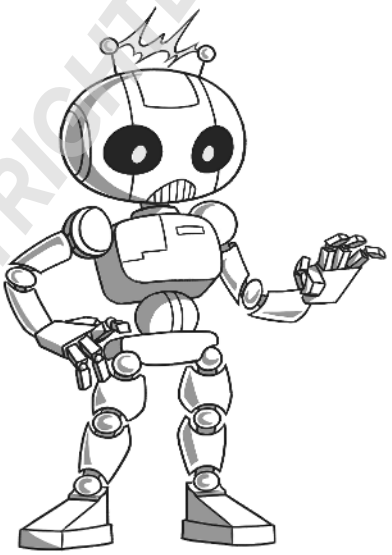
1 A Short History of the Internet

Our journey begins with a description of the evolution of the Internet and the emergence of a new type of fraud and abuse that leverages botnets.



From ARPANET to the Metaverse

The Internet is so ingrained in our day-to-day life that it seems as though it's always been around. However, the Internet is a relatively new invention—and it keeps evolving. The precursor of the Internet, called the Advanced Research Projects Agency Network (ARPANET), was invented in the 1960s, in the middle of the Cold War, to ensure continuity of availability of the network and computing resources even after a portion of it is removed or destroyed. Government researchers could also share information quickly without requiring them to travel to another location. ARPANET was a closed system using proprietary protocols, and only explicitly authorized people could access it. The idea of a network where one could share information and computing resources sparked the interest of academics, and the need for standardized communication protocols arose. Various communication protocols, including Transmission Control Protocol/Internet Protocol (TCP/IP), Hypertext Transfer Protocol (HTTP), and Domain Name System (DNS), were developed in the 1980s, marking the birth of the Internet as we know it today. TCP/IP defines how information is exchanged between two machines on the Internet. DNS, the equivalent of the phone book, transforms a hostname into the IP address where the service can be found. HTTP defines how web content is to be requested and shared between the browser running on the client and the web server. These protocols enable communication between systems from different vendors and connect them. Secure Sockets Layer (SSL) and, later, the Transport Layer Security (TLS) protocols add a layer of security and safety to the HTTP protocol. Newer languages like HyperText Markup Language (HTML) and JavaScript were invented to help develop websites and make content available in a structured and dynamic way.

Initially, the Internet was reserved for the technical elite who knew the protocols, had the right equipment, understood how to connect to the network, and knew how to query it to retrieve information. The development of web browser software in the 1990s, like Netscape and Internet Explorer, compatible with all of the aforementioned protocols and languages, made the Internet accessible to all. Web search engines such as AltaVista, Yahoo! Search, and Google Search also made it easier to query and find information online. When I was a college student in the 1990s, the Internet was in its infancy. All you could do was visit various websites to find information. Most news outlets would have a website with the latest sports results or events of the world. Major retailers started to create websites to showcase their products, and airlines advertised their flights. But e-commerce wasn't quite a thing just yet, and we still had to go to a brick-and-mortar shop to buy products or to a travel agency to book a flight.

Rapid technological advancement, including faster modems and expansion of the network infrastructure, supported the growth of the Internet. As the Internet grew more popular, investors started pouring money into a multitude of Internet companies with the hope of turning a

profit one day. These companies' valuations, which were purely based on speculative future earnings and profits, surged in the late 1990s with record-breaking initial public offerings (IPOs) that saw their stock triple within a day. These events fueled an irrational investment strategy from venture capital firms to companies that sometimes did not have a strong business plan or viable products for fear of missing out. In March 2000, large stock sell orders from leading high-tech companies like Cisco or Dell caused a panic sale and marked the beginning of the decline of the "Internet bubble." Investors became more rational, and capital became harder to find for startups that were not profitable. Many of these cash-strapped startups disappeared rapidly. Companies that reorganized and refocused their effort on developing valuable services and products survived, and some, like Akamai Technologies, Google, Amazon, and Apple, became very successful and key players in the development of the Internet.

When the bubble burst, it felt like a setback, but eventually, the Internet not only survived but started to thrive. As the quality of the Internet network improved, so did the content. The classic dial-up modem connection that had a maximum speed of 56Kbps was soon replaced by a more advanced and reliable network and telecom infrastructure. Integrated Services Digital Network (ISDN) offered speeds of up to 128Kbps, more than double what a dial-up modem could achieve. At the turn of the century, digital subscriber lines (DSLs), which offered high-speed Internet, became more widely available through conventional telephone networks, cable, and fiber optics. Today, Internet service providers offer connections as fast as 10Gbps, which is 178,571 times faster than the fastest dial-up modem. Advancements in mobile telecommunication and the emergence of smartphones meant that consumers could access the Internet from anywhere at any time for the first time. Mobile network expansion also helped expand the reach of the Internet to rural areas. Today, one can even browse the Internet while on a plane or cruising on the ocean, thanks to satellite networks.

As more and more people were drawn to the Internet, the distribution of rich content became a real issue. The networks that carried the Internet traffic did not always have the adequate capacity to handle the demand. Telecom operators would do their best to route the traffic, but frequent congestion and often long distances between the client and the server led to slow page load or stream buffering for Internet users, especially during popular events. Content Delivery Network (CDN) companies like Akamai Technologies, Fastly, and Cloudflare, to name a few, became the backbone of the Internet. CDNs helped fix the problem by avoiding transporting the content long distances and bringing it closer to the user. CDNs helped make the Internet faster and more reliable. I've worked on and off for the biggest CDN company in the world, Akamai Technologies, since 2006 and saw the Internet evolve from a front-row seat.

Let's look at different types of websites and services that became available on the Internet and how they managed to turn their online presence into a revenue stream.

