April 2022

The new ABCs of application security

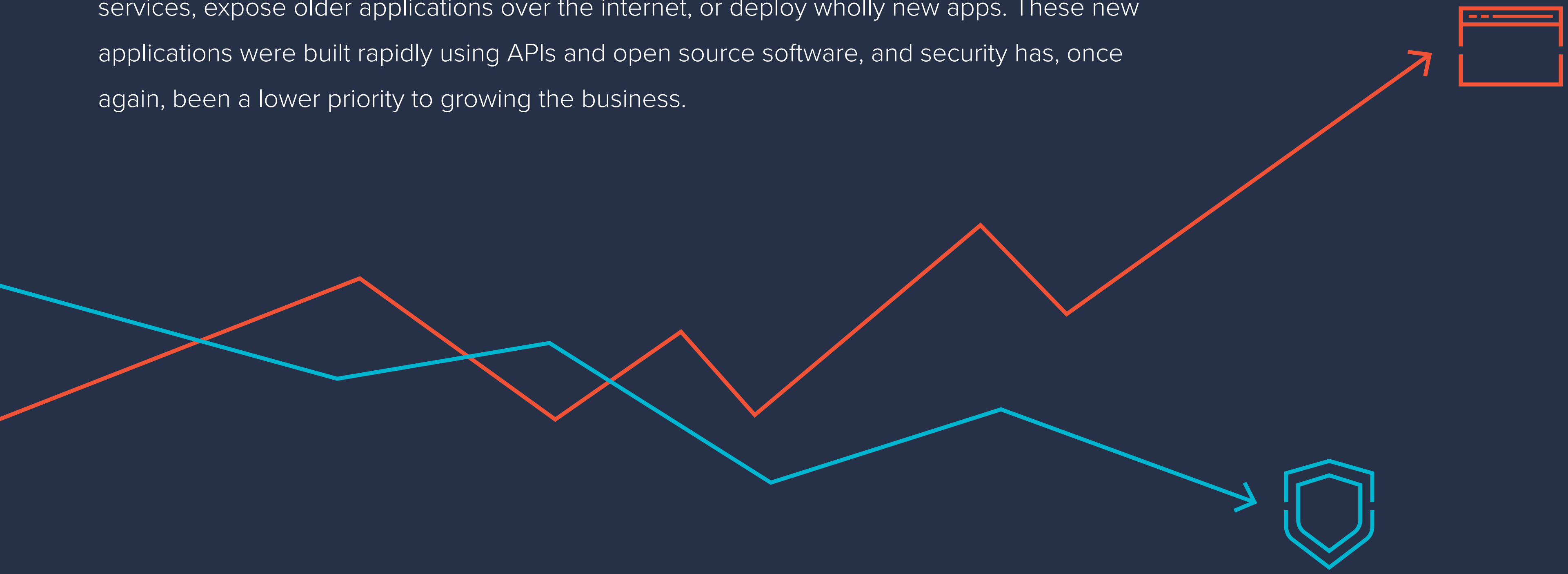
From API vulnerabilities and bots to client-side protection

Barracuda

Table of contents

Introduction: Newer, more dangerous attack vectors

Applications are the building blocks of how digital businesses work and how they engage with their end users and customers. The move to remote work in 2020 intensified the importance of web apps, and many organizations have had to pivot rapidly to upgrade their existing web services, expose older applications over the internet, or deploy wholly new apps. These new applications were built rapidly using APIs and open source software, and security has, once again, been a lower priority to growing the business.



Organizations have always faced an array of challenges when it comes to application security. Applications are a top attack vector for data breaches — something that is borne out by the Verizon Data Breach Investigation Report for some years now — and are one of the top two breach reasons. Starting with the traditional attacks against web applications, such as SQL injection, cross-site scripting, and command injection, these attacks have spread to APIs and mobile apps as well.

Over the past few years, the threats to applications have multiplied, and newer, more dangerous attack vectors have emerged. The fastest-growing attack vectors are now API vulnerabilities, automated bot attacks, and client-side attacks. In fact, respondents surveyed for Barracuda's report The state of application security in 2021 pointed to bot attacks, web application vulnerabilities, software supply chain attacks, and flawed API security as the top four reasons for a successful security breach at their organization.

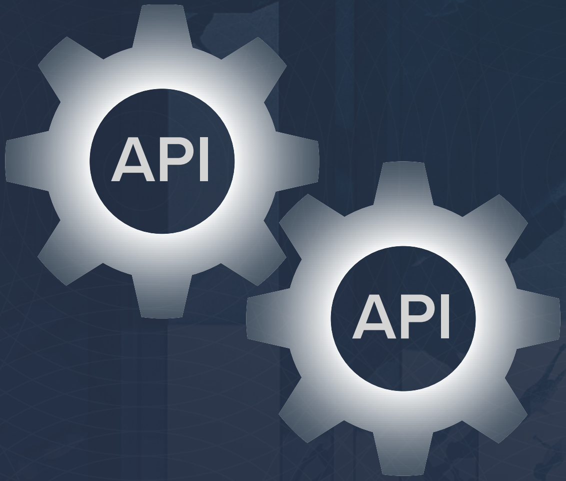
The number of vulnerabilities and breaches attributed to APIs and client-side attacks has grown exponentially, and some of these breaches like T-Mobile and British Airways have made headlines for all the wrong reasons. Client-side attacks, also known as supply chain attacks, were first discovered around 2018 and dubbed Magecart because they were primarily targeted at Magento-based online shops. The attackers first identified third-party JavaScript that was popularly used in checkout pages. Once they did this, they hacked these source files and inserted their card-skimming code. When a user loaded the site, the now malicious JavaScript was loaded, and it stole the user's credentials.

Bot attacks have impacted businesses in other ways. For example, one major attack is automated purchase of limited-edition items for resale. Known as scalping attacks, these lead to shortages for actual customers, such as the PlayStation 5 shortage. For some time now, bots have also been performing a variety of attacks, the most damaging of which are account takeover attacks and DDoS attacks.

This e-book takes an in-depth look at these three critical attack vectors — API vulnerabilities, bot attacks, and client-side attacks — as well as how organizations can fill the gaps in their application security and protect against these evolving threats.

A is for API security

For many years, APIs were primarily used in the backend of business applications, performing machine-to-machine communication. Today, APIs are everywhere, and they enable most of the applications that we use in our day-to-day life. Most mobile and web applications that we use for work and fun use APIs to function. They are the core of businesses, powering modern digital platforms and enabling digital transformation.



Organizations have turned to developing applications “API first” in a big way because this method allows them to innovate and go to market rapidly. APIs enable fast delivery when used with agile and DevOps practices, allowing developers to quickly build and release new functionalities for web and mobile applications. As API usage grows, they have become the foundation of critical services for the applications they enable, and in turn their access to critical data has increased exponentially.

The growth of APIs and their direct access to critical data has made them a prime target for attackers. APIs are built for automation, and this makes finding and exploiting insecure ones very profitable for attackers. Automated attacks make it faster and easier for cybercriminals to exfiltrate data than with a web application. API-based applications also encode their business logic on the application itself, as opposed to traditional applications where this logic is hidden in the backend server. This means that an attacker can easily sniff out application traffic to identify API endpoints and perform attacks against them.

The fact that APIs are a big new target for attackers is borne out by the BugCrowd PriorityOne report for 2021. In it, they disclose that the API vulnerabilities have doubled in a single year. These vulnerabilities are poised to grow more rapidly and evolve into one of the top vectors for application breaches in the coming years.

API vulnerabilites have

doubled

in a year

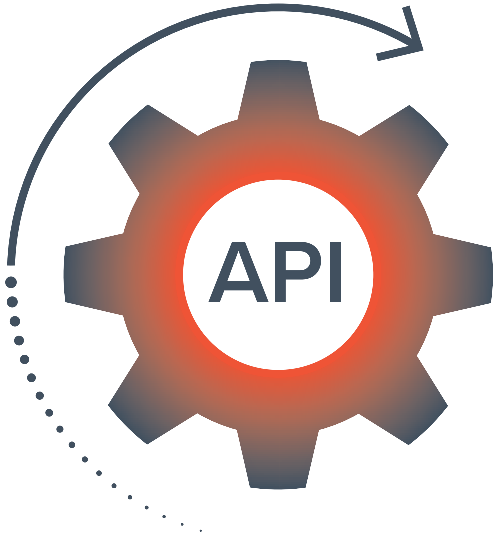
Example: Experian API exposed credit scores

A researcher recently discovered a large API vulnerability while shopping for student loans online. When he visited a lender's site, they offered to check his loan eligibility with his name, address, and date of birth. Being a researcher, he looked at the code running behind this search and found that it was an API call to Experian. The API that was in use was allowing lenders to send automated queries and fetch FICO credit scores from the credit bureau.

The researcher identified that the Experian API could be accessed directly without using any kind of application security, and simply entering all zeros in the date of birth field let him pull up anyone's credit score. He then proceeded to build a handy tool to automate the lookups. The API also returned, in addition to the credit scores, four "risk factors" that may explain the reason for a score.

Experian, when contacted about this, simply removed the API access for this single endpoint.

The danger of this exposure is shown by the tool that the researcher created. He was a researcher and reported the issue to Experian to be fixed. But, if this API endpoint was found by a malicious actor, they could have easily used it to collect the credit scores of anyone whose name and address are public information — possibly causing significant damage. It is unknown if the API call resulted in a hard pull or soft pull and how viewing the score affected the person's credit score.



Example: Brute forcing private meeting passwords on Zoom

Zoom meetings were, by default, protected by a six-digit numeric password. This means that any meeting had one of 1 million possible passwords in use. A researcher discovered that these passwords were brute-forceable, leading to Zoom bombing and similar attacks.

When Zoom had the numeric password, a user could use the link to the zoom meeting to open a webpage that prompted for the password. At this point, when the user filled out the required fields and hit enter, they could observe the backend API interactions to discover the vulnerability.

The important thing that was discovered about this process was that it had no rate-limiting enabled. This means the researcher was able to continually try passwords, and after 43,164 attempts that took about 29 minutes, they were able to discover the correct password — a rate of about 25 passwords per second. If multiple parallel machines were used in an attack, they could be used to crack the password very quickly.

This hack had massive implications. Given the number of high-level government agencies and similar organizations using Zoom, this issue could have caused severe damage by malicious actors dropping into meetings to eavesdrop on them. The researcher provided this information to Zoom, which implemented multiple changes to fix the issue.

