

گزارش بررسی پیکربندی نامناسب سرویس‌های NoSQL در مقیاس پروژه‌های در سطح جهان به تفکیک ابزارها و موتورهای NoSQL

علیرضا سلطانی نشان استاد راهنما: آقای دکتر شجاعی مهر

۹ دی ۱۴۰۲

دانشگاه آزاد اسلامی، واحد تهران-شمال، دانشکده فنی مهندسی کامپیوتر، گرایش مهندسی نرم‌افزار، مقطع کارشناسی
ارشد

۱ مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLv3 منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می‌دهد.

۲ تعریف مسئله

ذخیره‌سازی اطلاعات از مهم‌ترین نیازهای تحلیل‌کنندگان داده است. امروزه با توجه به پیشرفت صنعت IoT و یادگیری ماشین، تولید داده‌ها بسیار افزایش یافته است به گونه‌ای که بتوان این داده‌ها را به سریع‌ترین روش ممکن در محلی مناسب ذخیره‌سازی و نگهداری کرد. افراد برای ذخیره‌سازی این داده‌ها نیاز به نصب و راه‌اندازی یک سیستم DBM دارند که از طریق یک واسط با زبانی مناسب بتوانند به آن متصل شده و داده‌های دریافتی را بعد از تجزیه و تحلیل آنها در این محل ذخیره‌سازی و مدیریت کنند. امروزه محققان ترجیح می‌دهند به دلیل مقیاس پذیری بیشتر، سیستم‌های توزیع شده و قابلیت پایداری بالا از دیتابیس‌های رابطه‌ای به سمت دیتابیس‌های NoSQL مهاجرت کنند. این نوع دیتابیس‌ها امروزه توسط تمام اپلیکیشن‌های جدید پشتیبانی می‌شوند و برای استفاده آسان طراحی شده‌اند. حتی می‌توان متذکر شد که تعداد زیادی از سرویس‌های ذخیره‌سازی ابری امروزه از سرویس‌های دیتابیس NoSQL پشتیبانی گسترده‌ای دارند. این ارائه دهندگان اغلب شرکت‌های معروفی مانند Amazon DynamoDB Google Cloud Database MS Azure CosmosDB می‌باشند. همچنین بیشتر این موتورهای دیتابیس به صورت متن‌باز هستند و توسعه دهندگان زیادی از سرتاسر جهان روی آنها مشغول توسعه هستند.

در سال‌های اخیر، با پدید آمدن و رشد سریع سرویس‌های دیتابیس NoSQL بین عموم توسعه‌دهندگان استفاده از این نوع سرویس‌ها افزایش یافته است. دلیل اصلی این محبوبیت نصب و راه‌اندازی و استقرار آسان آنها در هر محلی است. همچنین قابل اعتماد هستند، روش‌ها و مکانیزم‌های زیادی برای تهیه نسخه‌های پشتیبان‌گیر به صورت منظم از داده‌ها را ارائه می‌دهند. دلیل اصلی آسان بود این سیستم آن است که در هنگام راه‌اندازی آنها زمان زیادی را صرف نمی‌کنید، زیرا بعد از نصب اولیه و طی کردن فرایند نصب با زدن روی دکمه "بعدی" دیتابیس شما آماده‌ست و می‌توانید از آن در برنامه خود استفاده کنید. بعد از این فرایند هیچ عملیاتی بر روی تعریف دسترسی‌ها، مدیریت کاربران در استفاده از دیتابیس مانند

اختصاص سطح دسترسی، توسط راهانداز سیستم DBM صورت نمی‌گیرد. نتیجه این موارد پیکربندی غیر اصولی و اشتباه^۱ سیستم ذخیره‌سازی داده می‌شود که در نتیجه افشای اطلاعات حساس^۲ را به دنبال خواهد داشت.

سوالی که ممکن است در اینجا مطرح شود آن است که چه زمانی پیکربندی نادرست موجب افشای اطلاعات می‌شود؟ در ابتدا بعد از راهاندازی این نوع دیتابیس‌ها اولین هدف استفاده از آنها در محیط لوکال در یک شبکه است. اما افشای اطلاعات و پیکربندی اشتباه زمانی رخ می‌دهد که این دیتابیس‌ها در شبکه اینترنت مورد دسترسی قرار گیرند.

محققان با توجه به موارد گفته شده بالا توانسته‌اند یک ابزار خودکار جهت آنالیز و جست و جوی سیستم‌های دیتابیس NoSQL را توسعه دهند که به وسیله آن می‌توانند پیکربندی نامناسب این سیستم‌های مستقر شده را متوجه شده، موارد آسیب‌پذیری را گزارش و سپس به صاحبان این دیتابیس‌ها هشدار در جهت در خطر بودن اطلاعاتشان ارسال کنند.

در این گزارش به طور خلاصه تمام موارد انجام شده را در پنج عنوان توضیح می‌دهیم. در ابتدا در مورد چالش‌ها و نحوه تحقیق روی این آسیب‌پذیری‌ها و عدم وجود پیکربندی مناسب می‌پردازیم. در بخش مدل پیشنهادی بیشتر ماهیت ابزار توسعه داده شده را مطرح می‌کنیم و سپس نتایج اجرای این ابزار را نمایش می‌دهیم و در نهایت به نوآوری و کارهای آینده می‌پردازیم.

۳ چالش‌ها

ابزاری توسعه داده شده است که در یک رنج گسترده‌ای از آدرس‌های IP می‌تواند اینگونه دیتابیس‌ها را اسکن کند و افشای سرویس آنها را تشخیص دهد. این تشخیص به شکل ایمن بدون هیچ نگهداری داده‌ها و یا افشای اطلاعات حساس آنها صورت می‌گیرد. بررسی ضعف پیکربندی‌های صورت گرفته بر روی ۶۷ میلیون ۷۲۶ هزار و ۶۴۱ آدرس IP بوده است که بین بازه زمانی اکتبر ۲۰۱۹ و مارچ ۲۰۲۰ تکمیل شده است. نکته جالب از آنجایی شروع می‌شود که این سرویس‌ها نه تنها به صورت شخصی راهاندازی شده‌اند بلکه تعداد ۱۲ هزار و ۲۷۶ نمونه از آنها در ارائه دهندگان سرویس‌های ابری معروف یافت شده است. با توجه به این موضوع در این تحقیق ۷۴۲ مورد آسیب‌پذیری پیدا شده است که به صورت مستقیم وب سایت این کاربران به دلیل ضعف در پیکربندی به دیتابیس‌های آنها ارجاع دارد این بدان معناست با وجود تنظیمات و پیکربندی پیش فرض و بدون هیچ گونه استراتژی امنیتی، هر کاربر ناشناس دیگری می‌تواند وارد این دیتابیس‌ها شده و آنها را با نظر و سلیقه خودش تغییر و حتی تخریب به قصد اخاذی کند.

۱.۳ بررسی نمونه‌ها در پیکربندی ضعیف راهاندازی

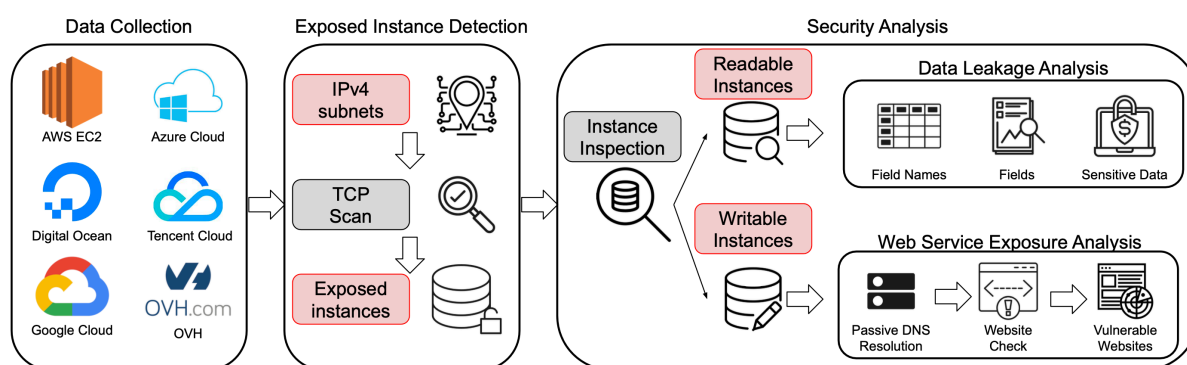
۱. در مارچ ۲۰۲۰، ۷ ترابایت از داده‌های سایت بزرگسالان به صورت صریح از یک نمونه دیتابیس Elastic Search با اطلاعاتی از قبیل، نام کاربران، جنسیت و گرایش‌ها، لاگ‌های مربوط به پرداخت‌هایشان، ایمیل، با ۱۰۸۸ میلیارد رکورد مورد افشا قرار گرفت.
۲. در نوامبر سال ۲۰۱۹ یک محقق توانست یک نمونه با پورت باز با بیشتر از ۱/۲ میلیارد رکورد از یک دیتابیس را پیدا کند که شامل اطلاعات حساس کاربران از قبیل آدرس ایمیل آنها بود.
۳. در ژانویه سال ۲۰۱۷، در یک حمله بیشتر از ۶۰۰ نمونه از دیتابیس Elastic search حذف شدند و برای بازیابی آنها از صاحبانشان اخاذی کردند [۴].
۴. براساس گزارشی در سال ۲۰۱۸ بیشتر از ده‌ها هزار نمونه از دیتابیس‌های Redis در دسترس کاربران مخرب، آسیب‌پذیر شناخته شدند که به دلیل دسترسی عموم افراد تعداد ۷۵۰۰ سرور یافت شد که در معرض خطر یک بدافزار

به نام Botnet بودند که هدف اصلی آنها دزدیدن ارزشهای دیجیتالی^۳ آن پلتفرم ارائه دهنده بود.

