Hacker Intelligence Initiative, Monthly Trend Report #16

Get What You Give: The Value of Shared Threat Intelligence

1. Executive Summary

There is a lot of hype and discussion around highly-specialized, targeted attacks, but this ignores large-scale attack campaigns – a class of attacks that does widespread damage. For example, on April 3, 2013, CNBC reported that, "Major U.S. bank websites have been offline a total of 249 hours in the past six weeks." This is what can occur when attack campaigns target an industry versus a specific target. This report analyzes the potential that community-based threat intelligence has in effectively defending against such large scale campaigns.

Imperva's ADC analyzed real-world traffic from sixty Web applications in order to identify attack patterns. The report demonstrates that, across a community of Web applications, early identification of attack sources and attack payloads can significantly improve the effectiveness of application security. Furthermore, it reduces the cost of decision making with respect to attack traffic across the community. Here's how, based on the traffic analyzed by the ADC:

- **Sources that attack multiple targets in the community are responsible for a disproportionate amount of attack traffic when compared to their share of the population**.
  - Multiple target SQL attackers generated nearly 6x their share of the population.
  - Multiple target comment spam attackers generated 4.3x their share of the population.
  - Multiple target RFI attackers generated 1.7x their share of the population (this amounted to 73% of total attacks).
- **Attack payloads which attack multiple targets within the community also are responsible for a disproportionate amount of attack traffic**.
  - Multiple attacking RFI URLs generated 2.5x their share of the attack population (this accounted for over three quarters of all RFI attacks or 78%).
- Multiple attacking sources and payloads gradually cover more and more targets thus affecting larger parts of the community.

These findings indicate that identifying a "noisy" attack source – an attacker, payload or tool that repeatedly attacks – is of high importance. Recognizing one "noisy" attack source can aid other organization in their defenses.

How can organizations detect these active "noisy" attack sources? Through the power of a community defense, each participant shares its knowledge of attack data to collectively identify common attack patterns to the benefit of all participants.

A community defense can be a three step process, where all steps are performed in parallel:

1. Continuously, and in real-time, collect attack characteristics from multiple targets.
2. Identify common attack patterns from aggregated attack data.
3. Rapidly disseminate actionable defensive items back to the community.

This report presents the value of a community defense, provides a detailed analysis of our data and offers recommendations to organizations. We also suggest ways in which the government and industry can work together to protect against the increasing number of cyber-attacks.

2. Methodology

The data for this report was collected through the real-time monitoring of attack data against more than 60 Web applications.[1] During the first quarter of 2013 (January-March), we extracted the more common Web application attacks targeting these applications. We then analyzed the behavior of these attacks over time and across targets. Early on in our analysis, it was obvious that three key attack characteristics were repeatedly identified across different attack campaigns and against separate targets:

1. **Attack sources**. IPs identified as having generated attack traffic.[2]
2. **Attack payloads**. Specific attack fingerprints.
3. **Attack tools**. Attacks exhibiting automated behavioral patterns.

Using our special-purpose software and knowledge base, we cross-referenced these characteristics against known attack signatures and made comparisons to blacklists of malicious hosts. We also performed calculations of statistical properties of the malicious traffic.

3. Common Web Attacks: Definitions

We examined four common Web attacks, specifically:

SQL Injection (SQLi) - an attack that exploits a security vulnerability occurring in the database layer of an application (like queries). Using SQL injection, the attacker can extract or manipulate the Web application's data. The attack is viable when user input is either incorrectly filtered for string-literal escape characters embedded in SQL statements, or user input is not strongly typed, and thereby unexpectedly executed.

Remote File Inclusion (RFI) - an attack that allows an attacker to include a remote file, usually through a script, on the Web server. This attack can lead to data theft or manipulation, malicious code execution on the Web server, or malicious code execution on the application's client side (such as JavaScript, which can lead to other attacks). This vulnerability occurs due to the employment of user-supplied input, without proper validation.

Local File Inclusion (LFI) - an attack that includes files on a server into the Web server. This attack can lead to malicious code execution on the Web server. The vulnerability occurs when a page included is not properly sanitized, and allows, for example, directory traversal characters to be injected. LFI attacks often append a Null character to the included file path in order to bypass value sanitization.

Comment Spam - a way to manipulate the ranking of the spammer's website within search results returned by popular search engines. A high ranking increases the number of potential visitors and paying customers to this site. The attack targets Web applications that let visitors submit content that contains hyperlinks. The attacker automatically posts random comments or promotions of commercial services to publicly accessible online forums, which contain links to the promoted site.

4. Extracting Actionable Intelligence

We demonstrate that multiple sources are targeted by attacks that exhibit at least one of these repetitive characteristics: attack source, attack payload or attack tool.

