بسمه تعالی

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

جرم‌شناسی در پایگاه‌داده MySQL

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 برخوارد باشد. مکانیزم‌های امنیتی موجود به سبب حفظ کارایی سامانه و دلایلی از این دست نمی‌توانند جلوی تمامی تهدیدها و جرایم اینترنتی در پایگاه‌های داده را بگیرند و لذا وقوع جرم در سامانه امکان‌پذیر است. از این‌رو، به منظور شناسایی شواهد جرم، جمع‌آوری اطلاعات مربوطه، تجزیه‌وتحلیل 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 گسترده و پیچیده معرفی گردد.

در این راستا بر آن شدیم تا جرم‌شناسی در پایگاه‌داده MySQL 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 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، ویرایش، مشاهده غیرمجاز محتوای جداول، تغییر غیرمجاز شمای پایگاه‌داده و تلاش برای ورود غیرمجاز به سامانه پایگاه‌داده مطالعه می‌شود. 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 و همچنین کلیه پیکربندی‌های ارائه شده در مستند حاضر بر روی MySQL 5.7.21 نسخه‌ی Enterprise می‌باشد.

فهرست مطالب

۱ جرم شناسی پایگاه‌داده

بسیاری از سازمان‌ها و آژانس‌های دولتی از پایگاه‌داده برای ذخیره و بازیابی اطلاعات استفاده می‌کنند. از آنجاییکه داده‌های سازمان‌ها می‌توانند حاوی اطلاعات حساس و حیاتی باشند، لذا حفاظت از پایگاه‌داده به عنوان یک سامانه ذخیره‌سازی اطلاعات، یک امر حیاتی و مهم به حساب می‌آید. داده‌های ذخیره شده در پایگاه‌داده 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 نهایتا ارائه‌ی شرح مفصلی از فعالیت‌های مجرمانه‌ی سایبری ایفا می‌کند. جرم‌شناسی پایگاه‌داده، شاخه‌ای از جرم‌شناسی دیجیتال است که در آن پایگاه‌داده و فراداده‌های مرتبط با آن به صورت قانونی مورد مطالعه قرار می‌گیرند. یکی از مهم‌ترین اهدفی که در جرم‌شناسی پایگاه‌داده دنبال می‌شود آن است که تشخیص دهیم چه کسی، چه زمانی، چه داده‌ای را تغییر داده است. لازم به ذکر است که پایگاه‌داده 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 پایگاه‌داده، مورد بحث و بررسی قرار می‌گیرد.

۱-۱ تعاریف و مفاهیم

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

سامانه مدیریت پایگاه‌داده: 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[1] 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): تجزیه‌وتحلیل، (۶): تفسیر، (۷): مستندسازی و ارائه‌ی شواهد دیجیتال به منظور تسهیل در بازسازی رویدادهای شناخته شده جنایی یا پیش‌بینی اقدامات غیرمجاز استفاده می‌شود، جرم شناسی دیجیتال اطلاق می‌گردد [۵].

جرم‌شناسی پایگاه‌داده: جرم‌شناسی پایگاه‌داده 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 پایگاه‌داده، 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 پایگاه‌داده، ممیزی اطلاق می‌شود [۶].

رویدادنگاری:[۴] به تاریخچه‌ای از فعالیت‌های اجرا شده توسط سامانه مدیریت پایگاه‌داده اطلاق می‌شود که برای تضمین ویژگی‌های جامعیت (ACID[۵]) به هنگام خرابی سخت افزاری[۶] یا از کار افتادگی ناگهانی[۷] مورد استفاده قرار می‌گیرد [۷].

مصنوعات[۸] پایگاه‌داده: رکوردها یا اطلاعاتی هستند که از پایگاه‌داده 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.

۲-۱ چالش‌ها

جرم‌شناسی پایگاه‌داده، چالش‌های زیادی به همراه دارد که آن را تبدیل به یک موضوع پیچیده می‌کند. سامانه‌های پایگاه‌داده، سرویس‌ها و زیرساخت‌های مختلفی دارند که از یک پایگاه‌داده به پایگاه‌داده 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 پایگاه‌داده

---

[2] Metadata
[3] Auditing
[4] Logging
[5] Atomicity-Consistency-Isolation-Durability
[6] Hardware failure
[7] Crash
[8] Artifact

دارای 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 ازجمله 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[9] 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.

۳-۱ اهداف

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

ردگیری[10] اعمال DML و DDL: 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

اشیای پایگاه‌داده 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. گاهاً 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 بازیابی داده‌های حذف شده خواهد بود.

