

# Optimizing Web Application Security for the New Bad Bot Threat Landscape

A collaborative effort with many thanks to:

**Yanky Askal**

Ecommerce Operations Manager

---



**John Stauffacher**

Security Architect

---



**Engin Akyol**

CTO

---



# Table of Contents

<b>Bad Bots – A Brief Overview.....</b>	<b>3</b>
Bad Bots are the Key Culprits Behind the Majority of Website Problems.....	3
Bots – The Good, Bad and the Ugly.....	4
Advanced Persistent Bots.....	5
Why Homegrown Solutions are Ineffective Against Bots.....	6
WAFs Designed to Solve a Different Problem – Ineffective Against Bad Bots.....	7
<b>Web Application Security Requires Complementary Solutions.....</b>	<b>8</b>
<b>The Business Impact of Bad Bots.....</b>	<b>9</b>
Competitive data mining.....	9
Watching the watchers.....	10
Overstressed infrastructure.....	10
Negative SEO.....	11
Skewed Analytics.....	11
<b>The Even Greater Business Impact of Ugly Bots</b>	<b>12</b>
Brute force account takeovers.....	12
How hard is it to obtain stolen credentials?.....	12
Carding.....	13
The stolen card market.....	13
<b>Bots Impact Multiple Departments.....</b>	<b>14</b>
<b>Case Study: IoT BotNet The Internet of Things.....</b>	<b>15</b>
<b>Digital Advertising Fraud.....</b>	<b>16</b>
Impression fraud.....	16
Click fraud.....	16
Retargeting.....	16
<b>Bot Mitigation Strategies.....</b>	<b>17</b>
Whitelist the Good Guys.....	17
Geo Fencing.....	17
Flow Enforcement.....	17
Login Enforcement.....	17
Client Enforcement.....	18
Do Independent Recon.....	18
Do Independent Testing.....	19
<b>Conclusion.....</b>	<b>20</b>
<b>Author Biographies.....</b>	<b>21</b>
<b>About Distil Networks.....</b>	<b>22</b>

# Bad Bots – A Brief Overview

## Bad Bots are the Key Culprits Behind the Majority of Website Problems

Bad bots comprise an average 19% of all Internet traffic, based on Distil Networks’ annual bot report. It’s this 19% that causes the majority of problems. These are quite diverse in nature and vary from website to website—depending on what industry you’re in and what type of data is of value to hackers or competitors.

API abuse can be caused by bots, business partners, runaway scripts, and integration bugs. In relation to online fraud, both bots and humans can generate transaction fraud and chargebacks.

Figure 1 – Bad Bots Cause Any Number of Website Problems



## Bots – The Good, Bad and the Ugly

Yanky Askal is head of eCommerce operations at Manhattan's [B&H Photo Video](#). He is responsible for website security, application and infrastructure administration, and site performance at the world's largest professional video, photo, and audio equipment supplier. Borrowing from the Clint Eastwood movie title, Askal frames the bot problem as *The Good, the Bad and the Ugly*.

- Good bots ensure online businesses and their products can be found by prospective customers
- Bad bots scrape publicly-available data from sites. They then manipulate and reuse that data (e.g., pricing, inventory levels) to gain a competitive edge
- The truly ugly bots undertake criminal activities, such as fraud and outright theft

**Table 1 – Good, Bad and Ugly Bots**

Bot Types	Definition	Problem	Management
Good Bots	Invited guests: Search engines, affiliates, vulnerability scanning & monitoring tools	Rapid site surfing without pauses. Indexing and cross-referencing large inventories can stress servers	Dedicate specific servers to handle traffic, request slowdowns from bot owners
Bad Bots	Uninvited guests: Web scrapers copying public content; white-hatters looking for vulnerabilities	Similar impact to infrastructure but also detrimental to the business by facilitating unethical competition	Much harder to control, as source is usually anonymous; actively avoids blocking
Ugly Bots	Burglars: Black-hatters, criminals, and other abusers of trust relationships	Brute force attacks, password and credit card testing, fraudulent accounts, flooding servers with form spam	Endless manual process that drains human and IT resources. Removal of targeted forms

## Advanced Persistent Bots

Distil is seeing more and more sophistication in the bot threat landscape, resulting in what is known as advanced persistent bots (APB). They comprise as much as 53% of all bot traffic. APBs have several advanced capabilities such as:

- Mimicking human behavior
- Browser automation
- Loading JavaScript and external resources
- Spoofing IP addresses and user agents
- Cookie support
- Other activities

Flying under the radar of many existing security solutions, these bots are much harder to identify and block than simple bots.

Their persistency aspect comes from bad bots' process for evading detection. They use such tactics as:

- Dynamic IP rotation (drawing from huge pools of IP addresses)
- Using Tor networks and peer-to-peer proxies to obfuscate their origins
- Distributing attacks over hundreds of thousands of IP addresses. For example, a bot might use 1000 IP addresses to make one request each, instead of one IP address to make 1000 requests.