براساس موارد مطرح شده در بندهای گفته شده بالا، اولین بررسی از ضعف پیکربندی دیتابیسهای NoSQL انجام شده است به گونه‌ای که می‌توان از آن برای تشخیص و تعیین معیاری برای بررسی پیکربندی درست در این دیتابیس‌ها از آن استفاده کرد. محققان یک فریمورک توسعه داده‌اند که به صورت کاملاً خودکار می‌تواند سرویس‌های معرض دید عموم را تشخیص و عملیات بررسی امنیتی روی آنها انجام دهد بدون ذخیره‌سازی داده‌های کاربران یا باز کردن داده‌های دیتابیس پلتفرم‌ها و دریافت اطلاعات حساس آنها.

۲.۳ فرایند کلی عملکرد فریمورک [۱]

این فریمورک در ابتدا لیستی از آدرس‌های IP که توسط بیشتر ارائه دهندگان سرویس‌های ابری استفاده می‌شود را اسکن کرده و به دنبال ارتباطی باز بر روی پورت پیش فرض دیتابیس NoSQL می‌گردد که بتواند به آن به صورت مستقیم متصل شود. (در شکل ۱، می‌توانید عملکرد فریمورک را در تصویر مشاهده کنید). سپس می‌تواند به یک نمونه از دیتابیس دسترسی داشته و عملیات بررسی امنیتی خود را شروع کند. به طور کلی این فریمورک به بررسی سطح دسترسی دیتابیس (همان دسترسی‌های خواندن و نوشتن روی یک سیستم مدیریت دیتابیس) متا دیتا از قبیل نسخه مورد استفاده از سرویس NoSQL، کاربران مجاز دسترسی به دیتابیس، سطوح دسترسی تعریف شده و جداول مرتبط به این دیتابیس‌ها، می‌پردازد.



شکل ۱: بررسی عملکرد ابزار توسعه داده شده

۱.۲.۳ تشخیص عمل خواندن از دیتابیس‌های فاش شده

اگر این ابزار تشخیص دهد که دسترسی خواندن را از این دیتابیس‌ها دارد تضمین افشای اطلاعات این سیستم‌ها را به طور قطعی می‌دهد که می‌تواند خطری برای محتوای داخل دیتابیس باشد. ابزاری که توسعه داده شده است کاملاً ایمن می‌باشد چرا که اصلاً وارد محتوای این دیتابیس‌ها و داده‌های آنها نشده و تنها از توابعی مانند تابع Count برای شمارش رکوردهایی که مربوط به فیلدهایی مانند نام کاربران، شماره تلفن یا آدرس ایمیل آنها می‌شود، استفاده می‌کند. اغلب داده‌های جمع‌آوری شده از این دیتابیس‌ها به صورت نمایش تعداد رکوردهای آنها مربوط به فیلدی مشخص است که در جداول صفحات بعدی آنها را مشاهده خواهید کرد.

۲.۲.۳ تشخیص عمل نوشتن از دیتابیس‌های فاش شده

زمانی که این ابزار بتواند به این دیتابیس‌ها متصل شود و بعد از آن قادر به ساخت یک workspace یا یک رکوردی از داده NoSQL یعنی همان Document باشد، تشخیص می‌دهد که مجوز نوشتن را در این سیستم دارد به همین خاطر یک پیام

^۳Cryptocurrencies

جدی را برای صاحبان دیتابیس می‌نویسد تا در جریان ضعف پیکربندی و ایمن نبودن ارتباطات آنها و باز بودن دسترسی‌ها، قرار بگیرند. با این کار محققان از افشا و آسیب به نمونه از دیتابیس جلوگیری می‌کنند. داشتن دسترسی نوشتن یکی از خطرناک‌ترین دسترسی‌های این دیتابیس‌ها می‌باشد به طوری که این ابزار علاوه عملیات گفته شده بالا یک استراتژی دیگری را در پیش می‌گیرد و آن این است که به جست و جوی DNS های آن به صورت غیر فعال می‌پردازد تا متوجه آن شود که آیا روی این IP که دیتابیس مستقر شده است، منابع دیگری مانند برنامه‌های وب و وبسایت‌ها و دیگر سرویس‌ها مستقر شده‌اند یا خیر؟ چرا که اگر منابع وب را از این طریق پیدا کند به این معنی است که این سرورها پتانسیل حمله آسیب‌زننده‌ای که باعث دستکاری داده‌ها می‌شود را دارند. در تمام وضعیت گفته شده بالا با ارائه دهندگان سرویس‌های ابری ارتباط برقرار شده و به آنها در مورد آسیب‌پذیری‌های یافت شده گزارشی به عمل آمده است.

۶۷ میلیون آدرس IP اسکن شده در ارائه دهندگان سرویس‌های ابری مختلف بین اکتبر سال ۲۰۱۹ تا مارچ ۲۰۲۰، تعداد ۱۲،۲۷۶ سرویس دیتابیسی با دسترسی‌های مختلف یافت شدند که ۸۷٪ آنها با دسترسی آزاد خواندن و نوشتن و ۸/۶٪ آنها تنها قابلیت خواندن اطلاعات را داشتند. بین این بررسی محققان مواردی از قابل دسترس بودن اطلاعات فقط خواندنی این دیتابیس‌ها پیدا کردند که ۷۴۲ نمونه پتانسیل افشای اطلاعات حساس کاربران مانند آدرس ایمیل، نام‌ها، گذرواژه‌ها و تمام منابعی که می‌تواند در اپلیکیشن‌های وب آنها استفاده شود، را داشتند. علاوه بر این ما دیتابیس‌های مختلفی را پیدا کردیم که توانایی افشای فایل‌های مهم و حساس مانند فایل‌های سرتیفیکیت سایت‌ها و لاگ‌های مربوط به آنها را داشتند. بین تمام سیستم‌های DBM سرویس MongoDB بیشترین مقدار ضعف پیکربندی را داشت به گونه‌ای که ۴،۸۵۹ نمونه از آن یافت شد و این سهم برای دیتابیس Elasticsearch به مقدار ۴،۷۲۵ نمونه بود.

۳.۲.۳ مرور سناریوهای تهدیدآمیز

افشای اطلاعات (سطح دسترسی خواندن)

زمانی که منابع دیتابیسی به صورت غیر عامدانه‌ای مورد دسترسی عموم قرار می‌گیرد که موجب مسائل شکسته شدن حریم خصوصی کاربران و افشای اطلاعات حساس و عدم محرمانگی می‌شود.

آلوده شدن منابع وب (سطح دسترسی نوشتن)

زمانی که دسترسی نوشتن روی یک میزبان فعال باشد به معنای آن است که تمام محتوای آن میزبان را می‌توان دستکاری کرد. اغلب وب سایت‌ها به این ترتیب تغییر چهره روی آنها اعمال می‌شود که مربوط به عملیات دستکاری Deface کردن این پایگاه‌های اطلاعاتی است. همچنین این عمل باعث تاثیر روی محتوای این وبسایت‌ها خواهد شد چرا که می‌توانند وارد دیتابیس شده و اطلاعات مربوطه را دستکاری کنند و به نفع خودشان ویرایشی انجام دهند. همچنین آسیب‌پذیری‌های دیگر نیز می‌تواند رخ دهد. برای مثال بعد از دسترسی نوشتن روی این میزبان‌ها می‌توانند از طریق وبسایت یک فایل مخرب و آلوده را قرار داده و کاربران آن را به عنوان فایل مورد نظر بارگیری کرده و باعث آلوده شدن دستگاه کاربران نهایی شود.

۴.۲.۳ اخاذی در ازای اطلاعات

مهاجمان می‌توانند با داشتن دسترسی نوشتن روی این دیتابیس‌ها حمله‌ای انجام دهند که موجب اخاذی از صاحبان اطلاعات شود. معمولاً استراتژی مهاجمان در این خصوص از بین بردن اطلاعات یا رمزنگاری آنها می‌باشد که در ازای اخاذی از صاحبان دیتابیس یا داده می‌توانند داده‌ها را به آنها برگردانند یا آنها کلید رمزنگاری آن داده‌ها را تحویل دهند.

