مرکز تخصصی
آپا دانشگاه کردستان

انواع حملات تزریق SQL

فرشته کیاست

شماره سند: A96004

۱۳۹۶/۱۱/۱۱

www.cert.uok.ac.ir
apa@uok.ac.ir
087-33662932

چکیده

برنامه‌های کاربردی وب تبدیل به یک بخش ضروری از زندگی روزمره ما شده‌اند و بسیاری از فعالیت‌های ما وابسته به قابلیت و امنیت این برنامه‌ها است. این برنامه‌ها به صورت گسترده در زمینه‌های مختلف مانند انجام وظایف مهم و حیاتی مورد استفاده قرار می‌گیرند و با داده‌های حساس کاربران سروکار دارند. با رشد روزافزون استفاده از این برنامه‌ها، آسیب‌پذیری‌های تزریق، مانند تزریق SQL، به یک چالش امنیتی عمده تبدیل می شود. مهاجمان با استفاده از این آسیب پذیری‌ها و تزریق کد مخرب، بصورت غیرمجاز به پایگاه داده دسترسی می‌یابند و باعث به خطر افتادن امنیت برنامه‌ها می‌شوند. بنابراین تشخیص پرس‌وجوهای مخرب ورودی یک سایت در حفظ امنیت آن از اهمیت بالایی برخوردار است. در این مقاله، انواع حملات تزریق SQL ارائه داده شده است.

کلمات کلیدی : پایگاه داده، تزریق SQL

۱- مقدمه

بسیاری از سازمان ها اطلاعات مهم، حساس و محرمانه مربوط به کارمندان، مشتریان و همکاران تجاری خود را در پایگاه‌های داده در سراسر جهان ذخیره می‌کنند. داده‌های ذخیره شده از اطلاعات با درجه حساسیت کمتر از قبیل نام، نام خانوادگی و تاریخ تولد و اطلاعات حساس تر از قبیل نام کاربری، رمز عبور، اطلاعات کارت بانکی و غیره تشکیل می‌شود. بنابراین، برای هر سازمانی مهم است که از پایگاه داده‌های خود محافظت کنند تا از هر گونه از دست رفتن اطلاعات جلوگیری کنند. در این راستا سازمان‌ها می‌بایست جهت دستیابی به اهداف مهمی مانند قابلیت اعتماد، جامعیت و دسترسی‌پذیری از یک استراتژی مناسب برای محافظت از پایگاه داده‌های خود استفاده نمایند.

در تفسیر اهداف فوق می‌توان مثال‌هایی را بیان نمود:

الف) نقض قابلیت اعتماد: افشای اطلاعات توسط کاربران غیرمجاز؛

ب) نقض جامعیت: تغییر تاریخ توسط مهاجمان؛

ج) نقض دسترسی‌پذیری: حذف داده‌ها از پایگاه داده

حمله تزریق SQL یک آسیب‌پذیری برنامه وب است که از طریق زبان‌های اسکریپتی پویا مانندPHP، ASP، JSP و CGI ایجاد می‌شود. تزریق SQL، تکنیکی است که مهاجمان از آن استفاده کرده تا از طریق ورودی‌های صفحه وب دستورات SQL را در عبارات SQL تزریق کنند. دستورات تزریق شده می‌تواند عبارات SQL را تغییر داده و امنیت برنامه کاربردی وب را به خطر اندازد[۳-۱].

همانطور که در شکل(۱) مشخص است، طبق آخرین گزارش OWASP، بیشترین نوع حملات وب در سال ۲۰۱۷ حملات SQL بوده است[۴]. حمله‌های تزریق SQL در میان حملات اعتبار ورودی، از نوع خطرناکترین حملات وب محسوب می‌شوند[۵].

