Contrast SECURITY

MARCH-APRIL 2020

Contrast Labs Application Security Intelligence Bimonthly Report

Table of contents

01 Executive Summary

01 | Executive Summary

Contrast Security's Application Security Intelligence Report for March-April 2020 leverages aggregate data collected by Contrast Assess and Contrast Protect for insights around both application vulnerabilities and targeted attacks. General findings of the bimonthly report include:

- While almost all applications have at least one vulnerability, nearly one-third have one or more serious vulnerabilities that require attention.
- Injection-based vulnerabilities (e.g., LDAP, SQL) had some of the highest number of serious vulnerabilities by rule per application. Plus, the frequency of command injection and expression language (EL) injection attacks both increased in March and April.
- While cross-site scripting (XSS) vulnerabilities may be decreasing in seriousness by some estimations, Contrast Labs' analysis shows the conditions for a successful breach still exist. Nearly half (46%) of applications are probed for XSS vulnerabilities. One in five (20%) have at least one XSS vulnerability in their code.

KEY FINDINGS

75

Applications with one or more vulnerabilities have an average of 75 vulnerabilities—and 55 of those are serious vulnerabilities

13%

Command injection and expression language (EL) injection attacks both increased since the last report (growing by 13% and 16% respectively)

63%

Nearly two-thirds of applications (63%) were probed for broken access controls, and 9% of the time they successfully touched a vulnerability in the code

46%

Nearly half (46%) of attacks targeted cross-site scripting (XSS), with 6% success rate of finding an XSS vulnerability—heightening the risk of a breach

Contrast SECURITY

02 Changing Business Models and New Vulnerabilities in 2020

02 | Changing Business Models and New Vulnerabilities in 2020

Contrast Labs' "Bimonthly Application Security Intelligence Reports" provide an update on the status of application security as observed by vulnerabilities and attacks pinpointed by telemetry from customer applications. 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, identifying actionable insights to aid developers and security teams as they refine their application security strategy. It is the only report in the industry that combines insights about vulnerabilities, library issues, and attacks in a single report.

By all accounts, 2020 is shaping up to be a historic year in terms of challenges and changes for businesses—across all business units and in virtually every sector. The impact of the COVID-19 pandemic has obligated major adaptations to enable remote operations, IP-based communications, and greater-than-ever reliance on web applications. Nearly three-quarters (73%) of IT operations and DevOps team leaders expect to either accelerate or maintain digital transformation initiatives and projects—indicating the value of digital products and services in an era of social distancing.[1]

At the same time, many cyber criminals have already accelerated their efforts to take advantage of potential opportunities for exploitation during the pandemic onset.[2] And the appearance of new vulnerabilities is only adding to the problems of the first third of the year. For example, Microsoft has seen a 44% jump in the number of vulnerabilities patched between January and April 2020, compared with the same period in 2019.[3] But the unfortunate reality is that an enterprise typically patches known Common Vulnerabilities and Exposures (CVE) vulnerabilities 85 days after publication, leaving hackers plenty of time to launch a successful attack.[4]

03 Application Vulnerability Trends

03 | App Lication Vulnerability Trends

For march-april 2020, we identified several trends from our analysis of data from contrast labs:

TREND: ALMOST EVERY APPLICATION HAS AT LEAST ONE VULNERABILITY.

During March and April 2020, 97% of applications had at least one vulnerability and nearly one-third (32%) had one or more serious vulnerability. On average, applications included two unique types of serious vulnerabilities (rule violations) during the two-month window. It should not come as a surprise that bad actors are increasingly taking advantage of the nearly ubiquitous opportunities that applications offer for exploitation. According to Verizon's "2020 Data Breach Investigations Report," nearly half (43%) of all successful data breaches can be traced back to an application vulnerability—a share that more than doubled year over year.[5]

Looking at month-to-month numbers for the year, applications with at least one serious vulnerability decreased from 39% in January down to 29% in March. In April, they rose back up to 35%. These fluctuations are mostly caused by the natural ebb and flow of development cycles (e.g., new projects, updates, fixes), which vary from month to month.

Contrast finds vulnerabilities by watching the ways an application is used throughout the development cycle: development, testing of the unit and system, quality assurance (QA), user acceptance (UA), and in production.

THE RISKS OF XSS?