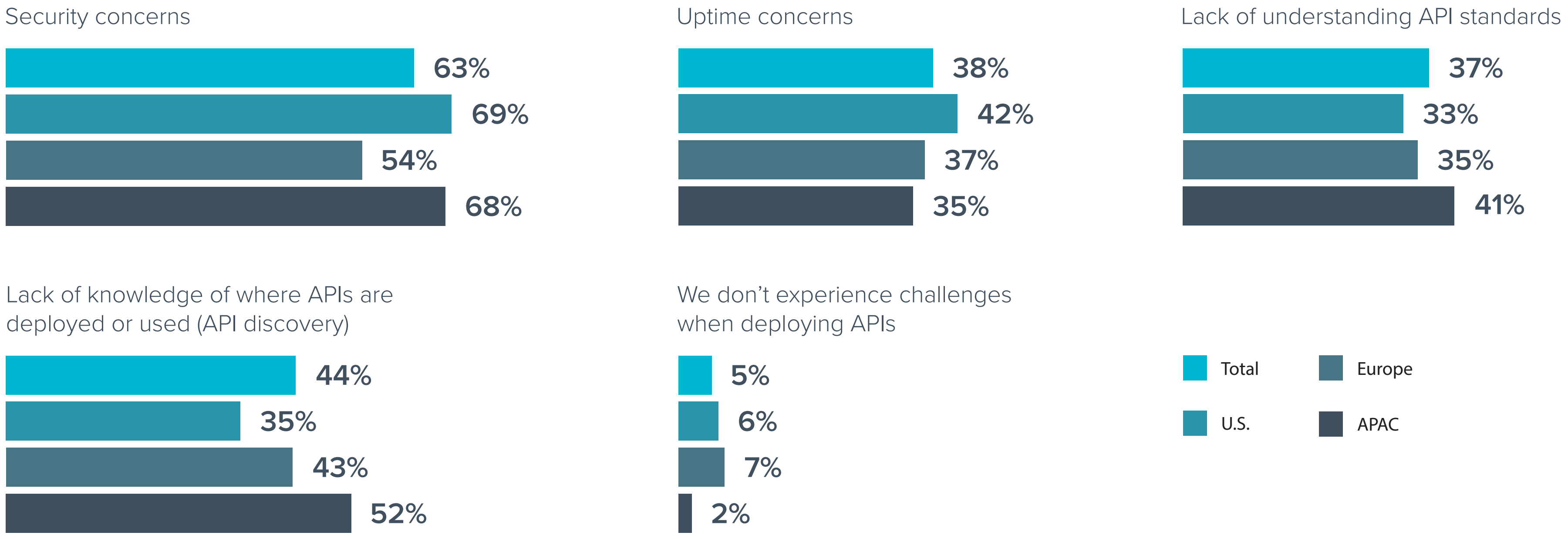
In addition to lack of rate limiting, another issue was discovered here — a lack of proper logging and monitoring, which would have relatively easily provided the Zoom team with notice that such attempts were occurring.

What application security professionals are saying

Given the outsize impact that an API breach can have, it is small wonder, then, that API security is top of mind for defenders. In Barracuda's recent survey of application security professionals, we asked them about the main challenges they face when deploying APIs — and security was the top concern. This is also borne out by the fact that the Open Web Application Security Project (OWASP) has released the API Security Top 10, providing a list of the unique vulnerabilities and security risks of APIs.

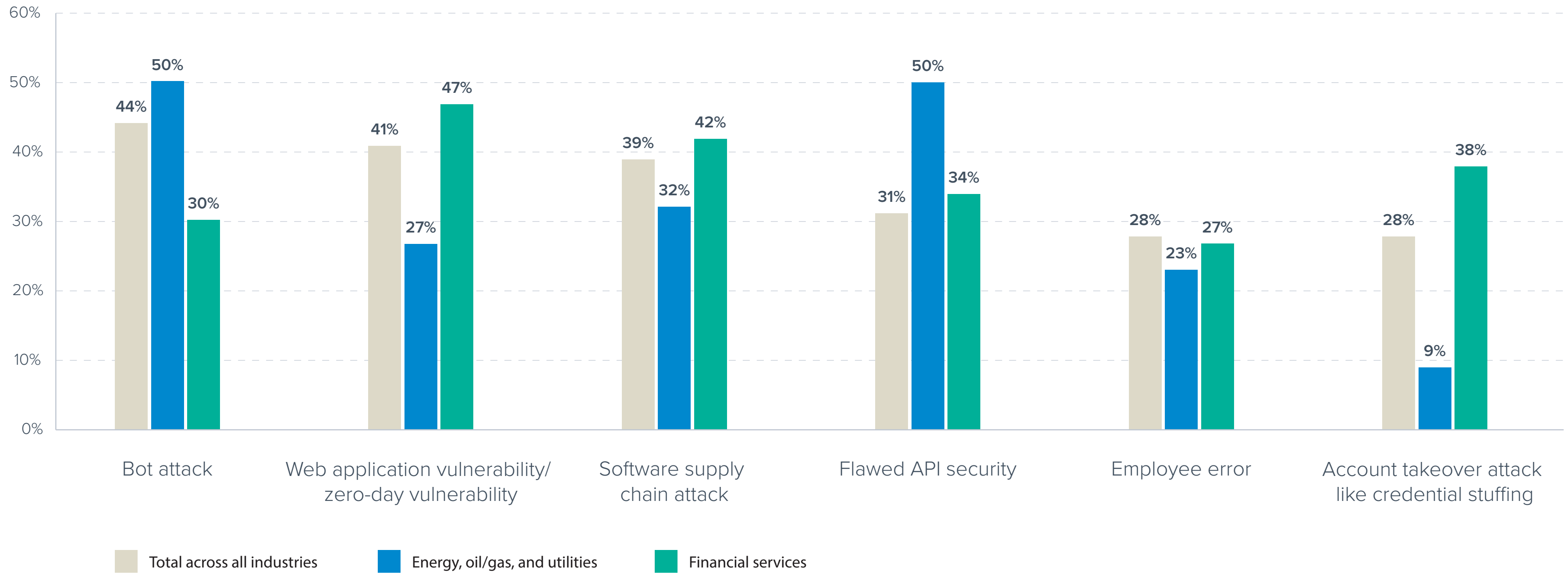
What are the main challenges your organization experiences when deploying APIs?

(n=728)



Which of the following contributed to the successful security breach that exploited a vulnerability in one of your organization's applications in the last 12 months?

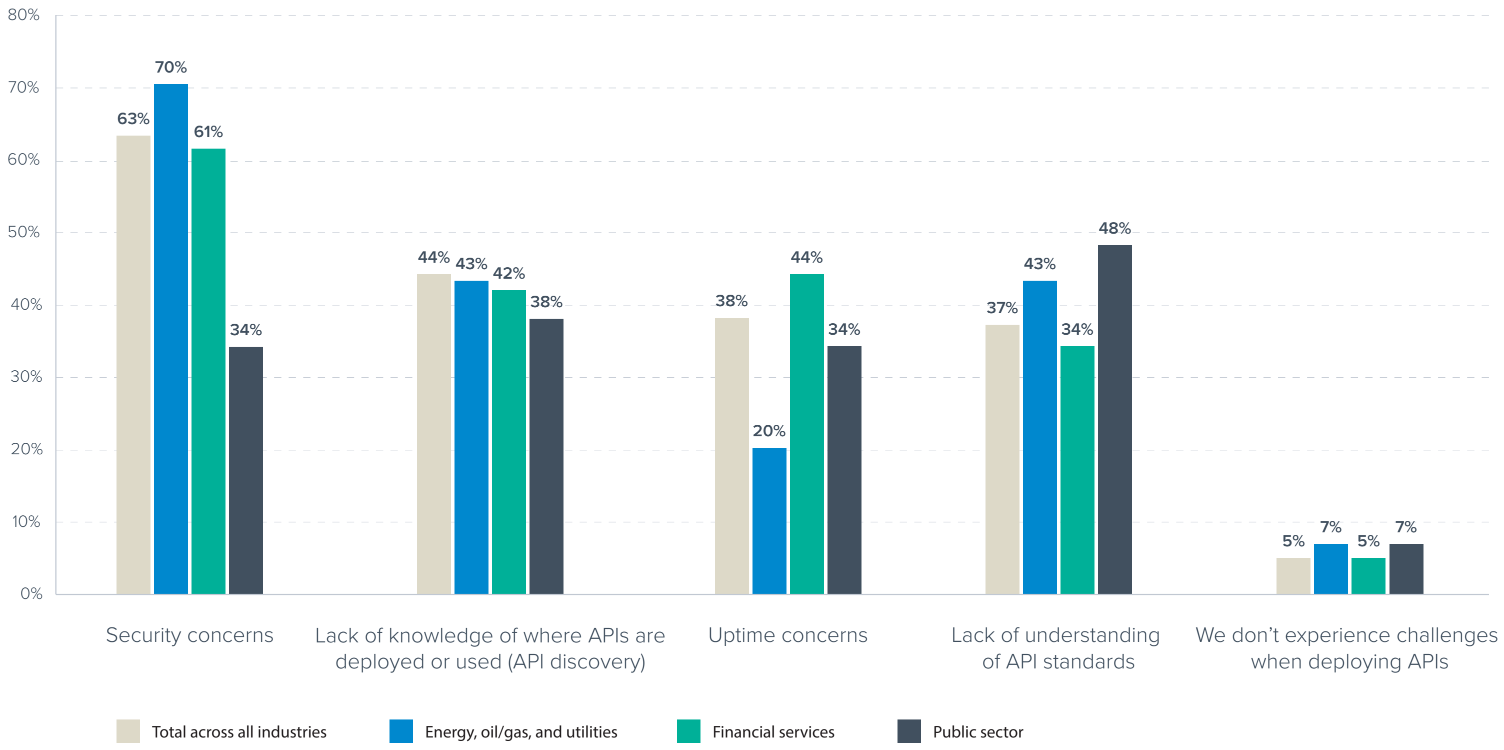
(n=541)



According to the survey results, the energy, oil, gas, and utilities industry was the most likely to be breached due to an API vulnerability. While the industry is mostly in the news due to targeted ransomware and IoT attacks, their APIs — many of which are public facing — are also under immense threat of attack. Respondents in the financial services sector, an industry where APIs are the lifeblood of automated transactions, also reported API breaches in their top breach reasons.

