دانشگاه تهران
پردیس دانشکده های فنی
دانشکده مهندسی برق و
کامپیوتر

آسیب‌پذیری در پایگاه‌داده‌های SQL و No-SQL

گزارش ارائه نهایی درس مبانی امنیت شبکه‌های کامپیوتری

نام دانشجو:
علی مهرانی - 810198542
طه ابراهیم‌زاده میاب - 810198335
علی افتخاری - 810198549

استاد راهنما:
جناب آقای دکتر صیاد حقیقی

خرداد ماه 1403

فهرست مطالب

فهرست اشکال

با تشکر بسیار از جناب آقای دکتر صیاد حقیقی بابت تمامی زحماتشان در طول ترم

چکیده

در این گزارش به بررسی آسیب‌پذیری‌ها و خطرات در پایگاه‌های داده SQL و NoSQL می‌پردازیم. ابتدا اشاره‌ای به اهمیت و کاربرد پایگاه‌های‌داده و کاربردهای وسیع آن‌ها می‌کنیم. در فصل دوم، مروری جامع بر پایگاه‌های داده SQL و NoSQL ، به همراه تفاوت‌ها و مزایای هر کدام می‌کنیم. سپس در فصل سوم به تحلیل و بررسی انواع مختلف آسیب‌پذیری‌ها در پایگاه‌های داده می‌پردازیم. خطرات مربوط به SQL/NoSQL Injection ، ارجاع مستقیم ناامن(IDOR) ، ناسازگاری داده‌ها، پیکربندی نادرست سیستم پایگاه داده، ارتقاء دسترسی، احراز هویت و مجوزدهی ضعیف، رمزنگاری نامناسب و buffer overflow از خطراتی هستند که در این ارائه به آن‌ها اشاره می‌کنیم و آن‌ا را مورد بررسی قرار می‌دهیم. هدف از این گزارش، افزایش آگاهی و ارائه راهکارهای امنیتی برای کاهش مخاطرات مرتبط با این آسیب‌پذیری‌ها می‌باشد.

کلمات کلیدی: پایگاه‌داده، SQL، No-SQL، امنیت، Vulnerability

فصل اول

مقدمه

امروزه پایگاه‌های داده در تمامی برنامه‌های تحت شبکه مورد استفاده قرار می‌گیرند. حفظ امنیت این سیستم‌ها برای بقا برنامه ایجاد شده الزامی و بسیار مهم می‌باشد. در این گزارش ابتدا مروری بر انواع پایگاه های داده می‌کنیم و سپس به خطرات و ریسک‌هایی اشاره می‌کنیم که امنیت هر یک از آن ها را تهدید میکند و در ادامه نیز به بررسی راه ها و روش‌هایی به جهت کم کردن و به حداقل رساندن این خطرات می‌پردازیم. با بررسی هرکدام از خطرات درباره آن ها مثال‌هایی زده تا میزان خطرناک بودن آن ها را بهتر مشاهده نماییم و همچنین می بینیم که در سناریوهای واقعی چگونه می‌توان با این تهدیدها مقابله کرد.

فصل دوم
مروری بر پایگاه‌های داده

3-1- پایگاه‌های داده SQL

پایگاه‌های داده SQL یا relational به عنوان یکی از مهم‌ترین و پرکاربردترین ابزارهای مدیریت داده‌ها در دنیای امروز شناخته می‌شوند. برای ایجاد و مدیریت داده ها در این پایگاه‌های داده از زبان SQL استفاده می‌شود که برای data definition و data manipulation در آن استفاده می‌شود. در این database ها داده‌ها در جداولی با سطرها و ستون‌ها سازماندهی و ذخیره می‌شوند. این پایگاه‌ها به کاربر امکان می‌دهند تا داده‌ها را به صورت ساختاریافته ذخیره، بازیابی و مدیریت کنند. طراحی این database ها مبتنی بر schema و با ساختار fixed است. این database ها تمام ویژگی‌های ACID را به کاربر گارانتی می‌دهند و یکپارچگی داده را تضمین می‌کنند. این database ها همچنین در بسیاری از cloud service ها نیز قابل استفاده هستند. از جمله معروف این پایگاه‌داده‌ها می‌توان به PostgreSQL، MySQL و Oracle Database اشاره کرد.

