Client or Email ($US)	Cost per 10,000 Clients ($US)
Per client acquisition[5]	$0.04 to $0.10	$400 to $1,000
DDOS attack (per hour)	$0.01 to $0.02	$100 to $200
DDOS extortion	$10,000 (common)	N/A
Spam emails	$0.005 to $0.015	$0.50 to $1.50
Click fraud[6]	$0.15 per client	$1,500
Adware[7]	$0.30 to $1.50 per install	$3,000 to $15,000

Figure 4: The economics of botnets

Cost of attack launch vs. incurred cost to target

It can be difficult to judge the exact size of a botnet from its advertising, because bot-herders have financial motives to make their botnets appear to be as large as possible. Still, botnets of 10,000 clients or more can be found for rent on underground software markets for a rate of $200 (US) per hour. Such a botnet can create a SYN flood exceeding 4 million packets per second or a sustained conn flood attack exceeding 4 million concurrent connections. To prove their capacity, these medium-sized botnets sometimes can be used for free for 3 minutes in a "try-before-you-buy" model. A botnet of this size, when launched against a competitor on a busy holiday shopping day, could cost that competitor $100,000 per hour. Larger botnets may rent for several thousand dollars per hour and are capable of attacking larger targets and causing larger losses.

Extortion

For the small- to medium-sized botnets, there is a more compelling way to make money—by threatening to launch an attack and then extorting blackmail from the intended target. According to researchers at CloudFlare, many of these DDoS extortion acts originate in Russia and China. A typical threat may begin like this:

```
"Dear <target>, we are a security firm in Shanxi province
 and we have received word that your company's website
```

> may be attacked one week from today. We have some influence with the attackers, and we believe that we may be able to convince them not to attack for $10,000 US. Please advise if you would like us to proceed. Instructions on how to wire funds are following."

The letter is sent by the attacker, and if the monies are not paid, the attack is executed. If the blackmail is paid, the target may receive a thank you letter:

> "Dear <target>, congratulations. The payment we received was used to cancel the attack. The individuals who were planning the attack enjoy doing business with you and would like to offer you a discount of 20 percent should you choose to hire them to attack someone on your behalf…"

Botnet	Estimated Size	DDoS Attack Types
Rustock	2.4 million	Conn flood
Cutwail	2.0 million	Fake SSL flood
akbot	1.3 million	DDOS (unknown type)
TFN2K	Unknown	SYN flood, UDP flood, ICMP flood, Smurf attack
LOIC	15,000	HTTP flood, SYN flood, UDP flood
HOIC	Unknown	HTTP flood
RefRef	Unknown	DoS via SQL server vulnerability

Figure 5: Some of the world's high-profile botnets

DDoS Attack Technologies on the Horizon

Historically, the majority of botnet clients have been relatively high performance utilities written in the low-level C/C++ language. This may be changing, as several new technologies will open opportunities to create different attack clients.

Java Applets can be programmed to launch an individual attack when a browser visits the hosting website and runs the applet. The new HTML5 specification brings

multi-threading to JavaScript's Web Workers, enabling a website to launch a multi-threaded attack from each browser that visits the hosting site. Mobile devices have the dangerous combination of ubiquity, muscular computational power, network access, and consumer-level security that may result in the mobile ecosystem becoming a jungle of botnets.

Attackers will use network and HTTP floods to activate DDoS mitigation defenses and then use CPU-vector attacks to bring down those defenses and cause a denial-of-service-via-mitigation failure. Similarly, an intriguing set of attacks known as ReDos target the use of regular expression engines in IPS and firewall defenses. A two-phase attack like this has already been used against a major online payment processing system[8] in 2010.

As new application frameworks continue to be developed, they bring with them additional layer 7 DDoS vectors. For instance, the Node.js language is seeing deployment among early adopters as a new server-side Javascript application server. But a novel attack vector that was quickly discovered persuaded Node.js to execute an infinite loop, allowing a single mobile phone to keep a site down indefinitely.

DDoS on the Horizon
Java Applets
HTML 5 Threads
Mobile Handset Bots
ReDos
Multi-Stage DDoS
IPv6
WebSockets

Conclusion

The DDoS threat spectrum has evolved from simple network attacks to DNS amplification attacks and finally application layer attacks. DDoS attacks are on the rise in 2012 as global participants increasingly engage in "hactivism" over digital rights and the changing landscape of intellectual property.

While HTTP floods currently account for over 80 percent of today's attacks, expect simple network attacks to make a resurgence as they are combined with HTTP floods into sophisticated multi-stage attacks that achieve denial-of-service. As new technological frontiers open, expect to see more distributed, low-bandwidth DDoS attacks.The DDoS threat spectrum will continue to evolve as attackers bend those new technologies to the political and commercial conflicts that will always be part of the human condition.

8 What We Learned from Anonymous/AntiSec, SC Magazine

F5 Networks, Inc. 401 Elliott Avenue West, Seattle, WA 98119 888-882-4447 www.f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com