INTRODUCTION

Many organizations are significantly vulnerable to man in the browser attacks, and to malware working from the machines and browsers of real users. Sophisticated bots play a key role in mounting these attacks, and represent a new category of threats. Although they are ghosts that fly under the radar, these bots are causing serious damage to businesses today.

Malicious bots /automated web attacks have a history of being improved relentlessly during the intellectual arms race between attackers and defenders, but the new breed is distinct from three earlier generations of bots. The latest bots, logically labeled “4th Generation”, are dangerously adept at operating under cover of legitimate users. The most recent evolutionary change is parasitism, by sneaking into a website after infecting a legitimate human site visitor who is unaware of being accompanied by the parasite bot. On a massive scale, Gen4 bots cheat companies without being detected by methods currently in place. If your company has a site with a substantial volume of user traffic or transactions, it is likely that you are currently a target.

This white paper will explain how Gen4 bots operate, their scope of operation, how we know if your website has been targeted, and the best methods of preventing the damage they cause.

Where Gen4 Bots Are Directed to Attack: In the Cracks

There are areas between IT, security, and business operations that receive less scrutiny, where bot attacks can go undetected longer. Typically, the attackers take aim at a company’s web application, and/or processes and business logic.

Web application firewalls (WAFs) were synonymous for years with good security, but they don’t protect against business logic abuse by bots. Signature-based defenses have quickly become relics of the recent past. We see a lot of security dollars spent on finding exploits, but no single approach is foolproof. We need to identify malicious bots as they begin testing and attacking, to stop them upon first access. Web behavior analytics (WBA) is an essential new approach in staying ahead of the rapid advances that criminals are steadily making in their hacking weaponry.
How the Bad Guys Get Badder

Hackers have learned some key lessons that guide their activities:

1. It’s not difficult to trick users, on a large scale, into accepting software that either contains malware or gives access to malicious bots.

2. When launching an attack from the browser of a real user, who innocently becomes an enabler of the attack against an organization, it is easier to overcome signature-based detection systems. One reason: the request is actually originating from a browser with genuine user history and characteristics.

3. Business logic has become an enticing target, if there is something behind it that is of value to steal or extract. Vulnerable logic is often situated between business, technical, and security domains, and this gap presents an opportunity for the attacker, since it is not always clear who should be guarding it.

4. It’s possible to engineer fraud that takes advantage of the victim organization’s business practices and goes completely undetected, while milking the victim for significant sums of money.

Automated Attacks Become Harder to Detect and Intercept

Automated attacks rely on a number of methods, and have different criminal purposes.

- **Account takeover** relies either on brute-force approaches, trying many combinations of usernames and passwords on a popular login page, often relying on stolen login combinations. The bad news: the brute force method is astoundingly successful, in large part because many users choose passwords that are foolishly obvious. For example, PerimeterX studied a brute force attack that had an incredibly high success rate of 8%. This bot attack tried 5 million combinations per day, which suggests they broke into about 400,000 accounts daily. Once the account is taken over, the hacker has instant access to any credit card data of the real account owner and other personal information that is stored in the account.

- **Fake user creation** may not sound nefarious—it can be used to collect a discount code, or to get thirty days of free movie streaming. Hackers use
this to amass millions of fake users, effectively giving them control over a large army of registered (though fake) users on particular websites. One danger is DDoS via hoarding; for example, having thousands of apparently legitimate users reserve all the cars that a particular rental car company has in a given city.

- **Carding**, or theft of gift card balances, are usually carried out by earlier-generation bots. Attackers understand the number structure of gift cards, and may try many millions of combinations to break into a gift card account and then steal the balance. With 93% of Americans giving or receiving a gift card this year, there is plenty of rich, low-hanging fruit here for thieves.

- **Marketing fraud** poses a serious threat to ecommerce and media businesses. Ever since companies began paying traffic sources for clicks and traffic, criminals have had a motive to create fake traffic. Marketing fraud has existed since the late 1990s, but has evolved significantly.

- **Content theft** often takes the form of scraping. If you own a commerce site, your competitors want your pricing, your current inventory, and your SEO-optimized product descriptions. If you own a news outlet or media content site, hackers want to steal your content and post it on third-party sites.

- **Checkout abuse** If you ever tried to buy a high-demand product online, like the latest Air Jordan sneaker, you know it’s impossible. Within minutes, all the inventory is gone. Bots are behind almost all of these near-instant purchases. The perpetrators quickly resell their inventory on the secondary market. This distortion of the efficient, natural marketplace causes a range of problems for both providers and consumers. For the website owner, it damages consumer trust and business profitability over the long term.