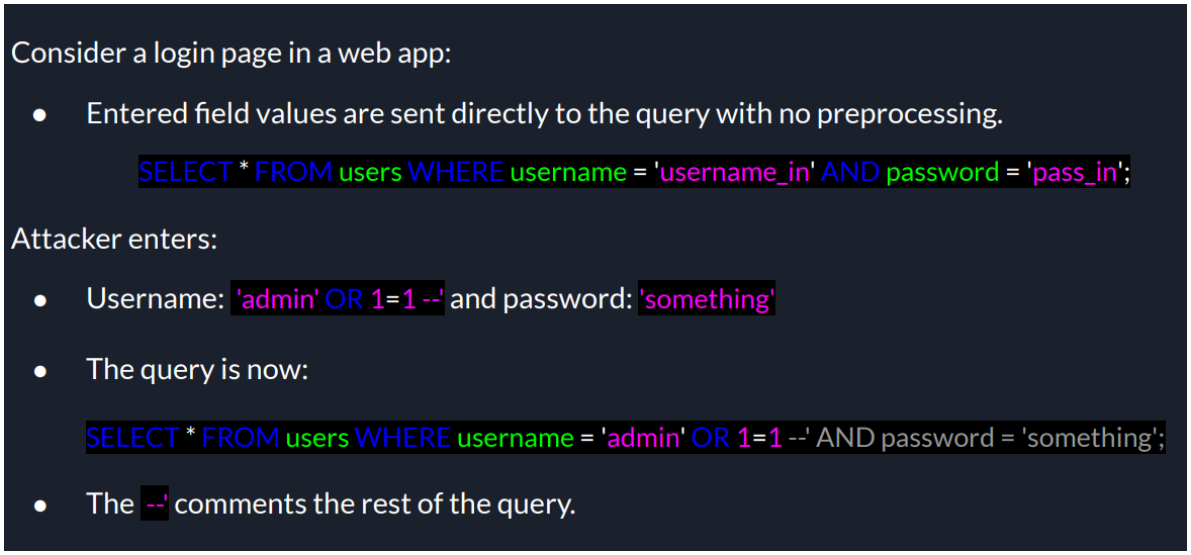
3-2- پایگاه‌های داده No-SQL

در شرایطی که نیاز به scalability بالا و مدیریت داده‌های حجیم وجود دارد، پایگاه‌های داده NoSQL به عنوان یک جایگزین مناسب برای پایگاه‌های داده سنتی SQL مطرح می‌شوند. این database ها برای کار با big data و برنامه‌های real-time بسیار مناسب هستند. برخلاف database های relational، این database ها نیاز به تعریف کردن یک schema مشخص برایشان ندارند و به راحتی با داده‌های بدون ساختار مشخص نیز کار می‌کنند. این database ها همچنین توانایی horizontal scaling را به ما می‌دهند تا با اضافه کردن node های بیشتر برنامه‌های بزرگ تر را handle کنند و این آن ها را برای ذخیره و پردازش داده‌های توزیع‌شده یا distributed، مناسب می‌کند. این database ها در برنامه‌های large-scale بسیار کاربردی هستند و در انواع مختلف می‌باشند که در ادامه به چند نوع از آن‌ها اشاره می‌کنیم.