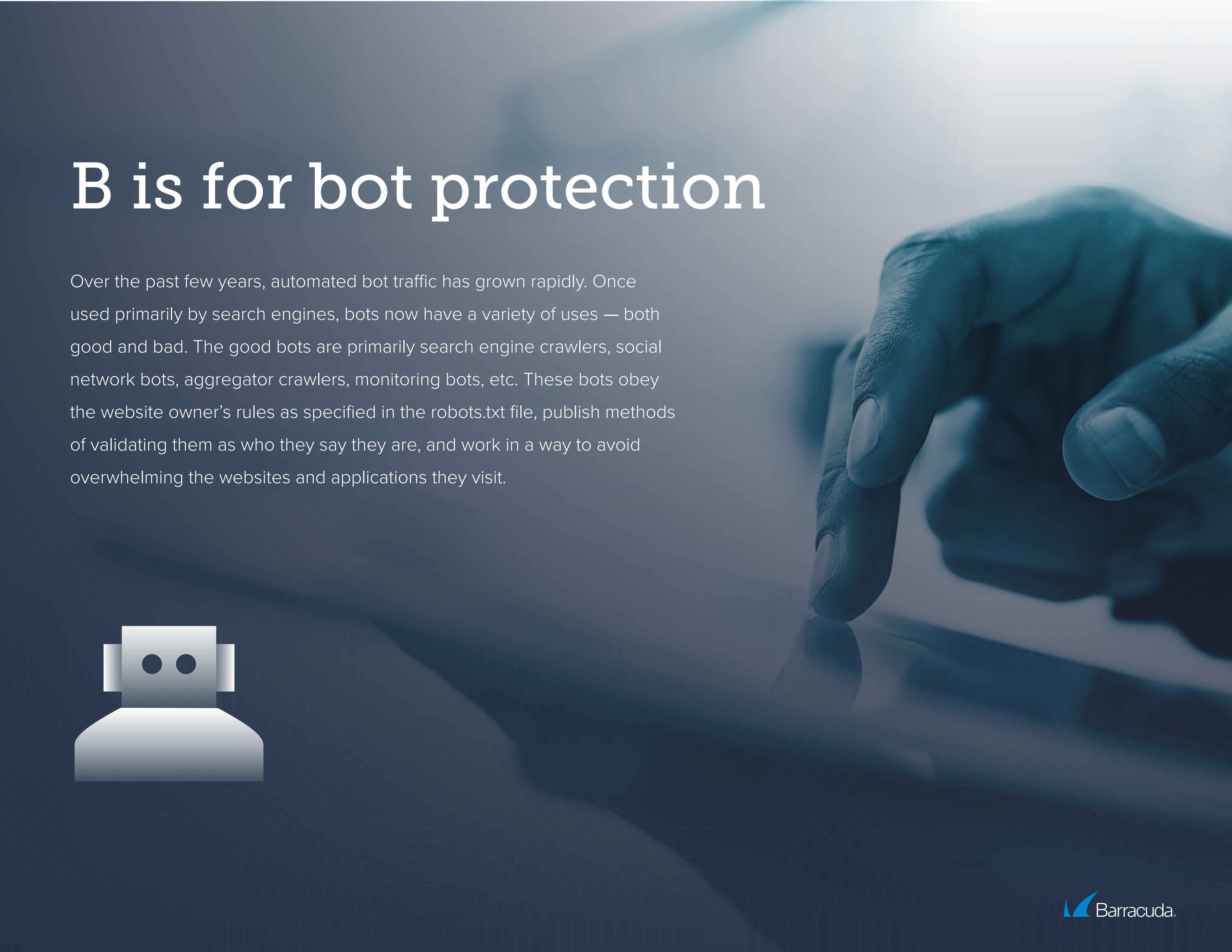
What are the main challenges your organization experiences when deploying APIs?

(n=728)



The public sector is the least likely to encounter challenges with API deployments according to their responses. Interestingly, their top problem is "lack of understanding of API standards," and API security is a distant third, in contrast with the other sectors that near universally consider API security to be the biggest concern. In comparison, energy, oil, gas, and utilities, and manufacturing lead the rest of the pack in their concerns about API security.

Unsurprisingly, respondents from financial services organizations were the most worried about uptime concerns among the various sectors. Respondents from the energy, oil, gas, and utilities industry were the least concerned about standards compliance, which is in a way concerning, given that standards compliance while building APIs enables better security.



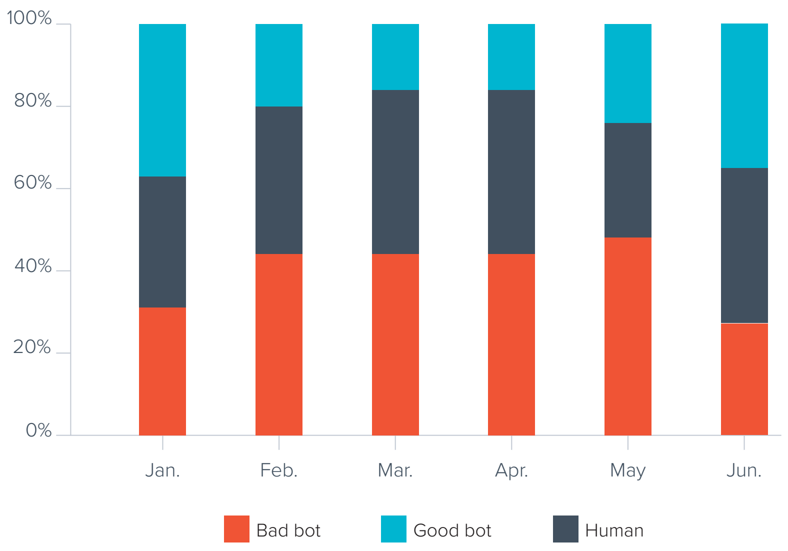
B is for bot protection

Over the past few years, automated bot traffic has grown rapidly. Once used primarily by search engines, bots now have a variety of uses — both good and bad. The good bots are primarily search engine crawlers, social network bots, aggregator crawlers, monitoring bots, etc. These bots obey the website owner's rules as specified in the robots.txt file, publish methods of validating them as who they say they are, and work in a way to avoid overwhelming the websites and applications they visit.

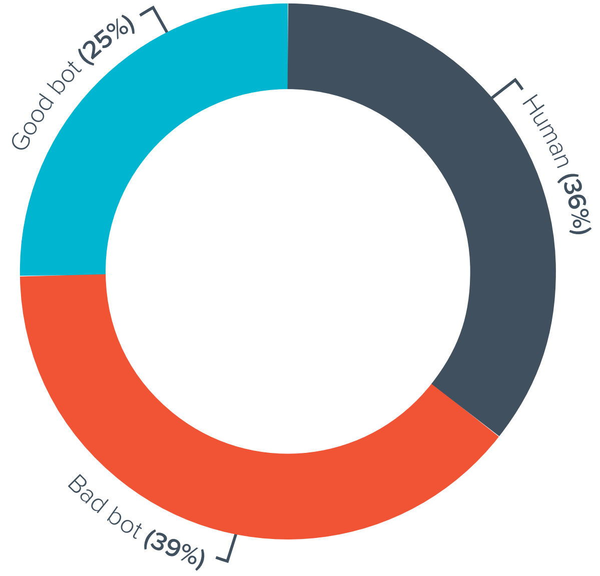
Bad bots are built to perform various malicious activities. They range from basic scrapers that try to get some data off an application (and are easily blocked) to advanced persistent bots that behave almost like human beings and look to evade detection as much as possible. These bots attempt attacks such as web and price scraping, inventory hoarding, account takeover attacks, distributed denial of service (DDoS) attacks, and much more. Bad bots make up a significant part of website traffic today, and detecting and blocking them is of critical importance to businesses.

Automated traffic makes up nearly two-thirds of internet traffic, as measured by Barracuda technology over the first six months of 2021. Roughly 25% of this traffic is from known good bots like search engine crawlers, social network bots, and monitoring bots.

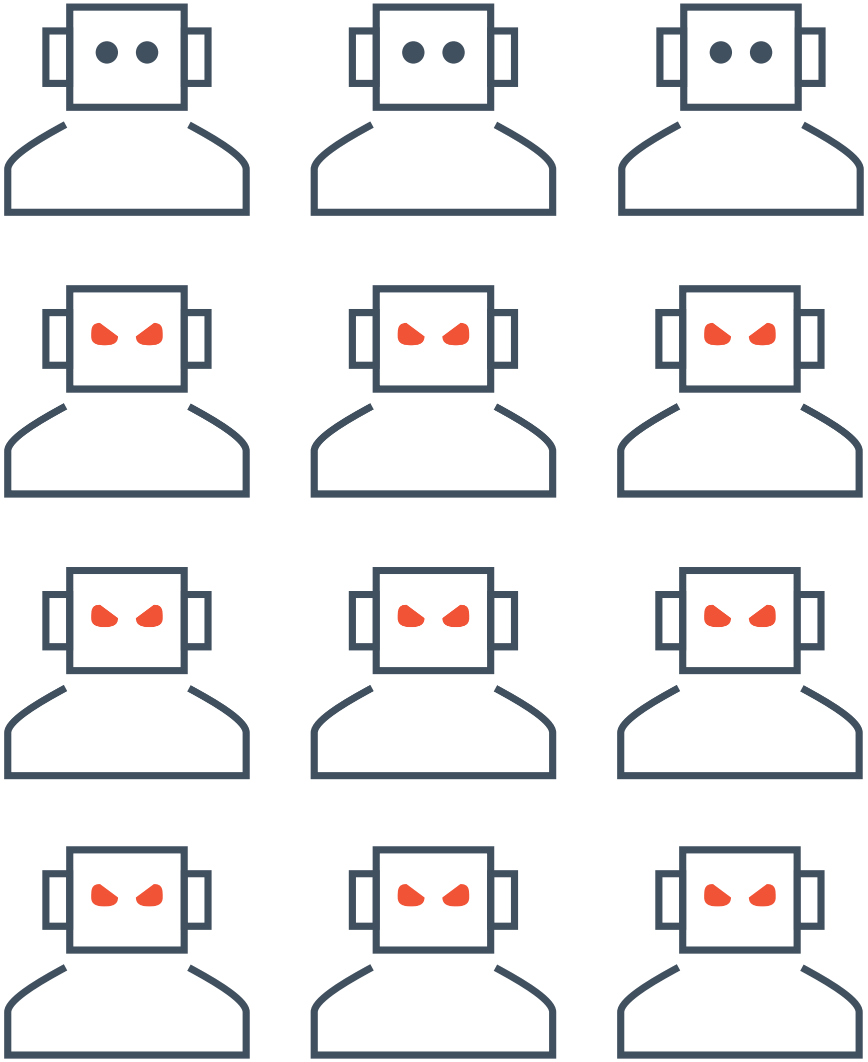
Distribution by month



Traffic distribution: Bots vs. humans (January – June 2021)



