KASPERSKY LAB REPORT

Java under attack — the evolution of exploits in 2012-2013

October 2013

KASPERSKY lab

TABLE OF CONTENTS

INTRODUCTION:

Why we decided to investigate Java

One of the biggest problems facing the IT security industry is the use of vulnerabilities in legitimate software to launch malware attacks. Malicious programs can use these vulnerabilities to infect a computer without attracting the attention of the user — and, in some cases, security software.

That's why cyber criminals prefer these attacks, known as exploits, over other infection methods. Unlike social engineering, which can be hit or miss, the use of vulnerabilities continues to produce the desired results.

Exploits still pose a threat even when the user is aware of the danger they pose, is well versed in IT security, and dilligently keeps their software updated. They operate surreptitiously, and users can fall victim simply by visiting a site that contains malicious exploit code or by opening a seemingly legitimate file with hidden malicious code. If the version of the software that is used to download a document or elements of a website contains a vulnerability, the exploit is triggered. It then loads additional malware from the criminals' server which, depending on the intended target, performs malicious activity such as stealing personal data, using the computer as part of a botnet to distribute spam or carry out DDoS attacks, etc.

Exploits still pose a threat even when the user knows they exist, is well versed in IT security and keeps track of software updates. That's because when a vulnerability is detected it can take weeks until a patch is released to fix it. During that time exploits are able to function freely and threaten the security of Internet users. That risk can be reduced significantly if users have high-quality security solutions installed on their computers, including technology capable of blocking attacks initiated by exploits.

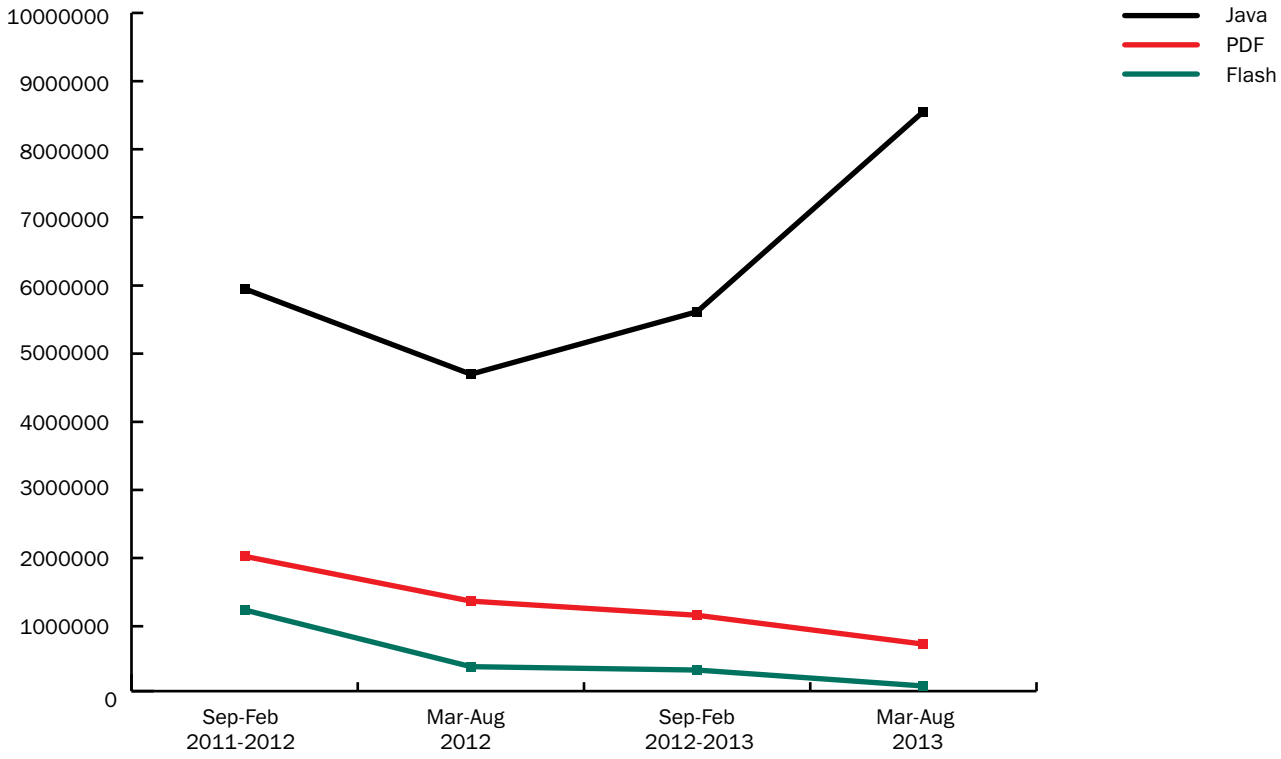
While developing high quality security solutions for home and corporate users, Kaspersky Lab constantly watches cyberthreats landscape. This helps Kaspersky Lab's experts regularly improve protection technologies and strengthen security against the most urgent threats. Along with other means, it includes data analysis, gained from **Kaspersky Security Network** — a cloud infrastructure that unites more than 60 million of Kaspersky Lab's users around the world and provides information, necessary to monitor the situation in field of cyberthreats regularly.

> **Exploits are special programs designed to take advantage of vulnerabilities in popular software used by millions of people around the world.**

The data, gained from Kaspersky Security Network, served as the basis for Kaspersky Lab's earlier report **Evaluating the threat level of software vulnerabilities**, in which the company's experts analysed the situation surrounding the most widely-used legitimate software. It revealed that a huge number of users do not update vulnerable versions of at-risk applications, even weeks after an update was released. The report also identified the most frequent targets for exploit attacks: the Oracle Java software environment, the multimedia content display application Adobe Flash Player, and its sister product Adobe Reader, used to view PDFs.

In our new study we decided to focus on Oracle Java. This choice reflects the huge increase in attacks on this product over the past year, as shown in the table below.

Number of exploit-led attacks in 2011-2013



QUICK Q&A ABOUT JAVA

What is Java?

Java is an object-oriented programming language that makes it relatively quick and easy to create cross-platform multimedia elements, including applications that can run on any virtual Java machine, regardless of the computer architecture. In other words, the developer does not need to rewrite the application every time it encounters a new operating system or browser. The main requirement is an existing version of the virtual Java machine for this operating system or browser. This had made the Java tool extremely popular among developers of websites and software for various devices.

Why are there so many vulnerabilities in Java?

First, the development of Java began when malicious attacks through vulnerabilities were virtually non-existent. As a result, software developers in general — not just those working on Java — could not anticipate this potential security risk, and the software was not built with security in mind.

Secondly, the large number of identified vulnerabilities in Java confirms the large numbers of specialists specifically searching for these vulnerabilities. According to Oracle, Java's owner, the product is used on over 3 billion different devices worldwide. This vast audience is one of the key parameters that guide cyber criminals when they choose a target for their attacks. The more people use a particular product, the greater the chances that the criminals can illegally enrich themselves.

Therefore, Java cannot be considered as the most vulnerable software platform — in fact, criminals are simply aware that millions of people use this product and feel it makes sense to spend resources on finding ways to take advantage of this situation.

What is an exploit pack?

It is an illegal software pack which includes a control panel and a set of exploits designed for a range of legitimate applications. Exploit packs are similar to a set of keys, but here each «key» is a separate exploit which is triggered depending on what software the victim is using. Exploits for Java vulnerabilities can also be used separately from an exploit pack. They are also seen, for example, in targeted attacks where the criminals prepare by detecting flaws in the target's IT infrastructure and create a web page which houses a Java exploit.

How are attacks using Java exploits carried out?

Typically cyber criminals lure users to a specially designed website which hosts a suitable selection of exploits. When the user loads the page, an built-in module identifies which browser is in use and which Java-plugin versions are installed. Using this information an exploit is chosen, and it automatically loads its malware onto the computer.

METHODOLOGY OF THE REPORT:

What and how we calculated

For this study we used information gained from more than 40 million users of Kaspersky Lab's products around the world who consented to provide statistics to Kaspersky Security Network. This data comes from computers containing any version of Oracle Java software. The report includes statistics from four protection modules used in Kaspersky Lab products: web antivirus, file antivirus, heuristic analyzer and the Automatic Exploit Prevention module.

The period under study:

- We chose the data collected within 12 months from September 2012 to August 2013. These 12 months are particularly interesting because of the comparatively large number of vulnerabilities detected in Java – twice as many as in the preceding 12-month period. To highlight this change we split this period into two halves: September 2012 to February 2013 and March to August 2013.

The subject of the study:

- The number of vulnerabilities in Java and their nature;
- The number of Java exploit attacks and their dynamics;
- The number of unique users under attack and their dynamics;
- The distribution of attacks and unique users by geographical location;
- The number and prevalence of 'sophisticated' attacks, detected using Kaspersky Lab's unique **Automatic Exploit Prevention**.
- The extent of the distribution of exploits developed for the vulerabilities detected within the study period.

MAIN FINDINGS

The study showed that Oracle and its users had a very difficult 12 months, facing a huge number of attacks.

- More than 160 vulnerabilities, six of them critical, were detected in this software by different companies and cybersecurity experts;
- Kaspersky Lab detected over 14.1 million attacks which used Java exploits. That's 33.3% more than the previous 12 months.
- The number of attacks increased throughout the year. From March to August 2013 over 8.54 million attacks were detected, up 52.7% on the previous six months.
- In total more than 3.57 million users of Kaspersky Lab's solutions were attacked all over the world during the 12-month study period.
- In the second half of the year the number of attacked users was up 21% compared to the first half and came to 2 million unique users.
- About 80% of attacks were concentrated in just 10 countries. Most of them were commited in the USA, Russia, Germany and Italy.
- Canada, the USA, Germany and Brazil were the countries with the fastest growing numbers of attacks. Along with France these countries also have the fastest growing numbers of unique users under attack.
- Near 50% of attacks used just six groups of exploits.
- Over the last 12 months Kaspersky Lab's Automatic Exploit Prevention blocked 4.2 million 'sophisticated' attacks, targeted at more than 2 million users.

Of course these trends have their own pecularities. We will study them in greater detail.

PART 1:

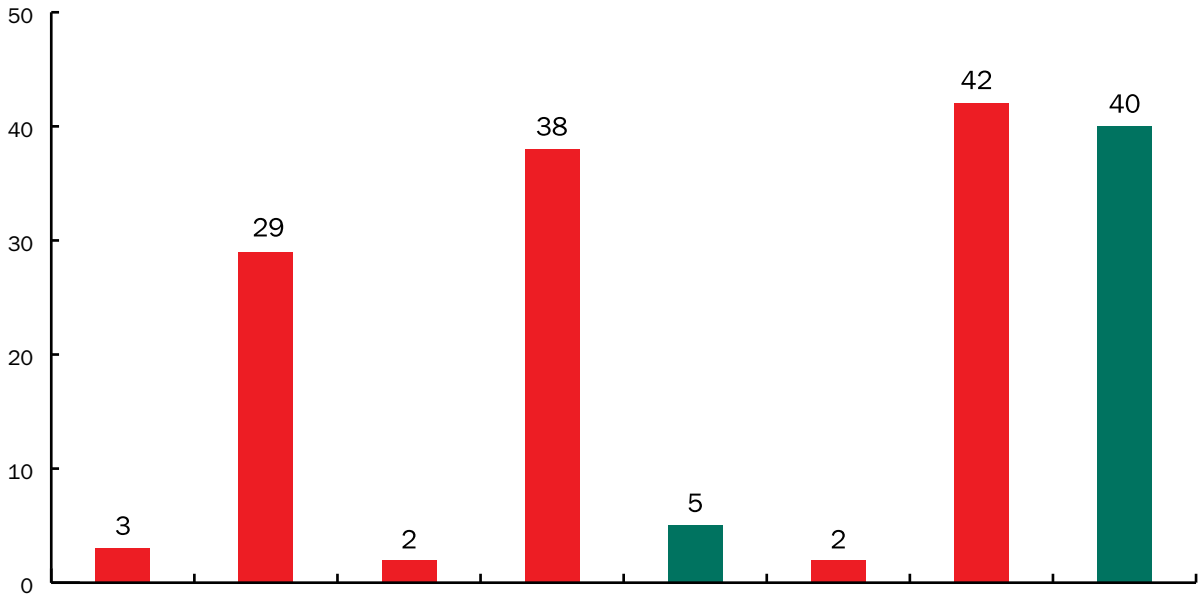
A year of vulnerabilities in Java

The period addressed in this report turned out to be particularly challenging for Oracle's Java in terms of unpatched vulnerabilities. During this time, 161 vulnerabilities were detected in various versions of Java. Most of these were in versions 1.5, 1.6, and 1.7, which are the most prevalent versions of the software. For comparison: over the same period in 2011-2012, just 51 unpatched vulnerabilities were reported.

The source data used to compile the chart below came from the Danish firm Secunia, which aggregates data about vulnerabilities detected in legitimate software. As can be seen in the chart, Secunia issued eight Advisories during this period (each announcing the detection of Java vulnerabilities), and some of them dealt with as many as 40 new reported vulnerabilities. During the same period in 2011-2012, five Secunia Advisories were issued.