An XSS attack hijacks HTML pages, deceives users, and steals sensitive data as it assumes control, redirects links, and rewrites site content. When a cyberattack identifies a vulnerability, malicious XSS code (e.g., JavaScript) is injected into the website.

The script then prompts a user to interact with it—typically through a link or by storing the code in the website to be later viewed by an unsuspecting victim.

Notwithstanding, while these attacks are extremely prevalent, XSS has slowly moved down on the OWASP Top 10 list. Some data suggests that risks associated with XSS may be lower on average than other CVEs.[7]

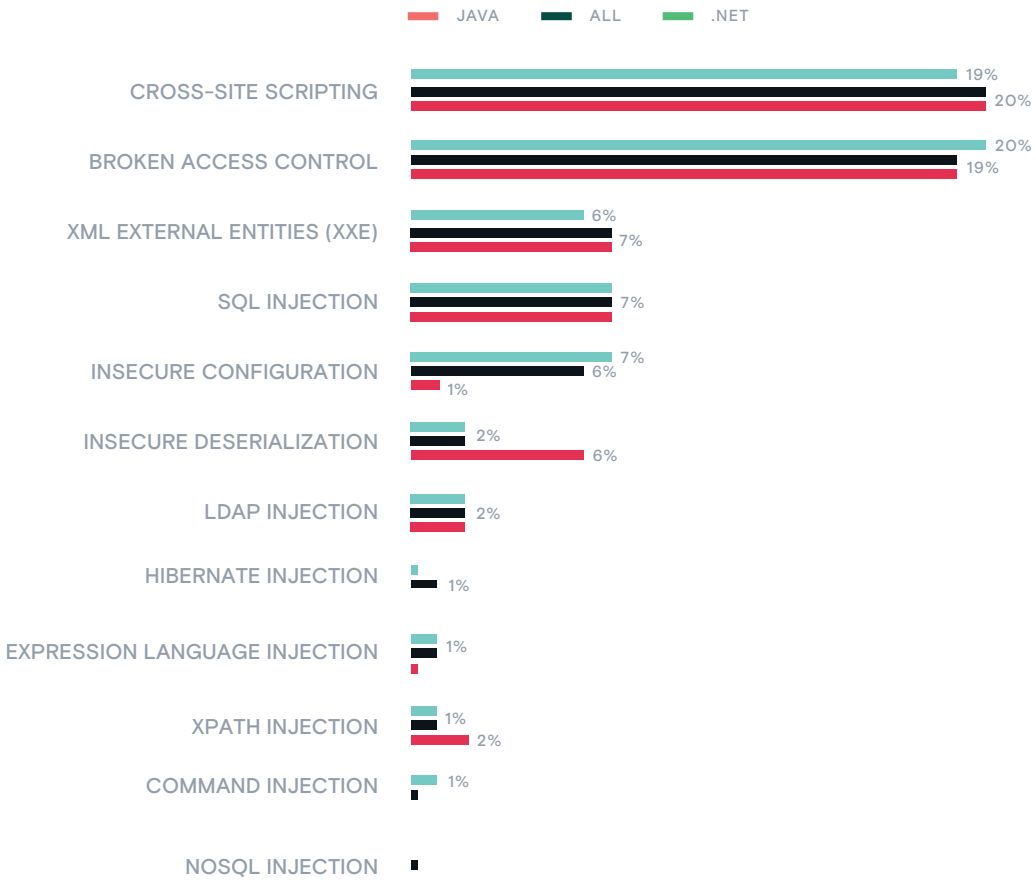
Throughout the software development life cycle (SDLC), there are points where vulnerabilities are more likely to be produced and/or reported. Over the months of March and April, Contrast found that new applications were 3x more likely not to have a serious vulnerability reported. There could be many reasons for this—including fewer lines of code (fewer opportunities for introducing vulnerabilities) and no completion of formal testing (fewer routes exercised over time means fewer vulnerabilities reported).

TREND: CROSS-SITE SCRIPTING (XSS) CONTINUES TO BE THE TOP SERIOUS VULNERABILITY DURING THE FIRST FOUR MONTHS OF 2020.

