

بسمه تعالی



سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

جرم‌شناسی در پایگاه داده MongoDB

پیشگفتار

رشد روز افزون حجم اطلاعات از یک سو و پیشرفت فن‌آوری‌های نوین از سوی دیگر سبب شده تا طیف وسیعی از تهدیدها و جرم‌ها در پایگاه‌داد مطرح گردد. وجود این تهدیدها و جرم‌ها سبب شده تا تمهیدات امنیتی چون حفظ محرمانگی، صحت و دسترس‌پذیری در این سامانه‌ها از جایگاه ویژه‌ای برخوردار باشد. مکانیزم‌های امنیتی موجود به سبب حفظ کارایی سامانه نمی‌توانند جلوی تمامی تهدیدها و جرایم اینترنتی در پایگاه‌های داده را بگیرند و لذا امکان وقوع جرم در سامانه امکان‌پذیر است. از این رو، به منظور شناسایی شواهد جرم، جمع‌آوری اطلاعات مربوطه، تجزیه و تحلیل آن‌ها و در نهایت تهیه مستندات لازم جهت اثبات وجود جرم، شاخه‌ی جرم‌شناسی پایگاه‌داده مطرح می‌گردد. وجود زیرساخت‌های مختلف برای سامانه‌های پایگاه‌داده، طبیعت چند بعدی سامانه‌ها، ابزارهای مختلف جرم‌شناسی و فقدان مدیریت دانش مرتبط با جرم‌شناسی را می‌توان از جمله چالش‌های اصلی در این حوزه دانست. وجود این چالش‌ها سبب شده است تا جرم‌شناسی پایگاه‌داده به عنوان یک موضوع پیچیده و بغرنج معرفی گردد.

در این راستا بر آن شدیم تا جرم‌شناسی در سامانه MongoDB را برای برخی از عملکردهای پر کاربرد، مورد بحث و بررسی قرار دهیم. بدین منظور، با استفاده از فرآیند استاندارد جرم‌شناسی، مراحل شناسایی، گردآوری اطلاعات، تحلیل، ترمیم و ارائه مستندات کافی برای اثبات جرم را تشریح می‌کنیم. لازم به ذکر است که اگر چه جرم‌شناسی سامانه پایگاه‌داده از حدود سال ۲۰۰۴ مطرح است، اما تا کنون پیشرفت خاصی در زمینه‌ی ابزارهای تجاری و رایگان برای رسیدگی به رویدادهای غیرمجاز ارائه نشده است.

در ادامه و در فصل اول، مفاهیم و تعاریف اولیه‌ی جرم‌شناسی پایگاه‌داده، چالش‌ها، اهداف و گام‌های اجرایی جرم‌شناسی تشریح می‌شود. در فصل دوم، فرآیند جرم‌شناسی و مدیریت جرم در هر سامانه پایگاه‌داده (سمپاد) مورد بحث و بررسی قرار می‌گیرد. در فصل‌های سوم تا پنجم، گام‌های شناسایی، جمع‌آوری شواهد، استخراج و تحلیل اطلاعات، ترمیم و ارائه مستندات مربوط به جرم، به ترتیب برای هر یک از رویدادهای درج، حذف، روزرسانی و مشاهده غیرمجاز محتوای جداول، تغییر غیرمجاز شمای پایگاه‌داده و تلاش برای ورود غیرمجاز به سامانه پایگاه‌داده مطالعه می‌شود. لازم به ذکر است که کلیه پیکربندی‌های ارائه شده در مستند حاضر بر اساس MongoDB 3.4.10 نسخه‌ی Enterprise می‌باشد.

فهرست مطالب

۴.....	۱	جرم‌شناسی پایگاه‌داده.....
۴.....	۱-۱	تعاریف و مفاهیم.....
۵.....	۱-۲	چالش‌ها.....
۶.....	۱-۳	اهداف.....
۷.....	۱-۴	گام‌های اجرایی.....
۱۰.....	۱-۵	جمع‌بندی.....
۱۰.....	۲	شناسایی و مدیریت جرم.....
۱۱.....	۲-۱	تمهیدات جرم‌شناسی.....
۱۴.....	۲-۲	فرآیند جرم‌شناسی.....
۱۸.....	۲-۳	رهنمون‌های فرآیند جرم‌شناسی.....
۲۳.....	۲-۴	جمع‌بندی.....
۲۴.....	۳	درج، حذف، بروزرسانی و مشاهده‌ی غیرمجاز محتوای جداول.....
۲۴.....	۳-۱	شناسایی جرم.....
۲۴.....	۳-۲	جمع‌آوری اطلاعات و شواهد.....
۳۳.....	۳-۳	استخراج و تجزیه و تحلیل اطلاعات.....
۳۶.....	۳-۴	ترمیم.....
۳۸.....	۳-۵	ارائه‌ی مستندات.....
۴۰.....	۳-۶	جمع‌بندی.....
۴۰.....	۴	تغییر غیرمجاز شمای پایگاه‌داده.....
۴۰.....	۴-۱	شناسایی جرم.....
۴۱.....	۴-۲	جمع‌آوری اطلاعات و شواهد.....
۴۴.....	۴-۳	استخراج و تجزیه و تحلیل اطلاعات.....
۴۷.....	۴-۴	ترمیم.....
۴۷.....	۴-۵	ارائه‌ی مستندات.....
۴۸.....	۴-۶	جمع‌بندی.....
۴۸.....	۵	تلاش برای ورود غیرمجاز به پایگاه‌داده.....
۴۸.....	۵-۱	شناسایی جرم.....
۴۸.....	۵-۲	جمع‌آوری اطلاعات و شواهد.....
۵۰.....	۵-۳	استخراج و تجزیه و تحلیل اطلاعات.....
۵۲.....	۵-۴	ترمیم.....
۵۲.....	۵-۵	ارائه‌ی مستندات.....
۵۲.....	۵-۶	جمع‌بندی.....
۵۳.....	۶	خلاصه مطالب.....
۵۵.....	۷	منابع.....

۱ جرم‌شناسی پایگاه داده

بسیاری از سازمان‌ها و آژانس‌های دولتی از پایگاه داده برای ذخیره و بازیابی اطلاعات استفاده می‌کنند. از آنجاییکه داده‌های سازمان‌ها می‌توانند حاوی اطلاعات حساس و حیاتی باشند، لذا حفاظت از پایگاه داده به عنوان یک سامانه ذخیره‌سازی اطلاعات، یک امر حیاتی و مهم به حساب می‌آید. داده‌های ذخیره شده در پایگاه داده ممکن است توسط کاربران غیرمجاز (از درون سازمان یا بیرون سازمان) تغییر داده شوند. لازم به ذکر است که بسیاری از مجرمان به دلیل فقدان دلایل کافی برای اثبات جرم، محکوم نمی‌شوند. در این شرایط، جرم‌شناسی نقش مهمی در ارایه روش‌های اثبات شده علمی برای جمع‌آوری اطلاعات، تحلیل و بررسی و نهایتاً ارائه‌ی شرح مفصلی از فعالیت‌های مجرمانه‌ی سایبری ایفا می‌کند. جرم‌شناسی پایگاه داده، شاخه‌ای از جرم‌شناسی دیجیتال است که در آن پایگاه داده و فراداده‌های مرتبط با آن به صورت قانونی مورد مطالعه قرار می‌گیرند. یکی از مهم‌ترین اهدافی که در جرم‌شناسی پایگاه داده دنبال می‌شود آن است که تشخیص دهیم چه کسی، چه زمانی، چه داده‌ای را تغییر داده است. لازم به ذکر است که پایگاه داده قربانی معمولاً شامل اطلاعاتی است که در طول تحقیقات قانونی به کار می‌آید. در ادامه برخی از تعاریف و مفاهیم کلیدی، چالش‌ها، اهداف و گام‌های اجرایی فرآیند جرم‌شناسی در سامانه پایگاه داده، مورد بحث و بررسی قرار می‌گیرد.

۱-۱ تعاریف و مفاهیم

در این بخش، برخی از مهم‌ترین تعاریف و مفاهیم جرم‌شناسی پایگاه داده که در سرتاسر مستند مورد استفاده قرار گرفته است، مورد بحث و بررسی قرار می‌گیرد.

سامانه مدیریت پایگاه داده: به نرم افزاری اطلاق می‌شود که برای نگهداری و مدیریت حجم وسیعی از اطلاعات طراحی شده و مورد استفاده قرار می‌گیرد [۴].

جرم‌شناسی: مجموعه‌ای از آزمایش‌ها یا روش‌های علمی است که در تحقیقات جنایی مورد استفاده قرار می‌گیرد. امروزه جرم‌شناسی به روشی برای به دست آوردن شواهد جنایی به منظور ارائه در دادگاه اشاره دارد.

تجزیه و تحلیل جرم‌شناسی: به فرآیند بررسی رویدادهای غیرمجاز با در نظر گرفتن خط زمانی موجود از شواهد فیزیکی مربوط به آن، تجزیه و تحلیل جرم‌شناسی اطلاق می‌شود. نتایج حاصل می‌تواند در شناسایی مجرم کمک کند و همچنین به منظور ارایه شواهد برای اثبات جرم، مورد استفاده قرار گیرد [۱].

¹ Timeline

جرم‌شناسی دیجیتال: به فرآیندی که در آن از روش‌های علمی مرسوم و اثبات شده برای: (۱): حفاظت، (۲): جمع‌آوری، (۳): اعتبارسنجی، (۴): شناسایی، (۵): تجزیه و تحلیل، (۶): تفسیر، (۷): مستندسازی و ارائه‌ی شواهد دیجیتال به منظور تسهیل در بازسازی رویدادهای شناخته شده جنایی یا پیش‌بینی اقدامات غیرمجاز استفاده می‌شود، جرم‌شناسی دیجیتال اطلاق می‌گردد [۵].

جرم‌شناسی پایگاه داده: جرم‌شناسی پایگاه داده فرآیندی است که تلاش می‌کند تا زمان/چگونگی/چرایی و عامل (های) رویداد غیرمجاز در سامانه را مشخص نماید. لازم به ذکر است که محتوای پایگاه داده، فراداده‌ها^۲ (به ویژه فایل‌های رویدادنگاری) و داده‌های موجود در حافظه از جمله مهمترین مولفه‌های تاثیرگذار در این فرآیند به حساب می‌آیند.

ممیزی:^۳ به نظارت و ثبت فعالیت‌های کاربران در پایگاه داده، ممیزی اطلاق می‌شود [۶].

رویدادنگاری:^۴ به تاریخچه‌ای از فعالیت‌های اجرا شده توسط سامانه مدیریت پایگاه داده اطلاق می‌شود که برای تضمین ویژگی‌های جامعیت (ACID) به هنگام خرابی سخت افزاری یا از کار افتادگی ناگهانی^۷ مورد استفاده قرار می‌گیرد [۷].

مصنوعات پایگاه داده: رکوردها یا اطلاعاتی هستند که از پایگاه داده قابل استخراج بوده و در تجزیه و تحلیل جرم‌شناسی مفید هستند [۲].

رویداد غیرمجاز: به رویدادی اطلاق می‌شود که به صورت خصمانه یا ناخواسته در روال عادی سیستم تغییر نامطلوبی را ایجاد می‌کند.

۱-۲ چالش‌ها

جرم‌شناسی پایگاه داده، چالش‌های زیادی به همراه دارد که آن را تبدیل به یک موضوع پیچیده می‌کند. سامانه‌های پایگاه داده، سرویس‌ها و زیرساخت‌های مختلفی دارند که از یک پایگاه داده به پایگاه داده دیگر، متفاوت هستند. علاوه بر این، پایگاه‌های داده مختلف دارای مصنوعات جرم‌شناسی متفاوت همچون روش‌ها، مدل‌ها، چارچوب‌ها، ابزارها، فعالیت‌ها و خط‌مشی‌های مختلف هستند از سوی دیگر، سامانه‌های پایگاه داده دارای

² Metadata

³ Auditing

⁴ Logging

⁵ Atomicity-Consistency-Isolation-Durability

⁶ Hardware failure

⁷ Crash

⁸ Artifact

طبیعت چند بعدی شامل سطح داخلی، سطح مفهومی و سطح خارجی هستند. سطح داخلی شامل فایل فیزیکی است و سطح مفهومی، سطح منطقی است که زیرساخت منطقی شمای پایگاه‌داده از جمله کاربران، جداول، شاخص‌ها و رویه‌ها را نمایش می‌دهد. سطح خارجی با کاربران واقعی سروکار دارد تا بتوانند داده‌ها را تغییر دهند. بنابراین، ابعاد مختلف پایگاه‌داده در جرم‌شناسی پایگاه‌داده ایفای نقش می‌کنند [۸].

یک چالش مهم دیگر در حوزه جرم‌شناسی پایگاه‌داده، تشخیص وجود نقض امنیتی در سامانه است. در واقع، فرآیند شناسایی و ترمیم به هنگام وقوع جرم، تا زمانی که شخصی فکر کند که نقض امنیتی رخ داده است، به تاخیر می‌افتد. در حالت کلی، شواهد برای وجود نقض امنیتی را می‌توان در سه دسته‌ی زیر خلاصه کرد:

- داده‌های سازمان در خارج از سازمان پیدا می‌شوند که مسلماً عادی و مجاز نیست.
- رویدادی مشاهده می‌شود که غیرمنتظره است؛ مثلاً فرآیندی در زمان اشتباه اجرا شده است یا دسترسی به سامانه، خارج از ساعت‌های اداری و مجاز اتفاق افتاده است.
- نشانه‌هایی از تغییر غیرمجاز داده‌ها وجود داشته باشد.

روش‌هایی برای جمع‌آوری و تجمیع شواهد مجرمانه در پایگاه‌داده وجود دارد. جرم‌شناسی پایگاه‌داده زمانی رخ می‌دهد که از مأمور ممیزی، نحوه وقوع نقض امنیتی و شخص مجرم، درخواست شود. روش‌هایی که برای جرم‌شناسی پایگاه‌داده وجود دارند، اصولاً دارای دو محدودیت زیر هستند:

- کاربر پس از هفته‌ها یا ماه‌ها متوجه نقض امنیتی در پایگاه‌داده می‌شود. در این صورت داده‌های ناپایدار در پایگاه‌داده به عنوان شواهد وجود ندارد.
- ممکن است هیچ ممیزی برای پایگاه‌داده فعال نباشد.

دو عامل فوق‌الذکر سبب می‌شوند که بررسی رویداد مورد تقاضا، زمان آن و نتیجه‌ی حاصل از بررسی در پیچیده‌ترین حالت ممکن قرار گیرد. آنچه جرم‌شناسی پایگاه‌داده را از جرم‌شناسی شواهد فیزیکی متمایز می‌کند، حجم پایگاه‌داده و نیاز به در حال اجرا ماندن آن در محیط عملیاتی است.

۱-۳ اهداف

برخی از مهم‌ترین اهدافی که در جرم‌شناسی پایگاه‌داده به دنبال آن هستیم به قرار زیر است [۸-۹]:

ردگیری اعمال DDL و DML: در چرخه حیات پایگاه‌های داده بارها نیاز می‌شود که تغییرات در داده‌ها همچون درج، بروزرسانی و حذف داده‌ها که از اعمال دستکاری داده (DML) به حساب می‌آیند و تغییرات در

⁹ Breach

¹ Trace

اشیای پایگاه داده همچون ایجاد، تغییر و حذف یک جدول که از اعمال تعریف داده (DDL) به حساب می‌آیند، ردگیری و بررسی شوند. با این کار، رویدادهای غیرمجاز تشخیص داده شده و مجرمان شناسایی می‌شوند.

شناسایی داده‌ها پیش و پس از تراکنش: در طول یک تراکنش، ممکن است داده‌ها دستخوش تغییرات زیادی شوند. گاهی داده‌های جدیدی ایجاد، داده‌های موجود حذف یا تغییر داده می‌شوند. شناسایی تغییرات اعمال شده بر روی داده‌ها، ما را در تشخیص رویدادهای غیرمجاز کمک خواهد کرد.

بازگشت به عقب اعمال غیرمجاز تغییر داده: در صورتی که داده‌ها طی رویدادهای غیرمجاز تغییر داده شوند، باید بتوان آن‌ها را به وضعیت پیش از رویداد غیرمجاز بازگرداند. همچنین در صورت حذف داده‌ها، نیاز به بازبایی داده‌های حذف شده خواهد بود.

اثبات یا رد وقوع نقض امنیتی: یک چالش مهم در حوزه جرم‌شناسی پایگاه داده، تشخیص وجود نقض امنیتی در سامانه است. در واقع فرآیند شناسایی و ترمیم به هنگام وقوع جرم، تا زمان تشخیص نقض امنیتی در سامانه به تاخیر می‌افتد. پس از آن با طی کردن گام‌های مطرح در فرآیند جرم‌شناسی پایگاه داده می‌توان ثابت کرد که نقض امنیتی رخ داده است یا خیر.

تعیین محدوده‌ی نفوذ به پایگاه داده: هنگامی که به پایگاه داده حمله می‌شود، تشخیص حمله‌ی انجام گرفته و محدوده‌ی نفوذ به منظور شناسایی خسارات وارد شده و محدوده‌ی تغییرات غیرمجاز، از اهمیت بالایی برخوردار است.

کشف اینکه چه اتفاقی در چه زمانی رخ داده است: با بررسی و تحلیل رویدادهای ثبت شده، اطلاعاتی همچون کاربر اجراکننده رویداد، زمان رخداد، داده متاثر از رویداد، چگونگی تغییر و علت آن مشخص می‌شود. با دانستن این اطلاعات، نه تنها جزییات رویدادهای رخ داده مشخص می‌شوند بلکه می‌توان با توجه به توالی زمانی رویدادها و تجمیع آن‌ها، اطلاعات جدیدی نیز کسب کرد. برخی از داده‌ها به تنهایی دارای ارزش کمی هستند ولی هنگامی که با سایر اطلاعات ترکیب می‌شوند، می‌توانند نقض امنیتی را آشکارتر سازند.

۴-۱ گام‌های اجرایی

گام‌هایی که برای جرم‌شناسی پایگاه داده باید برداشته شوند به موقعیت و نوع سامانه مدیریت پایگاه داده‌ی وابسته است. در ادامه، برخی از مهم‌ترین گام‌هایی که در این راستا می‌بایست لحاظ شود، آورده شده است.

۱. **مرحله‌ی شناسایی:** در مرحله‌ی شناسایی، رویداد و نوع آن با توجه به نشانه‌های موجود در سامانه شناسایی می‌شود. این مرحله از آن جهت مهم است که سایر مراحل را تحت تأثیر قرار می‌دهد. برخی از مهم‌ترین اهداف این مرحله، شناسایی مفاهیم مرتبط با بررسی جرم‌شناسی همچون منابع پایگاه داده، منابع سیستم‌عامل، منابع شبکه، تیم‌های بررسی، روش‌های بررسی، محیط بررسی، خط‌مشی‌ها، قوانین و مجوزها هستند.