Number of Java vulnerabilities in 2012-2013

The red bars denote the release of one or more Secunia Advisories announcing the detection of critical vulnerabilities. Only two advisories (green bars) contained no critical vulnerabilities. Source: Secunia



It's worth pointing out that the overwhelming majority of the vulnerabilities found in Java do not pose any great threat. Yet at the same time, six of these Advisories addressed at least one vulnerability that could potentially lead to computer infection. In total, only two Advisories published between September 2012 and August 2013 did not report critical vulnerabilities — the reports issues on February 20 and June 19. During that period Kaspersky Lab highlighted the six most dangerous Java vulnerabilities posing and "taught" its antivirus technologies to respond to the six exploit families targeting these vulnerabilities.

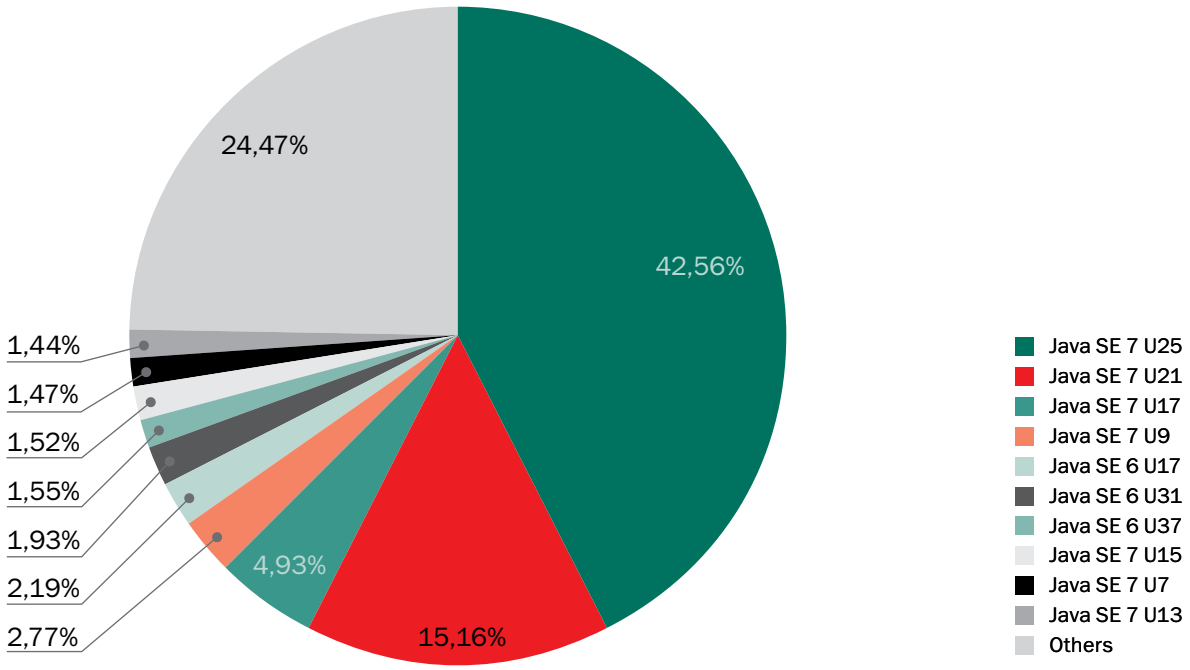
To Oracle's credit, all of the critical vulnerabilities detected over this period were already patched by the time this report was written. The latest version of Java (1.7) was released in June this year (Update 25), when version 1.6 (Update 51) was most prevalent.

Unfortunately, Kaspersky Lab statistics — which help us gain a fuller picture of which versions of Java people are using on computers running Kaspersky Lab security products — do not confirm that all users are protected against exploit attacks once an update is released.

One and a half months after the release of the latest version of Java, most people are still using vulnerable versions of the software.

Top 10 versions of Java, August 2013

This pie chart was compiled using data from 26.82 million individual users of Kaspersky Security Network reporting the use of any version of Java on their personal computers. Source (here and below): Kaspersky Security Network.



Less than half (42.5%) of all Java users in Kaspersky Security Network installed the update for the newest Java version. Over 15% (or more than 4 million users) are using the previous SE7 U21 version, which was released in mid-April. Roughly 1.3 million users (4.93%) are still using SE7 U17, which was released back in March 2013.

Remarkably, SE 6 U37 — released back in October 2012 — was the most recent version of Java 1.6 in the Top 10 most commonly used versions.

The conclusions are obvious: one and a half months after the release of the latest version of Java, most users are still working with vulnerable versions.

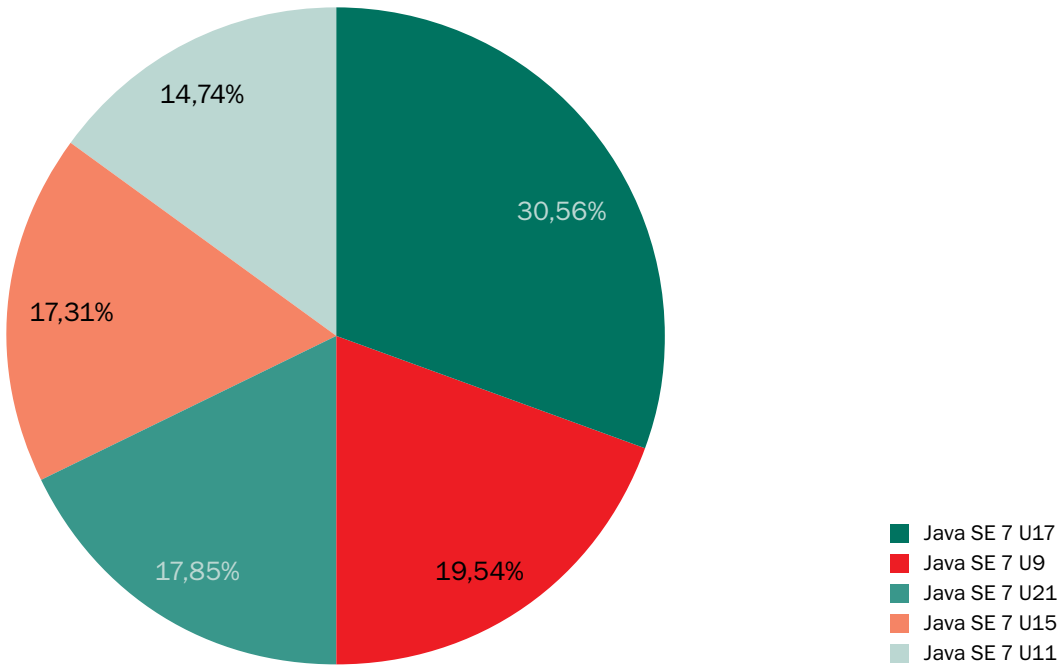
This situation happens again and again. In late June, less than two weeks after Update 25 was released, just 291,000 users reported that they were using the most up-to-date version. There was little time for many users to update their software, but more worryingly by the end of June we still saw almost twice as many people using the vulnerable outdated version U17 as the more recent U21 version released in mid-April. The number of people running U21 was 3.5 million and number of people still using the outdated and vulnerable SE U17 was more than 6 million

An analysis of the preceding period reveals the same situation. Most users are working with Java versions that are 2-3 generations older than the most up-to-date version at the time of the research.

At the same time, while this trend is dangerous, it also demonstrates some positive signs. If we compare the five most widely-used versions of Java in August with the same data from the end of June, then we can see that with a similar number of individual users (18.65 million in August, and 19.7 million in June), the number of users running the most up-to-date version of Java in August was significantly higher than in June.

The chart below illustrates the percentages of users running the Top 5 versions of Java in June.

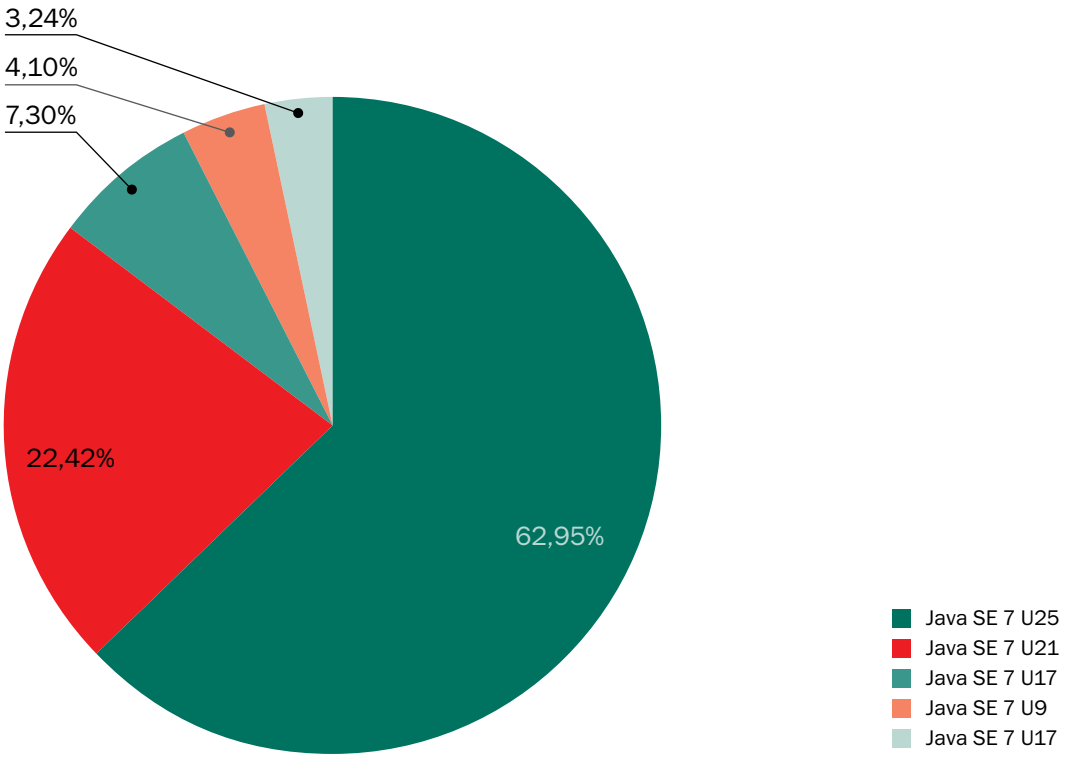
Top 5 Java versions, June 2013



Since there was a relatively short period of time (less than two weeks) from the release of the latest version (U25) and the collection of data for this report, the "latest" version used for the purposes of this analysis was U21. As the chart clearly shows, only 17.85% of users were running that version of Java, while most people (nearly one-third) were using the outmoded and vulnerable U17.

In August, we observed a much more encouraging trend.

Top 5 Java versions, August 2013



In other words, during the summer, Java users appeared much more willing to update to the latest version of the software than in the spring. It's difficult to pinpoint the exact reasons behind this acceleration in the update process. It's possible that mass media coverage of the detection of Java vulnerabilities prompted people to act — there were many stories of this type in the news throughout the spring of 2013. As will become clear from our further analysis, this past spring turned out to be the most challenging period for Java, particularly if you consider the number of attacks and the number of targeted users.

