MAY-JUNE 2020

State of Serverless Application Security Bimonthly Report

Table of contents

01 Executive Summary

02 Evolving Threats in 2020

03 Vulnerability Trends

- TREND: Too Many Applications Continue to Have Serious Vulnerabilities
- TREND: More Java Applications Have Serious Vulnerabilities
- TREND: Vulnerable Applications Tend to Have Multiple Vulnerabilities

04 Attack Trends

- TREND: SQL Injection Attacks Trended Sharply Upward, Becoming the Most Common Attack Type
- TREND: Specific Attacks Impacted Many More .NET Applications

05 Application Security Watch List

06 Conclusion

01 Executive Summary

01 | Executive Summary

The Contrast Labs Application Security Intelligence Report for May–June 2020 leverages aggregate data from Contrast Security customers to provide insights about the vulnerabilities in software that we protect—and attacks on those applications. Findings of note include:

- **One in three applications** has at least one serious vulnerability—27% higher than the annual average.
- **A larger percentage of Java applications** have serious vulnerabilities compared with other languages, owing to the increased flexibility of that platform.
- While the volume of attacks subsided, the percentage of applications impacted by **SQL injection, broken access control, and command injection attacks** spiked.
- The increase in the percentage of applications targeted by specific attacks was **especially pronounced for .NET applications**, with five attack categories seeing double-digit increases in the percentage of applications impacted.

These findings accentuate the continuing struggle to deliver secure applications for developers, security, and operations professionals. These problems have been magnified in recent months by the COVID-19 pandemic—which has forced most developers to work remotely while facing increasingly aggressive deadlines as businesses scramble to adapt to a marketplace that has been turned upside down.[1] In this environment, a comprehensive approach to application security (AppSec) is more important than ever—one that builds testing and protection into every step of the software development life cycle (SDLC).

KEY FINDINGS

33%

of applications had a serious vulnerability in May–June 2020—27% higher than the annual average

5

Applications with expression language injection vulnerabilities have a median of 5 instances per application

81%

of applications saw a SQL injection attack—47% MORE than the prior two-month period

5

attack categories saw a DOUBLE-DIGIT INCREASE in the percentage of applications impacted compared to the prior two-month period.
Only 3 ATTACK CATEGORIES saw a decrease

02

Evolving Threats in 2020

02 | Evolving Threats in 2020

Contrast Labs' bimonthly Application Security Intelligence Reports provide an update on the status of AppSec as observed by vulnerabilities identified by telemetry directly measured from customers' applications running in development, test, and production environments. The dataset includes vulnerabilities identified by Contrast Assess and attacks detected by Contrast Protect.

Every two months, Contrast Labs analyzes this data to determine which types of vulnerabilities and attacks are most prevalent in protected applications. The report also identifies actionable insights that can aid development, security, and operations teams as they refine their application security strategy. While the fluctuations over a two-month period are sometimes small, publishing the reports on a regular basis helps readers to identify trends on an ongoing basis.

COVID-19 continues to threaten public health and the economy, especially in the United States. At the same time, agitation for social change has made the headlines every day for more than two months. In the midst of the chaos, successful companies are adjusting their business models to emphasize digital engagement, and the underlying digital transformations are continuing on schedule—or, in many cases, accelerating.[2]

At the same time, cyber criminals are adjusting their tactics to take advantage of the situation.[3] For example, Google is blocking more than 18 million coronavirus-related scam emails on a daily basis.[4]

Tactics used in attacks on web applications are evolving as well, with more sophisticated account takeover (ATO) attacks, more widely distributed botnets, and more.[5] In other instances, federal indictments were issued against suspected Chinese hackers targeting companies doing COVID-19 research[6] and a New York City man for stealing payment card information for thousands of accounts.[7]

Responding to this changing landscape is complicated by the accelerating volume of new vulnerabilities discovered.[8] Microsoft has averaged 90 Common Vulnerabilities and Exposures (CVE) fixes per month in recent months,[9] and several high-profile critical vulnerabilities have been identified in open-source Java libraries recently. Overall, 2,460 vulnerabilities were added to the CVE database in May and June, more than 11 times the prior year total of 297.

One vulnerability in SAP NetWeaver could enable unauthenticated attackers to take over applications using HTTP,[10] and a dozen critical and high-severity vulnerabilities found in OpenClinic GA could expose critical healthcare infrastructure to attacks.[11] This makes it even more important to prioritize vulnerabilities according to risk, as only 0.6% of CVEs are ever exploited in the wild.[12]

03 Application Vulnerability Trends

03 | Application Vulnerability Trends

For may–june 2020, contrast labs identified several application vulnerability trends from analysis of its aggregate data:

TREND: TOO MANY APPLICATIONS CONTINUE TO HAVE SERIOUS VULNERABILITIES

Similar to previous months, nearly every application (98%) had at least one vulnerability in May and June (Figure 1). That number is actually up slightly from the prior two-month period, which saw 97% of applications impacted. It is even higher than the 96% reported in Contrast's annual 2020 Application Security Observability Report, which covered the 12-month period ending May 31, 2020. While a 1% or 2% increase seems relatively small in this case, full percentage-point changes at volumes can be fairly dramatic.

But the larger concern is that one in three applications has at least one serious vulnerability—that is, defined by Contrast as posing either high or critical risk. This number fluctuates more than the total number of vulnerabilities; it is down from the 36% noted in March and April but much higher than the 26% identified in the annual report. It seems that this metric increased when developers started working from home in March—an environment that may increase the likelihood of serious vulnerabilities being introduced. While the number settled a bit during May and June—perhaps because coders became more accustomed to working from home—it is still up by 27% over the annual percentage.

