Hacker Intelligence Summary Report
The Anatomy of a Hacktivist Attack

Web Application Security

Web Apps

Executive Summary

Hacktivism exploded onto the scene in late 2010, causing Website outages, defacements, and data breaches. Since that time, hacktivism has become a costly security threat as well as a justification for cybercrime and a cloak for government-sponsored attacks. Imperva witnessed a recent hacktivist attack that lasted for 25 days. Our observations give insight into hacktivist attack methods and an examination of how social media enables recruitment and attack coordination. Understanding how hacktivists operate will help organizations prepare for a future attack.

Our observation of the hacktivist campaign reveals:

» The process used to pick victims as well as recruit and use needed hacking talent.

» How hacktivists leverage social networks to recruit members and promotes hack campaigns.

» The specific cyber reconnaissance and attack methods used by hackers. We detail and sequence the steps that hacktivists perform to bring down Websites and steal data.

Finally, we recommend key mitigation steps that organizations need to help protect against attacks.

Methodology

This report is based on a hacktivist attack observed by the Imperva Application Defense Center (ADC). The target organization of the attack had a Web application firewall deployed which recorded and repelled the attacks. By analyzing traffic logs, we analyzed the attacks on these applications and categorized them according to the attack method, as well as identified patterns

iMPERVA®

DatabaseFileWeb

and trends within these attacks. We also analyzed social media communications related by the hacktivists in the days leading up to and after the attack. We believe this is the first end-to-end record of a major hacktivist attack.

To prevent a repeat attack, we will not name the targeted organization.

The Plot

Hacktivists emerged in 2010 and rocketed to prominence in mid-2011 after a string of successful application attacks. These attacks fell into two categories:

> » **Reactive**: An incident inspired hacktivist members to attack a target. For example, when MasterCard, Visa, and others stopped allowing payments to Wikileaks, the hacktivist group Anonymous began Operation Payback intended to bring down Websites with excessive traffic. When BART police blocked the use of cell phones in certain stations, Anonymous hacked into BART computers, exposing the data of dozens of employees.

> » **Proactive**: A hacktivist group announces its intention to attack a target before the assault. Proactive attacks are less common; there have only been a few incidents. For example, threats against Facebook and Mexican drug lords were made, but attacks either fizzled or never even materialized. It is difficult to estimate how many proactive attacks have occurred since, like terrorist attacks; only successful campaigns become public.

The attack Imperva witnessed was of the proactive variety. In this case, hacktivists hoped to disrupt an event that would take place on a specific date. The attack targeted a Website designed to support the event. Though we cannot identify the target, it is a large, well-known organization.

The attack occurred over a period of 25 days in three phases. During the recruiting and communications phase, a small group of instigators elicited support and recruited for the attack. Hacktivist members created a Website rationalizing an attack on their target. Twitter and Facebook promoted traffic to this site. Additionally, YouTube videos were produced to help rationalize attacks. Once a critical mass was achieved, the second phase, reconnaissance and attack, could begin. During this phase, around 10 to 15 skilled hackers probed the Website's applications in an effort to identify weaknesses that could lead to a data breach. The third and final phase was a Distributed Denial of Service (DDoS). Having failed to expose data, hackers obtained help from nontechnical hacktivist recruits. Several hundred to a few thousand people either downloaded attack software (such as was done in Operation Payback) or went to custom-built Websites that perform DDoS attacks. When the DDoS failed, the attack ended.

< **2** >

iMPERVA®

DataBaseFileWeb

Introduction: What have we learned about hacktivists?

Since the emergence of hacktivism, news reporters and security experts have followed hacktivists' exploits closely. Although the results are well known – publicly exposed data and interrupted Web services – the methods are much less clear. Our findings show:

» **Hacktivists are real people with real techniques – but they use conventional black hat methods and technologies.** In fact, their hacking methods very much mirror what profiteering hackers do daily. For example, hacktivists use many of the same tools for hacking, such as Havij, a SQL injection tool (probably invented in Iran[1]) designed to penetrate applications and steal data. In other words, they are able to take advantage of common application vulnerabilities found in many Websites, the same thing that fuels today's black market, data-driven cyber crime economy. The main innovation seen from hacktivists is the creation of many Websites that perform denial of service attacks.

» **Hacktivists will try to steal data first and, if that fails, attempt a DDoS attack.** The first major hacktivist attack, Operation Payback, was a DDoS attack targeting PayPal, Visa, MasterCard and others. Though the attack attracted a lot of attention, it failed to disrupt these companies' operations. Other attacks on companies such as Sony succeeded. The impact? Sony suffered a public relations debacle in the period following the attack and subsequent data breach. The lesson was not lost on hacktivists who continued with data-centric attacks on PBS, BART, and other organizations.

» **Hacktivist groups are comprised of two types of members:**

  • **Skilled hackers**: In the campaign Imperva witnessed, a small group of skilled hackers organized the assault. In total, this group numbered no more than 10 to 15 individuals. Given their display of hacking skills, one can surmise that they have genuine hacking experience and are quite savvy.