محققان چهار تا از محبوب‌ترین دیتابیس‌های NoSQL را مورد بررسی قرار دادند تا نشان دهند که تحقیقات آنها کافی بوده و تقریباً مهم‌ترین سرویس‌های NoSQL را پوشش داده است. محققان تحقیقاتی را نسبت به محبوب‌ترین سیستم‌های دیتابیسی براساس وب سایت db-engines.com به عمل آوردند. مهم‌ترین سوال آن است که چگونه یک سیستم به عنوان محبوب‌ترین سیستم دیتابیسی انتخاب می‌شود؟

انتخاب این دیتابیس‌ها براساس درصد استفاده آنها در وبسایت‌ها، بحث و گفت و گوهای گروه‌های فنی، پیشنهادات شغلی در رابطه با متخصص مربوط به این دیتابیس‌ها و ارتباطشان در شبکه‌های اجتماعی می‌باشد. براساس جدول ۱، رنک دیتابیس‌های مختلف براساس سایت db-engines آمده است. لازم به ذکر است که این جدول نسبت به جدول داخل مقاله به روز شده که طی ۳ سال گذشته دیتابیس‌های Redis و Elasticsearch به ترتیب مقال ۶ و ۷ را بدست آوردند. در حالی که بین سال ۲۰۱۹ تا ۲۰۲۰ مقام Redis و Elasticsearch به ترتیب ۷ و ۸ بود [۴].

۵.۲.۳ اهداف تحقیق

اهداف این تحقیقات به شرح زیر می‌باشد:

۱. نتیجه عدم تدابیر امنیتی
 ۲. بررسی تاثیر ضعف پیکربندی
 ۳. افزایش آگاهی برای جلوگیری از فاش شدن و دستکاری اطلاعات
- همچنین در بخش‌های بعدی در مورد آگاهی از نگرانی‌های اخلاقی مطرح شده است که در آن به جمع‌آوری داده‌های نتیجه این آزمایشات صرفا برای بررسی محاسبات محققان نسبت به آسیب‌پذیری داده‌ها در آینده است.

جدول ۱: رنکینگ موتورهای دیتابیس: ۱۰ دیتابیس محبوب از نظر سایت db-engines

رتب			دیتابیس	مدل	امتیاز		
۲۰۲۳ نوامبر	۲۰۲۳ اکتبر	۲۰۲۲ نوامبر			۲۰۲۳ نوامبر	۲۰۲۳ اکتبر	۲۰۲۲ نوامبر
۱	۱	۱	Oracle	R	1277.03	+15.61	+35.34
۲	۲	۲	MySQL	R	1115.24	-18.07	-90.30
۳	۳	۳	MSSQL Server	R	911.42	+14.54	-1.09
۴	۴	۴	PostgreSQL	R	636.86	-1.96	+13.70
۵	۵	۵	MongoDB	NS	428.55	-2.87	-49.35
۶	۶	۶	Redis	NS	160.02	-2.95	-22.03
۷	۷	۷	Elasticsearch	NS	139.62	+2.48	-10.70
۸	۸	۸	IBM Db2	R	139.62	+2.48	-10.70
۹	۹	۱۰	SQLite	R	124.58	+1.13	-13.56
۱۰	۱۰	۹	Microsoft Access	R	124.49	+0.18	-10.53

۴ مدل پیشنهادی

۱.۴ جمع‌آوری داده با پیدا کردن آدرس‌های IP سرویس دهندگان

اولین مرحله از این رویکرد جمع‌آوری داده با استفاده از لیست آدرس‌های IP نسخه ۴ بوده است که می‌تواند امکان داشته باشد روی هر کدام از آدرس‌ها حداقل یک نمونه دیتابیس NoSQL وجود داشته باشد. همچنین محققان لیستی از ارائه دهندگان سرویس‌های ابری را مطرح کردند که در آنها امکان نصب و راه‌اندازی این نوع دیتابیس‌ها میسر بوده است. این ارائه دهندگان به کاربران اجازه می‌دهند تا سرویس‌های دیتابیس خود را در شبکه اینترنت مستقر کنند و یک Connection String معتبر برای دسترسی آنها به اپلیکیشن خود راه‌اندازی نمایند. این سرویس‌ها عبارت‌اند از:

- AmazonEC2
- Microsoft Azure Cloud
- Google Cloud
- Tencent Cloud
- DigitalOcean
- OVH

هر کدام از نمونه ارائه‌دهندگان سرویسی که در بالا نام برده شد، قابلیت آن را دارند که کاربر بتواند در آن به صورت دستی دیتابیس NoSQL مورد نظر خود را نصب و پیکربندی کند. اما باید توجه داشت بعد از نصب اولیه این دیتابیس‌ها روی این سرویس‌ها هیچ پیکربندی در رابطه با کنترل دسترسی و تغییر شماره پورت و غیره انجام نمی‌شود که این به خودی خود نشان دهنده پیکربندی ضعیف این دیتابیس‌ها می‌باشد. اکثر تقاضا برای راه‌اندازی اولیه این دیتابیس‌ها صرفاً جهت داشتن فرایند آزمایشی توسط هر توسعه دهنده تازه کار است. دیگر به آن روند مهم پیکربندی توجه نمی‌کند و بعد از آن ممکن است داده‌های مهمی را بدون توجه به ضعیف بودن پیکربندی در دیتابیس خودش انتقال دهد.

بین شش سرویس ابری بالا، از سه مورد آنها (AmazonEC2، Microsoft Azure Cloud، Google Cloud) محققان توانستند به subnet آدرس پابلیک IP برسند. اما سه سرویس آخر یعنی (OVH، DigitalOcean، Tencent Cloud) آدرس IP پابلیک خود را ارائه نمی‌داند. برای دریافت اطلاعات مورد نظر محققان آدرس‌های IP را از ipinfo.io بدست آوردند و سپس بعد از توانستند به زیر آدرس‌های شبکه مورد نظر دسترسی پیدا کنند و برنامه خود را براساس لیست بدست آمده اجرا کرده و پیکربندی دیتابیس‌های مطرح شده را بررسی و بعد از آن گزارشی را تهیه کنند.

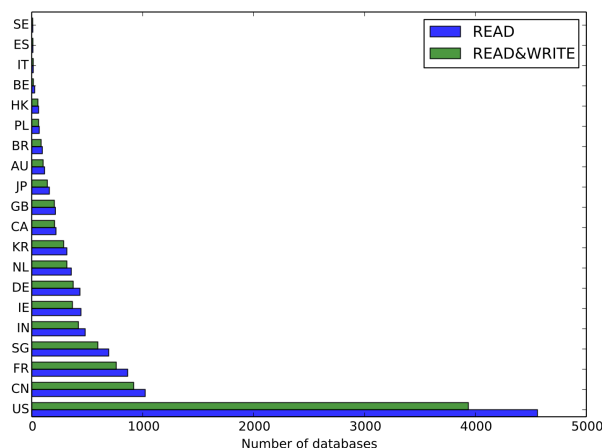
۲.۴ شناسایی نمونه‌های افشا شده

در مرحله دوم رویکرد، محققان موضوع پیدا کردن دیتابیس افشا شده در فضای عمومی اینترنت را بررسی کردند. با استفاده از لیست آدرس‌هایی که در مرحله اول بدست آمده است [؟]، محققان با به کارگیری Nmap آدرس‌های IP را برای یافت پورت‌های باز سرویس NoSQL بررسی می‌کنند. در نتیجه این مرحله دو حالت به وجود می‌آید:

۱. نتیجه Close را نمایش می‌دهد. این بدان معناست که سرور توسط آدرس IP در دسترس بوده است اما با استفاده از پورت پیش فرض دیتابی، برنامه محققان نتوانسته است به آن سرویس NoSQL متصل شود.

۲. نتیجه Filtered را نمایش می‌دهد. این بدان معناست که آدرس سرور و شماره پورت پیش فرض که این شماره پورت مربوط به یکی از سرویس‌های NoSQL است بر روی آن هیچ سرویس دیتابیس NoSQL مستقر نشده است و بجای آن اپلیکیشن‌های دیگری از این شماره پورت پیش فرض استفاده می‌کنند.

پس در این مرحله می‌توان نتیجه گرفت که برنامه محققان تمام آدرس‌هایی را بررسی کردند که بر روی شماره پورت پیش فرض آن‌ها یکی از دیتابیس‌های NoSQL مستقر شده است تا بتوانند ضعف پیکربندی آن‌ها را در این مقاله بیشتر مورد بررسی قرار دهند.



شکل ۲: ۲۰ رتبه اول دیتابیس‌های استفاده شده در مناطق جغرافیایی مختلف

۳.۴ بررسی‌های امنیتی

فاز اصلی و عملی این تحقیق می‌باشد. به گونه‌ای که به سه وظیفه برای جست و جوی نمونه‌های باز دیتابیس‌های NoSQL انجام گرفته است:

۱. ساخت یک نمونه یا Instance از دیتابیس مربوطه برای اتصال به آن و بررسی‌های اولیه ورود بدون احراز هویت
۲. بررسی و ورود به داده‌ها برای دریافت اطلاعات بدون دسترسی به محتوای اصلی آنها برای اثبات انتشار داده‌های حساس کاربران
۳. در معرض دید عموم قرار گرفتن سرویس‌های وب

بررسی و نمونه‌گیری از دیتابیس جهت اتصال و انجام عملیات

برای هر یک از آدرس‌های IP که در مرحله قبل بدست آورده شد، این فریمورک یک نمونه به ازای هر سرویس NoSQL ایجاد می‌کند تا ضعف پیکربندی را مورد بررسی قرار دهد.