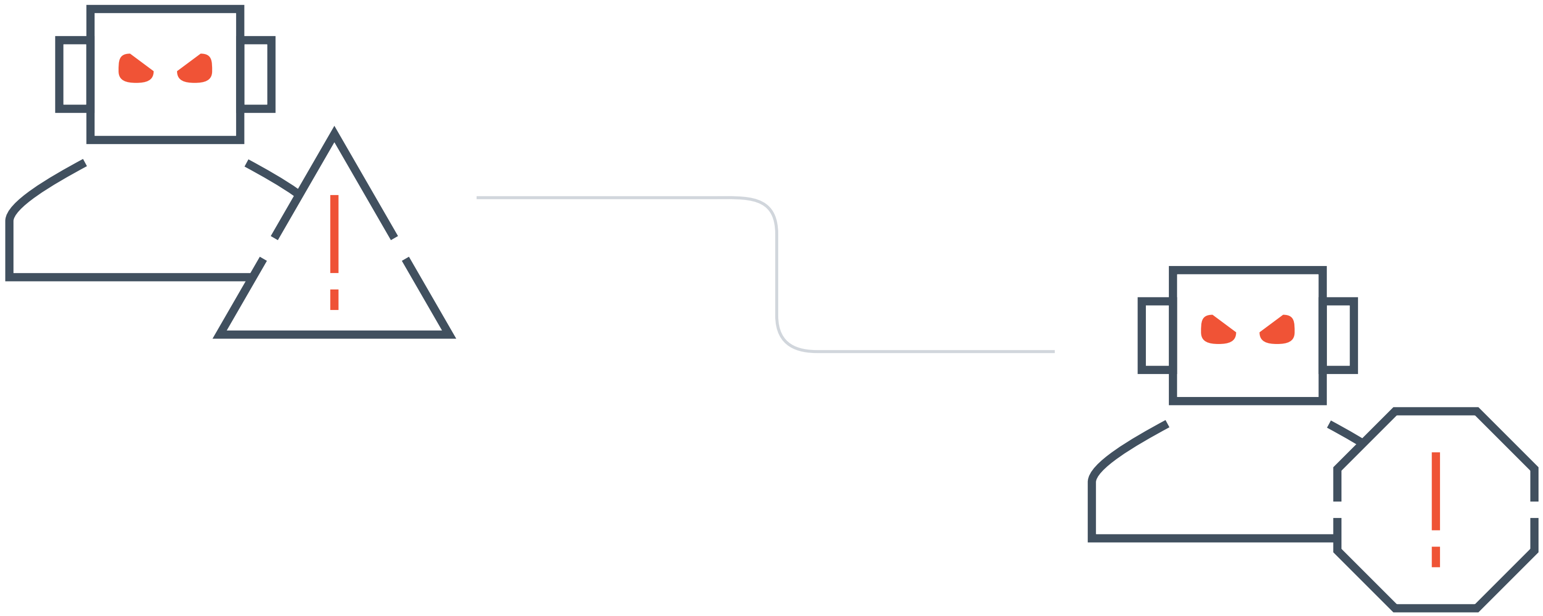
Today, bots are highly sophisticated and can be almost human in their behavior to bypass most defenses. The standard defenses employed to block them, primarily Google reCAPTCHA, is in no way a problem for them. In fact, the image-based CATPCHAs are easier to solve for bots than they are for humans. There is an entire ecosystem built around bots — from the people who build these intelligent bots, to services that provide "high-reputation" Google accounts to bypass CAPTCHA, to services that offer residential IP addresses (or Resis) to bypass IP reputation blocks, to escrow services that prevent bot purchasers from being scammed. With an increasing number of people turning to bots to make a quick buck, like the PlayStation 5 scalping in December 2020, bots are becoming mainstream and a big problem.



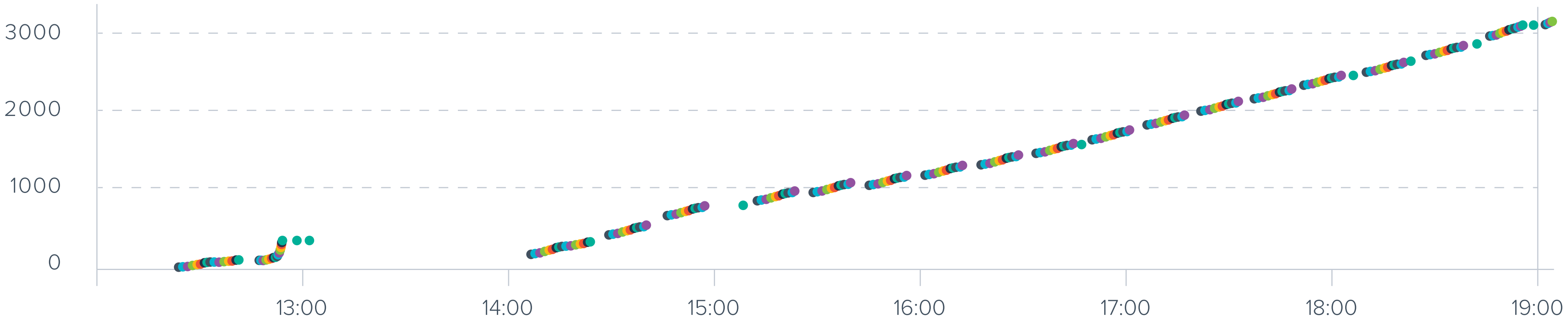
Example: Price scraping an e-commerce store in Eastern Europe

Barracuda detected and stopped a price scraping attempt on an e-commerce store based in Eastern Europe. The store was running a discount on Apple products, and there were some suspicious patterns of behavior in the traffic. The suspicious traffic came with standard browser clients, through multiple local residential IP addresses. However, these local IP addresses were from VPS hosting providers, and each client would only access a standard set of pages.

The attackers were caught using this correlation — and the price scraping attempt was stopped.

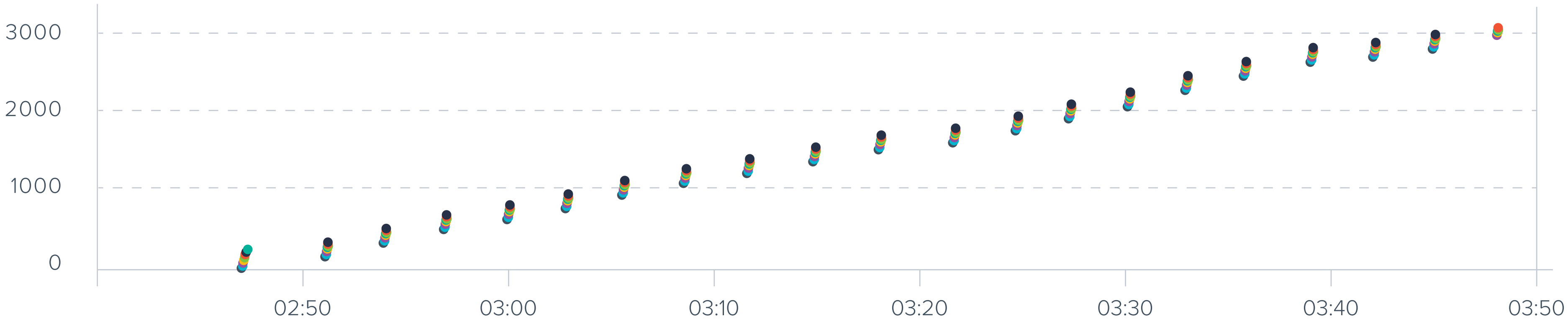


Repeating pattern of a price scraping bot



Bots were accessing the same set of product URLs multiple times in an hour after the initial burst was blocked.

Bot changing patterns to try to avoid detection

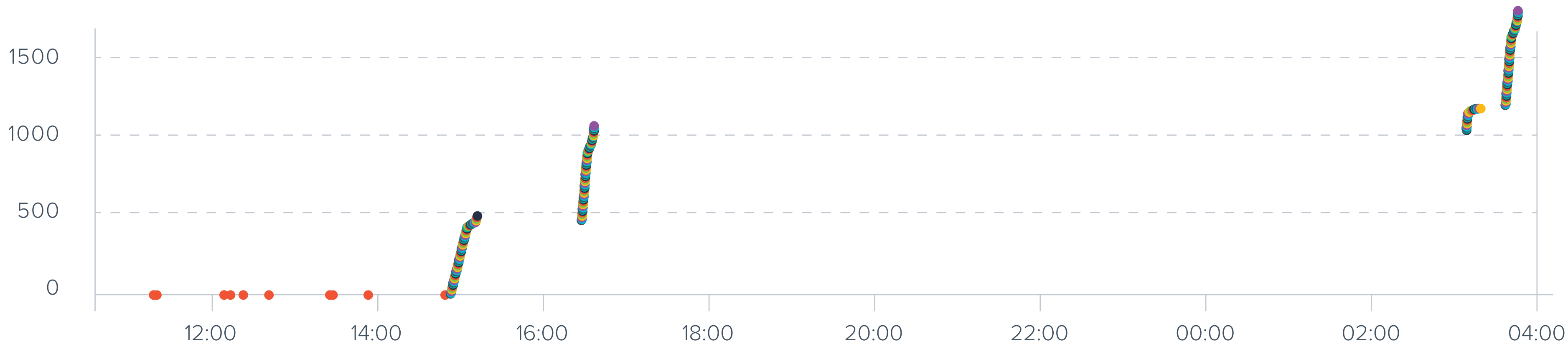


Bots attempting to access a smaller set of product pages in a different browsing pattern multiple times an hour.

Example: Attempting to overwhelm the login portal of an Indian manufacturing company

The login portal of an Indian manufacturing company was seeing unusually high traffic. The traffic was coming in primarily from mobile networks, which was unusual, but not unexpected for this website. However, on further analysis, the system determined that the incoming traffic was more likely from a desktop browser that was impersonating a mobile device while connected to a hotspot. The multiple clients attempting to overwhelm this login page were blocked successfully, and the page response time came back to normal.

Spikes in traffic to login portal



The first few dots were a bot pretending to be human and spreading out its accesses. After that, there are clusters seen, and each dot represents a different client attempting to access the login page.