  • **Laypeople**: This group can be quite large, ranging from a few dozen to a few hundred volunteers. Directed by the skilled hackers, their role is primarily to conduct DDoS attacks by either downloading and using special software or visiting Websites designed to flood victims with excessive traffic. The technical skills required range from very low to modest.

  In this incident, there was about a 10:1 ratio of laypeople to skilled hackers.

» **Hacking operation fell into three distinctive phases:**

  1. **Recruiting and communications phase (Day 1-18)** – In this phase, hacktivists leverage social media to recruit members and promote their campaigns. In particular, they use Twitter, Facebook, and YouTube to suggest and justify an attack. If a sufficient number of volunteers are persuaded to participate, the skilled hackers begin initial reconnaissance.

  2. **Reconnaissance and application attack phase (Day 19-22)** – During this phase, the skilled hackers carefully hide their true identity and place of operation. They probe applications in an effort to identify weaknesses that could lead to a data breach. They use well known vulnerability assessment tools, such as Nikto and Acunetix, to identify potential holes that could lead to data theft.

  During the attack phase, skilled hackers raise the bar and use attack software specifically designed to take data. As mentioned previously, one tool, probably developed in Iran, conducts a high volume of SQL injection attacks. Havij picks up where traditional penetration testing tools stop, actually performing data extraction and harvesting instead of just pointing to potential vulnerabilities.

  3. **DDoS phase (Day 24-25)** – If the data breach attempts fail, the skilled hackers elicit help from the laypeople. At this point, a large volume of individuals download attack software or go to custom-built Websites that perform DDoS attacks.

---

[1]   Havij in Farsi means "carrot" and is used in Iran as slang for the male sexual organ.

DatabaseFileWeb

»» **They have developed some custom-attack software that can be used on computers as well as mobile devices.** In the past, they refined an open-source stress testing and DDoS tool to develop the so-called Low-Orbit Ion Canon (LOIC). In this case, they also developed a DDoS tool that allows users to attack sites with mobile browsers. However, their mobile tool, though innovative, is not complicated. In fact, it is probably just a few hundred lines of JavaScript code which enables any device – PC, Apple, mobile device – to perform an attack by virtue of just having a Web browser.

In this attack, hackers created a Web page that contains a JavaScript. The script iterates endlessly (as long as the page is open in the browser) and generates a new image attribute. The source of the image is the victim's Web page, and the script creates multiple requests to the victims' Website as the page is rendered by the browser. In other words, all it takes for an attacker to participate in the attack is to browse to the specific Web-page and leave the page open. No need to install or download any software. This is what makes this technique so simple to use, as opposed to other methods. Since the code is written in JavaScript, it enables any device equipped with a standard browser to take part in the attack – and indeed we have seen mobile devices participating in the attack.

»» **Attack velocity is critical.** Hacktivists can't attack at will. Rather, they are subject to the dynamics of crowd-sourced hacking. This means someone must make a compelling case for attack, which requires persuasion and recruitment. This takes time – and if there's a specific event to disrupt – then a deadline looms. From a hacking perspective, this restricts the available hacking activity to taking targeted shots as opposed to setting cyber traps. This is in strong contrast to the hacking methods of government-sponsored hackers who can be more patient. For example, government-sponsored groups rely heavily on phishing, whereas hacktivists do not.

»» **Hacktivists use inexpensive, off-the-shelf tools as opposed to inventing new techniques or developing complex attacks.** Advanced, hard-to-detect attacks are a hallmark of government-sponsored cyber-attacks – but hacktivists operate out in the open. Their use of off-the-shelf-attack tools that are commonly available and cheap – in some cases free – to acquire. A typical hacktivist attack requires virtually no financial investment.

»» **There are several key differences with profiteering hackers.** The crowd-sourced hacking model restricts the use of several commonly used hacking techniques, including:

 • **Sporadic use of bots** – Bots are typically rented, incurring a cost. Since hacktivists rely on volunteers, bots are not always available. In the campaign we observed, no bots were used. Analyses of chat discussions for other hacktivist campaigns, such as Operation Payback, show that some botnet owners have offered the use of their bot armies to help conduct attacks, though no direct evidence exists that botnets were used.

 • **No reliance on malware** – There is no current evidence that hacktivists deployed malware. In the event we observed, malware was not used.

 • **No phishing or spear phishing** – Developing alluring emails with malware attachments or malicious links typically takes time to execute. This does not fit into hacktivists' modus operandi of conducting rapid attacks.

 • **Public recruitment phase** – In private hacking, recruitment takes place on hacker forums, typically in private communications. By contrast, hacktivists recruit through social media outlets in broad, public view. If IT security teams monitor social media, they can anticipate proactive attacks.

Database File Web

Hacktivism in Brazil

Brazil is famous for many things: Ayrton Senna, jiu-jitsu and caipirinhas. In cyber security, however, Brazil may add hacktivism to the list. Though unrelated to the incident, which is the subject of this report, Brazil's experience with hacktivism sheds light on how it is born and then thrives.