۲. تعیین روش جمع‌آوری داده‌های جرم‌شناسی: به منظور بررسی پایگاه داده آسیب‌دیده یا مورد سوءاستفاده، سه روش جمع‌آوری داده به شرح زیر وجود دارد:

- جمع‌آوری داده‌ها به صورت زنده^۱: این روش جمع‌آوری داده زمانی اتفاق می‌افتد که سامانه مورد تجزیه و تحلیل به طور همزمان در حال سرویس‌دهی نیز می‌باشد.
- جمع‌آوری داده‌ها به صورت غیرزنده^۲: روش جمع‌آوری داده به صورت غیرزنده، شامل نسخه‌برداری داده‌ها از سیستم مورد بررسی است.
- جمع‌آوری داده‌ها به صورت ترکیبی^۳: روش جمع‌آوری داده‌ها به صورت ترکیبی با بهره‌گیری از ویژگی‌های کلیدی هر دو روش قبل، ترکیبی از این دو روش را برای جمع‌آوری داده در پیش می‌گیرد.

لازم به ذکر است که صرف‌نظر از روش مورد استفاده می‌بایست این اطمینان حاصل شود که شواهد دیجیتال حفظ و نگهداری می‌شوند و داده‌ها به صورت ناخواسته تغییر نمی‌کنند یا از بین نمی‌روند.

۳. جمع‌آوری مصنوعات ناپایدار^۴ و پایدار: مصنوعات و اطلاعات مختلف را می‌توان از پایگاه داده، سیستم‌عامل، سرورهای وب یا فایل‌های رویدادنگاری استخراج کرد. لازم به ذکر است که جمع‌آوری مصنوعات و شواهد در سامانه می‌تواند سبب تغییر در پایگاه داده شود. از این‌رو، پیش از استخراج اطلاعات از پایگاه داده باید نسبت به این موضوع و پایدار یا ناپایدار بودن اطلاعات، آگاهی پیدا کرد. هر پایگاه داده شواهد مربوط به اعمال مختلف را در فایل‌های رویدادنگاری مختلفی پایگاه داده ذخیره می‌کند. این بدین معناست که برای تجزیه و تحلیل جرم‌شناسی می‌بایست نسبت به چگونگی عملکرد پایگاه داده، محل فایل‌ها و مصنوعات مختلف اطلاع داشت. مصنوعات در سطح پایگاه داده به دو دسته تقسیم می‌شوند: (۱): داده‌های فرار و (۲): داده‌های غیر فرار که در ادامه به تشریح هر یک از آنها می‌پردازیم.

- داده‌های فرار: به برخی از داده‌ها که به منظور افزایش سرعت، قابلیت اطمینان و بهره‌وری پایگاه داده در حافظه‌های ناپایدار ذخیره می‌شوند، داده‌های فرار اطلاق می‌شود. به عنوان نمونه،

1	Live acquisition	1
1	Dead acquisition	2
1	Hybrid acquisition	3
1	Volatile artifacts	4

پایگاه‌داده اوراکل اطلاعات زیادی را در SGX به منظور افزایش کارایی ذخیره می‌کند. لازم به ذکر است که به طور معمول این دسته از داده‌ها دائماً و با سرعت بالایی در حال تغییر هستند.

- **داده‌های غیرفرار:** به رکوردهای ذخیره شده در پایگاه‌داده، داده‌های غیرفرار یا پایدار اطلاق می‌شود.

ذکر این نکته ضروری است که فرآیند جرم‌شناسی در سامانه پایگاه‌داده نباید تنها بر روی پایگاه‌داده متمرکز باشد. پایگاه‌داده در یک محیط مجزا در حال اجرا و سرویس‌دهی نیست و متکی به زیرساخت مهمی چون سیستم عامل است. بنابراین باید سایر مصنوعات و اطلاعات جانبی از سیستم عامل‌ها و رویدادهای ثبت شده در سرور را نیز جمع‌آوری و آنها را بررسی نمود.

۴. **حفاظت و احراز اصالت داده‌های جمع‌آوری شده:** هدف از این مرحله آن است که مقدار شواهدی که مجدداً بر روی آن‌ها اطلاعاتی نوشته می‌شود، کاهش پیدا کند. مراقبت‌های شدیدی برای تضمین عدم تغییر غیرمنتظره داده‌ها باید انجام شود. اگر چه داده‌ها را می‌توان با ارسال پرسمان به پایگاه‌داده‌ی تغییر یافته به دست آورد، باید از اجرای هر نوع پرسمانی که باعث حذف اطلاعات از پایگاه‌داده شود، اجتناب نمود. علاوه بر این، در صورت وجود پایگاه‌داده آسیب‌دیده یا مورد سوء استفاده قرار گرفته، صرف‌نظر از روش بدست آوردن داده، هیچ عبارت SQL ای نباید اجرا شود؛ زیرا باعث تغییر داده‌های ذخیره‌شده در حافظه و صفحات داده مربوط به پایگاه‌داده می‌شود. این کار همچنین باعث تقسیم شدن صفحه داده داخلی و محل ذخیره‌سازی داده‌های جدید در حافظه پنهان^۱ می‌شود و فرآیند بررسی را پیچیده‌تر می‌کند. بنابراین باید پیش از فرآیند جمع‌آوری، نسخه پشتیبان از داده‌ها و فایل‌های مهم تهیه گردد. لازم به ذکر است که روش‌ها و ابزارهای مورد استفاده برای جمع‌آوری اطلاعات و شواهد می‌بایست تا حد امکان قابل اطمینان باشند.

۵. **تجزیه و تحلیل شواهد و تعیین فعالیت‌های مهاجم:** تجزیه و تحلیل داده‌های جمع‌آوری شده به نوع داده‌ها، سامانه پایگاه‌داده و رویداد خاصی که قرار است مورد بررسی قرار گیرد، بستگی دارد. مرحله‌ی تجزیه و تحلیل می‌بایست ابعاد مربوط به هر رویداد و محلی که اطلاعات مربوطه یافت می‌شود را در نظر بگیرد. بنابراین در مرحله‌ی تجزیه و تحلیل، اطلاعاتی همچون شخص مجرم، زمان ارتکاب

^۱ قسمتی از حافظه‌ی سیستم که میان تمامی پردازنده‌های مربوط به یک نمونه‌ی واحد از پایگاه‌داده‌ی اوراکل مشترک است.

1	Preservation	6
1	Authentication	7
1	Cache	8

جرم، داده هدف، دلایل ارتکاب و نحوه اجرای جرم تعیین می‌گردد. تجمیع داده‌ها در فرآیند جرم‌شناسی پایگاه‌داده از اهمیت بسزایی برخوردار است. برخی از داده‌ها به تنهایی دارای ارزش کمی هستند اما هنگامی که با سایر اطلاعات ترکیب می‌شوند، می‌توانند نقض امنیتی را آشکارتر سازند.

۶. **بازسازی پایگاه‌داده:** در بررسی پایگاه‌داده آسیب‌دیده یا مورد سوءاستفاده، داده‌هایی که قبلاً حذف شده‌اند، باید بازیابی و اقدامات انجام شده توسط مهاجم شناسایی شوند.

۷. **ارائه مستندات:** در مرحله آخر، کلیه‌ی بررسی‌های صورت گرفته در یک قالب استاندارد مستند و به مدیر سامانه و دادگاه ارائه می‌شود. لازم به ذکر است که مستندات تهیه شده برای سایر بررسی‌کنندگان که سناریوی مشابه‌ای را تجربه می‌کنند و همچنین برای حفاظت از پیگردهای قانونی بررسی‌کنندگان در آینده مفید خواهد بود.

۱-۵ جمع‌بندی

در این فصل به طور مشروح به معرفی مفاهیم جرم‌شناسی پایگاه‌داده و همچنین بررسی چالش‌ها، اهداف و گام‌های اجرایی در فرآیند جرم‌شناسی پایگاه‌داده پرداخته شد. در این راستا و در ابتدا، وجود تنوع‌های گسترده از سامانه‌ها، سطوح مختلف داخلی، مفهومی و خارجی پایگاه‌داده و نحوه تشخیص وقوع جرم به عنوان مهم‌ترین چالش مطرح در حوزه جرم‌شناسی پایگاه‌داده بررسی گردید. در ادامه، برخی از مهم‌ترین اهداف جرم‌شناسی پایگاه‌داده، تحت عناوینی چون شناسایی و اثبات وقوع جرم، کشف رویداد غیرمجاز و زمان وقوع آن، تعیین محدوده نفوذ و بازگرداندن وضعیت سامانه به وضعیت قبل از وقوع جرم معرفی گردید. در نهایت نیز، گام‌های اجرایی فرآیند جرم‌شناسی پایگاه‌داده از مرحله شناسایی جرم تا ارائه مستندات مربوط به شواهد وقوع جرم، تشریح گردید.

۲ شناسایی و مدیریت جرم

پژوهش‌های انجام شده در زمینه‌ی جرم‌شناسی دیجیتال منجر به توسعه‌ی روش‌ها و مدل‌های فرآیندی مختلفی شده است. بسیاری از این روش‌ها به سبب ویژگی‌های خاص هر یک از سامانه‌های پایگاه‌داده، به طور کامل قابل انطباق با جرم‌شناسی پایگاه‌داده نبوده و می‌بایست در آن‌ها تغییراتی صورت پذیرد. با استفاده از مدل‌های فرآیند جرم‌شناسی پایگاه‌داده می‌توان به صورت ساختاریافته، عملیات مربوط به جرم‌شناسی پایگاه‌داده را پیش برد. در این فصل، ابتدا تمهیدات مورد نیاز برای فراهم کردن بستر مناسب برای جرم‌شناسی در پایگاه‌داده مورد مطالعه قرار می‌گیرد. در ادامه و در بخش دوم نیز به شرح و بررسی فرآیند جرم‌شناسی در سامانه پایگاه‌داده می‌پردازیم. در بخش پایانی نیز ملاحظات و رهنمون‌های مورد نیاز برای رهگیری و انجام مراحل موجود در فرآیند جرم‌شناسی تشریح می‌گردد.

۲-۱ تمهیدات جرم‌شناسی

در این بخش، تمهیدات لازم برای جرم‌شناسی پایگاه‌داده را در طیف وسیعی از نیازمندی‌ها از پیش از وقوع جرم تا نیازمندی‌های نهایی مربوط به ارائه مستندات و اثبات وقوع جرم، مورد بحث و بررسی قرار می‌دهیم. لازم به ذکر است که داشتن طرح و برنامه دقیق، مهمترین گام در فرآیند جرم‌شناسی است. در یک دسته‌بندی کلی می‌توان تمهیدات مورد نیاز برای فرآیند جرم‌شناسی پایگاه‌داده را در موارد زیر خلاصه کرد [۱۵-۱۷]:

۱. **تعیین مدیر فرآیند:** این تمهید پیش از وقوع جرم انجام می‌شود. پیش از آنکه جرمی رخ دهد، شخصی باید به عنوان هماهنگ‌کننده تعیین شود. این شخص باید فرآیند تجزیه و تحلیل جرم‌شناسی را رهبری کند و از مستند تهیه شده از فرآیند به عنوان یک چک لیست برای اطمینان از اجرای گام به گام فرآیند استفاده نماید. علاوه بر این می‌بایست تیمی تحت رهبری هماهنگ‌کننده نیز وجود داشته باشد که از مهارت‌های امنیتی و مدیریتی در حوزه پایگاه‌داده برخوردار باشند. همچنین وجود یک مجموعه از ابزارها برای استفاده در فرآیند تجزیه و تحلیل بسیار مهم است. هریک از ابزارها و هر آنچه که هر ابزار انجام می‌دهد، می‌بایست مستند شود. بزرگترین چالش در فرآیند تجزیه و تحلیل، عدم وجود ابزارهای استاندارد رایگان به منظور تجزیه و تحلیل فایل‌های ردیابی و رویدادهای تولید شده توسط پایگاه‌داده است. در صورتی که در طول فرآیند جرم‌شناسی از ابزاری استفاده شود، باید بتوان ثابت کرد که ابزارهای مورد استفاده، شواهد جمع‌آوری شده را تغییر نمی‌دهند و حذف نمی‌کنند. در صورتی که ابزار توسط خود افراد ایجاد شده باشد، اثبات عدم تغییر در شواهد، می‌تواند دشوار باشد. بنابراین بهتر است از ابزارهای تجاری از پیش تعیین شده‌ای که در دادگاه‌ها قابل قبول و استناد هستند، استفاده شود. لازم به ذکر است که ابزارهای مورد استفاده نیز می‌بایست از حیث امنیتی قابل اعتماد باشند و اجازه ندهند مهاجم از طریق آن‌ها به شواهد موجود خدشه‌ای وارد کند.
۲. **تعیین مولفه مرکزی برای اطلاع‌رسانی:** تعیین مولفه مرکزی برای گزارش جرم‌های حادث شده به عنوان یک تمهید پیش از وقوع جرم، از اهمیت بالایی برخوردار است. وجود این مولفه سبب می‌شود تا ذینفعان به سرعت از وقوع نقض امنیتی مطلع و بررسی آن را در دستور کار خود قرار دهند.
۳. **تشخیص وقوع نقض امنیتی:** به محض تشخیص نقض امنیتی در سامانه می‌بایست اطلاعات کافی برای مولفه مرکزی ارسال گردد. مولفه مرکزی نیز اطلاع‌رسانی‌های لازم را در این رابطه برای سایر ذینفعان ارسال می‌کند.

۴. **انتقال کنترل فرآیند به مدیر هماهنگ‌کننده:** به محض تشخیص وقوع نقض امنیتی در سامانه و اعلام آن توسط مولفه مرکزی، کنترل فرآیند به مدیر تجزیه و تحلیل جرم‌شناسی منتقل می‌شود تا این اطمینان حاصل شود که فرآیند به درستی و با دقت توسط تیم دنبال می‌شود.
۵. **پرهیز از قطع اتصال شبکه و خاموش کردن سامانه:** در این گام، به هیچ وجه نباید سامانه پایگاه داده خاموش یا اتصال آن به شبکه قطع گردد. توجه به این نکته حائز اهمیت است که داده‌ها و شواهد ناپایدار با خاموش شدن سیستم از بین می‌روند. اینکه مهاجم از ادامه‌ی فعالیت‌های مخرب باز بماند، ایده‌ی خوبی است ولی از دست دادن شواهد ناپایدار، بررسی‌های آتی را ممکن است با مشکل روبرو کند.
۶. **بررسی واقعی بودن حمله:** در این مرحله می‌بایست وجود نقض امنیتی و حمله به سامانه بررسی گردد. لازم به ذکر است که بررسی در دسترس بودن داده‌های حساس به طور ویژه برای مهاجم و در حالت کلی در بستر عمومی وب از اهمیت بالایی در این مرحله برخوردار است.
۷. **جمع‌آوری داده‌های فرار:** در این گام، رسیدگی به نقض امنیتی را آغاز نموده و داده‌های فرار را از روی سرور پایگاه داده جمع‌آوری می‌نماییم. از جمله داده‌های فرار می‌توان به اطلاعات مربوط به کاربران وارد شده به سامانه، پردازش‌های در حال اجرا، پورت‌های و فایل‌های باز، اشاره کرد. همچنین تمامی فایل‌های رویدادنگاری پایگاه داده، فایل‌های ردیابی و فایل‌های پیکربندی نیز باید جمع‌آوری و از آن‌ها به درستی نسخه‌برداری شود. علاوه بر این، از رویدادهای ثبت شده توسط سرور وب و برنامه‌ی کاربردی و سایر فایل‌های رویدادنگاری مهم نیز باید نسخه‌ای تهیه شود. اطلاعات موجود در حافظه نیز باید تخلیه و ثبت شوند. به عنوان نمونه، در پایگاه داده اوراکل اطلاعات موجود در SGA باید تخلیه و ثبت شوند. می‌توان آخرین پرسمان SQL اجرا شده را از SGA به دست آورد. در صورتی که بررسی جرم‌شناسی خیلی سریع انجام شود، شانس این وجود دارد که عبارت‌های SQL که قسمتی از حمله بوده‌اند، در SGA وجود داشته باشند. همچنین می‌توان نشست‌ها و پردازش‌های موجود در SGA را نیز استخراج و جمع‌آوری کرد. تقریباً هرکاری که در پایگاه داده انجام می‌شود، آن را تغییر می‌دهد. بنابراین استخراج شواهد از پایگاه داده آسیب‌دیده می‌بایست با دقت و با ترتیب درستی انجام شود. لازم به ذکر است که داده‌هایی که بیشتر در معرض تغییر قرار دارند، باید سریعتر استخراج شوند.
۸. **جمع‌آوری داده‌های غیر فرار:** پس از جمع‌آوری داده‌های فرار که حساسیت بیشتری برای جمع‌آوری اطلاعات نسبت به سایر اطلاعات را دارند، سایر شواهد را نیز از پایگاه داده جمع‌آوری می‌کنیم. از جمله این شواهد می‌توان به لیست کاربران، مجوزهای کاربران و عضویت در نقش‌ها اشاره کرد.
۹. **قطع اتصال پایگاه داده به شبکه:** پس از جمع‌آوری اطلاعات و شواهد مورد نیاز برای بررسی جرم، به منظور کاهش مخاطرات احتمالی ناشی از آن می‌بایست اتصال سامانه به شبکه را قطع نمود.

۱۰. تهیه نسخه پشتیبان: در صورت امکان از کل دیسک سخت افزاری و یا در صورت وجود محدودیت، از شواهد موجود در سرور پایگاه‌داده، نسخه پشتیبان تهیه شود.

۱۱. انجام تجزیه و تحلیل بر روی داده‌ها: در این گام، بر روی داده‌ها و شواهد جمع‌آوری شده تجزیه و تحلیل صورت می‌گیرد و سعی می‌شود تا حد امکان زمان شروع و خاتمه‌ی حمله به دست آید. لازم به ذکر است که زمان‌های به دست آمده ممکن است با بررسی‌های بیشتر، تغییر نمایند.

۱۲. ایجاد جدول زمانی از رویدادها: جدول زمانی، تمامی اطلاعات مربوط به اعمال انجام شده بر روی پایگاه‌داده را در خود جای داده است. بدین ترتیب می‌توان پی برد که:

- مهاجم چگونه به سیستم دسترسی پیدا کرده است،
- از طریق چه نام کاربری وارد شده است،
- چه اطلاعاتی را مشاهده نموده است،
- چه اعمالی را در پایگاه‌داده انجام داده است،
- با چه مجوزهایی به سیستم وارد شده است،
- و چه کارهای بیشتری را با مهارت بیشتر می‌توانست در سامانه اعمال کند.