در ابتدا این ابزار در دیتابیس‌های مذکور لاگین کرده و سطوح دسترسی^۴ آنها را مورد بررسی قرار می‌دهد. اگر این ابزار نتواند به صورت مستقیم مجوزهای خواندن و نوشتن کاربران تعریف شده در دیتابیس را دریافت کند وارد استراتژی دوم شده و یک workspace جدید در دیتابیس افشا شده ایجاد می‌کند. اگر نتیجه ایجاد این workspace موفقیت آمیز بود این بدین معناست که این دیتابیس دسترسی نوشتن را برای تمام کاربران چه داخلی و چه خارج از سیستم دارد. بعد از این ایجاد این workspace ابزار سعی می‌کند که یک پیام هشدار برای صاحبان دیتابیس ایجاد کند در این پیام در مورد پیکربندی اشتباه و ضعیف این نمونه توضیح خواهد داد و صاحبان دیتابیس را با این پروژه آشنا می‌کند که قصد هیچ تخریب اطلاعاتی ندارد بلکه

^۴ Access permission

می‌خواهد آنها را از بابت این نقص‌ها با خبر سازد که هر چه زودتر با سیاستی مناسب از این دسترسی‌ها به کاربران خارج از سیستم جلوگیری انجام شود.

بررسی نشت داده

برای بررسی نشت اطلاعات، اگر ابزار بتواند بررسی کند که دسترسی خواندن را دارد که آن را اعلام می‌کند در غیر این صورت به صورت مستقیم به دیتابیس متصل شده و تمام متادیتاها از قبیل ورژن کنونی دیتابیس، نقش‌های کاربران تعریف شده در دیتابیس و همچنین نام تمام تسون‌ها (ویژگی‌های داده) را دریافت می‌کنند. لازم به ذکر است که این ابزار این داده‌ها را با استفاده از توابع داخلی دیتابیس مانند استفاده از Regular Expression توابع شمارشی و غیر بدست می‌آورد که هیچ داده‌ای از کاربران را در ابزار تجزیه و تحلیل نکند. این داده‌ها از قبیل آدرس ایمیل، حساب‌های شبکه‌های اجتماعی، شماره تلفن‌ها شماره حساب‌ها و گذرواژه هستند. این نتیجه مقادیر این داده‌ها را می‌توانید در جدول ۲ مشاهده کنید.

بررسی در معرض دید قرار گرفتن سرویس‌های وب

در این مرحله پتانسیل احتمال آسیب پذیری سرور افشا شده در برابر حمله‌های مبتنی بر وب مورد بررسی قرار گرفته است. در عمل با استفاده از ابزاری به نام VirusTotal آدرس‌های IP با دسترسی عمومی را وارد این برنامه کرده و تمام وب سایت‌هایی که روی این IP مستقر شده اند را بر می‌گرداند. در نهایت محققان به این نتیجه رسیدند که ممکن است بر روی نمونه دیتابیس NoSQL مستقر شده روی یک IP با دسترسی عمومی، وب سایت مربوط به این نمونه‌ها نیز مستقر شده باشد. اگر وب سایت به صورت عمومی از این روش در دسترس همه افراد باشد (با ارسال درخواست به سمت این وب سایت‌ها وضعیت درخواست شما ۲۰۰ خواهد بود). محققان به این نتیجه رسیدند که می‌توانند از طریق این اتصال بین وب سایت و دیتابیس به منابع دیگر آنها دسترسی داشته باشند.

جدول ۲: آنالیز نشت محتوای حساس

Category	FileType	MongoDB	Elasticsearch	cassandra	Total No.
Text/Data	.log	۳۵۰,۲۴۳	۶۹,۹۶۵,۶۱۳	۳,۳۲۲	۷۰,۳۱۹,۱۷۸
	.zip	۸,۷۹۰,۳۰۱	۱۸۱,۷۴۵	۳۳۵	۸,۹۷۲,۳۸۱
	.json	۷۰,۶۸	۵,۶۱۲,۰۷۲	۷,۴۱۳	۵,۶۸۹,۵۵۳
	.xml	۲۰,۴۰,۵۳۳	۹۸۱,۲۳۳	۴,۴۸۰	۳۰,۲۶,۲۴۶
	.txt	۱۵۰,۳۷۹	۱,۴۵۹,۴۹۵	۱,۷۳۳	۱,۶۱۱,۶۰۷
	.pdf	۸۱۱,۰۰۳	۷۵۶,۵۱۸	۱۰۰,۰۹۶	۱,۵۷۷,۶۱۷
	.text	۲۷,۲۱۹	۴۴۳,۹۹۰	۱۳۲	۴۷۱,۳۴۱
	.gz	۶,۴۱۶	۱۹۱,۳۸۷	۱۳۲	۱۴,۴۲۶
	.docx	۱۸,۹۱۰	۴۵,۳۷۵	۸۲۴	۶۵,۱۰۹
Image/Video	.jpg	۲۲,۷۷۲,۶۲۳	۲۵,۲۶۷,۹۳۴	۷۳۹,۵۰۸	۴۸,۷۸۰,۰۶۵
	.png	۴,۵۵۸,۳۷۲	۶۰,۷۲,۵۵۸	۱۱۲,۵۶۱	۱۰,۷۴۳,۴۹۱
	.jpeg	۶۱۳,۲۰۲	۷۰۷,۴۶۱	۳۴,۳۴۸	۱,۳۵۵,۱۲۱
	.gif	۵۹۴,۴۲۸	۵۸۱,۳۸۲	۱,۹۲۶	۱,۳۳۵,۱۲۱
	.mp4	۴۲۹,۰۳۸	۴۷۰,۲۳۱	۴,۳۳۳	۹۰۳,۶۰۲
	.webp	۳۹,۹۹۴	۷۵,۱۶۳	۲۴۹,۶۰۹	۳۶۴,۷۶۶
Code	.html	۱,۶۱۸,۰۱۰	۱۱,۶۸۴,۱۸۳	۲۷۲,۸۸۳	۱۳,۵۷۵,۰۷۶
	.ts	۸,۳۸۳	۳۳۵,۵۳۹	۱۱,۱۳۲	۳۵۵,۰۵۴
	.js	۵۸,۲۱۳	۱۳۵,۱۶۷	۱,۸۹۸	۱۹۵,۲۷۸
	.hll	۴۷	۵۸۵	-	۶۳۲
File Keys	.key	۱۰,۴۴۴	۱۰,۸۳۲,۴۶۴	۲,۶۲۰	۱۰,۸۴۵,۵۲۸
Crypto Files	.pem	۲۴۷	۱۳۰,۳۷۰	۱۳	۱۳۰,۶۳۰
	.pfx	۸۹	۱,۲۵۸	۴	۱,۳۵۱
	.p1۲	۳۱	۴۵۷	۷	۴۹۵
DB	.sql(Dumpls)	۲,۲۷۰	۲۳۷,۳۰۶	۱۸۶	۲۳۹,۷۶۲
Backup	.bak	۱,۴۸۷	۵,۳۶۴	۳۰۰	۷,۱۵۱
Password	.kdb(Keepass)	۱۰	۴۲	-	۵۲

۵ آزمایش‌ها و تحلیل نتایج

در این قسمت به نحوه عملکرد کلی فریمورک توسعه داده شده توسط محققان می‌پردازیم. همان طور که بالاتر بارها اشاره شد، این فریمورک طی تمام آزمایش‌ها، محققان اصلاً ماهیت داده‌های دیتابیس و محتوای آنها را مورد بررسی قرار نداده بلکه تماماً روی دسترسی در استفاده از داده‌ها تأکید زیاد داشتند.