فصل سوم
آسیب‌پذیری در پایگاه‌های داده

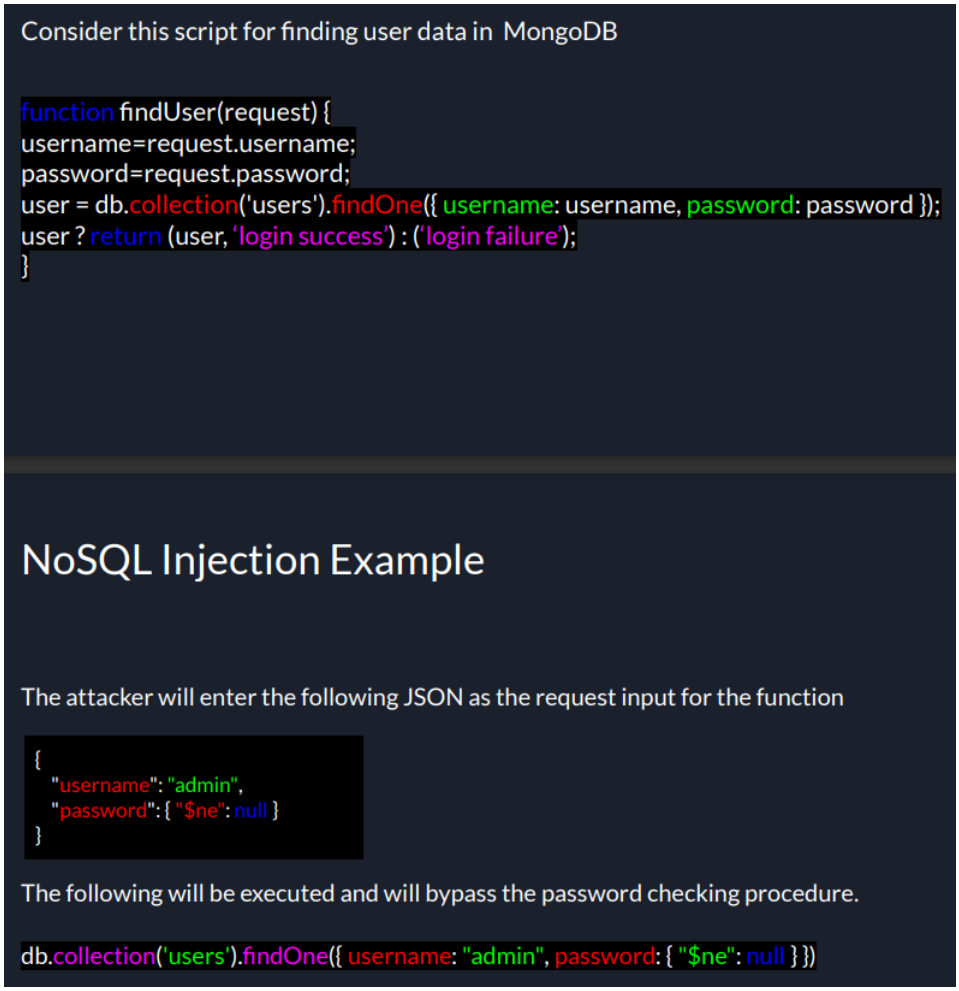
4-1- خطر SQL/No-SQL Injection

SQL Injection یکی از رایج‌ترین و خطرناک‌ترین حملات سایبری به database ها است که بر پایگاه‌های داده مبتنی بر SQL تاثیر می‌گذارد. در این نوع حمله، مهاجم از نقاط ضعف در کد برنامه‌های وب (معمولا در قسمتی که اطلاعات از پایگاه‌داده fetch می‌شود) استفاده می‌کند تا دستورات SQL مخرب را به پایگاه داده ارسال کند. این حملات می‌توانند منجر به افشای اطلاعات حساس یا تخریب داده‌ها شوند. این حمله معمولا از طریق ورود ناامن اطلاعات به جهت اجرای کوئری‌های خطرناک یا malicious query صورت می‌گیرد. شکل زیر نمونه‌ای از این حمله را نمایش می‌دهد:



شکل ۱-۳- نمونه حمله SQL injection

مطابق با شکل بالا،Attacker پس از وارد کردن ورودی پسوورد، بخشی از کوئری که قرار بود اجرا شود را کامنت کرده و تمامی اطلاعات را دریافت می‌کند. مشابه این فرآیند همچنین برای بسیاری از پایگاه‌های داده NoSQL نیز وجود دارد و به آن حمله NoSQL Injection نیز گفته می‌شود که شکل زیر نمونه‌ای از آن را نشان می‌دهد:



شکل ۲-۳- نمونه حمله No-SQL injection

همانگونه که از شکل نیز مشخص است، در MongoDB کاربر با وارد کردن اطلاعاتی خاص در قالب JSON باعث اجراشدن یک Malicious query می‌شود و کاربر در این سناریو، می‌تواند login کند.