Cross-site scripting (XSS) and broken access control continue to be the most common serious vulnerability categories, impacting 18% and 17% of applications, respectively (Figure 2)—down slightly from 20% and 19% in March and April. XSS vulnerabilities have now been present in 20% or less of applications for five straight months after being above 20% in January 2020 and throughout much of 2019. This is because current coding practices use application programming interfaces (APIs) and JavaScript in the browser instead of generating HTML on the server side. In other words, technology may be moving beyond this long-prominent vulnerability. In addition, languages and frameworks used by developers are doing more to protect against XSS vulnerabilities by default.

FIGURE 1

Percentage of applications with vulnerabilities and serious vulnerabilities.

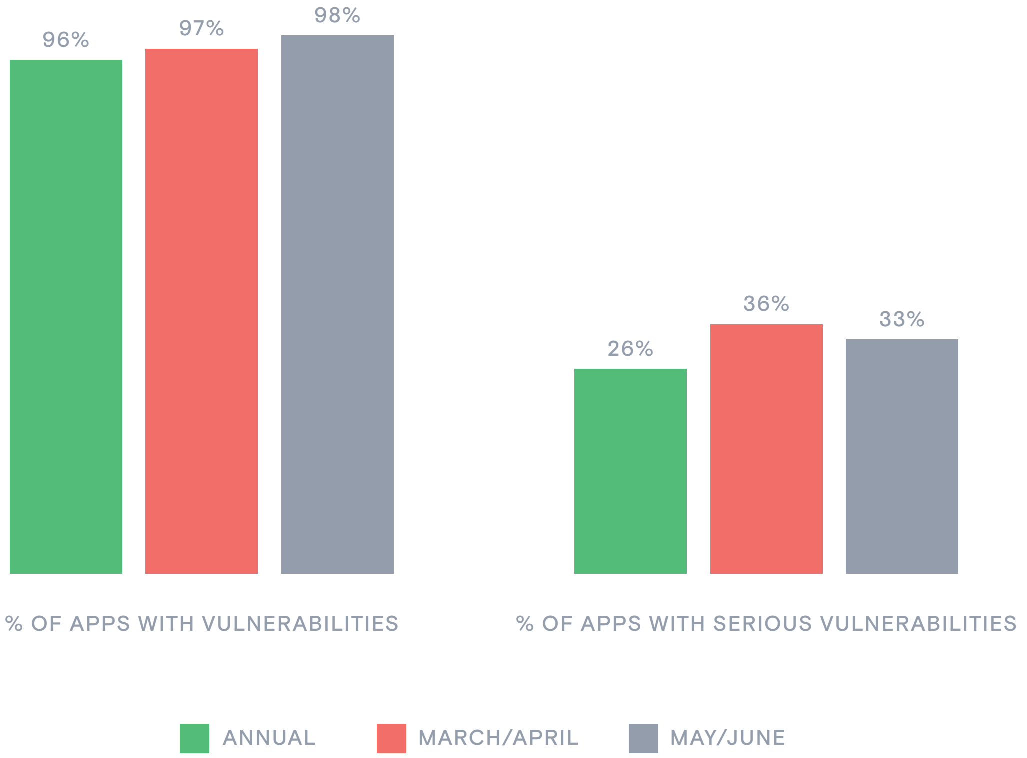
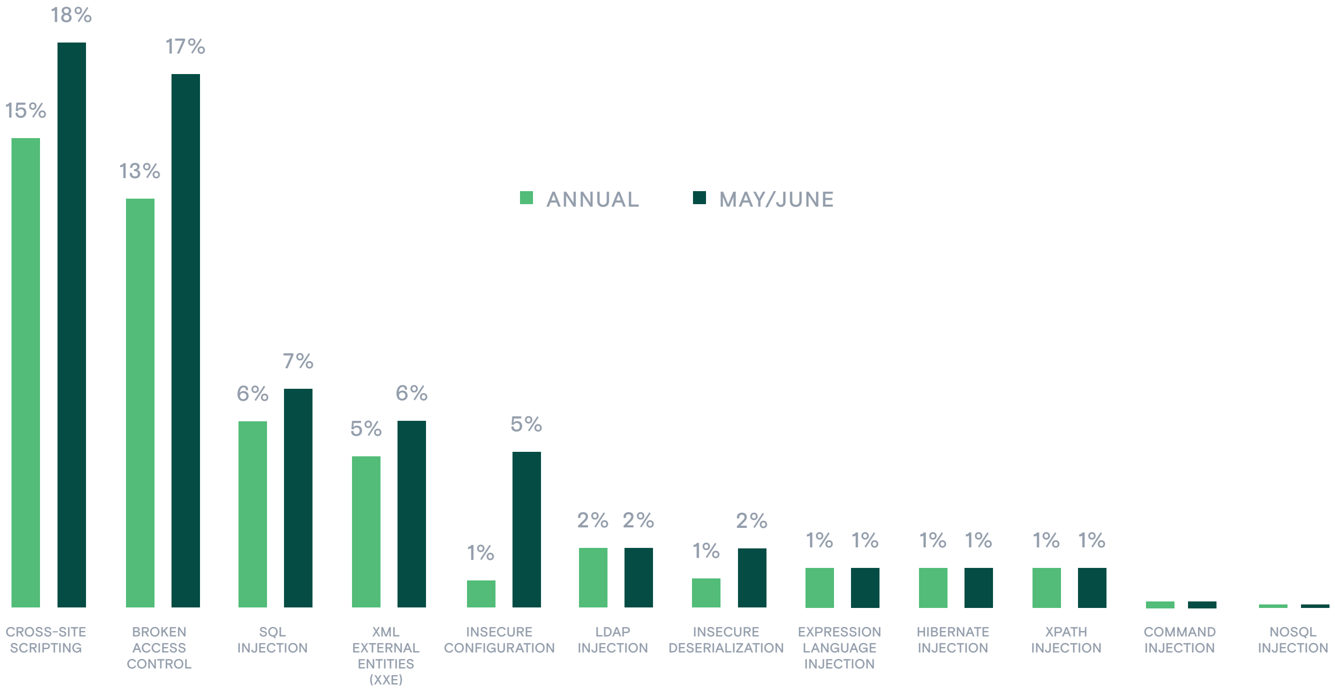