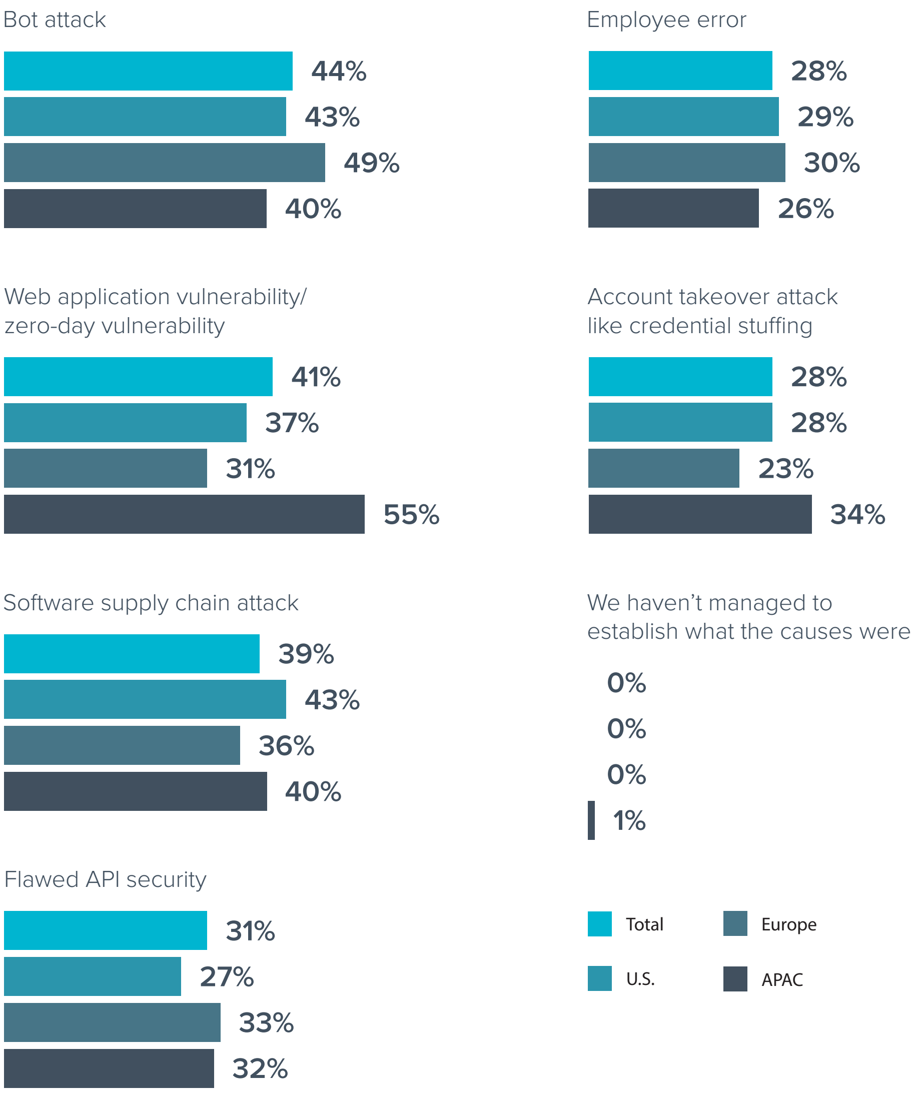
What application security professionals are saying

Bot attacks have been growing quite rapidly over the past few years, and this has led to some significant compromises. Two years ago, the biggest threats from bots were account takeover or credential stuffing attacks, and attacks were being publicized frequently, as were public disclosures of “credential dumps” from sites such as LinkedIn.

According to Barracuda’s recent survey of application security professionals, bot-based attacks were the most likely contributor to successful security breaches resulting from application vulnerabilities in the past 12 months.

Which of the following contributed to the successful security breach that exploited a vulnerability in one of your organization’s applications in the last 12 months?

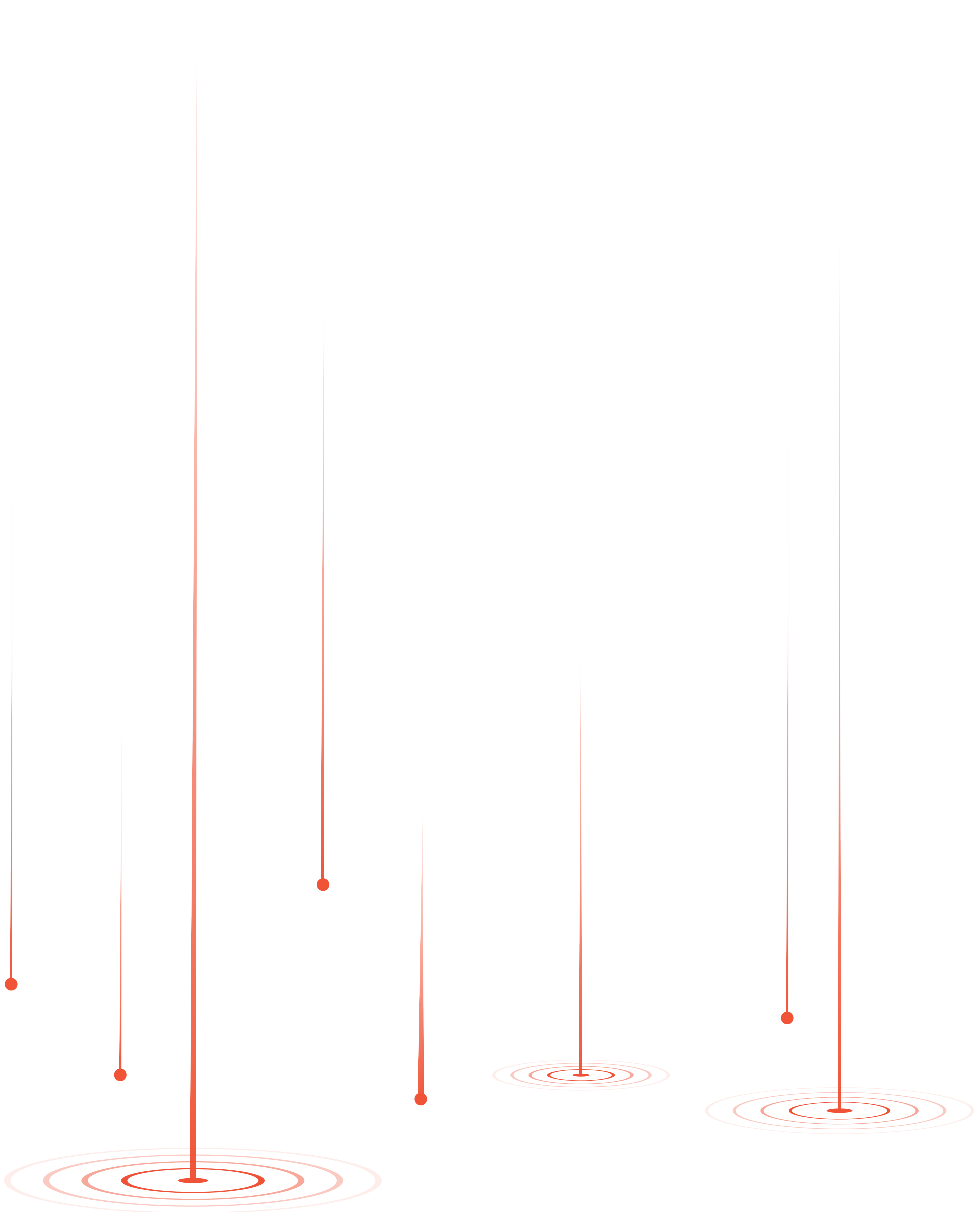
(n=541)



The wide array of bot attacks that target applications makes it a struggle for defenders to block them.

With so much variety in this attack vector, it's no surprise that so many organizations are struggling to defend their applications against bots. While bot spam is more of a nuisance attack, it is often used as a smokescreen to hide something more malicious, so it must be dealt with, not ignored. Depending on its frequency and authenticity, bot spam can become tricky to defend against and can evolve from benign annoyances to operational problems quite easily.

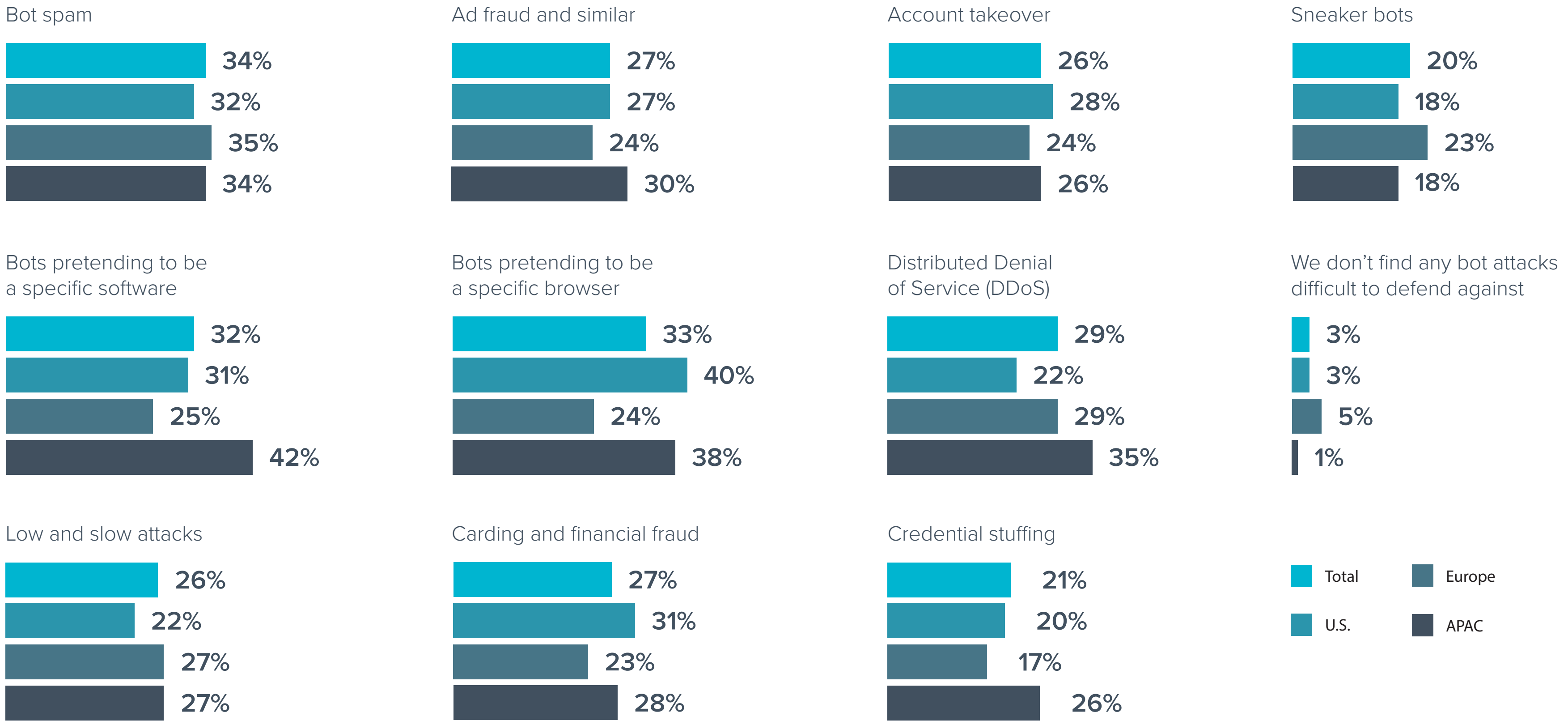
Bots spoofing browsers and apps are also major problems. These spoofs range from simple to complex, from a user trying to hide their real browser to bots running compromised versions of apps in click farms for ad-fraud or other malicious purposes.



Any of these bots used in conjunction with one another increases their chances of being successful. Multi-vector low and slow bot attacks are at the core of the problem and most likely contributed to successful breaches in the past year.

Which types of bot attack targeted at applications does your organization find it difficult to defend against?