### The Four Generations of Bots

As the Web has evolved, we have witnessed evolution of bots and automation engines. We can identify four distinct generations of bots, as the graphics below indicates. Generation 4, the newest, is an actual browser running on a device that belongs to an actual - and presumably innocent - user, exploiting that user’s device, credentials and environment.
**Gen 1 Bots**

A Gen1 bot is a simple script. These early bots of the late 1990s were used for brute-force break-in attacks, and for content and price scraping. They were limited and stateless. Gen1 bots have no sense of context, and they don’t keep a session for the user. These basic bots were not difficult to block; put a cookie in place, and the Gen1 bot will not store or use a cookie value properly.

**Gen 2 Bots**

Second-generation bots appeared in the early 2000s and added support for cookies and state. They can parse HTML, maintain cookies, and can go through the login process. In fact, they can carry over the CSRF token between pages, and overcome simple security mechanisms typically found in the login and checkout processes.
An example of a Gen2 bot is Sentry MBA, a brute-force tool for hackers. It has been used for account takeover and breaking into gift and credit cards. They effectively bypass security that relies on cookies as a test, but you can reliably block Gen2 bots with a simple JavaScript challenge. They will not be able to return the right answer, because they parse the pages, but aren’t capable of actually rendering the pages, and they cannot execute JavaScript.

**Gen 3 Bots: Headless Browsers**

Third-generation bots became much more capable in 2010. They are built on top of testing tools, like PhantomJS and Selenium. These are real browser cores, like Webkit or Firefox, that have an automation layer on top of them. There is no UI, and no user controlling the session for the browser, but there is a scripting and rendering engine that can execute any payload and render a page. In many cases, to help hide their identity these bots are executed through a proxy network.

**GEN 3 BOTS: HEADLESS BROWSERS**

**Gen 3 bots**
- Based on a real browser core, adding a scripting engine
  - Behaves like a browser - run Javascript, can render images, render sound
  - But no human being behind them, usually in a data center or single location

**Detect/block Gen 3 Bots**
- Signature approach

The Gen3 bot often is sent from a data center or single location (but through a network of proxies, as previously mentioned, or a botnet), and has been used for, among other things, Denial of Service attacks, marketing fraud, advertising fraud, and checkout abuse. Detection is much harder than with earlier generation bots because the attacks can mimic real browsing sessions. For example, the bot may appear to be a human, navigating web pages and clicking on links.

The best way to detect the Gen3 bot is a fingerprint-based approach to profile and challenge the browser.

The involves determining whether the browser originates from a data center - which is highly suspect - or from the IP address of an individual user. Reliable determination also requires identification of nuances in the browser. This is done by challenging the browser, such as how it renders images or sound, then comparing the responses with expected values for that browser / version / platform combination. These clues are needed to accurately and quickly determine, for example, that the user is an iPhone emulator, not an iPhone – or that rather than Chrome 53, the user is actually PhantomJS trying to mimic a newer version of Chrome.
Gen 4 Bots – Parasite Bots and Infected Users

Fourth-generation bots are the most sophisticated yet, and represent today’s greatest threat. Gen4 bots are malware that infect a real user, either by injecting a malicious extension into the user’s browser, or by simply executing the browser in a hidden window. From the viewpoint of the website that is being targeted, this is a destructive bot operating alongside and camouflaged by the human user’s session. The bot gains access by accompanying the human user. This explains why, at PerimeterX, we often refer to them as parasite bots.

**GEN 4 BOTS: INFECTED USERS**

**Gen 4 bots**
All tests through browser OK, so now we need to focus on
user behaviour

- Nuances in behaviour to positively ID humans (exonerate)
- More perfect / linear / non-random behavior of bots (incriminate)
- Even when a human is present, it can find a malware parasite in
  the background with unlikely parallel behavior.

**Detect/block Gen 4 Bots** Behavioral-based detection

Not all Gen4 bots are parasites, but it’s very instructive
to look at this insidious form of the latest bots.

**Gen4 bots: You don’t even know I’m here**

The great problem in detecting Gen4 bots: the traffic
appears as real users and is in fact legitimate traffic.
There is real human user interaction coming from the
same browser sessions as the bots, which do their
damage under cover of the legitimate users. Let’s look
at how one can recognize and stop Gen4 bots.