FIGURE 2

Percentage of applications with serious vulnerabilities May–June 2020, compared to 12-month averages.



TREND: MORE JAVA APPLICATIONS HAVE SERIOUS VULNERABILITIES

While Python seems poised to displace it,[13] Java has been the world's most commonly used programming language for many years, thanks largely to its portability, flexibility, and scalability.[14] But these same qualities that make it so attractive for businesses—especially large enterprises—also make it more susceptible to vulnerabilities. In May and June, 39% of Java applications contained serious vulnerabilities—compared with only 26% of .NET applications (Figure 3). Nearly one-quarter of Java applications (24%) have a broken access control vulnerability, and 23% have an XSS vulnerability—unchanged from March and April (Figure 4).

In contrast to the open world of Java development, Microsoft exercises stricter control over the .NET language, with fewer integrated development environments (IDEs), fewer open-source libraries and frameworks, and more standardization across the board. XSS and injection are the only two vulnerability categories to impact more than 10% of .NET applications.

These differences should be noted, not to disparage the use of Java but rather to aid development teams in delivering secure Java applications. Writing secure code is very possible with a variety of languages, if developers learn to do so by learning from feedback gleaned from continuous testing. With discipline and a commitment to write secure code, Java's higher propensity to vulnerabilities should not be a factor in whether to choose it or another language.

FIGURE 3

Percentage of applications with serious vulnerabilities by language, May–June 2020.

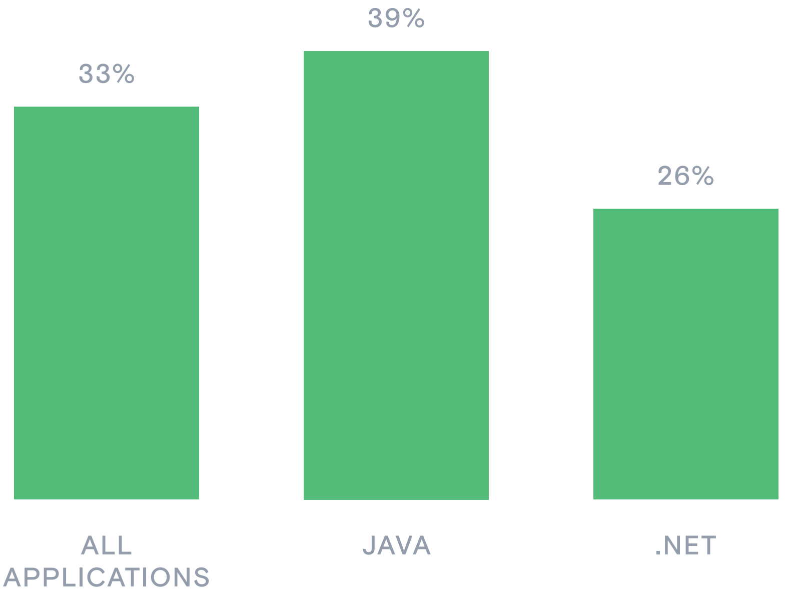
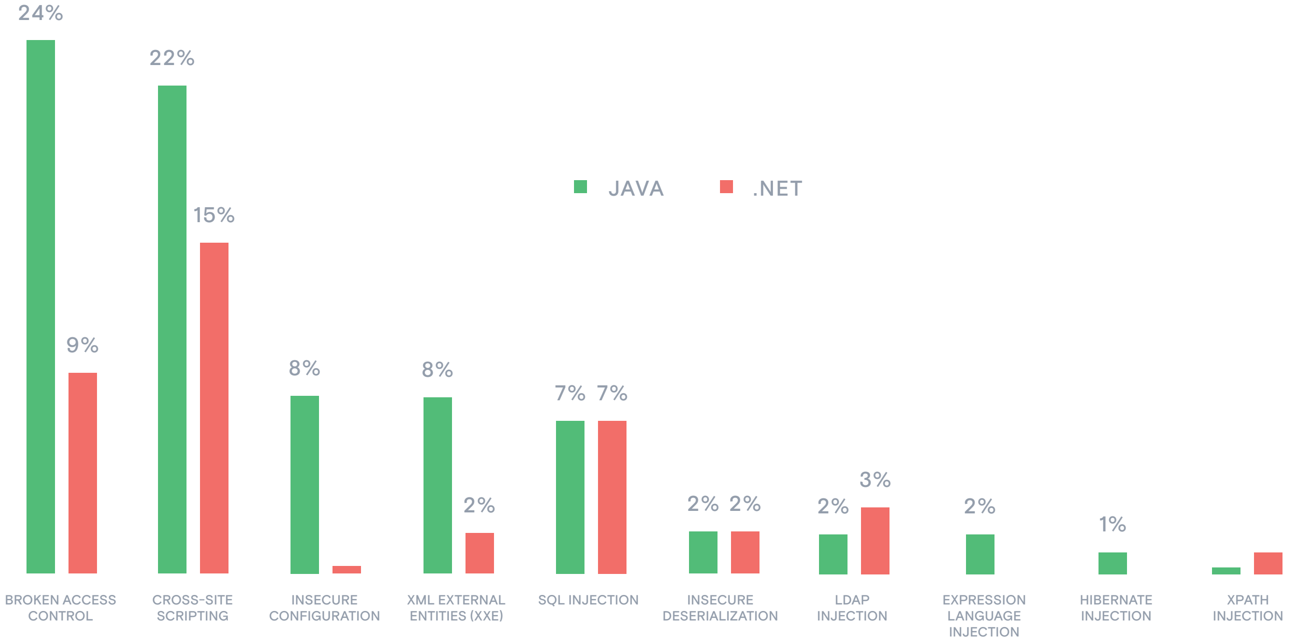