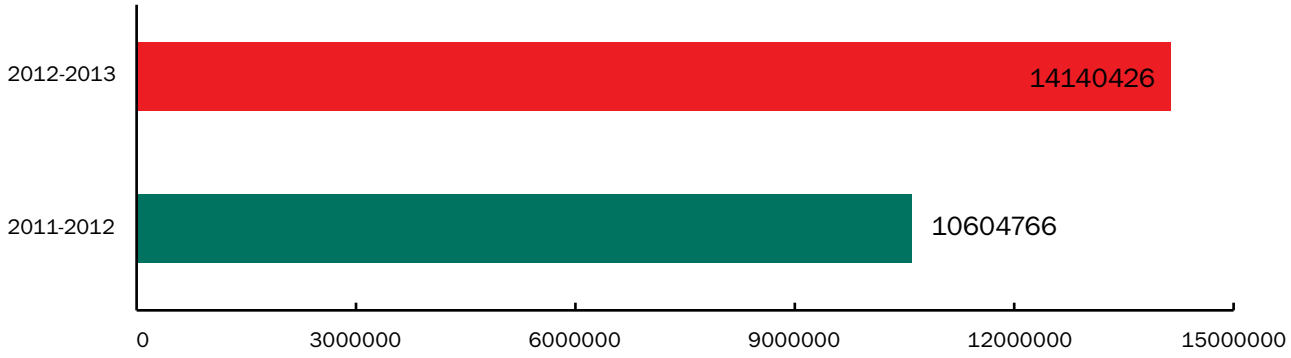
PART 2:

A year of attacks on Java users

A steady rise in attacks

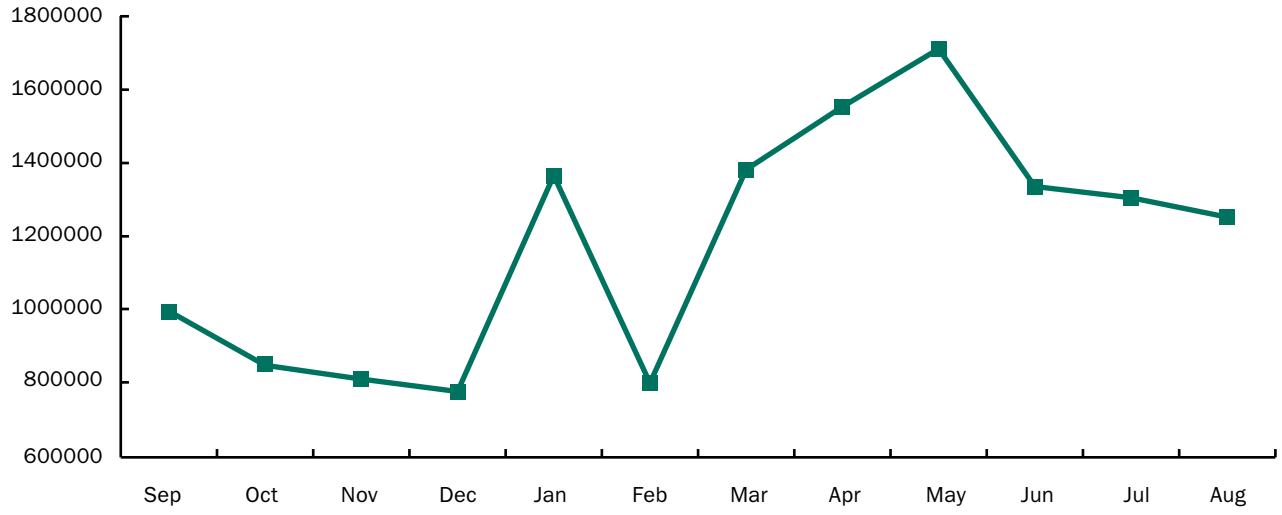
Over the course of 12 months from September 2012 through August 2013, Kaspersky Lab recorded over 14.1 million attacks launched against users around the world. Compared to the same period in 2011-2012, the number of attacks rose 33.3%.

Number of attacks: Y-on-Y comparison



At the same time, if we break that 12-month span into two six-month periods, the dynamics are even more striking. There was a 52.7% increase in the number of attacks in the second half-year – 8.54 million from March-August 2013, compared with 5.59 million in the previous six months. The dynamics of attacks over the previous 12 months:

Attacks, 2012-2013



At the start of this period, we saw stable numbers of attacks over the autumn months, and a week downward trend in December. Then in the beginning of January we saw a sharp surge, peaking in the middle of the month and followed by a similarly dramatic fall.

> **Compared to the same period in 2011-2012, the number of attacks in 2012-2013 increased 33.3%.**

Starting in February and through the end of May, the number of attacks grew rapidly. This was a result of, among other things, the detection of all of new vulnerabilities in Java (87 vulnerabilities in February-May, three of which were ranked as critical) and the slow transition among Java users to patched, more secure versions.

From June through August, we observed a steady decrease in the number of attacks. In mid-June Oracle released a new version of Java which was greeted with a more active transition to the updated version, as described in the preceding section of this report. The summer vacation season is traditionally slow in terms of cybercriminal activity, which is another factor to consider.

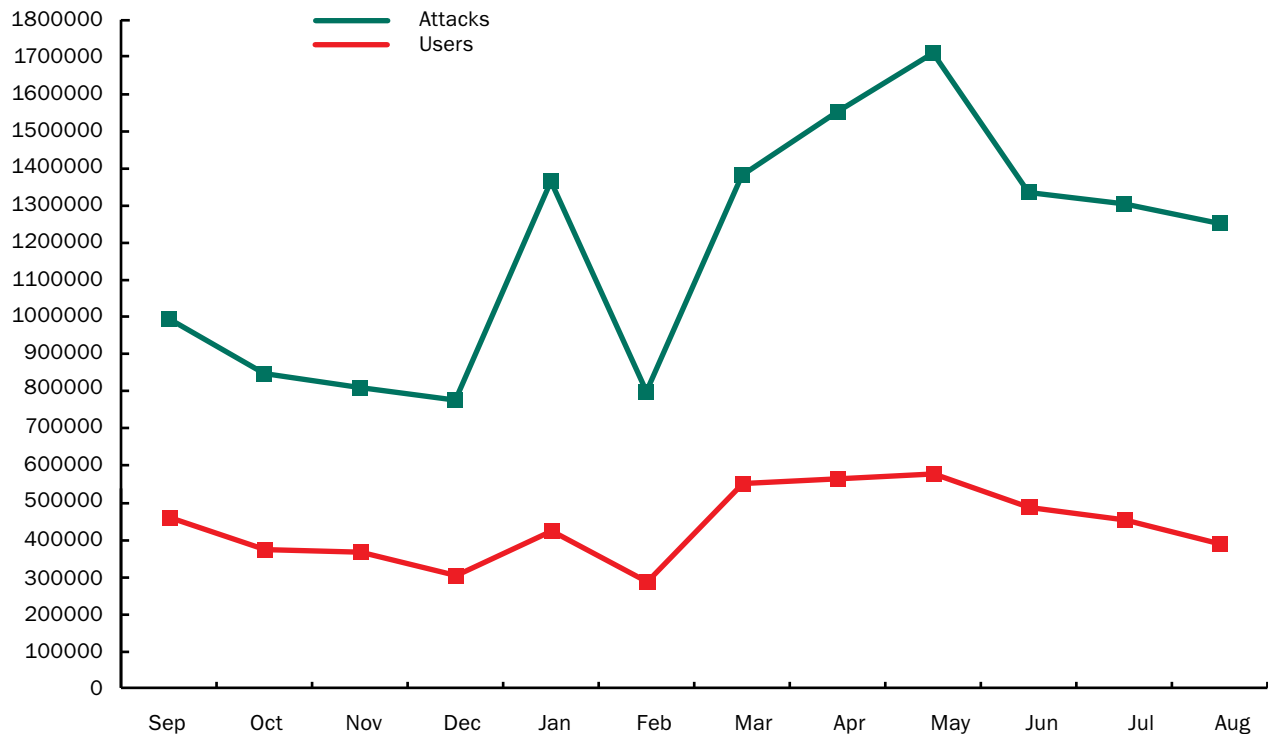
In addition to the number of attacks over the course of the year, the number of users who encountered these attacks also increased substantially.

More users in the crosshairs

Over the past year more than 14 million attacks targeted 3.75 million users in 226 countries. During the first six months of the year, 1.7 million individual users faced attacks involving Java exploits, compared with over 2 million in the next six months. That increase in March through August was over 21% compared with the previous six months.

> **The overwhelming majority of targeted users (roughly 79.6%) live in just 10 countries, and 10 countries represented the vast majority of attacks (82.2%).**

Number of Attacks and Attacked Users, 2012-2013



The number of users attacked over the entire year changed in line with the number of attacks themselves. In September — February, the average intensity came to 3.29 attacks per user, while in March — August, that number was 4.15. From one half of the year to the next, the intensity of attacks increased by 26.1%. On average, over the course of the year, each user encountered 3.72 attacks. As we addressed above, the busiest period both for the number of attacks and the number of targeted users was spring 2013.

Remarkably, however, after the "spring shake-up," the number of attacks fell more quickly than the number of affected users. For example, in June the number of attacks compared to May fell by 21.9%, while the number of users attacked fell by 15.5%. It is still possible that Oracle's release of the U25 update played a major role in this development, as it would mean patching up critical vulnerabilities in Java software. Due to the decrease in the number of users of vulnerable versions of Java, cyber crimnals could have taken measures to attract more new individual users to malicious websites. The logic is simple: the more users there are, the greater the chance that someone will have an outdated version of Java.

In general, these fluctuations in the number of attacks and the number of victims are worrying. Both numbers rose considerably over the past year. However in addition to the changes in the dynamics of the overall number of attacks, we have also seen some other curious trends in their distribution by country over the past 12 months.

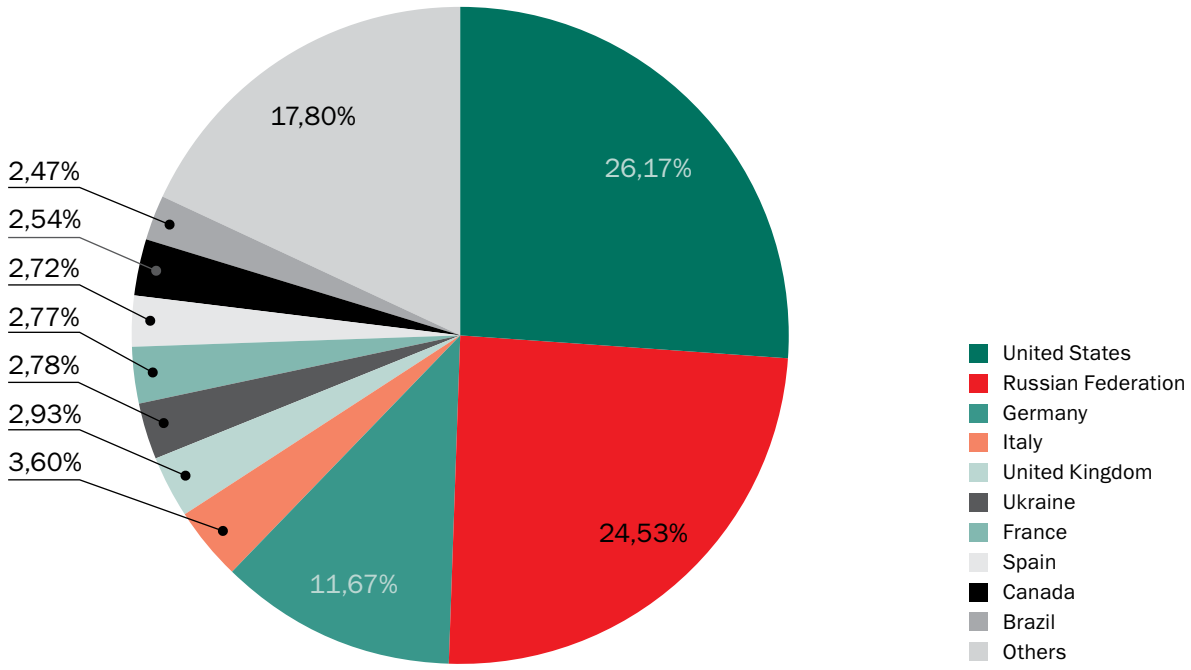
PART 3:

International threat — geography of users, attacks and sources

Geography of attacks — Ten major victims

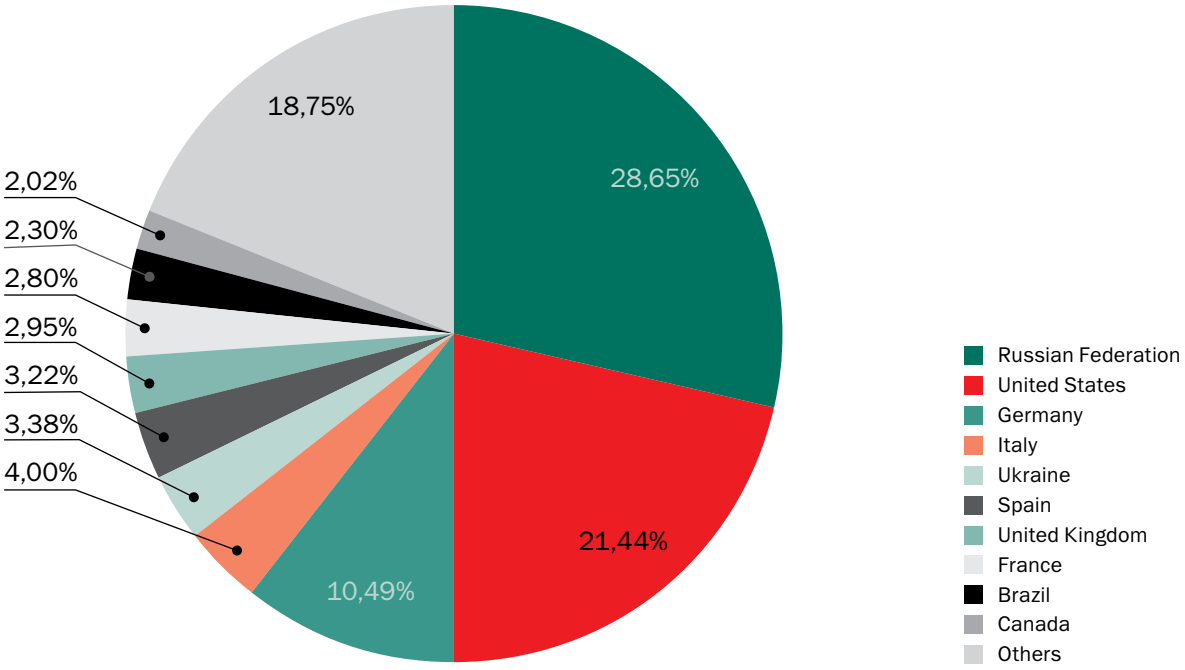
One of the most interesting results of this part of the study is the fact that the overwhelming majority of attacked users (about 79.6%) live in 10 countries. The vast majority of attacks (82.2%) also occurred in these ten states.