برای مقابله با این نوع حمله می‌توان کارهایی انجام داد از جمله:

- **استفاده از prepared statements**: اطمینان حاصل می‌کند که query ها ابتدا compile و سپس بعد از آن، پارامترها ارسال می‌شوند.
- **استفاده از Web Application Firewall**: استفاده از WAF جلوی بسیاری از حملات از جمله SQL inject را می‌گیرد.
- **استفاده از ORM**: ORM ها برای database های relational و ODM برای database های No-SQL، ورودی‌های malicious را چک می‌کنند. (Object-Document Mapper & Object-Relational Mapper)
- همچنین می‌توان به صورت دستی نیز ورودی‌های مشکوک را چک کرد اما کار پر ریسکی می‌باشد.

4-2- خطر Insecure Direct Object Reference (IDOR)

این حمله در اصل بیشتر مربوط به پیاده‌سازی سیستم و نحوه Authorization و Access-control می‌باشد اما در صورت عدم پیاده‌سازی مناسب، امکان سرقت اطلاعات از database و صدمه به آن وجود دارد. مثال زیر را در نظر بگیرید:

فرض کنید اطلاعات دانشجویی در سیستم دانشگاه در یک فایل txt با لینک زیر ذخیره شده که نام فایل شماره دانشجویی اوست:

https://university.com/stdinfo/810100000.txt

در این حالت، Attacker با تغییر شماره دانشجویی در لینک، می‌تواند بدلیل عدم پیاده‌سازی سیاست‌های مناسب access control و همچنین نام گذاری نامناسب فایل‌ها اطلاعات دیگر در پایگاه‌داده را استخراج کند. با انجام موارد زیر می‌توان از سیستم در برابر این خطر حفاظت کرد:

- پیاده‌سازی مناسب سیاست‌های access control و middleware و Authorizazation
- استفاده از سیاست نام گذاری مناسب برای فایل‌های static در سرور، مثلا در این بخش نباید نام فایل شماره دانشجویی باشد یا کلا چیزی که برای Attacker الگوی آن قابل حدس زدن باشد.

4-3- خطر Data Inconsistency

این خطر بیش‌تر پایگاه‌داده‌های NoSQL را تهدید می‌کند زیرا در این نوع پایگاه‌داده‌ها اصول ACID به صورت کامل رعایت نمی‌شوند و این باعث بروز مشکلاتی در زمینه consistency می‌شود.

همچنین اگر از مدل eventual consistency استفاده شود، پس از اعمال تغییرات در یک data center، مقداری زمان باید بگذرد تا داده‌ها با یک‌دیگر هم‌گام شوند. این مدل زمانی استفاده می‌شود که هم‌گام بودن داده در تمامی زمان‌ها برای کاربران اهمیتی نداشته باشد. به طور مثال مشاهده‌ی پست‌های یک کاربر در یک شبکه‌ی اجتماعی با کمی تاخیر، اهمیت زیادی برای کاربران ندارد.

اگر پایگاه‌داده به صورت توزیع‌شده در شبکه قرار گرفته باشد، یکی از مشکلاتی که منجر به inconsistency در داده‌ها می‌شود، مشکلات شبکه‌ای می‌باشد. اختلال در ارتباط میان دو گره می‌تواند باعث شود که یک سری از عملیات‌ها به درستی انجام نشوند و ناهماهنگی رخ بدهد.

4-4- خطر Database System Misconfiguration

یکی از مشکلات دیگری که امنیت یک پایگاه داده را تهدید می‌کند پیکربندی اشتباه است که معمولا به دلیل استفاده از تنظیمات پیش‌فرض ایجاد می‌شوند.