The most commonly seen serious vulnerabilities by Contrast consisted of XSS and broken access control vulnerabilities. In March and April combined, XSS vulnerabilities were reported in 20% of applications, broken access control in 19%, SQL injection in 7%, and XML external entities (XXE) in 7%. XSS attacks could allow an attacker to hijack the user's session and take over the account. Other damage can include phishing attacks, disclosure of end-user files, installation of Trojan horse programs, or redirection of users to a malicious URL.[6]

FIGURE 1

Percent of applications with reported serious vulnerabilities, March–April 2020.



Month by month, XSS vulnerabilities dipped to 16% in March (down from 23% in January 2020). XSS vulnerabilities increased again to 20% in April. Similar ranges of variation can be seen in all the top vulnerability categories in each of the first four months of 2020. Beyond the number of vulnerabilities found, XSS-targeted attacks showed activity during the period. Most notably, several vulnerable WordPress plugins were exploited in widescale attacks against more than 900,000 WordPress websites in April and May of this year. Some of the successfully compromised sites redirected visitors to “malvertising” (the use of online advertising to spread malware).[8]

Broken access controls were the second most frequently seen vulnerability during the bimonthly period. According to OWASP, exploitation of broken access controls (including path traversal vulnerabilities and cross-site request forgeries) is a core skill of cyber criminals. Attackers can exercise administrator privileges, take actions as users with restricted functions, or tamper with records.[9]

TREND: INJECTION-BASED VULNERABILITIES REMAIN A SIGNIFICANT PROBLEM BY VOLUME.

In March and April, vulnerable applications had an average number of 75 vulnerabilities—and the vast majority of those vulnerabilities (55) were serious. When looking at vulnerability types (by rule), EL injection and NoSQL injection had the highest median number of serious vulnerabilities per application. This means that 50% of applications that contained the vulnerability have more or less than the median number of vulnerabilities.

FIGURE 2

The median number of vulnerabilities (by rule) per vulnerable application.