Social Media The first decade of the 21st century saw the emergence of social networks with Myspace, Hi5, Friendster, Tagged, Bebo, Pinterest, Instagram, Facebook, Twitter (now X), Google+, YouTube, and LinkedIn. Storing and delivering user-generated content (photos, videos, articles) to someone's restricted circle was challenging and costly. CDN providers like Akamai had to adapt to the new trend and develop a multi-tier caching strategy to store and deliver content efficiently. Many of the early social media companies did not survive, mainly because they could not figure out how to monetize their content. Facebook, Instagram, X, YouTube, and LinkedIn fared the best and remain the biggest social networks in America and Europe. But these established platforms are getting some competitive pressure from new entrants like TikTok, favored by younger crowds. The primary source of revenue for social media companies comes from online advertisements or premium membership.

Dating Websites Dating sites such as Match, eharmony, and Tinder piggybacked on the social networking model. The business model and monetization aspects were much more straightforward for them. Instead of flooding their users with ads, they would charge a monthly subscription fee to give them access to millions of profiles and connect them with compatible people who share their interests.

Media Websites Websites belonging to the largest broadcasters, like NBC, first published news articles or content about shows on their site. Then, progressively, they started streaming their programs online or making them available on demand. It took broadcasters a while to find a way to monetize the Internet, but, in the end, the solution to monetize free content consisted of building technology to interrupt playback or a live stream with commercials. Later, media sites also introduced online subscriptions or pay-per-view for premium content. What was challenging initially for broadcasters was the need to support multiple proprietary formats such as QuickTime from Apple, Windows Media, and Adobe Flash. It was also a significant headache for CDN companies, as they had to maintain several networks to support all these formats. Standardization of protocols like HTTP Live Streaming (HLS) or WebRTC normalized the streaming methods. What made things even more challenging was that the screens that users used to watch the content became bigger and bigger with the introduction of smart TVs. Media websites wanted to offer the same quality of picture whether the user watched through traditional cable or satellite services or online. The image quality also had to be the same whether the user watched from a phone connected to a mobile network, a tablet connected to a medium-speed residential Internet service provider (ISP), or a large-screen TV connected to a high-speed Internet connection. This required CDN companies to support different bitrates and ever-increasing standards, starting with standard resolution, followed by High Definition (HD), Ultraviolet, 4K, and Over-the-Top (OTT).

Retailers The Internet allowed retailers to have a point of sale open 24/7. The consumer could browse products and shop from the comfort of their home any time of the day or night, and the product would be delivered a few days later. It required retailers to open new procurement centers to fulfill the orders, but this proved a very lucrative business, generating millions of dollars daily. It took a few more years for grocery stores to offer an online shopping experience with a delivery service since dealing with fresh produce is more complex. E-commerce transformed how retailers interacted with customers and opened new opportunities to expand their brand outside their traditional audience or borders. I've seen many well-known brands offer international shipping over time, turning their website into a global store overnight.

Gaming In the early days of the Internet, the gaming industry used the Web to advertise its products, which were available only as physical media to run on a personal computer or game console. In the past, if someone wanted to play video games with their friends, they first needed a game that supported multiplayer capability and a game console and they needed to go to their friend's house. Little by little, consoles could connect to the Internet. New games were developed to support playing online with other people. At the same time, gaming applications for mobile devices that offered an online experience by default emerged. An avatar represents each player, and a player can even buy digital content to outfit their avatar and interact with others who may live on the other side of the planet. While before the main revenue stream for video game studios was only selling the game, online gaming opened up new opportunities to sell additional packages or extensions to enhance the playing experience. For the increasing number of free games available on the Apple App Store or Google Play, the in-game purchase option that generally opens a more exciting or interactive experience becomes the only source of revenue for the publisher.

Banks and Financial Institutions Banks and financial institutions adopted the Web and offered online services in the mid-1990s. However, because the industry is more regulated and due to the nature of their business, this led to a more conservative attitude toward adopting CDN technologies to accelerate and secure the user's transaction. Their main hesitation was allowing the CDN service to handle their certificates. The reluctance was understandable, considering the risk associated with the transactions and content they were dealing with. They needed to ensure their user identity and life savings would stay safe and secure.

Healthcare Providers Like the banking industry, healthcare providers were conservative in adopting the Web to ensure they could stay compliant with regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and not inadvertently leak their patients' medical records.

Looking back over the last 20 years, the Internet has changed everything: it has altered the way we shop, bank, interact with our healthcare providers, interact with each other, book our vacations, explore the world, and even work! In the past, someone starting a new business was required to open a physical location. It was only possible to sell products with points of sale at various strategic locations that consumers visited. Now, one can quickly open an online shop and make their product available worldwide. A company doesn't even need to have physical offices anymore. With collaboration tools like Zoom, Webex, Teams, and Skype, meeting virtually from anywhere and running a business is easier. This has been a lifesaver and allowed businesses to survive during the COVID-19 pandemic. The rare few companies that did not have a solid online strategy struggled or disappeared.

Now that the Internet is very well established, we have started to hear about the Metaverse. But what is the Metaverse? The Metaverse is a vision of what the Internet and the online experience may become in the future. But not everyone has the same idea of what that looks like. Some say virtual reality (VR) or augmented reality (AR) technology is key to the Metaverse. The experience will be similar to playing games through an Oculus device. The Metaverse may include some existing components, such as cryptocurrency, that would represent the main means of buying goods or services. But what would those goods be? Are we talking about real products that will be delivered to our doorstep or a virtual shirt and hat to dress an avatar? Maybe all of the above? What about people? Will we interact with real people or AI entities trained to converse on various topics? During the pandemic, I participated as a speaker and attendee in virtual trade shows and security conferences where one could attend pre-recorded or live presentations and ask questions to the presenter through a chat application. One could also visit virtual booths on the virtual show floor to discuss with a sales representative about products solving various security problems. The experience was a mix of video games, social media, video streaming, Zoom meetings, and a good dose of awkwardness. One could navigate the various areas of the conference and join group activities, presentations, and chatrooms to meet other professionals and discuss topics or even join a virtual happy hour to sip wine sent by mail by the event sponsor. The experience was so different from an in-person conference that it fits what the Metaverse means to me. Now, one could argue that none of this is it since all I described already exists. It's always difficult to envision the technology of the future, as we can only imagine it based on what we know is possible today. What the Metaverse will be also depends on what technology or experience users will be the most comfortable with and adopt. Further technological advancement in computing will also be required to support the complex and rich interactions. If we look at how our parents' generation envisioned the future 30 years ago, they got a few things right but also many things wrong. For example, I remember seeing a documentary about the house of the future when I was a kid, and it featured a device that looked like a tablet where one could read the news. That part became a reality with tablets and