(n=750)



Bot spam

Ad fraud and similar

Account takeover

Sneaker bots

Bots pretending to be a specific software

Bots pretending to be a specific browser

Distributed Denial of Service (DDoS)

We don’t find any bot attacks difficult to defend against

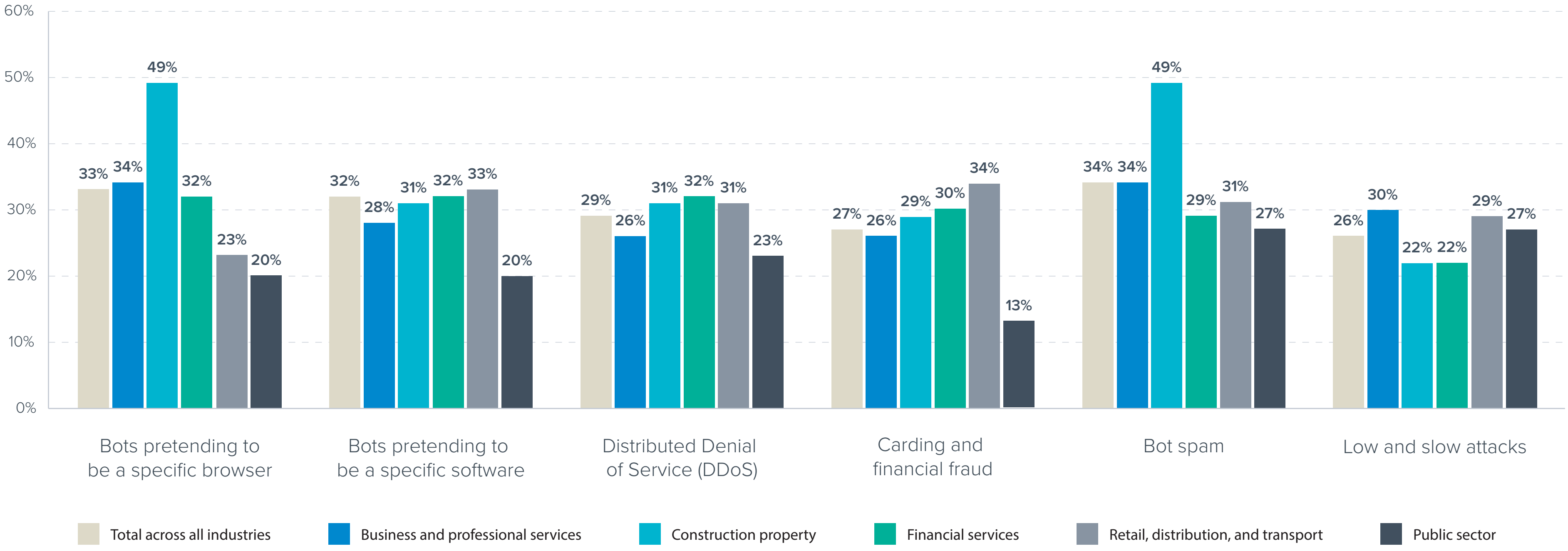
Low and slow attacks

Carding and financial fraud

Credential stuffing

Which of the following types of bot attack targeted at your organization's applications do you find it difficult to defend against?

(n=750)



For respondents from the financial services industry, three types of bot attacks tied as the most challenging to defend against: DDoS, bots pretending to be a specific software, and bots pretending to be a specific browser. These types of spoofing become a problem for financial applications, and attackers use cracked versions to perform malicious actions against these organizations. DDoS means significant financial losses simply because these systems are unavailable. Carding and financial fraud ranking second on the list is also quite interesting because you would expect it to be the top threat. This shows the prevalence of apps or browser-based access is massive for financial organizations, and it continues to grow.

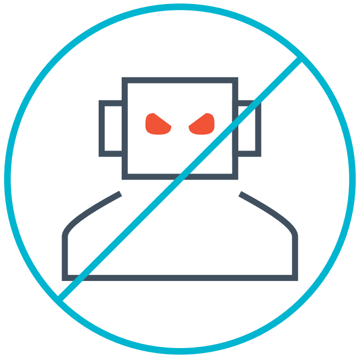
In general, bots pretending to be specific software or browsers are in the top five challenges for most sectors. The other interesting response is the construction and property sector, and the public sector are quite worried about bot spam. This suggests real estate sites may be seeing a lot of spam in their listings. Public sector spam is also a big concern. For instance, a few years ago the amount of "spam" entries in the FCC's net neutrality discussions were a concern.

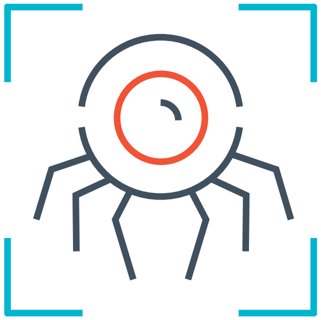
Public sector, retail, business, and professional services are the most worried about low and slow bots. Public sector organizations often have a lot of file downloads, which are ungated and become regular targets for attackers trying application DDoS attacks. Retail organizations also have a lot to lose from low and slow bots trying various attacks like account takeovers, price scraping, scalping, and much more.

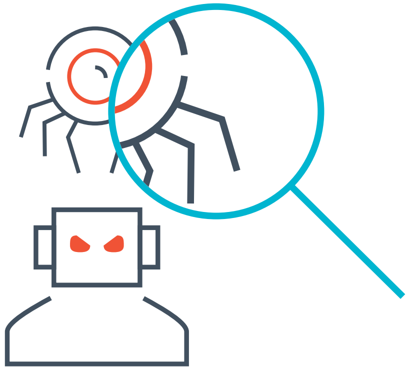
Preventing, detecting, and identifying bots will be critical functionality for vendors that help organizations defend against bot-based attacks.

Preventing, detecting, and identifying bots will be critical functionality for vendors that help organizations defend against bot-based attacks.

Whether organizations are struggling with spam bots, fraudulent bots, or bots that can spoof software and browsers, they urgently require improvements when it comes to defending against this attack vector. After all, bots were the most prolific contributor to successful breaches against applications in the past year. According to survey respondents, there are three clear areas that would be among the most desirable when choosing a security solution to fend off bots: fraud prevention, spam detection, and bot identification.



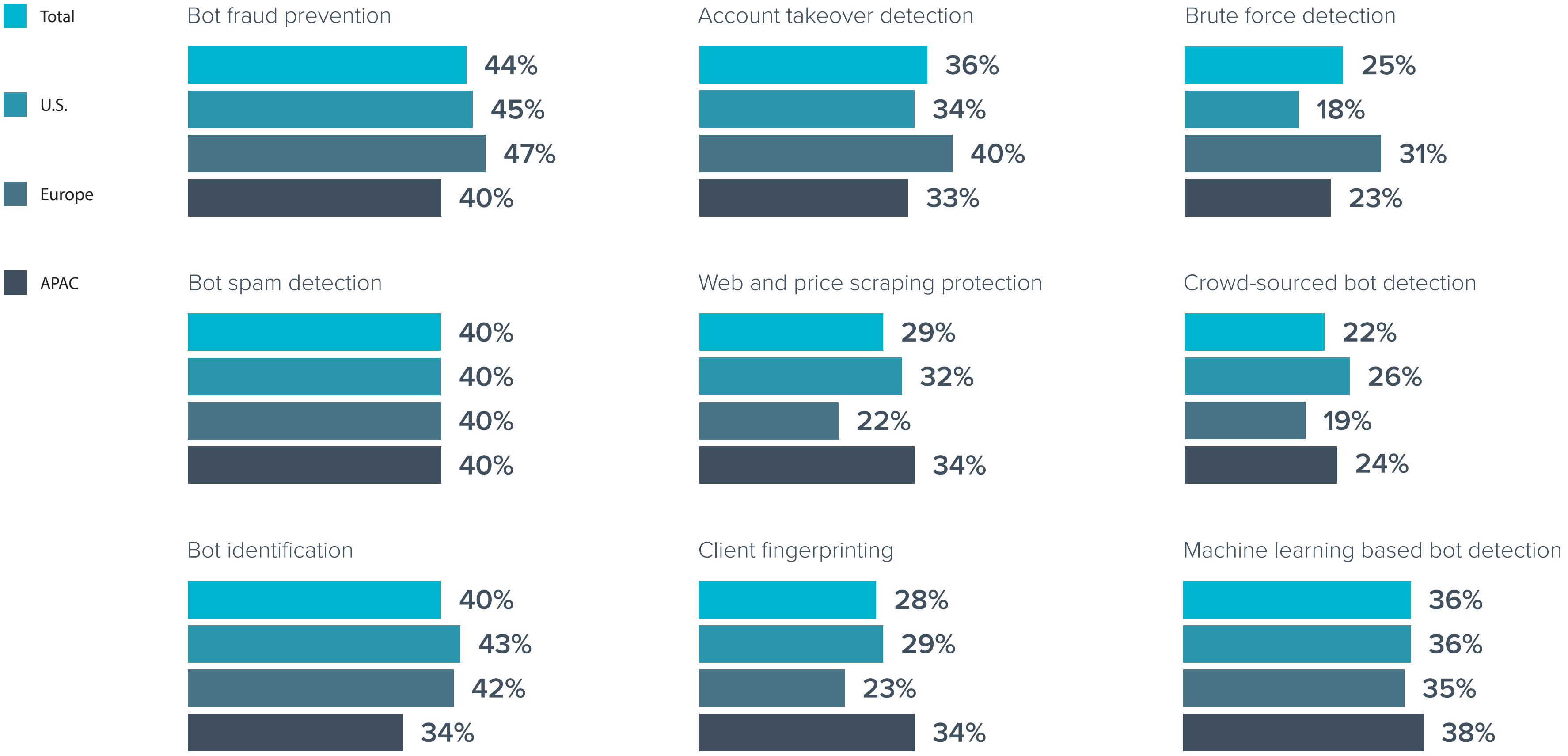




These types of features would be critical for defending against most attack types, but any vendor that can provide these characteristics within a single solution will be improving the existing bot protection capabilities of most organizations beyond measure.

Which features would be most important to your organization when choosing a security solution to help defend applications against bot attacks?