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

شکل (۱) :گزارش شایعترین حملات OWASP در سال ۲۰۱۷

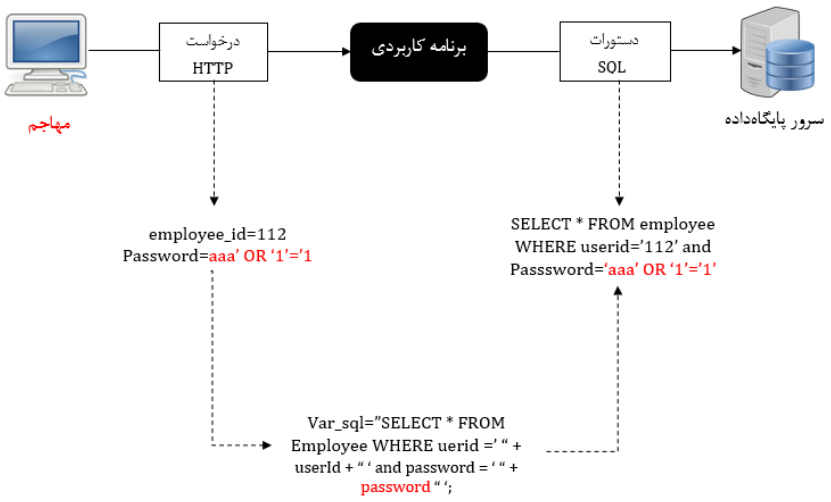
به عنوان مثال زمانی که مدیر یک سایت برای احراز هویت مقادیر id=112 و pass=admin را وارد می‌کند این نمونه‌ای از پرس‌وجوی نرمال است[۶]. در مقابل، شکل (۲) ورود یک کاربر مهاجم با استفاده از آسیب‌پذیری SQL را نشان می‌دهد. این حمله از سه مرحله تشکیل شده‌است.

۱) مهاجم یک درخواست HTTP مخرب را به برنامه وب ارسال می‌کند.

۲) دستور SQL ایجاد می‌شود.

۳) دستور SQL را به پایگاه داده تحویل می‌دهد.

یک برنامه کاربردی، بر اساس مدل فوق، ورودی را از کاربران برای بازیابی اطلاعات از پایگاه داده دریافت می‌کند. برخی از برنامه‌های کاربردی وب بدون اعتبارسنجی، فرض می‌کنند که ورودی نرمال است و از آن برای ساختن پرس و جوهای SQL استفاده می‌کنند. از آنجاییکه در این برنامه‌های کاربردی اعتبارسنجی پیش از ارسال ورودی‌ها برای بازیابی داده صورت نمی‌گیرد، در مقابل حملات تزریق SQL حساس می‌باشند. برای مثال، مهاجمان، به عنوان کاربران عادی، از ورودی مخرب حاوی دستورهای SQL استفاده می‌کنند تا پرس و جوهای SQL را در برنامه وب ایجاد کنند. پس از پردازش توسط برنامه کاربردی، درخواست‌های مخرب پذیرفته شده ممکن است خط مشی های امنیتی معماری پایگاه داده را شکست دهد؛ زیرا نتیجه جستجو ممکن است باعث آسیب به پایگاه داده و انتشار اطلاعات حساس شود. انواع حملات SQL را می توان به هشت دسته : ۱- تتولوژی،۲- پرس و جوی‌های غیرقانونی/منطقی نادرست، ۳- پرس و جوی اجتماع ،۴- PiggyBacked، ۵- رویه‌های ذخیره‌شده، ۶- استنتاج، ۷- حمله کدگذاری متناوب ، تقسیم‌بندی کرد[۱۰-۷]. تست امنیتی یک فعالیت محوری در مهندسی نرم افزار امن است. این تست شامل دو مرحله است: تولید ورودی های حمله برای تست سیستم و ارزیابی اینکه آیا اقدام‌های اکتشافی هر آسیب پذیری را در معرض آشکار قرار می دهند[۱۱].