4.1 Attack Sources

To illustrate the exact relationship between the number of attacked targets per attack source and the duration of the attacker's activity, we designed an "Attack-Source Reputation Quadrant" graph. (Figures 1 & 3)

In an "Attack-Source Reputation Quadrant" graph, the Y-axis represents the number of Web applications that were attacked, and the X-axis represents the duration of an attack. Accordingly, each dot in the graph represents an attack source and corresponds to the source's longevity and the number of Web applications it has attacked during the course of our analysis.

[1] To protect the anonymity of the applications yet, maintain their identity for analysis, the applications were numbered.

[2] We recognize the concern for identifying an attacker based on the source IP of the malicious request. However, we find this a fitting definition. First, for the sake of simplicity. Second, IPs may not necessarily signify the exact attack entity but instead, a group of attackers, possibly those renting the same botnet.

To express the "Attack-Source Reputation Quadrant" as a graph we added two more divisions. The first is a vertical line along the Y-axis that separates attack sources of those active only during a single day or less, from those active for more than a single day. The second is a horizontal line which similarly isolates attack sources that attacked only a single target from those that attacked multiple targets.

There are four different quadrants:

- The upper left quadrant (in pink) includes all attack sources that were active for only one day and attacked more than one target.
- The lower left corner (in green) includes all attack sources that were active for only one day and attacked only a single target.
- The upper right quadrant (in blue) includes all attack sources that were active for more than one day and attacked more than one target.
- The lower right quadrant (in orange) includes all attack sources that were active for more than one day and attacked only a single target.

To quantify the data, we've enhanced the "Attack-Source Reputation Quadrant" with two pie-charts (color-coded to the quadrants, respectively):

- The top pie chart represents the number of attack sources within each quadrant.
- The bottom pie chart represents the number of attacks each attack source generated.

We conducted this analysis separately for SQL Injection and for Comment Spam:

- **SQL Injection**
  The "Attack-Source Reputation Quadrant" for SQL Injection is displayed in Figure 1. As shown, 3% of attack sources generating SQLi targeted multiple sources for more than a day, although they account for 17% of the SQLi traffic. In other words, the same attack sources generated attack traffic at a rate nearly six times their share in the source population. Accordingly, a participant sharing information on their attackers within a community can help all members of the community in blocking a large portion of attack traffic targeting their applications.

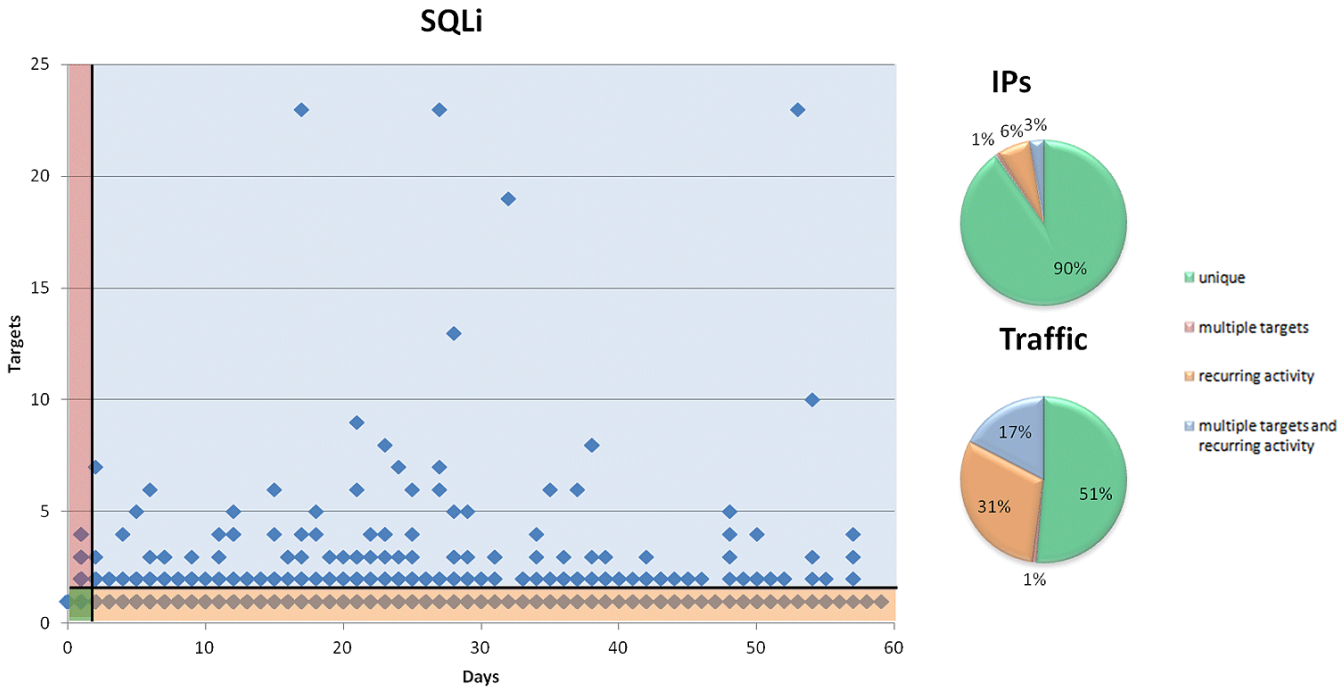
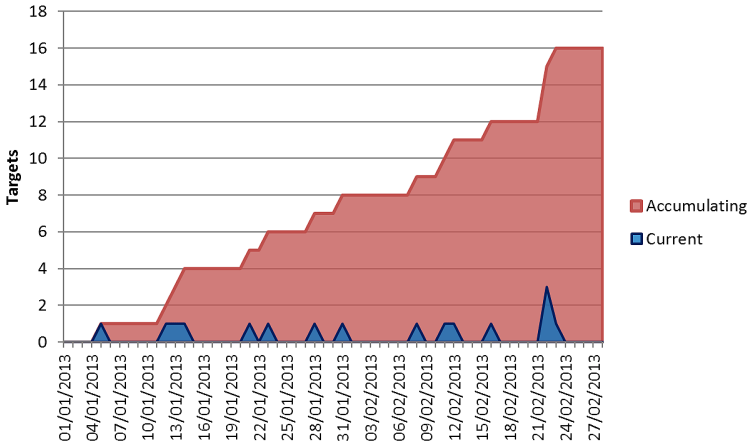