(n=750)

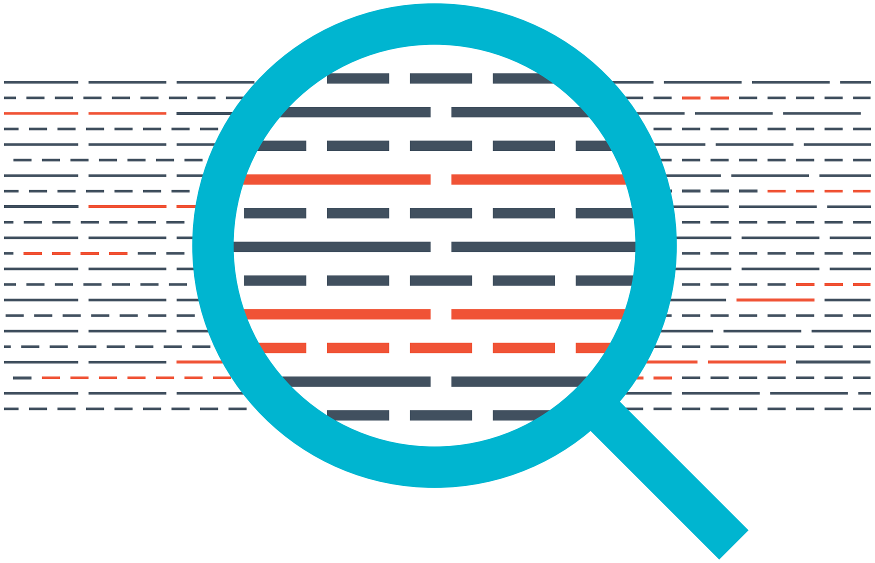


C is for client-side protection

Over the years, a multitude of novel, web-specific vulnerabilities have emerged, such as clickjacking and cross-site scripting (XSS), many of them manifesting on the client side. And, unfortunately, injection vulnerabilities that appeared on the client side more than a decade ago are still alive and well.

Client-side attacks, also known as supply-chain attacks or Magecart attacks (named for the Magento shopping application that was originally targeted) are difficult to detect and block.

Since the beginnings of the web in the late 1980s, apps have evolved continuously to satisfy our insatiable appetite for everything internet. This change has happened not only on the server side, but also on the client side (aka the browser). Just as dynamic content replaced static content, single-page applications replaced simple JavaScript-based rendering with an experience more suited to scrolling on phones or tablets. As an increasing amount of the application logic moves to the client side, attackers have turned their attention to the client side as well. Much of this client-side logic is implemented using either open source or other third-party code — and in the process, security falls by the wayside.



So why do developers use third-party code? Well, to put it simply, the modern web would not be possible without this. Modern web pages comprise dozens or even hundreds of third-party or fourth-party external scripts. Tools such as webpagetest.org can be used to see the surprising number of third-party scripts on any given webpage. This is an accepted approach in web development because the alternative is unthinkable: reinventing thousands of lines of code. The problem is one of trust: A script that is good today may be hacked tomorrow. Attackers target the sources hosting this third-party code because their hack will transform every application using that code into a victim.

Since the third-party code is the one that is maliciously modified, most application owners do not realize that the scripts have been compromised until much later in the cycle. The scripts themselves are loaded from other sources like CDNs and code repositories, and they are typically not delivered directly to browser from the website, which means detecting and stopping them is difficult with current tools and practices.

Example: British Airways supply-chain attack

In 2018, a supply-chain compromise resulted in the data of between 380,000 and 500,000 British Airways customers being breached. The breach resulted in the loss of personal and payment information for those affected.

The breach was one of the highest profile Magecart attacks at the time. Magecart was a group that were first seen in 2016 skimming card details online. They injected scripts that would specifically steal the data from online payment forms, and then they would either use the stolen data themselves or sell it to other cybercriminals.

In the case of British Airways, the Magecart group modified a specific JavaScript used on their web and mobile apps called Modernizr. Within this script, they embedded a small function that would run at the end of the script. When this function ran, it collected data filled into the payment form and sent it out to a third-party website the criminal group operated. This resulted in massive data exfiltration, and British Airways was fined £20 million, which at the time was the largest ever fine in the UK (reduced from the original £183.39 million due to the economic impacts of the pandemic on the airline and travel industries.)

Customers affected by **data breach**

380-500K

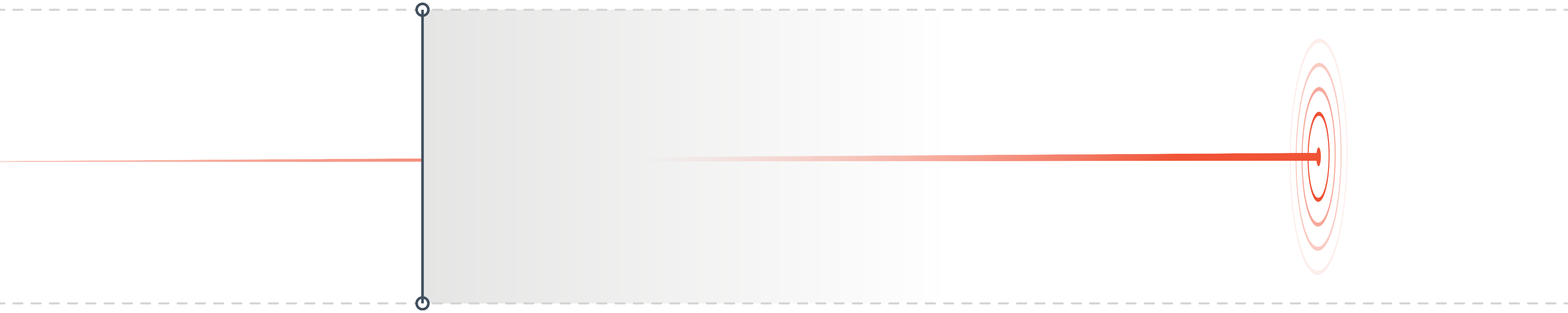
British Airways **fined**

£20 million

Example: Visa warns of online skimmer

In September 2020, Visa issued a warning over a new online skimmer called Baka that was performing client-side skimming attacks. The skimmer had some interesting mechanisms to prevent detection. The skimmer loaded dynamically into the client machine's memory at the time of execution — this meant it could not be detected by standard scanning or page inspection. It was designed to run only from memory, so that no traces of it would be found in the storage of the browser. The skimmer creators took great efforts to ensure that it was fully encrypted and difficult to identify when running on a website or application.

At the time of the notification, the skimmer was in active use on many online stores, and it was clearly built by skilled developers who designed it to evade detection as long as possible.



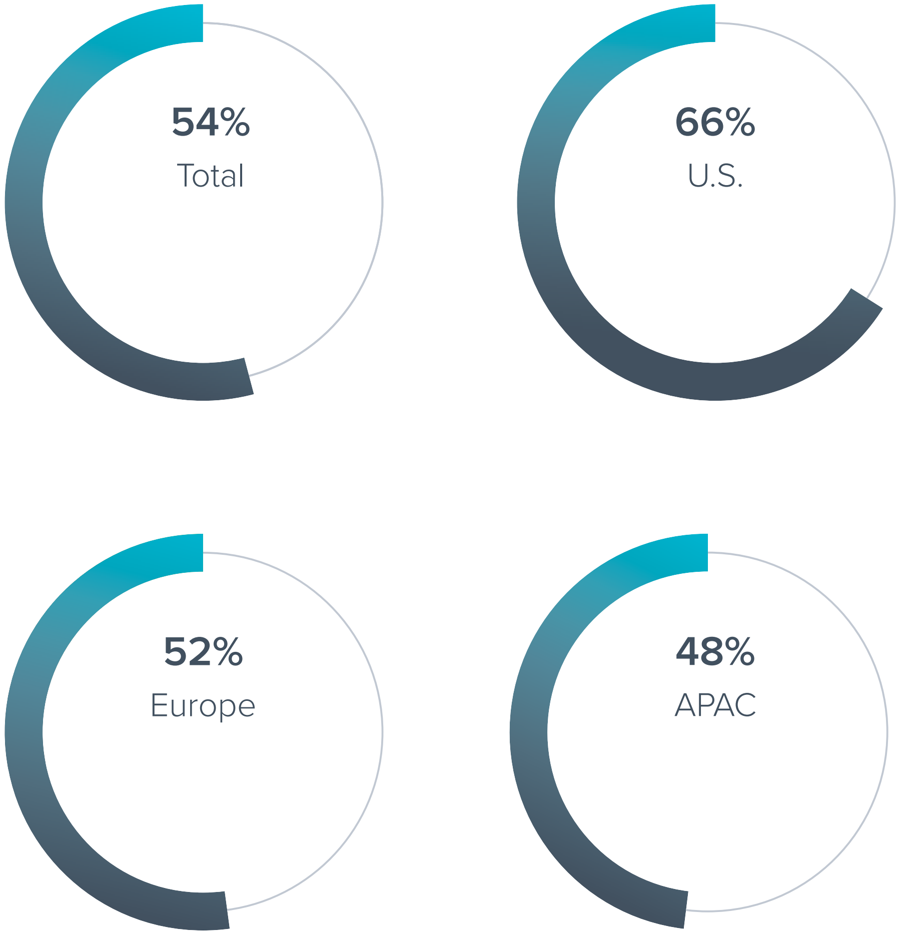
What application security professionals are saying

The use of third-party scripts for web applications is widespread, with organizations using varying methods to deliver scripts to a browser.

The drive for efficiency when it comes to application development is once again evident in the responses to Barracuda’s recent survey of application security professionals, with more than half of organizations using ready-made third-party scripts for web applications. Security should be a big concern when using third-party code. This is especially true when code is being delivered to a browser directly from the source platform, such as GitHub. If the code has been tampered with, a software supply-chain attack, such as Magecart, could be just around the corner. Organizations should be wary of this approach to application development.

Approximately, what percentage of your organization’s web applications use third-party scripts?