In the past twenty four months, there has been an increase in high-profile hacktivism in Brazil. Note the recent, high volume of attacks on several government and private enterprises. Starting in May 2011, TIM Mobile Operator suffered a DDoS attack linked to Anonymous.[2] Next month, inspired by the World Wide #AntiSec Operation, LulzSecBrasil was formed. Wasting little time, they attacked Websites belonging to Brazil's President, Brazil's biggest gas and oil firm Petrobras, Brazil's tax agency (Receita Federal), the Ministry of Sports, and Brazil's biggest newspaper, *Rede Globo*. The group also disclosed the personal information of Brazilian president, Dilma Rousseff and Mayor of São Paulo, Gilberto Kassab.[3]

A month later, Brazilian LulzSec ceased operating. This opened the door for Anonymous Brazil. They quickly hacked a Brazilian Federal Police Agent and some of her e-mails were leaked and exposed online. Credentials from Petrobras employees and Municipal Chambers of Uberlandia were leaked and exposed. Several Brazilian political parties now were targeted and had their Websites hacked.[4] In January 2012, a total of nine major bank or government agencies were attacked, including Ciello (Visa in Brazil), Citibank Brazil,[5] Banco Central, Caixa Economica Federal, HSBC, Bank Itau and Bradesco.[6]

Why is hacktivism hitting Brazil with such intensity? Short answer: Twitter. Today, Brazil ranks second after the US in Twitter usage. Ironically, the widespread use of Twitter has given hacktivism an elevated communications platform for recruitment.

First, a note on some socioeconomic history in Brazil as it pertains to internet use. According to a *Time* magazine article:

> *Much of Brazil's transformation can be seen through the spread of telecommunications and the growth of social media. With telephone landlines once the preserve of wealthy elites, millions have turned over the years to mobile phones as their primary connection. That shift worked well with Twitter, which entered the Brazilian market first as an SMS service. To confront the gap between rich and poor, both the government and private NGOs sought to introduce computer technology to the poorest classes as early as the beginning of the 1990s. "Brazil was a pioneer in creating democratic access to computers and Internet for the poor, well ahead of the United States," says Green. And while many favelas are still excluded from the electric grid, the country's "Popular PC Project" of installing cheap computers in poorer areas has become a model the world over.*

How has this translated into hacktivism? When hacktivism first started, hacktivists attracted the attention of many Brazilians. More importantly, many of the hacktivist targets, Brazilians perceived, deserved attack. In the minds of many Brazilians, the cyber mayhem was no crime.

Anonymous Brazil touched a populist nerve. The main innovation? They made DDoS accessible to the masses. Anyone with a browser – even a mobile browser – could participate in an attack. You could see the fruits of your labor as target sites went down with a populist cyber riot. No pitchfork required.

Are there any lessons for countries to learn? While no two countries are identical culturally or economically, there are some factors to consider. Clearly, having strong government or corporate resentment in a population is a key factor – it's hard to see the same dynamic ever occurring in Norway. Second, monitoring Twitter will be a critical corporate and law enforcement activity. Recently, the FBI issued a request for information for social media monitoring.[7] It's a safe bet that Brazilian enterprises will be doing the same – if they aren't already.

2   https://twitter.com/#!/Anony_Ops/status/65303916757266432
3   http://g1.globo.com/tecnologia/noticia/2011/06/hackers-divulgam-supostos-dados-de-politicos-na-internet.html
4   http://www1.folha.uol.com.br/poder/934535-policia-federal-investiga-ataques-de-hackers.shtml
5   http://online.wsj.com/article/SB10001424052970203889904577200964142208498.html?mod=WSJ_Tech_LEFTTopNews
6   http://topstories.foxnews.mobi/quickPage.html?page=17224&external=1313500.proteus.fma&pageNum=-1
7   https://www.fbo.gov/index?s=opportunity&mode=form&id=c65777356334dab8685984fa74bfd636&tab=core&_cview=1

@iMPERVA®

How can companies prepare for a hacktivist attack?

If companies have implemented solid defenses to stop application layer attacks like SQL injection, cross site scripting, local file inclusion and DDoS, then they will be equipped to handle a hacktivist attack.

What are the lessons?

» **Any organization can be a target.** There is not a lot of consistency to hacktivists' campaigns; their targets include a wide range including religious organizations, pornography sites, consumer electronics firms, banks, Mexican drug lords, law enforcement, and government.

» **The threat is real if applications are vulnerable.** By using good application security standards, potential targets can reduce their risk.

Attack Timeline

A notorious hacktivist group launched a 25-day attack on a well-known Website. Imperva's ADC team was able to obtain and analyze the attack traffic from start to end.

Our research concludes that the attack consisted of three phases:

» **Recruiting and communications phase (Day 1-18)** – This is really the essence of all hacktivism campaigns. The raison d'être of hacktivism is to attract attention to a cause, so this phase is critical. Messages were spread via social media applications, in particular, Facebook, Twitter and YouTube.

» **Reconnaissance phase and application attack phase (day 19-22)** – A few days before the attack's "D-Day", a group of savvy hackers conducted a survey on the security of the target Website. This phase lasted three days.

The reconnaissance group members:

• Had knowledge of hacking tools.

