THREATX

WHITEPAPER

The Role of Bots in **API Attacks**

Defending Against Bot-Based API Attacks

APIs have come a long way since the Remote Procedure Calls (RPCs) of the early 1980s. Today's RESTful APIs enable developers to integrate systems much faster and more easily. As a result, APIs are proliferating, and threat actors have taken notice of these small programs that expose business logic.

Technology does not evolve in a vacuum. The tools and techniques attackers use have also evolved to make adversarial hacking faster and easier. With the help of massive, complex networks of bots — or botnets — threat actors can automate and orchestrate large scale API-based attacks while evading detection and consuming compute resources to distract security analysts.

These are not your old-school spammer's botnets, and APIs are not your run-of-the-mill web apps. You cannot protect APIs if you can't detect and stop botnets. These are sophisticated attacks that require a sophisticated defense.

What's a Bot?

To comprehend the role bots play in an API-based attack, it's important to understand what bots are and how they work.

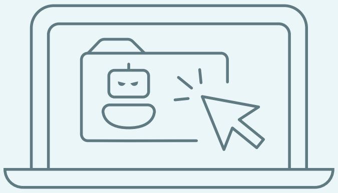
A BOT IS...

An autonomous software robot used for machine-to-machine communication. A bot can be any machine, but it looks like a unique user as represented by an IP address.

Botnets

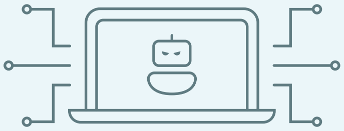
A single bot can perform one task at a time. Many bots working together can do many tasks at a time. When a collection of bots are used together, they form a botnet. A botnet can consist of hundreds of thousands of IP addresses. These IP addresses can be purchased from the dark web or hijacked from legitimate users.

1



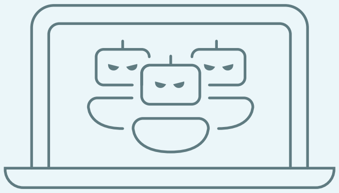
A bot takes over when an unsuspecting user downloads a malicious file that gives an attacker control over their computer.

2



The machine acts as a node in the attacker's botnet and is used, unbeknownst to the user, to execute any step of an attack.

3



Threat actors compromise large companies, universities — even home routers — to acquire massive amounts of IP addresses they can leverage with bots.

BOT USES

Not all bots are used for nefarious purposes.

Search engines, for example, crawl the Internet as bots to index their search patterns. However, a great number of bots are used maliciously.

> In fact, over 20% of all traffic to web applications comes from malicious bots.
>
> *Source: ZDNET*

These bots range in sophistication from basic crawlers and scrapers after content and pricing data, to bots emulating real browsers and performing more advanced attacks like account take over, credential stuffing, and credential manipulation and escalation. Bots are also used to find and exploit vulnerabilities like excessive data exposure, security misconfigurations, and authorization and authentication gaps.

BOTS ARE HERE TO STAY

Botnets allow threat actors to work smarter, not harder, and for that reason they are here to stay.

In the art and craft of software engineering, scripting automates an otherwise manual task.

Threat actors use bots to automate attacks by scripting bots to perform various hacking techniques, thereby increasing the efficiency and scale of what skilled attackers do manually.

Attackers can both scale out their efforts and achieve their objectives faster, doing more in less time — and giving security organizations less chance of detecting the activity. Bots also allow threat actors to increase their capacity, simultaneously carrying out multiple attacks.

The Next Generation of Bot Attacks

Like APIs, bot attacks have been around for a long time and are becoming more sophisticated.

For example, attackers leverage advanced malicious bots to look like human web application traffic. They can hide behind a network of tens of thousands of anonymous proxy servers and compromised systems to widely distribute the sources from which their traffic is seen, defeating volume-based detection. Advanced bot attacks may also use headless browsers, which can easily defeat user-agent based detection and include other features to fool legacy web application firewalls (WAFs) and web applications into thinking they are in fact a normal human user.

Mimicking human behavior helps bots evade detection.

A single IP address coming in and hitting every URL or path is easy to detect — it's a scanner. But when distributed across multiple IP addresses, the same activity appears to be different users in different parts of the application.