Top 10 most frequently attacked countries, 2012-2013

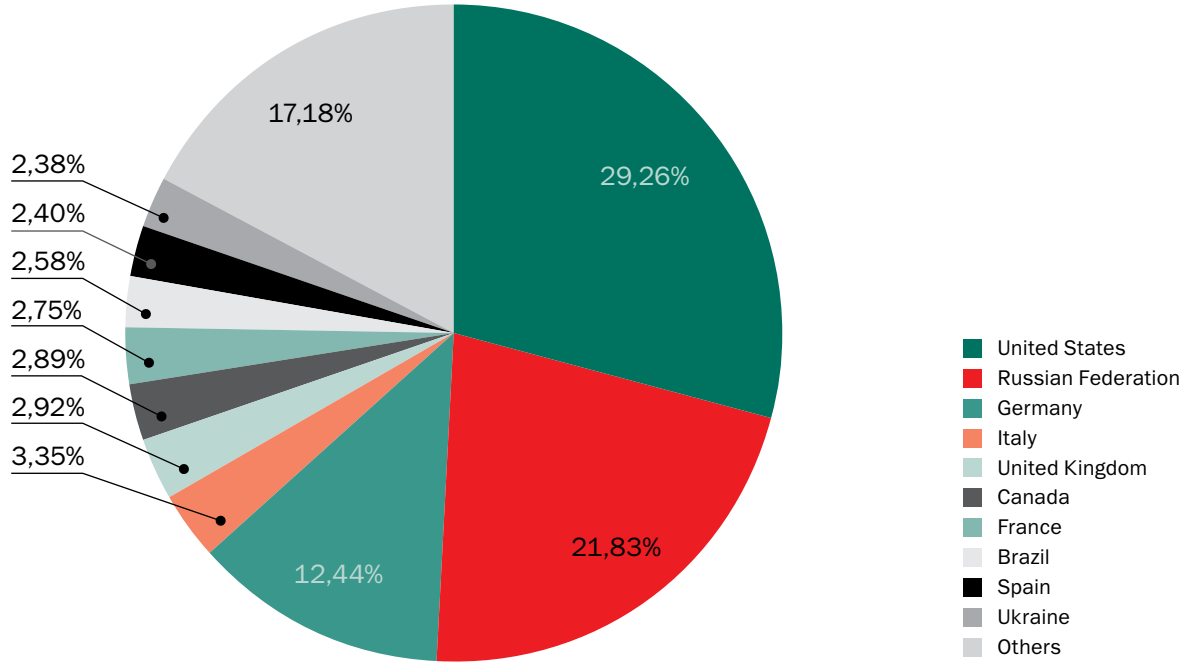


The table above shows the USA in first place, receiving 26.17% of all attacks. Russia comes second with 24.53%, and Germany (11.67%) is third. Meanwhile, the list of most attacked countries changed during the year.

Top 10 most frequently attacked countries, September-February 2012-2013



Top 10 most frequently attacked countries, March-August 2013



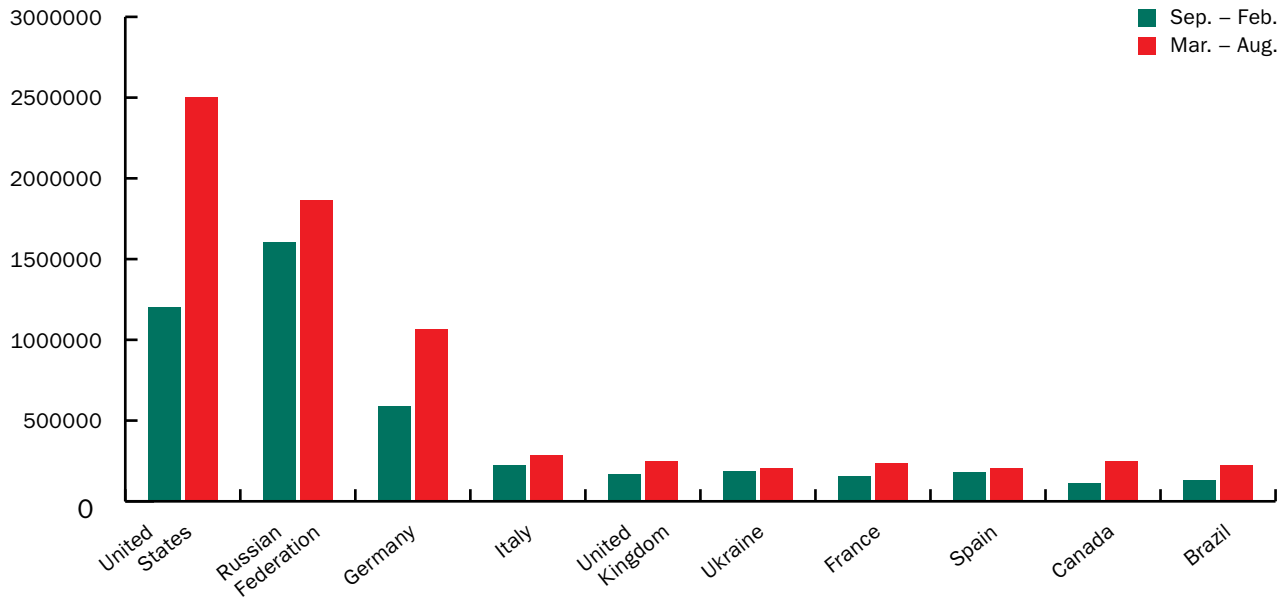
In March-August 2013 the USA and Russia swapped places. Attacks in the USA increased from 21.44% to 29.26%. Attacks on Russian users fell 6.82 percentage points to 21.83%.

In Canada the number of attacks increased by more than 118% and reached 0.24 million; in Brazil they climbed by 72% (0.22 million attacks). The number of incidents in the UK increased by 51%; in the second half of the year over 0.25 million attacks were detected.

The German "contribution" to the common attack picture increased noticeably — by 1.95 percentage points. Other countries also reported growth. Only Ukraine's share decreased — by 1 percentage point from 3.38% to 2.38%. The rates of other countries varied insignificantly.

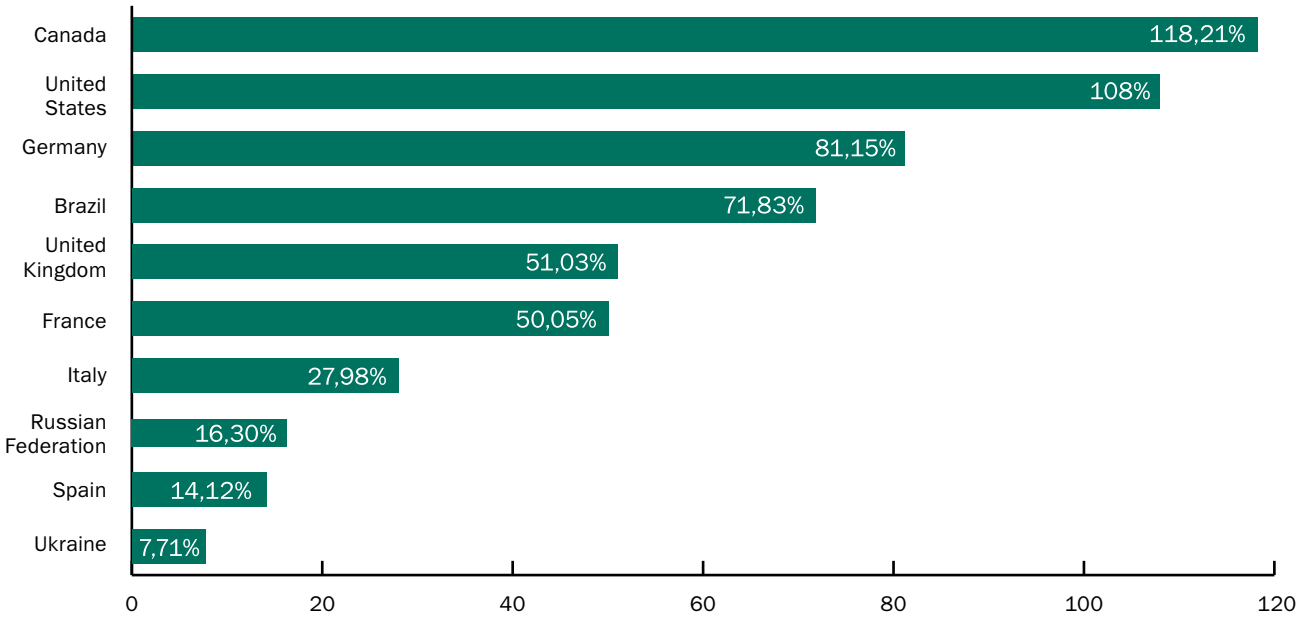
In total it looks like this:

Most frequently attacked countries, 2012-2013



Over the period from March till August the USA faced twice as many attacks — 2.5 million in comparison with 1.19 million attacks in the previous 6 months. In March-August 2013 more than a million attacks were detected in Germany, while over the period from September to February the number was 0.58 million. In general, the USA, Russia and Germany are leading countries in terms of number of attacks. However, the dynamics of the increase among the leaders varies significantly.

Attack increase dynamics in 2012-2013



In Canada the number of attacks increased by more than 118% to 0.24 million; in Brazil the increase was by almost 72% to 0.22 million attacks. The number of incidents in the UK increased by 51%; in the second half of the year more than 0.25 million of attacks were detected. The lowest growth was seen in Russia, Spain and Ukraine.

Not surprisingly, as the number of attacks increases, so does the number of users facing attack. The ratings of the leading countries are similar: almost half of all users attacked during the research period (48.27%) live in Russia and the USA. Every tenth victim lives in Germany.

In terms of the intensity of attacks on individual users, the leaders are Brazil and the USA — during the study period these countries respectively saw 5.75 and 4.79 attacks per user, which is significantly higher than the average rating for other countries. Germany is third with 4.04 attacks per user, and Italy came fourth with 3.82.

The Top 10 countries for attack intensity are:

Country	Attacks per user
Brazil	5.75
United States	4.79
Germany	4.04
France	3.65
Canada	3.58
Spain	3.42
United Kingdom	3.39
Russian Federation	3.32
Ukraine	3.04

Why is the intensity of attacks increasing? The large number of vulnerabilities discovered in Java and users who pay no attention to software updates create the perfect conditions for cyber criminals to launch more intensive attacks.

> **Almost a half of all attacked users for the period in question (48.27%) live in Russia and the USA.**

Having learnt who was most often attacked and where, we will now analyze information on the sources of these attacks to gain a better understanding of the problem.

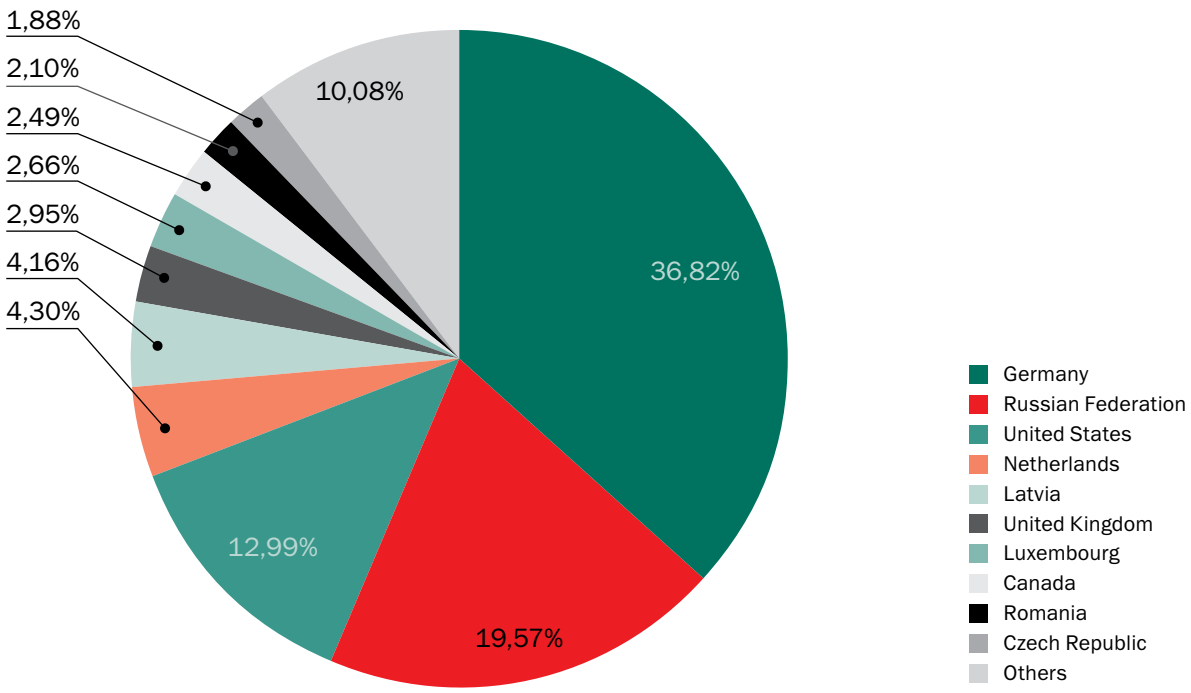
Sources of attacks — The new players are in

One striking detail of the geographical location of many attack sources — servers, in plain terms — is the fact that the countries hosting these servers are often far removed from the countries which are on the receiving end of attacks.