You can’t use IP reputation to incriminate the user, and
you cannot rely on cookies or JavaScript. What does
that leave for investigation to detect them? It comes
down to the user’s interaction with the application. For
example, real humans move the mouse in certain ways,
and react to elements on the page in specific ways. It is
this user action, and many others, which web behavior
analytics can study to detect when a Gen4 bot is
running the interaction, and block it.

**Humans are random, but not that random**

When the human user clicks on a checkbox or a button
that is preceded by a focus-in movement of the mouse.
Hackers realized that WBA defenses were looking for
the focus-in pattern, so they programmed that behavior
into their bot. PerimeterX noticed for example that in
some cases, after circling in, the bots would click on
the same pixel every time, giving away their non-
humanness. WBA can recognize this behavior with
accuracy. Hackers soon realized that was a red flag, so
they adopted a random pattern to choose the pixel to
click on. That drove us to evaluate the randomness of
the click location; too random, and we can be confident
that it’s a bot.
The bot will randomize patterns convincingly, and hackers will continue to improve the resemblance to human behavior, but the right WBA tools can identify which are malicious bots. When you understand how humans use a mouse and browser, bot behavior doesn’t pass for human, because the mouse action of a bot displays either too much linearity or too much randomness, for example. As Gen4 bots steadily become more sophisticated, it becomes increasingly difficult for traditional approaches to even detect the bots. Behavior-based approaches however, turn the tables by being ahead in the arms race. Hackers can’t guess which collection of behavioral nuances are being used to ferret out their bots. And even when they hide under cover of actual human users, the behavioral approach can see the malicious bot activity in the session. Only behavior-based approaches are capable of detecting and blocking the newer malicious bots.

The Frontier in Deciding “Bot or Not”

Now that we understand what we are fighting, how do you determine if a specific user is bot or human? Very few solutions can handle the most modern advanced attacks. You need to know how your real users interact with your web pages. That will vary from page to page.

Incriminate when stakes are low, exonerate when stakes are high

There are two principal decision-making models:

Incrimination, which presumes innocence until there is evidence of malicious activity, and Exoneration, which assumes the website must be protected against this user until there’s evidence there is no malicious activity.

Incrimination

Exoneration

Website operators need to consider these two approaches, and apply them appropriately. Do we incriminate the user or exonerate the user? Presumed innocent or presumed guilty? It’s up to the evidence to change your mind.

EXONERATE OR INCrimINATE

Incriminate We assume visitor is a human, it’s up to the evidence to prove us wrong

<table>
<thead>
<tr>
<th>ALLOW</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>![User Icon]</td>
<td>![Block Icon]</td>
</tr>
</tbody>
</table>

Exonerate We want proof that visitor is a human, not just lack of proof that a bot

<table>
<thead>
<tr>
<th>ALLOW</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>![User Icon]</td>
<td>![Block Icon]</td>
</tr>
</tbody>
</table>

? NOT YET KNOWN HUMAN BOT
On the homepage and landing pages, most will use the Incriminate model, which says the user is innocent until proven to be a bot. You don’t want to impact the user experience, particularly when they are just arriving at your site. You don’t know enough yet to interfere, and on these initial pages, a bot probably cannot do much damage. By default, you allow the user.

The Incriminate approach makes it easy to block out Gen1 and Gen2 bots at this early stage; if the user doesn’t run JavaScript or cookies, then block them.

However, on sensitive pages with business logic or anything you want to protect from bots, PerimeterX would recommend that most websites shift to the Exonerate model; presumed guilty until proven innocent. Only after we collect sufficient evidence to feel sure it’s a human at the steering wheel, do we let them through.

To summarize, on most of your website pages, where there is little risk of damage or unauthorized data access during the start of a new session, we recommend “incrimination mode” which is the presumption of innocence. But at login and at checkout, where a malicious bot can quickly and directly commit fraud, you go with exoneration mode: assume the worst, until you have sensor information that proves a user is human.

The Newest Attacks by the Latest Bots - Examples

It is likely that your website is under attack at this time, and that all your websites are under attack at every hour, every day.

Classic account takeover attack, with twists

A recent automated attack we detected and categorized as an IoT attack came from many different hacked devices, including Canon printers, as shown below. With over 5 million attempts per day, it achieved an 8% success rate, or a stunning 400,000 break-ins per day. This makes it highly likely that the perpetrators had access to a very good user list, probably one stolen from another website. It had to be very current. The login attempts originated from thousands of IP addresses.