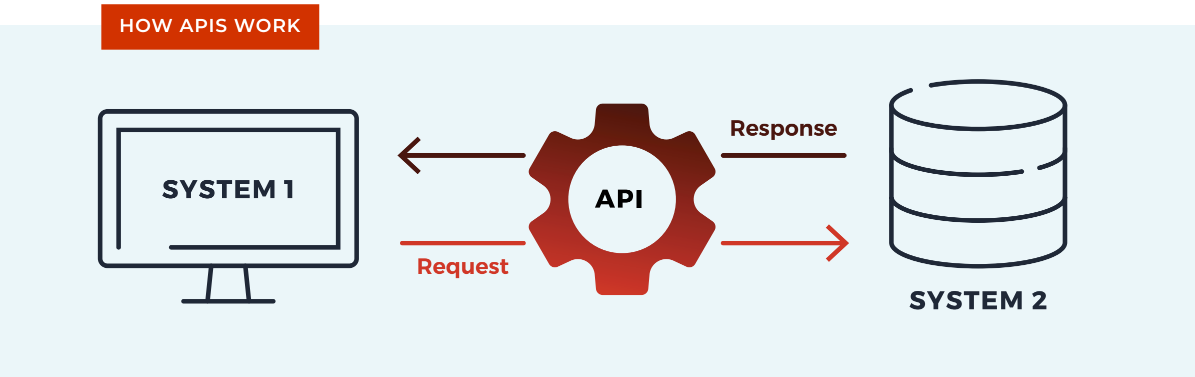
For example, if a threat actor wants to identify the presence of a vulnerability in an organization's environment, their efforts are less likely to be noticed if test conditions are sent from 500,000 IP addresses versus a single IP address. Attackers strategically distribute the attack across a botnet to stay below detection thresholds, rotating IP addresses and reusing them only after a safe period of time.

Why Bots Are Used Against APIs

Bots and botnets clearly give attackers an advantage when attacking APIs, but why attack APIs in the first place?

APIs exist to expose high-value functionality that can be easily integrated and accessible to multiple clients (users). APIs work by connecting various applications so they can share data. For example, an insurance app may use an API to send a patient's coverage information to a healthcare provider system. APIs aren't new, but they have evolved to become easier to use. The whole point of the RESTful design pattern is that the resources are discoverable. Submitting a request provides the public interface and the parameters and methods that they expose.

HOW APIS WORK



In addition to exposing valuable business logic, APIs are attractive targets because they can be weak points in the system. Within the software development lifecycle, APIs tend to be less scrutinized than traditional web business logic because organizations often struggle with visibility into their API landscape. However, just like any piece of code, APIs are susceptible to vulnerabilities. The Open Web Application Security Project (OWASP) has even established a Top 10 list dedicated just to API-related vulnerabilities.

APIs also operate behind the scenes. Most organizations don't pay attention to how many APIs are in production, their intended purpose, and how they're actually being used. But when considered at scale, the risk footprint represented by APIs grows exponentially. As more users adopt or use the application, its APIs are called more frequently. Furthermore, developers tend to reuse APIs. Not only are they reused internally, but they may also be used by the business' partners and customers as well.

How Bots Are Used Against APIs

BOTS FOR RECONNAISSANCE

In an API-based attack, threat actors often use bots and botnets to conduct reconnaissance before executing an actual attack. Bots can do everything from web application mapping to actual identification of vulnerabilities. They may apply a known payload or an anomalous input and review the outputs to find areas that can be further exploited. At this stage, threat actors also use bots to test detection thresholds to avoid setting off alerts and drawing attention to themselves.

BOTS FOR ATTACK

When using bots to execute an attack, threat actors often take one of two approaches. They may target the vulnerable APIs identified during reconnaissance, or they may target all the APIs because they know what's behind them is lucrative. They spread the attack across different IP addresses, being mindful of detection thresholds. If they know three failed login attempts during a credential stuffing attack gets an IP address blocked, they'll stop after two failed login attempts and move to the next IP address.

BOTS FOR EVASION

Threat actors also use botnets to distribute attack traffic so that it maximizes resource utilization in a distributed denial-of-service (DDoS) attack. The goal here isn't to avoid detection, but rather to create a distraction while conducting a more insidious attack. Attackers make a point of triggering thousands of alerts so that they can exfiltrate data unnoticed.

How Not to Block a Bot

Simple bot attacks can often be mitigated with traditional WAF functionality. A bot scraping content isn't terribly difficult to distinguish from human users of the web application, for example. A group of IP addresses attempting to log in to your web application repeatedly or sending multiple malicious requests in a short period of time can be identified by volume alone.

Unfortunately, the botnets threat actors use today are more advanced and more difficult to detect.

In addition to mimicking user behavior, threat actors will leverage proxies, TOR nodes, and NAT'd networks as a further level of evasion. Everything from residential proxies to compromised IoT device networks are fair game.