- **باز بودن دسترسی شبکه**: اگر هیچ محدودیتی روی IP هایی که به شبکه‌ی ما متصل می‌شوند نداشته باشیم، راحت‌تر توسط افراد مهاجم به آن‌ها حمله شود و اطلاعات آن نشر پیدا می‌کند که برای جلوگیری از این حمله‌ها می‌توان محدودیت‌هایی روی IPهایی که می‌توانند دسترسی داشته باشند اعمال کرد.
- **عدم به‌روزرسانی‌های امنیتی:** بعضی وقت‌ها یک سری مشکلات امنیتی در یک نرم‌افزار وجود دارد که در به‌روزرسانی‌ها، رفع می‌شود و اگر آن به‌روزرسانی انجام نشود، مهاجم‌ها از همین حفره‌ی امنیتی می‌توانند استفاده کنند و به پایگاه‌داده‌ی ما نفوذ کنند. برای جلوگیری از این باید سعی کرد تا به صورت مرتب، به‌روزرسانی‌ها به خصوص به‌روزرسانی‌های امنیتی، انجام شود و همچنین به صورت منظم آزمون‌های نفوذ به سیستم گرفته شود و در صورت وجود یک حفره، شناسایی شود و مشکلش رفع شود.
- **نامناسب بودن تنظیمات بک‌آپ:** ممکن است این مشکل باعث از دست رفتن داده‌های پایگاه‌داده شود و برای جلوگیری از این موضوع باید به صورت منظم و اتوماتیک، بک‌آپ‌هایی گرفته شود و در صورت نیاز با استفاده از آن‌ها اطلاعات را بازیابی کرد.

4-5- خطر Privilege Escalation

هنگامی که یک کاربر بتواند امتیازی را به دست بیارد که نباید آن امتیاز را داشته باشد، privilege escalation رخ داده است که دو نوع دارد:

- Horizontal Privilege Escalation: در این حالت یک کاربر با یک امتیاز خاص، به حساب کاربری شخصی دیگر با همان امتیاز دسترسی دارد. این حمله می‌تواند به شیوه‌های مختلفی مانند SQL injection یا استفاده از باگ‌های نرم‌افزاری رخ دهد.

- Vertical Privilege Escalation: در این حالت یک کاربر امتیازی بالاتر از امتیازی که باید داشته باشد را به دست می‌آورد. به طور مثال یک کاربر معمولی، می‌تواند مانند admin دسترسی و توانایی داشته باشد.

برای جلوگیری از privilege escalation می‌توان ورودی‌ها را اعتبارسنجی کرد تا از حمله‌هایی مثل SQL injection جلوگیری کرد. از multi-factor authentication استفاده کرد تا احتمال این‌که کسی به عنوان شخصی دیگر وارد نشود. از principle of least privilege استفاده کرد که یعنی به هر کاربر به میزانی دسترسی و امتیاز بدهیم که به آن نیاز دارد تا از آن دسترسی‌ها و امتیازها سوء استفاده نشود. در نهایت باید patch management داشته باشیم تا به‌روزرسانی‌هایی که از نظر امنیتی حیاتی هستند، اعمال شوند.

4-6- خطر Weak Authentication and Authorization

احراز هویت و مجوزدهی از اجزای حیاتی در امنیت سیستم‌ها و داده‌ها هستند. این دو فرآیند تضمین می‌کنند که فقط کاربران معتبر به منابع خاص دسترسی داشته باشند و اقدامات مطابق با سطح دسترسی خود را انجام دهند. با این حال، نقاط ضعف رایج در این زمینه‌ها می‌توانند به آسیب‌پذیری‌های جدی امنیتی منجر شوند.

احراز هویت: احراز هویت فرآیندی است که هویت یک کاربر یا سیستم را تأیید می‌کند. این فرآیند تضمین می‌کند که موجودیت درخواست کننده دسترسی همان کسی است که ادعا می‌کند. مکانیزم‌های رایج احراز هویت شامل کلمه‌های عبور، داده‌های بیومتریک و توکن‌ها هستند.

مجوزدهی: مجوزدهی سطح دسترسی کاربران تأیید شده را تعیین می‌کند. پس از تأیید هویت کاربر، مکانیزم‌های مجوزدهی کنترل می‌کنند که چه اقداماتی کاربر می‌تواند انجام دهد و به کدام منابع می‌تواند دسترسی داشته باشد. این موضوع معمولاً از طریق سیاست‌های کنترل دسترسی مدیریت می‌شود.