۱۳. خاموش کردن سیستم و بازگرداندن آن به وضعیت پیش از حمله: در این گام می‌بایست با توجه به میزان اهمیت پایگاه‌داده، برای خاموش کردن آن تصمیم‌گیری شود. پیش از خاموش کردن یا بازگرداندن پایگاه‌داده، باید کاملاً مشخص شود که مهاجم چه کارهایی را انجام داده است. در صورتی که داده‌ها تنها توسط مهاجم خوانده شده باشند و هیچ تغییری در آن‌ها اعمال نشده باشد، نیازی به بازگرداندن پایگاه‌داده به نقطه‌ای پیش از حمله نیست.

۱۴. تهیه مستند از حمله: مستندسازی فرآیند و کلیه‌ی شواهد جمع‌آوری شده، بسیار مهم است. همچنین باید یافته‌ها و آنچه با بررسی به دست آمده است نیز مستند شود. تمامی اعمال مهاجم به همراه نشانه‌هایی از سرقت داده‌ها باید ثبت شوند. ثبت اطلاعاتی همچون سیستم عامل سرور و نسخه‌ی آن، نوع و نسخه‌ی پایگاه‌داده، رویداد غیرمجاز، نوع رویداد (خصمانه/ناخواسته)، شیوه ممیزی، منابع رویدادنگاری، شیوه یا ابزار تحلیل و امکان ترمیم در یک قالب استاندارد ضروری است. این مستندات برای شناسایی نقاط ضعف سامانه از اهمیت ویژه‌ای برخوردار است. بنابراین شناسایی نقاط ضعف سامانه و پیشنهادهایی برای حل آن‌ها نیز در این مستند ثبت می‌شوند.

۱۵. گزارش رویدادها و فرآیند طی شده: پس از تهیه‌ی مستند از رویدادها و فرآیند طی شده، مجموعه مستند گردآوری شده قابل ارائه به دادگاه یا سایر مقامات قانونی است.

۲-۲ فرآیند جرم‌شناسی

با استفاده از مدل‌های فرآیند جرم‌شناسی پایگاه داده می‌توان به صورت ساختاریافته‌ای عملیات مربوط به جرم‌شناسی پایگاه داده را پیش برد. در ابتدا و پیش از انجام هر عملی، باید نسبت به وقوع رویداد غیرمجاز اطمینان حاصل کرد و آن رویداد را شناسایی نمود. به عنوان مثال، در صورتی که مدیر پایگاه داده نسبت به حذف برخی از نام‌های کاربری از پایگاه داده مطلع شود، فرآیند جرم‌شناسی برای یافتن اطلاعاتی پیرامون این رویداد و ترمیم داده‌های حذف شده، آغاز می‌گردد.

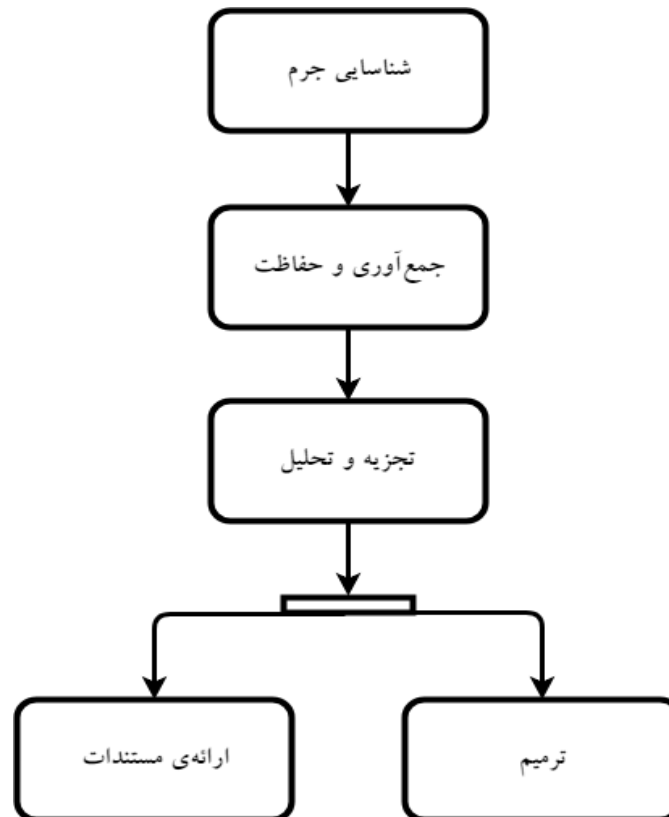
پس از شناسایی وقوع رویداد غیرمجاز می‌بایست شواهد و اطلاعات پیرامون رویداد، جمع‌آوری گردد. با توجه به اینکه پایگاه داده در یک محیط مجزا نبوده و با سیستم عامل و برنامه‌های کاربردی در ارتباط است، می‌توان شواهد را از منابع مختلف به دست آورد. در جمع‌آوری اطلاعات و شواهد توجه به این نکته حائز اهمیت است که برخی از اطلاعات همچون داده‌های موجود در حافظه، فرار بوده و هر لحظه احتمال از دست رفتن آن‌ها وجود دارد؛ بنابراین، اینگونه از اطلاعات باید هرچه سریعتر و پیش از فرا رسیدن موعد حذف، جمع‌آوری شوند. پس از آن، می‌توان اطلاعات پایدار و غیرفرار را استخراج و در مکانی امن نگهداری کرد.

اطلاعات و شواهد جمع‌آوری شده در صورتی برای تجزیه و تحلیل و همچنین ارائه در دادگاه‌های قانونی قابل قبول هستند که به درستی حفاظت شده باشند و بتوان ثابت کرد که در حین فرآیند جرم‌شناسی تغییری در آن‌ها صورت نگرفته است.

در ادامه و در مرحله‌ی تجزیه و تحلیل، با بررسی و تحلیل داده‌های جمع‌آوری شده، اطلاعاتی همچون چه کسی تغییر را ایجاد کرده، چه زمانی تغییر رخ داده، چه داده‌ای تغییر داده شده، چرا و چگونه تغییر رخ داده است، مشخص می‌شود. لازم به ذکر است که تجمیع داده‌ها در فرآیند جرم‌شناسی پایگاه داده از اهمیت بالایی برخوردار است. برخی از داده‌ها به تنهایی دارای ارزش کمی هستند ولی هنگامی که با سایر اطلاعات ترکیب می‌شوند، می‌توانند نقض امنیتی را آشکار سازند. تمامی شواهد جمع‌آوری شده و فرآیند طی شده در جرم‌شناسی می‌بایست مستند شود. مستندات و گزارش‌های تهیه شده قابل ارائه به مدیریت سازمان و دادگاه در صورت نیاز خواهند بود.

به سبب گستردگی حوزه جرم‌شناسی پایگاه داده، تمرکز ما در فرآیند جرم‌شناسی پایگاه داده تنها بر روی اطلاعات ثبت شده در پایگاه‌های داده می‌باشد و استفاده از اطلاعات ثبت شده بر روی سیستم عامل، داده‌های موجود در حافظه و شبکه نادیده گرفته شده است. لازم به ذکر است که برای داشتن طرحی جامع در این

زمینه، در نظر گرفتن کلیه مولفه‌های درگیر ضروری است. شکل ۱، گام‌های فرآیند جرم‌شناسی پایگاه‌داده در رویکرد اتخاذی را به تصویر کشیده است. در ادامه، هر یک از این گام‌ها مورد بحث و بررسی قرار گرفته است.



شکل ۱: فرآیند پیشنهادی جرم‌شناسی پایگاه‌داده منطبق با فرآیند استاندارد

شناسایی جرم:

فرآیند جرم‌شناسی با مشاهده رویدادهای غیرمجاز در سامانه آغاز می‌شود. منظور از رویدادهای غیرمجاز، رویدادهای خصمانه یا ناخواسته‌ای هستند که مدیر پایگاه‌داده انتظار وقوع آن‌ها را در شرایط فعلی ندارد. به عنوان مثال، در صورتی که مدیر پایگاه‌داده کاربرانی را در جدولی از پایگاه‌داده تعریف کرده باشد، درج کاربر جدیدی که توسط مدیر انجام نشده است، یک رویداد غیرمجاز به شمار می‌آید.

برخی از رویدادهای قابل بررسی در فرآیند جرم‌شناسی در سامانه مدیریت پایگاه‌داده عبارتند از:

- **درج، حذف و مشاهده‌ی غیرمجاز محتوای جداول:** رویدادهای غیرمجازی هستند که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شوند و موجب درج سطر (های) جدید، حذف سطر (های) موجود یا مشاهده تمامی یا بخشی از جدول (ها) در پایگاه‌داده

می‌گردند. سه رویداد فوق، از جمله دستورات دستکاری داده (DML)^۲ در پایگاه‌داده به حساب می‌آیند.

- **بروزرسانی غیرمجاز داده‌های جداول:** رویداد غیرمجازی است که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شود و سبب تغییر سطر (های) موجود می‌گردد. این رویداد نیز از جمله دستورات DML در پایگاه‌داده است. لازم به ذکر است که به سبب اهمیت بروزرسانی داده‌ها و چالش‌هایی که برای شناسایی وقوع جرم وجود دارد، این رویداد در برخی از سمپادها از دیگر رویدادهای مربوط به دستورات دستکاری داده متمایز شده است.

- **تغییر غیرمجاز شمای پایگاه‌داده:** رویداد غیرمجازی است که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شود و سبب تغییر شمای جدول (ها) در پایگاه‌داده می‌گردد. این رویداد توسط دستورات تعریف داده (DDL)^۳ در پایگاه‌داده قابل اعمال است.

- **تلاش برای ورود غیرمجاز به پایگاه‌داده:** رویداد غیرمجازی است که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شود و به طور معمول دو هدف زیر را دنبال می‌کند:

- حدس نام کاربری و کلمه عبور کاربران مجاز جهت ورود به سامانه.
- تلاش برای شناسایی شناسه یکتای سامانه^۴ (SID) و نهایتاً دسترسی به حساب‌های کاربری و اعمال نفوذ در سامانه.

لازم به ذکر است که در برخی سمپادها، این تلاش تنها از طریق حدس نام کاربری و کلمه عبور کاربران مجاز جهت ورود به سمپاد امکان‌پذیر است.

- **تغییر غیرمجاز در فایل‌های رویدادنگاری و ممیزی:** رویداد غیرمجازی است که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شود و هدف آن از بین بردن فایل‌های رویدادنگاری به منظور پاک کردن شواهد رویدادهای مجرمانه در پایگاه‌داده است.

جمع‌آوری اطلاعات و شواهد:

² Data Manipulation Language 3
² Data Definition Language 4
² System Identifier 5

مصنوعات و اطلاعات مختلف برای شناسایی رویدادها را می‌توان از پایگاه داده، سیستم عامل، سرورهای وب یا فایل‌های رویدادنگاری استخراج کرد. جمع‌آوری مصنوعات و شواهد می‌تواند سبب تغییر در پایگاه داده شود. از این‌رو، پیش از استخراج اطلاعات از داخل یا خارج پایگاه داده می‌بایست نسبت به پایدار یا ناپایدار بودن اطلاعات آگاهی پیدا کرد. هر پایگاه داده شواهد مربوط به اعمال مختلف را در بخش‌ها و فایل‌های رویدادنگاری مختلف ذخیره می‌کند. این بدان معناست که به منظور تجزیه و تحلیل جرم‌شناسی می‌بایست نسبت به چگونگی عملکرد پایگاه داده، محل فایل‌ها و مصنوعات مختلف اطلاع داشت. از آنجا که تمرکز ما در فرآیند جرم‌شناسی پایگاه داده تنها بر روی اطلاعات حاصل از رویدادنگاری و ممیزی در پایگاه داده است، لذا اطلاعات مورد هدف برای گردآوری تنها به پایگاه داده محدود شده و شامل اطلاعات موجود در سیستم عامل و حافظه‌ی اصلی نمی‌باشد.

در صورتی که بر روی فایل‌های رویدادنگاری و ممیزی به صورت دوره‌ای اطلاعاتی ثبت شود، به هنگام شناسایی رویداد غیرمجاز می‌بایست ابتدا سرور پایگاه داده متوقف شود تا از اجرای پرسمان‌های جدید بر روی سرور خودداری گردد. بدین ترتیب از نگاشته شدن بر روی اطلاعات ثبت شده در فایل‌های رویدادنگاری و ممیزی جلوگیری می‌شود. در نهایت نیز به منظور انجام تحلیل‌های آتی می‌توان از فایل‌های مورد نیاز، یک نسخه‌ی پشتیبان تهیه نمود [۱۲].

مرحله جمع‌آوری اطلاعات و شواهد، شامل دو گام زیر می‌باشد:

۱. **تعیین نحوه‌ی جمع‌آوری اطلاعات:** در صورت مشاهده علائمی از وقوع رویداد غیرمجاز در پایگاه داده، با توجه به نوع رویداد می‌توان از منابع مختلفی برای جمع‌آوری اطلاعات استفاده کرد. به عنوان نمونه، برای جمع‌آوری اطلاعات مربوط به برخی از رویدادها می‌بایست از رویدادهای ثبت شده در فایل‌های رویدادنگاری و برای برخی دیگر می‌بایست از تعریف خط‌مشی‌هایی ممیزی استفاده کرد.
۲. **بررسی تنظیمات فعلی:** پس از آنکه منابع جمع‌آوری اطلاعات برای یک رویداد ناخواسته مشخص شدند، تنظیمات فعلی سیستم بررسی می‌شود. با بررسی اولیه‌ی تنظیمات مشخص می‌شود که آیا اقدامات اولیه برای ثبت اطلاعات قبل از وقوع رویداد انجام شده‌اند یا خیر. به عنوان نمونه، بررسی می‌شود که آیا در پایگاه داده، خط‌مشی‌هایی برای ثبت اطلاعات ممیزی مربوط به جرم تعریف شده است یا خیر.

استخراج و تجزیه و تحلیل اطلاعات:

در این مرحله ابتدا ابزارها و پرسمان‌های هدف به منظور استخراج اطلاعات از منابع تعیین می‌شوند. یک روش محبوب در میان تحلیل‌گران جرم‌شناسی، استخراج عبارات SQL اجرا شده در پایگاه داده است. همچنین بی‌شک یکی از المان‌های بسیار مهم در تجزیه و تحلیل جرم‌شناسی، شناسایی هویت مجرم است. هر عملی که ثبت می‌شود را باید بتوان به یک شخص واقعی نسبت داد [۱۵]. سپس اطلاعات هدف از منابع مشخص شده استخراج

و مرحله بررسی و تحلیل اطلاعات آغاز می‌گردد. در این مرحله با بررسی و تحلیل داده‌های جمع‌آوری شده، اطلاعاتی همچون عامل وقوع رویداد، زمان وقوع، دلایل وقوع، نوع تغییر حاصل از اجرای رویداد، داده متأثر از تغییر و چگونگی اعمال رویداد غیرمجاز، آشکار می‌شود [۱۰، ۱۱]. لازم به ذکر است که تجمیع داده‌ها در فرآیند جرم‌شناسی پایگاه‌داده بسیار حائز اهمیت است. به منظور سادگی در بررسی هر یک از رویدادهای ناخواسته، بخش استخراج اطلاعات و تجزیه و تحلیل با یکدیگر ادغام شده و تحت عنوان استخراج و تجزیه و تحلیل اطلاعات بیان می‌شود.

ترمیم:

پس از محرز شدن رخداد جرم در سامانه پایگاه‌داده، متناسب با نیاز و شواهد اطلاعاتی موجود، ممکن است ترمیم پایگاه‌داده امکان‌پذیر باشد. در واقع، ترمیم سامانه پایگاه‌داده بدین معناست که وضعیت پایگاه‌داده را به وضعیتی پیش از وقوع رویداد برگردانیم. به عنوان نمونه، در صورتی که داده‌های حساس از یک جدول حذف شده باشند، این مرحله به بازیابی داده‌های حذفی می‌پردازد.

ارائهی مستندات:

در گام نهایی می‌بایست کلیه‌ی بررسی‌های صورت پذیرفته را در یک قالب استاندارد از پیش تعیین شده به نحوی مستند نمود که قابل ارائه به مدیریت سازمانی یا دادگاه قانونی برای طرح دعوی باشد. مستندسازی به سایر بررسی‌کنندگانی که سناریوی مشابهی را تجربه می‌کنند نیز کمک خواهد کرد. به منظور ارائه مستندات مربوط به رویدادهای غیرمجاز کشف‌شده در سامانه، فرم استاندارد زیر ارائه شده است.

جدول ۱: تهیه‌ی مستند از فرآیند جرم‌شناسی پایگاه‌داده

عنوان رویداد		
وقوع جرم	<input type="checkbox"/> عدم وقوع	<input type="checkbox"/> وقوع خصمانه
شیوه ممیزی	<input type="checkbox"/> وقوع ناخواسته	
منابع رویدادنگاری		
شیوه یا ابزار تحلیل		
امکان ترمیم		

۲-۳ رهنمون‌های فرآیند جرم‌شناسی

در این بخش، برخی از مهم‌ترین رهنمون‌های مربوط به مراحل مختلف فرآیند جرم‌شناسی پایگاه‌داده آورده شده است. لازم به ذکر است که ملاحظات ذکر شده در این بخش برای هر سامانه پایگاه‌داده در هر سازمانی

حیاتی است. با این وجود، ممکن است متناسب با نوع سازمان و نوع سامانه پایگاه‌داده، ملاحظات دیگری نیز افزوده شود [۱۲-۱۵].

