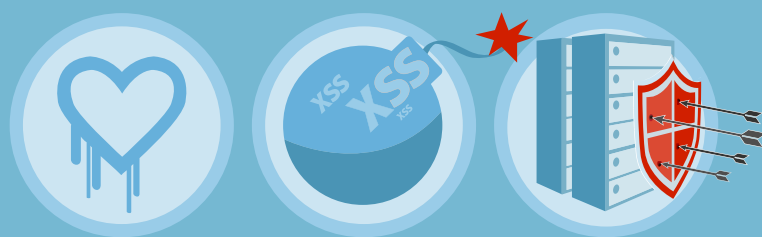


2014 Mid Year Security Threats Review

Indusface Security Threat Report

07 August 2014



CONTENTS

03 Introduction

04 Heartbleed

What is Heartbleed?	05
How Critical is Heartbleed?	05
What were the fixes?	05
Effect of Heartbleed: Then & Now	05
Initial Heartbleed victims	06
Is Heartbleed all cured?	07

09 Cross Site Scripting or XSS


What is XSS?	10
How does XSS work?	10
Types of XSS attacks	10
Creme de la creme of XSS attacks	11
How to protect your application from Cross Site Scripting of XSS?	12
Can your Web Application Firewall (WAF) protect you from Cross Site Scripting (XSS)?	13
State of Application Security in India: Infographic.	14

15 DDoS Attacks and all things Vulnerable

What are DDoS attacks?	16
Recent DDoS attacks	17
Types of DDoS attacks	17
Motives of DDoS attacks on Businesses	18
How to protect yourself against DDoS attacks?	18

19 About Indusface

Introduction



These past few months have kept us so busy! Branding it the year of Internet apocalypse might not be a far cry from truth. Barely six months have passed, and the vulnerabilities found have already registered themselves in the Internet hall of fame. With so much going on, we are sure you would have

missed out on some of the key blog stories...and since we understand your need to stay updated on everything important, we have worked on creating an eBook for you with all the key blog stories from Indusface, at one place. We hope you enjoy them and keep coming back for more.

All that you knew, and a lot more about Heartbleed:

The event that shook the World Wide Web



On April 7th, a major vulnerability called Heartbleed was discovered in OpenSSL, the most prevalent software used for encryption and other purposes on the web and the internet. SSL, known as secure socket layer, is the preferred protocol used for encryption. OpenSSL is its most common and open-source implementation. Websites that use encryption, payment gateways, VPNs, apps

— including mobile apps, all use SSL and a large majority of them use OpenSSL. The two most common web servers Apache and Nginx, that comprises of more than 60% of web servers on the internet use OpenSSL when they use the https (that is the encrypted version) version of http. Most operating systems use OpenSSL for various modules, so these modules are also affected.

What is Heartbleed Vulnerability?

Heartbleed vulnerability was discovered by three researchers — Neel Mehta from Google and two others. Heartbleed allows a malicious user to steal sensitive information such as private keys, passwords etc. The vulnerability is present in a module of OpenSSL called TLS heartbeat extension which is used to generate heart beat messages. Hence the name Heartbleed for this vulnerability. This heartbeat handshake is usually done during the negotiation time of the SSL protocol and much before https takes over, in case SSL is used under https. Thus, the vulnerability is not present in layer 7 but rather at layer 4.

How critical is Heartbleed?

Heartbleed is a critical vulnerability. To get into a bit more detail, the Heartbleed vulnerability allows a malicious user using a client to get 64K of memory from the server. Now, this memory can potentially contain sensitive data such as private keys. Once one gets the private keys, the server can be impersonated. Thus Heartbleed needs to be fixed or taken care of immediately.

While Heartbleed existed in the OpenSSL software since about two years back, it was discovered only recently. The disclosure was made public on April 7th along with a version of OpenSSL (1.0.1g) that has the fix. It could not be found if Heartbleed had been exploited. Further, if exploited, Heartbleed leaves no trace in the logs etc. unless one logs every SSL/TLS transaction which is hardly practical. This made it more difficult to find if it had been exploited in the past, but any server running a vulnerable version of OpenSSL was at risk.