در این بخش با نمایش یک نمونه کد، به طور کلی در مورد نحوه عملکرد این فریمورک صحبت خواهیم کرد.

```
1 # Connect to the database
2 mongo_client.connect(ip_address)
3 # Get database names
4 db_names = list_database_name()
5 # Mongo-shell method used to get role mappings
6 db.getRoles(showBuiltinRoles: false)
7 # Get collection names
8 collections = list_collection_names()
9 # Map-reduce function used to get field-mapping for each table
10 map = Code("function() {
11     for (key in this) {
12         emit(key, null);
13     }
14 }")
15 reduce = Code("function(key, stuff) { return null; }")
16 results = collection.map_reduce(map, reduce, "myresults")
17 # Query methods for field and object detection
18 count({ fieldname: {"$exists": True} })
19 count({ fieldname: {r'.*\.(fieldvalue)'} })
20 # Write the message in a new database
21 writedb = mongo_client[MONGO_DB]
22 writecollection = writedb[MONGO_COLLECTION]
23 writecollection.insert_one(MONGO_DOC)
```

۱.۵ بررسی قسمتی از فرایند مطرح شده

در خط ۲ با استفاده از آدرس IP حاضر در لیست آدرس‌ها، قصد اتصال به دیتابیس مانگو را به صورت ریموت داریم. سپس بعد از اتصال موفق می‌توانیم با استفاده از واسط Mongo Shell دستوراتی را برای انجام عملیاتی روی دیتابیس مورد نظر انجام دهیم. برای مثال در خط بعدی آن نام تمام دیتابیس‌هایی که در این آدرس IP وجود دارد را دریافت می‌کنیم و در متغیر dbnames قرار می‌دهیم. با استفاده از این عمل یعنی عاملی که توانسته به دیتابیس متصل شود قادر به خواندن محتوا بوده است که در این جا با اطمینان اعلام می‌کنیم که دسترسی خواندن در این دیتابیس برای یک کاربر عادی و خارج از سیستم وجود دارد. بعد از آن با استفاده از متد getRoles می‌توانیم تمام کاربرانی که در این دیتابیس تعریف شده‌اند براساس سطح دسترسی آنها، را دریافت کنیم. برای دسترسی به جداول (اصطلاحاً در دیتابیس‌های مانگو به آن کالکشن می‌گویند.) از تابع که نوشته‌ایم استفاده کردیم تا لیست تمام کالکشن‌های مربوط به یک دیتابیس را دریافت کنیم. برای آن که بتوانیم فیلدهای (کلیدهای) مربوط به یک کالکشن (مجموعه‌ای از داکيومنت‌ها) را دریافت کنیم از یک تابع Map-Reduce استفاده کردیم که تنها نام کلیدهای استفاده شده در این کالکشن را به ما برگرداند که بدایم دیتابیس شامل چه فیلدهایی است. در نهایت با استفاده از Regex توانستیم تمام داده‌های مربوط به متادیتا هر کالکشن را توسط الگوی "exists" برای تشخیص تعداد فیلدها جست و جو کنیم. در سه خط کد آخر می‌توان متوجه شد که ما می‌خواستیم که از داشتن دسترسی نوشتن در دیتابیس اطمینان حاصل کنیم که به همین خاطر سعی کردیم نتیجه تحقیقات خود را در یک کالکشن جدید در همان دیتابیس به عنوان یک داکيومنت جدید اضافه کنیم.

۲.۵ یادآوری تابع Map-Reduce [۱]

تابع Map-Reduce یک تابع دلخواه و قابل سفارشی‌سازی توسط برنامه‌نویس در زبان جاوا اسکریپت است (اساس کلی دیتابیس‌های مانگو بر پایه موتور NodeJS و زبان Javascript می‌باشد). که به واسطه آن عملیات مختلفی مانند بدست آوردن کلیدهای استفاده شده در یک داکيومنت، گروه‌بندی اجزا و یا دستکاری روی گروه‌بندی اجزا با استفاده از توابعی مانند `sum()` یا `count()` می‌باشد.

```
1 db.cardata.find()
2 { "_id": ObjectId("9dd74a***2340***"), "name": "Peugeot", "Series": "2", "qty": 10 },
3 { "_id": ObjectId("5f94*****"), "name": "BMW", "Series": "X", "qty": 112 },
4 { "_id": ObjectId("2e127a*****"), "name": "Peugeot", "Series": "5", "qty": 105 },
5 { "_id": ObjectId("8m113*****"), "name": "BMW", "Series": "M", "qty": 25 },
6
7 var map = function() { emit(this.name, this.qty) }
8 var reduce = function(key, value) { return Array.sum(values) }
9 db.cardata.mapreduce(map, reduce, "myResults")
10 db.myResults.find()
11 { "_id": "BMW", "value": 137 },
12 { "_id": "Peugeot", "value": 115 }
```

۳.۵ بررسی در انواع دیتابیس‌های اشاره شده

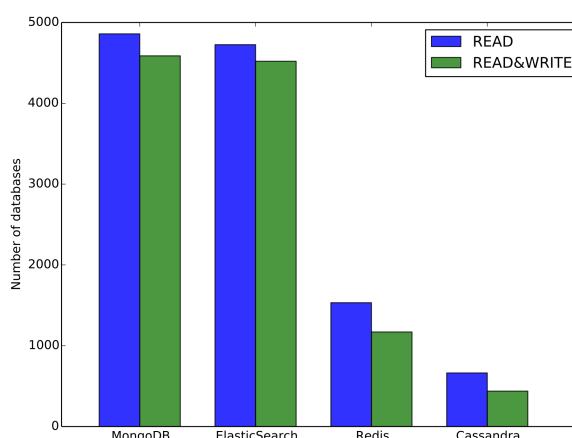
دقیقا همانند کد بالا در دو دیتابیس Cassandra و Elasticsearch به شکل مناسب برای انجام همان فرایندها، عملیات مورد نظر را اعمال کردیم اما در دیتابیس مموری Redis به شکل قبلی نتوانستیم عمل کنیم چرا که نمی‌توانستیم به طور مستقیم برخلاف سرویس‌های قبلی صحت دسترسی به لیست‌ها و کالکشن‌ها را بررسی کنیم. (این ویژگی در نسخه ۶ به بعد معرفی شد به گونه‌ای که در این مقاله امکانش میسر نبوده است). همچنین از آنجایی که در دیتابیس Redis نمی‌توان تعداد آبجکت‌ها و فیلدها را بدست آورد (به دلیل `key:value` بودن) از انجام این عملیات جلوگیری کرده‌ایم چرا که برای بدست آوردن هر فیلد ممکن بود که بتوانیم به مقدار آن فیلد دسترسی داشته باشیم که این قوانین ما را نقض می‌کند. در نهایت برای عدم آسیب رساندن به بقیه داده‌ها سعی کردیم یک کلید جدید ایجاد کنیم و پیام هشدار خود را در آن ایجاد کنیم. در زیر می‌توانید عملیات (کلی) انجام شده در دیتابیس Redis را بررسی کنید.

```
1 # Connect to the database
2 r = redis.Redis(ip_address)
3 # Get key names
4 keys = r.keys(pattern=u'*.')
5 # Get eventual number of occurrences of key fieldnames
6 occurrences = scan_iter(match="fieldname")
7 for match in occurrences:
8     counter += 1
9 # Write the warning message
10 r.append(key, value)
```

نتایج بدست آمده بر اساس آزمایشاتی که در بالاتر توضیح داده شد به تقنیک سرویس‌های NoSQL به طور دقیق در این بخش بررسی می‌شود. به طور کل ۶۷ میلیون آدرس IP برای این تحقیق جمع‌آوری شده است بطوری که سهم هر سرویس از این آدرس‌های IP در جدول ۳ آمده است.

جدول ۳: آدرس‌های IP یافت شده بین بازه اکتبر ۲۰۱۹ تا مارچ ۲۰۲۰

NoSQL Service name	IP Address Quantitiy
AmazonEC2	۴۱,۲۴۵,۶۴۸
AzureCloud	۱۳,۷۱۲,۷۶۱
Google Cloud	۷,۳۴۶,۹۴۴
OVH	۳,۰۱۹,۰۰۸
TencentCloud	۱,۳۸۳,۶۶۴
DigitalOcean	۱,۰۰۷,۶۱۶



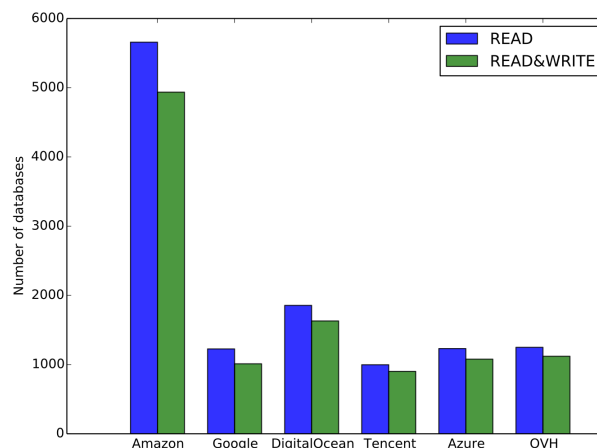
شکل ۳: بررسی پیکربندی سرویس‌های مختلف NoSQL

۴.۵ توزیع دیتابیس‌های NoSQL به تفکیک پارامترهای مختلف

جدول ۴: تعداد سرویس‌های افشا شده NoSQL

Misconfiguration	MongoDB	Elasticsearch	Redis	Cassandra	TotalNo.
Read	۴,۸۵۹	۴,۷۳۵	(۲,۰۲۹)۱,۵۳۲	۶۶۳	۱۲,۲۷۶
Read Only	۲۷۲	۲۰۵	۳۶۲	۲۲۵	۱,۰۶۴
Read & Write	۴,۵۸۷	۴,۵۲۰	۱,۱۷۰	۴۳۸	۱۰,۷۱۵

با توجه به نمودار شکل ۲ می‌توان دریافت که محبوبیت استفاده از دیتابیس‌های MongoDB نسبت به Cassandra بسیار زیاد است. (البته بر اساس آمار بدست آمده در سایت db-engines.com از سال ۲۰۲۰ تا نوامبر ۲۰۲۳) به همین خاطر کمتر توسعه دهنده یا شرکتی وجود دارد که از دیتابیس Cassandra استفاده کند. در نمودار شکل ۳ می‌توان متوجه شد که بیشتر توسعه دهندگان علاقه زیادی به استفاده از سرویس‌های ابری دارند که از دیتابیس‌های NoSQL پشتیبانی می‌کنند. همانطور که می‌توانید مشاهده کنید سرویس AmazonEC2 بیشترین میزان ضعف پیکربندی را در بین تمام سرویس‌ها داشته است (به تعداد ۵,۵۶۸ سرویس دیتابیس). همچنین در نمودار شکل ۴ می‌توانید تعداد دیتابیس‌های NoSQL را به تفکیک کشورهای مختلف مشاهده کنید. همانطور که انتظار می‌رفت بیشتر سرویس‌ها در ایالت متحد آمریکا مستقر شده‌اند که بیشترین مقدار تعداد افشای دیتابیس را در بین تمام کشورها دارند تقریباً ۴۴/۶٪ را به خودش اختصاص می‌دهد.