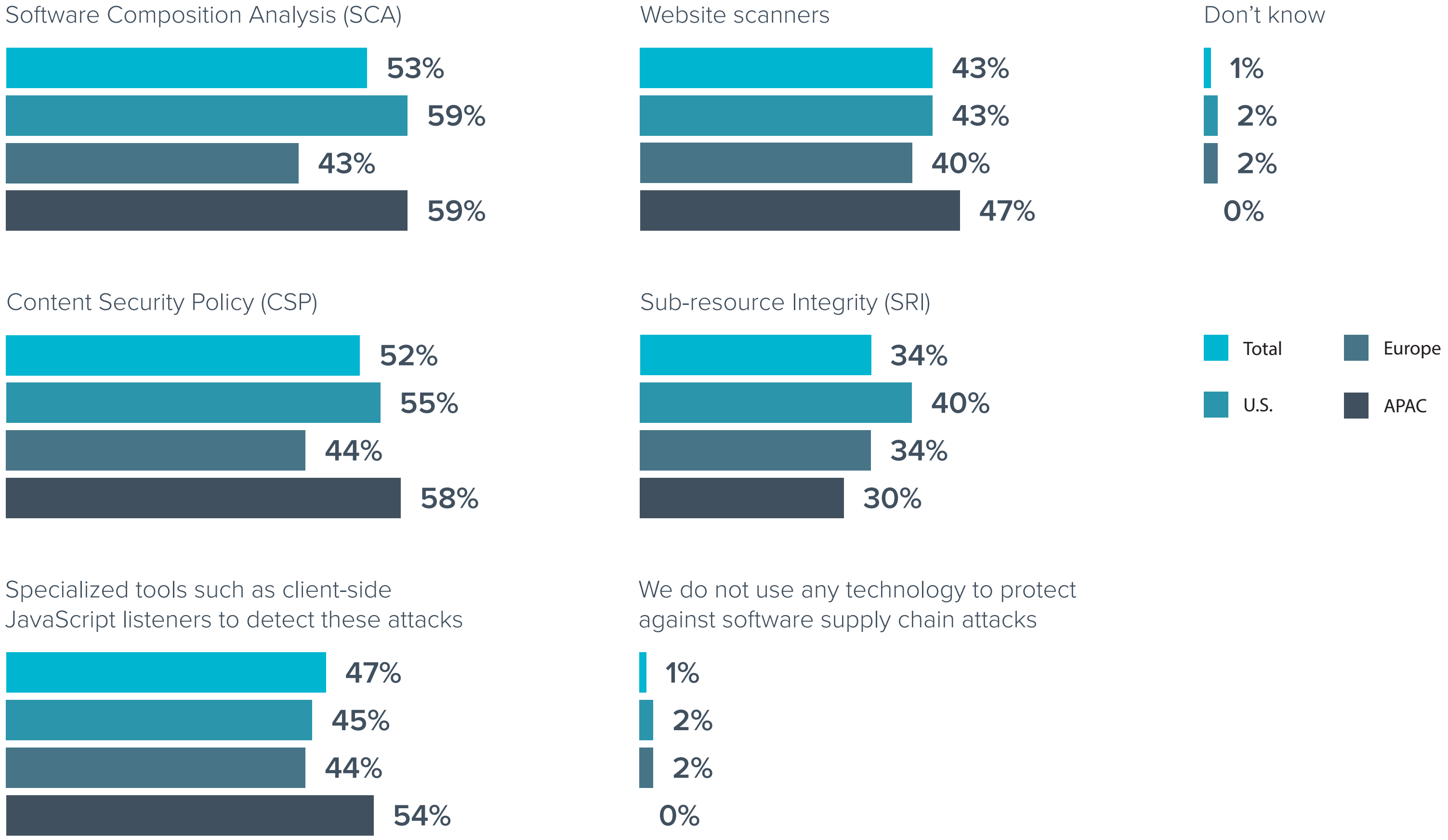
(n=750)



Relatively standard protections are in place for supply-chain attack mitigations. APAC respondents use specialized tools, including client-side JS listeners, to detect attacks more than other regions. Such listeners have a better chance of detecting the more advanced attackers than website scanners. Website scanners are the fourth most popular technology on this list, but they are easily spoofed, as evidenced by the Baka skimmer detected by Visa. Sub-resource integrity (SRI) is difficult to set up and maintain, which could be a reason for its low popularity.

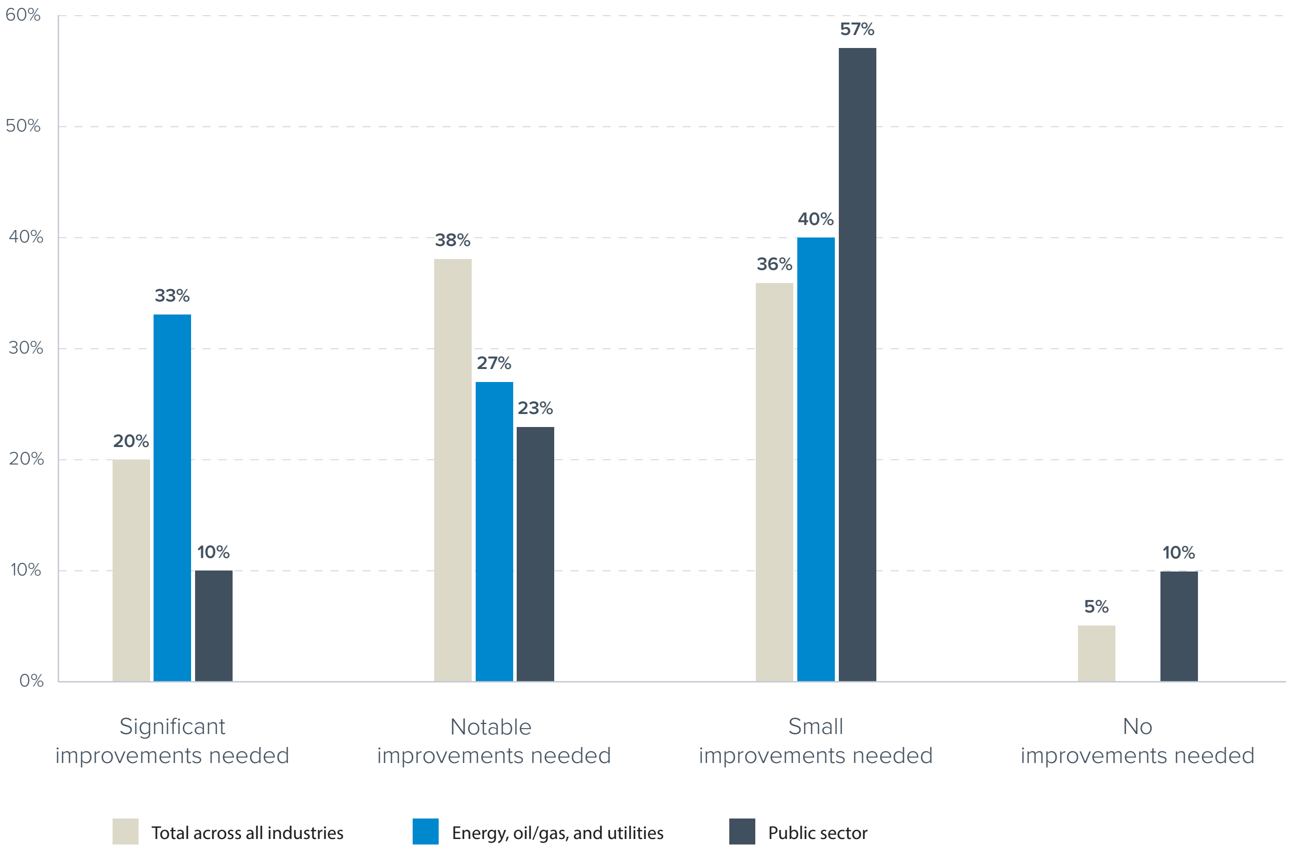
Which technologies does your organization use to protect against software supply chain attacks?

(n=750)



What level of improvement do you believe is needed in your organization when it comes to defending against software supply chain attacks?

(n=728)



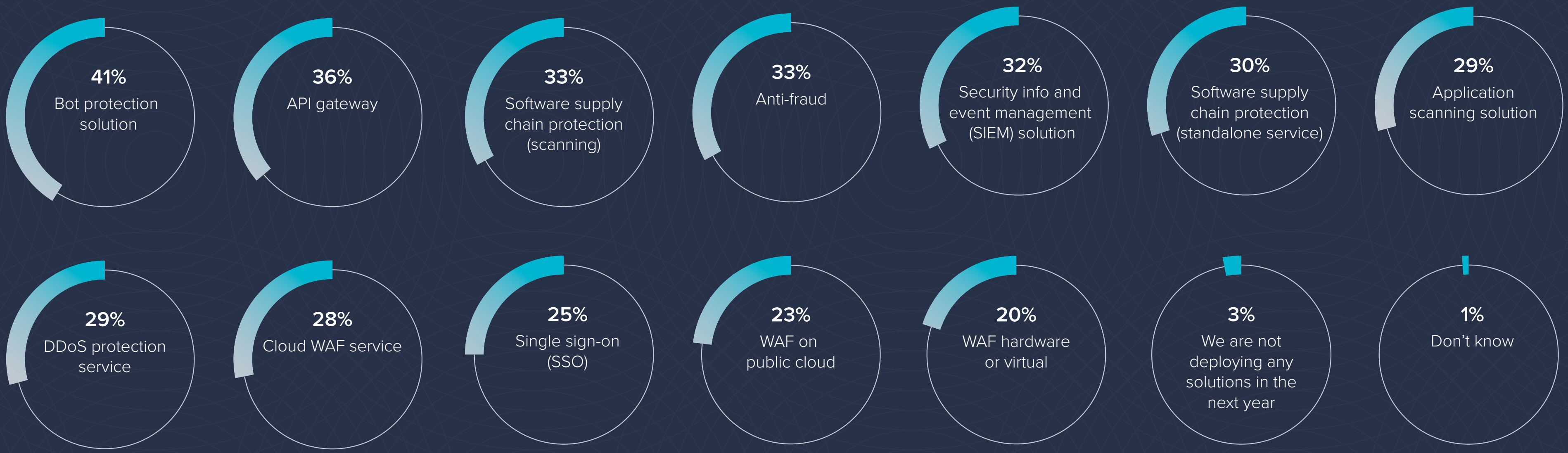
Most organizations are split on the improvements that are required for their website supply chains. Respondents working in the public sector were the most likely to state that small improvements or no improvements are needed for their protection. Only the energy, oil, gas, and utility sector indicated that significant improvements are needed. This is more a reflection of the attack vector only emerging relatively recently and the impact not being fully understood. As more of these attacks come to light, the attack vector will become much more prominent.

Conclusion: Preparing for the new ABCs of application security

Organizations are getting breached more than ever through their web and API applications. As newer technologies proliferate, attackers work to identify ways to bypass their security measures and breach them. APIs, bot attacks, and client-side attacks are the latest ways they are working to breach applications for fun and profit.

Which of the following solutions will your organization be deploying in the next year?

(n=750)



In our research, we find that organizations seem to understand this, with many looking to deploy new solutions in the coming year, such as bot protection (41%), API gateway (36%), and software supply chain protection (scanning) (33%).

It is a good sign that organizations are moving to fill these gaps, but the more solutions they add, the more complex application security becomes. To provide effective protection, an application security solution needs to be a platform that is capable of protecting customers against all of these attack vectors. A platform approach to application security provides powerful protection against both traditional and emerging threats while remaining easy to use and manage.

About Barracuda

At Barracuda we strive to make the world a safer place. We believe every business deserves access to cloud-first, enterprise-grade security solutions that are easy to buy, deploy, and use. We protect email, networks, data, and applications with innovative solutions that grow and adapt with our customers' journey. More than 200,000 organizations worldwide trust Barracuda to protect them — in ways they may not even know they are at risk — so they can focus on taking their business to the next level. For more information, visit barracuda.com.

Barracuda®

The new ABCs of application security • US 1.0 • Copyright 2022 Barracuda, Inc. • 3175 S. Winchester Blvd., Campbell, CA 95008 • 408-342-5400/888-268-4772 (US & Canada) • barracuda.com