What were the fixes?

OpenSSL had released a version of OpenSSL called 1.0.1g which would fix the vulnerability. This was released on 7th April. So, the best option was to upgrade from a vulnerable version to 1.0.1g. The other option was to recompile from source the version of OpenSSL which you were using but with the flag `-DOPENSSL_NO_HEARTBEATS`. This would disable heartbeats and so circumvent the vulnerability.

Further, it was important to regenerate all keys generated from a vulnerable version. In case one was using an OS such as Ubuntu, as versions of Ubuntu from 12.04 were vulnerable. Ubuntu had made an upgrade version (1.0.1-4ubuntu5.12) available at

<http://askubuntu.com/questions/444702/how-to-patch-the-heartbleed-bug-cve-2014-0160-in-openssl>

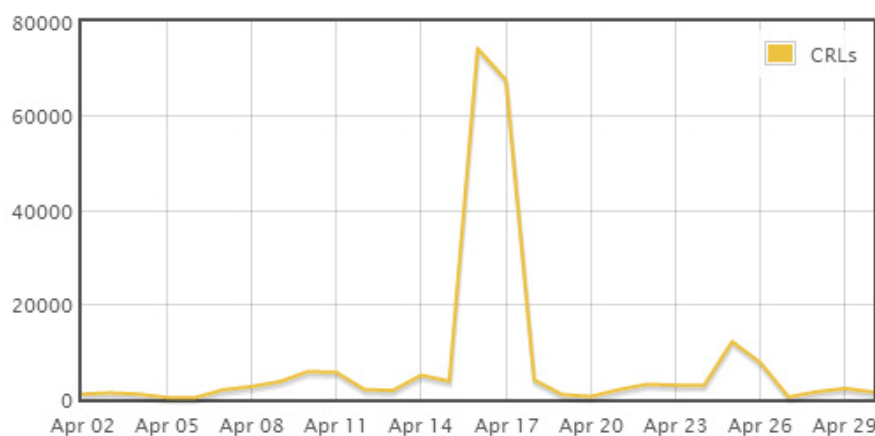
Effect of Heartbleed: Then and Now

The news of Heartbleed spread like a wildfire. Within days, multiple articles and websites came up with “The Heartbleed Hit List” - A list which comprised of most of the websites which were compromised and whose users were requested to change their password immediately. Big and popular names

like Google, Yahoo, and Dropbox were listed. There was a great confusion on changing the passwords- or not changing them.

In June, six more bugs were found in OpenSSL and OpenSSL came up with a security advisory detailing seven vulnerabilities with their fixed versions. While most of the vulnerabilities could lead to denial of service attacks and arbitrary code execution, one of the vulnerabilities (CVE-2014-024) allowed a hacker to launch a MITM (man in the middle attack) and snoop on unencrypted data. The hacker could thus look at sensitive data in the clear, beating OpenSSL's encryption. The impact for this could be enormous.

Heartbleed OpenSSL pulled with itself into limelight all the digital certificate-issuing authorities in the world. Emphasis on the need for support from them became overwhelming and none of them could shy away from that. The massive impact of this can be seen in the below graph, which shows revocations of a huge number of certificates within a few days.



The greater good that came from Heartbleed is the recent announcement of the Core Infrastructure Initiative (CII), which funds open-source projects that are in the critical path for core computing functions. Many large technology firms came in support to financially support the key open-source initiatives, signaling the beginning of the time where critical open source projects will be adequately backed up by the biggies.

CII will be supporting Network Time Protocol, OpenSSH, and OpenSSL and other key projects, such as the Open Crypto Audit Project.