Since there's a significant potential profit involved in an API-based attack, attackers can afford to distribute the attacks across tens of thousands of IP addresses, avoid IP addresses already flagged on various threat intel feeds, and develop extremely complex rotations in order to evade detection.

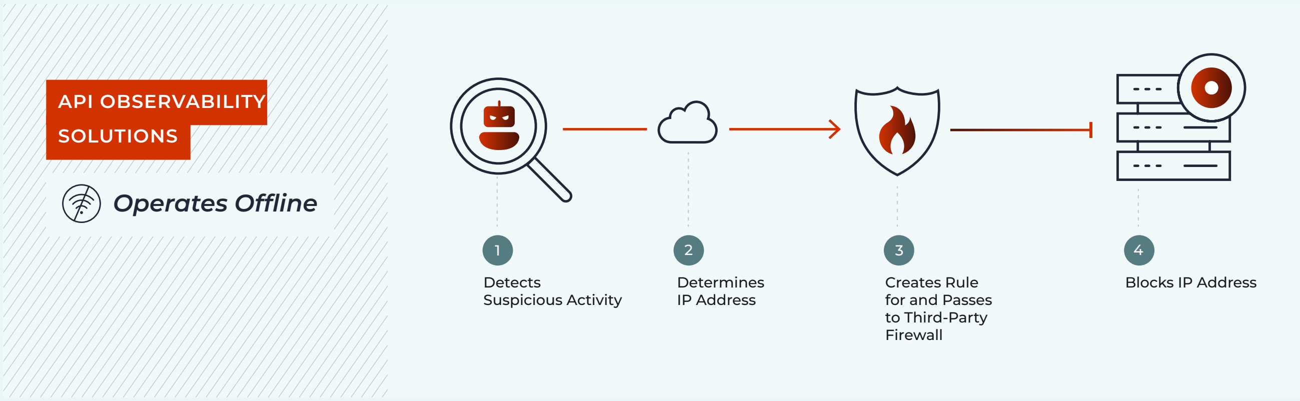
They also bypass traditional web-based authentication and authorization checks by targeting vulnerable APIs supporting web and mobile storefronts. Because there are multiple clients involved with an API call, it's sometimes unclear what "normal" API usage looks like. This increases the complexity of spotting suspicious activity and the risk of blocking legitimate users.

Legacy WAFs vs. Bots

Traditional WAFs rely on signatures to detect attacks, but API-based attacks have no clear signature. Attackers rarely, if ever, follow a linear path in their efforts to breach a target. Furthermore, legacy WAFs determine whether individual transactions are good or bad without considering the greater context. Without this big picture view, WAFs are incapable of recognizing bot-driven API attacks.

API Observability Solutions vs. Bots

API observability solutions are also limited in their ability to detect and stop API-based attacks because they operate offline. These tools may accurately identify an attack, but generally it's well after the fact. They can't interrogate the client IP address to determine if the user is really who they say they are or using multiple IP addresses to launch the attack. They can simply determine the IP address involved in that stage of the attack and then pass it to a third-party firewall solution to block the IP address. They can't, however, replicate the complex correlations and detection techniques required when leveraging a third-party firewall with basic capabilities based on static rules and signatures. API observability solutions also lack IP interrogation and fingerprinting techniques to correlate behavior across IP addresses.