Figure 2 – Advanced Persistent Bots (APBs)

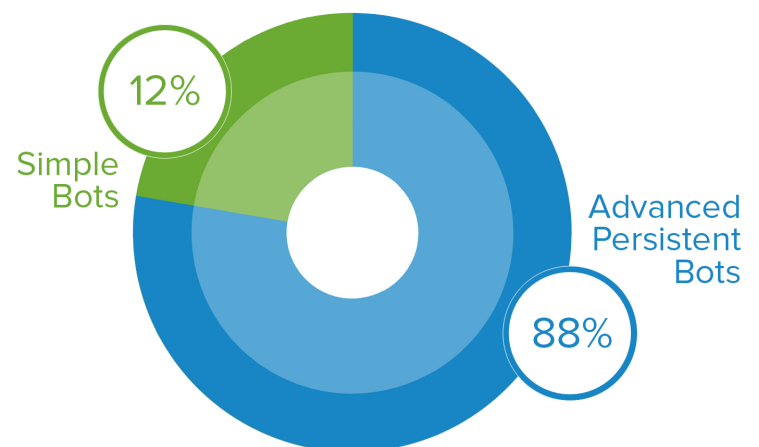
### Advanced Persistent Bots (APBs)...

#### Advanced

Mimic human behavior  
Load JavaScript  
Load external resources  
Support cookies  
Browser automation (Selenium, PhantomJS)

#### Persistent

Dynamic IP rotation  
Distribute attacks across IP addresses  
Hide behind anonymous and peer-to-peer proxies



## Why Homegrown Solutions are Ineffective Against Bots

Many of Distil's customers have tried to tackle the bot problem on their own by using CAPTCHAs or through log analysis—after which they write custom scripts or rules to block IPs or rate-limit their traffic.

CAPTCHAs lower conversion rates and can be easily defeated by advanced bots.

Log analysis is also not very effective. Not only is it expensive to perform, but most sophisticated bots masquerade as humans in logs—especially those using random IPs or distributing their attacks over hundreds of thousands of IPs. Once you find and block an offender, it can simply change IPs and return unhindered.

IP blocking and rate limiting are also reactive tactics. They can be thwarted by dynamic IP rotation and distributed attacks.

**Figure 3 – Home-grown Bot Solutions**



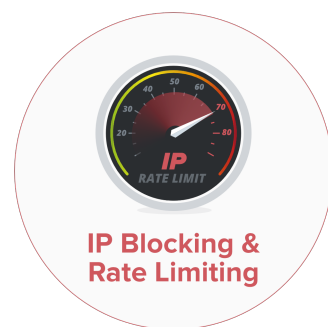
**CAPTCHAs**

- Creates a poor user experience
- Defeated by advanced bots
- Defeated by CAPTCHA farms
- Reduces conversions by up to 27%



**Log Analysis**

- Bots appear human in logs
- Labor intensive
- Distributed attacks hard to pinpoint
- Reactive in nature



**IP Blocking & Rate Limiting**

- Defeated by distributed IP attacks
- Defeated by low and slow crawlers
- Defeated by peer-to-peer / proxies
- Reactive in nature

## WAFs Designed to Solve a Different Problem – Ineffective Against Bad Bots

When discussing bot problems with prospective clients, Distil often hears, “I’ve got a web application firewall (WAF) to handle that.”

However, WAFs were never designed to to manage the volume, variety, and sophistication of today’s bots. Instead, they identify and block application exploits looking to attack a coding vulnerability. They are IP-centric and use attack signatures.

But in the world of bots, there are no signatures. Bots aren’t limited to perpetrating website attacks, rather they programmatically abuse and misuse websites—resulting in a wide assortment of problems.

Bots are dynamic; they can attack anything (right column, Fig. 4). If a hacker can dream up a way to misuse a site or its data, then they can create a bot to do it. For this reason WAFs and bot detection solutions solve different problems.

**Figure 4 – WAFs vs. Bot Detection & Mitigation**

	Web Application Firewalls	Bot Detection & Mitigation
Focus	Application Coding Exploits	Automated Abuse, Misuse & Attacks
Vulnerabilities / threats	OWASP Top 10 App Security Flaws Misconfiguration Issues Data Leakage Correlates to CVE Numbers	Brute force Login Attacks Reconnaissance Attacks Man-in-the-browser / man-in-the-middle Application Denial of Service Carding Payment Fraud Account Hijacking New Account Fraud Content Theft Competitive Data Mining Spammy Content API Abuse & Misuse

# Web Application Security Requires Complementary Solutions

IT have been fighting hackers, volumetric attacks, bad guys flooding infrastructure and paralyzing businesses since the advent of the Internet. Today a number of tools exist to manage those issues. Many Distil customers have a DDoS provider, firewall, and a WAF, so their security looks something like the chart in Fig. 5.

Bot mitigation should be deployed in conjunction with this security stack.

