SC MEDIA
A CyberRisk Alliance Resource

Data security, Cloud security, Application security

How the latest SQL injection attacks threaten web application firewalls

Paul Wagenseil February 6, 2023

Getty Images

Web application firewalls (WAFs) are designed to stop SQL injection and other common attack techniques that target websites, online apps and servers. The newest form of SQL injection attack using JSON can bypass traditional protections, threatening organizations that use WAFs to protect their online assets.

"Attackers using this novel technique could access a backend database and use additional vulnerabilities and exploits to exfiltrate information via either direct access to the server or over the cloud," stated a research report issued in early December 2022 by Claroty, an operational-technology-protection firm based in New York and Tel Aviv.

"This is especially important for OT and IoT platforms that have moved to cloud-based management and monitoring systems," the report added. "WAFs offer a promise of additional security from the cloud; an attacker able to bypass these protections has expansive access to systems."

doesn't notice.

In plain English, the attackers are secretly telling the database what to do while feeding it supposedly harmless data. Because many websites and web applications that use SQL-based databases also permit users and visitors to input data, this creates a vast opportunity for SQL injection attacks.

Generally, an attacker would launch a SQLi attack by inputting malicious data into a webpage form field or a web app. In the most egregious cases, attackers might only need to modify database-query strings — everything to the right of the "?" in a URL — in browser address bars.

For example, if the database management software is expecting a name like "Smith" as an input, but instead receives a SQL command like `' DROP TABLE names; --` , the software will execute the command instead of treating it as an input. In this case, the result might be the deletion of an entire table of data labeled "names".

To take another example, this SQL statement would search a database for all examples of "Smith" or "smith":

```
select * from person where name = 'smith'
```

However, if someone were to enter this input into any name field:

```
' or 1=1; --
```

then the statement becomes

```
select * from person where name = '' or 1=1; --'
```

This input will force the management software to return every name in the database, giving the attacker a trove of sensitive data. (In SQL syntax, "*" returns any piece of data that meets the statement parameters; single quotations denote the beginnings and endings of parameters; "1=1" is invariably true; and the double hyphen "--" indicates that all following text should be ignored.)

More complex versions of this attack could force the database to cough up credit-card numbers, hashed passwords or user session cookies. Other types of SQL injection attacks can trick online databases into revealing their structures and management software. The most damaging SQLi attacks can add or delete data or even add files to the database server.

SQL injection attacks date back to at least 1998. Many database management systems have evolved mitigations against them, such as by "sanitizing" inputs so they cannot be parsed as commands. Yet the attacks remain common.

"SQL injections present themselves as the most likely guarantee an attacker has to easily and illegitimately gain access to a website or other SQL-backed system, simply based on the probability of success," stated application-security company Invicti on its blog in 2013.

queries are so extremely easy to create, and secure SQL queries are still mildly complex (or at least more complex than generic and typical in-line and often insecure queries)."

How WAFs stop SQL injection

One of the most effective ways to minimize the chances of successful SQL injection is by using a web application firewall (WAF).

While regular network firewalls are put up client-side by organizations to defend users and devices, WAFs are implemented server-side to protect websites and web applications. Barracuda Networks, Cloudflare and F5 are among the best-known providers of commercial WAFs, and there are also free open-source alternatives such as ModSecurity.

"The primary functionality of a WAF is to filter out web attack attempts in real time, mitigating OWASP Top 10 application vulnerabilities such as SQL injections and cross-site scripting (XSS) as well as blocking other attack vectors," explained Invicti in a blog post.

WAFs can be run from dedicated network appliances on the same premises as the servers they protect or can be added to the server software itself. There are also cloud-based WAFs operated from third-party servers. All work best in conjunction with intrusion detection/protection systems (IDS/IPS) and next-generation firewalls (NGFWs) to protect networks.

Because WAFs operate at Layer 7 of the network model, they have more insight into what kind of data is coming in than a traditional Layer 3-based firewall. WAFs monitor and filter incoming HTTP GET and POST requests, blocking data packets they deem malicious. They can be very helpful when complying with regulations and standards such as the payment-card-industry data-security standard (PCI DSS).

Some WAFs whitelist known good Internet Protocol (IP) addresses and block the rest, which would work for web apps dedicated to specific clients. Public-facing WAFs generally blacklist known bad IP addresses and screen everything else, a process that's more resource-intensive than whitelisting. "Hybrid security" WAFs blend the two approaches.

Although each brand and version of WAF will handle things a bit differently, they generally block SQL injection attacks by screening incoming POST packets for SQL syntax.

Some bits of syntax are known to be malicious, e.g., `' DROP TABLE xxxx; --` as above and will be blocked automatically. In other cases, the WAF will parse the SQL syntax to try to figure out if it might be malicious. It may add "escape" characters to SQL syntax to render it harmless, a method many database-management systems also use.

How WAF protections can be bypassed

Sometimes, WAFs can be outdone. While probing cnMaestro (a wireless device management platform) for vulnerabilities, the Claroty researchers noticed that a SQL injection attack used for successful exploitation with on-premises cnMaestro implementations did not work on the cloud version.

that no WAF would recognize?" the researchers wondered.