شکل (۲) :نمونه‌ای از یک کد تزریق SQL

محققان طیف گسترده ای از تکنیک ها را برای حل مشکل تزریق SQL پیشنهاد کرده‌اند. این تکنیک ها از بهترین شیوه های کاملا اتوماتیک برای تشخیص و جلوگیری از تزریق SQL استفاده می‌کنند. از نمونه‌های این تکنیک‌ها می‌توان به بررسی نوع ورودی[۱]، کدگذاری ورودی[۲]، تست جعبه سیاه[۳]، چک کننده‌های کد ایستا[۴]، رویکردهای مبتنی بر آلودگی[۵]، سیستم‌های تشخیص نفوذ[۶]، اشاره کرد[۷, ۱۲, ۱۳]. اخیرا روش‌های مبتنی بر هوش مصنوعی، تشخیص تزریق SQL با استفاده از ماشین بردار پشتیبان[۷] [۱۴]و تشخیص حمله تزریق بر اساس تکنیک‌های یادگیری ماشین(مبتنی بر شبکه عصبی[۸])[۱۵]، پیشنهاد شده است. در روش تشخیص تزریق SQL با استفاده از ماشین بردار پشتیبان با استفاده از مجموعه آموزشی مدلی از پرس‌وجوهای مخرب ایجاد می‌شود و هر نمونه ورودی با این مدل مقایسه می‌شود. در روش مبتنی بر شبکه عصبی، از ۳۲ ویژگی برای تشخیص چرس‌وجوی مخرب استفاده می‌کند. این ویژگی‌ها به شبکه عصبی پس انتشار سه لایه داده می‌شود تا مدلی از پرس‌وجوهای مخرب ایجاد کند.

۲- انواع حملات تزریق SQL

۲-۱- تتلوژی[۹]

هدف کلی یک حمله مبتنی بر حشو این است که کدی صحیح را در یک یا چند عبارتی شرطی تزریق کند تا همیشه آنها را به درستی ارزیابی کند. عواقب این حمله به این بستگی دارد که چگونه نتایج پرس‌و جو در برنامه کاربردی مورد استفاده قرار می‌گیرد. رایج ترین نوع استفاده، این است که صفحات احراز هویت را دور بزنند و اطلاعات را استخراج کنند. در این حمله، مهاجم عبارت " or 1=1 -- " را در فیلد ورودی login وارد می‌کند. نتیجه پرس و جو به صورت زیر خواهد بود.

$$select\ account\ from\ users\ where$$
$$login = '' \ or\ 1 = 1 - -and\ pass = ''\ and\ pin =$$

کد تزریق شده در شرط "or 1=1" تمام عبارت where را به یک تتلوژی تبدیل می‌کند. پایگاه داده از شرط به عنوان مبنا برای ارزیابی هر ردیف استفاده می‌کند و تصمیم می‌گیرد که کدام یک از آنها به برنامه بازگردد. از آنجا که شرط یک تتلوژی است، پرس و جو برای هر سطر در جدول به درستی برای هر ردیف ارزیابی می شود و تمام آنها را باز می‌گرداند[۱۶].

۲-۲- پرس و جوی‌های نادرست غیرقانونی/منطقی[۱۰]

این حمله به مهاجم اجازه می‌دهد اطلاعات مهمی در مورد نوع و ساختار پایگاه داده یک برنامه وب جمع آوری کند. این حمله یک گام کمکی برای جمع‌آوری اطلاعات برای حملات دیگر محسوب می‌شود. هنگام انجام این حمله، مهاجم سعی می‌کند پرس‌وجوهایی را ایجاد کند که سبب نحو، یا خطای منطقی در پایگاه داده شود. خطاهای نحو برای تعیین پارامترهای تزریقی استفاده می‌شود. خطاهای منطقی اغلب نام جدول‌ها و ستون‌هایی که باعث خطا شده‌اند را نشان می دهد[۷, ۱۷, ۱۸].