• Used anonymity services to disguise their identity.

• Used just a few attack sources (compared to the attack phase).

• Kept a low profile. Their attack traffic levels during this phase were relatively low, especially when compared to the attack phase. However, the reconnaissance traffic was relatively high compared to ordinary days.

Eventually, the attackers tried to penetrate Web applications with "off-the-shelf" attack tools. After failing to find such vulnerabilities, they had fallen back to the option of launching a DDoS attack against the Website. The application attack phase lasted about three days.
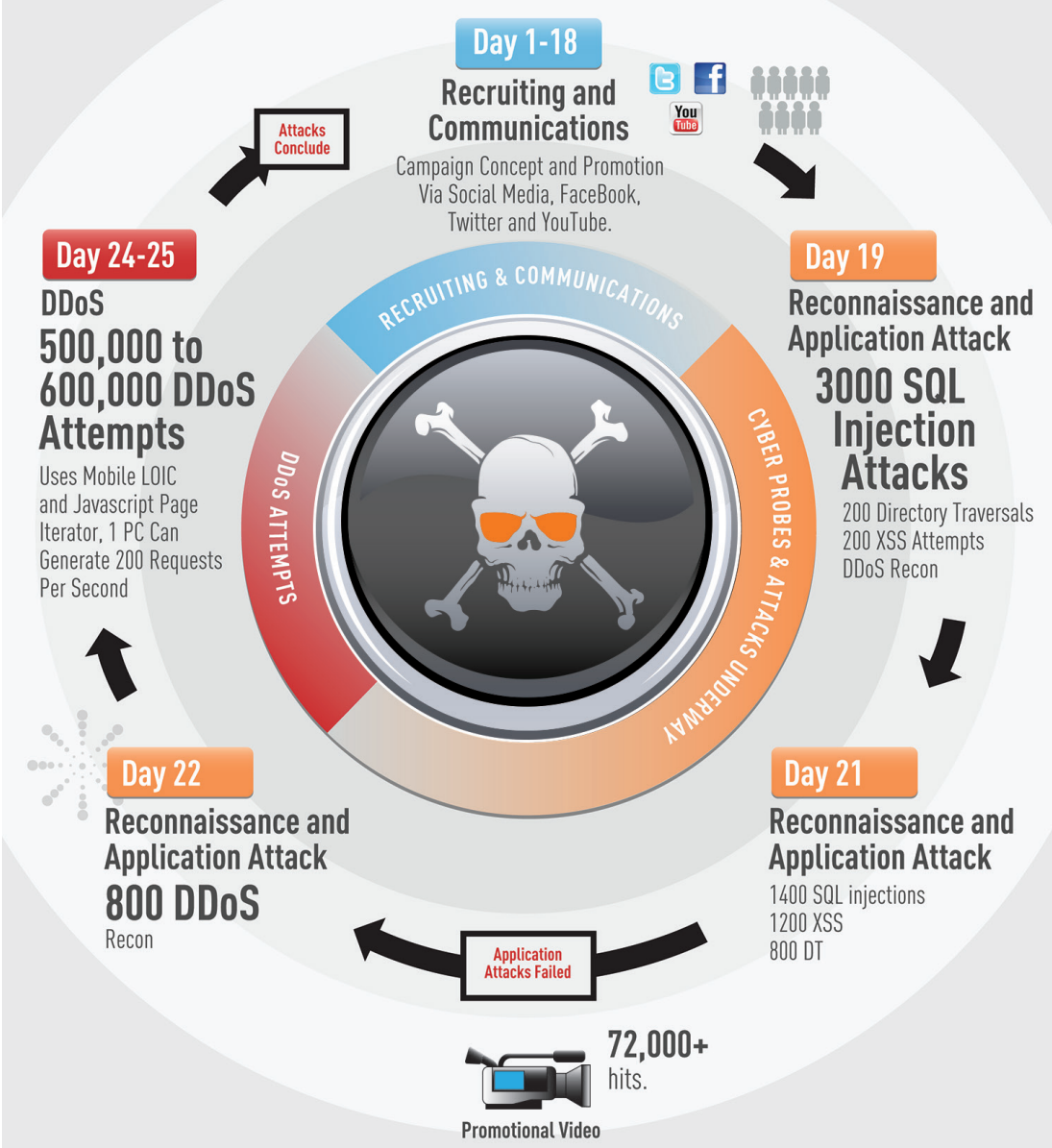
» **DDoS Attack phase (day 24-25)** – Following the findings of the reconnaissance group, a DDoS attack was launched against the targeted Website.

The attack tool used for the DDoS attack was coded in the JavaScript language and hosted on a Website. This type of attack is commonly referred to as "Mobile LOIC." In this campaign, some portion of the attacks was conducted from mobile devices.

The ADC was able to obtain the attack tool, analyze its source code, and test its performance in a captive environment.

The attack traffic was considerably higher than on the reconnaissance phase and conducted by many more attackers, but it was done with less security awareness as most of the attackers didn't use the protection of anonymity services.