۱. در صورت بروز نقض امنیتی، کل سازمان باید از آن مطلع شده و نشست آموزشی کوتاهی در زمینه رسیدگی به آن با حضور تمامی افراد برگزار شود. در صورتی که اطلاع‌رسانی به بخش‌های مختلف سازمان و تجزیه و تحلیل جرم‌شناسی توسط بخش‌های مختلف، طبق برنامه‌ی مشخصی انجام نشود، بخش‌هایی ممکن است موضوع را نادیده بگیرند و بخش‌های دیگری ممکن است آن را به اطلاع عموم برسانند. بنابراین، هرآنچه که در روند جرم‌شناسی انجام می‌شود می‌بایست طبق برنامه از پیش تدوین شده، صورت پذیرد.
۲. یکی از چالشی‌ترین مسائل در بررسی پایگاه‌های داده آن است که هر چه بیشتر در پایگاه‌داده جستجو و بررسی انجام شود، تغییرات بیشتری در آن رخ می‌دهد. لازم به ذکر است که تقریباً هر عملی که در پایگاه‌داده انجام می‌شود، می‌تواند باعث تغییر در رکوردهای دیکشنری شود.
۳. استفاده از حساب کاربری با مجوزهای بالا می‌تواند خود زمینه ایجاد تغییرات در پایگاه‌داده را فراهم آورد. بنابراین، در نظر گرفتن حساب کاربری با دسترسی‌های محدود فقط خواندنی برای این منظور ایده‌آل است. اگر چنین حساب‌های کاربری قبل از وقوع جرم تعریف نشده باشد، ایجاد حساب کاربری جدید توصیه نمی‌شود؛ چرا که خود موجب تغییرات جدیدی در سامانه می‌شود و با وجود اینکه استفاده از کاربر SYS می‌تواند خطرناک باشد ولی منطقی‌تر است.
۴. برای قابل قبول بودن شواهد در دادگاه قانونی به هنگام طرح دعوی، دو اصل اساسی باید رعایت شود: (۱): اولین اصل آن است که می‌بایست ثابت شود که اطلاعات و شواهد گردآوری شده به صورت قانونی به دست آمده است و (۲): دومین اصل نیز آن است که می‌بایست ثابت شود که به هنگام جمع‌آوری شواهد و اطلاعات تغییری در آنها اعمال نشده است. برای اثبات اصل اول، پیش از جمع‌آوری شواهد باید قانونی بودن آنها بررسی و تأیید شود. همچنین اصل دوم نیز با مستندسازی شواهد و اعمال اجرا شده بر روی آنها قابل اثبات است. در حقیقت جمع‌آوری شواهد باید طی گام‌های از پیش تدوین شده‌ای صورت گیرد تا اطمینان حاصل شود که شواهد جمع‌آوری شده، تغییر نکرده‌اند. برای اثبات صحت داده‌ها می‌توان از روش مجموع مقابله‌ای^۲ استفاده کرد. این روش می‌تواند بر روی شواهد مختلف از جمله یک فایل یا کل دیسک سخت افزاری اعمال شود. با این روش می‌توان ثابت کرد که شواهد جمع‌آوری شده همان اطلاعاتی است که در ادامه از آنها برای تجزیه و تحلیل استفاده می‌شود و بدون تغییر به دادگاه قابل ارائه است. توجه به این نکته حائز اهمیت است که برای استفاده از روش

مجموع مقابله‌ای نباید به راحتی از بسته‌های موجود در پایگاه‌های داده استفاده شود. به عنوان مثال، در پایگاه‌داده اوراکل، نباید از بسته‌ی DBMS_SQLHASH استفاده شود چون ممکن است خود این بسته توسط مهاجم تغییر داده شده باشد. بنابراین در طول بررسی‌های جرم‌شناسی باید بتوان اعتبار ابزارها و عدم تغییر آن‌ها را ثابت کرد. مهاجم همچنین می‌تواند دیدهای موجود در پایگاه‌داده را برای مخفی نگه داشتن اعمال خود تغییر دهد. بنابراین باید یا از جدول‌های پایه در بررسی‌های جرم‌شناسی استفاده شود و یا پیش از استفاده از دیدها از صحت آن‌ها اطمینان حاصل گردد.

۵. یکی از مصنوعات مهم پایگاه‌داده برای تجزیه و تحلیل جرم‌شناسی، رویدادهای ثبت شده توسط ممیزی است. استفاده از ممیزی این اطمینان را ایجاد می‌کند که همیشه شواهدی برای استفاده در تحلیل جرم‌شناسی وجود دارد. می‌توان پایگاه‌داده را برای ثبت رویدادهای مشخص همچون ایجاد یک کاربر، تغییر رمز عبور و دسترسی به محتوای یک جدول تنظیم کرد. در صورتی که ممیزی فعال نباشد می‌توان از سایر قابلیت‌های رویدادنگاری در پایگاه‌های داده برای ثبت تغییرات استفاده کرد. با این وجود، به کارگیری ممیزی با جزئیات کافی و ثبت تمامی اعمال اجرا شده توسط مهاجم، تجزیه و تحلیل جرم‌شناسی را ساده‌تر می‌کند. تنظیمات مربوط به ممیزی می‌بایست بر اساس برنامه‌ای از قبل تعیین شده، مشخص گردد. در واقع بر اساس این برنامه، هرآنچه که نیاز به دانستن است، مشخص می‌گردد. در ادامه، برخی از امکان‌ها برای انجام تنظیمات ممیزی ارائه شده‌اند:

- افرادی که به پایگاه‌داده وارد یا از آن خارج می‌شوند.
- افرادی که خود را به جای مدیر پایگاه‌داده نمایش می‌دهند.
- تلاش برای حمله‌ی تزریق^{۲۷} SQL
- تغییر در پروفایل کاربر
- تغییر در ساختار پایگاه‌داده

یک راه حل جامع برای تهیه‌ی ممیزی باید شامل ممیزی از خود ممیزی نیز باشد. در صورتی که مهاجم تلاش به حذف یک رکورد ممیزی کند، باید این عمل ثبت شود. همچنین تلاش برای تغییر تنظیمات ممیزی نیز باید ثبت گردد. در صورتی که ممیزی در پایگاه‌داده فعال باشد، اولین گام شناسایی تنظیمات ممیزی است. پس از آن می‌توان به رویدادهای ممیزی و استفاده از آن‌ها در بررسی‌های جرم‌شناسی پرداخت.

۶. شواهد و اطلاعات مربوط به جرم‌شناسی اغلب در مکان‌های مختلفی از پایگاه داده وجود دارند. شناخت شواهد و اطلاعات مهم و اولویت‌بندی آنها از اهمیت بالایی برخوردار است. به غیر از اطلاعاتی که می‌توان از طریق اجرای پرسمان بر روی پایگاه داده‌ی تغییر یافته به دست آورد، می‌توان از طریق ابزارهایی که توسط سامانه پایگاه داده استفاده می‌شوند؛ همچون نهان‌گاه طرح اجرایی^۸ و رویدادنگاری تراکنش‌ها^۹ نیز شواهدی را جمع‌آوری کرد. یک طرح اجرایی، کارآمدترین روش اجرای درخواست‌های داده است که در نهان‌گاه طرح اجرایی برای استفاده‌ی مجدد ذخیره می‌شوند. رویدادنگاری تراکنش‌ها در شناخت پرسمان‌های اجرا شده بر روی پایگاه داده و برای بررسی‌ها و تحقیقات مختلف، مفید است. سایر منابع شامل فایل‌هایی هستند که تاریخچه‌ی مربوط به پایگاه داده را ذخیره می‌کنند. برخی از این فایل‌ها به طور اختصاصی در پایگاه داده کاربرد دارند، همچون فایل رویدادنگاری پایگاه داده و فایل‌های داده در حالی که سایر فایل‌ها همچون رویدادنگاری سرور وب و رویدادنگاری رویدادهای سیستمی یک سیستم عامل به طور خاص به سرور پایگاه داده اختصاص داده نمی‌شوند. در هنگام تصمیم‌گیری در مورد اینکه کدام داده اول جمع‌آوری شود، مهم است که سطح بی‌ثباتی یک فایل در نظر گرفته شود.

۷. یکی از بزرگترین مسائل در تجزیه و تحلیل جرم‌شناسی در پایگاه داده آن است که به صورت معمول پایگاه‌های داده، رویدادهای مربوط به دسترسی و خواندن محتوای جداول را ثبت نمی‌کنند ولی در تمامی مواقع، تغییرات در پایگاه داده همچون بروزرسانی، درج و حذف داده‌ها ثبت می‌شوند. گاهی نیاز است که شواهدی برای بررسی سرقت داده‌ها شناسایی شود. سرقت داده‌ها، ممکن است به نحوی باشد که داده‌ها را از پایگاه داده حذف نکند. بنابراین می‌بایست به دنبال شواهدی برای دسترسی به داده‌ها از طریق پایگاه داده بود. دسترسی به این گونه شواهد معمولاً پیچیده است مگر اینکه ممیزی فعال باشد. در حقیقت تنها روشی که می‌توان از آن به طور حتم برای اثبات دسترسی به داده مشخص استفاده کرد، فعال کردن ممیزی روی آن پیش از دسترسی است. به طور طبیعی، ثبت فعالیت خواندن داده‌ها ایده‌آل است هرچند همیشه انجام نمی‌شود. بنابراین باید بتوان در کنار آن، از راه‌های جایگزین نیز استفاده کرد. بدین منظور اصولاً از همبستگی^{۱۰} میان شواهد استفاده می‌شود. به عنوان مثال، در پایگاه داده اوراکل، ممکن است شواهدی مبنی بر ورود مهاجم و ایجاد اتصال به پایگاه داده وجود داشته باشد. مهاجم ممکن است پرسمان SELECT را وارد کرده باشد و بهینه‌ساز^{۱۱} اوراکل وارد عملی

2	Execution plan cache	8
2	Transaction logs	9
3	Correlation	0
3	Optimizer	1

برای کامپایل دستور SQL شده باشد. همچنین در صورتی که حمله بلافاصله مورد بررسی قرار گرفته باشد، پرسمان SQL استفاده شده توسط مهاجم ممکن است در SGA وجود داشته باشد. در این شرایط، در صورتی که حمله از طریق سرور وب انجام شده باشد، پرسمان SQL ممکن است در فایل رویدادنگاری مربوط به برنامه‌ی کاربردی تحت وب موجود باشد. بنابراین در صورتی که ممیزی بر روی سیستم فعال نباشد، باید از همبستگی میان شواهد مختلف برای نتیجه‌گیری در مورد سرقت اطلاعات استفاده شود.

۸. برای جلوگیری از حجیم شدن فایل‌های رویدادنگاری و ممیزی می‌توان خط‌مشی‌هایی را برای بازنویسی^{۳۲} رویدادها در این گونه از فایل‌ها اعمال کرد. به عنوان مثال می‌توان مشخص کرد که در صورتی که حجم فایل‌های رویدادنگاری به ۱۰۰ مگابایت رسید، اطلاعات جدید از ابتدای فایل بر روی اطلاعات قبلی نوشته شود یا در یک دوره هفت روزه، فایل‌های رویدادنگاری جدید ایجاد شوند. در صورتی که بر روی فایل‌های رویدادنگاری و ممیزی به صورت دوره‌ای اطلاعات ثبت شود، هنگام تشخیص رویداد غیرمجاز، باید ابتدا سرور پایگاه‌داده متوقف شود تا از اجرای اعمال و پرسمان‌های جدید بر روی سرور خودداری شود. سپس می‌توان از فایل‌های مورد نیاز، نسخه‌ی پشتیبان تهیه کرد تا تحلیل‌های آتی بر روی نسخه‌های پشتیبان انجام شود.

۹. در صورتی که خط‌مشی‌هایی برای تهیه‌ی پشتیبان از پایگاه‌داده به صورت دوره‌ای وجود داشته باشد، وضعیت مطلوبی که پایگاه‌داده پیش از این، در آن قرار داشته است در دسترس خواهد بود. در نتیجه، در صورت وقوع رویداد غیرمجاز با بازگرداندن فایل‌های پشتیبان می‌توان به حالت مطلوب پیش از رویداد غیرمجاز تغییر وضعیت داد.

۱۰. در صورت وجود فایل‌های رویدادنگاری و ممیزی بر روی سیستمی که در آن پایگاه‌داده وجود دارد، مدیر پایگاه‌داده یا کاربر مخرب می‌تواند به طور موقت تهیه‌ی ممیزی و رویدادنگاری را متوقف کرده و اعمال مورد نظر خود را اجرا کند و یا تغییراتی را به صورت دستی در فایل‌های ممیزی و رویدادنگاری ایجاد کند. در صورتی که چندین نسخه از داده‌های حساس در مکان‌های مختلف وجود داشته باشد، امکان از دست رفتن داده‌های حساس در صورت حذف آن‌ها از یک مکان، کاهش می‌یابد. همچنین مطلوب است که تمامی رویدادها به سیستم مرکزی ارسال شده و از آن‌ها به طور مرکزی حفاظت و نگهداری شود.

۱۱. معمولاً مهاجمین پس از حمله به پایگاه‌داده، سعی به حذف ردپای خود در فایل‌های رویدادنگاری و ممیزی می‌کنند. بنابراین محدود کردن دسترسی به این فایل‌ها و کپی‌برداری از آن‌ها و ذخیره‌سازی در سرور مرکزی از اهمیت زیادی برخوردار است. همچنین با تشخیص اعمال تغییر غیرمجاز در فایل‌های رویدادنگاری و ممیزی می‌توان متوجه شد که رویدادهای ثبت شده در این فایل‌ها فاقد اعتبار هستند. به عنوان نمونه، هنگام اجرای پرسمان بر روی فایل‌های رویدادنگاری redo log مربوط به پایگاه‌داده اوراکل، در صورتی که به صورت دستی تغییری در این فایل‌ها اعمال شده باشد، یکی از خطاهای موجود در شکل ۲ و شکل ۳ نشان داده می‌شود که نشان‌دهنده تغییر غیرمجاز در این فایل‌ها و عدم اعتبار اطلاعات ذخیره شده، است.

```
ORA-00308: cannot open archived log 'C:\Users\Desktop\prade\oradata\ord\REDO02.LOG'
ORA-27046: file size is not a multiple of logical block size
OSD-04012: file size mismatch (OS 209715713)
00308. 00000 - "cannot open archived log '%s'"
*Cause: The system cannot access a required archived redo log file.
*Action: Check that the off line log exists, the storage device is
         online, and the archived file is in the correct location.
         Then attempt to continue recovery or restart the recovery
         session.
```

شکل ۲: خطای تغییر غیرمجاز در فایل رویدادنگاری

```
ORA-00368: checksum error in redo log block
ORA-00353: log corruption near block 132010 change 21987320 time 02/02/2018 10:11:30
ORA-00334: archived log: 'C:\USERS\DESKTOP\ORACLE\ORADATA\ORCL\REDO02.LOG'
00368. 00000 - "checksum error in redo log block"
*Cause: The redo block indicated by the accompanying error, is not
         valid. It has a checksum that does not match the block contents.
*Action: Do recovery with a good version of the log or do time based
         recovery up to the indicated time. If this happens when archiving,
         archiving of the problem log can be skipped by clearing the log
         with the UNARCHIVED option. This must be followed by a backup of
         every datafile to insure recoverability of the database.
*Action: Restore correct file or reset logs.
```

شکل ۳: خطای تغییر غیرمجاز در فایل رویدادنگاری

۲-۴ جمع‌بندی

در این فصل، ابتدا بستر مورد نیاز برای جرم‌شناسی پایگاه‌داده را تحت عنوان تمهیدات جرم‌شناسی به طور مشروح مورد بحث و بررسی قرار دادیم. سپس فرآیند جرم‌شناسی را از شناسایی جرم تا تهیه و ارائه مستندات به مدیریت سازمان و دادگاه‌قانونی برای طرح دعوی تشریح کردیم. همچنین به منظور ارائه مستندات مربوط به رویدادهای غیرمجاز کشف‌شده در سامانه، فرم استاندارد زیر ارائه گردید.

جدول ۲: تهیه‌ی مستند از فرآیند جرم‌شناسی پایگاه‌داده

عنوان رویداد

وقوع جرم	عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>
شیوه ممیزی			
منابع رویدادنگاری			
شیوه یا ابزار تحلیل			
امکان ترمیم			

۳ درج، حذف، بروزرسانی و مشاهده‌ی غیرمجاز محتوای جداول

در این فصل به بحث و بررسی رویدادهای غیرمجازی که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شوند و موجب درج سطر (های) جدید، حذف و تغییر سطر (های) موجود یا مشاهده تمامی یا بخشی از جدول (ها) در پایگاه‌داده می‌گردند، پرداخته می‌شود. رویدادهای فوق از جمله دستورات دستکاری داده (DML)^۳ در پایگاه‌داده به حساب می‌آیند. از آنجاییکه رویدادهای مورد بحث در این فصل از نقطه نظر جرم‌شناسی به هم نزدیک هستند، لذا تنها بر روی رویداد حذف غیرمجاز به عنوان یک نماینده از این گروه رویدادها متمرکز شده و در صورت نیاز تفاوت‌های دیگر رویدادها ذکر می‌گردد.

۳-۱ شناسایی جرم

فرآیند جرم‌شناسی با مشاهده‌ی رویدادهای غیرمجاز در سامانه آغاز می‌شود. بنابراین در صورتیکه مدیر پایگاه‌داده خود به طور مستقیم یا غیر مستقیم (از طریق کاربران)، نسبت به حذف داده‌ها از جدولی آگاهی پیدا کند که انتظار حذف آنها در شرایط فعلی را نداشته است، فرآیند جرم‌شناسی در سامانه آغاز می‌شود.

۳-۲ جمع‌آوری اطلاعات و شواهد

از آنجاییکه هر پایگاه‌داده شواهد مربوط به اعمال مختلف را در بخش‌ها و فایل‌های رویدادنگاری مختلف ذخیره می‌کند، در این بخش قصد داریم منابع مربوط به شواهد رویداد غیرمجاز حذف را تعیین نماییم. لازم به ذکر است که تمرکز ما در فرآیند جرم‌شناسی پایگاه‌داده تنها بر روی اطلاعات حاصل از رویدادنگاری و ممیزی در پایگاه‌داده است و شامل اطلاعات موجود در سیستم عامل و حافظه‌ی اصلی نمی‌باشد. از این‌رو، در ادامه به معرفی منابع مربوط به شواهد رویداد غیرمجاز حذف و نحوه آگاهی و تنظیم آنها می‌پردازیم.