Figure 1: Attack Source Reputation Quadrant for SQL Injection

- **Accumulated effect**
  Zooming to one of the points in the upper right (blue) quadrant, in Figure 1 highlights how, over time, SQLi attack activity accumulates to a high number of attacks against a large number of targets. However, a single target observing only its own attack data witnesses only a small portion of overall SQLi attack activity.

  The blue area in Figure 2 represents the number of targets that were attacked by the same attack source. Normally, this attack source attacked only one target at a time (the exception is on February 22nd, when this attack source attacked three separate targets on the same day). The red area represents the accumulated number of targets attacked by this same source. It was possible to plot this area only through the collective analysis of attack data against multiple targets.

  It is important to note that from the date in Figure 2 we can easily conclude that rapidly disseminating the information of a "noisy" source to the rest of the members in the community would block the attacks. By January 13th, there is enough evidence in the community to flag this source as malicious, and thus easily mitigate any attack by that source against the other 12 applications.

› **Comment Spam**

The "Attack-Source Reputation Quadrant" for comment spam is displayed in Figure 3. The upper pie chart shows that only 13% of the attack sources fall within the blue quadrant. In other words, only 13% of the attack sources attacked more than one target and were active for longer than a day. However, these attack sources generated 56% of all comment spam traffic – more than four times attack traffic than their share in the population.

This finding shows that by analyzing attack patterns across a multitude of targets it is possible to detect a relatively small amount of attack sources and consequently, protect against a large portion of the malicious traffic.