> **In total during the year 1.21 million unique attack sources were detected in 95 countries. Over the half of them were located in the USA, Germany and Russia.**

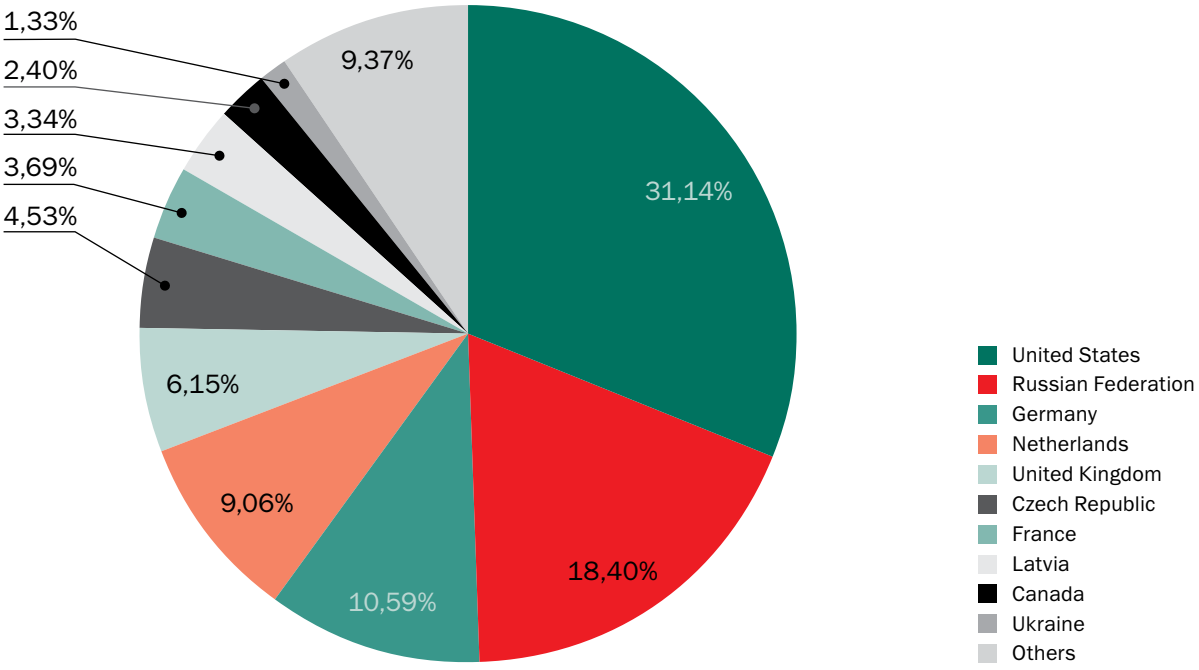
From September to February Kaspersky Security Network detected 0.41 million malicious servers located in 86 countries.

Top 10 countries from where attacks originated, September-February 2012-2013



From March to August Kaspersky Security Network collected information about 0.80 million malicious servers in 78 countries. The total growth in attack sources is 95.2%.

Top 10 countries from where attacks originated, March-August 2013

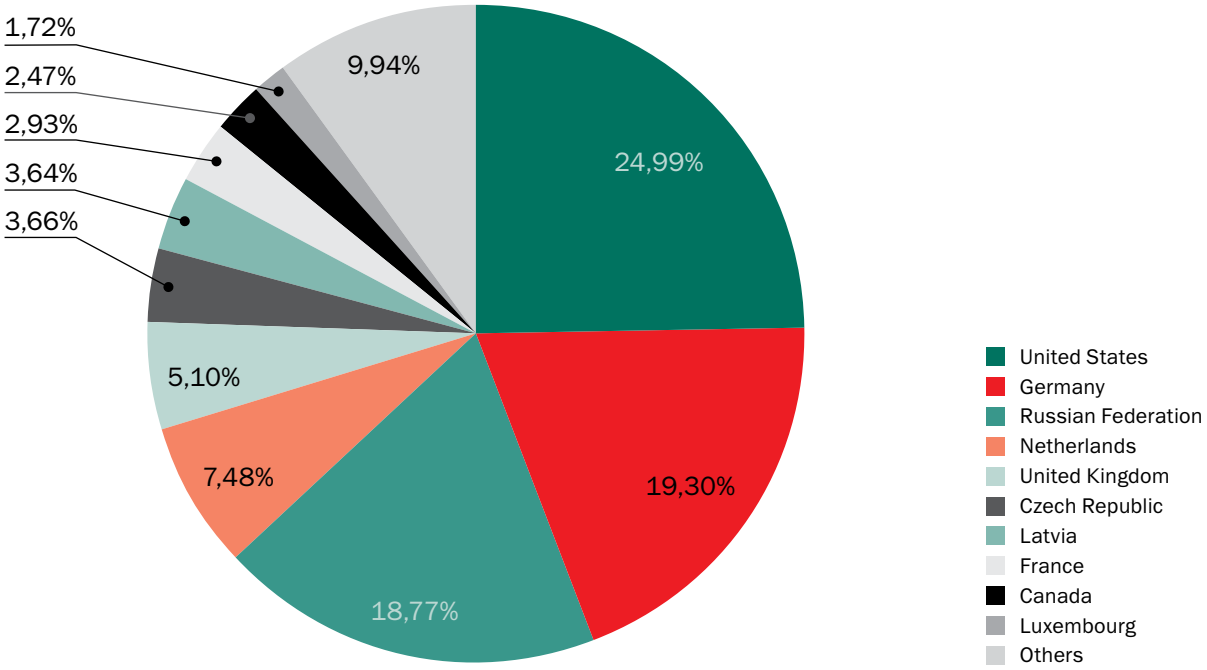


However, the most interesting thing here is not that the number of sources doubled within six months, but the fact the top 10 countries changed dramatically. Germany's numbers fell more than threefold from 36.82% to 10.59%, pushing it down to third place in the second half of the year. Russia remained in second place and its figures did not change much — 19.57% from September to March and 18.40% from March till August.

Germany and the USA swapped places. In the first half of the year only 12.99% of attack sources were located in the United States; in the second half this rose to 31.14%.

In total, after 12 months of studies the Top 10 countries serving as attack sources are as follows:

Top 10 countries from where attacks originated, 2012-2013



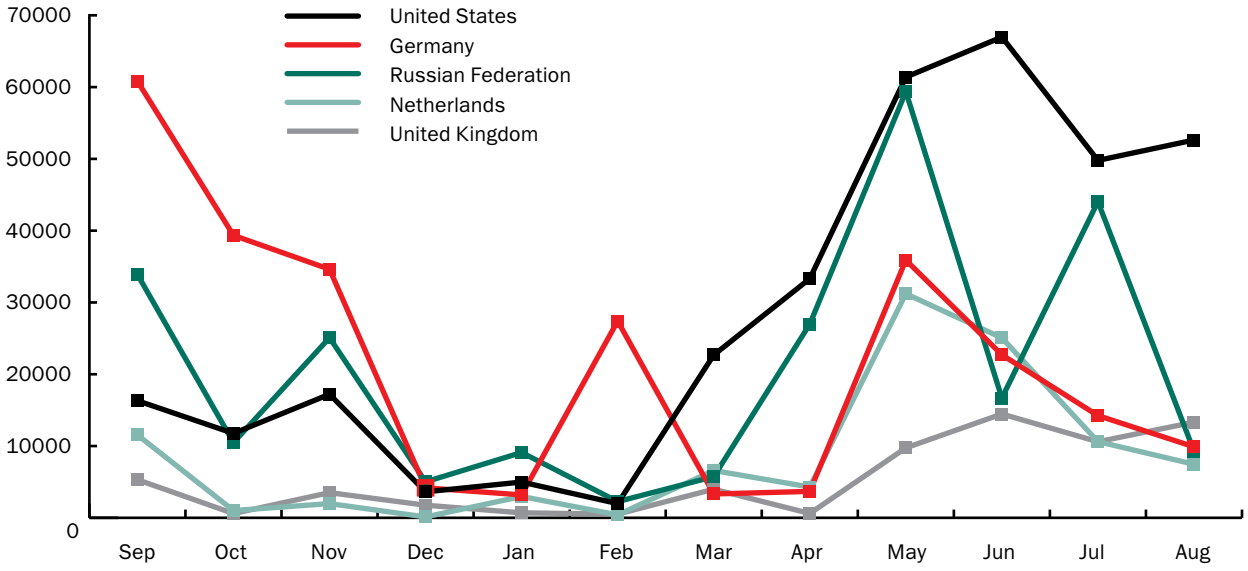
In total during the year 1.21 million unique attack sources were identified in 95 countries. More than half of them — 63.06% — are located in the USA, Germany and Russia. Other significant sources of exploit attacks can be found in the Netherlands (7.48%), the UK (5.1%), the Czech Republic (3.66%), France (2.93%), Canada (2.47%), and Luxembourg (1.72%). The remaining 10% of sources are located in 85 other countries.

The USA shows a strong and steady increase in the number of attack sources, with the numbers rising since February and peaking in July.

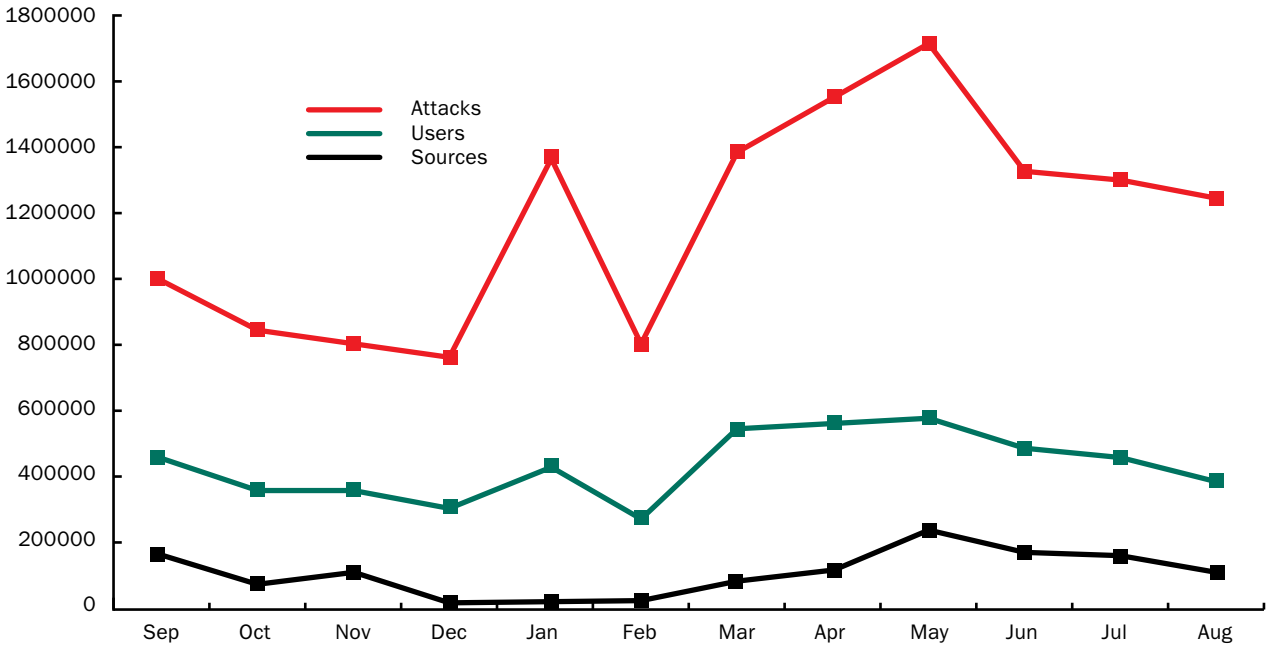
It is striking that several countries are highly-rated among “attackers” but are not among the most prominent victims. This could be connected to the fact these countries — the likes of the Netherlands, the Czech Republic, Latvia or Luxembourg — are not regarded as high-risk addresses for cybercriminals, so hosting providers are less likely to react to complaints about malicious sites hosted on these servers. Of course, these companies can be found in many other countries as well, but the numbers are greater in the nations mentioned above.

During the year attack sources changed as follows:

Top 5 countries from where attacks originated in dynamics, 2012-2013



At the start of the study period Germany was a consistent leader, even though the number of attack sources fell apart from a spike between January and March. The USA showed the steadiest and biggest increase in the number of attack sources, climbing from February and peaking in July. In March the number of malicious servers located in Russia increased rapidly. It is interesting that by the end of the period of the study three members of the Top 5 — Russia, Germany and the Netherlands — were on a downward path while the number of malicious servers in the USA and the UK grew. However, as in the table below, it did not greatly affect the total number of malicious servers. As with the attacks and attacked users, the number of servers decreased during the summer.