Initial Heartbleed Victims

Many high profile websites came forwards as confirmed targets of Heartbleed attacks:

According to a Forbes article, Yahoo was exposed for about 24 hours where other sites like the Canadian Revenue Agency immediately took their website down while they worked on patching and remediation. The CRA commissioner Andrew Treusch later announced that 900 taxpayers' details including social security numbers, which could be used to gain access to

government benefits or perform identity theft, were exposed by an attacker using Heartbleed. Shortly following the announcement the Royal Canadian Mounted Police announced the arrest of Mr. Solis-Reyes who was accused of stealing the 900 records and is due to appear in court on July the 17th 2014.

Another victim of Heartbleed, Mumsnet, also announced to its users that it had been attacked. Mumsnet, is a popular British parenting website with 1.5 million users. As per Forbes, the e-mail to users stated “On Thursday 10 April we at Mumsnet HQ became aware of the bug and immediately ran tests to see if the Mumsnet servers were vulnerable. As soon as it became apparent that we were, we applied the fix to close the OpenSSL security hole... However, it seems that users’ data was accessed prior to our applying this fix”. Mumsnet posted an article outlining how the attacker was able to log in as the founder of Mumsnet, Justine Roberts after using Heartbleed to steal her username and password.

Is Heartbleed all cured?

Marco Ostini, an Information security analyst working at Australian Computer Emergency Response Team (AusCERT) at the University of Queensland, recently stated that the OpenSSL vulnerabilities are very close to being universal, and are not restricted to server-side computing. As a result, they are affecting almost every operating system, many of which are yet to receive the patches for OpenSSL vulnerabilities.

With time, IT infrastructure has become complex, and the security of them more so. Trying to manage them on your own can be tricky. It’s best to rely on a professional for the same. Many times, in the effort to save money, organizations decide to rely on their IT teams for security problems, which can work in some cases, but complex issues require more expertise. It’s best to nip the problem at the bud stage, rather than when it becomes a weed and becomes too difficult to manage.

The same issue has been faced by IT teams in the case of Heartbleed. The Heartbleed bug was a programming mistake which went un-noticed for many years. Being used by a very large section of the world’ internet, the remedial action to be taken was also of massive proportion. Securing the infected system not only required the updating of the software but also obtaining new “master keys” to re-establish their corporate electronic identity. In many cases, the users needed to be requested to change their passwords.

When six more bugs were found in June, more remedial efforts were required by the IT teams. We won’t be surprised if the cost of dealing with Heartbleed, globally, has already touched millions. And the money would be well spent if Heartbleed was fixed for good, but unfortunately the problem still persists.

There is a long list of products affected by OpenSSL Heartbleed, encompassing almost any IT product or service imaginable. One example is the 4.1.1 Version of Android’s Jelly Bean OS, which is still vulnerable, keeping

Australian security expert, Robert Graham’s research has shown that out of the 600,000 vulnerable servers identified by him post Heartbleed, 300,000 servers were still exposed, as recent as the end of June. While reason for this can easily be attributed to incompetency, but this will not be the truth.

the many android users at risk. While the fix is there, deploying it on such a massive scale is becoming difficult. This has led to a fear that many people might have stopped trying to install patches.

One might think that all is well now, and a few broken servers are not going to affect anyone, but that is not how IT security works. There is threat of data loss, yes...but more than that, a server which is compromised is like a huge gap in your protective fence, which if not mended will sooner than later give entry to lurking cybercriminals. Therefore, securing IT infrastructure requires diligent vigilance. One must have the right security tools in place and perform continuous website security checks that will share regular security updates to the business owners.

Internet is a crucial part of our everyday life. It is vulnerable to cybercrimes and cybercriminals and the aftereffects of Heartbleed have increased their vulnerability. The onus to ensure the security of our IT infrastructure falls on us, and this is a part which is non-negotiable.

Is XSS the KING of Vulnerabilities?