API OBSERVABILITY SOLUTIONS

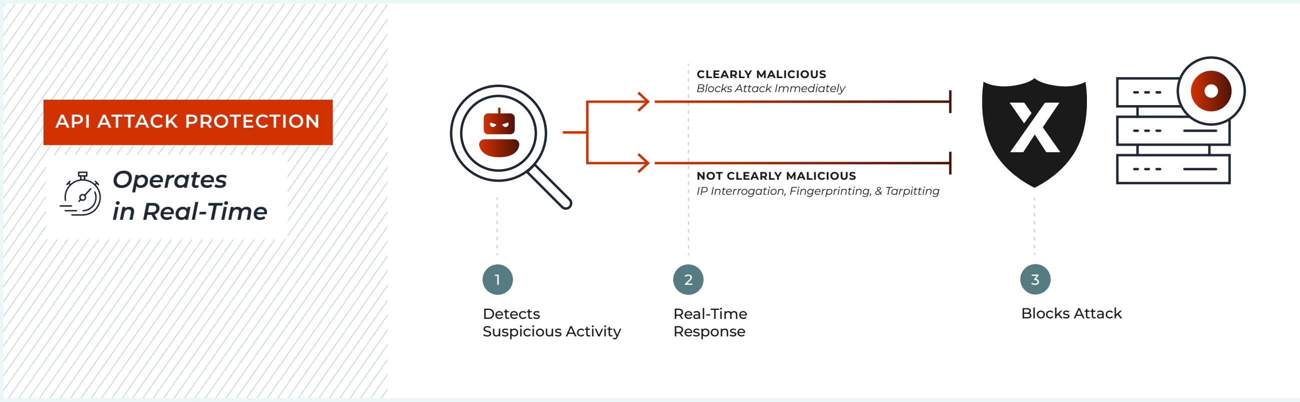
Operates Offline

1 Detects Suspicious Activity

2 Determines IP Address

3 Creates Rule for and Passes to Third-Party Firewall

4 Blocks IP Address



API ATTACK PROTECTION

Operates in Real-Time

CLEARLY MALICIOUS
Blocks Attack Immediately

NOT CLEARLY MALICIOUS
IP Interrogation, Fingerprinting, & Tarpitting

1 Detects Suspicious Activity

2 Real-Time Response

3 Blocks Attack

The Best Way to Combat Bot-Based API Attacks

Defending APIs Against Botnet Attacks

THE GOAL

1. Make it too inconvenient and expensive for threat actors to continue their attack
2. Simultaneously buy enough time to fix the application so that it's no longer vulnerable

THE SOLUTION

Combatting attacks is accomplished by detecting and blocking as many malicious IP addresses as quickly as possible so that threat actors eventually run out and can't use them against you.

When leveraging a botnet, attackers do everything they can to avoid creating patterns that can be detected. However, analyzing the HTTP request can uncover giveaways — identification markers or data points — that can be correlated to understand the full scope of an attack. By matching multiple data points within an HTTP request, it is possible to detect malicious activity with a high degree of certainty and significantly reduce false positives. APIs allow you to have stronger assumptions of what the machine-to-machine interaction should look like. So, if there is behavior that simply doesn't make any sense or you detect OWASP Top 10 threats like SQL injection or attempts to exploit known vulnerabilities, then it can be simply blocked.

Tracking and interrogating to identify bots

When it's unclear whether activity is malicious, then a multi-disciplined approach is needed. Real-time behavioral profiling looks at large volumes of contextual data, monitoring every request live from every user to characterize their behavior and map their intent. By seeing more transactions, the system can recognize a broader pattern much faster and automatically craft a complex behavioral signature to block the attack in real time. In addition to behavioral profiling, advanced threat engagement techniques, such as IP fingerprinting, interrogation, and tarpitting, help shed light on the "user's" intent.

Organizations also need DDoS protection to help shoulder some of the burden of the infrastructural load as well as access to experts who watch bot activity on a global scale and can help put API activity in perspective.

ThreatX vs. Bots

The ThreatX Platform protects APIs from all threats, including bot attacks, DDoS attempts, API abuse, exploitations of known vulnerabilities, and even zero-day attacks.

Rather than relying on a single, significantly risky event or a known signature, ThreatX identifies and blocks more threats, more accurately by analyzing behavior from multiple vantage points. In this way, ThreatX can correlate several behaviors back to one attacker and identify behavior that is suspicious but wouldn't be flagged by other security solutions.

ThreatX scans all inbound API traffic in real time. When it recognizes attacker behavior indicative of an API attack, ThreatX flags and watches that user. This real-time monitoring enables ThreatX to execute advanced threat engagement techniques, such as IP fingerprinting, interrogation, and tarpitting. When a series of user interactions indicate a certain risk threshold has been met, ThreatX blocks the user in real time.

Modern threats, modern solutions

Ten years ago, a vulnerability in a web application may have never been discovered by an attacker who had to manually test each and every path in search of a way in. The same can't be said for an API. Botnets allow attackers to work faster and more effectively, distributing and automating the attack across hundreds of thousands of individual bots. If there is a vulnerable API, a threat actor will find and exploit it — unless you can stop them first.

FOR MORE INFORMATION

Watch our recent Live Q&A: Malicious Bots in Modern Threats →

To learn more about bot-based API attacks

Request a Demo →

To learn more about how ThreatX can help protect against bot-based attacks

THREATX

www.threatx.com
info@threatx.com

ABOUT THREATX

ThreatX's API protection platform makes the world safer by protecting APIs from all threats, including DDoS attempts, BOT attacks, API abuse, exploitations of known vulnerabilities, and zero-day attacks. Its multi-layered detection capabilities accurately identify malicious actors and dynamically initiate appropriate action. ThreatX effectively and efficiently protects APIs for companies in every industry across the globe. For more information, visit: www.threatx.com.