FIGURE 4

Percentage of applications with serious vulnerabilities, by vulnerability and language, May–June 2020.



TREND: VULNERABLE APPLICATIONS TEND TO HAVE MULTIPLE VULNERABILITIES

Another continuing trend is that a small subset of applications contains a large number of serious vulnerabilities. The mean number of serious vulnerabilities within vulnerable applications held steady at 55 in May and June. But when the numbers are broken down, only 5% of applications have 50 or more serious vulnerabilities (Figure 5). This handful of applications inflates the mean number, of course, and the median number of serious vulnerabilities per vulnerable application is only four.

Both numbers would be worse for these applications if they were protected by legacy AppSec tools. Line-by-line scanning using static application security testing (SAST), for example, can create significant alert fatigue in an application with many legitimate vulnerabilities—and the large number of false positives for which SAST solutions are notorious. A report by one SAST vendor showed an average of 175 identified vulnerabilities per application, many of which are likely false positives.[15]

When looking at vulnerability counts by category, applications having expression language injection vulnerabilities have the most instances of that vulnerability—a median of five. XSS has a median of four, and broken access control, SQL injection, and XML external entity (XXE) injection had medians of two each (Figure 6).

It makes sense in certain instances that multiple occurrences of the same vulnerability would be present in a particular application. For instance, if a developer misses validating a piece of data once, they could miss it in multiple places.

FIGURE 5

Count of serious vulnerabilities per vulnerable application, May–June 2020.

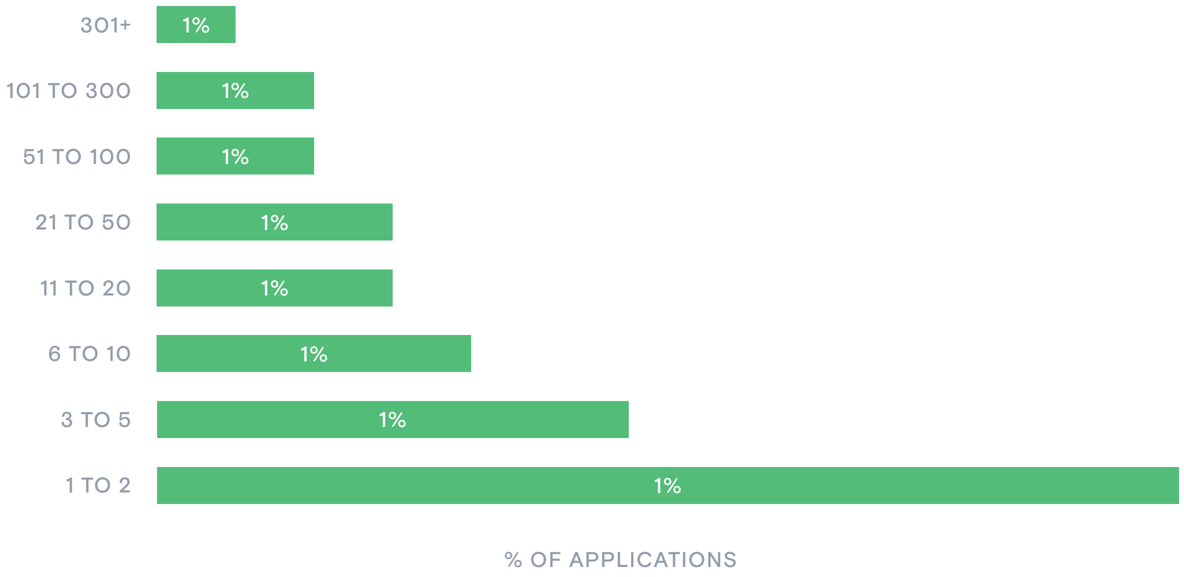
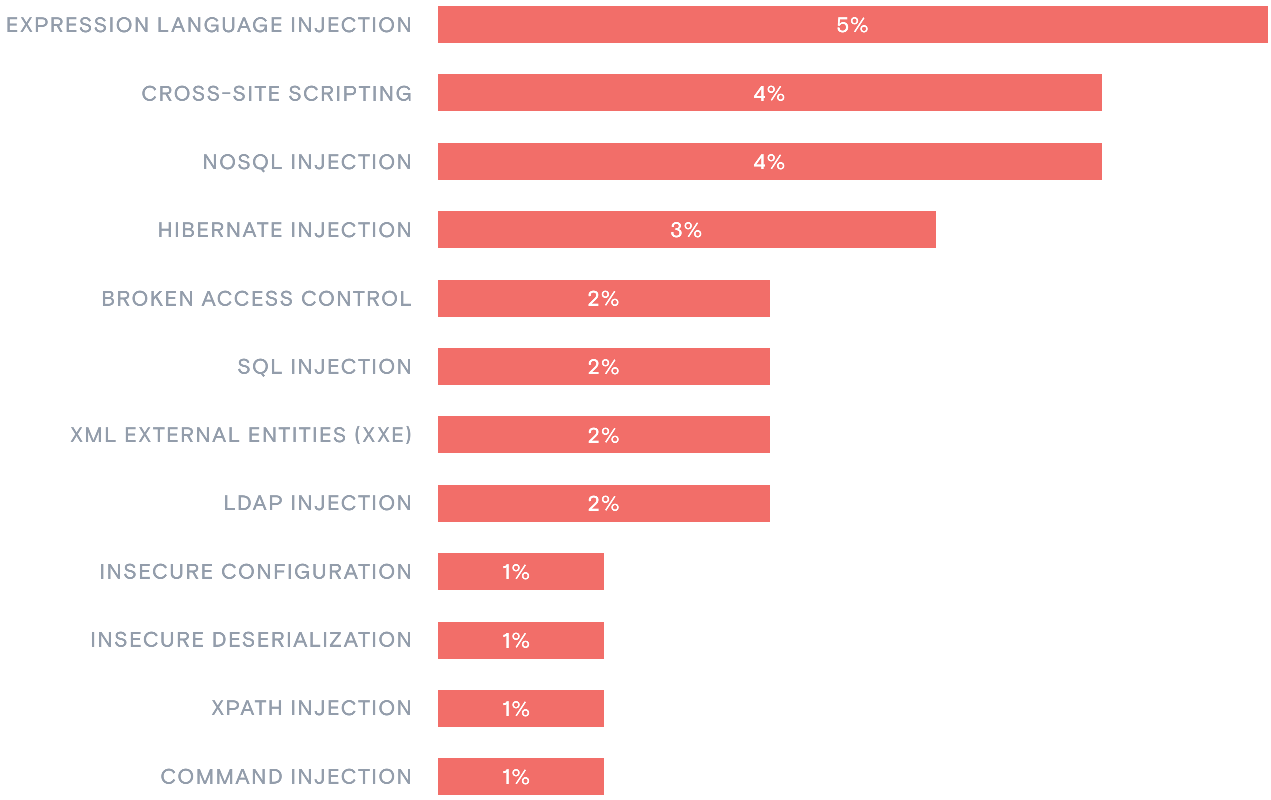