DatabaseFileWeb

THE ANATOMY of A HACKER ATTACK
HACKER INTELLIGENCE SUMMARY REPORT

Day 1-18

Recruiting and Communications

Campaign Concept and Promotion Via Social Media, FaceBook, Twitter and YouTube.

Attacks Conclude

Day 24-25

DDoS
500,000 to 600,000 DDoS Attempts

Uses Mobile LOIC and Javascript Page Iterator, 1 PC Can Generate 200 Requests Per Second

Day 19

Reconnaissance and Application Attack

3000 SQL Injection Attacks

200 Directory Traversals
200 XSS Attempts
DDoS Recon

RECRUITING & COMMUNICATIONS

DDoS ATTEMPTS

CYBER PROBES & ATTACKS UNDERWAY

Day 22

Reconnaissance and Application Attack

800 DDoS
Recon

Day 21

Reconnaissance and Application Attack

1400 SQL injections
1200 XSS
800 DT

Application Attacks Failed

72,000+
hits.

Promotional Video

Learn more at: blog.imperva.com

Source: Imperva

Recruiting and Communications Phase (Day 1-18)

The attack campaign targeted the disruption of a specific event. The hacktivist group recruits and communicates through Web 2.0 media outlets. The hacktivists used multiple channels, including Facebook, Twitter, and YouTube.

The content during this phase:

» Explained their political agenda for the campaign. In this case, a Website was created that rationalized the attack. Twitter and Facebook were used to bring attention to the Website and its arguments. Further, YouTube videos further rationalized the attack by denigrating the target and exposing perceived transgressions.

» Declared the dates and targets of protest to recruit protesters and hackers.

Reconnaissance and Application Attack Phase (Day 19-22)

The reconnaissance phase took place for four days and consisted of:

» Scanning for general Web application vulnerabilities

» Scanning for DDoS relevant pages.
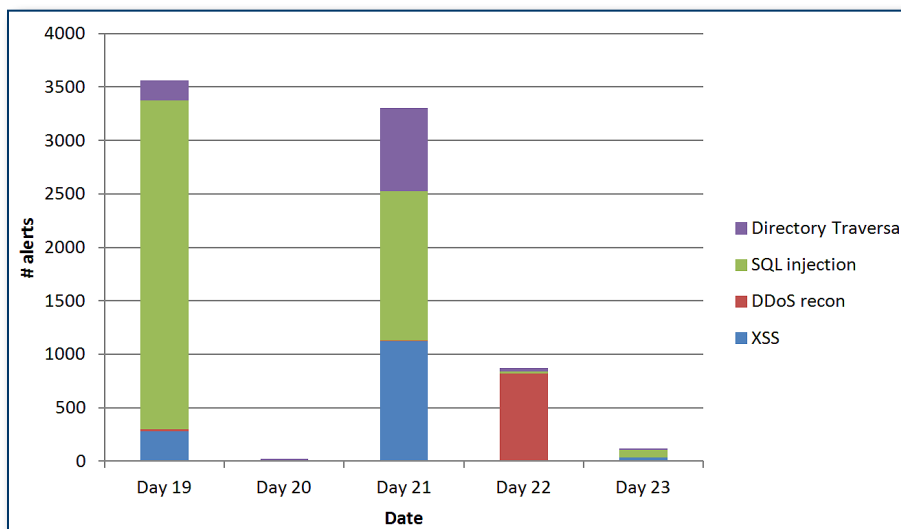
Figure 1 – Attacks used during reconnaissance phase.

Figure 1 shows the number of different types of attacks during the reconnaissance phase. The attackers first tried to check for general Web-application vulnerabilities, such as XSS, SQL injection, and Directory Traversal. Notice the high volume of attack traffic (thousands) on these days.

Only when failing to find such vulnerabilities, the attackers resorted to searching a resource suitable for DDoS. Such resources involve actions which are time and resource consuming and might lead to exhaustion of the server resources. Eventually, attackers spotted a specific URL that was later used in the attack itself.

Attack Tools

There was an extensive usage of various automated tools for Web application vulnerability scanning. These scanners are general "off-the-shelf" products and weren't coded specifically for this attack. All attacks on the application were blocked and logged by a Web application firewall.

A few prominent examples include:

Havij Scanner

Havij is an automated SQL Injection tool that helps penetration testers to find and exploit SQL Injection vulnerabilities on a Web page. By using this software, a user can perform back-end database fingerprint, retrieve DBMS users and password hashes, dump tables and columns, fetch data from the database, running SQL statements and even access the underlying file system and executing commands on the operating system.[8] We believe Havij was probably developed in April 2010 in Iran. Today, Havij earns kudos for its accuracy and ease of use.[9]

Note the User-Agent Havij keyword:

In the following example, the "Havij magic number" was used in order to find the number of columns:

Acunetix Scanner

The Acunetix Web Vulnerability Scanner is an automated black box scanner that checks Websites and Web applications for vulnerabilities such as SQL injection, Cross Site scripting, and other vulnerabilities.[10]

The scanner was used mostly to look for Remote File Inclusion (RFI) vulnerabilities:

---

[8]   http://www.vulnerabilitydatabase.com/2011/06/havij-v1-15-advanced-sql-injection-released
[9]   http://itsecteam.com/en/projects/project1.htm
[10] http://www.acunetix.com/vulnerability-scanner/

Nikto Scanner

Nikto Web Scanner is a Web server scanner that tests Web servers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received. The Nikto code itself is Open Source (GPL).[11]

```
GET        .php ? root= </script><script>alert("Vulnerable")/<script>   HTTP/1.1
User-Agent: Mozilla/5.00  (Nikto/2.1.4)  (Evasions:None) (Test:000831)
X-Akamai-CONFIG-LOG-DETAIL: true
TE: chunked;q=1.0
Connection: TE
Accept-Encoding: gzip
```

Web Application Vulnerabilities Reconnaissance Traffic Analysis

The ADC had analyzed the attack sources of the Web Application Vulnerabilities Reconnaissance phase.
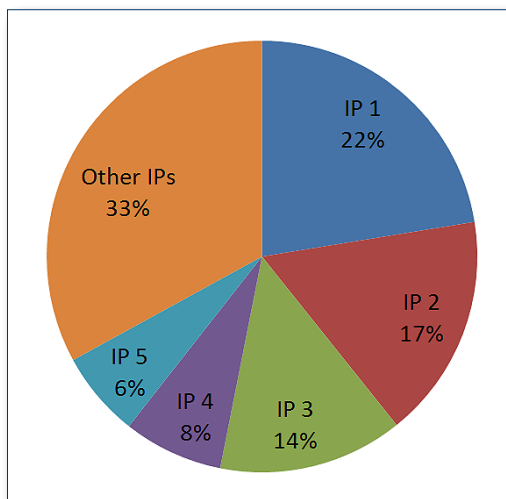
Figure 2: Source IPs distribution for the XSS, SQL injection, and Directory Traversal attack traffic across 1860 IP addresses.

Figure 2 shows the distribution of IPs that produced the XSS, SQL injection and Directory Traversal traffic on the reconnaissance phase. The Figure shows only the percentage of traffic from unique IP addresses; the actual IP addresses are not listed for anonymity.
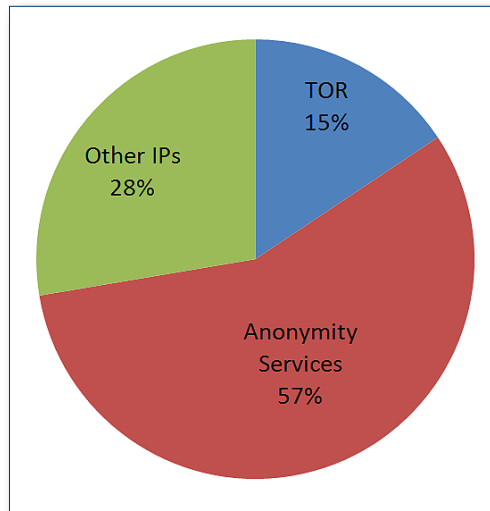
11 http://www.cirt.net/nikto2

DataBaseFileWeb

Figure 3 – Source IPs anonymity analysis that produced XSS, SQL injection and DT attack traffic

Figure 3 shows the identity of the IPs mentioned in Figure 2. Our analysis showed that the vast majority (at least 75%) of the attack traffic was produced by users who used means of anonymity. This specific group of attackers is obviously more familiar with security and privacy issues.

DDoS reconnaissance (Day 22)

The DDoS reconnaissance was on the 22nd day of the attack. Most reconnaissance traffic was related through that TOR, identify obfuscation network.

Below is an example:

» We suspect that the hackers had profiled the site in order to find resources that will require the server to work harder than others – expressed by slower time to respond. However, we didn't find evidence for such activity. Another option is that they had selected the search URL with no previous profiling, as a target as it is known to involve fetching data from database which is a relatively resource intensive action. Repeatedly searching the site may lead to the exhaustion of server resources and create denial of service. The DDoS reconnaissance phase contained a large volume of requests to the search functionality; all of them contain the 'test' word. As this URL was later used to the DDoS attack itself, it can be deduced that those requests were used in order to test the system resources.

» The search reconnaissance sources were two TOR IPs.

DatabaseFileWeb

DDoS Attack (Day 24-25)

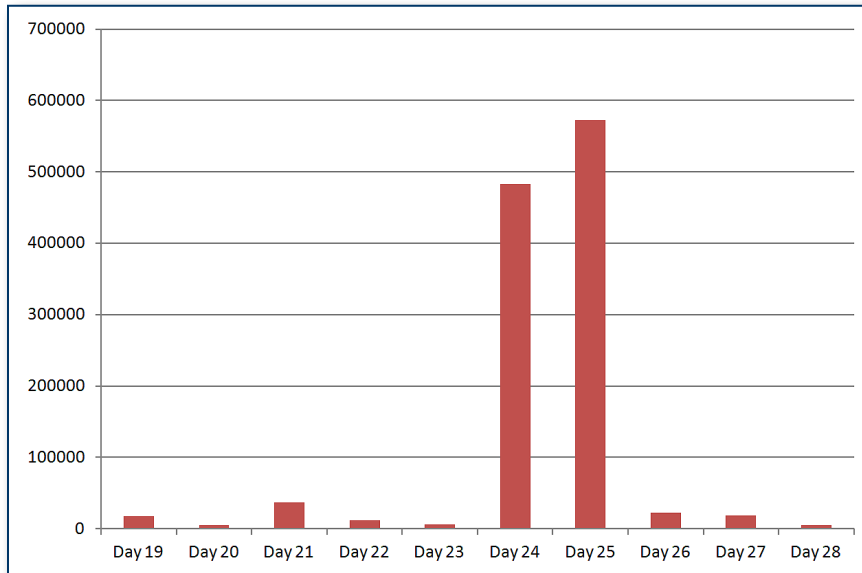Following the reconnaissance phase, the real DDoS attack lasted two days.