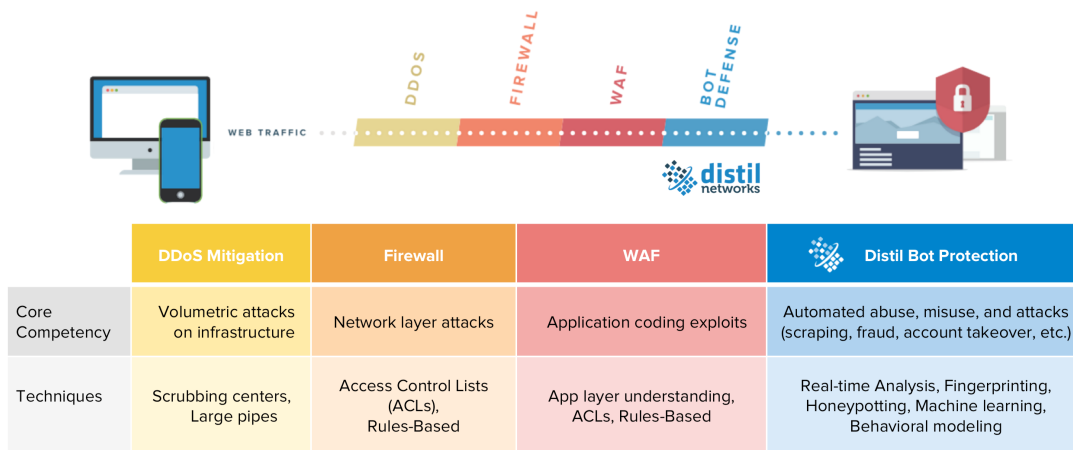
In looking at these various solutions, each uses different techniques to solve a variety of problems. DDoS protection uses scrubbing centers and large pipes to handle volumetric attacks.

Firewalls use ACLs and rules to stop network layer assaults. WAFs inspect web (HTTP/s) traffic and use ACLs and rules to enforce protection.

That leaves bot protection. It leverages real-time analysis, device fingerprinting, honeypot injection, machine learning, and behavioral modeling to identify and block malicious automation (i.e., bad bots).

Table 2 below summarizes why traditional tools don't, and can't, work against bots.

**Figure 5 – A Typical Application Security Stack for a Distil Networks Customer**



**Table 2 – Traditional Tools vs. Bot Detection and Mitigation**

	Hackers	Volumetric Attacks	Bots
Attack Vector	OWASP Top 10	DDoS	Multiple, Fly under the radar, appear to be human or a good bot
Target of Attack	Application code, to steal information or hijack a website	Network infrastructure, to paralyze the business	Multiple systematic abuses of web infrastructure and applications
Tool	WAF, secure coding practices	DDoS mitigation, scrubbing centers	Dedicated tool that can block bad bots without impacting legitimate users



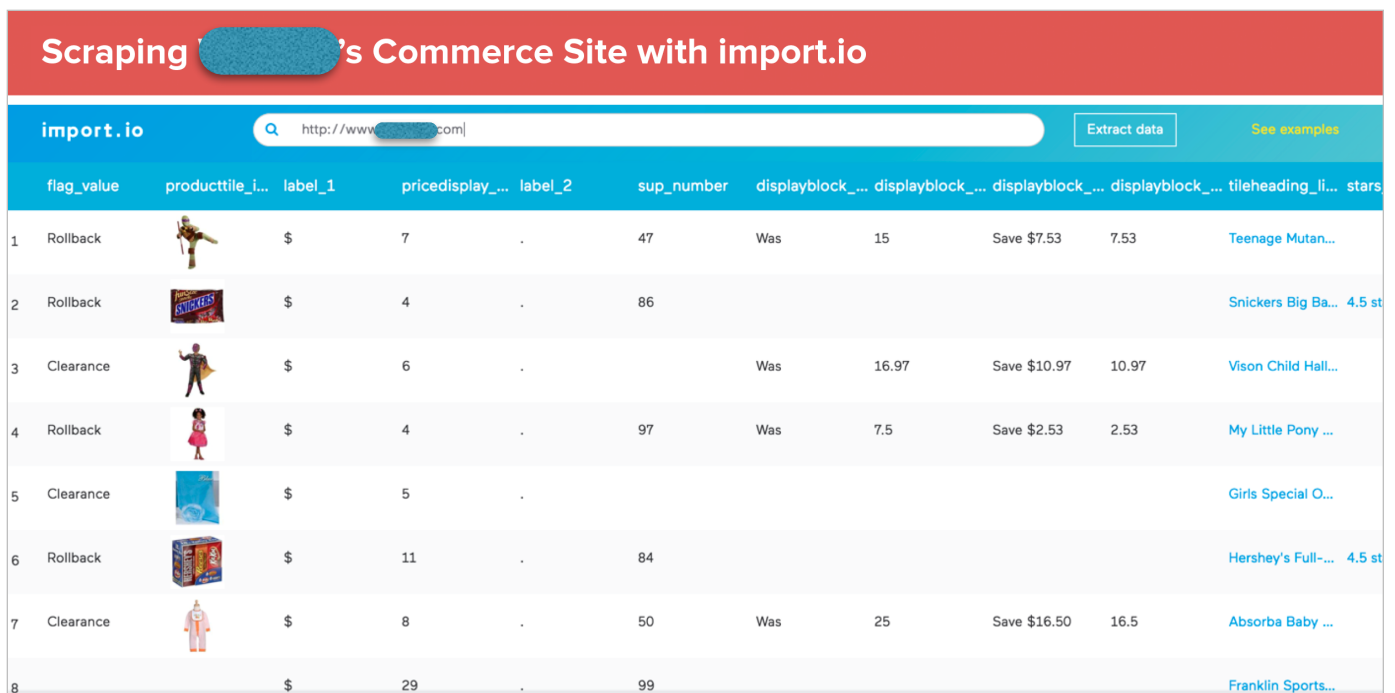
# The Business Impact of Bad Bots

Scraping content from a competitor’s website might be considered fair game, since the data is in the public domain, but this act is much more than just stealing content. Bots can duplicate entire product portfolios, continuously monitor dynamic pricing, and report out-of-stock status at competing sites.