A month ago, TweetDeck, the popular social media dashboard application for management of Twitter accounts, had to be temporarily shut down, after being found vulnerable to cross site scripting (XSS). A teenager found an open vulnerability in TweetDeck's software and within hours the hackers community ensued a mass TweetDeck hijacking, despite TweetDeck being notified timely. The XSS in TweetDeck allowed JavaScript to become plain text where a computer code

was inserted, which when viewed in a user's TweetDeck, retweeted itself as a code. The result was a worm which even though unable to force the user to follow the attacker, did cause considerable damage as it replicated with the simple act of viewing and did not restrict itself to infect, only when clicked upon. The cause for all this fanfare was an XSS vulnerability that existed in TweetDeck since 2011, which they missed fixing...and the result was there for all to see.

What is XSS?

Cross site scripting, commonly known as XSS, is a very powerful hacking technique, which is much preferred by the hackers. XSS refers to a weakness in the design of a website which can then be used by an attacker to inject a malicious code into a website or web application, causing it to sway from its determined function. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites.

You are vulnerable if you do not ensure that all user supplied input is properly escaped, or you do not verify it to be safe via input validation, before including that input in the output page. Without proper output escaping or validation, such input will be treated as active content in the browser. If Ajax is being used to dynamically update the page, are you using safe JavaScript APIs? For unsafe JavaScript APIs, encoding or validation must also be used.

Automated tools can find some XSS problems automatically. However, each application builds output pages differently and uses different browser side interpreters such as JavaScript, ActiveX, Flash, and Silverlight, making automated detection difficult. Therefore, complete coverage requires a combination of manual code review and penetration testing, in addition to automated approaches. Web 2.0 technologies, such as Ajax, make XSS much more difficult to detect via automated tools.

How does XSS Work?

Cross site scripting, unlike other attacks like SQL injection, attacks the user of the application. Code is injected on client-side script such as JavaScript, on any of the many injection points in a website. Most websites have numerous injection points, such as search fields, feedback forms, cookies and forums, which can be vulnerable to cross-site scripting.

XSS can be used by a hacker to send a malicious script to the target, who will execute the script. The target's browser cannot test the authenticity of the script and will execute the script. The malicious script can then access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

XSS attacks work even over an SSL connection, as the script is run in the pretense of a secured site, takes advantage of the incapability on the part of browser to distinguish between legitimate and malicious content served up by a Web application. The only difference is that now the attack is happening on an encrypted connection.

Types of XSS Attacks

There are three types of XSS attacks- Non-Persistent, Persistent and DOM based Attack

Non-Persistent (or reflected) XSS attack is the most common type, which requires a user to visit the specially crafted link by the attacker. The crafted

code gets executed by the innocent user's browser when s/he visits the link.

The Persistent (or stored) XSS vulnerability is the more devastating variation of an XSS flaw, which when injected by the attacker, gets stored for a period of time in the server or a secondary storage device. The user need not interact with any malicious form or link, but simply view the page containing the malicious code.

In Document Object Model or DOM based XSS attack, the attacker tricks the web application to run the client side code in an "unexpected" manner. This means that the HTTP response or the page itself, does not change, but the clients side code behaves in a different manner due to malevolent modifications in the DOM environment. Therefore unlike the Persistent and Non-Persistent attacks, DOM based XSS is not dependent on the web server to receive the malicious XSS payload.

The principal reason behind XSS vulnerability is that the user-input is not properly validated and encoded. Neither are applications timely tested for vulnerabilities which can give rise to XSS attacks. Even if you have tested your applications once for vulnerability, the test needs to be done again, periodically, to ensure that no new technical vulnerabilities have arisen.

Crème de la Crème of XSS Attacks:

Despite being around since 1990s, XSS has now suddenly become the basis of most of the cyber-attacks. They can be avoided as security vendors can fix them easily. We are sharing few example here, of some major organizations, which were found to be vulnerable to XSS:

2011

In early 2011, several viral attacks were noticed on user walls by Facebook. The root cause of the attacks was the presence of three separate cross-site scripting (XSS) vulnerabilities on Facebook sites, which were uncovered within a period of about 10 days.