smartphones, which is how most people consume news today. That same documentary also described something like virtual reality. However, they made it sound like holograms would be common by now, and no one ever talked about wearing those bulky goggles. What things did they get wrong? Well, nothing to do with the Internet, but where's my hoverboard, and what about flying cars? It's probably a good thing they got those concepts wrong!

The Different Layers of the Web

The Web is not uniform, and not all content is readily visible and accessible by all. Figure 1.1 presents the different layers of the Web as an iceberg.

The visible and easy-to-access part is known as the *surface web*. Its content is discovered, indexed, and found through web search engines. It comprises all e-commerce, travel, news, social media, gaming websites, and more. The content can be accessed freely. This part of the Web represents only about 4% of the Web.

Next comes the *deep web*. Access to the content and resources is generally restricted behind authentication mechanisms and may contain private information. Most industries are represented in this part of the Web. For example, in the case of e-commerce, the deep web corresponds to the user's purchase history, shopping reward, or tailored services. For social media, it consists of posts and pictures that users publish and share only with their network. This corresponds to an account holder's bank statement in banking or a patient's medical records for healthcare. The deep web represents about 90% of the Web and is legitimate. Access to the

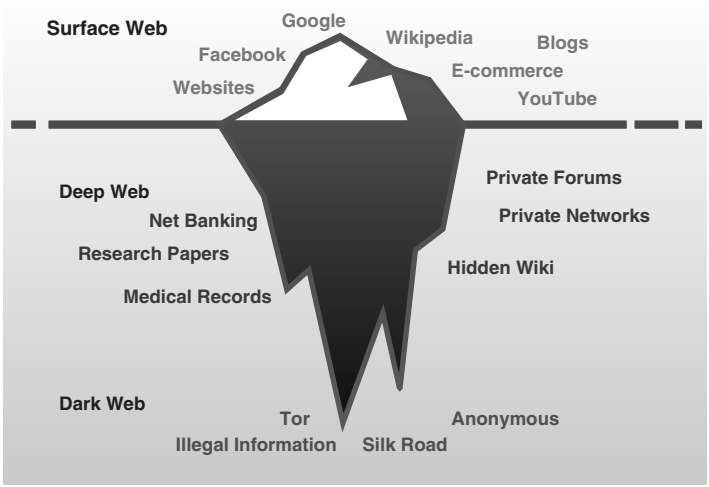


Figure 1.1 The different layers of the Web

content of resources is restricted to the population subscribing to specific services, and web search engines cannot discover and index the content.

Finally comes the mysterious *dark web*, often associated with criminal and illegal activity. This is true but somewhat of a misconception since the dark web was initially designed to provide Internet users with anonymity. For example, the infamous The Onion Router (TOR) network was initially invented to protect American intelligence online communication. The code was released in 2004 under a free license and later launched as a free service to all Internet users who wanted to browse without being tracked. Anonymity is, of course, what criminals and hackers look for so that they can carry out their schemes without being disturbed or traced. The dark web is where one can find marketplaces to acquire stolen data from various security breaches, accounts harvested through credential stuffing attacks, or stolen credit cards. You'll also find forums for hackers who exchange methods and best practices to carry out various fraud schemes without being detected or caught. The dark web accounts for about 6% of the Internet. Sellers offer multiple categories of products or services on the dark web. Figure 1.2 shows the now-defunct AlphaBay marketplace that once offered various categories of products, including fraud, hacking and spam, malware, drugs, and illegal chemicals. The FBI announced the takedown of AlphaBay in 2017.