FIGURE 6

Median number of serious vulnerabilities per vulnerable application, May–June 2020.



04 Attack Trends

04 | Attack Trends

Data from contrast protect during may and june revealed a number of trends regarding attacks:

TREND: SQL INJECTION ATTACKS TRENDED SHARPLY UPWARD, BECOMING THE MOST COMMON ATTACK TYPE

The overall volume of attack attempts was down, with the average application experiencing just over 9,000 attacks per month—significantly lower than the nearly 13,500 in March and April. This may reflect more precise targeting on the part of cyber criminals. The percentage of attacks that hit an exploitable vulnerability—that is, targeted a vulnerability that actually existed in an application—held steady at 3%.

While the overall volume of attacks was down somewhat, the number of applications that saw SQL injection attacks was up sharply (Figure 7). This could be related to an influx in new applications being onboarded or a targeted concentration of attacks on SQL from bad actors. More than four in five applications (81%) saw an attack of this type in May and June, up from 55% in the prior two-month period—a 47% increase (Figure 8). This was enough to move SQL injection into first place as the most common attack type, while XSS moved from first to third on the list despite impacting 66% of applications—up from 63% in March–April.

While only 7% of applications had SQL injection vulnerabilities during May and June, the spike in this attack type is worrying because it may indicate that attackers are finding some success with the approach. Media reports have highlighted SQL injection vulnerabilities in applications in critical fields like remote learning16 and operational technology.[17] As businesses scrambled to accommodate remote employees and more digital engagement with customers during the COVID-19 pandemic, they may have deployed more of this vulnerability type into production.

The same data shows that two other vulnerability categories surged in May and June as compared with March and April. Broken access control spiked from 48% to 72% of applications, moving this category into second place in the list of most common attacks. Command injection increased from 46% to 63% of applications. Again, cyber criminals may be deploying more of these attacks in hopes of a successful infiltration at a time when development activity is spiking. Some broken access control attacks are easy to automate and deploy to multiple applications, making them a low-cost attack with medium to high potential return.

Of course, organizations can avoid problems associated with these attacks by making it a priority to deploy secure code into production—something that is possible with an instrumentation-based approach to application security. Such solutions provide continuous vulnerability data to the developers as they write code, while protecting an application in production from within the software.

FIGURE 7

Percentage change in number of applications targeted by attacks by rule, March–April vs. May–June 2020.