Figure 4 – Number attacks per day

Figure 5 shows the volume of attack traffic in each day. The actual attack can be seen clearly with a significantly higher number of attacks in the hundreds of thousands.

Attack Traffic Characteristics

All of the attack's requests were of the following type:

When examining this example, a few details should be noted:

» It is a request to the search URL – the same one as was tested on the Reconnaissance phase.

» The value to search (search word parameter in the request) was carefully chosen in order to massively overload the server and exceed the server's resources faster than an un-related keyword.

» Again, the hacktivists are loud, and they mark their attack with a distinctive message that was redacted in the picture above.

» The Referrer[12] header is constant throughout the attack and reveals interesting information on the attack source, a specific URL that was also redacted.

» The id parameter was incremented in every request to create a different request each time, so the response wouldn't be delivered from cache (such as Akamai's CDN cache, or the attacker browser cache).

---

[12] Although headers (Referrer included) can be forged by the attackers, we had found evidence that the attack was indeed originated from that host.

ⓘMPERVA®

Attack Tool Analysis

By analyzing the format of attack messages and comparing it to code we found on the Web, we believe that we were able to identify the tool used for generating attack traffic.

In previous hacktivist attacks, hackers used the LOIC tool. The LOIC tool was coded in the C# language and required installation of the tool on the attacker machine. Since installation is required, not all platforms could run the LOIC application.

In this attack, other methods and tools were used. The hacktivist group created a Web page that contains a JavaScript. The script iterates endlessly (as long as the page is open in the browser) and generates a new image attribute. The source of the image is the victim's Web page, thereby creating multiple requests to the victims' Website as the browser renders the Web page.

All it takes for an attacker to participate in the attack is to browse to the specific Web page and leave the page open. No need to install or download any software. This is what makes this technique so simple to use, as opposed to other methods.

Since the code is written in JavaScript, it enables any device equipped with a standard browser to take part in the attack – and indeed we have seen mobile devices participating in the attack.

This is the reason why such tools are often being referred to as "Mobile LOIC."

It contains links to information from the DDoS attack:

» The source code refers to a specific site (which has since been removed). This site is the source of the attack as established by the Referrer header as noted in the picture above.

» Same URL characteristics for the attack:

1. Fixed part

2. Parameter named "id" that contains current time in milli-seconds

3. Parameter named "msg" that contains a hacktivist slogan

To conclude, the tool generates the same attack we had seen on the wire.

JavaScript Code Structure

The tool JavaScript code creates requests by continuously changing the source of an HTML image with JavaScript code.

Each time the image source is changed the browser need to create a request in order to fetch it.

```
img.setAttribute("src", targetURL + "?id=" + rID + "&msg=" +
  messageNode.value)
```

Where rID is the date:

```
rID =Number(new Date())
```

This is consistent with the traffic we have seen.

As stated before, the purpose of the dynamic changing parameter is to avoid a state where the response will be in the server from some cache and to make sure that the attack will be actually received on the attacked server.

Finally, we note that the DDoS tool itself doesn't provide anonymity; one will have anonymity only if his browser will use specific tools for it – such as working with TOR or Proxy.

***Testing the Tool in "Captive" Environment***
In order to assess the attack tool capabilities we had set it to attack a server in our lab. The tool user interface is depicted below (NOTE: This is a screenshot of a generic tool used in another hacktivist attack and is not the tool used in the particular incident we recorded):

During the attack, the attack counters are incremented:
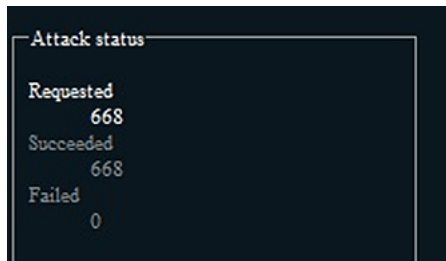
On a regular PC, the attack tool was able to create up to 200 requests per second.
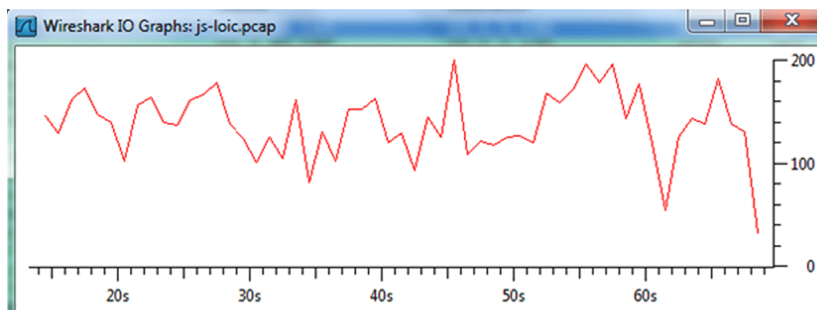
Sources Analysis