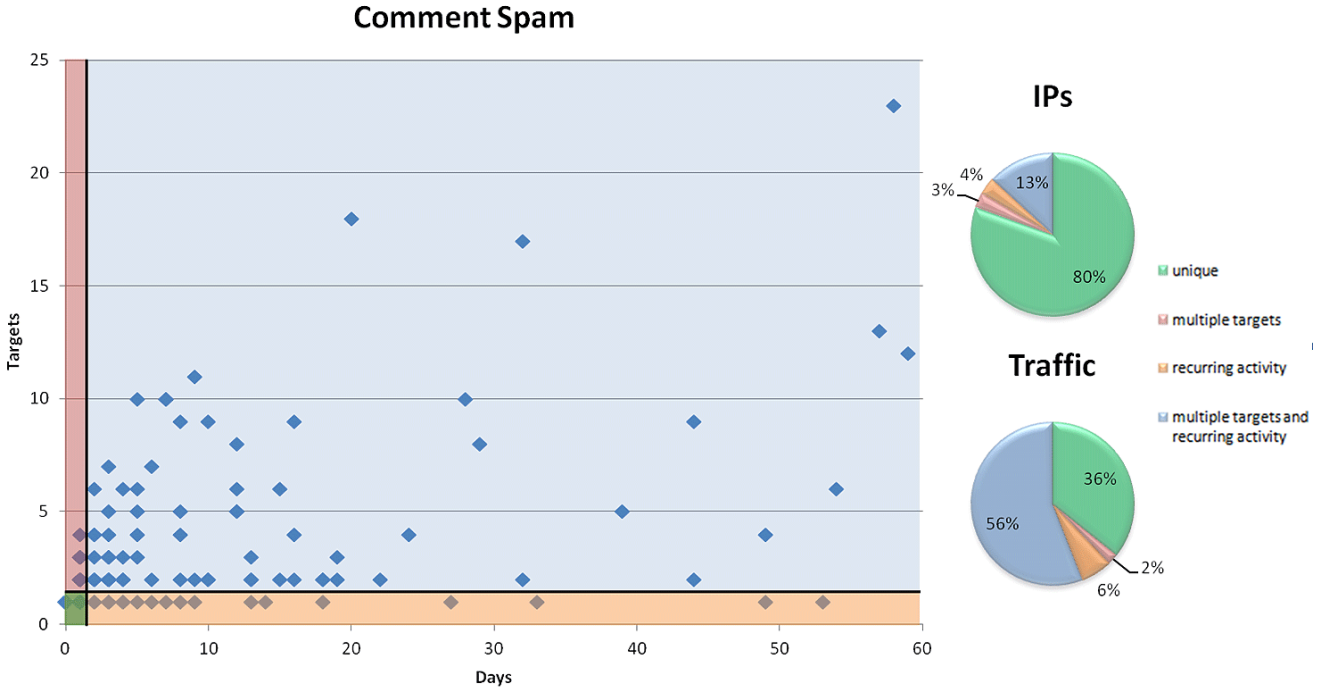


Figure 3: Attack Source Reputation Quadrant for Comment Spam

4.2 Attack Payload

The attack payload usually contains particular fingerprints. Once these fingerprints are tagged as malicious, any future request containing that payload may automatically be flagged as an attack.

Similarly to our "Attack-Source Reputation Quadrant" the "Attack-Payload Reputation Quadrant" in Figure 4 displays RFI attack campaigns according to their included URLs. The vertical black line separates distinct URLs that appeared for only a single day of attacks, from those used over multiple days. The horizontal line separates distinct URLs sent against only a single target, from those sent against multiple targets.

Accordingly, the upper right quadrant of Figure 4 (in blue) includes the number of distinct URLs that were active for more than one day, and sent against more than one target. The "Attack-Payload Reputation Quadrant" for the RFI payload illustrates that 31% of the URLs were repeated over multiple days and attacked multiple targets (blue quadrant). However, these URLs were included in 78% of RFI traffic.

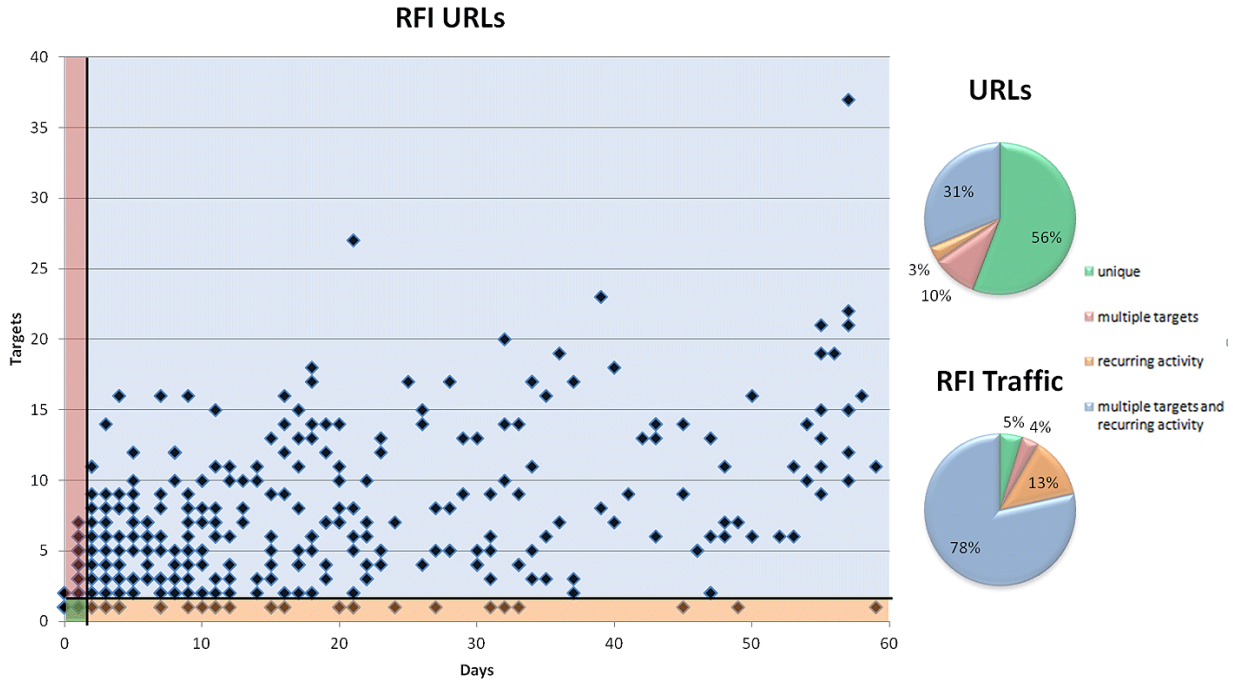


Figure 4: Attack Payload Reputation Quadrant for RFI

4.2.1 Case Study

We present the following attack campaign observed during Q1 2013, to demonstrate the behavior of different URLs included within RFI attacks yet used by a single attack source and attacking multiple targets.