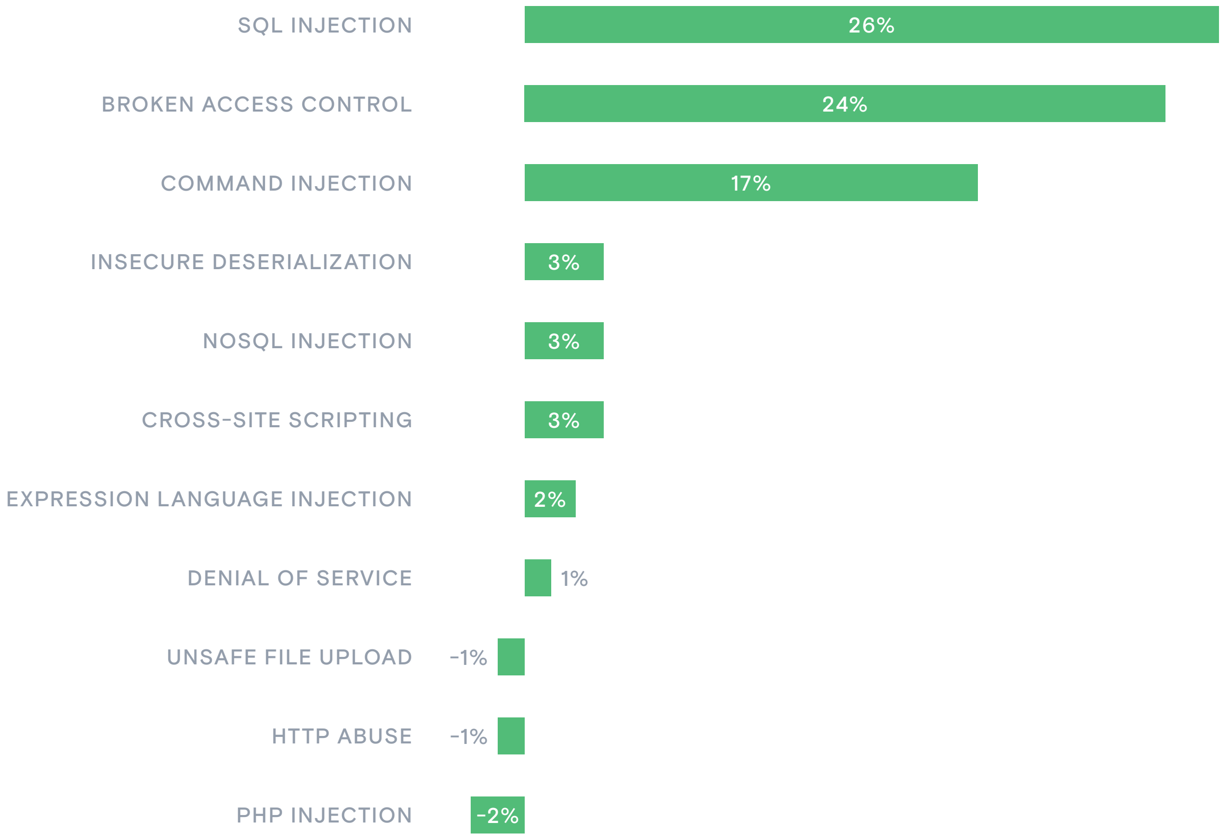
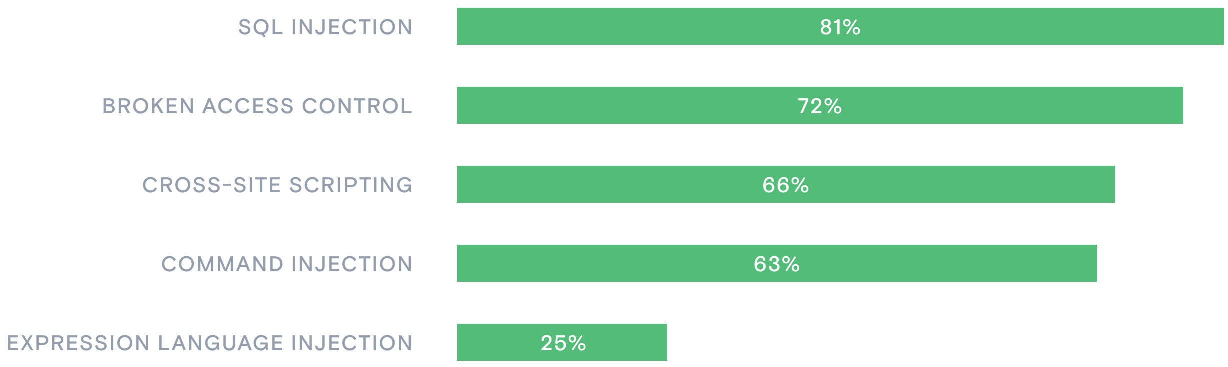


FIGURE 8

Percentage of applications targeted for the top 5 attack types, May–June 2020.