In the same year, a popular anti-virus security vendor's website was found to be suffering from a number of vulnerabilities which could easily allow XSS attacks.

2012

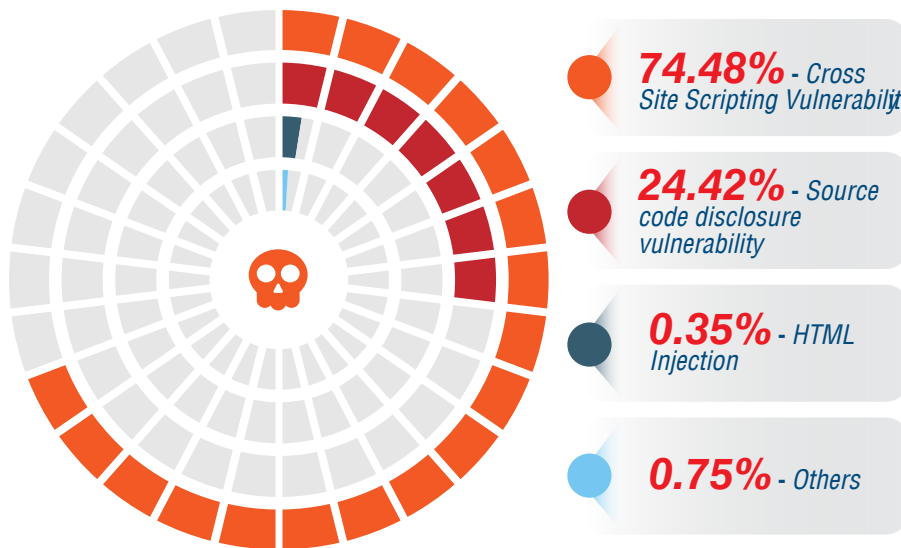
In 2012, Yahoo was affected by an XSS flaw which could be exploited by miscreants to hijack yahoo webmail user's accounts. A hacker was reportedly ready to sell the information about this flaw to a "serious contender" for \$700. What made the situation serious at that time was that Yahoo had been unable to find out the flaw, and thereby was unable to fix it!

2013

In 2013, a 17 year old school boy found a cross site scripting (XSS) vulnerability in the popular money transfer site PayPal.

2014

As we have shared before, Tweetdeck had to be shut down temporarily in 2014, due to an XSS flaw being discovered by a teenager and then exploited, by hackers.



How to protect your applications from Cross Site Scripting or XSS?

1. The preferred option is to ensure the proper escape all untrusted data based on the HTML context (body, attribute, JavaScript, CSS, or URL) that the data will be placed into. See the OWASP XSS Prevention Cheat Sheet for details on the required data escaping techniques.
2. Positive or “whitelist” input validation is also recommended as it helps protect against XSS, but is not a complete defense as many applications require special characters in their input. Such validation should, as much as possible, validate the length, characters, format, and business rules on that data before accepting the input.
3. For rich content, consider auto-sanitization libraries like OWASP’s AntiSamy or the Java HTML Sanitizer Project .
4. Consider Content Security Policy (CSP) to defend against XSS across your entire site.
5. Turn off the HTTP TRACE support on all web servers. Cookie data can be stolen by the attacker via JavaScript even if the document cookie is disabled.
6. It is possible to patch XSS but it can never be ensured that the filter

cannot be broken into. The best solution to protect yourself from XSS is to sign up with a security vendor, who study most of the available XSS vectors as part of their regular research and work on the codes to identify the attack pattern and thereby provide total protection against it

7. Scan your applications continuously for vulnerabilities. Continuous vulnerability assessments of your applications help you in finding the vulnerabilities before the hacker does, and fixing them.
8. Add a Web Application Firewall (WAF) to your defense layers. Gartner mentioned in a recent report that any organization which owns a public website, makes internal Web Applications available to partners and clients, or has business critical internal web applications, should consider investing in WAF. In the attack on TweetDeck, the tweet that started it all, was added with a script which forced the simulation of the retweet button. A WAF could have blocked the keywords resulting in the formation of the malicious script and could have prevented the attack altogether.
9. Sign up for a Hybrid analysis - combine web application scanning with a managed web application firewall, which works on behavioral analysis. A managed WAF scans your traffic and creates rule based on that, thereby evolving the protection as per the traffic.