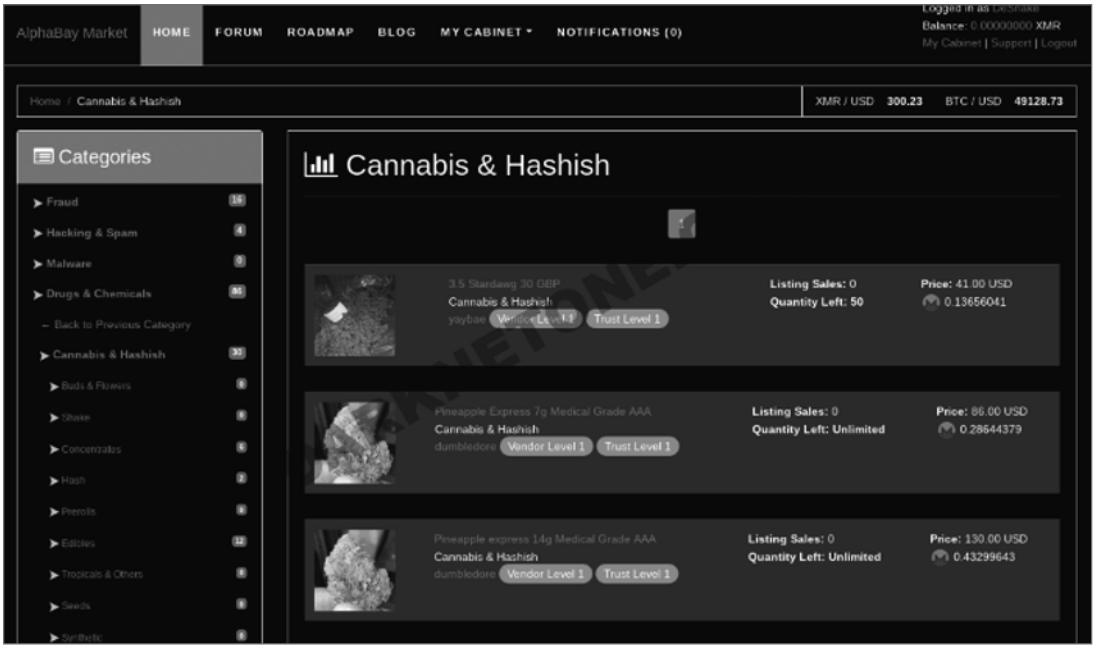


Figure 1.2 The AlphaBay marketplace on the dark web

The Emergence of New Types of Abuses

When companies develop a website, they usually focus on specific use cases. For example, suppose a retailer makes its inventory available online. In that case, its focus is on showcasing its products, giving its users the best possible experience, and making the shopping experience as easy as possible to increase its revenue. Security is not part of their core business, and they would not necessarily think about how someone with not-so-good intentions could exploit their site to defraud them and their users. The expansion of commerce on the Internet provided many opportunities for new abuses. In retrospect, it's easy to criticize past design choices made by software companies or website developers when new vulnerabilities that seem evident and outrageous today are found, especially when one realizes that the vulnerability existed for years. However, as a longtime product and software designer, I can attest to the difficulty of anticipating how a feature or workflow could be misused and exploited to commit fraud or abuse a resource. Some hackers specialize in developing viruses that exploit vulnerabilities in the operating system or browser running on the user's machine. Computer viruses have been omnipresent for years, and the ever-increasing ease of exchanging information through a growing number of communication channels has made their spread easier. Viruses can serve multiple purposes, including spying on the user's activity and collecting information like credentials or credit card numbers by logging key presses, stealing or encrypting the content of a disk (ransomware), or serving as a relay in a botnet. Antivirus is part of the enterprise security strategy and is beyond the scope of this book. Other hackers choose to exploit public website's resources and application programming interfaces (APIs) to collect information and defraud users. Protecting against such attacks is within the realm of application security, which we'll discuss further in the following chapters.

As product architects and developers, we follow certain design principles to achieve a specific goal, and overall, we want to keep things as simple as possible to make a product as easy as possible to use and maintain. Thankfully, the software development community has learned about common exploits and past mistakes over time. Most developers also go through yearly secure coding training. Security awareness has improved along with coding practices, but all this knowledge is relatively new and did not exist when the first e-commerce sites started to appear.

How attacks were classified changed over time based on the improved collective understanding of the attacker's motivations. The 2000s saw the rise of what is now known as *application-layer attacks*. SQL injection, cross-site scripting (XSS), command injection, and cross-site request forgery (CSRF) are commonly used to exploit vulnerabilities on a site to steal information or money or sometimes deface a website. The Open Worldwide Application Security Project (OWASP), MITRE's Common Vulnerabilities and Exposures (CVE) Program, and other organizations have

been vital in spreading awareness of new vulnerabilities, providing guidance on how to solve them, and encouraging better coding practices. The development of open-source web application firewalls (WAFs) such as ModSecurity or equivalent commercial offerings has also been vital in proactively preventing such exploits even when vulnerabilities exist on the site.

By the early 2010s, there was much talk about denial-of-service (DoS) attacks—or their distributed denial-of-service (DDoS) variant. DoS/DDoS attacks come in two primary flavors. First, there are activist attacks, like the infamous Anonymous group that targeted companies or organizations because of their position on specific issues. I'm not going to debate whether Anonymous' motivations were right or wrong, but they were undoubtedly disruptive when they managed to rally a large crowd to their cause. Those with experience dealing with them probably remember the infamous Low Orbit Ion Cannon (LOIC) or the High Orbit Ion Cannon (HOIC) DoS tools, both point-and-shoot attack tools that offered various options. Sites that were not well protected would soon get overwhelmed by the load and essentially be taken offline, which sometimes led to me having to do an emergency integration of the web security product from the company I was working for at the time to protect a site. It often happened on a Friday evening. (Come on, guys, have some compassion for the hard-working web security community here!) A well-executed DDoS attack against a retailer on a critical day like Black Friday can lead to millions of dollars in losses. When these attacks became more common, web security companies extended their WAF, offering DoS/DDoS protection. This mainly consisted of rate-limiting features, IP reputation, and a set of rules designed to detect known attack tool signatures like LOIC or HOIC, most of the time also coupled with a dedicated security analyst who could craft new rules to deal with more recent attack signatures or so-called zero-day attacks.

Beyond the activist attacks, a lot of what is perceived as a DDoS is, sometimes, overzealous botnet traffic scraping a site to collect product descriptions, pricing, and inventory, running a credential stuffing attack to test if a username/password combination is valid, or other types of attacks we'll discuss in detail in Chapter 2, "The Most Common Attacks Using Botnets." In the early 2010s, no one talked much about bot attacks. But I saw firsthand that a lot of the traffic causing site availability issues was poorly calibrated bot activity. Now, you must remember that in the case of content scraping or a credential stuffing attack, an attacker who takes the site offline is making their attack less effective since it will increase the time it takes to complete the task and, ultimately, their cost. So, a denial of service caused by a botnet is an unintended consequence since their intent is not to take the site down but rather exploit it. Botnet operators quickly learned how to work around the DDoS detection in place, making them ineffective.