They found that while the Amazon WAF had no trouble spotting and blocking regular SQL syntax, it didn't always recognize statements written in JavaScript Object Notation (JSON), a widely used data-interchange format that helps web developers structure data.

JSON can be abused to mount JavaScript-based attacks on web pages but was not known to be a vector in SQLi attacks. Yet because JSON is so commonly used, many relational-database management systems now support it along with SQL.

There's the problem. The most widely used SQL management systems, including Microsoft SQL Server, MySQL, PostgreSQL and SQLite, have added support for JSON syntax over the past decade. But the Amazon WAF didn't always see JSON syntax as a possible attack vector and sometimes ignored it.

"If we could supply a SQLi payload that the WAF will not recognize as valid SQL, but the database engine will parse it, we could actually achieve the bypass," the Claroty report said. "As it turns out, JSON was exactly this mismatch between the [Amazon] WAF's parser and the database engine. When we passed valid SQL statements that used less prevalent JSON syntax, the WAF actually did not flag the request as malicious."

More specifically, the JSON operator `@>` "threw the WAF into a loop and allowed us to supply malicious SQLi payloads, allowing us to bypass the WAF."

Using a JSON-based SQL injection attack, the Claroty researchers were able to successfully steal an administrative-user session cookie from their own vulnerable web app hosted on AWS.

"By simply prepending simple JSON syntax to the start of the [SQL] request," said the researchers, "we were able to exfiltrate sensitive information using our SQLi vulnerability over the cloud!"

To be fair, the Claroty researchers may not have been the first to uncover this flaw. Security researcher Ivan Novikov documented a theoretical JSON-in-SQL attack in 2017, but his findings were apparently not widely acted upon.

A widespread problem

The Claroty team found that the same JSON-in-SQL attack worked against WAFs from Cloudflare, F5, Imperva and Palo Alto Networks. An advisory issued by the state of New Jersey on Dec. 15, 2022, specified them as "Palo Alto Next-Generation Firewall, F5 Big-IP, Amazon AWS ELB, Cloudflare, and Imperva." (Check Point's CloudGuard AppSec and its open-source spin-off, open-appsec, were not affected by the vulnerability.)

"While JSON support is the norm among database engines, the same cannot be said for WAFs," the Claroty report said. "Vendors have been slow to add JSON support, which allowed us to craft new SQL injection payloads that include JSON that bypassed the security WAFs provide."

submitted its findings to the open-source penetration-testing tool SQLMap, and while that change has not yet been fully ingested, anyone will soon be able to try this attack against a SQL database.

"We also tried to notify some other smaller WAF vendors, but they did not respond to us," Claroty's Noam Moshe told TechTarget in December 2022. "However, since all the major WAF vendors are now blocking these types of attacks we felt it's the right time to publish."

If your organization is using a WAF, check with the vendor or code maintainer to make sure it has been patched against the JSON-in-SQL vulnerability, and update your software.

"This is a dangerous bypass, especially as more organizations continue to migrate more business and functionality to the cloud," said the Claroty report. "IoT and OT processes that are monitored and managed from the cloud may also be impacted by this issue, and organizations should ensure they're running updated versions of security tools in order to block these bypass attempts."

Paul Wagenseil
Paul Wagenseil is custom content strategist for CyberRisk Alliance, leading creation of content developed from CRA research and aligned to the most critical topics of interest for the cybersecurity community. He previously held editor roles focused on the security market at Tom’s Guide, TechNewsDaily.com, SecurityNewsDaily.com, and spent nearly 10 years at FoxNews.com.

RELATED

DATA SECURITY

California school district confirms data breach

SC Staff June 29, 2023

California's Sweetwater Union High School District has confirmed that its computer network had been compromised in February, resulting in the theft of data from its students and their families, as well as current and former employees and their dependents.

DATA SECURITY

Prolonged data breach impacts US Patent and Trademark Office

SC Staff June 29, 2023

Prolonged data breach impacts US Patent and Trademark Office The U.S.

PRIVACY

Separate cyber incidents impact Kannact, other health organizations

SC Staff June 25, 2023

Separate health data breaches have been disclosed by Oregon-based digital health firm Kannact, Massachusetts-based mental health and addiction treatment center New Horizons Medical, and Philadelphia-based orthopedic clinic Vincera Institute, HealthITSecurity reports.

RELATED EVENTS

SC MEDIA
A CyberRisk Alliance Resource

ON-DEMAND EVENT

GET DAILY EMAIL UPDATES

SC Media's daily must-read of the most current and pressing daily news

Business Email*

By clicking the Subscribe button below, you agree to SC Media Terms and Conditions and Privacy Policy.

SUBSCRIBE

ABOUT US

SC Media | CyberRisk Alliance | Contact Us | Careers | Privacy

GET INVOLVED

Subscribe | Contribute/Speak | Attend an event | Join a peer group | Partner With Us

EXPLORE

Product reviews | Research | White papers | Webcasts | Podcasts