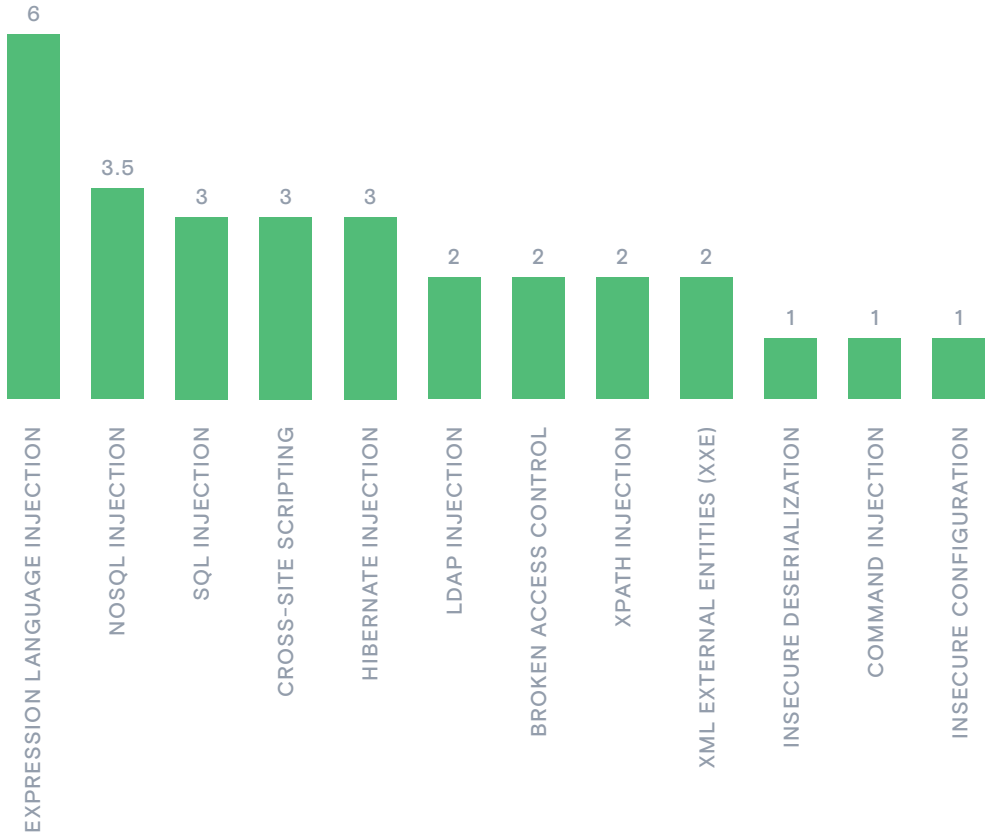
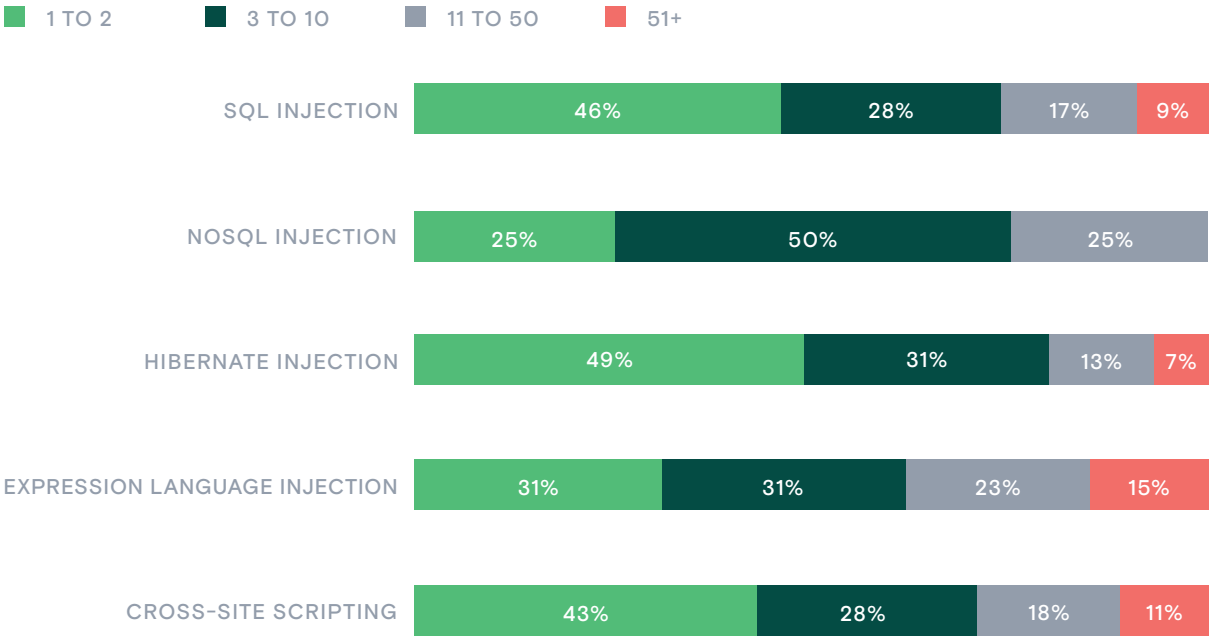


FIGURE 3

Distribution of serious vulnerabilities with the top five medians.



Injection vulnerabilities have been a problem for the last 20 years and come in a variety of types (e.g., EL, SQL, NoSQL, XPath, LDAP, Command, Hibernate). According to OWASP, injection attacks occur when untrusted data is sent to an interpreter as part of a command or query.[10] The malicious data can then trick the interpreter into executing unintended commands or accessing data without authorization.[11] Specifically, OWASP ranks injection vulnerabilities as a Top 10 security problem for both web applications and application programming interfaces (APIs).

But successful injection attacks can have far-reaching consequences across an organization—beyond application security alone. An SQL injection vulnerability in the management interface of the Sophos XG firewall was recently used to successfully exfiltrate user data (usernames, passwords, local device admins).[12]

04 Attack Trends

04 | Attack Trends

For march-april 2020, individual applications received an average of 13,429 attacks each month (slightly below the year-to-date average of 14,216). Contrast labs identified several attack trends in its aggregate data.

TREND: COMMAND INJECTION AND EL INJECTION ATTACKS TARGETING APPLICATIONS HAVE BOTH RISEN OVER THE LAST TWO MONTHS..