In this attack campaign, we witnessed a Brazilian attack source generating both RFI and SQLi attack traffic against multiple applications. On closer inspection, we found that this source was actually a server belonging to a Brazilian medical research facility which was compromised by attackers, injected with malicious code, and converted into an attack machine under the hacker's botnet.

Most of the server's requests had an innocent looking User-Agent field. However, a few of the requests included a very unique string: **Mozilla/5.0 (compatible; indonesiancoder/1.0; +http://xxxxxxxxxxxxxxx.com)**. As emphasized in previous HII reports, the User-Agent field is an important identifier in detecting automation and the usage of hacking tools[3].

We found out that http://xxxxxxxxxxxxxxx.com is a website belonging to an Indonesian independent hacker who is related to several Indonesian hacking communities. The site contains hacking tips, news review, and a selection of tutorials and code examples.

[3] http://www.imperva.com/docs/HII_Automation_of_Attacks.pdf

Figure 5 is a network graph representing our analysis of different attack payloads included within RFI attacks sent by the same attack source (where the source is the Indonesian hacker via the compromised Brazilian server).

The graph clearly shows that the attacker used more than one payload to attack a single application. App #1, for example, was attacked using no less than nine different payloads. Similarly, payloads within RFI attacks were re-used to attack multiple targets. As the graph indicates, different URL payloads, URL #1 and URL #2, were included in different RFI requests which targeted four applications in addition to App #1. URL #1 and URL #2, as well as others used by this attacker, refer to the same malicious file, "jahat.php"[4]: http://xxxxxx.com.xxxxx.es/jahat.php, http://xxxxxxxxx.com.lp-host.com/jahat.php. URL #3 refers to a different file, http://xxxxxxxxx.com.xx.xxxxxxx.com/jos.php. Based on the pattern from the graph, it is likely that URL#3 targeted other applications for which we do not have insight.