³ Data Manipulation Language

رویدادننگاری **mongod.log**: تمامی پرمسرها و اعمال اجرایی در پایگاه‌داده‌ی MongoDB را می‌توان در فایل رویدادننگاری **mongod.log** ثبت و ذخیره کرد. پس از نصب MongoDB بر روی ویندوز ۷، می‌توان در مسیر نصب، فایل پیکربندی برای راه‌اندازی سرویس Mongod با نام **mongod.cfg** ایجاد کرد و تنظیمات لازم را در آن قرار داد. نمونه‌ای از این فایل در شکل ۴ دیده می‌شود.

```
systemLog:
  destination: file
  path: c:\data\log\mongod.log
  verbosity: 1
storage:
  dbPath: c:\data\db
auditLog:
  destination: file
  format: JSON
  path: c:\data\log\auditlog.json
setParameter: { auditAuthorizationSuccess: true, logUserIds: 1 }
replication:
  oplogSizeMB: 100
  replSetName: rs0

operationProfiling:
  mode: all
```

شکل ۴: فایل پیکربندی سرویس mongod

همانطور که در شکل ۴ دیده می‌شود، محل ذخیره‌سازی محتوای پایگاه‌های داده در قسمت **storage.dbpath** و محل ذخیره‌سازی رویدادها در قسمت **systemLog.path** تنظیم می‌شود. سطح جزئیات پیام‌های خطا با اعداد صفر تا پنج تنظیم می‌شود. صفر، سطح پیش‌فرض است و دارای جزئیات کمتری است. پنج، سطح با جزئیات بیشتر است. در شکل ۴ با استفاده از پارامتر **verbosity** سطح جزئیات پیام‌های تمامی مؤلفه‌ها، یک تنظیم شده است. علاوه بر این، برای مشاهده‌ی سطح جزئیات پیام‌ها برای هر یک از مؤلفه‌ها می‌توان از دستور زیر استفاده کرد:

```
db.getLogComponents()
```

سطح پیش‌فرض برای تمامی مؤلفه‌ها صفر است. همان‌طور که در شکل ۵ دیده می‌شود، این سطح برای تمامی مؤلفه‌ها به یک تغییر داده شده است (عدد 1- به معنی ارث‌بری سطح جزئیات از مؤلفه‌ی والد است).

```

"verbosity" : 1,
  "accessControl" : {
    "verbosity" : -1
  },
  "command" : {
    "verbosity" : -1
  },
  "control" : {
    "verbosity" : -1
  },
  "executor" : {
    "verbosity" : -1
  },
  "geo" : {
    "verbosity" : -1
  },
  "index" : {
    "verbosity" : -1
  },
  "network" : {
    "verbosity" : -1,
    "asio" : {
      "verbosity" : -1
    },
    "bridge" : {
      "verbosity" : -1
    }
  }
},

```

شکل ۵: سطح جزئیات پیغام‌های خطا

با استفاده از دستور زیر می‌توان محل فایل‌های رویدادنگاری و داده در MongoDB را استخراج کرد (شکل ۶).

```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

```

{
  "argv" : [
    "C:\\Program Files\\MongoDB\\Server\\3.4\\bin\\mongod.exe",
    "--config",
    "C:\\Program Files\\MongoDB\\Server\\3.4\\mongod.cfg",
    "--service"
  ],
  "parsed" : {
    "auditLog" : {
      "destination" : "file",
      "format" : "JSON",
      "path" : "c:\\data\\log\\auditlog.json"
    },
    "config" : "C:\\Program Files\\MongoDB\\Server\\3.4\\mongod.cfg",
    "operationProfiling" : {
      "mode" : "all"
    },
    "replication" : {
      "oplogSizeMB" : 100,
      "replSetName" : "rs0"
    },
    "service" : true,
    "setParameter" : {
      "auditAuthorizationSuccess" : "true",
      "logUserIds" : "1"
    },
    "storage" : {
      "dbPath" : "c:\\data\\db"
    },
    "systemLog" : {
      "destination" : "file",
      "path" : "c:\\data\\log\\mongod.log",
      "verbosity" : 1
    }
  },
  "ok" : 1
}

```

شکل ۶: یافتن محل فایل‌های رویدادنگاری

جدول زیر، گام‌های جمع‌آوری اطلاعات و شواهد مربوط به رویداد غیرمجاز حذف محتوای جداول را با استفاده از رویدادنگاری mongod.log نشان می‌دهد.

تنظیم رویدادنگاری در فایل پیکربندی		
destination: file	تعیین ذخیره‌سازی رویدادها در فایل	۱
path: c:\data\log\mongod.log	تعیین مسیر ذخیره‌سازی رویدادها	۲
verbosity: 1	تعیین سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۳
برخی دستورات سودمند		
db.getLogComponents()	مشاهده‌ی سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۱
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به رویدادنگاری	۲

استفاده از **پروفایلر**^۳: پروفایلر اطلاعات دقیقی در مورد دستورات اجرا شده بر روی نمونه‌ی mongod را جمع‌آوری می‌کند. این دستورات شامل اعمال CRUD و دستورات مدیریتی و پیکربندی است. پروفایلر تمامی داده‌های جمع‌آوری شده را در مجموعه‌ی system.profile ذخیره می‌کند. قابلیت پروفایلر به صورت پیش‌فرض غیرفعال است. پروفایلر سه سطح دارد:

- سطح صفر: پروفایلر را غیرفعال می‌کند.
- سطح یک: اعمال زمانبری که اجرای آن‌ها از حد آستانه‌ای بیشتر زمان صرف آنها می‌شود را ثبت می‌کند.
- سطح دو: تمامی اعمال را ثبت می‌کند.

با استفاده از دستور زیر می‌توان وضعیت پروفایلر برای یک پایگاه‌داده را مشاهده کرد:

```
db.getProfilingStatus()
```

³ Profiler 4
³ Instance 5

خروجی دستور بالا به صورت زیر است:

```
{ "was" : 2, "slowms" : 100 }
```

در این خروجی، عدد ۲ نشان دهنده‌ی ثبت تمامی اعمال توسط پروفایلر است.

اطلاعات تکمیلی: با استفاده از دستور زیر می‌توان سطح پروفایلر در یک پایگاه داده را تغییر داد:

```
db.setProfilingLevel(2)
```

همچنین می‌توان در فایل پیکربندی `mongod.cfg`، تنظیمات پروفایلر را وارد کرد:

```
operationProfiling:
```

```
mode: <string>
```

```
slowOpThresholdMs: <int>
```

```
slowOpSampleRate: <double>
```

به عنوان مثال با وارد کردن تنظیمات به صورت زیر در فایل `mongod.cfg`، سطح پروفایلر برای تمامی پایگاه‌های داده ۲ می‌شود:

```
operationProfiling:
```

```
mode: all
```

جدول زیر، گام‌های جمع‌آوری اطلاعات و شواهد مربوط به رویداد غیرمجاز حذف محتوای جدول را با استفاده از پروفایلر نشان می‌دهد.

تنظیم سطح پروفایلر		
<code>db.setProfilingLevel(<LEVEL_NUMBER>)</code>	تنظیم سطح پروفایلر با دستور	۱
<pre>operationProfiling: mode: <LEVEL_STRING></pre>	تنظیم سطح پروفایلر در فایل پیکربندی	۲
برخی دستورات سودمند		
<code>db.getProfilingStatus()</code>	بررسی وضعیت پروفایلر برای یک پایگاه داده	۱

Oplog: به منظور ثبت اعمال نوشتاری برای استفاده در تکرار‌گاری‌ها دارد. Oplog یک مجموعه است که در آن تمامی اعمالی که به نحوی داده‌های ذخیره شده در پایگاه‌داده را تغییر می‌دهند، ثبت می‌شود. MongoDB اعمال و پرسمان‌ها را بر روی سرور اصلی اعمال و بر روی Oplog در سرور اصلی ذخیره می‌کند. اعضای ثانویه، Oplog را کپی و عملیات‌های درون آن را اعمال می‌کنند. تمامی اعضای مجموعه‌ی تکرار‌گاری یک نسخه از Oplog هستند. به منظور مشاهده‌ی تنظیمات مربوط به مجموعه‌ی کپی‌ها، می‌توان از دستور زیر استفاده کرد (شکل ۷).

```
db.adminCommand( { replSetGetStatus : 1 } )
```

```
{
  "set" : "rs0",
  "date" : ISODate("2018-09-27T05:37:42.710Z"),
  "myState" : 1,
  "term" : NumberLong(29),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1538026656, 1),
      "t" : NumberLong(29)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1538026656, 1),
      "t" : NumberLong(29)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1538026656, 1),
      "t" : NumberLong(29)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "127.0.0.1:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 4282,
      "optime" : {
        "ts" : Timestamp(1538026656, 1),
        "t" : NumberLong(29)
      },
      "optimeDate" : ISODate("2018-09-27T05:37:36Z"),
      "electionTime" : Timestamp(1538022414, 1),
      "electionDate" : ISODate("2018-09-27T04:26:54Z"),
      "configVersion" : 1,
      "self" : true
    }
  ],
  "ok" : 1
}
```

شکل ۷: مشاهده‌ی تنظیمات مربوط به مجموعه‌ی کپی‌ها

همانطور که در شکل ۷ دیده می‌شود، نامی که برای مجموعه‌ی کپی‌ها در نظر گرفته شده است، به همراه اعضای آن مشخص شده‌اند.

اطلاعات تکمیلی: به منظور پیکربندی تکرار در MongoDB می‌توان از مدل ارباب/برده^۸ یا از مجموعه‌ی کپی‌ها^۹ استفاده کرد.

در مدل ارباب/برده کافی است که سرویس mongod در گره^{۱۰} اصلی با دستوری مشابه دستور زیر راه‌اندازی شود:

```
mongod.exe --config "C:\Program Files\MongoDB\Server\3.4\mongod.cfg" --master
```

حتی در صورتی که گره‌ی برده وجود نداشته باشد، دستور بالا کمک می‌کند که بتوان از قابلیت‌ها و اطلاعات ثبت شده در Oplog استفاده کرد. در این صورت در پایگاه داده‌ی local، مجموعه‌ای با نام oplog.\$main وجود خواهد داشت.

برای پیکربندی مجموعه‌ی کپی‌ها، کافی است که در فایل پیکربندی mongod.cfg تنظیماتی مشابه آنچه در ادامه بیان شده است، وارد شود:

replication:

oplogSizeMB: 100

replSetName: rs0

پس از آن می‌توان دستورات زیر را وارد کرد:

```
var conf = {
  "_id": "rs0",
  "members": [{
    "_id": 0,
    "host": "127.0.0.1:27017"
  }]
}
rs.initiate(conf)
```

حال می‌توان رویدادهای ثبت شده را در مجموعه‌ی oplog.rs در پایگاه داده‌ی local مشاهده کرد.

جدول زیر، گام‌های جمع‌آوری اطلاعات و شواهد مربوط به رویداد غیرمجاز حذف محتوای جداول را با استفاده از Oplog نشان می‌دهد.

3	Master/slave	8
3	Replica set	9
4	Node	0

پیکربندی مجموعه‌ی کپی‌ها		
<pre>replication: oplogSizeMB: 100 replSetName: rs0</pre>	وارد کردن تنظیمات مقابل در فایل پیکربندی	۱
<pre>var conf = { "_id": "rs0", "members": [{ "_id": 0, "host": "127.0.0.1:27017"}]} rs.initiate(conf)</pre>	تنظیم اعضای مجموعه‌ی کپی‌ها	۲
برخی دستورات سودمند		
<pre>db.adminCommand({ replSetGetStatus : 1 })</pre>	مشاهده‌ی تنظیمات مربوط به مجموعه‌ی کپی‌ها	۱

ممیزی: MongoDB نسخه‌ی Enterprise، دارای قابلیت ممیزی از اعمال مختلف است. ممیزی به مدیران و کاربران این امکان را می‌دهد که فعالیت‌های سیستمی را ردگیری کنند. یک راه‌حل کامل برای ممیزی باید تمامی سرورهای mongod و پردازنده‌های mongos را شامل شود. در خوشه‌های توزیع‌شده، در صورتی که ممیزی روی نمونه‌های mongos فعال باشد باید ممیزی روی تمامی نمونه‌های mongod در خوشه یعنی shard-ها و سرورهای پیکربندی فعال باشد. علاوه بر بررسی فایل پیکربندی mongod.cfg، می‌توان با استفاده از دستور زیر، تنظیمات مربوط به ممیزی را بررسی کرد (شکل ۸).

```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

```
{
  "argv" : [
    "C:\\Program Files\\MongoDB\\Server\\3.4\\bin\\mongod.exe",
    "--config",
    "C:\\Program Files\\MongoDB\\Server\\3.4\\mongod.cfg",
    "--service"
  ],
  "parsed" : {
    "auditLog" : {
      "destination" : "file",
      "format" : "JSON",
      "path" : "c:\\data\\log\\auditlog.json"
    },
    "config" : "C:\\Program Files\\MongoDB\\Server\\3.4\\mongod.cfg",
    "operationProfiling" : {
      "mode" : "all"
    },
    "replication" : {
      "oplogSizeMB" : 100,
      "replSetName" : "rs0"
    },
    "service" : true,
    "setParameter" : {
      "auditAuthorizationSuccess" : "true",
      "logUserIds" : "1"
    },
    "storage" : {
      "dbPath" : "c:\\data\\db"
    },
    "systemLog" : {
      "destination" : "file",
      "path" : "c:\\data\\log\\mongod.log",
      "verbosity" : 1
    }
  },
  "ok" : 1
}
```

شکل ۸: مشاهده‌ی تنظیمات ممیزی

همانطور که در شکل ۸ دیده می‌شود، رویدادهای ممیزی در فایل JSON در مسیر مشخص شده در `auditLog.path` ذخیره می‌شوند. همچنین پارامتر `auditAuthorizationSuccess` به منظور تهیه‌ی ممیزی از اعمال CRUD (Create, Read, Update, Delete) فعال شده است.

اطلاعات تکمیلی: در صورتی که در فایل پیکربندی `mongod.cfg` در قسمت `auditLog` پارامتر `destination` به عنوان محل خروجی رویدادهای ممیزی مشخص شده باشد، ممیزی فعال می‌شود. رویدادهای ممیزی می‌توانند در میز فرمان^۴ `syslog` (این گزینه در ویندوز قابل تنظیم نیست)، فایل JSON یا فایل BSON نوشته شوند.

auditLog:


```
destination: file
format: JSON
path: c:\data\log\auditlog.json
true همچنین به منظور تهیهی ممیزی از اعمال CRUD باید پارامتر auditAuthorizationSuccess با مقدار
تنظیم شود.
setParameter: { auditAuthorizationSuccess: true }
```

جدول زیر، گام‌های جمع‌آوری اطلاعات و شواهد مربوط به رویداد غیرمجاز حذف محتوای جداول را با استفاده از ممیزی نشان می‌دهد.

تنظیم ممیزی در فایل پیکربندی		
destination: file	تعیین ذخیره‌سازی رویدادهای ممیزی در فایل	۱
path: c:\data\log\auditlog.json	تعیین فایل ذخیره‌سازی رویدادهای ممیزی	۲
format: JSON	تعیین فرمت ذخیره‌سازی رویدادهای ممیزی	۳
setParameter: { auditAuthorizationSuccess: true }	تهیهی ممیزی از اعمال CRUD	۴
برخی دستورات سودمند		
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به ممیزی	۱

۳-۳ استخراج و تجزیه و تحلیل اطلاعات

در این بخش نحوه‌ی استخراج اطلاعات برای هر یک از منابع مربوط به شواهد اطلاعات مورد بحث و بررسی قرار می‌گیرد. با استفاده از اطلاعات جمع‌آوری شده، تحلیل‌گر باید داده‌های حذف‌شده را بررسی و در صورت مشاهده‌ی رویداد حذف غیرمجاز، آن را به عنوان جرم تلقی نماید. در ادامه برای هر یک از منابع حاوی شواهد برای رویدادهای سامانه، نحوه استخراج و تجزیه و تحلیل شواهد تشریح می‌گردد. در مثال‌هایی که در ادامه ارائه شده است، نام جدول حذف شده مشخص بوده و بنابراین یافتن اطلاعات مرتبط با آن، کار آسانی است. در دنیای واقعی، بررسی‌های بیشتری برای یافتن جدول حذف شده نیاز است.

استخراج و تحلیل اطلاعات موجود در فایل رویدادنگاری **mongod.log**: می‌توان وارد فایل **mongod.log** در مسیر مشخص شده در فایل پیکربندی **mongod.cfg** شد و اطلاعاتی در مورد دستور اجرا شده حذف و زمان اجرای آن به دست آورد (شکل ۹).

```
2018-06-01T20:53:17.946+0430 D COMMAND [conn1] run command test.$cmd { delete: "test", deletes: [ { q: { book: "b1" }, limit: 0.0 } ], ordered: true }
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Beginning planning...
=====
Options = INDEX_INTERSECTION
Canonical query:
ns=test.testTree: book == "b1"
Sort: {}
Proj: {}
=====
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Index 0 is kp: { _id: 1 } unique name: '_id_' io: { v: 2, key: { _id: 1 }, name: "_id_", ns: "test.test" }
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Predicate over field 'book'
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Rated tree:
book == "b1" || First: notFirst: full path: book
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Planner: outputted 0 indexed solutions.
2018-06-01T20:53:17.947+0430 D QUERY [conn1] Planner: outputting a collscan:
COLLSCAN
---ns = test.test
```

شکل ۹: محتوای فایل رویدادنگاری **mongod.log**

استخراج و تحلیل اطلاعات با استفاده از پروفایلر: می‌توان وارد پایگاه داده‌ی مورد نظر شد و با استفاده از دستور زیر محتوای پروفایلر را استخراج کرد.

```
db.system.profile.find({"op": "remove"}).pretty()
```

خروجی دستور بالا برای یک عمل حذف در ادامه آورده شده است. همانطور که دیده می‌شود اطلاعاتی همچون تعداد اسناد حذف شده بر اثر اجرای پرسمان، زمان حذف داده و کاربر اجراکننده‌ی پرسمان در خروجی مشخص شده‌اند.

```
{
  "op": "remove",
  "ns": "test.test",
  "query": {
    "book": "b1"
  },
  "keysExamined": 0,
  "docsExamined": 3,
  "ndeleted": 1,
  "keysDeleted": 1,
  "numYield": 0,
  ...
  "millis": 2,
  "planSummary": "COLLSCAN",
```

```
...
"ts" : ISODate("2018-06-01T16:23:17.949Z"),
"client" : "127.0.0.1",
"appName" : "MongoDB Shell",
"allUsers" : [
{
"user" : "test_user",
"db" : "test"
}
],
"user" : "test_user@test"
}
```

استخراج و تحلیل اطلاعات پروفایلر

db.system.profile.find({"op": "remove"}).pretty()	محتوای پروفایلر برای عمل حذف	۱
---	------------------------------	---

استخراج و تحلیل اطلاعات با استفاده از **Opllog**: با استفاده از دستور زیر اطلاعاتی در مورد عمل حذف از Opllog قابل استخراج است. در شکل ۱۰ نمونه‌ای از اطلاعات موجود در Opllog برای عمل حذف نشان داده شده است. اطلاعاتی همچون زمان حذف داده و شناسه‌ی یکتای سند حذف شده در اختیار قرار می‌گیرد.

```
db.oplog.rs.find({"op": "d"}).sort({$natural: -1})
```

```
MongoDB Enterprise rs0:PRIMARY> use test
switched to db test
MongoDB Enterprise rs0:PRIMARY> db.test.insert({"name": "b", "age": 40})
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.find({"name": "b"})
{ "_id" : ObjectId("5bac77387e93f5020e8f3e9e"), "name" : "b", "age" : 40 }
MongoDB Enterprise rs0:PRIMARY> db.test.remove({"name": "b"})
WriteResult({ "nRemoved" : 1 })
MongoDB Enterprise rs0:PRIMARY>
MongoDB Enterprise rs0:PRIMARY> use local
switched to db local
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({"op": "d"}).sort({$natural: -1})
{ "ts" : Timestamp(1538029404, 1), "t" : NumberLong(29), "h" : NumberLong("8016336715114222703"), "v" : 2, "op" : "d", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac77387e93f5020e8f3e9e") } }
{ "ts" : Timestamp(1538029166, 1), "t" : NumberLong(29), "h" : NumberLong("-390876406742121819"), "v" : 2, "op" : "d", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac76607e93f5020e8f3e9b") } }
{ "ts" : Timestamp(1538028709, 1), "t" : NumberLong(29), "h" : NumberLong("-5510235655690852101"), "v" : 2, "op" : "d", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac74294b87a1bf1ee07d411") } }
MongoDB Enterprise rs0:PRIMARY>
```