شکل ۴: بررسی ضعف پیکربندی دیتابیس‌های NoSQL در سرویس‌های ارائه دهنده خدمات ابری

۵.۵ بررسی معیار کیفی دیتابیس‌های فاش شده

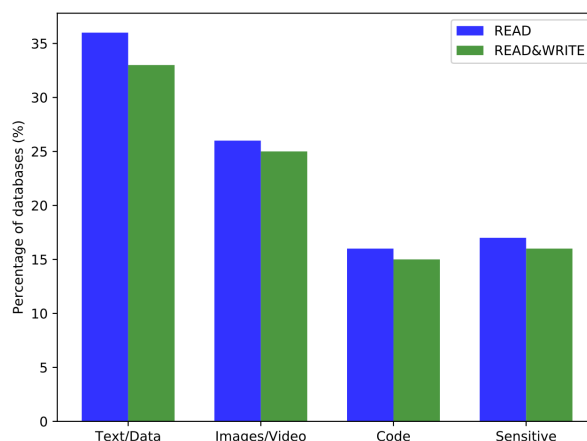
در طی گزارشی که در مورد این مقاله می‌نوشتیم همواره سوالی که در ذهنم وجود داشت این بود که چه معیاری برای بررسی کیفیت داده‌ها در طی این همه فرایند بررسی و تحقیق وجود داشته است؟ در این قسمت به بررسی این سوال می‌پردازیم. اما در کوتاه‌ترین جواب می‌توان گفت که تعداد زیادی از این دیتابیس‌های فاش شده در سرویس‌های NoSQL تنها برای آزمایش توسط برنامه‌نویسان بوده است نه همه برای استفاده در اپلیکیشن‌های بزرگ نهایی در دست کاربران! معیار اصلی برای سنجش کیفیت دیتابیس‌های فاش شده بر اساس حجم دیتابیس‌ها و متادیتای مربوط به جداول و فیلدها می‌باشد. اینکه هر چقدر دیتابیس سنگین‌تر باشد به معنای آن است که احتمالاً شامل جداول بسیار زیاد و داده‌های مختلف در آن می‌باشد. تعدادی از دیتابیس‌ها به حجم چند مگابایتی می‌رسیدند که بیشتر در آنها از کلماتی مانند Test استفاده شده بود یا اینکه کلاً خالی بودند. محققان این اطلاعات را از طریق توابعی متنوعی که در هر سرویس دیتابیس‌ای ارائه می‌شود (استفاده از درایورهای مانند pymongo و غیره) دست یافتند. همانطور که در قسمت ردیس گفته شد، در این دیتابیس (در نسخه‌ای که این مقاله نوشته شده است) نمی‌توان به این اطلاعات بدون نشت و دیدن داده‌ها، دست یافت.

۱.۵.۵ دیتابیس Mongo

میانگین اندازه حجم این دسته از دیتابیس‌های فاش شده به مقدار 299 ± 40 مگابایت می‌رسید که بیشتر از ۲۷۱ دیتابیس (۵/۵٪) به اندازه بیشتر از ۱۰۰ مگابایت می‌رسند. تعداد ۹۳ نمونه از این دیتابیس‌ها که حدوداً (۱/۹٪) را شامل می‌شد اندازه حجم بیشتر از ۵۰۰ مگابایت را داشتند. بین این بررسی، ۴۴۵ (۹/۱٪) حاوی کلمه test به عنوان نام کالشن‌ها بودند که تعداد ۲۶۲ نمونه از آنها (۵/۳٪) دارای اندازه حجم کمتر از ۱ مگابایت بودند.

۲.۵.۵ دیتابیس Elasticsearch

میانگین اندازه حجم این دیتابیس‌ها به طور کلی 109.04 ± 422.57 مگابایت بوده است. این دیتابیس نسبت به MongoDB حجم بیشتری از داده‌ها را نگهداری می‌کردند. ۶۵۲ نمونه از این دیتابیس‌ها (۱۳/۷٪) اندازه حجم بیشتر از ۱۰۰ مگابایت را داشتند. ۲۳۹ نمونه دیگر (۵/۲٪) اندازه حجم بیشتر از ۵۰۰ مگابایت را داشتند و در نهایت در این بررسی بیشتر از ۶۶۱ نمونه دارای کلمه test در نام اندیس‌هایشان بودند که ۱۲۰ نمونه از آنها حجم کمتر از ۱ مگابایت داشتند.



شکل ۵: درصد موجودیت داده‌های حساس دیتابیس‌ها در ۴ دسته‌بندی ذکر شده

۳.۵.۵ دیتابیس Redis

در این دیتابیس ۱۰۱ نمونه از کلمه test به عنوان نام کلید مورد نظر استفاده کرده بودند.

۴.۵.۵ دیتابیس Cassandra

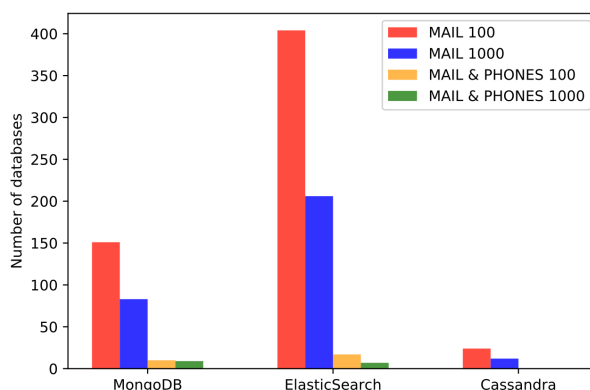
میانگین اندازه حجم این دیتابیس به اندازه 10.78 ± 33.61 مگابایت بوده است. ۲۱ دیتابیس (۳/۱٪) اندازه حجم بیشتر از ۱۰۰ مگابایت داشتند. همچنین ۱۸۰ دیتابیس (۲۶/۵٪) از کلمه test به عنوان نام جدول استفاده کرده بودند و در نهایت ۴۵ نمونه از آنها (۶/۶٪) کمتر از ۱ مگابایت اندازه حجمشان بود.

۶.۵ بررسی محتوای فاش شده

در جدول ۲ می‌توانید انواع فایل‌هایی که در این تحقیق در این ۴ دیتابیس فاش شده‌اند را مشاهده کنید. اما نکته جالب آن است که بیشترین مقدار یافت شده در مورد لاگ‌ها می‌باشد مخصوصاً در دیتابیس Elasticsearch. چرا که برخی شرکت‌ها برای راه‌اندازی یک سیستم مانیتورینگ ساده از ابزارهایی مانند Kibana، Elasticsearch Grafana، Logstash استفاده می‌کنند. برای نگهداری داده‌های مربوط به یک برنامه مانیتورینگ از دیتابیس Elasticsearch استفاده می‌شود و برای مدیریت و ذخیره‌سازی لاگ‌ها به صورت بحران‌های زمانی^۵ از ابزار logstash استفاده می‌شود. ۴۵۲ نمونه از دیتابیس Elasticsearch که شامل ۹/۵٪ از کل دیتابیس‌های تحقیق شده می‌باشد از فایل‌های log استفاده می‌کنند که مقدار این نوع از فایل‌ها ۵۰٪ از داده‌ها را تشکیل می‌دهد. مابقی داده‌ها با فرمت‌های webp mp۴ git jpeg png jpg از عنوان فایل‌های چندرسانه‌ای در این دیتابیس یافت شده است. در نمودار شکل ۴ می‌توانید درصد استفاده از دیتابیس‌ها از ۴ دسته‌بندی داده متنی، رسانه‌ای، کد و داده‌های حساس مانند نام کاربری، گذرواژه و حتی فایل‌های امنیتی وب را مشاهده کنید. در همین راستا، محققان برای پیدا کردن داده‌های حساس‌تر روی فیلدهای نام کاربری، گذرواژه و شماره تلفن در دیتابیس‌های Elasticsearch، MongoDB و Cassandra تحقیقاتی انجام دادند. علاوه بر این آنها به دنبال فیلدهای مخصوصی مانند حساب‌های کاربری در شبکه‌های LinkedIn و Facebook گشتند. نتیجه این قسمت را می‌توانید در جدول ۵ مشاهده کنید.