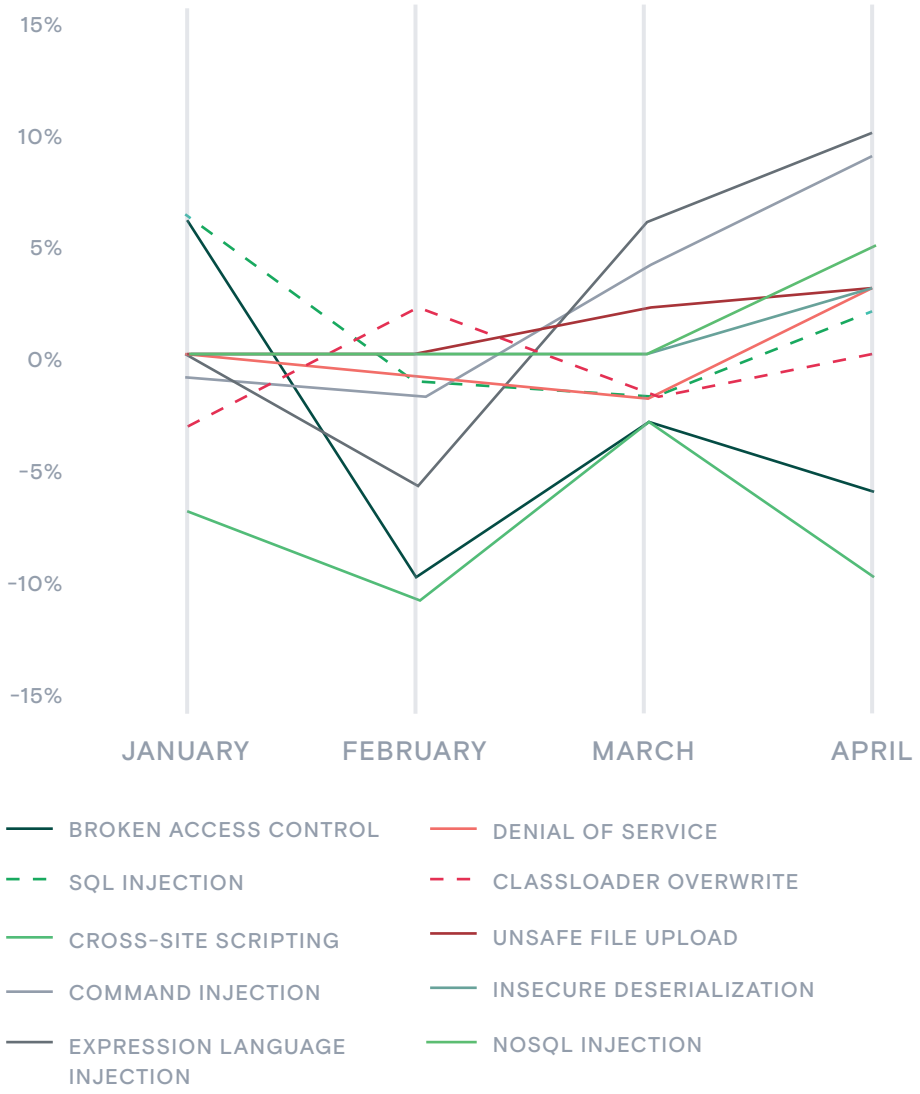
A greater portion of applications were targeted by command injection attacks than in the previous two months. The percentage increased in March to 51% and to 60% in April (a 13% total increase since February 2020). Command injection represents a significant risk to application security. These attacks, if successfully exploited, can execute arbitrary commands on a host operating system (OS) via the vulnerable code. These attacks become possible when an application passes unsafe user-supplied data (e.g., forms, cookies, HTTP headers) to a system shell. The attacker's OS commands are then typically executed using the vulnerable application's privileges.[13]

Serious command injection attacks continue to exploit application security—as well as network device firmware,[14] Internet-of-Things (IoT) firmware,[15],[16] and even basic physical security systems like CCTV.[17] GitHub recently patched a critical command injection issue on its website as a result of its million-dollar bug bounty program. The vulnerability existed because the branch names were not correctly sanitized in the Mercurial import feature. The root cause was ultimately found to be an outdated dependency.[18]

Command injection is an ongoing threat. For example, last fall, cybersecurity researchers uncovered hackers who were exploiting two critical remote command injection vulnerabilities using a web-based exploit that runs code on the host in order to eavesdrop on network traffic and install backdoors in enterprise-grade networking devices.[19]

FIGURE 4

Change in percent of applications targeted by attack type January–April.