شکل ۱۰: مشاهده‌ی رویدادهای ثبت شده در Opllog

استخراج و تحلیل اطلاعات Olog		
db.oplog.rs.find({"op": "d"}).sort({\$natural: -1})	محتوای Olog برای عمل حذف	۱

استخراج و تحلیل اطلاعات با استفاده از ممیزی: در صورتی که محتوای فایل auditlog.json مشاهده شود، اطلاعاتی همچون زمان اجرای پرسمان حذف و کاربر اجراکننده برای عمل حذف نشان داده می‌شوند.

```
{ "atype" : "authCheck", "ts" : { "$date" : "2018-06-01T20:53:17.946+0430" }, "local" : { "ip" : "127.0.0.1", "port" : 27017 }, "remote" : { "ip" : "127.0.0.1", "port" : 57984 }, "users" : [ { "user" : "test_user", "db" : "test" } ], "roles" : [ { "role" : "dbAdmin", "db" : "test" } ], "param" : { "command" : "delete", "ns" : "test.test", "args" : { "delete" : "test", "deletes" : [ { "q" : { "book" : "b1" }, "limit" : 0 } ], "ordered" : true } }, "result" : 0 }
```

۳-۴ ترمیم

در صورتی که پیش از حذف داده‌ها اقدامات لازم جهت تهیه فایل پشتیبان صورت پذیرفته باشد، با بازیابی فایل می‌توان اطلاعات از دست رفته را مجدداً در اختیار گرفت. در این بخش سعی داریم روش‌هایی به غیر از روش‌های مربوط به تهیه‌ی نسخه‌های پشتیبان را برای بازیابی داده‌های از دست رفته مورد بحث و بررسی قرار دهیم. برای بازیابی اسناد حذف شده از یک مجموعه می‌توان از Olog کمک گرفت. ابتدا در پایگاه داده‌ی test و در مجموعه‌ی test، اسنادی مطابق دستورات زیر ایجاد و برخی از آن‌ها حذف می‌شوند. کلیه‌ی مراحل در شکل ۱۱ نشان داده شده است.

```
for (i=0; i < 100; i++) {db.test.insert({_id: i})}
for (i=100; i < 200; i++) {db.test.insert({_id: i, name: "test_name"})}
db.test.remove({"name": "test_name"})
```

```
MongoDB Enterprise rs0:PRIMARY> for (i=0; i < 100; i++) {db.test.insert({_id: i})}
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.find().count()
100
MongoDB Enterprise rs0:PRIMARY> for (i=100; i < 200; i++) {db.test.insert({_id: i, name: "test_name"})}
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.find().count()
200
MongoDB Enterprise rs0:PRIMARY> db.test.remove({"name": "test_name"})
WriteResult({ "nRemoved" : 100 })
MongoDB Enterprise rs0:PRIMARY> █
```

شکل ۱۱: ایجاد و حذف اسناد در مجموعه

مجموعه‌ی Olog اصطلاحاً capped collection است؛ بدین معنا که اندازه این مجموعه ثابت بوده و در صورت پر شدن حجم آن، رویدادهای قدیمی‌تر را حذف کرده و بر روی آن‌ها می‌نویسد. بنابراین در صورتی که هنوز

بر روی رویدادهای مربوط به دستورات درج اسناد نوشته نشده باشد، می‌توان داده‌های حذف شده را بازیابی کرد. بنابراین با استفاده از دستورات زیر، داده‌های درج شده از Oplog استخراج شده و مجدداً در مجموعه درج می‌شوند. کلیدی مراحل در شکل ۱۲ نشان داده شده است.

```
use local

var deletedDocs = db.oplog.rs.find({op: "i", ns: "test.test", "o.name": "test_name"}, {"o": 1}).toArray();

use test

for (var i = 0; i < deletedDocs.length; i++) {
... db.test.insert({_id: deletedDocs[i].o._id, name: deletedDocs[i].o.name});
... }
```

```
MongoDB Enterprise rs0:PRIMARY> use local
switched to db local
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({op: "i", ns: "test.test", "o.name": "test_name"}).count()
100
MongoDB Enterprise rs0:PRIMARY> var deletedDocs = db.oplog.rs.find({op: "i", ns: "test.test", "o.name": "test_name"}, {"o": 1}).toArray();
MongoDB Enterprise rs0:PRIMARY> deletedDocs.length
100
MongoDB Enterprise rs0:PRIMARY> use test
switched to db test
MongoDB Enterprise rs0:PRIMARY> db.test.count()
100
MongoDB Enterprise rs0:PRIMARY> for (var i = 0; i < deletedDocs.length; i++) {
... db.test.insert({_id: deletedDocs[i].o._id, name: deletedDocs[i].o.name});
... }
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.count()
200
MongoDB Enterprise rs0:PRIMARY> db.test.count({name: "test_name"})
100
```

شکل ۱۲: بازیابی داده‌های حذف شده از مجموعه

همچنین هنگام حذف یک سند، در فایل Oplog شناسه‌ی یکتای سند حذف شده، مشخص است و با استفاده از آن شناسه‌ی یکتا در صورتی که رویداد درج آن هنوز از مجموعه‌ی Oplog حذف نشده باشد، می‌توان سند را مجدداً بازگرداند. در حقیقت با توجه به زمان حذف سند، می‌توان رویداد حذف را از Oplog استخراج کرد که در این رویداد، شناسه‌ی یکتای سند حذف شده وجود دارد. حال با استفاده از شناسه‌ی یکتا می‌توان رویداد درج را در صورتی که هنوز در Oplog باشد، یافت و داده را بازیابی کرد. کلیدی مراحل در شکل ۱۳ نشان داده شده است.

```
MongoDB Enterprise rs0:PRIMARY> use test
switched to db test
MongoDB Enterprise rs0:PRIMARY> db.test.insert({"goal": "test", "try": "recover deleted data"})
WriteResult({ "nInserted": 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.find({"goal": "test"})
{ "_id" : ObjectId("5bac7fb7e93f5020e8f3e9f"), "goal" : "test", "try" : "recover deleted data" }
MongoDB Enterprise rs0:PRIMARY> db.test.remove({"goal": "test"})
WriteResult({ "nRemoved": 1 })
MongoDB Enterprise rs0:PRIMARY> use local
switched to db local
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({"op": "d", "o._id": ObjectId("5bac7fb7e93f5020e8f3e9f")})
{ "ts" : Timestamp(1538031584, 1), "t" : NumberLong(29), "h" : NumberLong("-2818423778825976716"), "v" : 2, "op" : "d", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac7fb7e93f5020e8f3e9f") } }
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({"op": "i", "o._id": ObjectId("5bac7fb7e93f5020e8f3e9f")})
{ "ts" : Timestamp(1538031551, 1), "t" : NumberLong(29), "h" : NumberLong("-5073497418000619574"), "v" : 2, "op" : "i", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac7fb7e93f5020e8f3e9f"), "goal" : "test", "try" : "recover deleted data" } }
MongoDB Enterprise rs0:PRIMARY>
```

شکل ۱۳: بازبازی سند حذف شده از مجموعه

در به روزرسانی داده‌ها اطلاعاتی که در Oplog ذخیره می‌شود، به صورت زیر خواهد بود:

```
{ "ts" : Timestamp(1528197532, 1), "t" : NumberLong(12), "h" :
NumberLong("7242159816216298254"), "v" : 2, "op" : "u", "ns" : "test.test", "
o2" : { "_id" : ObjectId("5b16719804c9f01a07bc5ba7") }, "o" : { "$set" : { "age" : 29 } } }
```

بنابراین برای بازبازی مستندات درج شده تا قبل از به روزرسانی غیرمجاز، باید رویدادهای مربوط به درج و به روزرسانی مستند، استخراج شده و موارد مجاز آن بر روی داده‌های بازبازی شده اعمال گردد. کلیه‌ی مراحل در شکل ۱۴ نشان داده شده است.

```
MongoDB Enterprise rs0:PRIMARY> use test
switched to db test
MongoDB Enterprise rs0:PRIMARY> db.test.insert({"name": "c", "age": 35})
WriteResult({ "nInserted": 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.update({"name": "c"}, {$set: {"age": 34}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
MongoDB Enterprise rs0:PRIMARY> db.test.find({"name": "c"})
{ "_id" : ObjectId("5bac822d7e93f5020e8f3ea0"), "name" : "c", "age" : 34 }
MongoDB Enterprise rs0:PRIMARY> use local
switched to db local
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({"o._id": ObjectId("5bac822d7e93f5020e8f3ea0")})
{ "ts" : Timestamp(1538032173, 1), "t" : NumberLong(29), "h" : NumberLong("-1711148711800495843"), "v" : 2, "op" : "i", "ns" : "test.test", "o" : { "_id" : ObjectId("5bac822d7e93f5020e8f3ea0"), "name" : "c", "age" : 35 } }
MongoDB Enterprise rs0:PRIMARY> db.oplog.rs.find({"o2._id": ObjectId("5bac822d7e93f5020e8f3ea0")})
{ "ts" : Timestamp(1538032219, 1), "t" : NumberLong(29), "h" : NumberLong("-2127704844347029173"), "v" : 2, "op" : "u", "ns" : "test.test", "o2" : { "_id" : ObjectId("5bac822d7e93f5020e8f3ea0") }, "o" : { "$set" : { "age" : 34 } } }
MongoDB Enterprise rs0:PRIMARY>
```

شکل ۱۴: درج و به روزرسانی روی یک سند

۳-۵ ارائه‌ی مستندات

در این گام، اطلاعات کسب شده در طول فرآیند جرم‌شناسی پایگاه داده مستند می‌شود. برای این منظور می‌بایست برای هر یک از رویدادهای درج، حذف و مشاهده جداول در پایگاه داده به هنگام بررسی جرم، جداول زیر را به طور دقیق تکمیل و به مدیریت سازمان برای پیگیری‌های لازم و طرح دعوی ارائه کرد.

جدول ۳: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد درج غیرمجاز در جدول

درج غیرمجاز در جدول			
<input type="checkbox"/>	وقوع جرم	<input type="checkbox"/>	عدم وقوع
<input type="checkbox"/>	وقوع خصمانه	<input type="checkbox"/>	وقوع ناخواسته
<input type="checkbox"/>	شیوه ممیزی	<input type="checkbox"/>	ممیزی

رویدادنگاری در فایل <input type="checkbox"/> mongod.log	رویدادنگاری پروفایلر <input type="checkbox"/>	رویدادنگاری در Opllog <input type="checkbox"/>	منابع رویدادنگاری
فاقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
حذف سطر(های) درج شده <input type="checkbox"/>			امکان ترمیم
			توضیحات

جدول ۴: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد حذف غیرمجاز محتوای جدول

حذف غیرمجاز محتوای جدول			
عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>	وقوع جرم
ممیزی <input type="checkbox"/>			شیوه ممیزی
رویدادنگاری در فایل <input type="checkbox"/> mongod.log	رویدادنگاری پروفایلر <input type="checkbox"/>	رویدادنگاری در Opllog <input type="checkbox"/>	منابع رویدادنگاری
فاقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
استفاده از Opllog <input type="checkbox"/>			امکان ترمیم
			توضیحات

جدول ۵: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد مشاهده غیرمجاز محتوای جدول

مشاهده غیرمجاز محتوای جدول			
عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>	وقوع جرم
ممیزی <input type="checkbox"/>			شیوه ممیزی
رویدادنگاری در فایل <input type="checkbox"/> mongod.log	رویدادنگاری پروفایلر <input type="checkbox"/>		منابع رویدادنگاری
فاقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
فاقد امکان ترمیم است.			امکان ترمیم
			توضیحات

جدول ۶: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد بروزسانی غیرمجاز محتوای جدول

بروزسانی غیرمجاز محتوای جدول			
عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>	وقوع جرم

		میزی <input type="checkbox"/>	شیوه ممیزی
<input type="checkbox"/> Olog در رویدادنگاری	<input type="checkbox"/> رویدادنگاری پروفایلر	رویدادنگاری در فایل <input type="checkbox"/> mongod.log	منابع رویدادنگاری
فاقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
<input type="checkbox"/> استفاده از Olog			امکان ترمیم
			توضیحات

۳-۶ جمع‌بندی

در این فصل به بحث و بررسی رویدادهای غیرمجازی که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال می‌شوند و موجب درج سطر (های) جدید، حذف و بروزرسانی سطر (های) موجود یا مشاهده تمامی یا بخشی از جدول (ها) در پایگاه می‌گردند، پرداختیم. برای این منظور، پس از شناسایی جرم، چهار رویکرد به نام‌های رویدادنگاری در فایل mongod.log، رویدادنگاری پروفایلر، رویدادنگاری در Olog و ممیزی برای جمع‌آوری شواهد و اطلاعات مربوط به آن را معرفی کردیم. برای هر یک از این رویکردها، تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و در پایان تهیه مستندات را تشریح کردیم.

۴ تغییر غیرمجاز شمای پایگاه‌داده

در این فصل به بحث و بررسی رویدادهای غیرمجازی که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال و موجب تغییر در شمای پایگاه‌داده می‌شود، می‌پردازیم. این دسته از رویدادها، از جمله دستورات تعریف داده (DDL) در پایگاه‌داده به حساب می‌آیند. از آنجاییکه رویدادهای مورد بحث در این فصل از نقطه نظر جرم‌شناسی بسیار به هم نزدیک هستند، لذا تنها بر روی رویداد حذف غیرمجاز یک جدول متمرکز می‌شویم.

۴-۱ شناسایی جرم

فرآیند جرم‌شناسی با مشاهده‌ی رویدادهای غیرمجاز در سامانه آغاز می‌شود. بنابراین در صورتیکه مدیر پایگاه‌داده خود به طور مستقیم یا غیرمستقیم (از طریق کاربران)، نسبت به تغییر در شمای پایگاه‌داده همچون

حذف یک جدول آگاهی پیدا کند که انتظار تغییر آن در شرایط فعلی را نداشته است، فرآیند جرم‌شناسی در سامانه آغاز می‌شود.

۴-۲ جمع‌آوری اطلاعات و شواهد

از آنجاییکه هر پایگاه‌داده شواهد مربوط به اعمال مختلف را در بخش‌ها و فایل‌های رویدادنگاری مختلف ذخیره می‌کند، در این بخش قصد داریم منابع مربوط به شواهد رویداد حذف غیرمجاز یک جدول را تعیین نماییم. لازم به ذکر است که تمرکز ما در فرآیند جرم‌شناسی پایگاه‌داده تنها بر روی اطلاعات حاصل از رویدادنگاری و ممیزی در پایگاه‌داده است و شامل اطلاعات موجود در سیستم عامل و حافظه‌ی اصلی نمی‌باشد. از این‌رو، در ادامه به معرفی منابع مربوط به شواهد رویداد حذف غیرمجاز یک جدول و نحوه آگاهی و تنظیم آن‌ها می‌پردازیم.

فایل رویدادنگاری mongod.log: تمامی پرسمان‌ها و اعمال اجرایی در پایگاه‌داده‌ی MongoDB را می‌توان در فایل رویدادنگاری mongod.log ثبت و ذخیره کرد. پس از نصب MongoDB بر روی سیستم عامل ویندوز، می‌توان در مسیر نصب، فایل پیکربندی برای راه‌اندازی سرویس Mongod با نام mongod.cfg ایجاد کرد و تنظیمات لازم را در آن قرار داد. با استفاده از دستور زیر می‌توان محل فایل‌های رویدادنگاری و داده در MongoDB را استخراج کرد.

```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد مربوط به رویداد تغییر غیرمجاز شمای پایگاه‌داده با استفاده از فایل رویدادنگاری mongod.log را نشان می‌دهد.

تنظیم رویدادنگاری در فایل پیکربندی		
destination: file	تعیین ذخیره‌سازی رویدادها در فایل	۱
path: c:\data\log\mongod.log	تعیین فایل ذخیره‌سازی رویدادها	۲
verbosity: 1	تعیین سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۳
برخی دستورات سودمند		
db.getLogComponents()	مشاهده‌ی سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۱
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به رویدادنگاری	۲

پروفایلر^{۴۳}: پروفایلر اطلاعات دقیقی در مورد دستورات اجرا شده بر روی نمونه‌ای mongod را جمع‌آوری می‌کند. این دستورات شامل اعمال CRUD و دستورات مدیریتی و پیکربندی است. پروفایلر تمامی داده‌های جمع‌آوری شده را در مجموعه‌ی system.profile ذخیره می‌کند. قابلیت پروفایلر به صورت پیش‌فرض غیرفعال است. پروفایلر سه سطح دارد:

- **سطح صفر**: پروفایلر را غیرفعال می‌کند.
- **سطح یک**: اعمال کند و زمانبر که اجرای آن‌ها از حد آستانه‌ای بیشتر زمان می‌برد را ثبت می‌کند.
- **سطح دو**: تمامی اعمال را ثبت می‌کند.

با استفاده از دستور زیر می‌توان وضعیت پروفایلر برای یک پایگاه‌داده را مشاهده کرد:

```
db.getProfilingStatus()
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد مربوط به رویداد تغییر غیرمجاز شمای پایگاه‌داده با استفاده از پروفایلر را نشان می‌دهد.