برای بررسی دقیق‌تر نشت داده‌هایی مانند ایمیل و شماره تلفن کاربران، محققان یک آستانه بین ۱۰۰ تا ۱۰۰۰ داده بین تعدادی از آدرس‌های ایمیل و شماره تلفن قرار دادند. بررسی آن در نمودار شکل ۶ آمده است. از آن می‌توان نتیجه گرفت

^۵Time Series Databases



شکل ۶: بررسی نشت داده، بین آستانه ۱۰۰ و ۱۰۰۰ ایمیل و شماره تلفن

که بین تمام دیتابیس‌های یافت شده در این بررسی تعداد کمی از آنها شامل داده‌هایی برای فیلدهای آدرس ایمیل و شماره تلفن هستند.

برای مثال در بین ۱،۱۲۶ دیتابیس MongoDB تنها ۱۵۱ دیتابیس دارای فیلد ایمیل هستند که بیشتر از ۱۰۰ داده در رابطه با ایمیل دارند و ۸۳ دیتابیس دیگر بیشتر از ۱۰۰۰ داده. در ۸۵۲ دیتابیس Elasticsearch در رابطه با فیلد آدرس ایمیل، تنها ۴۰۴ دیتابیس بیشتر از ۱۰۰ داده ایمیل دارند و ۲۰۶ دیتابیس دیگر بیشتر از ۱۰۰۰ ایمیل دارند. همچنین در ۴۵ دیتابیس Cassandra فقط ۲۵ دیتابیس بیشتر از ۱۰۰ داده ایمیلی دارند و ۱۲ دیتابیس دیگر بیشتر از ۱۰۰۰ داده ایمیلی را شامل می‌شوند.

جدول ۵: آنالیز و بررسی پتانسیل نشت پذیری داده‌های حساس کاربران

Fields	Elasticsearch		Cassandra		MongoDB	
	No.Databases	No.resources	No.Databases	No.resources	No.Databases	No.resources
E-mail	۸۵۲	۲،۱۸۹،۰۷۸	۴۵	۲۱۰،۲۴۴	۱،۱۲۶	۲،۷۱۵،۷۶۵
Password	۶۹	۹۴،۶۴۷	۹	۱۹	۱۴۷	۲۰۸،۳۴۳
Phone Number	۶۸	۱۶۷،۲۳۴	۵	۳۴	۵۱	۶۵۴،۶۳۴
Nominative	۲۳۶	۲،۲۷۴،۳۲۷	۲۰	۱۳،۱۶۸	۲۷۰	۱،۲۴۸،۵۹۲
Facebook	۲۶۶	۲۲۱،۵۷۶	۴۰	۶۲۳	۸۴	۳۴،۷۷۲
Linkedin	۱۳۳	۲۶،۶۹۵	۶	۴۹۷	۳۹	۱۵۰،۸۹

۷.۵ افشای وب سرویس‌ها

همانطور که می‌دانید دیتابیس‌ها داده‌های مربوط به سایت‌ها و اپلیکیشن‌های مختلفی را نگهداری می‌کنند. در صورتی که یک هکر بتواند وارد این دیتابیس‌ها شود با دسترسی‌های خواندن/نوشتن می‌تواند داده‌های دیتابیس این اپلیکیشن‌ها را تخریب کند یا در بدترین حالت می‌تواند موجب تغییر چهره^۶ آن وب سایت یا اپلیکیشن شود که در نهایت موجب انتشار داده‌های نامناسب و غیر معتبر خواهد شد. محققان توانستند با بررسی آدرس‌های IP و تبدیل آنها به آدرس‌های معنادار DNS به وب‌سایت‌هایی برسند که ممکن است آن وب‌سایت‌ها از این دیتابیس‌ها خوراک^۷ دریافت کنند. نتیجه دسترسی به این وب‌سایت‌ها در جدول ۶ آمده است. (نتیجه تنها روی وب‌سایت‌هایی بوده است که در حین تحقیق قابلیت دسترسی داشتند و بدون timeout request می‌توانستند به آنها دسترسی داشته باشند). ۲۹۱ وب‌سایت به وسیله دیتابیس MongoDB،

^۶Deface
^۷Feed

۳۳۴ وبسایت به وسیله دیتابیس، Elasticsearch ۲۷ مورد برای دیتابیس Redis و ۱۱ مورد برای دیتابیس Cassandra که به ترتیب ۱۶/۸٪، ۱۹/۳٪، ۱/۵٪ و ۰/۶٪ وبسایت‌هایی به صورت عمومی در دسترس بودند. به طور کلی، با دسترسی هکر به یک دیتابیس دو نوع از حمله می‌تواند رخ دهد:

۱. حمله به روش تخریب چهره^۸

۲. حمله به روش تزریق کد^۹

جدول ۶: افشای وب سرویس‌ها: لیست تعداد سایت‌هایی که محققان توانستند از آن پاسخ بگیرند

Http code	No. Websites	Summary
۲۰۰	۱۰۷۲۳	Request Succeeded
۳۰۲	۴۷۳	Found. URI changed temporary
۴۰۴	۲۴۴	Not Found
۴۰۳	۱۳۶	Forbidden
۳۰۱	۱۰۵	Moved Permanently
۴۰۱	۸۴	Unauthorized
۵۰۲	۵۳	Bad Gateway
۳۰۷	۴۹	Temporary Redirect
۵۰۳	۲۹	Service Unavailable
۵۰۰	۲۴	Internal Server Error
۳۰۸	۲۱	Permanent Redirect
۴۰۰	۱۹	Bad Request
۲۰۳	۱۶	Request succeeded but not from the origin server
۴۰۵	۹	Method Not Allowed
۴۰۶	۷	Not Acceptable
۲۰۴	۴	No content
۵۰۴	۳	Gateway Timeout
۵۲۲	۳	Server Connection Timeout
۳۰۴	۱	No content
۴۰۲	۱	Payment required
۴۲۴	۱	Failed dependency
۵۲۱	۱	Web Server down
۵۲۶	۱	Invalid SSL certificate

۱.۷.۵ حمله به روش تخریب چهره

هکرها از طریق تغییر داده‌های مربوط به دیتابیس سعی می‌کنند فضای تخریب و دزدی اطلاعات را فراهم کنند. برای مثال تصویر یا متنی را در دیتابیس تغییر می‌دهند. کاربران هم این داده‌ها را در صفحه وبسایت‌ها می‌بینند و از آنجایی که به این صفحات اعتماد دارند فکر می‌کنند که این داده‌ها از منابع شناخته شده‌ای منتشر می‌شود و در دام هکرها می‌افتند. محققان در این تحقیق ۴۹۴ وبسایت که پتانسیل آسیب پذیری و تغییر چهره برای اهداف فیشینگ داشتند را پیدا کردند چرا که اطلاعات مهمی را داشتند و حساسیت این اطلاعات برای هکرها بسیار حائز اهمیت است.

^۸Defacement
^۹Code injection

۲.۷.۵ حمله به روش تزریق کد

این حمله از سناریو قبلی حتی خطرناک‌تر است. تزریق کد همانطور که از نامش پیداست روشی است که به هکرها این امکان را می‌دهند تا براساس آسیب پذیری‌هایی که در یک سایت وجود دارد، کد جاوا اسکریپت مخصوصی را اجرا کنند و بدون داشتن مجوز اجرا در وبسایت می‌توانند به صورت کامل اجرا شود و از آنجایی که دسترسی نوشتن را روی دیتابیس دارند می‌تواند برای محتوای داخل دیتابیس خطرناک باشد. طبق این تحقیق، ۲۹۹ وبسایت را یافتند که پتانسیل این آسیب پذیری را دارند تا موجب تخریب اطلاعات کاربران شود.

۸.۵ آنالیز در معرض خطر بودن نمونه‌ها

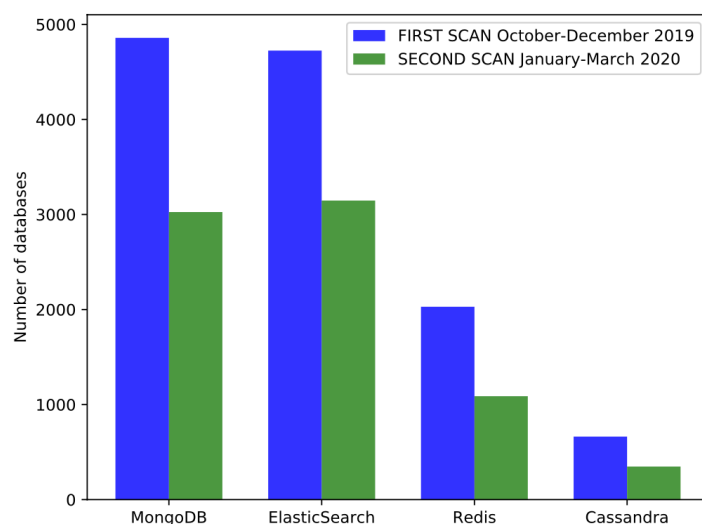
در بین تمام دیتابیس‌ها، نمونه‌های دیتابیس Redis بیشترین تعداد حمله را داشته است. معیار محققان برای یافتن نمونه‌های در معرض خطر، در حقیقت جست و جوی کلمات مشکوک در نام جداول بوده است. در دیتابیس MongoDB کلماتی مانند "hacked by unistellar" و "how to restore" جست و جو شده است به گونه‌ای که ۳۸۹ دیتابیس در این جست و جو پیدا کردند. در دیتابیس Elasticsearch محققان کلمه "readme" را جست و جو کردند که به طور مستقیم به هکرها رمزنگاری و دزد اطلاعات مربوط می‌شود که در نهایت ۶۲۲ مورد از این کلمه یافت شد. در دیتابیس Redis محققان کلمات "Back"، "1"، و "trojan" را جست و جو کردند که مربوط به حمله سال ۲۰۱۸ به نمونه‌های دیتابیس Redis در حمله سایبری بوده است [۲]. ۶۲۵ مورد از این کلمات در دیتابیس Redis یافت شد.