The table above shows the correlation of the number of attacks, attacked users and servers from which attempts were made to download exploits.

On average, during the year 11.64 attempts to download an exploit were made from each of the more than 1.2 million of unique identified IP addresses.

Is that a lot? Firstly, one should take into consideration these figures reflect only Kaspersky Lab's users who have encountered such pages. In reality there could be many more downloads from each IP address. Secondly, exploits are a very dangerous type of malware. A successful download may lead to serious damage, including financial losses. In other words, the damage caused by attacks conducted with exploits is disproportionate to their comparatively small number.

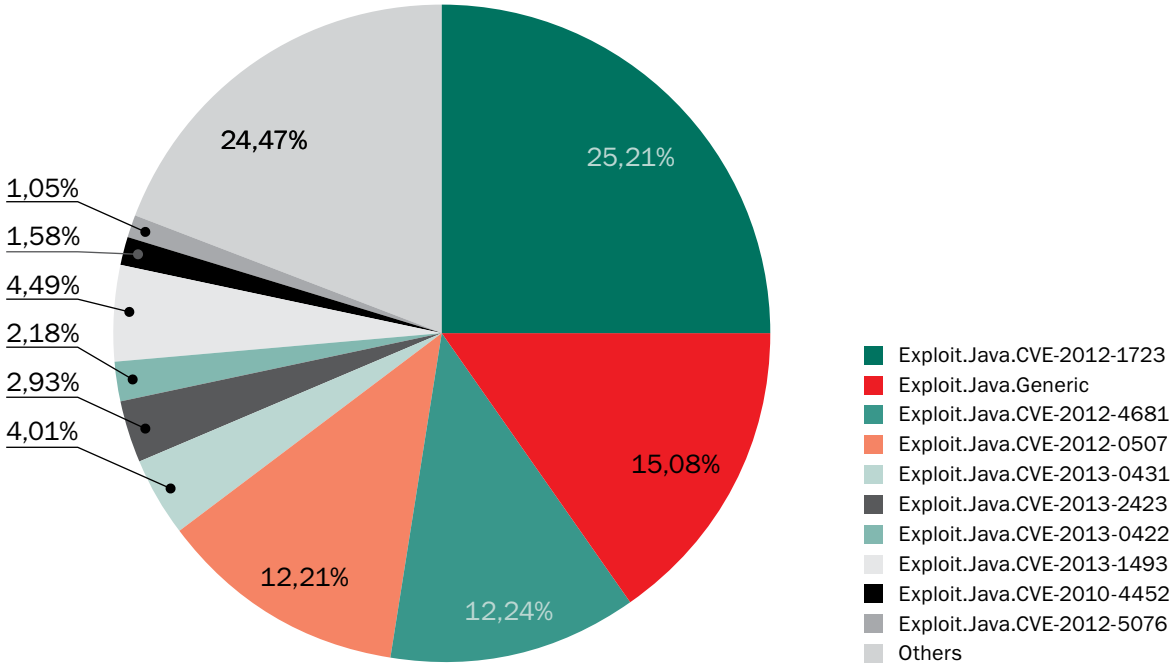
Speaking of the attacks themselves, during the study period only six critical vulnerabilities were detected in Java. Could they really cause serious harm to unprotected users? We will look at that in detail in the next part of the report.

PART 4:

A closer look at exploits "In the wild"

Over the reporting period, Kaspersky Lab products detected 2,047 different malicious families categorized as exploits. However — and this is typical — just nine of these families account for the majority of the attacks that were launched.

Top 10 exploits, 2012-2013

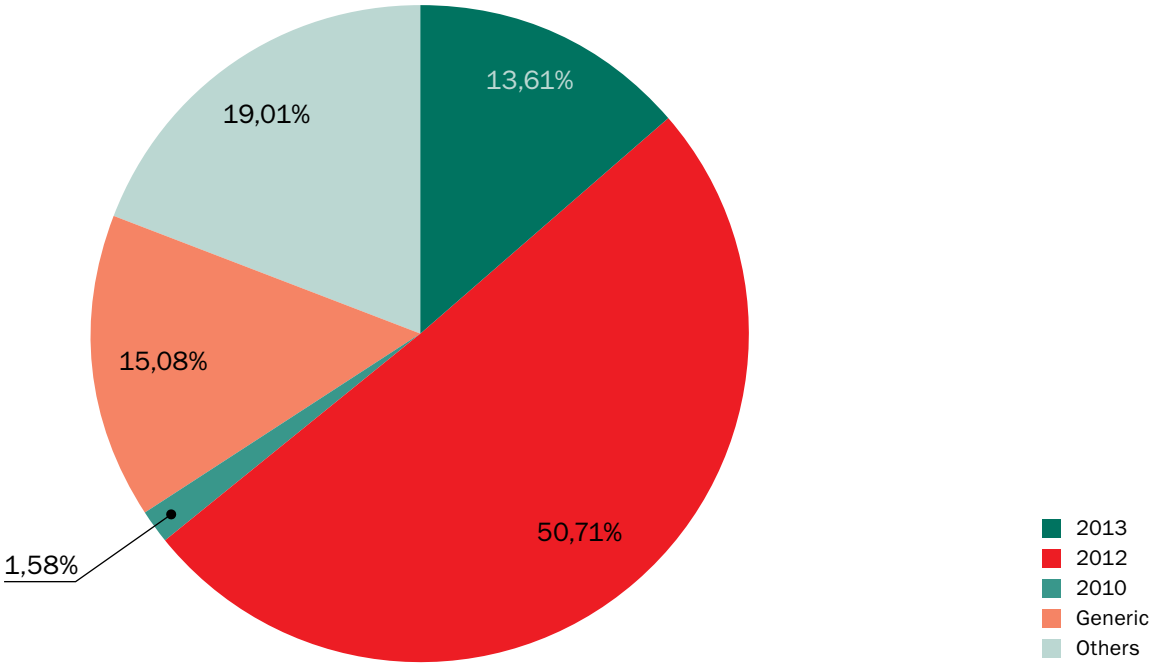


Roughly 81% of all attacks were launched from just 9 exploit families targeting 9 vulnerabilities.[1]

The chart below is another illustration of the sad state of updates for vulnerable software programs. Over 50% of the detections processed by Kaspersky Lab technologies came from exploits specifically targeting vulnerabilities that were discovered in 2012. In 2013, they represented about 13.61% of attacks.

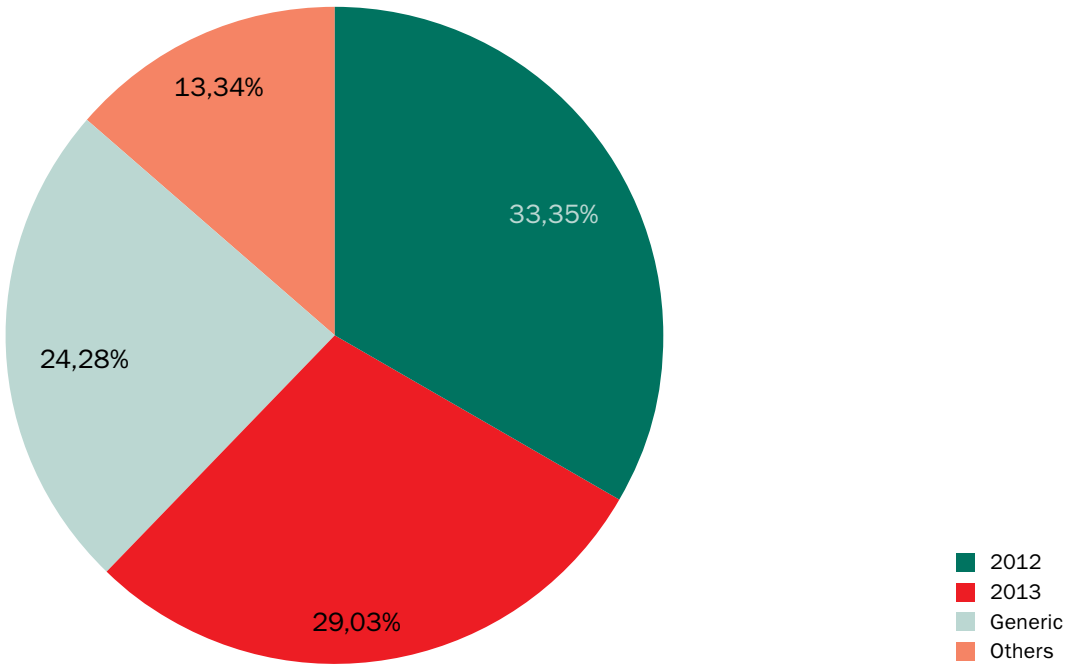
[1] The Exploit.Java.Generic signature means that Kaspersky Lab's cloud security technologies detected behavior that is typical of exploits, but at the time that statistics were prepared, these detections were not identified in terms of which Java vulnerability they were targeting. Just over 2 million such detections were recorded over the reporting period.

Top 10 Java exploits by year of discovery of the vulnerabilities, 2012-2013



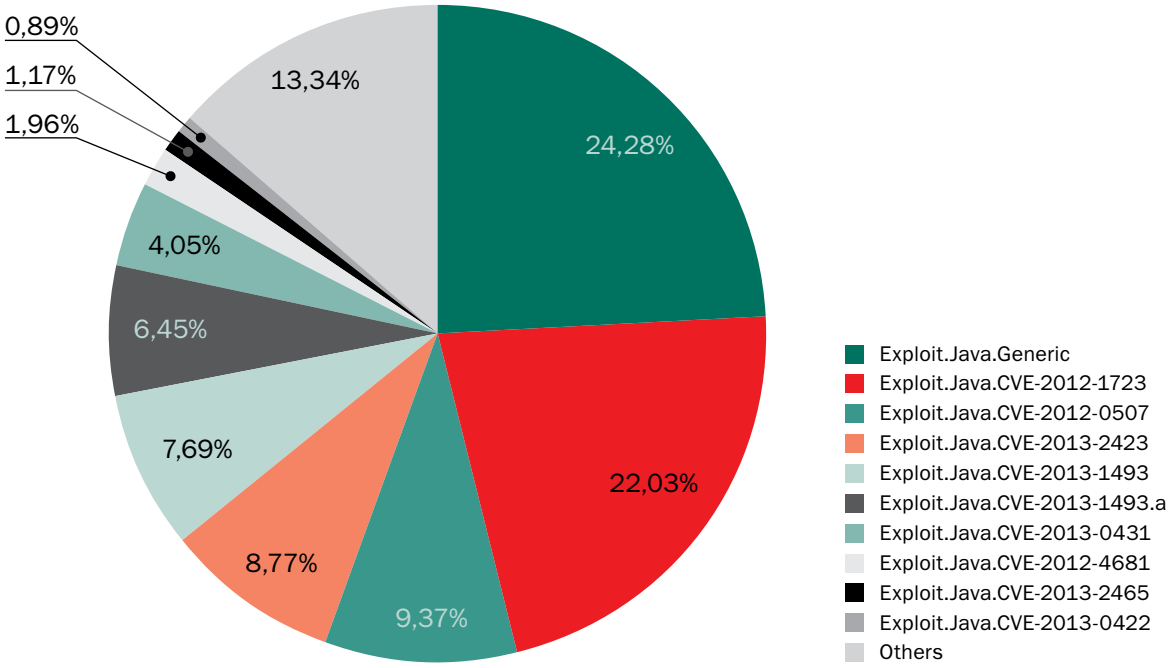
One could presume that the aggregate statistics for the past 12 months don't really give a clear picture of the situation, since these exploits were designed to target vulnerabilities that weren't patched in 2012 and had more time to spread. In fact, if you break the numbers down for August 2013, it becomes clear that the unpatched exploits in 2013 were responsible for a much larger number of attacks in the last month of summer than they were over the year on average.

Top 10 Java exploits by year of discovery of the vulnerabilities, August 2013



However, if you look at the Top 10 exploits for the same month, the situation looks different.