At first glance, excessive activity on the login API, for example, can prevent real users from logging in and purchasing products. This can be perceived as a DDoS attack from an activist group wanting to impact the revenue of the targeted company. When looking closer into the

attack, one will discover that each request is for a different username/password combination. Let's look at it from the attacker's point of view and reflect for a minute on why login is such a popular target. In general, e-commerce sites encourage their customers to create an account to get discounts, facilitate their purchases, and improve their overall experience when visiting the site. Consumers can then manage their accounts remotely and review their purchases and credits. They also provide their personal information, including phone number, postal address, credit card or bank account number. . .basically the type of information or services hackers can exploit or monetize.

Gone are the times when criminals would break into a facility or home, storm the filing cabinet or safe, and steal whatever paperwork or money they could find. Now, this source of information is potentially available right at their fingertips with far less risk involved. What's more, we're no longer looking at potentially hardened criminals. Anyone can be a hacker, like computer-savvy teenagers, who may be committing crimes without realizing that what they are doing is illegal. The tool of choice is no longer a crowbar to break a door or window. A laptop, some programming skills, a well-crafted Python script, a few virtual machines running in the cloud, and an armada of proxy servers are the essential ingredients of botnets and have become more effective tools for criminals. There were only a few bot management offerings in the early 2010s; this gave me the idea to look closer at the characteristics of attack traffic, and I started building my first bot management product.

The Proliferation of Botnets

The term *bot* is short for *robot*, which generally designates software running on a computer designed to perform a specific task. In the web context, this task mainly consists of collecting the data available on the Web or automating certain transactions. In the fraud and abuse context, the task consists of abusing various site functionalities and performing multiple types of attacks known as *account takeover*, *credential stuffing*, or *account opening abuse*, to name a few. Chapter 2 discusses in greater detail the various types of attacks where bots are used.

A *botnet* is a network of bots that run the same software designed to accomplish a specific task. A command-and-control center coordinates the activity of the individual bots. The botnet's size can vary depending on the task and the level of sophistication required to defeat any protection in front of the targeted website. Figure 1.3 represents the high-level structure of a botnet.

Let's establish one thing: not all botnets are equal, and not all of them are bad. Some of them are vital for the Internet to function well. For example, companies like Google or Bing that run web search engines depend on botnets known as Googlebot and Bingbot, respectively, to crawl the Internet to index its content and discover new sites.

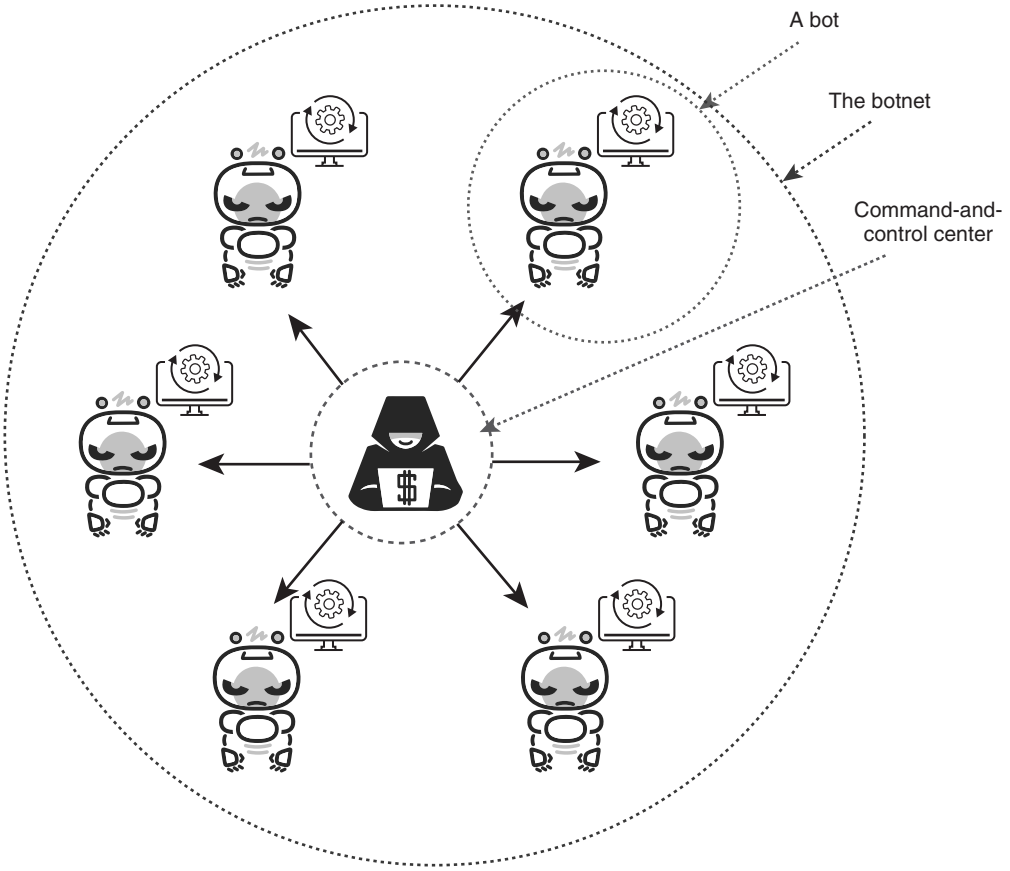


Figure 1.3 The high-level architecture of a botnet

Legitimate enterprises also use automation to handle repetitive yet business-critical tasks. Services exist to monitor the performance or availability of business-critical sites. Online marketers leverage botnets to scrape a site's content and analyze customer sentiments regarding a particular product or service. Online advertisers also scrape the Web to evaluate their ad placement to get the most return on investment. These ads may be annoying, but let's be honest: they're necessary to keep a significant portion of the Internet free and offset the cost of developers, facilities, and hosting.