۹.۵ تجزیه و تحلیل تاثیر تحقیقات امنیتی

در انتهای این مقاله، تاثیر این تحقیق و پروژه را مورد بررسی قرار می‌دهیم. از اکتبر ۲۰۱۹ تا دسامبر ۲۰۱۹ محققان این پروژه را در مرحله اجرا قرار دادند. بعد از اولین اجرای خود دوباره در ژانویه سال ۲۰۲۰ تا مارس ۲۰۲۰ این برنامه را اجرا کردند تا بررسی کنند که آیا برنامه‌نویسان و مدیران سیستمی پیام آنها را دریافت کردند و پیکربندی خود را به حالت امن تبدیل کردند؟ نتیجه در نمودار شکل ۷ آمده است.

در این نمودار می‌توان دریافت که از ۶۶۳ نمونه دیتابیس Cassandra ۳۱۵ نمونه یعنی حدوداً ۴۷/۵٪ دیتابیس‌ها بعد از دریافت پیام محققان پیکربندی خود را متناسب با شرایط امنیتی تنظیم کردند و دیگر به صورت عمومی قابل دسترسی نبودند. از ۲،۰۲۹ نمونه دیتابیس Redis، ۹۱۴ نمونه دیگر قابل ردیابی و دسترسی عمومی نبودند به طوری که ۲۷ نمونه از آنها برای ورود به سیستمشان نیازمند احراز هویت بودند. (تقریباً ۴۶/۳٪ آنها دیگر آسیب پذیری اولیه مانند احراز هویت را نداشتند!)

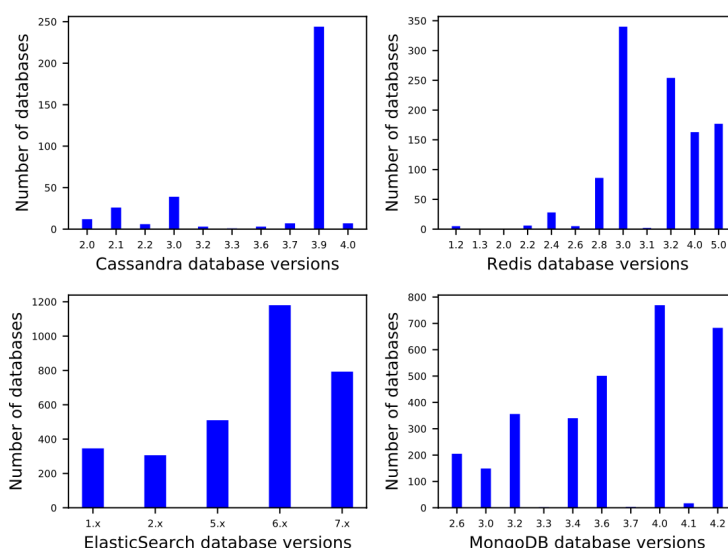
از ۴،۷۲۵ نمونه دیتابیس Elasticsearch، ۱،۵۳۹ نمونه دیگر در دسترس عموم نبودند و ۴۰ نمونه از آنها درخواست احراز هویت برای ورود به سیستم دیتابیس داشتند. (مقدار ۳۳/۴٪ از دیتابیس‌های Elasticsearch جست و جو شده). از ۴،۸۵۹ نمونه دیتابیس MongoDB، ۹۱ نمونه دیگر در دسترس عموم نبودند و ۱،۷۴۳ نمونه از آنها درخواست احراز هویت برای ورود به آنها را ارسال می‌کردند (۳۷/۷٪ از کل دیتابیس‌های MongoDB جست و جو شده).



شکل ۷: تاثیر آنالیز امنیتی: تعداد دیتابیس‌هایی که قبل و بعد از این آزمایش در خصوص پیکربندی نامناسب بررسی شدند

۱۰.۵ تجزیه و تحلیل نسخه سرویس‌های NoSQL

خالی از لطف نیست که اشاره کنیم، محققان نسخه‌های استفاده شده از دیتابیس‌ها را نیز مورد بررسی قرار دادند و متوجه شدند که تعداد زیادی از آنها از نسخه‌های قدیمی از DBMS استفاده می‌کردند که دیگر توسط توسعه دهندگان ارائه دهندگان آنها دیگر پشتیبانی نمی‌شد. به یاد داشته باشید که اگر از نسخه‌های قدیمی استفاده کنید ممکن است آسیب پذیری را برای هکرها آزاد کرده باشید که بتوانند از طریق آن به دیتابیس‌ها و حتی اپلیکیشن‌های شما حمله کنند. مهم‌ترین هدف به روز رسانی‌ها دریافت تمام پیچ‌های امنیتی در سیستم‌های نرم‌افزار مخصوصاً سیستم‌های خاص دیتابیس است. همچنین آگاه باشید که از چه نسخه‌ای از نرم‌افزار (در اینجا دیتابیس) استفاده می‌کنید. نسخه مورد استفاده اگر در محیط Dev باشد آسیب پذیری بیشتری نسبت به نرم‌افزارهایی دارند که در محیط Stable هستند. پس همیشه سعی کنید که از نرم‌افزارهای به روز Stable استفاده کنید.



شکل ۸: بررسی نسخه سرویس‌های NoSQL

۶ جمع‌بندی

در این گزارش در مورد محبوب‌ترین سرویس‌های دیتابیس NoSQL تحقیق کردیم. از طریق ابزاری که محققان توسعه دادند پی‌بردیم که راه‌اندازی ساده و آسان درست است که کم‌هزینه‌ست اما بعد از گذشت مدتی خیلی کم‌متوجه می‌شویم که هر چقدر برای راه‌اندازی سیستم‌های دیتابیس هزینه زمانی و مالی بگذاریم باز هم کم است. فریم‌ورک محققان براساس روندی امن این تحقیق را تسریع بخشید. فرایندهای مختلفی در این بین بررسی شدند. مانند حضور یک سیستم احراز هویت مناسب، عدم استقرار و راه‌اندازی سیستم‌های داده‌ای در شماره پورت‌های پیش فرض و غیره. این بررسی بر روی ۶۷ میلیون آدرس IP بین اکتبر سال ۲۰۱۹ و مارس سال ۲۰۲۰ انجام شد. آسیب پذیری‌هایی که در حین خواندن این گزارش شاهد بودید تنها ساده‌ترین آسیب پذیری‌هایی بودند که مورد بررسی قرار گرفتند. همچنین انواع اهداف نفوذگران را بررسی کردیم که به چه شکلی می‌توانند تخریب اطلاعات را انجام دهند. قلب تپنده یک سیستم اپلیکیشنی، کاربران و داده‌های آنها هستند. در صورتی که داده‌ها فاش شود اعتماد به آن شبکه کم و به تدریج از بین می‌رود. این تحقیق تنها روی ۴ دیتابیس مطرح شده صورت گرفت. به عنوان نمونه هنوز فرایندی که شاهد آن بودید بر روی دیتابیس‌های InfluxDB و دیگر دیتابیس‌های محبوب مانند Prometheus انجام نشده است. همچنین محققان این مقاله به دنبال انجام این روند در بستر شبکه TCP با استفاده از شناسایی آدرس‌های نسخه ۶ IP هستند.

مراجع

- [1] Atlas MongoDB. Map Reduce. <https://www.mongodb.com/docs/manual/core/map-reduce/>, 2023.
- [2] Catalin Cimpau.
- [3] DB-Engines. DB-Engines Ranking. <https://db-engines.com/en/ranking>, 2023.
- [4] Ferrari, Dario and Carminati, Michele and Polino, Mario and Zanero, Stefano. Nosql breakdown: A large-scale analysis of misconfigured nosql services. In *Annual computer security applications conference*, pages 567–581, 2020.
- [5] Liam Tung. MongoDB ransacked: Now 27,000 databasae hit in mass ransom attacks. <https://www.zdnet.com/article/mongodb-ransacked-now-27000-databases-hit-in-mass-ransom-attacks/>, 2017.
- [6] Nmap Security Scanner. What is Nmap method? <https://nmap.org/>, 2023.