## Competitive data mining

A company like B&H Photo Video has spent years, and millions of dollars, to build a world-leading ecommerce site with over 300,000 SKUs. What gives anyone the right to use that investment to create their own competing business for next to nothing? Once scrapers have access, they can track price and availability shifts in real-time, while the original site is penalized by suppliers for ostensibly leaking confidential pricing.

Figure 6 – Scraping Made Easy



The screenshot shows the import.io interface with a search bar containing a URL and an 'Extract data' button. Below the search bar is a table of scraped product data. The table has columns for various attributes including flag\_value, producttile\_i..., label\_1, pricedisplay\_..., label\_2, sup\_number, displayblock\_..., and tileheading\_li... The data rows show products like Teenage Mutant, Snickers, Vison Child Hall..., My Little Pony, Girls Special O..., Hershey's Full..., Absorba Baby, and Franklin Sports.

flag_value	producttile_i...	label_1	pricedisplay_...	label_2	sup_number	displayblock_...	displayblock_...	displayblock_...	displayblock_...	tileheading_li...	stars
1	Rollback		\$	7	.	47	Was	15	Save \$7.53	7.53	Teenage Mutan...
2	Rollback		\$	4	.	86					Snickers Big Ba... 4.5 st
3	Clearance		\$	6	.		Was	16.97	Save \$10.97	10.97	Vison Child Hall...
4	Rollback		\$	4	.	97	Was	7.5	Save \$2.53	2.53	My Little Pony ...
5	Clearance		\$	5	.						Girls Special O...
6	Rollback		\$	11	.	84					Hershey's Full-... 4.5 st
7	Clearance		\$	8	.	50	Was	25	Save \$16.50	16.5	Absorba Baby ...
8			\$	29	.	99					Franklin Sports...

## So simple, anyone can do it

Import.io is a tool that makes scraping websites as simple as drag-and-drop. Figure 6 shows the results of running it against a major online retailer’s website.

In 2.5 seconds, Import.io broke out everything a competitor might need and presented it as an XLS file. This particular ecommerce site allows this because it believes its prices cannot be beaten. But the same thing can happen to any website. The barrier of entry has been lowered to the point where anybody can get into the scraping game. It capitalizes on other businesses’ hard-earned effort and work product without any real investment.

## Watching the watchers

Given advances in bot-specific threat detection technology, some companies have been able to turn the tables on the scrapers. A large tire discounter expects price-matching to occur. When it's about to raise prices, it turns off its protection to allow bots to price match, knowing their competitors will follow suit.

More aggressive sites even feed scraper bots false information. One airline, for example, feeds an unsavory competitor bogus pricing information—whatever listings the latter is showing its own customers is incorrect.

## Overstressed infrastructure

Web infrastructure has a capacity limit—there's only so much a business can get out of its servers, bandwidth, firewall, and routers. All the non-human traffic coming in—the bots—is using up resources and can cause slowdowns and downtime. It's like turning a retail store into a public park—all the same resources are there, but now it's serving a bunch of people who aren't customers and may even want to cause damage. Moreover, the non-customers can come from anywhere on the planet and give no warning as to when they might appear. Predicting hardware requirements becomes a huge gambling exercise.

**Figure 7 – You WILL be Assimilated**

### Bezos used bots to crush Diapers.com

#### 2009

Amazon's pricing bots were tracking Diapers.com.

#### 2010

Amazon's pricing drops ate into Diapers.com's growth. Investors grew wary of pouring more money into the startup, given the competition.

#### Late 2010

Amazon rolls out Amazon Mom, offering huge discounts and free shipping. Soon after, Amazon announces the acquisition of Diapers.com



"If you don't sell, I'm going to squash you like **this**."

**Source:** [http://www.slate.com/blogs/future\\_tense/2013/10/10/amazon\\_book\\_how\\_jeff\\_bezos\\_went\\_thermonuclear\\_on\\_diapers\\_com.html](http://www.slate.com/blogs/future_tense/2013/10/10/amazon_book_how_jeff_bezos_went_thermonuclear_on_diapers_com.html)

## Diapers.com and Amazon

Diapers.com was once a successful ecommerce site providing a comprehensive line of baby products. Meanwhile, Amazon.com was a successful online bookstore considering entering other target markets—including products for infants. They ran a sniffer against Diapers.com, learning its business strategy and tracking its pricing and margins. Eventually Amazon was able to mimic Diapers.com's product line while also getting real-time price updates.

Within a year, Diapers.com's investors became wary (Fig. 6). Amazon had launched Amazon Mom and shortly thereafter acquired Diapers.com—all of this being much quicker, cheaper, and easier than building their own baby products business from scratch.

## Negative SEO

Google's search engine isn't happy when it finds duplicate content on websites. When it does, it assumes that the site hosting the original content is trying to game the system. It then lowers that site's ranking, relegating it from page one results to pages two or three. The entity that copied the information can sometimes even end up with a higher ranking than the originating site. This scenario is yet another way to lose business without actually doing anything wrong.