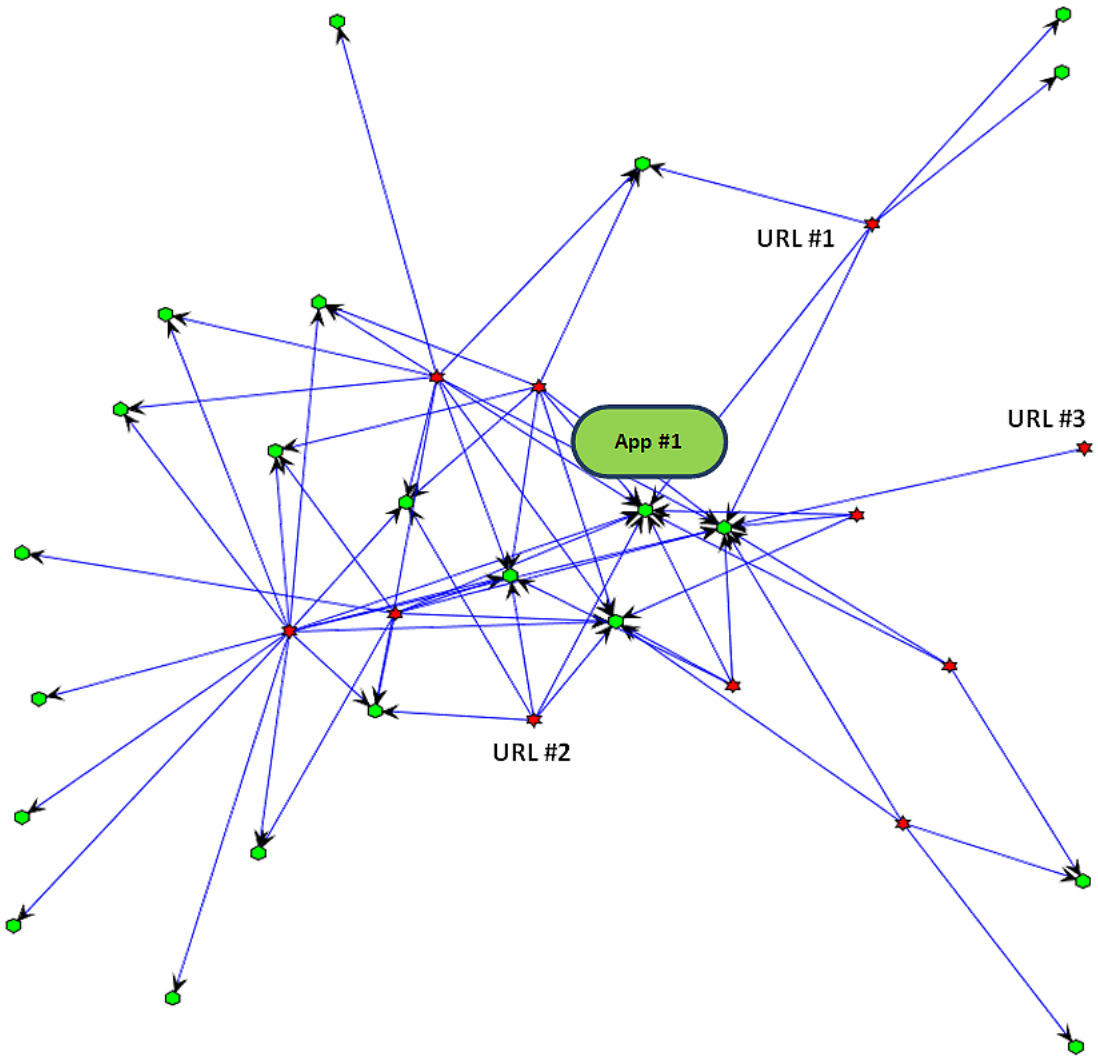


Figure 5: RFI Payload to Target Network Graph

[4] "jahat" translates to "evil" in Indonesian.

It is important to note that payloads directing to malicious code are not only re-used to attack different targets, but can be used by more than one attacker. In the example above, URL#1 was used during February 2013 by twelve different attack sources located in ten different countries – among which were Thailand, Germany, Slovenia and Russia. This might imply a single attacker was using servers from around the globe to execute the attacks.

Consequently, it is clear that payloads provide important and valuable attack identifiers. Since a single payload can be used against multiple targets and by multiple attackers, it is of great importance to share this kind of information to mitigate future attacks.

4.3 Attack Tools

Hackers are increasingly using automated tools to carry out their attacks more efficiently on Web applications. Once an automated attack is under way, different components of the attack may signify an ongoing campaign, or the hacker's field preparation for an upcoming attack.

The histogram in Figure 6 demonstrates the potential value of detecting and fingerprinting attack tools. In our last Web Application Attack Report[5], we clustered all attacks that exceeded the rate of thirty requests per five minutes, and named these clusters "attack incidents". Our conclusion about these incidents is that, based on the frequency and speed of attacks, they are likely automated and generated by specific hacking tools. Our WAAR report concluded that for common Web application attacks (RFI, LFI, SQL Injection, Comment Spam, XSS and Directory Traversal), more than half of the malicious traffic is generated by automated tools. In Figure 6, at least 81% of all SQL Injection attacks were automated.

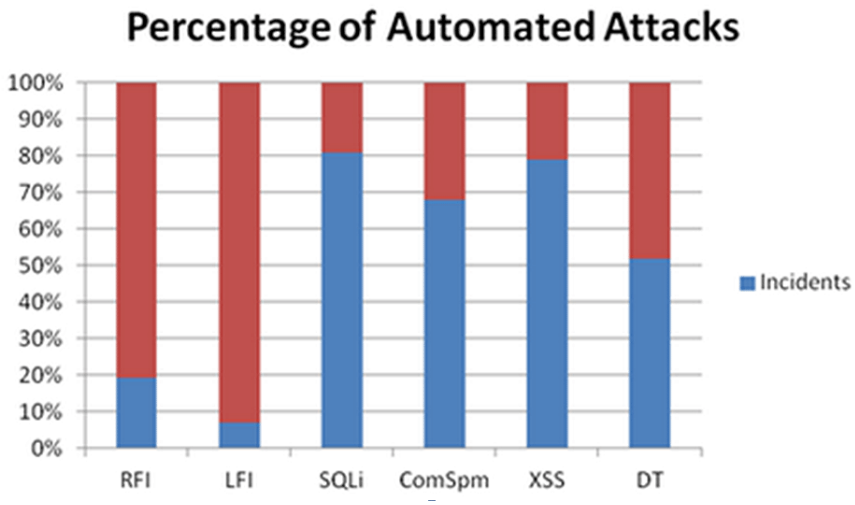


Figure 6: Proportion of Automated Attacks of all Malicious Traffic

Taking this type of analysis further, tools which behave in an automated manner may signify a reconnaissance attack where the hacker is testing a vulnerable application. Recognition of a reconnaissance attack allows quick identification of similar vectors targeting other applications and allows for blacklisting of the suspicious source IPs – *before they actually start to attack*[6].

[5] http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed3.pdf

[6] Updating the blacklist may also occur when these IPs start their actual attacks against their first group of targets, and before they start to scan the next batch of potential targets.