اثبات یا رد وقوع نقض امنیتی: یک چالش مهم در حوزه جرم‌شناسی پایگاه‌داده، 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 پایگاه‌داده می‌توان ثابت کرد که نقض امنیتی رخ داده است یا خیر.

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

---

[1] Trace [0]

۱-۴ گام‌های اجرایی

گام‌هایی که برای جرم‌شناسی پایگاه‌داده باید برداشته شوند به موقعیت و نوع سامانه مدیریت پایگاه‌داده‌ی 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 اهدف این مرحله، شناسایی مفاهیم مرتبط با بررسی جرم‌شناسی همچون منابع پایگاه‌داده، منابع سیستم‌عامل، منابع شبکه، تیم‌های بررسی، روش‌های بررسی، محیط بررسی، خط‌مشی‌ها، قوانین و مجوزها هستند.

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

- **جمع‌آوری داده‌ها به صورت زنده**[1]: 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**[2]: روش جمع‌آوری داده به صورت غیرزنده، شامل نسخه‌برداری داده‌ها از سیستم مورد بررسی است.
- **جمع‌آوری داده‌ها به صورت ترکیبی**[3]: 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 دیجیتالی حفظ و نگهداری می‌شوند و داده‌ها به صورت ناخواسته تغییر نمی‌کنند یا از بین نمی‌روند.

۳. **جمع‌آوری مصنوعات ناپایدار[4] و پایدار:** مصنوعات و اطلاعات مختلف را می‌توان از پایگاه‌داده، سیستم‌عامل، سرورهای وب یا فایل‌های رویدادنگاری استخراج کرد. لازم به ذکر است که جمع‌آوری مصنوعات و شواهد در سامانه می‌تواند سبب تغییر در پایگاه‌داده شود. از این‌رو، پیش از استخراج اطلاعات از پایگاه‌داده باید نسبت به این موضوع و پایدار یا ناپایدار بودن اطلاعات، آگاهی پیدا کرد.

---

1 Live acquisition
1 Dead acquisition
1 Hybrid acquisition
1 Volatile artifacts

1
2
3
4

هر پایگاه‌داده شواهد مربوط به اعمال مختلف را در فایل‌های رویدادنگاری مختلفی پایگاه‌داده ذخیره می‌کند. این بدین معناست که برای تجزیه و تحلیل جرم‌شناسی می‌بایست نسبت به چگونگی عملکرد پایگاه‌داده، محل فایل‌ها و مصنوعات مختلف اطلاع داشت. مصنوعات در سطح پایگاه‌داده 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 پایگاه‌داده درحافظه‌های ناپایدار ذخیره می‌شوند، داده‌های فرار اطلاق می‌شود. به عنوان نمونه، پایگاه‌داده اوراکل اطلاعات زیادی را در SGA[۱۵] به منظور افزایش کارایی ذخیره می‌کند. لازم به ذکر است که به طور معمول این دسته از داده‌ها دائما 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[۶] و احراز اصالت[۷] 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 تغییریافته 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

[۱۵] قسمتی از حافظه‌ی سیستم که میان تمامی پردازه‌های مربوط به یک نمونه‌ی واحد از پایگاه‌داده‌ی اوراکل مشترک است.

[1] Preservation
[1] Authentication
[6]
[7]

حافظه پنهان[۸]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 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.

---

[1] Cache [8]

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[۹] 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[۱۰] 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

---

[1] Checklist
[2] Trace files
[9]
[10]

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 مولفه‌ی 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 پایگاه‌داده 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 پایگاه‌داده جمع‌آوری می‌نماییم. از جمله داده‌های فرار می‌توان به اطلاعات مربوط به کاربران وارد شده به سامانه، پردازه‌های در حال اجرا، پورت‌های و فایل‌های باز، اشاره کرد. همچنین تمامی فایل‌های رویدادنگاری پایگاه‌داده، 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 پایگاه‌داده 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 پردازه‌های موجود در SGA را نیز استخراج و جمع‌آوری کرد. تقریباً هرکاری که در پایگاه‌داده انجام می‌شود، آن را تغییر می‌دهد. بنابراین استخراج شواهد از پایگاه‌داده 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 غیرفرار:** پس از جمع‌آوری داده‌های فرار که حساسیت بیشتری برای جمع‌آوری اطلاعات نسبت به سایر اطلاعات را دارند، سایر شواهد را نیز از پایگاه‌داده جمع‌آوری می‌کنیم. از جمله این شواهد می‌توان به لیست کاربران، مجوزهای کاربران و عضویت در نقش‌ها اشاره کرد.