## Skewed Analytics

Google Analytics and other tracking systems let you understand what your customer is looking for; you can then optimize your site based on actual customer experience. However, when bots enter the picture and accept JavaScripts, they're welcomed as human users. Since 60% or more of traffic now comprises bots, it's easy to see how inaccurate analytics can become. This skewing also shows up in conversion rates, making life difficult for your marketing team.

# The Even Greater Business Impact of Ugly Bots

Today's bots—the ugly ones—go much further than simple web scraping and wreaking havoc on your IT infrastructure. They can turn legitimate sites into unwitting participants in criminal activities.

## Brute force account takeovers

Say a bad guy has acquired a database of stolen usernames and passwords. They construct a bot that runs all those user credentials against a targeted ecommerce site. Most will fail, but given users' proclivity for using the same credentials on multiple sites, a few bots will receive a welcome message, validating it as a valid account on that site.

Now that username/password combination has a resale value on the black market. Anyone buying it can access as much personal data as the valid account owner has posted—name, mailing address, phone number, email address, and more. In turn, that information may be used to access bank and credit card accounts, date of birth, Social Security number—the list goes on.

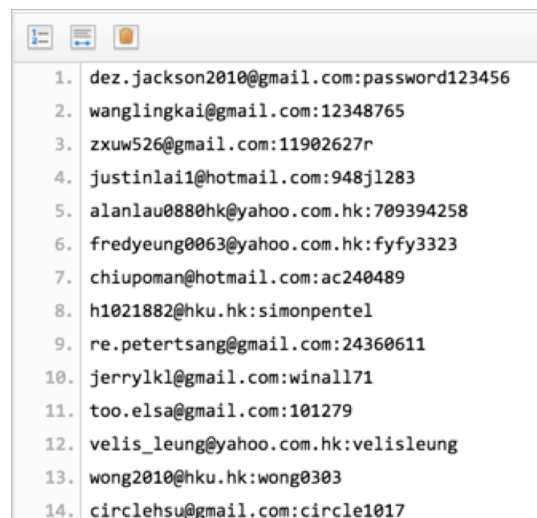
Account details such as purchase delivery addresses can also be altered, further impacting existing customers. The operator of the targeted site has not been breached in the traditional sense, but has unwittingly become part of a chain of criminal activity.

It costs almost nothing to test credentials for validity. All it takes are a few Amazon or T-1 instances to cycle through a lot of logins, learn which ones are useful, and then sell them on the black market for pennies on the dollar.

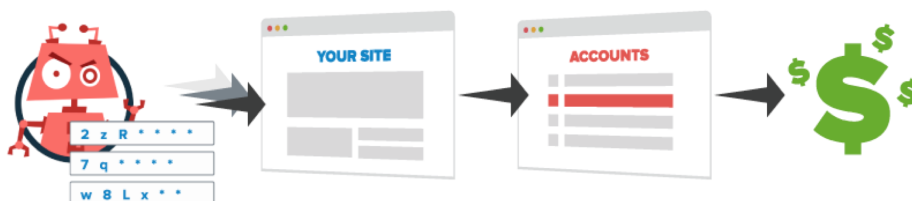
## How hard is it to obtain stolen credentials?

Procuring stolen credentials is easy as running Pastebin. There is no need to risk leaving traces by shopping on the black market. Simply visit any of the random sites where people post everything they ever wanted to share with anyone else, quickly and anonymously pull down the data dump, and then run a bot through usernames and passwords.

**Figure 9 – Stolen credentials to feed a bot are posted on sites like Pastebin**



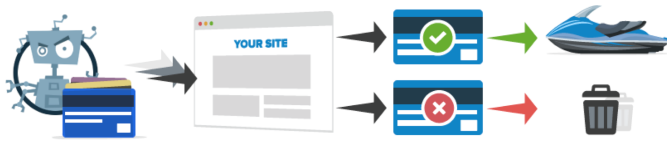
**Figure 8 – Bots Are Used for Brute Force Login Attacks**



## Carding

Carding is when a credit card is used to charge a very small amount. It's not enough to alert a verification investigation, but serves to validate that the card is still active and hasn't been reported as stolen. The card information is now ready for resale, or for direct use by a criminal to purchase a high-value item, thereby defrauding the respective owners of both the website and the credit card.

**Figure 10 – Bots and fraudulent credit card purchases**



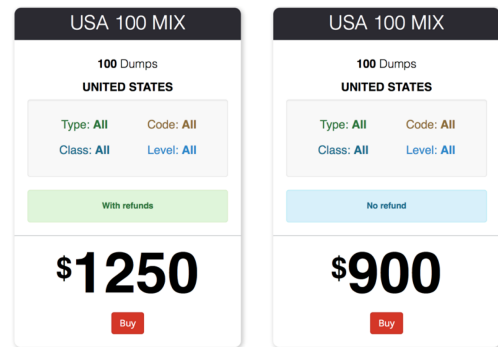
## The stolen card market

The dark side of the web is rife with black markets operating as economic ecosystems. Credit card account details can be purchased “as is” or at a premium for guaranteed-active status.

The latter is important for maintaining a reputation and in being able to command a premium price—so much so that some stolen card dealers offer refunds.