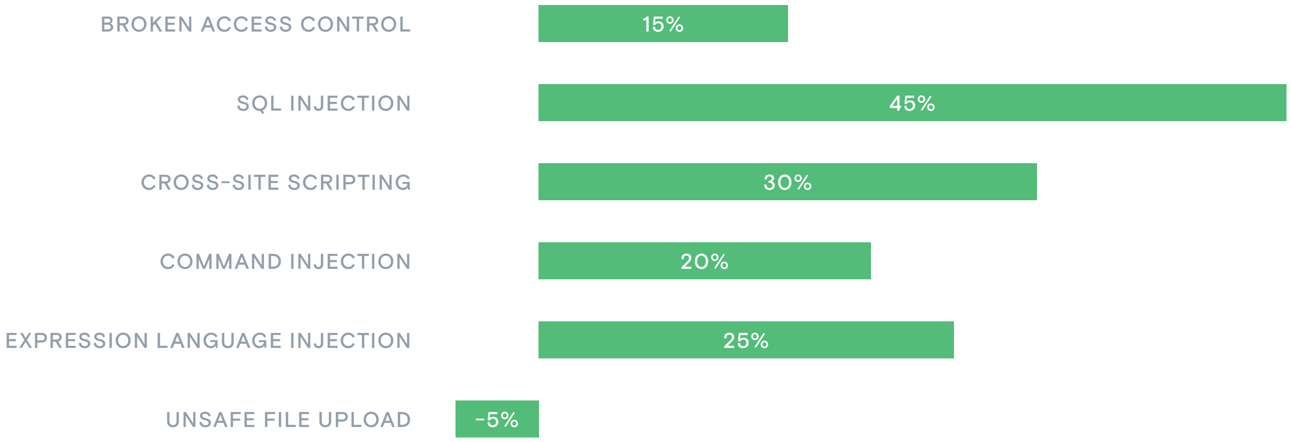
TREND: SPECIFIC ATTACKS IMPACTED MANY MORE .NET APPLICATIONS

As noted above, the total volume of attacks was down in May and June compared with earlier months, but a larger percentage of applications were impacted by several specific attack types. Both trends were especially true with .NET applications. For applications in that language, five vulnerability categories—broken access control, SQL injection, XSS, command injection, and expression language injection—saw double-digit increases in the percentage of applications impacted (Figure 9). Three of these five saw increases of between 25 and 40 percentage points.

This bimonthly period saw an interesting combination of fewer attacks per application per month with increases in the percentage of applications on the receiving end of many attack types. This difference was especially pronounced with .NET applications, which only saw declines in one attack type. This suggests that adversaries are targeting a broader array of applications, which are also seeing a greater number of vulnerabilities targeted. This might be due to attackers trying to exploit all of the additional code that is being deployed faster due to more aggressive business release cycles during COVID-19.

FIGURE 9

Percentage increases in number of applications targeted by attacks by rule, .NET applications, March–April vs. May–June 2020.



05 Application Security Watch List

05 | Application Security Watch List

When development and security professionals look at vulnerability and attack data such as what is presented in this report, their most important consideration is how much risk each vulnerability poses to their organizations. Contrast Labs performs continuous analysis of these two datasets to compile an Application Security Watch List for each bimonthly period. The ranking is according to something we call a “risk factor”—the comparison of the likelihood that a vulnerability type will occur and the likelihood that that specific vulnerability will be attacked. The Watch List for May and June of 2020 is as follows:

FIGURE 10

Top 12 vulnerability categories by risk factor, May–June 2020.

1. SQL INJECTION
2. BROKEN ACCESS CONTROL
3. COMMAND INJECTION
4. CROSS-SITE SCRIPTING
5. EXPRESSION LANGUAGE INJECTION
6. INSECURE CONFIGURATION
7. XML EXTERNAL ENTITIES (XXE)
8. DENIAL OF SERVICE
9. CLASSLOADER OVERWRITE
10. NOSQL INJECTION
11. UNSAFE FILE UPLOAD
12. INSECURE DESERIALIZATION

Given the 47% spike in the percentage of applications targeted by SQL injection attacks, it is not surprising that this vulnerability moved to the top of the Watch List for May and June (Figure 11). Its vulnerability prevalence is up by 15% in that period compared with the annual average for the 12 months ending on May 31. SQL injection attacks were especially impactful in May, when 85% of applications experienced them.

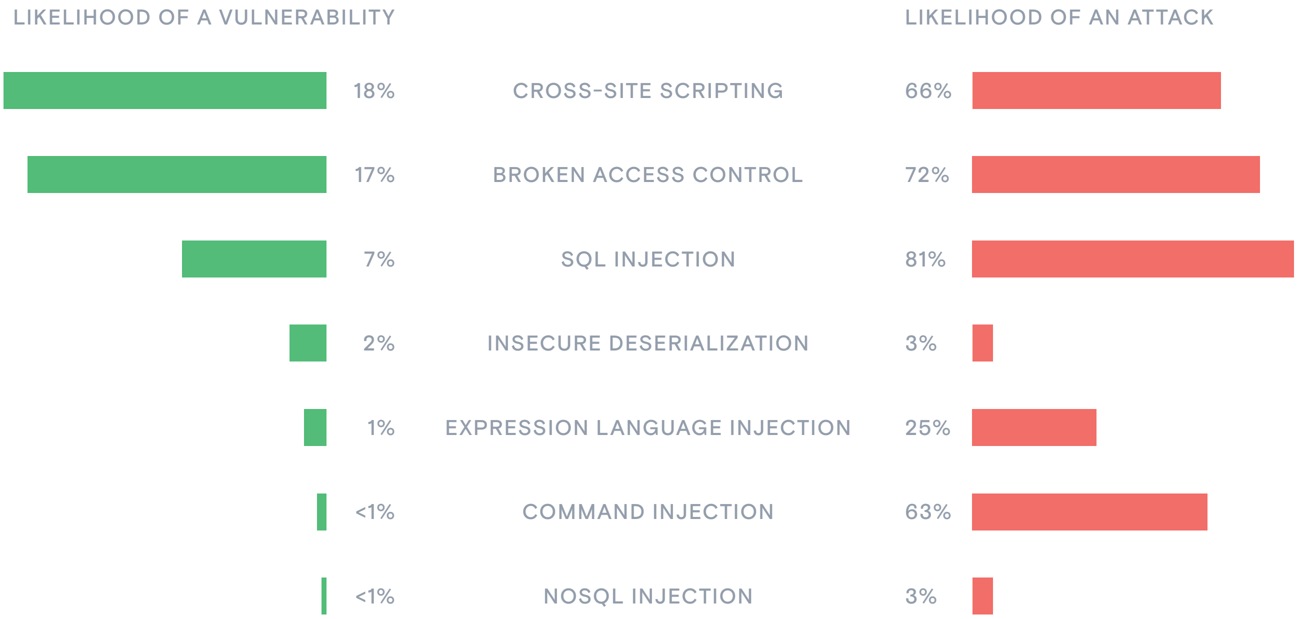
Indeed, SQL injection attacks can yield useful information when they are successful, including login credentials, transaction files, and database information. For example, in late May, a New York City resident was charged with using SQL injection to hack into ecommerce sites, steal payment card information, and sell it on the dark web.[18]