۲-۳- پرس و جوی اجتماع[۱۱]

در حملات اجتماع ، مهاجم از یک پارامتر آسیب پذیر برای تغییر مجموعه داده ها برای یک پرس و جو داده شده استفاده می‌کند. در این تکنیک با استفاده از عملگر اجتماع دو پرس‌وجوی SQL اجتماع می‌شوند بطوریکه پرس‌وجوی اول نرمال و پرس‌وجوی دوم مخرب و توسط مهاجم تزریق شده است. در مثال زیر نمونه‌ای از پرس‌وجوی اجتماع نمایش داده شده است.

$$select * from\ accounts\ where\ id =' 212' UNION\ select * from\ credit_{card}\ where\ user =' admin' --' and\ pass =' pass'$$

نتیجه این حمله، یک مجموعه داده است که پایگاه داده در پاسخ به اجتماع دو پرس و جو بازمی‌گرداند[۱۹].

۲-۴- پرس و جوهای Piggy-Back

در این نوع حمله، یک مهاجم سعی می‌کند پرس‌وجوهای بیشتری را به پرس و جوی اصلی تزریق کند. این پرس‌وجوها به پرس‌وجوی اول اضافه می‌گرند. در نتیجه، پایگاه داده چندین پرس و جو SQL را دریافت می‌کند. اولین پرس و جو که به صورت عادی اجرا می شود پرس‌وجوی اصلی است، موارد بعدی که اجرا می‌شوند پرس و جوهای تزریقی هستند. به عنوان مثال اگر مهاجم ورودی "'; drop table users - -" را در فیلد pass وارد کند، برنامه پرس وجوی زیر را تولید می‌کند.

$$select\ accounts\ from\ users\ where$$

$$login =' doe' and\ pass =''; drop\ table\ users$$

$$--' and\ pin = 123$$

نتیجه اجرای دومین پرس و جو این است که کاربران جدول را حذف کنند، که احتمالا اطلاعات ارزشمندی را از بین می برد[۴, ۱۵].

۲-۵- رویه‌های ذخیره شده[۱۲]

این نوع تزریق سعی می کند رویه‌های ذخیره شده موجود در پایگاه داده را اجرا کند. برای ایجاد این نوع تزریق، مهاجم پرس و جوی ساده‌ی $'' ; shutdown; --$ را می‌تواند به فیلدهای نام کاربری و پسورد تزریق

کند. این تزریق باعث می‌شود که رویه ذخیره شده پرس و جوی زیر را تولید کند.

$$select\ accounts\ from\ users\ where\ login =' doe' and$$

$$pass =''; shutdown; --\ and\ pin =$$

در این مرحله، این حمله مشابه حمله piggy-back عمل می کند. اولین پرس و جو به طور معمول انجام می‌شود، و سپس پرس و جو مخرب اجرا می شود، که نتیجه آن باعث خاموشی پایگاه داده می‌شود[۷, ۱۶, ۲۰].

۶-۲- استنتاج[۱۳]

در این نوع تزریق، مهاجمان عموما سعی دارند به سایتی حمله کنند که به اندازه کافی امن باشد. از آنجا که در سایت‌های امن پیام‌های خطای پایگاه داده در دسترس نیستند تا به مهاجم بازخورد ارائه دهند، مهاجمان باید از روش متفاوتی برای دریافت پاسخ از پایگاه داده استفاده کنند. در این وضعیت، مهاجم دستورات را به سایت تزریق می کند و سپس مشاهده می‌کند که پاسخ وب‌سایت چگونه تغییر می‌یابد. زمانی که رفتار سایت در پاسخ، تغییر می کند، مهاجم نتیجه می‌گیرد که نه تنها این پارامترهای خاص آسیب پذیر هستند، بلکه اطلاعات اضافی در مورد مقادیر موجود در پایگاه داده را به دست می‌آورد[۲۱].

۷-۲- حمله کدگذاری متناوب[۱۴]

در این روش، مهاجم پرس‌وجوی تزریق را با استفاده از کدگذاری متناوب مانند هگزا دسیمال، اسکی و یونیکد تغییر می‌دهد. به این ترتیب از فیلتر برنامه نویس، که پرس‌وجوی ورودی را چک می کند تا شامل کاراکترهای خطرناک نباشد، میتواند فرار کند. به عنوان مثال از کد char(44) به جای کاراکتر نقل قول تک استفاده می‌کند[۷].