More sophisticated operators convert stolen credit cards to gift cards, these being less trackable. These they provide to “mules,” who use the gift cards to acquire merchandise. These items are then shipped to the original operator for resale at temptingly low prices. From a single investment in stolen credit cards, and by using traditional marketing and SEO techniques, operators build complete illicit businesses on networks such as Tor.

**Figure 11 – Stolen Card Dealer Offering Two Options, One with a Refund**



**Figure 12 – Sophisticated operator advertising his business**

**Why So Cheap?**

As already stated in our FAQ, our products are sold at competitive prices mainly because they are obtained using leaked credit card & PayPal information. Data is acquired, giftcards are bought using the data and then used to purchase goods on various Clearnet stores in order to further anonymize the purchase. Here's how it works:

Simply put, it goes like this:

- 1) You order an item (or items);
- 2) We process the order and obtain a gift card of sufficient value using leaked Credit Card and PayPal information;
- 3) We order the item(s) using the giftcard(s) and receive them to one of our dropships;
- 4) Immediately or within 2 days of receiving the item(s), we ship the parcel to You.
- 5) You receive the parcel 1 to 3 weeks after ordering (depending on the availability of goods in a selected area).

# Bots Impact Multiple Departments

The escalating amount of damage caused by bots is now at a level where it must be factored in as part of the budgeting process across multiple departments. In addition to the aforementioned IT security, infrastructure and SEO impacts, businesses must also consider:

- **Customer Service** agents who are weighed down with calls from carding victims that aren't spending that time on valid customers. The latter may lose patience, take their business elsewhere, and/or badmouth your company on social media
- **Verification** teams have extra work to do, tracking down what is happening with carding victims accounts
- **Fulfillment** teams have to deal with fraudulent delivery alerts, wasting not only their time but also delivery service resources

All the while these people are being paid, but with an increasingly large chunk of their time being wasted on dealing with bot-driven activity. This has a direct impact on budgets and profits.

And then there is the damage that can never be accounted for, such as random traffic spikes when bots launch a targeted assault on an ecommerce site. Most often this occurs around the year-end holidays, that time of year when most ecommerce entities expect to pull in the bulk of their revenues. It's a hyper competitive time, with every outlet competing for every order, for every sale. Promotions are everywhere. Prices drop. Every company uses this time to ensure they meet their year-end goals.

At the same time, the competition is crawling your prices in real-time. For a company like B&H with 300,000 SKUs, competitors crawling its prices over such a short duration can increase bot traffic by a factor of 100. It's almost impossible to be equipped to deal with that kind of spike at (literally) a moment's notice. And what happens to that holiday business—and all future business—if the site crashes under the bot crawler load? No business wishes to deal with such consequences.





# Digital Advertising Fraud

Ad spend continued its healthy trajectory towards an estimated \$56 billion in 2015 and fraud is tracking closely behind. Of that \$56 billion in circulation in the digital advertising world, we estimate that for every \$3 spent on digital advertising, \$1 is being siphoned out of the advertising ecosystem into the pockets of the bad guys.

## Impression fraud

Impression fraudsters put up fake websites and load a huge number of ads on to them - display ads, rich media, video ads, anything will do (including mobile display ads, which are also bought on a CPM basis). Bots are then used to repeatedly load the pages to generate fake ad impressions, disguising the origins of those ads to avoid detection.

## Click fraud

Click fraud is more sophisticated in that it takes two steps. Fake sites are loaded up with search ads so they're part of search ad networks. Then the fraudsters have bots type in the keywords (usually very expensive ones) to cause the ads to load. When those ads are served up by the network, bots click on the ads to generate the CPC revenue for that site.

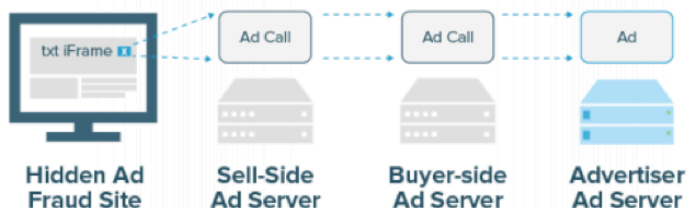
## Retargeting

The bad guys own and control millions of these sites, using bots to earn revenue from CPM and CPC. A new and more sophisticated approach known as "retargeting fraud" sends bots to legitimate publishers' websites for the purpose of collecting a cookie. If the bot is targeting, for example, an outdoor site like REI.com, it will visit a number of sites related to outdoor activities. This builds up a cookie profile about that "user" that appears attractive to advertisers targeting outdoor enthusiasts. When the "cookie collector bot" visits a site that's controlled by the bad guys, the advertisers, through retargeting, try to show an ad to that "user", and pay a premium CPM to do so. Diverting ad dollars Click and impression fraud also mess with analytics. When bots go to premium websites to generate page views without clicks, the CTR for the entire site is lowered. The result is that optimizers divert their ad dollars to sites with higher CTRs, operated by the bad guys. The same thing happens with viewability: when the bad guys stack dozens of ads behind each other so that they are all viewable, those sites attract optimizer dollars. Then there's pixel stuffing, which enables many ads to be hidden on a single page. Another growth industry for the bad guys, thanks to programmatic ad buying.