Top 10 exploits, August 2013



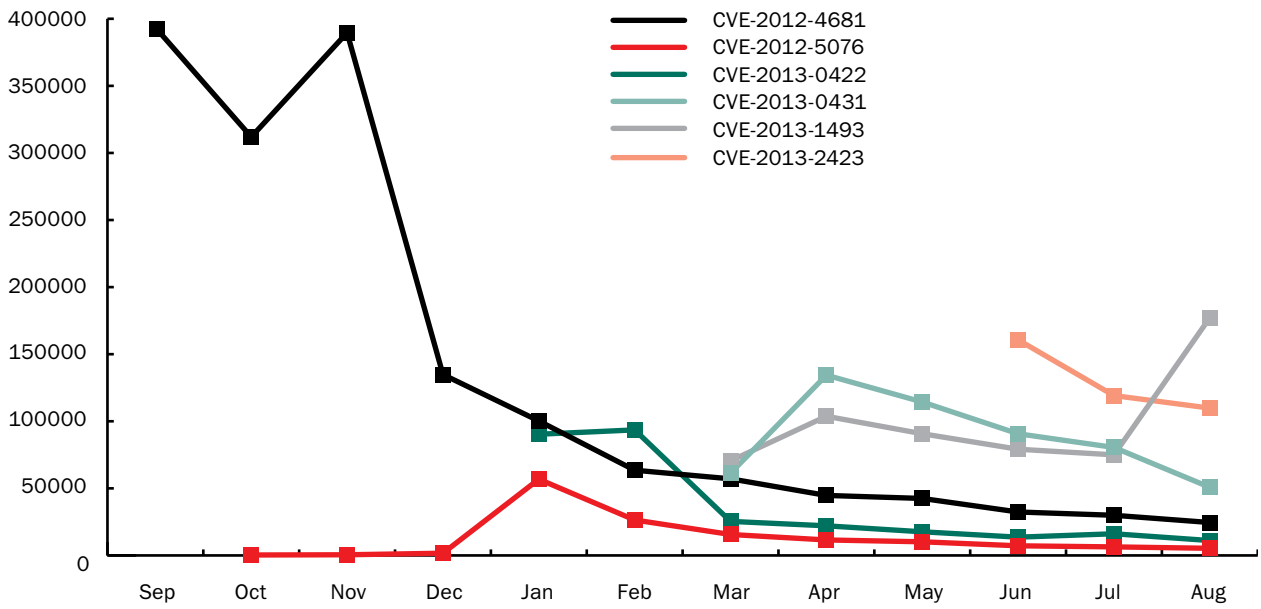
CVE-2012-1723 is still the leader among identified exploits. The first public announcement about this vulnerability was made in June 2012. Oracle issued a patch just a few days later. As you can see from the pie chart, more than one year later, 20% of attacks still involve CVE-2012-1723.

Over 50% of the detections processed by Kaspersky Lab technologies over the 12-month period involved exploits targeting vulnerabilities that were unpatched in 2012. Vulnerabilities detected in 2012 account for roughly 13.61%.

Nevertheless, as we already mentioned above, over the reporting period, six new critical vulnerabilities were detected. This is how the landscape of attacks has been changing as new holes in Java security are detected.

"The relay race" of the new exploits, 2012-2013

The lines in the chart represent exploit-led attacks registered over the course of the year.



The dynamics of attacks involving exploits that appeared during the reporting period are reminiscent of a relay race. As the number of attacks targeting one vulnerability begins to fall, a new one arises, and the exploits targeting that one take the baton and lead the race. This chart also shows how CVE-2012-4681 was involved in the attacks. That particular exploit was detected just three days before the period covered by this study began, and was a Zero Day-type vulnerability posing a critical threat. Although Oracle did release an emergency patch on 30 August 2012, the chart shows that even after a slight dip in the number of attacks in October, it was climbing back up the chart by late November.

The exploit targeting CVE-2012-5076 was discovered in mid-October 2012 and took up the baton from CVE-2012-4681. It immediately became popular among cyber criminals, since it is capable of infecting several versions of Java. Although the patch for the vulnerability was released that same month, in March 2013, this exploit was still listed on Blackhole, one of the most common exploit packs on the black market, because it can target a broad range of versions and infects victims more reliably.

CVE-2013-0422 is another Zero Day vulnerability that was discovered in January this year. A patch was released quickly, but the number of attacks increased, right up until February. Then, the drop in the number of attacks began to decelerate, and settled in at a level of 25,000 detections per month right up until August. Just like its predecessor, this exploit was listed on Blackhole together with other well-known exploit packs. CVE-2013-0422 along with CVE-2012-1723 were used in **Icefog** spying campaign discovered by Kaspersky Lab researchers recently.

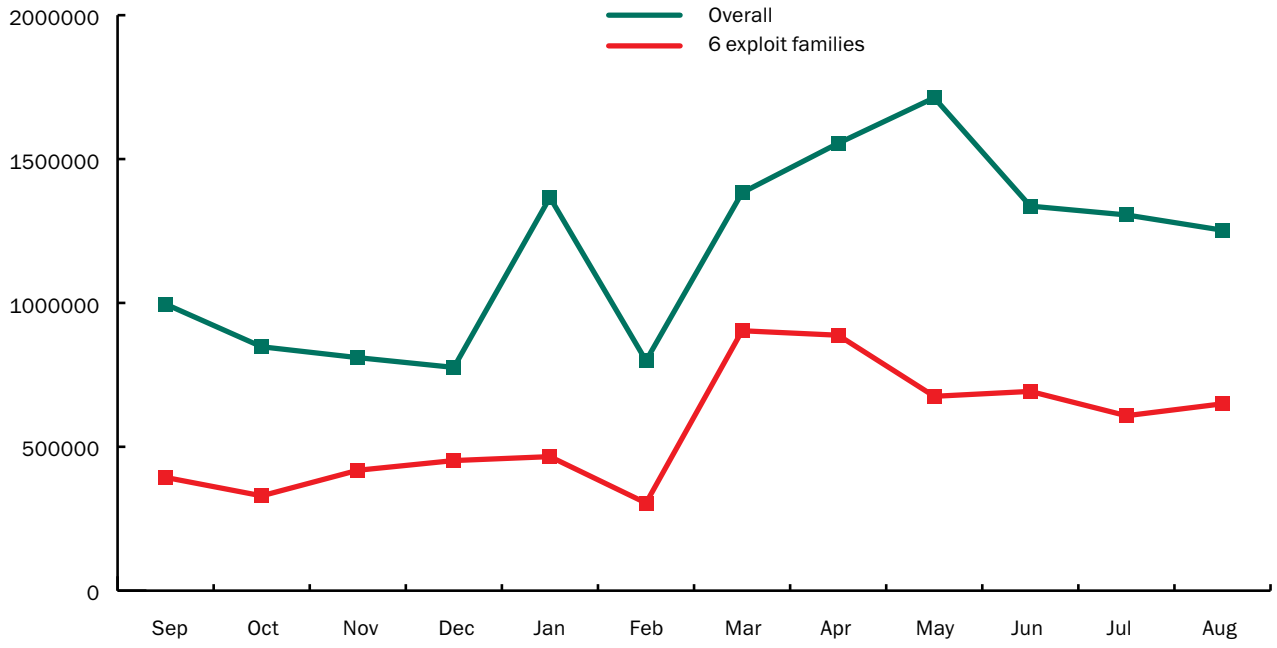
The CVE-2013-0431 vulnerability, which appeared in early February, was actively used including by a gang of cyber criminals spreading **Reveton**, a scareware Trojan. This program blocks the victim's computer and displays a message allegedly from the FBI stating that the user has violated a law and has to pay a fine. The group behind these types of attacks was arrested by the Spanish police in early February 2013, although that did not stop the spread of this Trojan. In mid-February, information appeared online about a computer infected by Reveton after an attack involving exploits targeting CVE-2012-0431.

The CVE-2013-1493 vulnerability that appeared in early March was also targeted in attacks that were mostly geared towards industrial espionage. Specifically, according to data from independent studies, the malicious program that was installed after the exploit was triggered connected with the malicious user's server on the same IP address as the C&C server used in the attack against Bit9 in February this year.

> **In absolute terms, attacks involving six families of exploits represent at least 47.95% of the total number of exploit attacks. That is nearly one-half of all Java exploit attacks detected by Kaspersky Lab products.**

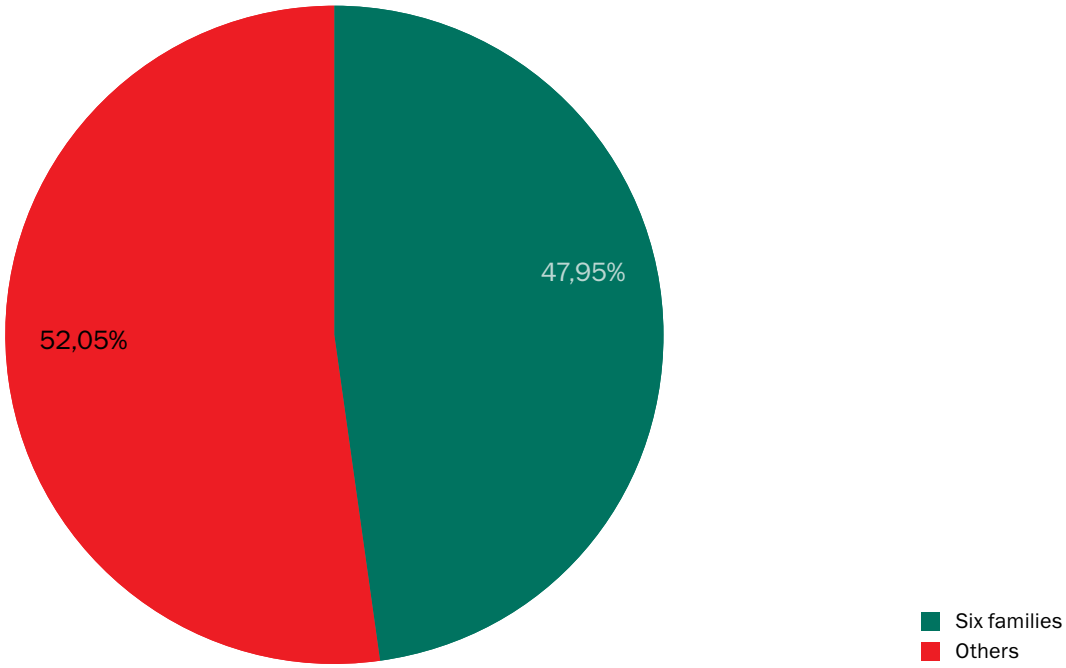
Exploits targeting CVE-2013-2423, which was detected in June 2013, were found by Kaspersky Lab experts to be involved in attacks against Apple users. It is thought that these attacks were launched as part of a campaign against people visiting the website of the exiled Tibetan government, something Kaspersky Lab **addressed previously**.

All of this data is just further illustration of how malicious users took advantage of vulnerabilities that were discovered during the reporting period. In general, there were six vulnerabilities that defined the flow of attacks against Java in September 2012 through August 2013.



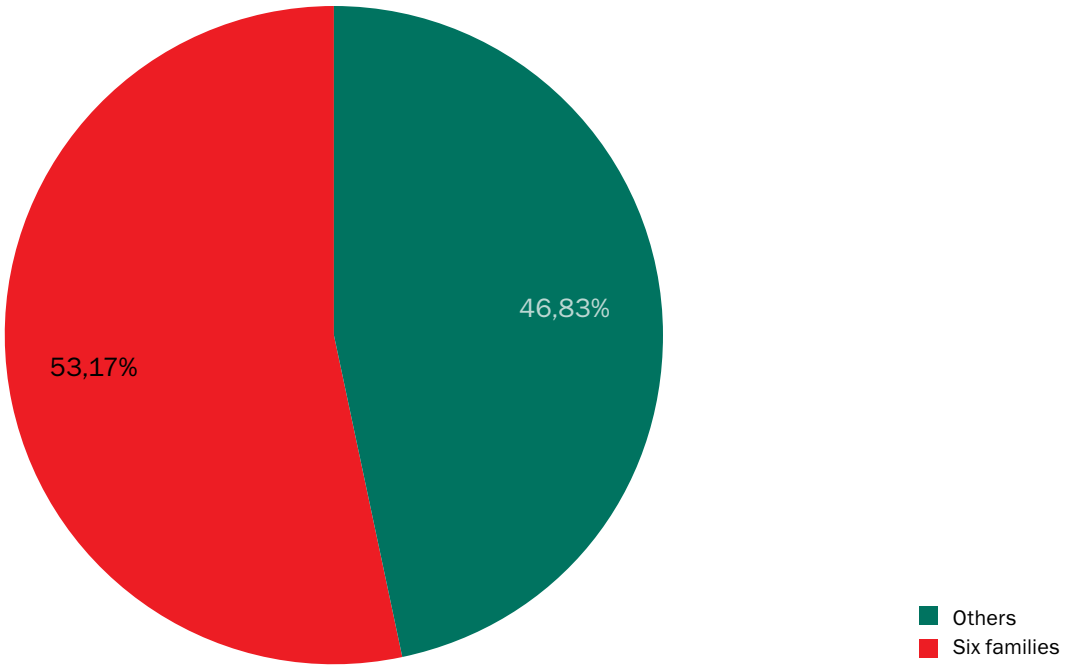
In absolute terms, the attacks involving these families represented at least 47.95% of the total number of attacks, or nearly one-half of all Java exploit detections processed by Kaspersky Lab products.

Six families in overall number of attacks via exploits



At the same time, more than half (53.17%) of the total number of users subjected to attacks encountered at least one attack involving an exploit from one of these six families.

Number of users attacked with the 6 new exploit families in overall number of users



And that was 12 months of attacks targeting Java. However, before we bring together the results of this study, let us first take a look at just one more dangerous type of attack encountered by Kaspersky Lab users over this 12-month period.