Following are two graphs of the distribution over time for DDoS alerts, each one for each day of the attack. One bar represents a time frame of half an hour.
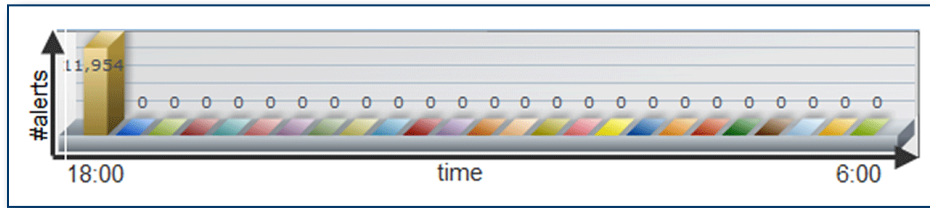

Figure 5 - Distribution over time for the DDoS alerts - day 9, total alerts: 11,954

Figure 6: Distribution over time for the DDoS alerts, day 10, total alerts: 319,792

As can be seen in the graph, there were two peaks to the attack – the first one was between 3:30 am to 4:30 am and the second one was between 6:30 am to 9:30 am, local time.

So, even while being under attack, there's an opportunity for security team to assess the attack details and, if needed, fine tune the rules in order to detect and block the entire attack more precisely.
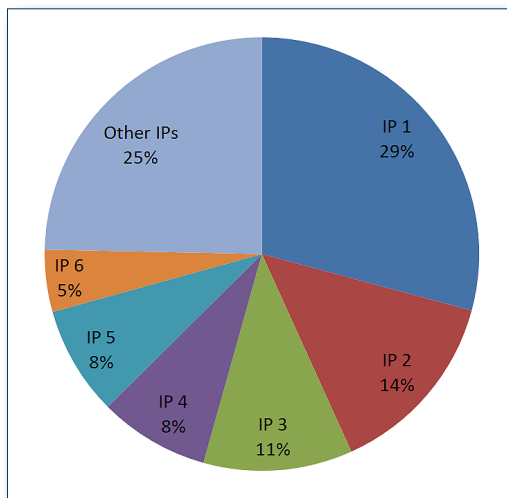

Figure 7: Number of DDoS alerts per IP

Figure 6 shows the distribution of the DDoS alerts produced by different IPs. Once more – the actual IPs are not listed for anonymity. As can be seen – the DDoS attack had originated mainly from a subset of six different IPs that produced most of the traffic. Note that the maximum DDoS alerts were 22 transactions/sec with a bandwidth of 124.5 kilo bits/sec.

Following is a table that shows the geographic origin for each IP that is represented as a percentile in Figure 6. The table focuses on the IPs that produced most of the traffic:

Percentile	Geo-Location/Proxy
29%	Country 1
14%	Anonymous Proxy
11%	Country 2
8%	Country 3
8%	Country 4
5%	Country 3

Table 1 - Percentile of traffic produced by a single IP and its Geo-location

As can be seen in the table above, most of the traffic originated from four different countries. That suggests that these IPs indeed represent individual attackers that don't bother with disguising their identity. Only a small portion of the traffic was delivered through an anonymous proxy.

It's important to note few thousands of attack requests originated from mobile sources, as established by their user agent's header.

Conclusions/ Detection and Mitigation

» **Monitor social media** – Hacktivism is loud by definition. Hacktivists use all of the channels the Web offers – Twitter, Facebook, YouTube, blogspot, pastebin etc. One should proactively scan the Web for hints of coming attacks (Google alerts, for example). The data obtained should be used to accommodate the attack as the data disclosed specifies attack date, means, etc.

» **Protect applications** – Exposing data transacted by applications can have a damaging impact. A strong application security program consisting of Web application firewalls, vulnerability assessments and code reviews can help mitigate the risk of a breach.

» **DDoS is the hacker's last resort** – Attackers prefer small scale, effective campaigns that do not require massive recruitment of willing participants. Therefore, possible attack victims should make it their priority to mitigate application vulnerabilities, even before mitigating DDoS attacks.

» **Analyze the alert messages generated by your security devices** – the DDoS attack was preceded by a few-days-long phase of reconnaissance. By examining these alerts, one can strengthen the security policy and be better prepared for the attack. Daily analysis of alert information may help better prepare for tomorrow's attack.

» **IP reputation is very valuable** – IP reputation is a very powerful tool, especially in high-volume attacks. Using IP reputation, most of the reconnaissance traffic could have been blocked. However, like any PoW they should be interrogated – scrutinizing the content may yield important insights on the purpose of the attackers.