۳- نتیجه‌گیری

دراین مقاله سعی بر آن شد انواع حملات تزریق پایگاه داده شرح داده شود. در مقابل این حملات تکنیک‌هایی برای پیشگیری و تشخیص این نوع حملات ارائه شده‌اند که در مقاله بعدی سعی می شود این تکنیک‌ها بررسی شوند.

مراجع

[1] Kim, M.-Y. and D.H. Lee, Data-mining based SQL injection attack detection using internal query trees. Expert Systems with Applications, 2014. 41(11): p. 5416-5430.

[2] Raji, V. and P. Ashokkumar, Protecting Database from Malicious Modifications Using JTAM. Journal of Computer Applications, 2012.

[3] Nicolett, M. and J. Wheatman, Dam Technology Provides Monitoring and Analytics with Less Overhead. Gartner Research (Nov. 2007). 2010.

[4] Williams, J. and D. Wichers, The Ten Most Critical Web Application Security Risks. rc1,//OWASP Foundation, 2017.

[5] Tajpour, A. and M.J. zade Shooshtari. Evaluation of SQL injection detection and prevention techniques. in Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on. 2010. IEEE.

[6] Ibrahim, S., M. Masrom, and A. Tajpour, SQL injection detection and prevention techniques. 2011.

[7] Halfond, W.G., J. Viegas, and A. Orso. A classification of SQL-injection attacks and countermeasures. in Proceedings of the IEEE International Symposium on Secure Software Engineering. 2006. IEEE.

[8] Moosa, A., Artificial neural network based web application firewall for sql injection. World Academy of Science, Engineering and Technology, 2010. 40: p. 12-21.

[9] Win, W. and H.H. Htun, A simple and efficient framework for detection of sql injection attack. IJCCER, 2013. 1(2): p. 26-30.

[10] Kindy, D.A. and A.-S.K. Pathan. A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques. in Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on. 2011. IEEE.

[11] Ceccato, M., et al. SOFIA: an automated security oracle for black-box testing of SQL-injection vulnerabilities. in Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. 2016. ACM.

[12] Bau, J., et al. State of the art: Automated black-box web application vulnerability testing. in Security and Privacy (SP), 2010 IEEE Symposium on. 2010. IEEE.

[13] Halfond, W.G. and A. Orso. AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. in Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. 2005. ACM.

[14] Rawat, R. and S. Raghuwanshi, SQL injection attack Detection using SVM. International Journal of Computer Applications, 2012. 42(13): p. 1-4.

[15] Sheykhkanloo, N.M. SQL-IDS: evaluation of SQLi attack detection and classification based on machine learning techniques. in Proceedings of the 8th International Conference on Security of Information and Networks. 2015. ACM.

[16] Anley, C., Advanced SQL injection in SQL server applications. 2002.

[17] Litchfield, D., Web application disassembly with ODBC error messages. Windows Security, 2001.

[18] McDonald, S., SQL Injection: Modes of attack, defense, and why it matters. White paper, GovernmentSecurity. org, 2002.

[19] Singh, J.P., Analysis of SQL Injection Detection Techniques. arXiv preprint arXiv:1605.02796, 2016.

[20] Bezdek, J.C., R. Ehrlich, and W. Full, FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 1984. 10(2-3): p. 191-203.

---

۱ Checking Input Type
۲ Encoding of inputs
۳ Black Box
۴ Static Code Checkers
۵ Taint Based Approaches
۶ IntrusionDetection Systems
۷ SVM
۸ Artifitial Neural Network
۹ Tautologies
۱۰ Illegal/logically incorrect queries attack
۱۱ Union quiery
۱۲ Stored Procedures attack
۱۳ Inference attack
۱۴ Alternate Encodings attack