Applications targeted by EL injection attacks saw a 16% increase between February 2020 (seen in just 12% of applications) and April 2020 (targeted 28% of applications). EL injection allows an attacker to view server-side data and other configuration details and variables (e.g., sensitive code, passwords, database queries). The attack takes advantage of server-side code injection vulnerabilities that occur whenever an application incorporates user-controllable data into a string that is dynamically evaluated by a code interpreter. If the user data is not strictly validated, an attacker can substitute input that modifies the code that will be executed by the server.

EL injection attacks are very serious, as they can lead to complete compromise of the application's data and functionality, not to mention the server hosting the application. Cyberattacks can also use the server as a platform for further attacks against other systems.[20] Adobe recently patched a number of application vulnerabilities, including XSS and EL injection issues, that had the potential to expose sensitive information.[21]

Several other attacks appeared in March and April. These include: **unsafe file upload** (seen in 5% of applications; successful attacks can have far-ranging consequences), **NoSQL injection** (seen in 5% of applications; targeting non-SQL databases), and **insecure deserialization** (seen in 3%; these can lead to remote code execution attacks). For NoSQL and unsafe file load, March and April were the highest occurrences in the last 12 months. This fluctuation is expected, as we generally see these attack types come and go.

05 Application Security Watch List

05 | Application Security Watch List

For development and security teams, the most important takeaway when looking at application vulnerability and attack data is how to calculate risk to an organization. Here, Contrast Labs performs continuous analysis of these two datasets to compile an Application Security Watch List for each bimonthly period. This list is based on a comparison of the likelihood that a vulnerability will occur and the likelihood that the specific vulnerability will be attacked.

FIGURE 5

The top twelve vulnerabilities by risk factor for March–April 2020. This list ranks the prevalence of vulnerabilities and likelihood of an attack for the period.

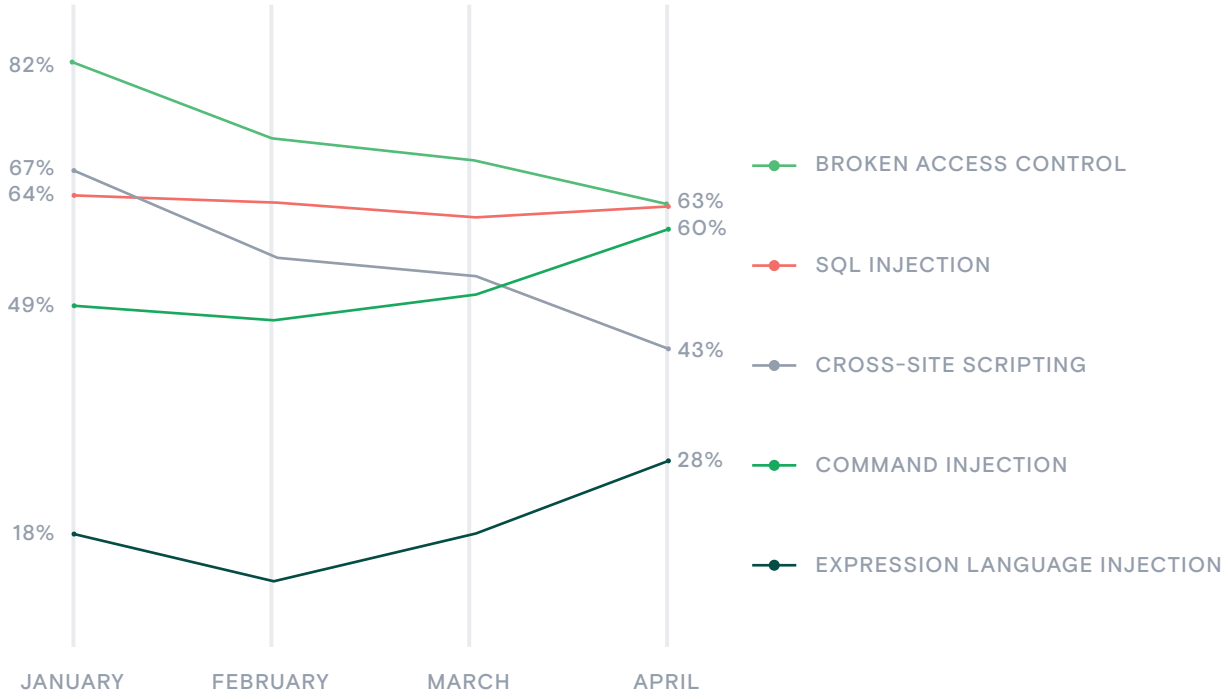
1. BROKEN ACCESS CONTROL
2. SQL INJECTION
3. CROSS-SITE SCRIPTING (XSS)
4. COMMAND INJECTION
5. EXPRESSION LANGUAGE (EL) INJECTION
6. XML EXTERNAL ENTITIES (XXE)
7. INSECURE CONFIGURATION
8. DENIAL OF SERVICE
9. CLASSLOADER OVERWRITE
10. UNSAFE FILE UPLOAD
11. INSECURE DESERIALIZATION
12. NOSQL INJECTION