Several familiar vulnerability types follow behind SQL injection, including broken access control, command injection, XSS, and expression language injection. Indeed, the top five entries on the Watch List are unchanged from March and April, and these vulnerabilities remain common targets for adversaries.

Insecure configuration moved up one spot to number six on the Watch List, driven by a 79% increase in vulnerability prevalence compared with the annual average. And further down the list, NoSQL injection moved up two spots from its customary 12th position on the list. As organizations deal with increasingly large sets of distributed data, NoSQL databases are more attractive to make sense of it.[19] As a result, NoSQL injection vulnerabilities may be targeted with more attacks going forward.

FIGURE 11

Likelihood of a vulnerability vs. likelihood of an attack, May–June 2020.



06 Conclusion

06 | Conclusion

This bimonthly report finds that software vulnerabilities and attacks continue unabated in a rapidly changing technology world. Given the rapid clip at which businesses have developed applications over the past two months—often in order to survive in a rapidly changing economy—the fact that vulnerabilities have not dramatically increased could be counted as good news.

Conversely, the decline in the volume of attacks compared with the prior two-month period might appear at first glance to be a positive development. Unfortunately, the notion that cyber criminals are simply targeting their attacks more precisely is likely closer to the truth. The dramatic increase in applications impacted by specific attack types may indicate some level of strategy behind the attacks.

In a fast-changing world, it is more important than ever to ensure that applications are delivered into production without serious vulnerabilities. To achieve this, organizations need to "shift left" with their AppSec processes—finding vulnerabilities earlier in the development process.[20] At the same time, organizations must "shift right" by building self-protection into applications in production.[21] Security instrumentation enables continuous security testing built into the application itself, providing real-time feedback to developers and enabling remediation on the fly.

[1] James Kobielus, "Coding together apart: Software development after COVID-19," InfoWorld, April 9, 2020.

[2] "OpsRamp Survey Shows IT Spending Remains Strong, With Focus on Minimizing Business Risk during COVID-19," GlobeNewswire, April 21, 2020.

[3] "How cybercriminals are taking advantage of Covid-19," Silicon Republic, May 26, 2020.

[4] Joe Tidy, "Google blocking 18M coronavirus scam emails every day," BBC, April 17, 2020.

[5] Ido Safruti, "5 Ways Web Attacks Will Change Post-COVID," SC Media, July 8, 2020.

[6] Ryan Lucas, "DOJ Charges 2 Suspected Chinese Hackers Who Allegedly Targeted COVID-19 Research," NPR, July 21, 2020.

[7] Akshaya Asokan, "Suspected Hacker Faces Money Laundering, Conspiracy Charges," GovInfoSecurity, May 30, 2020.

[8] Liam Tung, "Open-source security: This is why bugs in open-source software have hit a record high," ZDNet, March 13, 2020.

[9] Todd Schell, "July 2020 Patch Tuesday forecast: Will the CVE trend continue?" Help Net Security, July 10, 2020.

[10] "Critical Vulnerability Hits SAP Enterprise Applications," Dark Reading, July 14, 2020.

[11] Eduard Kovacs, "Vulnerabilities in Popular Open Source Management Tool Expose Hospitals to Attacks," SecurityWeek, July 9, 2020.

[12] Roger A. Grimes, "Are zero-day exploits the new norm?" CSO, February 21, 2019.

[13] Liam Tung, "Programming language popularity: Python overtakes Java-as Rust reaches top 20," ZDNet, July 28, 2020.

[14] Ben Putano, "A Look At 5 of the Most Popular Programming Languages of 2019," Stackify, August 30, 2019.

[15] "State of Software Security Report 2019," Veracode, October 22, 2019.

[16] Robert Lemos, "Researchers Find Vulnerabilities in Popular Remote Learning Plug-ins," Dark Reading, April 30, 2020.

[17] Eduard Kovacs, "Hackers Can Target Rockwell Industrial Software With Malicious EDS Files," SecurityWeek, May 22, 2020.

[18] John Leyden, "New Yorker charged in e-commerce cybercrime, bitcoin laundering scam," The Daily Swig, May 29, 2020.

[19] Avantika Monnappa, "The Rise of NoSQL and Why it Should Matter to You," Simplilearn, December 3, 2019.

[20] Jakob Pennington, "Shifting Left: DevSecOps as an Approach to Building Secure Applications," Medium, July 18, 2019.

[21] Alan Shimel, "DevOps Chat: Shifting Security Left and Right, With Contrast Security," Security Boulevard, October 7, 2019.

240 3rd Street
2nd Floor
Los Altos, CA 94022
Phone: 888.371.1333
Fax: 650.397.4133

Contrast
SECURITY

contrastsecurity.com