PART 5:

Plus 4.2 million of attacks caught by Automatic Exploit Prevention technology

The key figure in this study is that Kaspersky Lab products blocked over 14 million attacks involving Java exploits. In fact, the company's security system registered 4.2 million more incidents. That is the number of attacks that were blocked by Kaspersky Lab's one-of-a-kind **Automatic Exploit Prevention technology**. Those 4.2 million attacks would have affected 2.25 million individual users around the world.

Why we decided to analyze Automatic Exploit Prevention data separately

The fight against attacks using exploits is a multi-step process in which several security subsystems of an antivirus product play a role. The first step takes place at the level of the webpage, when the security product, guided by a database of malicious websites, will attempt to block any user redirects to a site seeded with an exploit. If that attempt fails, perhaps because the website was just created and is not yet included on the blacklist, then the file antivirus module gets involved. It scans the page for any sign of malicious code, based on the signature database and heuristic entries. The latter are expanded with signatures that help detect as-of-yet unknown malicious code using signs that are typical of known malware. If that doesn't work, then the next step is to perform a scan using the exploit signature database. If this doesn't produce any results, then the proactive exploit detection technology is activated — Automatic Exploit Prevention is one of the components of this technology built into Kaspersky Lab products. You can learn more about how it fights exploits in **this article** published by Kaspersky Lab's Vulnerability Research Group Manager Vyacheslav Zakorzhevsky.

At this point, we should note that in the exploit counteraction process described above, Automatic Exploit Prevention is, in a way, a kind of "final frontier". It is not the final frontier of security — if for some reason, malicious code manages to get past Automatic Exploit Prevention (which is highly unlikely) and download malware onto a user's computer, it will still be detected and blocked by other components of the security product. Nevertheless, Automatic Exploit Prevention is unique in that it considerably lowers the probability of that scenario playing out.

While most other security technologies are built to search out malicious components in code that is entering a computer system from outside, Automatic Exploit Prevention analyzes the behavior of legitimate components, rather than malicious components. In Java's case, this means the components of the software environment installed on the user's computer. Generally speaking, Automatic Exploit Prevention has a "basic idea" of how one or another Java component should or should not operate. In the case of analytical systems, the technologies note anomalies in the behavior of Java components, and if that software begins to operate in a way that the developer did not intend it to, Automatic Exploit Prevention activates and blocks the exploit.

This technology is critically important for ensuring top-quality user protection against cyber-attacks for one simple reason: malicious users are ready and willing to invest in developing ways to get around the security systems installed on the computers of their potential victims.

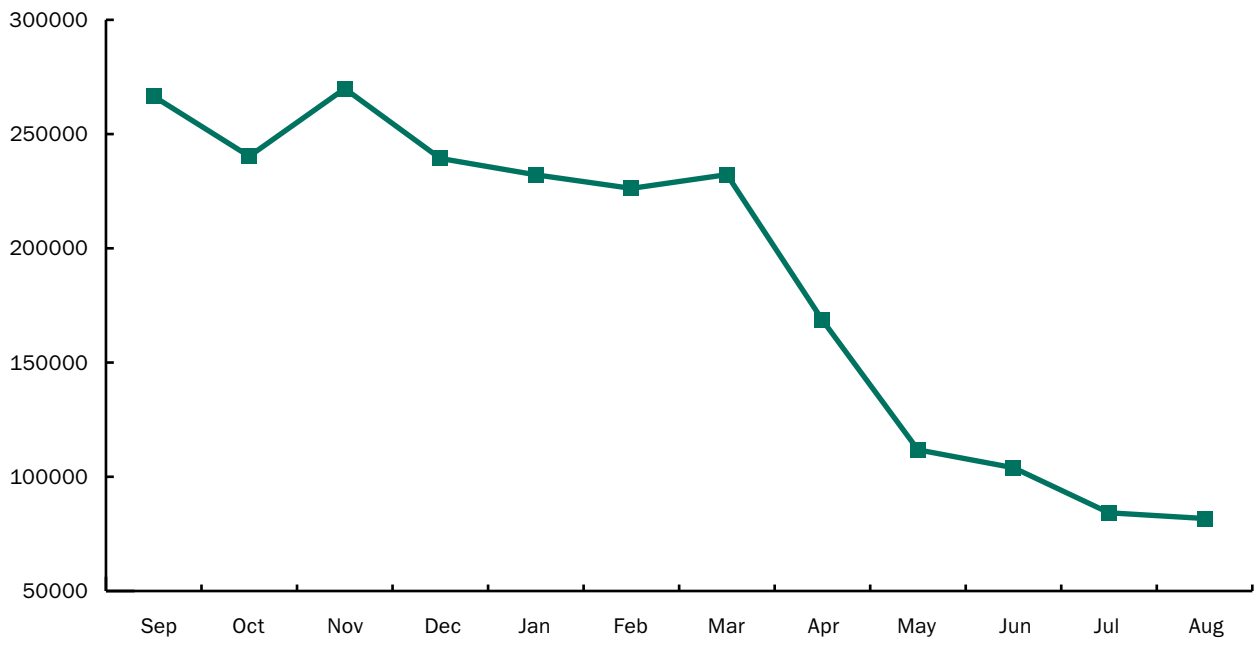
According to information that can be found on any hacker forum — even one that isn't closed to the public — successfully using exploits to download malware 1,000 times runs about $80-$120. And if a "client" of these services wants to see any type of potential "profits" on infected computers, such as online banking Trojans or encryption Trojans, then the price per thousand downloads could be as much as $140-$160.

> **There is no reason to believe that all 4.2 million attacks blocked by Automatic Exploit Prevention were targeted attacks, but it can be presumed with a fair degree of certainty that much more care was taken in preparing these attacks, compared to others.**

The high success rate of these types of attacks makes the creation of effective exploit packs a lucrative endeavor for cyber criminals. That is why they are willing to spend time and resources painstakingly studying antivirus programs in order to figure out how to beat the detection technologies in any antivirus component.

The chart below shows the dynamics of these more sophisticated exploit-led attacks:

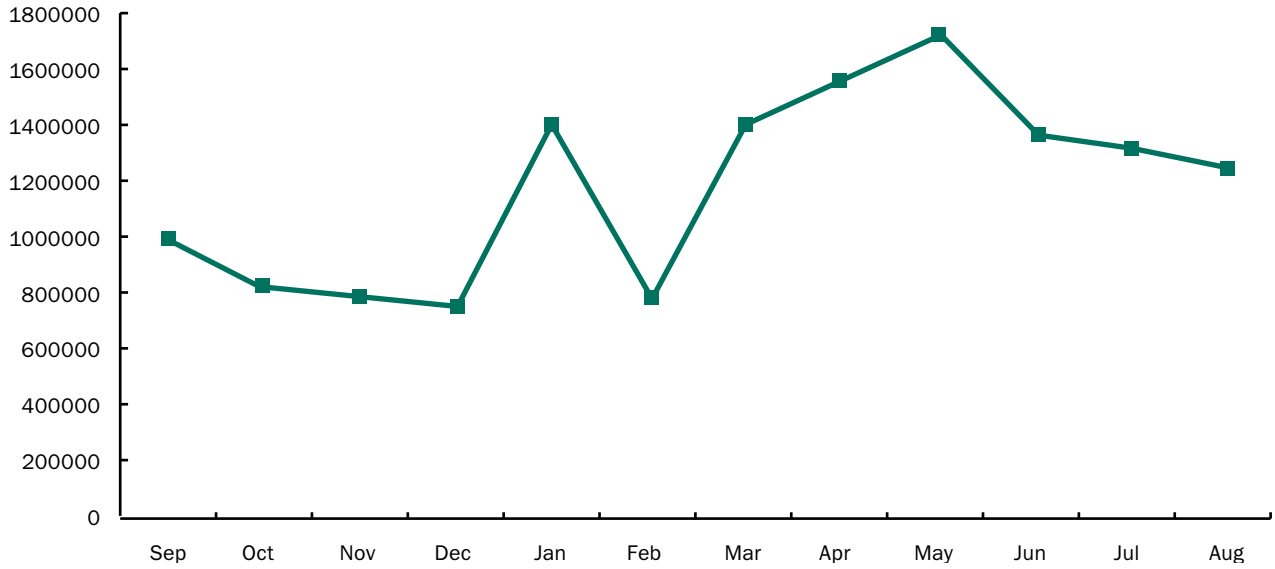
Dynamics of attacks blocked by AEP



The distinct decrease in the number of AEP detections after March 2013 can be explained by two key factors: the constant improvement by Kaspersky Lab experts of other exploit detection technologies, and the fact that cyber criminals simply did not need to make their attacks so sophisticated.

It suffices to turn back to the chart from the start of this study:

Attacks, 2012-2013



While the number of AEP detections has gone down, the number of detections of other, "higher-level" Kaspersky Lab security components has, on the contrary, gone up.

At the same time, three critical vulnerabilities were discovered in Java, and exploits targeting them already successfully attacked users. In other words, malicious users have a large supply of targets to attack, and they had no need to take extra care to disguise already known exploits or search out new ones. By June, Oracle had already patched up most of the security breaches in its product, and the situation began to change. We don't have any reason to believe that all 4.2 million attacks blocked by Automatic Exploit Prevention were targeted attacks, but we can say with a fair degree of certainty that these attacks were prepared with much more care than other attacks. After the **outstanding results** in expert lab tests, Automatic Exploit Prevention has now proven itself in the battlefield.

CONCLUSION:

The importance of sophisticated technologies in an age of sophisticated attacks

The key conclusion of the entire study can be boiled down to one sentence: no matter how fast developers release updates to patch vulnerabilities, this is still not solving the problem of exploit attacks. Oracle has patched all critical vulnerabilities, and information about these patches was disclosed during the period addressed in this report. In some cases, just a couple of days passed before a patch was released. All the same, the number of attacks and the number of users subjected to these attacks continued to rise.

Users do not install security updates quickly enough and cyber criminals, knowing that Java software is used by an enormous number of people, never tire of searching for vulnerabilities in that software so they can generate illegal profits from those errors.

Security solutions which can effectively counteracting exploits are becoming a kind of frontier in the security field, affording developers of vulnerable software more time to prepare patches, and users with a safe way to use the Internet until the patch is released. Furthermore, these solutions generally lower the profitability of cyber criminal efforts. In money terms, 14.1 million attacks blocked by Kaspersky Lab means the loss of at least $1.4 million (assuming $100 per each 1,000 successful “installs”) for cyber criminals trying to make money off of the sale of installs that ultimately do not take place.

In order to avoid attacks involving vulnerabilities and the losses that these attacks could cause, Kaspersky Lab experts recommend that corporations and home users adhere to the following rules.

For Businesses:

- A huge proportion of the risks associated with vulnerabilities will be neutralized if the corporate IT infrastructure includes a solution for administrating corporate personal computers with Patch Management functions, such as those found in **Kaspersky Lab Systems Management**. The Patch Management function helps to promptly install critical updates in a centralized system, keeping admins abreast of the condition of software running on any workstation within the company’s network.

- Most attacks that exploit vulnerabilities in legitimate software (including Java) begin with a link to a malicious website. These attacks can be prevented by blocking links to these sites with Web Control, a specialized resource that helps control user access. These resources can help strictly limit the list of websites that corporate computers are permitted to visit.

- Make workstation security even more reliable with Application Control, which allows only a restricted list of applications to launch on corporate computers. In the event that a potentially dangerous unpatched vulnerability is discovered in a program widely used within the company, Application Control can prevent that program from being run on any computer within the company network. The advanced Web Control and Application Control functions are performed using the robust platform for corporate IT infrastructure security provided in **Kaspersky Endpoint Security for Business**.

- Malicious users are often able to develop exploits targeting unpatched vulnerabilities some time before critical security updates for corporate software can be released. That is why total protection for the corporate infrastructure should make use of advanced security solutions with technologies that can counteract exploit attacks. Kaspersky Lab's **Automatic Exploit Prevention** effectively counteracts exploit attacks in partnership with the numerous other advanced technologies available in Kaspersky Endpoint Security for Business.

For Consumers:

- Most modern software, including Java, Adobe Reader, and Flash Player, contains a built-in software update system. Don't forget to use them, and don't ignore the prompts to install updates, since that is the most reliable way to keep the software running on your home computer up to date.

- Users can encounter exploits in places other than just a malicious web page. Often, malicious users will find errors in a legitimate, popular website, such as a media outlet, a social network, or an online store, and use it to spread exploits. In order to reliably protect yourself against all kinds of potential threats, including attacks through vulnerable software, use a quality Internet Security solution — such as **Kaspersky Internet Security** — with specialized technologies capable of counteracting phishing, spam, viruses, Trojans, and complex attacks.

- Java is a very widespread software that a computer typically requires to function normally when running multimedia content on the Internet. However, Java is not necessarily required in order to work with web content. So even if you have installed all relevant updates and the best possible security solution, if you still feel that your computer is not secure, simply switch off Java on your computer. This action can prevent some functions on some websites from working, but at the same time, these resources can always be accessed using a safer alternative technology.

KASPERSKY lab