In an attack we witnessed in early March 2013, the usage of attack tools alerted us to a sudden epidemic of RFI reconnaissance attacks. On the face of it, the requests weren't doing any damage and would have been able to fly under the attack-detection radar. However, the high volume of requests within such a short time frame indicated that a reconnaissance attack was under way and the attacker was testing vulnerable sites by simply referencing a file on Google's server (namely, "humans.txt").

The injected URL (i.e. the "humans.txt" file) contained the following simple non-malicious text:

```
Google is built by a large team of engineers, designers, researchers, robots, and
others in many different sites across the globe. It is updated continuously, and
built with more tools and technologies than we can shake a stick at. If you'd like
to help us out, see google.com/jobs.
```

The attackers attempted to inject this page into eleven different Web applications, with one application being targeted 730 times (through slightly different attack vectors containing the same URL). In total, we saw 5144 requests within a three week period.

Figure 7 is an example of the RFI requests:

```
Occured : Fri Mar 15 04:42:32 PDT 2013
Source :
Country Code : ru,
Ip : 91.227.68.33,
Url : /auth/auth.php
Method : GET
Http Headers :
Host : www.————.com,
Referer : http://www.————.com/,
User-Agent : Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko Firefox/11.0
Http Parameters :
phpbb_root_path : http://google.com/humans.txt
```

Figure 7: Sample RFI Request

Figure 8, shows source attackers vs. target applications that used RFI with this URL. The source attackers are in red, the targets (applications) are in green. Groups of different sources are attacking the same target (for example, App #1), and some sources (for example, Source #2) are attacking multiple targets.

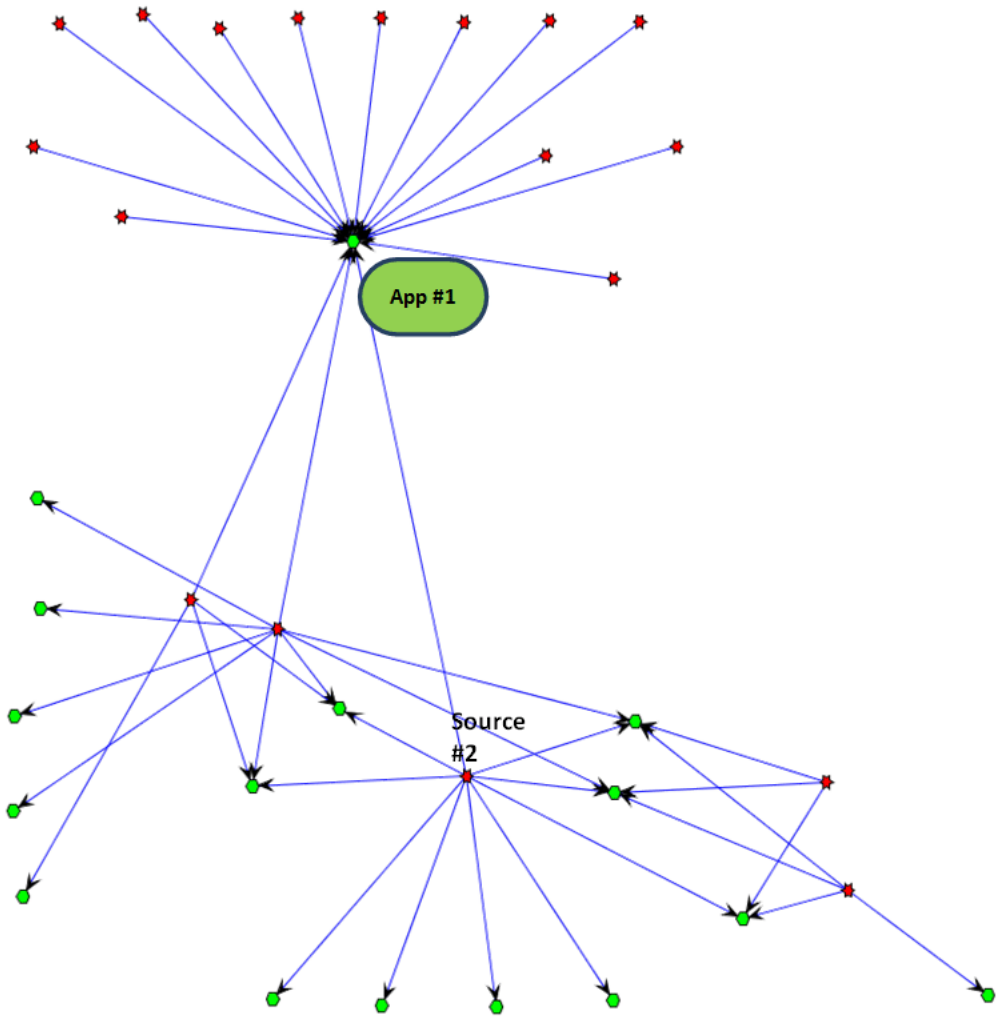


Figure 8: Source Attackers vs. Target Applications Using RFI

A large part of the attack traffic is generated by attack tools. These tools are being reused by their owners for multiple campaigns against multiple targets. Some of these tools are actually sold by their creators, and are being used by different attack groups against different targets. In fact some of these tools have received a good share of media coverage (Havij and Sqlmap for SQL Injection, LOIC and Railgun for DDoS. The most effective way to quickly identify new and updated tool marks is by collecting attack traffic from the community, analyzing it for common patterns, and then sending the resulting identifiers back to community members.). By being able to detect "tool marks" for specific attack tools, application layer security solutions can become more effective.