Can your Web Application Firewall (WAF) protect you from Cross Site Scripting (XSS)?

A web application firewall protects web applications against threats like XSS and SQL. It monitors the inbound/outbound HTTP/S traffic and blocks keywords that lead to XSS. A hacker trying to XSS a site inputs JavaScript or other similar scripts. By blocking these scripts, one prevents hackers from pushing such malicious scripts into browsers.

Many might argue that the best way to protect your applications is for developers to produce secure codes and any vulnerable codes should be tried to fix at the source itself, but the practicality of the situation is that there are simply too many vulnerabilities to actually keep remediating them, before they cause harm. And new vulnerabilities keep coming to limelight every day. The latest example which will not be forgotten in times to come is the Heartbleed vulnerability.

Along with filtering traffic to clear illegitimate traffic, a managed WAF with custom rule sets can protect your vulnerable web applications from attacks.

The state of Application Security in India

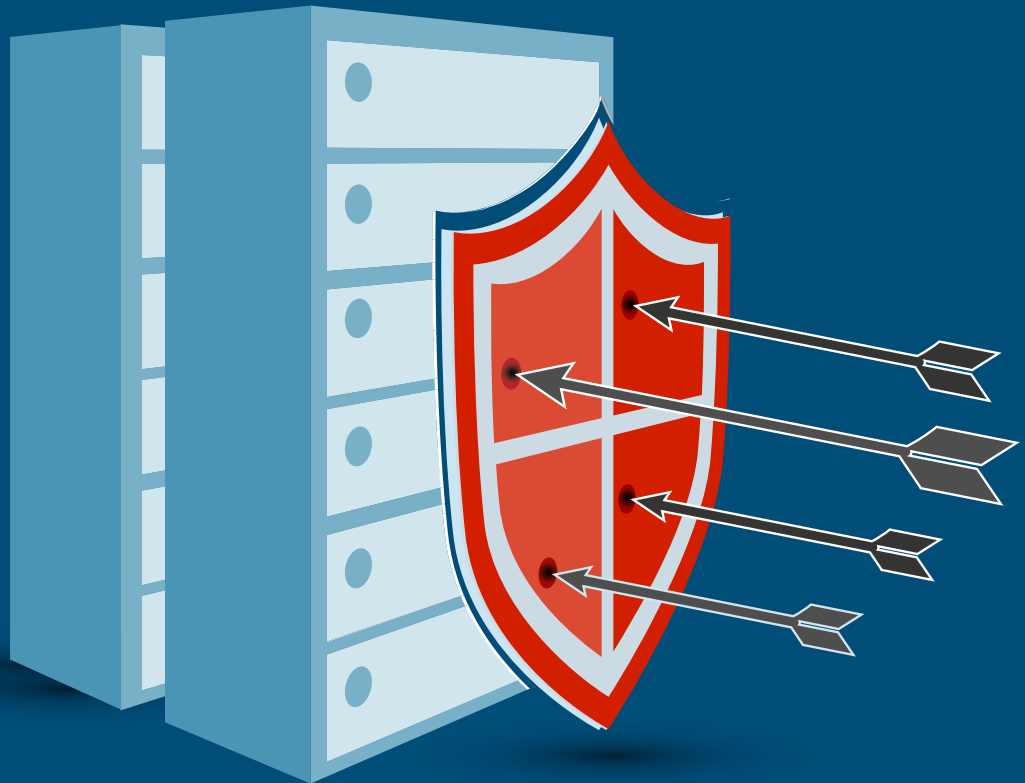
XSS TOPS THE CHART!