In this very aggressive yet quiet attack, Perimeter X detected 3,500 IPs hitting the site once per minute. From the website perspective, each attempt appeared as a different user on a different home network, attempting a normal login every minute or so.
Scraping attack

In another example, the hackers scraped content and application data to steal content across an entire website every three hours. Selenium would render the JavaScript. Without checking the behavior of every data request, it is very difficult to block this scraping attack.

Checkout and business logic abuse

This attack targets business logic and involves checkout abuse. Bots buy up the entire inventory of scarce hot products in limited supply, and then seek to hawk them at tremendous profit margins on eBay, Craigslist, StubHub or elsewhere. Bots often are the buyers of 100% of these immediate sellouts. Product makers are now paying attention, and have been pushing for new laws to criminalize this activity, but rendering the activity illegal is hardly a solution.

Gen4 attacks via large numbers of infected users

The Gen4 attack begins with malware infection of many users, often with a malicious browser extension. From there, attacks are limited only by the fraudsters’ creativity. The malware may execute an intricate series of steps, piggybacking along with a legitimate but oblivious human host who logs into bank, ecommerce, or social media sites, taking advantage of social logins to open new accounts, and exploiting the user’s personal and financial information it can access.

Traditional defense systems perceive only the real, human website visitors, rather than the bot. The attack may involve tens of thousands of genuine but bot-controlled user accounts taking part. Newer, sophisticated bots are designed to not trigger alarms before major damage is committed. Analyst group 451 Research (in a Report by Eric Ogren published January 2017) concludes that web behavior analytics (WBA) must be the cornerstone of an effective anti-bot defense today. WBA works by collecting as much information as practical about how real human visitors and customers interact with a particular website’s pages. WBA defenses use that knowledge to sniff out robotic behavior in real time and challenge the bot, or otherwise block the attack.
Effective Defense against Infected Users

The only way to incriminate a user infected with Gen 4 bots is by applying a behavior-based approach. This requires a sophisticated sensor framework and big data and machine learning. The result is a highly capable approach to identifying these malicious bots. Once you have the human-behavior framework in place, identification of the attacks is immediate.

You Can and Should Act Now Against Automated Web Attacks

Some best practices in the defense against Gen4 bot attacks have emerged already.

1. **Understand your web traffic.** Profile your users, and get to know their normal behavior on each page of your sites: Site owners must understand authentic user behavior and how it appears.

2. **Collaborate among security, IT and business owners:** Self-defense is a business survival issue; not just an IT issue. It requires collaboration with your application owners. It needs to be cross-departmental.

3. **Pay only for legitimate traffic:** Bot activity can escalate fraud costs, data loss, theft, and customer loyalty. If you strip out 40% of undesirable bot traffic you can boost your site performance, stop counting impostors, stop paying bonuses to hijackers. Use correct data in your analytics, and pay only the marketing affiliates that are legitimate.

**How to Start**

To get started with behavior-based web protection, it’s advisable to use it on a test portion of your website. First, monitor your traffic and understand what is normal and what is abnormal behavior. Next, get the reporting and forensics running before you block users. With this approach, you can quickly get familiarity with no risk. Some CISOs see results within hours. The speed of improvement can be astonishing once you start to block based on WBA, but expect to wait a few weeks for your analytics to show improvements.
Conclusion

Automated web threats constantly evolve. Their sophistication increases steadily. When we at PerimeterX reverse engineer malware, we see a high level of attackers’ capability. It’s evident that companies face criminal organizations with large, well-trained, organized software teams that perform better than most enterprise software development organizations. These organized crime groups update their malware multiple times per day, and their code is very well-written and tested. To protect your site from advanced automated web attacks, in particular Gen 3 and 4, your organization’s defense needs to evolve and get ahead.

A Web behavior based approach to protection is the next level in the evolution of IT security. Once you implement a robust behavior-based layer of defense, you will have established a troublesome barrier against bot attacks that helps prevent damage to your business.

About PerimeterX

PerimeterX prevents automated attacks by detecting and protecting against malicious web behavior. By analyzing the behavior of humans, applications and networks, PerimeterX catches automated attacks in real-time with unparalleled accuracy. Its proprietary technology protects your business and web infrastructure by preventing known automated attacks, as well as those that do not trigger security alarms. Businesses deploy PerimeterX and gain visibility within minutes, and easily integrate it into their existing infrastructure. PerimeterX empowers companies across numerous industries including enterprise SaaS, e-commerce and media to protect against advanced automated attacks. To learn more, please visit www.perimeterX.com