### Ad Stacking (dozens in one call)



### Pixel Stuffing (hiding more ads)





# Bot Mitigation Strategies

Despite the lack of efficacy among traditional security approaches, a number of strategies exist for IT and web security teams to mitigate the effect of bots on the business.

## Whitelist the Good Guys

Every website requires good bots, so start by whitelisting Google and other search engine indexers—bots that abide by `robot.txt` rules and only go where they're permitted to go. Without them, websites get delisted and online businesses disappear. Monitoring tools also fall into this category.

Next, the field is clear to classify other automated agents as malicious unless and until proven otherwise.

## Geo Fencing

Easy to implement, geo fencing is an often-missed strategy. It exposes a website only to geographic locations in which it does business. If a business doesn't export its goods or services, there isn't a need to have its website available to anyone outside of the United States.

Legitimate customers likely don't use the Tor network for shopping, so its exit nodes should be blocked as part of any geo fencing strategy. It's simple to configure SIEM or IPS to block them, since they're relatively static. Refer to this online list: <https://check.torproject.org/exit-addresses>.

That said, anyone who is familiar with Tor knows that its exit nodes are widely known, and so there is a darker side—Tor bridge nodes. They're undocumented dynamic exit nodes that can only be obtained four at a time from a special email address, or from <https://bridges.torproject.org/bridges>.

## Flow Enforcement

Another way to address bot control is to enforce the flow, or route, legitimate customers take through a site or web application that validates them as they progress. Many bots are hardwired to go after certain high-value targets and encounter problems if forced to follow a typical user's predetermined flow.

## Login Enforcement

Many applications display a login page for their first or second entry points, but then fail to do this for subsequent entry points. This is the equivalent of leaving a house key under the doormat. It's essential to protect all assets within applications by requiring authentication at every sensitive entry point.

### Set account lockouts and brute force limiters

A botnet, created on AWS for less than eight-one-hundredths of a cent, can spend all day hitting your site as it blasts through a massive list of usernames and passwords gleaned from Pastebin. Therefore, while account lockouts and login attempt limits are inconvenient for users who have forgotten their password and have to wait for system access, most ecommerce sites enact these steps to meet PCI compliance requirements.

Development teams should be able to create applications in a way that such brute force, automated attacks can be identified and blocked before they can interface with critical site areas. This is one approach used in Distil's solution; it sets a reverse proxy and uses it to enforce limits.

By encouraging legitimate users during registration to sign up for proactive notification via SMS, any adverse impact can be mitigated if their account ever needs to be locked for security reasons. They can easily authenticate themselves via a return text message, after which

they receive a one-time password reset link. Appropriate blocking can then be initiated against those that cannot authenticate themselves.

Another approach might be to insert a random delay into every login process. Genuine users are unlikely to notice, but even a short delay impacts an automated process that is throwing thousands of credentials at a site.

Until recently, it might also have been possible to use this approach to help track IP addresses behind this type of assault. Now, however, instead of 60,000 hits coming from one or two IP addresses, one or two hits come from 60,000 IP addresses. Validating that many accounts, even with a one or two-second delay, requires advanced machine learning and botnet detection, something out of the reach of in-house defenses.

### **Monitoring session usage**

It's unlikely a valid user would be simultaneously logging into a site from two widely separated locations, nor would such users need two concurrent sessions. Unless an application is specifically written to accommodate such behavior, the simultaneous sessions conflict with one another. Therefore, capping sessions at one per user is a reasonable step to take.

### **Two-factor authentication**

Two-factor authentication (2FA) isn't always feasible, and it can be more work than it's worth. But in many cases, 2FA solves many botnet issues. This is because it requires an additional credential that a bonafide user knows, possesses, or is inseparable from them—none of which a botnet has access.

### **Client Enforcement**

Several characteristics can red flag a suspect user agent, starting with its non-appearance on the list of [all known user agents](#) in the world. Of course, any of these user agent strings can be

modified, but this isn't something browsers tend to do. Since legitimate clients will likely have a user agent stream that appears on the list, this makes it a good resource for whitelisting acceptable traffic sources.

Another suspicious tell-tale sign is a client's inability to execute JavaScript, which most sites use. Such a client isn't able to interact with a site in any meaningful way, such as placing an order. If the inability to execute JavaScript is a blocking criterion, therefore, it's unlikely there will be any pushback from legitimate customers.

This same evaluation applies to cookies; if a site visitor can't store them, it's unlikely to be able to complete a purchase transaction.

And then there are those clients that simply don't look right. This may require a little more analysis to determine whether each has an attribute for the window element; if it doesn't, then it's probably a script. This is one characteristic that sets Phantom JS apart, for example—they don't set a window attribute for the title, so a simple two-second call to a library function shows that they're not actually running with a head, but rather on a terminal somewhere.

### **Do Independent Recon**

Not many companies have their own reconnaissance policy. Instead, they're going through Pastebin, GitHub, Gist, or .onion sites to look for company names, email addresses, information they've seeded out there. It's very likely they'll find some of that information, which means they need a policy in place to mitigate those discoveries. People (especially developers) often need to be reminded not to post credentials, especially if they've been hardcoded, on a public website. (An email provider in Colorado found this out the hard way

when credentials that allowed access to core systems were left on GitHub.)