We had analyzed the vulnerabilities data collected by us during the course of 3.2 billion ethical hacks and 2 million scans that we had conducted for our customers. Our web application security solution, found that Cross Site Scripting (XSS) tops the high vulnerabilities list, while SQL injection is the top most critical vulnerability.

You can access this infographic on “*The state of Application Security in India*” by [clicking on the link below](#).

DOWNLOAD INFOGRAPHIC

DDoS attacks and all things Vulnerable



DDoS attacks have been in news a lot lately. They have evolved from the non-dangerous threats into a more deadly occurrence which the hackers are using to attack on a much more severe and larger scale than before. DDoS attacks are targeting and attacking more people in a lesser time using more advanced techniques.

The month of June, alone, played host to a series of high-level DDoS attacks. Evernote, Feedly, World cup websites, again Feedly, Hong Kong Voting Site were the victims. On 19th June, the social media Giant, Facebook, went down for half an hour, during early hours of 19th June/Thursday, resulting in the longest downtime in the recent history of Facebook last week. The event was widely attributed to a DDoS attack, a news which was not confirmed, but was broadly believed.

What are DDoS attacks?

A Distributed denial of service (DDoS) attack is one in which a multitude of compromised computers attack a single target, thereby stalling traffic for the legitimate users of the targeted system. Attackers build botnets (a network of infected computers), and take control of them remotely, without the actual owner having any knowledge of this. The infection is dispersed to these computers, via spreading malicious software through various online activities like emails, websites etc. The attacker can then, at the chosen time, use it's botnets to attack the target application, by sending huge amount of traffic to the victim.

The large flow of requests from the compromised systems, to the targeted system essentially forces the target system to shut down or report as out of service due to bandwidth issues, thereby paralyzing the targeted system.

Lately, the DDoS attacks have become much more concentrated and sophisticated. The hackers have started directing 200 Gbps to 400 Gbps of traffic towards the victim's servers, which is typically hundreds of times the normal bandwidth of websites. Also, while initially DDoS attacks were simple and mainly involved sending lots of traffic towards the victim's server, the new brand of attackers are doing their homework and changing strategies according to the target. They are finding the least point of resistance and using it against the target.

In 2013, majority of banks were targeted by Hackers with DDoS attack and nearly 50% of the banks stated that they had been targeted more than once.

Ponemon

Recent DDoS Attacks

Major online biggie's victim of DDoS attacks:

DDoS attacks have rapidly become hacker's choice of attack, with evidently many major businesses falling at the receiving end. On June 10th, Evernote, popular online-note taking and web clipping saving service, became a victim of a similar attack. The attack disrupted their services for almost four hours, affecting many of its 100 million registered users. Members were unable to synchronize their filings during the attack.

Unlike before, researchers are now finding that such DDoS attacks are no longer a one-time event, but a precursor to more attacks and sometimes, these attacks also acts as smokescreens for various other malicious activities. This fact was proven true by the series of attacks on Feedly. On 11th June, Feedly, the very popular news aggregator which provides content from various online sources at one place, became the next victim of a DDoS attack. They confirmed in a post that they were being targeted by a DDoS attack and the attackers were trying to extort money from them in lieu of stopping the attack. They refused to give in to the extortion and announced that they had neutralized the attack, within 12 hours of the first post, but the very next morning, they came back with the news of second attack.

The threat was again neutralized within hours and then they were attacked again...the third time. The news of Feedly hitting the third day of downtime due to DDoS attack was everywhere. This was the final attack and was neutralized in a very short time. Feedly laid emphasis on the news that none of the customer data was lost in this attack.

DDoS Attacks Hit the World Cup!

While football fever struck worldwide, a major DDoS attack struck the official government World Cup website, which went down for more than a day.