تنظیم سطح پروفایلر		
db.setProfilingLevel(<LEVEL_NUMBER>)	تنظیم سطح پروفایلر با دستور	۱
operationProfiling: mode: <LEVEL_STRING>	تنظیم سطح پروفایلر در فایل پیکربندی	۲
برخی دستورات سودمند		
db.getProfilingStatus()	بررسی وضعیت پروفایلر برای یک پایگاه‌داده	۱

Oplog: Oplog به منظور ثبت اعمال نوشتاری برای استفاده در تکرار کاربرد دارد. Oplog یک مجموعه است که در آن تمامی اعمالی که به نحوی داده‌های ذخیره شده در پایگاه‌داده را تغییر می‌دهند، ثبت می‌شود. MongoDB اعمال و پرسمان‌ها را بر روی سرور اصلی اعمال کرده و بر روی Oplog در سرور اصلی ذخیره می‌کند. اعضای ثانویه، Oplog را کپی و عملیات‌های درون آن را اعمال می‌کنند. تمامی اعضای مجموعه‌ی

⁴ Profiler
⁴ Instance

³
⁴

تکرار گارای یک نسخه از Oplog هستند. به منظور مشاهده‌ی تنظیمات مربوط به مجموعه‌ی کپی‌ها، می‌توان از دستور زیر استفاده کرد.

```
db.adminCommand( { replSetGetStatus : 1 } )
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد مربوط به رویداد تغییر غیرمجاز شمای پایگاه داده با استفاده از oplog را نشان می‌دهد.

پیکربندی مجموعه‌ی کپی‌ها		
<pre>replication: oplogSizeMB: 100 replSetName: rs0</pre>	وارد کردن تنظیمات مقابل در فایل پیکربندی	۱
<pre>var conf = { "_id": "rs0", "members": [{ "_id": 0, "host": "127.0.0.1:27017"}]} rs.initiate(conf)</pre>	تنظیم اعضای مجموعه‌ی کپی‌ها	۲
برخی دستورات سودمند		
<pre>db.adminCommand({ replSetGetStatus : 1 })</pre>	مشاهده‌ی تنظیمات مربوط به مجموعه‌ی کپی‌ها	۱

ممیزی: نسخه‌ی Enterprise از MongoDB دارای قابلیت ممیزی از اعمال مختلف است. ممیزی به مدیران و کاربران این امکان را می‌دهد که فعالیت‌های سیستمی را ردگیری کنند. یک راه‌حل کامل برای ممیزی باید تمامی سرورهای mongod و پردازنده‌های mongos را شامل شود. در خوشه‌های توزیع شده در صورتی که ممیزی روی نمونه‌های mongos فعال باشد، باید ممیزی روی تمامی نمونه‌های mongod در خوشه یعنی shard-ها و سرورهای پیکربندی فعال باشد. علاوه بر بررسی فایل پیکربندی mongod.cfg، می‌توان با استفاده از دستور زیر، تنظیمات مربوط به ممیزی را بررسی کرد.

```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد مربوط به رویداد تغییر غیرمجاز شمای پایگاه‌داده با استفاده از ممیزی را نشان می‌دهد.

تنظیم ممیزی در فایل پیکربندی		
destination: file	تعیین ذخیره‌سازی رویدادهای ممیزی در فایل	۱
path: c:\data\log\auditlog.json	تعیین فایل ذخیره‌سازی رویدادهای ممیزی	۲
format: JSON	تعیین فرمت ذخیره‌سازی رویدادهای ممیزی	۳
setParameter: { auditAuthorizationSuccess: true }	تهیه‌ی ممیزی از اعمال CRUD	۴
برخی دستورات سودمند		
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به ممیزی	۱

۳-۴ استخراج و تجزیه و تحلیل اطلاعات

در این بخش، نحوه‌ی استخراج اطلاعات از منابع مربوط به شواهد مورد بحث و بررسی قرار می‌گیرد. با استفاده از اطلاعات جمع‌آوری شده، تحلیل‌گر باید تغییرات در شمای پایگاه‌داده را بررسی و در صورت مشاهده‌ی رویداد غیرمجاز، آن را به عنوان جرم تلقی نماید. در ادامه برای هر یک از منابع حاوی شواهد برای رویدادهای سامانه، نحوه استخراج و تجزیه و تحلیل تشریح می‌گردد.

استخراج و تحلیل اطلاعات موجود در فایل رویدادنگاری mongod.log: می‌توان وارد فایل mongod.log در مسیر مشخص شده در فایل پیکربندی mongod.cfg شد و اطلاعاتی در مورد دستور اجرا شده و زمان اجرای آن به دست آورد (شکل ۱۵).

```

2018-06-22T20:23:34.085+0430 I COMMAND [conn1] CMD: drop test.test
2018-06-22T20:23:34.085+0430 D STORAGE [conn1] dropCollection: test.test
2018-06-22T20:23:34.085+0430 D INDEX [conn1] dropAllIndexes dropping: { v: 2, key: { _id: 1 }, name: "_id_", ns: "test.test" }
2018-06-22T20:23:34.086+0430 D STORAGE [conn1] test.test: clearing plan cache - collection info cache reset
2018-06-22T20:23:34.086+0430 D STORAGE [conn1] dropIndexes done
2018-06-22T20:23:34.086+0430 D STORAGE [conn1] deleting metadata for test.test @ RecordId(17)
2018-06-22T20:23:34.122+0430 D STORAGE [conn1] WT drop of table:index-1--2523936265598001586 res 0
2018-06-22T20:23:34.122+0430 D STORAGE [conn1] ~WiredTigerRecordStore for: test.test
2018-06-22T20:23:34.241+0430 D STORAGE [conn1] WT drop of table:collection-0--2523936265598001586 res 0
2018-06-22T20:23:34.241+0430 I COMMAND [conn1] command test.test appName: "MongoDB Shell" command: drop { drop: "test" } numYields:0 r
2018-06-22T20:23:34.245+0430 I COMMAND [conn1] command admin.$cmd appName: "MongoDB Shell" command: replSetGetStatus { replSetGetStatu

```

شکل ۱۵: رویداد حذف یک سند در mongod.log

همانطور که در شکل ۱۵ دیده می‌شود، مجموعه‌ی حذف شده به همراه زمان حذف آن در فایل mongod.log نشان داده شده است.

استخراج و تحلیل اطلاعات با استفاده از پروفایلر: می‌توان وارد پایگاه‌داده‌ی مورد نظر شد و با استفاده از دستور زیر محتوای پروفایلر را استخراج کرد.

```
db.system.profile.find({}).pretty()
```

دستور بالا تمامی رویدادهای مربوط به پایگاه‌داده را استخراج می‌کند. به منظور استخراج اطلاعات در مورد حذف یک مجموعه می‌توان دستور بالا را محدودتر کرد.

```
db.system.profile.find({"command": {"drop": "test"}}).pretty()
```

خروجی دستور بالا برای یک عمل حذف مجموعه، در ادامه آورده شده است. همانطور که دیده می‌شود اطلاعاتی همچون زمان حذف مجموعه، مجموعه‌ی حذف شده و کاربر اجراکننده‌ی پرسمان در خروجی مشخص شده‌اند.

```

{
  "op" : "command",
  "ns" : "test.test",
  "command" : {
    "drop" : "test"
  },
  "numYield" : 0,
  ...
  "millis" : 2,
  "ts" : ISODate("2018-06-22T16:04:55.706Z"),
  "client" : "127.0.0.1",
  "appName" : "MongoDB Shell",
  "allUsers" : [

```

```
{
  "user" : "test_user",  "db" : "test"
}
],
"user" : "test_user@test"
}
```

استخراج و تحلیل اطلاعات پروفایلر

محتوای پروفایلر	۱
db.system.profile.find({"command": {"drop": "test"}}).pretty()	برای عمل حذف جدول

استخراج و تحلیل اطلاعات با استفاده از **Oplog**: با استفاده از دستور زیر می‌توان اطلاعاتی در مورد رویدادها به دست آورد. منظور از عملیات "c" کلیدی اعمالی است که در سطح بالا بر روی پایگاه داده تأثیر می‌گذارند.

```
db.oplog.rs.find({"op": "c"}).sort({$natural: -1})
```

در شکل ۱۶ نمونه‌ای از اطلاعات موجود در Oplog برای عمل حذف مجموعه نشان داده شده است. اطلاعاتی همچون زمان حذف مجموعه و مجموعه‌ی حذف شده در اختیار قرار می‌گیرد.

```
{ "ts": Timestamp(1538032866, 1), "t": NumberLong(29), "h": NumberLong("4110408161315415711"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "drop": "test" } }
{ "ts": Timestamp(1538026369, 1), "t": NumberLong(29), "h": NumberLong("6606404383445823730"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "create": "test", "idIndex": {
"v": 2, "key": { "_id": 1 }, "name": "_id_", "ns": "test.test" } } }
{ "ts": Timestamp(1529684130, 1), "t": NumberLong(2), "h": NumberLong("205666964492939174"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "drop": "test" } }
{ "ts": Timestamp(1529684123, 1), "t": NumberLong(2), "h": NumberLong("-4597887325383036674"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "create": "test", "idIndex": {
"v": 2, "key": { "_id": 1 }, "name": "_id_", "ns": "test.test" } } }
{ "ts": Timestamp(1529683495, 1), "t": NumberLong(2), "h": NumberLong("6540867271443540361"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "drop": "test" } }
{ "ts": Timestamp(1529683492, 1), "t": NumberLong(2), "h": NumberLong("4283341467273589467"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "create": "test", "idIndex": {
"v": 2, "key": { "_id": 1 }, "name": "_id_", "ns": "test.test" } } }
{ "ts": Timestamp(1529682814, 1), "t": NumberLong(2), "h": NumberLong("7108854286840645133"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "drop": "test" } }
{ "ts": Timestamp(1529682809, 1), "t": NumberLong(2), "h": NumberLong("-5345811354632198343"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "create": "test", "idIndex": {
"v": 2, "key": { "_id": 1 }, "name": "_id_", "ns": "test.test" } } }
{ "ts": Timestamp(1529682611, 1), "t": NumberLong(1), "h": NumberLong("3952132406321316278"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "drop": "test" } }
{ "ts": Timestamp(1529682598, 1), "t": NumberLong(1), "h": NumberLong("1371724587365640687"), "v": 2, "op": "c", "ns": "test.$cmd", "o": { "create": "test", "idIndex": {
"v": 2, "key": { "_id": 1 }, "name": "_id_", "ns": "test.test" } } }
```

شکل ۱۶: محتوای Oplog برای عمل حذف مجموعه

استخراج و تحلیل اطلاعات Oplog

db.oplog.rs.find({"op": "c"}).sort({"\$natural": -1})	محتوای Oplog برای عمل حذف جدول	۱
---	--------------------------------	---

استخراج و تحلیل اطلاعات با استفاده از ممیزی: در صورتی که محتوای فایل auditlog.json مشاهده شود، اطلاعاتی همچون زمان اجرای پرسمان حذف مجموعه و کاربر اجراکننده نشان داده می‌شوند.

```
{ "atype" : "authCheck", "ts" : { "$date" : "2018-06-22T20:45:30.321+0430" }, "local" : { "ip" : "127.0.0.1", "port" : 27017 }, "remote" : { "ip" : "127.0.0.1", "port" : 55375 }, "users" : [ { "user" : "test_user", "db" : "test" } ], "roles" : [ { "role" : "dbAdmin", "db" : "test" } ], "param" : { "command" : "drop", "ns" : "test.test", "args" : { "drop" : "test" } }, "result" : 0 }
```

۴-۴ ترمیم

در صورتی که پیش از تغییر در شمای پایگاه‌داده و به طور خاص حذف یک جدول، اقدامات لازم جهت تهیه فایل پشتیبان صورت پذیرفته باشد، با بازیابی فایل می‌توان اطلاعات از دست رفته را مجدداً در اختیار گرفت. متأسفانه برای پایگاه‌داده‌ی MongoDB به جز روش پشتیبان‌گیری و بازیابی فایل‌های پشتیبان، روش و ابزار مناسبی یافت نشد. لازم به ذکر است که راه دیگری که همواره برای تمامی پایگاه‌داده می‌توان بهره گرفت، استخراج تمامی رویدادهای ثبت شده از زمان ایجاد تا پیش از زمان حذف و اجرای مجدد دستورات مربوط به این رویدادها می‌باشد که این امر نیز مستلزم آن است که تمامی رویدادها از زمان ایجاد تا پیش از حذف در اختیار باشند.

۴-۵ ارائه‌ی مستندات

در این گام، اطلاعات کسب شده در طول فرآیند جرم‌شناسی پایگاه‌داده مستند می‌شود. برای این منظور می‌بایست به هنگام بررسی جرم، جدول زیر به طور دقیق تکمیل و به مدیریت سازمان برای پیگیری‌های لازم و طرح دعوی ارائه گردد.

جدول ۷: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد تغییر غیرمجاز شمای پایگاه‌داده

تغییر غیرمجاز شمای پایگاه‌داده			
وقوع جرم	عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>
شیوه ممیزی	ممیزی <input type="checkbox"/>		
منابع رویدادنگاری	رویدادنگاری در فایل <input type="checkbox"/>	رویدادنگاری پروفایلر <input type="checkbox"/>	رویدادنگاری در Oplog <input type="checkbox"/>
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		

امکان ترمیم	استفاده از پشتیبان‌گیری و بازگرداندن فایل‌های پشتیبان <input type="checkbox"/>
توضیحات	

۴-۶ جمع‌بندی

در این فصل به بحث و بررسی رویدادهای غیرمجازی که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی بر روی پایگاه‌داده اعمال و موجب تغییر در شمای پایگاه‌داده می‌گردند، پرداخته شد. برای این منظور، پس از شناسایی جرم، چهار رویکرد به نام‌های رویدادنگاری در فایل mongod.log، رویدادنگاری پروفایلر، رویدادنگاری در Oplog و ممیزی برای جمع‌آوری شواهد و اطلاعات مربوط به جرم معرفی گردید. سپس برای هر یک از این رویکردها، تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و در پایان تهیه مستندات تشریح شد.

۵ تلاش برای ورود غیرمجاز به پایگاه‌داده

در این فصل، رویدادهای غیرمجازی که به صورت خصمانه یا ناخواسته توسط کاربر/برنامه‌کاربردی برای ورود به پایگاه‌داده صورت می‌پذیرد را مورد بحث و بررسی قرار می‌دهیم. این رویداد در واقع به هنگام تلاش برای ورود از طریق بدست آوردن نام کاربری و کلمه عبور به سمپاد تحمیل می‌شود. در ادامه، فرآیند جرم‌شناسی مربوط به این رویداد مورد بحث و بررسی قرار می‌گیرد.

۵-۱ شناسایی جرم

در صورتی که تلاش‌های ناموفق برای ورود به سیستم پایگاه‌داده ثبت شوند، با مشاهده‌ی رویدادهای ثبت‌شده می‌توان حمله به پایگاه‌داده برای یافتن نام کاربری و کلمه‌ی عبور را تشخیص داد. در این شرایط فرآیند جرم‌شناسی در سامانه آغاز می‌شود.

۵-۲ جمع‌آوری اطلاعات و شواهد

از آنجاییکه هر پایگاه‌داده شواهد مربوط به اعمال مختلف را در بخش‌ها و فایل‌های رویدادنگاری مختلف ذخیره می‌کند، در این بخش قصد داریم منابع مربوط به شواهد رویداد تلاش برای ورود به پایگاه‌داده را تعیین نماییم.

استفاده از فایل رویدادنگاری **mongod.log**: تمامی پرسمان‌ها و اعمال اجرایی در پایگاه‌داده‌ی MongoDB را می‌توان در فایل رویدادنگاری **mongod.log** ثبت و ذخیره کرد. پس از نصب MongoDB بر روی سیستم عامل ویندوز، می‌توان در مسیر نصب، فایل پیکربندی برای راه‌اندازی سرویس **Mongod** با نام **mongod.cfg** ایجاد کرد و تنظیمات لازم را در آن قرار داد. با استفاده از دستور زیر می‌توان محل فایل‌های رویدادنگاری و داده در MongoDB را استخراج کرد.


```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد را برای رویداد ورود غیرمجاز به پایگاه داده با استفاده از فایل رویدادننگاری mongod.log نشان می‌دهد.

تنظیم رویدادننگاری در فایل پیکربندی		
destination: file	تعیین ذخیره‌سازی رویدادها در فایل	۱
path: c:\data\log\mongod.log	تعیین فایل ذخیره‌سازی رویدادها	۲
verbosity: 1	تعیین سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۳
برخی دستورات سودمند		
db.getLogComponents()	مشاهده‌ی سطح جزئیات پیغام‌های تمامی مؤلفه‌ها	۱
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به رویدادننگاری	۲

استفاده از ممیزی: MongoDB نسخه‌ی Enterprise، دارای قابلیت ممیزی از اعمال مختلف است. ممیزی به مدیران و کاربران این امکان را می‌دهد که فعالیت‌های سیستمی را ردگیری کنند. یک راه‌حل کامل برای ممیزی باید تمامی سرورهای mongod و پردازنده‌های mongos را شامل شود. در خوشه‌های توزیع‌شده، در صورتی که ممیزی روی نمونه‌های mongos فعال باشد، باید ممیزی روی تمامی نمونه‌های mongod در خوشه یعنی shard-ها و سرورهای پیکربندی فعال باشد. علاوه بر بررسی فایل پیکربندی mongod.cfg، می‌توان با استفاده از دستور زیر، تنظیمات مربوط به ممیزی را بررسی کرد.

```
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})
```

جدول زیر، گام‌های مربوط به جمع‌آوری اطلاعات و شواهد مربوط به رویداد ورود غیرمجاز به پایگاه داده با استفاده از ممیزی را نشان می‌دهد.

تنظیم ممیزی در فایل پیکربندی

destination: file	تعیین ذخیره‌سازی رویدادهای ممیزی در فایل	۱
path: c:\data\log\auditlog.json	تعیین فایل ذخیره‌سازی رویدادهای ممیزی	۲
format: JSON	تعیین فرمت ذخیره‌سازی رویدادهای ممیزی	۳
setParameter: { auditAuthorizationSuccess: true }	تهیه‌ی ممیزی از اعمال CRUD	۴
برخی دستورات سودمند		
db.getSiblingDB("admin").runCommand({getCmdLineOpts:1})	مشاهده پیکربندی مربوط به ممیزی	۱

۳-۵ استخراج و تجزیه و تحلیل اطلاعات

در این بخش نحوه‌ی استخراج اطلاعات برای هر یک از منابع ذکر شده، بیان می‌شود. با استفاده از اطلاعات جمع‌آوری شده، تحلیل‌گر باید تلاش برای ورود به پایگاه‌داده را بررسی نماید.

استخراج و تحلیل اطلاعات با استفاده از فایل رویدادنگاری mongod.log: نمونه‌ای از فایل ثبت رویدادها در شکل ۱۷ نشان داده شده است.