While XSS was the top vulnerability to watch for January–February 2020, **broken access control** and **SQL injection** vulnerabilities both jumped ahead for the March–April period. Broken access control allows attackers to bypass authorization safeguards and perform tasks as if they were privileged users. This includes unauthorized access to functionality and data. This can include access to other users' accounts, sensitive files, the ability to modify user data, and changing access rights.[22]

SQL database injections can give an attacker unauthorized access to valuable data such as username and password information, transaction files, or database tables with the ability to compromise an entire database server through command execution.[23] Recent breach examples include a hacker using SQL injection to steal and leak the usernames and passwords of nearly 23 million players of an online children's game.[24]

FIGURE 6

Percent of applications with top 5 vulnerabilities targeted across the first four months of 2020.



Contrast Labs analysis differentiates between effective attacks and ineffective probing. For example, while nearly half (48%) of applications were probed for command injection vulnerabilities, a full 100% of the attacks Contrast observed in March and April did not touch vulnerable code. Similarly, 63% of applications were targeted for broken access control exploitation, but 91% of those were unsuccessful because these kinds of vulnerabilities were not present. And while 46% of applications saw XSS probes, they only made contact with vulnerable code 6% of the time. This means that 94% of the time those XSS attacks could generate a false-positive alert—that is, if security did not understand that a corresponding vulnerability was not present.

However, there are two particular areas where both the frequency attack probes and the success rate of attacks touching vulnerable code represent significant risk to the organization:

- Broken access control: Nearly two-thirds (63%) of all applications were targeted by broken access control attacks; 9% of the time, these attacks successfully found a broken access control vulnerability.
- XSS: Almost half (46%) of all applications were targeted by XSS attacks, compounded by a 6% success rate that one of these attacks touches vulnerable code.

Attacks with 9% and 6% success rates represent a significant amount of risk, particularly when they are the top two attack types in terms of likelihood to target an application for the period. In comparison, while SQL injection attacks were also frequent in March–April, these reached vulnerable code less than 1% of the time.

When data shows both that specific types of serious vulnerabilities are present and that attackers are increasing their efforts to exploit those vulnerabilities, these should be areas of focused risk management in the near term. Broken access control and XSS vulnerabilities currently pose heightened risks of leading to successful exploitation. Developers should either remediate these types of vulnerabilities and/or protect them against attacks in production with effective application security, such as a runtime application self-protection (RASP) solution. Development and security teams need tools that allow them to focus attention on the <10% of attacks that actually pose a significant threat, rather than playing whack-a-mole against every single attack type hitting their applications.

FIGURE 7

Likelihood of a vulnerability vs. likelihood of an attack, March–April 2020.

06 Conclusion

06 | Conclusion

Findings within this bimonthly report show that—regardless of whatever else is happening around the globe—application vulnerabilities and attacks continue to be a problem for both developers and their customers. While the data from March and April may not reveal any dramatic discoveries since the previous report, data clearly shows that bad actors are continuing to do whatever they can with whatever opportunities present themselves.

With an average application seeing thousands of attacks each month targeting serious vulnerabilities (including current popular attacks by command injection and EL injection), organizations must proactively look at ways to reduce the chances of exploitation. And as the attack data shows, DevOps teams should focus their remediation efforts on broken access control and XSS vulnerabilities—ensuring that application security extends from development through production.