It's simple to check for and scrub credentials when posting to a repository, and is safer than relying on the repository's own security.

## Do Independent Testing

Independent testing of Distil's technology is what brought Engin Akyol, John Stauffacher, and Askal together. The trio built a bogus airline company, doing everything a normal company would do, and hooked it into Distil's solution. And then they observed what happened as they threw everything bad they could think of at it.

Their first step was to build a separate testing infrastructure, such that the environment was known to be stable. The goal was to make it resemble a production environment without actually going live. (QA and development environments are not stable enough for this kind of testing.)

The next step was to select a scraper. Import.io, Automation Anywhere, or Outwit are all good choices. The latter two both sidecar IE and browse right along with the user, extracting content into .CSV files and Excel spreadsheets faster than the blink of an eye.

For brute force testing, [THC Hydra](#) is an old application but one of the easiest and best to use, because it provides everything needed to bypass any mitigations already in place. It also retrieves the intelligence needed to figure out a response to this type of attack. [Metasploit](#) has its own modules, or home-grown options can be created using Ruby, JavaScript, or Python.

Several commercial tools are available for DDoS testing. For example, Loader.io is a commercial stress tester and includes a number of interesting features, such as specifying 1) scripts

to run and, 2) particular behavior patterns. Most support authentication and provide graphical reports that can be used to support architectural and response decisions.

Being highly scriptable, LoadImpact is a professional cloud-based tool for load testing, supporting different geographic regions. It runs as a Chrome plugin, has a huge reporting library, and a small web server agent can retrieve an impressive array of statistics.

Then there are the booters and stressers. These are slimmed down, less polished tools that offer a plethora of ways to lay a heavy hand on applications. No longer restricted to layer seven and using commercial botnets such as Zeus, they're extremely flexible and effective. Since they're using hijacked computers to achieve their goal, these are tools that should be used with care.

Stressor.io is yet another good DDoS testing resource. For about \$500, Stressor.io will power up 30 GB of zombie machines harnessed to cripple one target in layer three or layer seven. An even lower-budget option is Bees With Machine Guns, available on GitHub. It uses AWS to spin up random resources as zombies for testing purposes, but can result in a hefty AWS bill.

To train WAFs and other anti-bot solutions, simulated user traffic is key. Selenium is a commercial application that provides a scripting interface for user interaction. Phantom JS is what most bot writers use; it's better at manipulating Chrome, in addition to looking like Chrome or Firefox than either browser is able to look like itself. Capybara is a Ruby library that makes interacting with user interfaces extremely simple through its use of plain English commands. Finally, Splinter is a huge Python library that, like Selenium, is overlaid on the UI for ease of scripting a user session.

# Author Biographies



**Engin Akyol** is CTO of Distil Networks. He is responsible for the network design, procurement, provisioning, testing, and deployment of all infrastructure hardware, as well as Distil's HTTP

security service. Prior to Distil, Akyol was part of a QA "SWAT team" at Riverbed Technology, where he investigated corner cases, reproducing them in a lab environment to speed up resolution. At Cisco Systems he provided IT security, networking, and testing consulting as part of its elite ECATS (Enhanced Customer-Aligned Testing Services) group, comprised of top Cisco engineers.



**John Stauffacher** is the author of *Web Application Firewalls: A Practical Approach* and a world-renowned expert in WAFs, network security, and active defense. His research into

WAF program development, anti-bot defense and active defense has enabled him to be a trusted advisor to industry leaders, security vendors, and many of the Fortune 100. He has spoken at many industry events including DerbyCon 4, BSidesLA, GRR Con, LayerOne, LA2600, the inaugural Circle City Con, and Pumpcon (the longest running continuous hacker conference in the US).



**Yanky Askal** is Head of Ecommerce Operations for B&H Photo Video, the world's largest source of professional video, photo, and audio equipment. Askal oversees website security, application

and infrastructure administration, and end-to-end website performance/availability. He and his team are responsible for protecting B&H from a wide range of threats, including web scraping, competitive data mining, payment fraud, new account fraud, account hijacking, carding, email spam, phishing, reconnaissance attacks, account takeovers, and application denial of service attacks.

## Conclusion

Winning on today's bot battlefield takes time, expertise, and a dedicated team. The information provided in this paper provides a glimpse into the world of anti-bot defenses. If you're interested in exploring further how teaming up with Distil Networks can help you win the battle of the bots on your site, contact us today for a no-strings attached analysis and trial deployment.

## About Distil Networks

Distil Networks, the global leader in bot detection and mitigation, is the only easy and accurate way to protect web applications from bad bots, API abuse, and fraud. With Distil, you automatically block 99.9% of malicious traffic without impacting legitimate users.

Distil Web Security defends websites against web scraping, brute force attacks, competitive data mining, account takeovers, online fraud, unauthorized vulnerability scans, spam, man-in-the-middle attacks, digital ad fraud, and downtime.

Distil API Security protects all types of APIs including those serving web browsers, mobile applications, and Internet of Things (IoT) connected devices. Distil API Security defends APIs against developer errors, integration bugs, automated scraping, and web and mobile hijacking.

For more information on Distil Networks, visit us at <http://www.distilnetworks.com> or follow @DISTIL on Twitter.