```

2018-09-27T11:00:44.165+0330 I ACCESS [conn10] SCRAM-SHA-1 authentication failed for test user on test from client 127.0.0.1:52902 ; AuthenticationFailed:
2018-09-27T11:00:44.166+0330 I COMMAND [conn10] command test.$cmd appName: "MongoDB Shell" command: saslContinue { saslContinue: 1, payload: BinData(0, 633D
2018-09-27T11:00:44.169+0330 I - [conn10] end connection 127.0.0.1:52902 (2 connections now open)
2018-09-27T11:00:46.421+0330 D REPL [NoopWriter] Writing noop to oplog as there has been no writes to this replica set in over 10s
2018-09-27T11:00:46.421+0330 D REPL [NoopWriter] Set last known op time to { ts: Timestamp 1538033446000|1, t: 29 }
2018-09-27T11:00:47.787+0330 I NETWORK [thread1] connection accepted from 127.0.0.1:52905 #11 (2 connections now open)
2018-09-27T11:00:47.798+0330 I NETWORK [conn11] received client metadata from 127.0.0.1:52905 conn11: { application: { name: "MongoDB Shell" }, driver: { na
2018-09-27T11:00:47.799+0330 I COMMAND [conn11] command admin.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1, client: { application: { name:
2018-09-27T11:00:47.800+0330 I COMMAND [conn11] command admin.$cmd appName: "MongoDB Shell" command: whatsmyuri { whatsmyuri: 1 } numYields:0 reslen:47 lock
2018-09-27T11:00:47.801+0330 I COMMAND [conn11] command admin.$cmd appName: "MongoDB Shell" command: buildInfo { buildInfo: 1.0 } numYields:0 reslen:1145 lo
2018-09-27T11:00:47.802+0330 I COMMAND [conn11] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0 } numYields:0 reslen:406 locks:
2018-09-27T11:00:47.823+0330 I ACCESS [conn11] SCRAM-SHA-1 authentication failed for test user on test from client 127.0.0.1:52905 ; AuthenticationFailed:
2018-09-27T11:00:47.823+0330 I COMMAND [conn11] command test.$cmd appName: "MongoDB Shell" command: saslContinue { saslContinue: 1, payload: BinData(0, 633D
2018-09-27T11:00:47.825+0330 I - [conn11] end connection 127.0.0.1:52905 (2 connections now open)
2018-09-27T11:00:50.071+0330 I NETWORK [thread1] connection accepted from 127.0.0.1:52908 #12 (2 connections now open)
2018-09-27T11:00:50.082+0330 I NETWORK [conn12] received client metadata from 127.0.0.1:52908 conn12: { application: { name: "MongoDB Shell" }, driver: { na
2018-09-27T11:00:50.082+0330 I COMMAND [conn12] command admin.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1, client: { application: { name:
2018-09-27T11:00:50.083+0330 I COMMAND [conn12] command admin.$cmd appName: "MongoDB Shell" command: whatsmyuri { whatsmyuri: 1 } numYields:0 reslen:47 lock
2018-09-27T11:00:50.084+0330 I COMMAND [conn12] command admin.$cmd appName: "MongoDB Shell" command: buildInfo { buildInfo: 1.0 } numYields:0 reslen:1145 lo
2018-09-27T11:00:50.085+0330 I COMMAND [conn12] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0 } numYields:0 reslen:406 locks:
2018-09-27T11:00:50.085+0330 I COMMAND [conn12] command test.$cmd appName: "MongoDB Shell" command: saslStart { saslStart: 1, mechanism: "SCRAM-SHA-1", payl
2018-09-27T11:00:50.102+0330 I ACCESS [conn12] SCRAM-SHA-1 authentication failed for test user on test from client 127.0.0.1:52908 ; AuthenticationFailed:
2018-09-27T11:00:50.102+0330 I COMMAND [conn12] command test.$cmd appName: "MongoDB Shell" command: saslContinue { saslContinue: 1, payload: BinData(0, 633D

```

شکل ۱۷: رویداد تلاش برای ورود به پایگاه‌داده در mongod.log

هر یک از رویدادهای ثبت شده مربوط به تلاش برای ورود به پایگاه‌داده، مشابه رویداد زیر است. همانطور که دیده می‌شود، زمان تلاش و نام کاربری مشخص شده‌اند.

2018-06-22T21:25:02.256+0430 I ACCESS [conn2] SCRAM-SHA-1 authentication failed for test_user on test from client 127.0.0.1:56714 ; AuthenticationFailed: SCRAM-SHA-1 authentication failed, storedKey mismatch

همانطور که در شکل ۱۷ دیده می‌شود، در فواصل زمانی کوتاه، نام کاربری برای ورود تلاش کرده است و با شکست روبرو شده است. این رویدادهای ثبت شده نشان‌دهنده‌ی حمله‌ی brute force برای یافتن کلمه‌ی عبور مربوط به یک نام کاربری است.

استخراج و تحلیل اطلاعات با استفاده از ممیزی: نمونه‌ای از فایل ممیزی در شکل ۱۸ نشان داده شده است.

```
[], "param" : { "user" : "test_user", "db" : "test", "mechanism" : "SCRAM-SHA-1" }, "result" : 18 }
"param" : { "command" : "isMaster", "ns" : "admin", "args" : { "isMaster" : 1, "client" : { "application" : { "nar
"param" : { "command" : "whatsmyuri", "ns" : "admin", "args" : { "whatsmyuri" : 1 } }, "result" : 0 }
"param" : { "command" : "buildinfo", "ns" : "admin", "args" : { "buildinfo" : 1 } }, "result" : 0 }
"param" : { "command" : "isMaster", "ns" : "test", "args" : { "isMaster" : 1 } }, "result" : 0 }
"param" : { "command" : "saslStart", "ns" : "test", "args" : { "saslStart" : 1, "mechanism" : "SCRAM-SHA-1", "payl
"param" : { "command" : "saslContinue", "ns" : "test", "args" : { "saslContinue" : 1, "payload" : { "$binary" : "\
[], "param" : { "user" : "test_user", "db" : "test", "mechanism" : "SCRAM-SHA-1" }, "result" : 18 }
"param" : { "command" : "isMaster", "ns" : "admin", "args" : { "isMaster" : 1, "client" : { "application" : { "nar
"param" : { "command" : "whatsmyuri", "ns" : "admin", "args" : { "whatsmyuri" : 1 } }, "result" : 0 }
"param" : { "command" : "buildinfo", "ns" : "admin", "args" : { "buildinfo" : 1 } }, "result" : 0 }
"param" : { "command" : "isMaster", "ns" : "test", "args" : { "isMaster" : 1 } }, "result" : 0 }
"param" : { "command" : "saslStart", "ns" : "test", "args" : { "saslStart" : 1, "mechanism" : "SCRAM-SHA-1", "payl
"param" : { "command" : "saslContinue", "ns" : "test", "args" : { "saslContinue" : 1, "payload" : { "$binary" : "\
[], "param" : { "user" : "test_user", "db" : "test", "mechanism" : "SCRAM-SHA-1" }, "result" : 18 }
```

شکل ۱۸: رویداد تلاش برای ورود به پایگاه داده در فایل ممیزی

هر یک از رویدادهای مربوط به تلاش برای ورود به پایگاه داده، مشابه رویداد زیر است. منظور از result: 18 شکست در تأیید اصالت است.

```
{ "atype" : "authenticate", "ts" : { "$date" : "2018-06-22T21:25:05.275+0430" }, "local" : { "ip"
: "127.0.0.1", "port" : 27017 }, "remote" : { "ip" : "127.0.0.1", "port" : 56716 }, "users" : [],
"roles" : [], "param" : { "user" : "test_user", "db" : "test", "mechanism" : "SCRAM-SHA-1" },
"result" : 18 }
```

همانطور که در شکل ۱۸ دیده می‌شود، در فواصل زمانی کوتاه، برای ورود با نام کاربری test_user تلاش صورت پذیرفته و با شکست روبرو شده است. این رویدادهای ثبت شده نشان‌دهنده‌ی حمله‌ی brute force برای یافتن کلمه‌ی عبور مربوط به یک نام کاربری است.

۴-۵ ترمیم

بررسی‌های انجام شده تا زمان نگارش سند نشان می‌دهد پایگاه‌داده MongoDB، پارامتر یا دستوری برای قفل‌یابی غیرفعال کردن دستی یا خودکار کاربر پس از تعداد مشخصی تلاش ناموفق برای ورود به پایگاه‌داده ندارد و تنها می‌توان حساب کاربری را ایجاد یا حذف کرد.

۵-۵ ارائه‌ی مستندات

برای این منظور می‌بایست به هنگام بررسی جرم، جدول زیر به طور دقیق تکمیل و به مدیریت سازمان برای پیگیری‌های لازم و طرح دعوی ارائه گردد.

جدول ۸: تهیه‌ی مستند از فرآیند جرم‌شناسی برای رویداد تلاش برای یافتن نام کاربری یا کلمه‌ی عبور

تلاش برای یافتن نام کاربری یا کلمه‌ی عبور			
وقوع جرم	<input type="checkbox"/> عدم وقوع	<input type="checkbox"/> وقوع خصمانه	<input type="checkbox"/> وقوع ناخواسته
شیوه ممیزی	<input type="checkbox"/> ممیزی		
منابع رویدادنگاری	<input type="checkbox"/> رویدادنگاری در فایل mongod.log		
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	<input type="checkbox"/> حذف حساب کاربری		
توضیحات			

۶-۵ جمع‌بندی

در این فصل، تلاش برای ورود به پایگاه‌داده در صورت نداشتن نام کاربری یا کلمه‌ی عبور مورد بحث و بررسی قرار گرفت. برای این منظور، پس از شناسایی جرم، دو رویکرد به نام‌های رویدادنگاری در فایل mongod.log و ممیزی برای جمع‌آوری شواهد و اطلاعات مربوط به جرم معرفی گردید. در پایان نیز برای رویکرد معرفی شده، تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و تهیه مستندات تشریح گردید.

۶ خلاصه مطالب

جرم‌شناسی پایگاه داده فرآیندی است که طی آن تلاش می‌شود تا اطلاعاتی چون زمان / چگونگی / چرایی و فرد مجرم برای یک رخداد غیرمجاز در سامانه مشخص شود. جرم‌شناسی پایگاه داده زمانی رخ می‌دهد که از مأمور ممیزی، کشف چگونگی وقوع نقض امنیتی و شخص مجرم درخواست شود. جرم‌شناسی پایگاه داده، چالش‌ها و مسائل زیادی به همراه دارد که آن را تبدیل به یک موضوع پیچیده کرده است.

با استفاده از مدل‌های فرآیند جرم‌شناسی پایگاه داده می‌توان به صورت ساختاریافته عملیات مربوط به جرم‌شناسی پایگاه داده را پیش برد. در مراحل مختلف فرآیند جرم‌شناسی، تمرکز اصلی بر روی پایگاه داده و اطلاعات موجود در آن برای شناسایی جرم است. در حقیقت تنها از اطلاعات ثبت شده در پایگاه‌های داده به منظور شناسایی جرم استفاده می‌شود و استفاده از اطلاعات ثبت شده بر روی سیستم عامل، داده‌های موجود در حافظه و شبکه خارج از حوزه مورد بحث است. در یک دسته‌بندی کلی، جرم‌شناسی پایگاه داده شامل گام‌های زیر است:

۱. شناسایی جرم،
۲. جمع‌آوری اطلاعات و شواهد،
۳. تجزیه و تحلیل،
۴. ترمیم،
۵. ارائه‌ی مستندات.

هر سامانه پایگاه داده، شواهد مربوط به رویدادهای مختلف را در فایل‌های مختلف برای استفاده در تجزیه و تحلیل جرم‌شناسی ذخیره می‌کند. این به معنای آن است که برای تجزیه و تحلیل جرم‌شناسی باید نسبت به چگونگی عملکرد پایگاه داده و محل فایل‌ها و مصنوعات مختلف اطلاع داشت. لازم به ذکر است که جمع‌آوری مصنوعات و شواهد می‌تواند سبب تغییر در پایگاه داده شود، بنابراین پیش از استخراج اطلاعات از پایگاه داده یا خارج از پایگاه داده باید نسبت به این موضوع و پایدار یا ناپایدار بودن اطلاعات آگاهی پیدا کرد.

لازم به ذکر است، در صورتی که مهاجم مجموعه‌ای را حذف کند و برای پاک کردن ردپای خود اطلاعات موجود در فایل‌های رویدادنگاری و ممیزی را دستی تغییر دهد، به عنوان نمونه، کاربر ثبت شده به عنوان کاربر اجراکننده‌ی پرسمان را تغییر دهد، مطلوب است که بتوان تغییر غیرمجاز در فایل‌های رویدادنگاری و ممیزی را متوجه شد. در MongoDB اگرچه هنگامی که سرور در حال اجرا است، نمی‌توان فایل رویدادنگاری جاری یعنی mongod.log و فایل ممیزی auditlog.json را تغییر داد ولی هنگامی که سرور متوقف شود، امکان تغییر فایل auditlog.json وجود دارد. همچنین می‌توان فایل‌های رویدادنگاری مربوط به زمان‌های قبل را تغییر داد بدون اینکه این تغییر مشخص شود. بنابراین برای جلوگیری از دسترسی‌های غیرمجاز به این فایل‌ها می‌بایست بر روی آن‌ها کنترل دسترسی و مجوزهای لازم اعمال شود.

برای هر یک از رویدادهای غیرمجاز در سامانه پایگاه‌داده به هنگام بررسی جرم، جداولی برای ارائه مستندات لازم در نظر گرفته شده است که می‌بایست در طول فرآیند جرم‌شناسی به طور دقیق تکمیل و به مدیریت سازمان برای پیگیری‌های لازم و طرح دعوی ارائه گردد. جدول (۲)، کلیه جداول اطلاعاتی مربوطه را به تصویر کشیده است.

جدول ۹: تهیه‌ی مستند از فرآیند جرم‌شناسی در پایگاه‌داده‌ی MongoDB

درج غیرمجاز در جدول			
وقوع جرم	عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>
شیوه ممیزی	ممیزی <input type="checkbox"/>		
منابع رویدادنگاری	رویدادنگاری در فایل <input type="checkbox"/>	رویدادنگاری پروفایلر <input type="checkbox"/>	رویدادنگاری در Opllog <input type="checkbox"/>
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	حذف سطر(های) درج شده <input type="checkbox"/>		
توضیحات			
حذف غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>
شیوه ممیزی	ممیزی <input type="checkbox"/>		
منابع رویدادنگاری	رویدادنگاری در فایل <input type="checkbox"/>	رویدادنگاری پروفایلر <input type="checkbox"/>	رویدادنگاری در Opllog <input type="checkbox"/>
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	استفاده از Opllog <input type="checkbox"/>		
توضیحات			
مشاهده غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع <input type="checkbox"/>	وقوع خصمانه <input type="checkbox"/>	وقوع ناخواسته <input type="checkbox"/>
شیوه ممیزی	ممیزی <input type="checkbox"/>		
منابع رویدادنگاری	رویدادنگاری در فایل <input type="checkbox"/>	رویدادنگاری پروفایلر <input type="checkbox"/>	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		

فقد امکان ترمیم است.			امکان ترمیم
			توضیحات
تغییر غیرمجاز شمای پایگاه‌داده			
<input type="checkbox"/> وقوع ناخواسته	<input type="checkbox"/> وقوع خصمانه	<input type="checkbox"/> عدم وقوع	وقوع جرم
			شیوه ممیزی <input type="checkbox"/> ممیزی
<input type="checkbox"/> Oplog در رویدادنگاری	<input type="checkbox"/> رویدادنگاری پروفایلر	رویدادنگاری در فایل <input type="checkbox"/> mongod.log	منابع رویدادنگاری
فقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
<input type="checkbox"/> استفاده از پشتیبان‌گیری و بازگرداندن فایل‌های پشتیبان			امکان ترمیم
			توضیحات
تلاش برای یافتن نام کاربری یا کلمه عبور			
<input type="checkbox"/> وقوع ناخواسته	<input type="checkbox"/> وقوع خصمانه	<input type="checkbox"/> عدم وقوع	وقوع جرم
			شیوه ممیزی <input type="checkbox"/> ممیزی
<input type="checkbox"/> mongod.log در رویدادنگاری			منابع رویدادنگاری
فقد شیوه یا ابزار تحلیل است.			شیوه یا ابزار تحلیل
<input type="checkbox"/> حذف حساب کاربری			امکان ترمیم
			توضیحات

۷ منابع

- [1]. DB audit and security 360, version 5.0, SoftTree Technologies, Inc.
- [2]. <http://www.dba-oracle.com>
- [3]. Al-Dhaqm, A., Razak, S.A., Othman, S.H., Nagdi, A. and Ali, A., 2016. A GENERIC DATABASE FORENSIC INVESTIGATION PROCESS MODEL. Jurnal Teknologi, 78.
- [4]. R. Ramakrishnan and J. Gehrke. Database Management Systems (Third Edition). McGraw-Hill, Inc. New York, NY, USA, 2003.
- [5]. G. Palmer, A Road Map for Digital Forensic Research, DFRWS Technical Report, DTR-T001-01 Final, Air Force Research Laboratory, Rome, New York, 2001.
- [6]. <https://docs.oracle.com>
- [7]. https://en.wikipedia.org/wiki/Transaction_log

- [8]. J. Shital, Forensic Investigation for Database Tampering using Audit Logs, International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 03, March 2015
- [9]. K. Fowler, SQL Server database forensics, presented at the Black Hat USA Conference, 2007.
- [10]. Fasan, O.M. and Olivier, M.S., 2012. On Dimensions of Reconstruction in Database Forensics. In WDFIA (pp. 97-106).
- [11]. Al-Dhaqm, A., Razak, S.A., Othman, S.H., Nagdi, A. and Ali, A., 2016. A GENERIC DATABASE FORENSIC INVESTIGATION PROCESS MODEL. Jurnal Teknologi, 78.
- [12]. <https://solutioncenter.apexsql.com/recover-sql-server-database-using-only-a-transaction-log-file-ldf-and-old-backup-files/>
- [13]. H. Q. Beyers, "Database forensics: Investigating compromised database management systems", 2013.
- [14]. Khanuja, H.K., Adane, D.S.: A Framework For Database Forensic Analysis. Published in Computer Science & Engineering: An International Journal (CSEIJ) 2(3) (2012)
- [15]. Finnigan, P., *Oracle Incident Response and Forensics: Preparing for and Responding to Data Breaches*, 2018, Apress, Berkeley, CA.
- [16]. <https://dbatricksworld.com/ora-38707-media-recovery-is-not-enabled/>
- [17]. <http://www.innovateus.net/science/what-forensics>
- [18]. R. Urbano, 2017, Oracle Database Administrator's Guide, 12c Release 2 (12.2)
- [19]. DB audit and security 360, version 5.0, SoftTree Technologies, Inc.
- [20]. <https://docs.mongodb.com/>
- [21]. <https://veliovgroup.com/article/2qsjtNf8NSB9XxZDh/mongodb-replica-set-with-oplog>
- [22]. <https://stackoverflow.com/questions/25802786/is-there-any-way-to-recover-recently-deleted-documents-in-mongodb>.