To manage application security risks, organizations need to "shift left" with their security processes.[25] This starts by integrating security testing into every step of the development process. Security testing must be built directly into applications so that they execute any time an application is running. At the same time, organizations must also "shift right" to protect applications in production.[26] Similar to securing development cycles, the best way to mitigate risk (while reducing the noise of false positives and false negatives) is to build continual testing into running applications in production.

[1] "OpsRamp Survey Shows IT Spending Remains Strong, With Focus on Minimizing Business Risk during COVID-19," GlobeNewswire, April 21, 2020.
[2] Emma Woollacott, "Cybersecurity And COVID-19: The First 100 Days," Forbes, May 5, 2020.
[3] Kelly Sheridan, "Microsoft Patches 113 Bugs, 3 Under Active Attack," Dark Reading, April 14, 2020.
[4] Simon Roe, "Too Many Vulnerabilities, Too Little Time," InfoSecurity, May 29, 2020.
[5] "2020 Data Breach Investigations Report," Verizon, May 2020.
[6] "Cross Site Scripting (XSS)," OWASP, accessed June 11, 2020.
[7] David Lindner, "Contrast Labs: Mapping Risk Profiles for Select OWASP Top 10 Vulnerabilities to Understand Their AppSec Risk," Contrast Security, May 19, 2020.
[8] Ionut Ilascu, "Massive campaign targets 900,000 WordPress sites in a week," BleepingComputer, May 5, 2020.
[9] "A5:2017-Broken Access Control," OWASP, accessed June 4, 2020.
[10] Jeff Williams, "Injection Theory," OWASP, accessed June 15, 2020.
[11] "OWASP Top 10 Application Security Risks – 2017," OWASP, accessed June 4, 2020.
[12] "Hackers exploited SQL injection flaw to compromise Sophos XG firewall devices," teiss, April 30, 2020.
[13] "Command Injection," OWASP, accessed June 5, 2020.
[14] Paul Wagenseil, "Thousands of Netgear routers are at risk of getting hacked: What to do," Tom's Guide, March 5, 2020.
[15] Charlie Osborne, "Smart IoT home hubs vulnerable to remote code execution attacks," ZDNet, April 22, 2020.
[16] Dan Goodin, "Critical bugs in dozens of Zyxel and Lilin IoT models under active exploit," Ars Technica, March 21, 2020.
[17] Tom Spring, "Hackers Actively Exploit 0-Day in CCTV Camera Hardware," Threatpost, March 23, 2020.
[18] Ionut Arghire, "GitHub Paid Out Over $1 Million in Bug Bounties," SecurityWeek, March 27, 2020.
[19] Swati Khandelwal, "Hackers Exploit Zero-Day Bugs in Draytek Devices to Target Enterprise Networks," The Hacker News, March 27, 2020.
[20] "What Is Expression Language Injection?" Contrast Security, accessed June 5, 2020.
[21] Swati Khandelwal, "Adobe Patches Vulnerabilities in Illustrator, Experience Manager," SecurityWeek, January 14, 2020.
[22] "OWASP Top Ten," OWASP, accessed June 8, 2020.
[23] "What Is SQL Injection & How Does It Happen?," Contrast Security, accessed June 8, 2020.
[24] Catalin Cimpanu, "Hacker leaks 23 million usernames and passwords from Webkinz children's game," ZDNet, April 18, 2020.
[25] Jakob Pennington, "Shifting Left: DevSecOps as an Approach to Building Secure Applications," Medium, July 18, 2019.
[26] Alan Shimel, "DevOps Chat: Shifting Security Left and Right, With Contrast Security," Security Boulevard, October 7, 2019.

Contrast Security provides the industry's most modern and comprehensive Application Security Platform, removing security roadblocks inefficiencies and empowering enterprises to write and release secure application code faster. Embedding code analysis and attack prevention directly into software with instrumentation, the Contrast platform automatically detects vulnerabilities while developers write code, eliminates false positives, and provides context-specific how-to-fix guidance for easy and fast vulnerability remediation. Doing so enables application and development teams to collaborate more effectively and to innovate faster while accelerating digital transformation initiatives. This is why a growing number of the world's largest private and public sector organizations rely on Contrast to secure their applications in development and extend protection in production.

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

Contrast SECURITY

contrastsecurity.com