نقاط ضعف رایج در احراز هویت و مجوزدهی

هم دیتابیس های SQL و هم NoSQL می‌توانند از نقاط ضعف مشابهی رنج ببرند:

- **استفاده از اعتبارنامه‌های پیش‌فرض:** استفاده از نام‌های کاربری و کلمه‌های عبور پیش‌فرض که به راحتی قابل حدس زدن هستند.
- **عدم وجود احراز هویت چندعاملی (MFA):** نداشتن مکانیزم‌های احراز هویت چندعاملی که می‌تواند امنیت بیشتری را فراهم کند.
- **سیاست‌های ضعیف کلمه عبور:** استفاده از کلمه‌های عبور ساده و ضعیف که به راحتی قابل شکستن هستند.
- **کنترل‌های دسترسی نادرست:** نبودن کنترل‌های مناسب برای محدود کردن دسترسی کاربران به منابع خاص.
- **ذخیره‌سازی کلمه‌های عبور به صورت متن ساده:** ذخیره کردن کلمه‌های عبور به صورت متن ساده که امنیت آن‌ها را به خطر می‌اندازد. درواقع با دستیابی به دیتابیس تمامی رمزهای عبور قابل دسترسی بوده و امنیت کاربران به خطر می افتد. حداقل اقدامی که برای ذخیره رمز عبور باید انجام داد این است که هش رمز های عبور را به جای خود آن ها ذخیره کرد.
- **نبود کنترل دسترسی مبتنی بر نقش:** نداشتن مکانیزم‌های کنترل دسترسی مبتنی بر نقش که سطح دسترسی کاربران را بر اساس نقش‌هایشان مدیریت کند.
- **مدیریت نامناسب نشست‌ها:** نداشتن مدیریت صحیح نشست‌ها که می‌تواند منجر به سوءاستفاده از نشست‌های فعال شود.

4-7- خطر Inappropriate Encryption

رمزگذاری نامناسب یکی از مشکلات جدی امنیتی است که می‌تواند در پایگاه‌های داده SQL و NoSQL به وقوع بپیوندد و اطلاعات حساس را به خطر بیندازد :

الگوریتم‌های ضعیف رمزگذاری: استفاده از الگوریتم‌های ضعیف مانند DES که به راحتی قابل شکستن هستند. این الگوریتم‌ها به دلیل ضعف در استانداردهای رمزنگاری امروزی، نباید برای حفاظت از اطلاعات حساس استفاده شوند.

استفاده از کلیدهای رمزگذاری کوتاه: استفاده از کلیدهای رمزگذاری کوتاه مانند کلیدهای 56 بیت به جای کلیدهای قوی‌تر مانند کلیدهای 256 بیت. استفاده از کلیدهای کوتاه می‌تواند باعث کاهش مقاومت در برابر حملات کرکرهای رمزگشایی شود.

پیاده‌سازی نامناسب : اشتباهات در استفاده از کتابخانه‌های رمزنگاری، عدم استفاده از حالت‌های امن مانند CBC به جای ECB برای الگوریتم‌های بلوکی، و استفاده نادرست از عملیات رمزگذاری. این مشکلات می‌توانند منجر به آسیب‌پذیری‌های امنیتی شوند.

مشکلات عملکرد رمزگذاری : بار اضافی از رمزگذاری قوی که بر عملکرد پایگاه داده تأثیر می‌گذارد. استفاده از رمزگذاری قوی می‌تواند باعث کاهش عملکرد و بازدهی پایگاه داده شود.

برای حفظ امنیت اطلاعات در پایگاه‌های داده، لازم است از الگوریتم‌های رمزگذاری قوی و استاندارد، استفاده از کلیدهای رمزگذاری بلند، پیاده‌سازی صحیح الگوریتم‌ها، و مدیریت مناسب عملکرد رمزگذاری اطمینان حاصل کرد. با این اقدامات، می‌توان بهبود امنیت پایگاه‌های داده را فراهم آورد و از بروز حملات و نقض‌های امنیتی جلوگیری کرد.