Another name in this list of DDoS victims was of Hong Kong Democracy Poll, where attack was fended off by diverting most of the traffic to sinkholes. But the problem with sinkholing or black-holing is that though it diverts the traffic to a sinkhole where it is discarded, segregation between good and bad traffic cannot be done. This means that all traffic, whether good or bad, is discarded. While DDoS is bad news for organizations, resorting to sinkholing cannot be considered as an alternative.

Types of DDoS attacks

Essentially, there are two types of DDoS attacks-Application layer attack and network Layer attack. The first takes place at the application layer (Layer 7) and the second at the network layer (Layer 3 and 4).

In the network layer DDoS attack, hackers bring down a website by overpowering the network resources and thereby hampering the ability of the network to respond to legitimate users' requests.

Application-layer attacks, on the other hand target applications. Also known as Layer 7 attacks, these are DDoS attacks which overload an application

server by sending mass requests like log-in, downloads, etc. These requests impersonate the user traffic such that they appear to be coming from legitimate users, and make the web server unavailable to the actual users, causing the business to lose customers. Application layer attacks are done with the specific intention of harming the victim's business by disrupting customer access and transactions, thereby causing financial and brand loss.

DDoS BOT traffic has increased by 240% and more than 25% of this traffic originates from India.

Motives of DDoS attacks on Businesses:

Application Layer Attacks are on rise lately, mainly due to the financial factor linked to them. Equally, DDoS attacks are targeted to serve diverse motives:

1. Competitive brand damage
2. Financial gain: This falls under two categories.
 - a. Monetary benefit thought extorting money
 - b. Gain advantage over Competitors
3. Cause business loss: Especially for organizations that operate online
4. Revenge: Attacks against Government websites fall under this category. Ex-employees, competitors are also the likes who in this list. A recent example is of a prominent player in online industry being targeted by a series of DDoS attacks, soon after the exit of an ex-partner.
5. Smokescreens to cloak other malicious activities
6. Cyber warfare
7. To create nuisance

How to protect yourself against DDoS attacks?

1. Special DDoS prevention boxes can be used to thwart high speed DDoS attacks. Many of them connect to routes upstream to figure out the origin of the DDoS attacks and then block them.
2. DDoS attacks can take place at both the network level and the application level. A network firewall can be used to block the traffic in case the DDoS attack is at the network level.
3. At application layer, a technology as the only solution to block DDoS attack is very risky but can be used effectively as a suspicious DDoS alerting mechanism with targeted rules and with human intervention for analysing and if it is indeed a DDoS, taking action to block it.
4. A Managed WAF with DDoS prevention rules with right thresholds configured for raising alerts along with human intervention to act on those alerts, can be used to block the traffic in case the DDoS attack is at the application level. In other words, your WAF vendor manages the incoming traffic by its behavior profiling, which is done with the help of manual intervention. Once this is done, the appropriate security policies can be applied to mitigate DDoS attack.
5. DDoS mitigation is required for defense against Layer 7 attacks. Proper DDoS mitigation requires the authentication of incoming traffic to separate legitimate human traffic from human-like traffic which is actually originating from bots.

TOTAL APPLICATION SECURITY THAT DETECTS, DEFENDS AND PROTECTS

The IndusGuard suite of products provides next generation, SECaaS and cloud based, customizable application security and regulatory compliance solutions that are easy to deploy and manage, highly efficient, cost effective, and work cohesively in capitalizing years of in-depth security intelligence, providing organizations with total application security that detects, defends and protects application assets.



Indusface is a privately-held, award winning, innovative, visionary, fast growing, cloud based, application Security-as-a-Service (SECaaS) company, which caters to over 600 customers worldwide.

Gartner ■ Magic Quadrant
for Application Security Testing

Finalist
DSCI
Excellence Awards
2013

PCI
Approved
Scanning Vendor

Deloitte.
Tech Fast 50
India & 500 Asia

RED
HERRING
100
ASIA
WINNER

certn
Empanelled