5. Conclusions and Recommendations

5.1 Closing the Loop

We have previously shown that by analyzing attacks of the same type for a common source we can extract information that would help us deter similar attacks from the same source in the future. We've also shown that if we analyze attacks of the same type for a common payload we can extract information that would help us mitigate similar attacks using that same payload in the future.

However, one of the most exciting possibilities encompassed within this type of collaboration is the value that can be obtained by repetitively applying this collection, analysis, and dissemination process. For example, after analyzing SQL injection attacks for common sources, we apply the results back to the protection devices. Then, we collect all attacks that are generated by the sources that have been identified as threat to the community. These attacks include SQL injection attacks but also RFI attacks (discussed in previous WAAR reports). We can analyze those RFI attacks to extract common payloads, and then apply those payloads back to the protection devices. Once these are applied, we are able to detect RFI attacks from sources we were not aware of earlier (or we were not aware how "noisy" they were). When we analyze these RFI attacks, we may identify a new group of attack sources that we can then feedback to the protection devices, repeating the process in an infinite loop.

5.2 The Role of Government

Whether it is mass attack campaigns on multiple targets, attack tools used repeatedly by attackers, or reconnaissance attacks, the value of information sharing outweighs that of keeping the data within. Our research shows that organizations, on their own, cannot defend their systems as well as organizations that coordinate their efforts.

The government poses another incentive for sharing attack information. Many governments worldwide have begun to recognize their role in protecting organizations under their jurisdiction from cyber-attacks.

On Capitol Hill, discussions addressing information sharing between the government and the private sector are currently underway. In February 2013, President Obama issued an executive order addressing protections from cyber-security threats for critical infrastructure by increasing "the volume, timeliness, and quality of cyber-threat information shared with US private sector entities." While this is a voluntary program for private sector entities, this executive order heralds not only the requirement, but also the government's responsibility, in supporting an information sharing platform.

In March 2013, the British government announced its initiative for cross-sector threat information sharing among the government, industry, and information security analysts.

In the past few years, as governments are formalizing these processes, we've been witnessing a more "informal" collaboration between industry and government. Globally, Computer Emergency Response Teams (CERT, Infagard, and ENISA) attempt to bridge the information sharing divide by providing recommendations based on the members' experience. However, for early information sharing "adopters", the fear of the shared data being leaked, resulting in damages, penalties, bad PR, and personal accountability, remains a strong inhibitor. Governments should recognize that organizations are holding back for self-protection reasons, and work to find ways to encourage more ubiquitous information-sharing.

The legislation that we would like to see would contain the following:

- Endorse information sharing initiatives. Provide a safe, trusted and relevant forum to entice organizations to voluntarily share their information.
- Provide incentives to share information and refrain from penalizing companies for losses due to threat information sharing.
- Enact forward-looking provisions that address a more active role for governments that include formalizing the standards for threat information sharing.

5.3 Recommendations

While surgically targeted web application attacks are still occurring, we also witness large campaigns with common characteristics across different organizations and verticals. Consequently, security cooperation between organizations that suffer from web attacks can create a "network effect" in which all members of the cooperating community can benefit by exchanging security and threat information.

› **We recommend that users actively participate in security intelligence communities when they are available**.

- The value of these communities is directly dependent on the willingness of organizations not solely to use the output of these communities, but to contribute data, as well.
- To overcome the common objection that data sharing may put the organization at risk, ensure that the community provides adequate safeguards and anonymization of data.

› **We recommend that industry and government regulatory bodies take a proactive approach to security intelligence sharing**.

- While enforcing minimum standards is a key function of regulatory bodies, , there is an opportunity to deliver a positive value by providing a safe, trusted, and relevant forum for sharing security intelligence.

Hacker Intelligence Initiative Overview

The Imperva Hacker Intelligence Initiative goes inside the cyber-underground and provides analysis of the trending hacking techniques and interesting attack campaigns from the past month. A part of Imperva's Application Defense Center research arm, the Hacker Intelligence Initiative (HII), is focused on tracking the latest trends in attacks, Web application security and cyber-crime business models with the goal of improving security controls and risk management processes.

Imperva
3400 Bridge Parkway, Suite 200
Redwood City, CA 94065

Tel: +1-650-345-9000
Fax: +1-650-345-9004
www.imperva.com