Fraudsters have also mastered automation and commonly leverage botnets to automate part of the workflow that will lead to a successful fraud attack. They, too, can pay top dollar for top developers with expertise in defeating bots if they don't have the skills to do so themselves. I occasionally come across legitimate-looking job ads with very competitive salaries, targeting

employees of companies that offer bot management products. Automation is commonly used in the following use cases:

Product and Price Scraping Legitimate companies specializing in business intelligence use botnets to scrape and extract data from public websites. They process the data to extract competitive intelligence. The data is sold to their customers, who use the information to adjust the product positioning and pricing in an attempt to attract more customers and increase revenue.

Inventory Hoarding and Scalping This use case consists of scraping a site for specific high-value items (most commonly sneakers and concert tickets). The bot is also designed to automatically check out as much of the inventory as possible. The bot is used only during sales events. Any inventory acquired will be sold on a secondary market at a premium to legitimate users who could not get the item before it was sold out.

Credential Stuffing This use case, also known as credentials enumeration or account checking, consists of verifying a known credential set (username and password) against various websites. An account verified through this process can be sold on the dark web. The account will eventually be taken over and used in various fraud schemes.

Account Opening Abuse This use case consists of creating fake accounts on a website. These accounts are later used as part of a broader fraud scheme. For example, scalpers create fake accounts to get a better chance to check out items with limited availability during sales events or take advantage of a promotion offered to new account holders.

Carding Attack Also known as a card enumeration attack, when performed against gift cards, the attack consists of "guessing" the gift card number and PIN. The attack is often performed against the website gift card balance application or using the "Add Gift Card" feature in the checkout workflow. Carding against credit cards works similarly, except in this case, the attacker must guess the CVV and zip code correctly.

Spamming This use case consists of posting spam content on forums, product review boards, or social media to advertise competing products or services in an attempt to hijack the audience.

These attacks affect all industries, but they affect e-commerce, travel, and hospitality websites the most. Successful attacks such as the preceding use cases may require sending hundreds of thousands of requests, which cannot realistically be done manually in a cost-effective manner. Like any regular business, fraudsters always look for ways to scale and automate their operations

to maximize profit. The scale and sophistication of the attack and infrastructure are usually proportional to the revenue incentive.

Quantifying the Bot Traffic Volume on the Internet

Out of all the use cases where bots are used, scraping represents by far the highest volume of traffic. Where a credential stuffing attack can amount to a few hundred thousand requests per day on a website, scraping activity typically results in millions of requests per day. This is why I chose to quantify how much Internet traffic comes from bots by looking at the scraping activity. The bot-to-human traffic ratio varies depending on who you ask. In truth, nobody knows because nobody can monitor the whole Internet and run bot detection on it. Even if one could run bot detection on 100% of the Internet, it also assumes that the detection is 100% accurate, which is challenging despite everyone's best efforts in the web security industry. Lastly, bot activity comes and goes. Today's scraping traffic may not be the same as tomorrow.

To answer this question, I took a one-week traffic snapshot from a representative sample of websites that includes global fashion brands, major U.S. retailers, international airlines, and hotel chains. The sample used for the study consisted of 1.16 billion HTML pages or API requests. Requests for static content such as images, style sheets, JavaScript, videos, or fonts were ignored for this study.

From the study, one fact became apparent: the level of bot activity a site may receive largely depends on the popularity of their products or services and their ranking on the market. Figure 1.4 shows an example of a sportswear brand catering to the U.S. market. Trendy brands can attract not only a lot of customers but also the attention of multiple bot operators. In this case, only about 18% of the traffic comes from real humans, and the rest (82%) comes from bots. Out of the 82% of bot traffic, only a small portion comes from good bots such as web search engines, search engine optimization (SEO), and social media. The rest comes from scrapers.

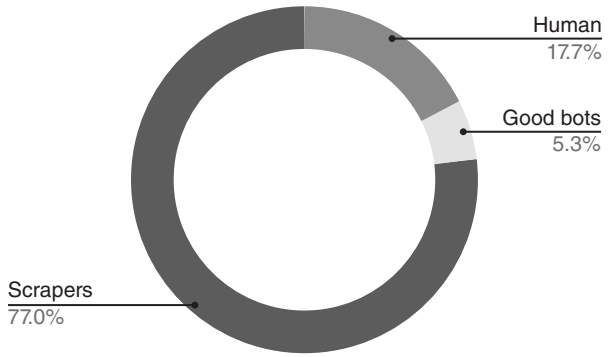


Figure 1.4 Traffic distribution for a sportswear company in the United States

In contrast, Figure 1.5 shows the traffic distribution for a retailer selling auto parts online with a mild bot problem. In this case, just over 15% of the traffic comes from bots, two-thirds of which comes from good bots.

Figure 1.6 shows another example from a home improvement company catering to the U.S. market, where the distribution between scraper, human, and good bots is more evenly distributed. Yet, the overall bot activity accounts for 60% of the total traffic to the product pages.

Looking at all the sampled websites, the average bot-to-human traffic ratio is about 70/30 (see Figure 1.7). The sample used for the study is not large enough to conclude that 70% of the Internet is bot traffic, but we can safely conclude that the great majority of the traffic on the Internet originates from bots.