4-8- خطر Buffer Overflow

سرریز بافر یکی از آسیب‌پذیری‌های رایج در برنامه‌نویسی است که می‌تواند منجر به مشکلات امنیتی جدی شود. در این حالت، داده‌ها به حافظه‌ای که برای آن‌ها تخصیص داده نشده وارد می‌شوند و می‌توانند عملکرد برنامه را تحت تأثیر قرار دهند یا به مهاجمان امکان اجرای کد مخرب را بدهند. در پایگاه‌های داده SQL و NoSQL، دو نوع سرریز بافر وجود دارد:

سرریز بافر مبتنی بر پشته (Stack-based Buffer Overflow)

این نوع، رایج‌ترین نوع سرریز بافر است. زمانی رخ می‌دهد که پشته، که یک ناحیه حافظه برای ذخیره متغیرهای محلی و کنترل ترتیب اجرای دستورات است، سرریز می‌شود. این اتفاق می‌تواند منجر به خرابی برنامه و اجرای کدهای غیرمجاز شود.

سرریز بافر مبتنی بر هیپ (Heap-based Buffer Overflow)

این نوع سرریز در ناحیه هیپ رخ می‌دهد، که حافظه‌ای برای تخصیص پویا است. در این نوع، داده‌ها می‌توانند به حافظه‌ای که به دیگر اشیاء اختصاص داده شده وارد شوند و باعث تخریب داده‌ها و رفتار نامطلوب برنامه شوند.

برای جلوگیری از وقوع سرریز بافر، می‌توان از روش‌های زیر استفاده کرد:

- نگهداری و مدیریت حافظه: استفاده از تکنیک‌های مناسب مدیریت حافظه برای جلوگیری از سرریز.
- اعتبارسنجی ورودی‌ها: بررسی و اعتبارسنجی داده‌های ورودی تا اطمینان حاصل شود که از اندازه‌های مجاز تجاوز نمی‌کنند.
- مدیریت صحیح حافظه: استفاده از ابزارها و روش‌هایی که به مدیریت صحیح حافظه کمک می‌کنند.
- فیلتر کردن داده‌ها: فیلتر کردن داده‌های ورودی به منظور جلوگیری از ورود داده‌های نامعتبر و مخرب.

سرریز بافر یک مشکل جدی امنیتی است که می‌تواند به مهاجمان اجازه دهد کنترل برنامه را به دست گیرند. با استفاده از روش‌های پیشگیری مناسب مانند مدیریت صحیح حافظه، اعتبارسنجی ورودی‌ها، و فیلتر کردن داده‌ها، می‌توان از بروز این نوع آسیب‌پذیری جلوگیری کرد و امنیت سیستم‌ها را افزایش داد.

مراجع

[1] https://ibm.com/topics/database-security

[2] https://owasp.org/

[3] https://www.proofpoint.com/us/threat-reference/privilege-escalation

[4] https://janmuhammadzaidi.medium.com/vertical-privilege-escalation-the-user-can-takeover-an-admin-account-via-response-manipulation-9237c8b2fefa

Abstract

In this report, we examine the vulnerabilities and risks in SQL and NoSQL databases. Initially, we highlight the significance and wide-ranging applications of databases. In the second chapter, we provide a comprehensive overview of SQL and NoSQL databases, discussing their differences and advantages. Following this, the third chapter delves into analyzing various types of database vulnerabilities. We address the dangers related to SQL/NoSQL Injection, Insecure Direct Object Reference (IDOR), data inconsistency, database system misconfiguration, privilege escalation, weak authentication and authorization, inappropriate encryption, and buffer overflow. This report aims to raise awareness and offer security solutions to mitigate the risks associated with these vulnerabilities.

Keywords: *Database, SQL, No-SQL, Security, Vulnerability*

University of Tehran

College of Engineering

School of Electrical and
Computer Engineering

Vulnerabilities in SQL and No-SQL Databases

Report paper for the final presentation of the Network Security Fundamentals course

By:

Ali Mehrani - 810198542

Taha EbrahimZadeh Miyab - 810198335

Ali Eftekhari - 810198549

Supervisor:

Dr. Sayyad Haghighi