vice.com

The History of SQL Injection, the Hack That Will Never Go Away

Iby Joseph CoxINov 20 2015, 8:00am

7-9 minutes

One of the hackers suspected of being behind the TalkTalk breach, which led to the personal details of at least 150,000 people being stolen, used a vulnerability discovered two years before he was even born.

That method of attack was SQL injection (SQLi), where hackers typically enter malicious commands into forms on a website to make it churn out juicy bits of data. It's been used to steal the personal details of World Health Organization employees, grab data from the Wall Street Journal, and hit the sites of US federal agencies.

"It's the most easy way to hack," the pseudonymous hacker w0rm, who was responsible for the Wall Street Journal hack, told Motherboard. The attack took only a "few hours."

But, for all its simplicity, as well as its effectiveness at siphoning the digital innards of corporations and governments alike, SQLi is relatively easy to defend against.

So why, in 2015, is SQLi still leading to some of the biggest breaches around?

SQLi was possibly first documented by Jeff Forristal in the hacker zine Phrack. Back then, Forristal went by the handle rain.forest.puppy, but he's now CTO of mobile security at cybersecurity vendor Bluebox security.

"According to Microsoft, what you're about to read is not a problem, so don't worry about doing anything to stop it."

SQL, or Structured Query Language, is a programming language used to manage databases. In essence, it's used when a website needs to call up a piece of information from its database, either to process it or present it to a user.

But Forristal had found that typing certain commands would force a server to reveal information stored on it. "People can possibly piggyback SQL commands," he wrote.

In the December 1998 issue of Phrack, Forristal wrote about a series of issues with a version of Microsoft SQL server. When Forristal's fellow researcher told Microsoft of the problems, "their answer was, well, hilarious," he wrote. "According to them, what you're about to read is not a problem, so don't worry about doing anything to stop it."

Today, over 15 years after it was first publicly disclosed, SQLi repeatedly sits at the number one spot of vulnerabilities in the OWASP Top 10 report, which is released every three years by the Open Web Application Security Project (OWASP) Foundation, a non-profit that monitors the threats that websites face.

Phrack's current logo. Image: Phrack

"SQL injection is always the number one risk. That is a reflection of just how many incidents are out there, as well as other factors that keep it very high up there," Troy Hunt, founder of breach site haveibeenpwned.com, told Motherboard in a phone interview.

"When you go to a webpage, and you make a request, that parses part of the data in the request back to a server," Hunt said. "For example, you read a news article, and the news article, in the address bar it has, "id=1", and that gives you news article number 1, and then you get another one with ID 2."

But, "with a SQLi attack, an attacker changes that ID in the address bar to something that forces the database to do something it's not meant to do," Hunt said, such as returning a piece of private data.

An individual attack might just return one piece or section of info, so an attacker is likely to "repeat it it over and over and over again, as many times as is necessary, so they get every piece of data from the database," Hunt said.

Naturally, that's going to be quite time consuming. So, a hacker might use tools that automate the process instead. Those include Havij, which "is popular amongst script kiddies as it's for Windows and has a [graphical user interface]," Mustafa Al-Bassam, a security researcher and former LulzSec hacker, told Motherboard in an online chat.

Another commonly used piece of software is sqlmap. "It crawls the pages on the website, similar to how a search engine crawler might, looks for input forms on the website, and submits the forms with inputs that might cause a MySQL syntax error," Al-Bassam added.

When the attacker is looking for a target to hit in the first place, that's just as simple to automate too.

"They would use Google to search for URLs that are known to be typically associated with scripts that are vulnerable to SQL injection," Al-Bassam said. "They would typically have a script that goes through all the URLs and tests them automatically to see if they're vulnerable."

"You could teach a 4-year-old to do it," Al-Bassam added, summing up how incredibly easy the whole process is. Indeed, Hunt has uploaded a video of him teaching his 3-year-old son how to carry out an SQLi attack with Havij.

"You put the URL in, here's all the data out," Hunt told Motherboard. There are also ample YouTube tutorials on how to carry out an SQLi attack.

The thing is, there are solutions ready to be deployed by website developers to stop SQLi attacks and the unnecessary leaking of customers data or corporate details. And those solutions have been around for years.

One of those is the adoption of "prepared statements": when SQL commands controlling the database can't be directly dictated by a user's input.

If the solutions are fairly straight forward, why are SQLi-based attacks still happening?

"The benefit of prepared statements is that they set the semantics of a query so that any incoming data can't surprise the developer by including syntax that changes a query intended to retrieve a single row into a query that extracts data from arbitrary tables," Mike Shema, senior manager, software development engineer from Yahoo!, told Motherboard in an email.

Another is to "use SQL libraries that take care of input sanitization for them," Al-Bassam suggested. This, in short, scrubs any data entered by the user to remove any potential malicious parts of it.

So, if SQLi is so easy that literally a child could do it, and the solutions are fairly straight forward, why are SQLi-based attacks still happening?

"Any serious programmer should know about SQLi, but there's a massive shortage of programmers, so companies hire anyone even if they don't have the right training or experience to mitigate basic vulnerabilities," Al-Bassam suggested. On top of this, "they're often put under pressure by their managers to develop functional software rather than secure software."

Shema from Yahoo! echoed this, and said that "Sometimes small apps with a narrow feature set just need to be written quickly," meaning that the developers might bypass some of the mitigations for each attacks, despite them being relatively straight forward to implement.

Hunt was slightly less forgiving, and didn't agree that it was because of pressure from higher management. Instead, he lamented about the large number of tutorials available to web developers online that, instead of providing decent advice, detail how to make systems that are vulnerable to SQLi. "I've seen multiple tutorials come up this year that have got blatant SQL injection risks in them," he said.

So just as script kiddies continue to share their SQLi tutorials on YouTube, there is parallel information sharing going on with website developers. "We've got this ability for anyone to stand up, and share their knowledge, and not always get it right," Hunt said.

Ultimately, the responsibility of the security of these sites, and the data they contain, boils down to web developers themselves. That means SQLi and the breaches it causes will remain, at least for a little while longer.