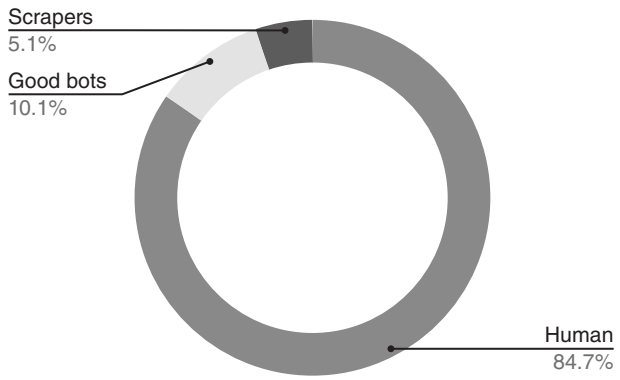


Figure 1.5 Traffic distribution for an auto part retailer in the United States

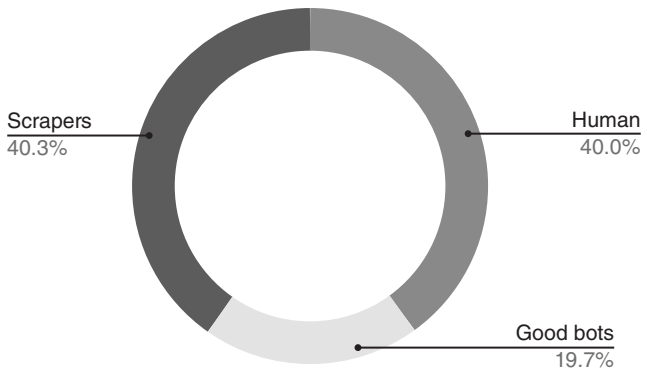


Figure 1.6 Traffic distribution for a home improvement company in the United States

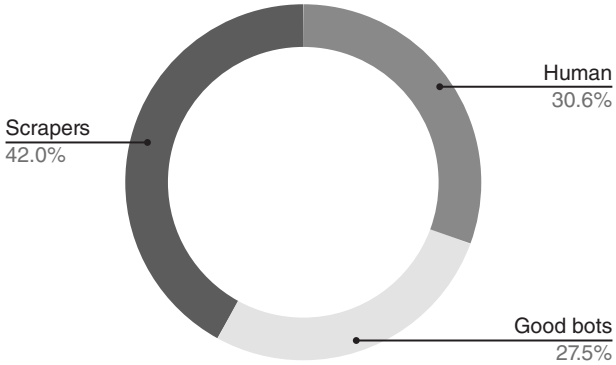


Figure 1.7 Typical traffic distribution for a website

Botnets Are Unpredictable

Botnet activity on a website is very unpredictable. The activity can be intense for a few days and disappear as quickly as it appeared. The pattern of the bot traffic varies as well. Here are a few examples of botnet traffic patterns. One often expects continuous and persistent activity, as shown in Figure 1.8. In this example, the botnet made more than 4 million daily requests using various randomization methods, such as randomizing the User-Agent and HTTP headers and leveraging 100,000 unique IP addresses.

With some botnets, the activity may stop and resume after several hours or days, as shown in Figure 1.9.

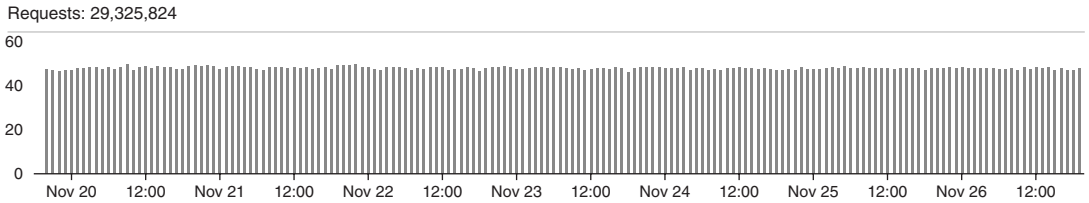


Figure 1.8 Continuous and persistent bot activity

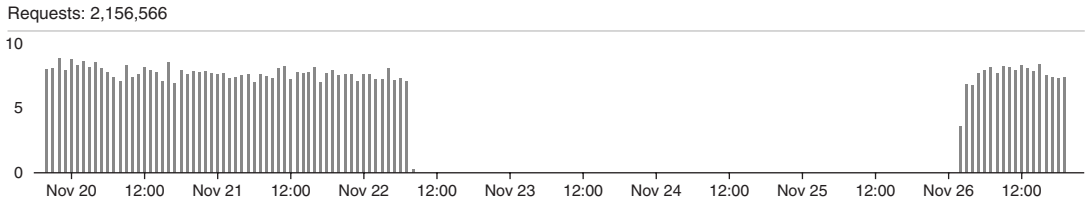


Figure 1.9 Persistent bot with long periods of inactivity

But sometimes, the activity is more punctual, as shown in Figure 1.10. The activity may start and stop at specific times, and the same content may be consistently requested daily from the same set of IP addresses.

One-off activities, as shown in Figure 1.11, may also occur where the botnet will get the information it's looking for, never to return.

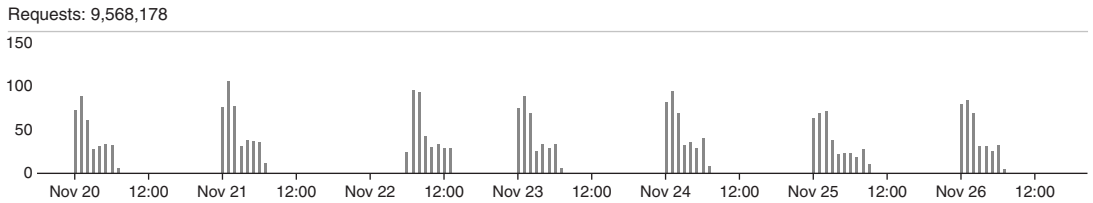


Figure 1.10 Punctual bot activity

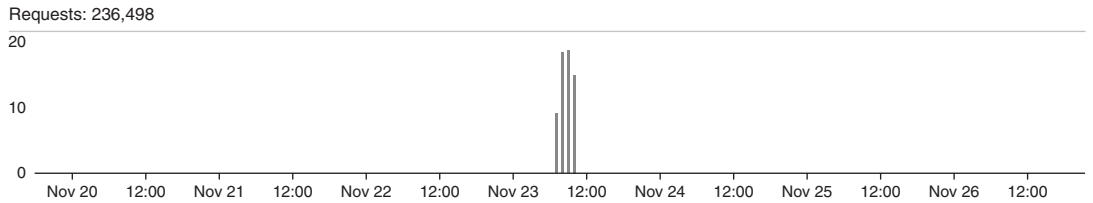


Figure 1.11 One-off bot activity

How botnets behave depends on the needs of the entity or operator that controls them. The quality of the detection and mitigation strategy plays a significant role in the botnet behavior. Mitigating the activity (for example, deny) will force the bot operator to take action. I remember one of my first fights against a botnet back in the days when the tool of choice for such occasions was a simple WAF. After analyzing the traffic, I found that the bot traffic was coming from a very specific network that belonged to a cloud provider. I could also confirm that no legitimate traffic was coming from that same network. Content with that finding, I updated the WAF policy to block the traffic coming from that network. Happy to have come up with a quick solution, I confirmed that the activity was getting blocked once the new policy was deployed. A few minutes later, the bot activity that I had identified with my detection seemed to stop. As part of the resolution, I expected the overall traffic volume to drop. Strangely, though, it remained the same.

After some more digging, I found that the bot traffic had "migrated" to a different cloud provider and had resumed after a short interruption. It took a few more iterations to come up with a more clever signature to block it once and for all. With a strong detection strategy that includes multiple detection methods looking at the traffic from different angles, bot operators

have far fewer opportunities to evade detection. They ultimately stop their activity and go after an easier target whenever possible. Seeing the bot activity disappear doesn't mean it is gone forever, though. Motivated and skilled bot operators will try various methods to defeat the detection to get access to the content they are after. It's common to see small bursts of activity that sometimes last only a few seconds. This is usually the sign of bot operators trying a new attack strategy and cutting their experiment short as soon as they are still getting blocked. It's essential, even during "peacetime," to maintain the bot management solution and adopt newer detections to prevent opportunistic returns from botnets.

It's hard to say if bot activity on the Internet is on the rise. The bot traffic volume does not represent the number of adversaries. When a bot operator faces resistance from an advanced bot management system, it's common to see the volume of bot activity increase, sometimes even doubling. The reason is simple: some botnets have a retry mechanism that kicks in when a request fails. This could be interpreted as an increase in the number of adversaries, whereas in reality, this is still the same botnet that is retrying some failing requests. Looking at the volume of bot requests matters since excessive activity can overwhelm a website and take it offline, but what matters the most is understanding the number of botnet-sending requests. Each botnet corresponds to an adversary. Having good visibility on the number of botnets attacking a site will provide a better understanding of the threat landscape affecting it.

In later chapters, you'll discover the complexity of bot attacks and how to build an effective bot management solution to defeat them consistently.

Bot Activity and Law Enforcement

Vendors selling bot management solutions often use terms like *criminals* or *criminal organizations* when talking about the adversary (the bot operator). Bot activity other than that coming from good bots is a nuisance and violates most websites' acceptable use policies. Undoubtedly, the adversary has strong intentions for some types of attacks and aims to defraud or take advantage of others for their benefit. This is the case, for example, for account takeover and carding attacks, which I'll discuss in more detail in Chapter 2. That something is considered criminal, however, assumes that laws are in place to deal with these situations. Unfortunately, few legal regulations exist regarding the usage of bots on the Internet.

The Better Online Ticket Sales Act, also known as the BOTS Act, was introduced into law in the United States in 2016. This act prohibits individual organizations from buying tickets using bot technology and reselling them at a profit. This activity, known as *ticket scalping*, has plagued ticketing companies like Ticketmaster and ultimately artificially inflated the price of concert tickets for legitimate users. The fine for a convicted bot operator is $16,000. This law is a step in the right direction. It was enforced in 2021 against Just In Time Tickets, Inc. and

Concert Specials, Inc. which were found guilty of purchasing tickets from Ticketmaster using a botnet and reselling them on their website at a premium. But this law seems to do little to discourage thousands of scalpers from buying and reselling tickets online, as seen regularly when tickets for famous pop stars go on sale. It is extremely difficult for legitimate fans to purchase tickets due to the excessive usage of botnets that grab up the inventory as soon as they are released. This situation led to a controversy in November 2022 when tickets for Taylor Swift's Eras Tour went on sale on Ticketmaster. The demand was such that the website struggled to keep up, but it also appeared that scalpers purchased many tickets.

Enforcing the BOTS Act—or any regulation around bot activity on the Internet—is difficult. Although the bot activity can be detected, bot operators are masters at hiding their tracks, making it challenging to attribute the activity to any particular person or entity for prosecution. Additionally, the bot activity may come from different countries (or even U.S. states) with different regulations from those where the site is hosted. This leads to difficulties enforcing acceptable use policies or determining what legal regulation applies.

Law enforcement is not powerless and scores some victories by taking down botnets regularly. However, as will be discussed in Chapter 3, "The Evolution of Botnet Attacks," bot operators are nimble. Shutting down a botnet doesn't always translate to a permanent operation shutdown. A botnet is just code that is waiting to be deployed on servers. The logistics involved in reactivating the botnet are relatively trivial. Thousands of companies offer cloud services from which a bot operator can choose. Most of these providers do their due diligence to prevent botnets from running from their servers since it could tarnish their reputation. When they find bot activity originating from their servers, they may shut down the activity and close the associated account. However, a provider with a self-service offering may apply less scrutiny to vet their customers or the services they are running from their servers. Considering this, companies with a major presence on the Internet must arm themselves with bot and fraud detection products to defend their website, prevent abuses of their resources and intellectual properties, and protect their users.

Summary

This chapter provided a brief history of the Internet, described the bot problem, and identified the main types of attacks that use them. It also looked at how prevalent bot activity is on the Internet and its typical behavior. There is little to no law at this time that can help website owners fight the initiator. But even if there were, attributing the activity to a particular person, group, or entity is always challenging. Now that you have a high-level understanding of bot attacks, the next chapter dissects how the various attacks work.