۹. **قطع اتصال پایگاه‌داده 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[1] 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[2] 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

[1] Timeline [2]

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 آمده‌است نیز مستند شود. تمامی اعمال مهاجم به همراه نشانه‌هایی از سرقت داده‌ها باید ثبت شوند. ثبت اطلاعاتی همچون سیستم عامل سِرور 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 پایگاه‌داده می‌توان به صورت ساختاریافته‌ای عملیات مربوط به جرم‌شناسی پایگاه‌داده 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 پایگاه‌داده نسبت به حذف برخی از نام‌های کاربری از پایگاه‌داده 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 پایگاه‌داده 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.

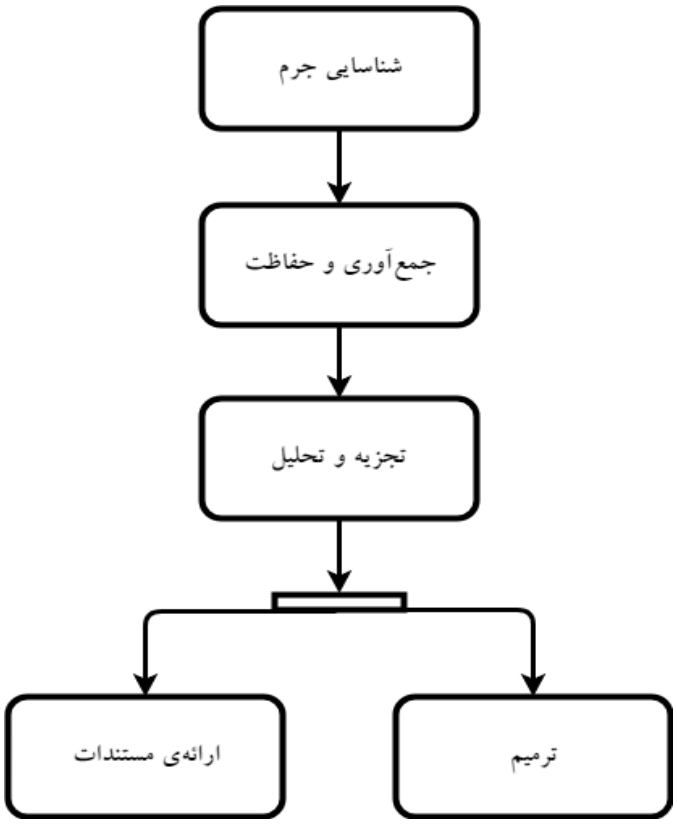
---

[2] Restore [2]

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 تجزیه‌وتحلیل، با بررسی و تحلیل داده‌های جمع‌آوری شده، اطلاعاتی همچون چه‌کسی تغییر را ایجاد کرده، چه‌زمانی 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 پایگاه‌داده 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 پایگاه‌داده، تمرکز ما در فرآیند جرم‌شناسی پایگاه‌داده تنها بر روی اطلاعات ثبت‌شده در پایگاه‌های داده می‌باشد و استفاده از اطلاعات ثبت‌شده 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)[۲۳]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)[۲۴]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.

---

2 Data Manipulation Language 3
2 Data Definition Language 4

- تلاش برای شناسایی شناسه یکتای سامانه[۵](SID) و نهایتا دسترسی به حساب‌های کاربری و اعمال نفوذ در سامانه.

لازم به ذکر است که در برخی سمپادها، این تلاش تنها از طریق حدس نام‌کاربری 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 رویدانگاری به منظور پاک کردن شواهد رویدادهای مجرمانه در پایگاه‌داده است.

جمع‌آوری اطلاعات و شواهد:

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

درصورتی که بر روی فایل‌های رویدادنگاری و ممیزی به صورت دوره‌ای اطلاعاتی ثبت شود، به هنگام شناسایی رویداد غیرمجاز می‌بایست ابتدا سرور پایگاه‌داده 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 پایگاه‌داده، با توجه به نوع رویداد می‌توان از منابع مختلفی برای جمع‌آوری اطلاعات استفاده کرد. به

---

[2] System Identifier [5]

عنوان نمونه، برای جمع‌آوری اطلاعات مربوط به برخی از رویدادها می‌بایست از رویدادهای ثبت‌شده 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 پایگاه‌داده، 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 پایگاه‌داده 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 مشابه‌ای را تجربه می‌کنند نیز کمک خواهد کرد. به منظور ارائه مستندات مربوط به رویدادهای غیرمجاز کشف‌شده در سامانه، فرم استاندارد زیر ارایه شده است.

جدول ۱: تهیه‌ی مستند از فرآیند جرم‌شناسی پایگاه‌داده

عنوان رویداد			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی			
منابع رویدادنگاری			
شیوه یا ابزار تحلیل			
امکان ترمیم			

۳-۲ 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[6]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 ارایه شده‌اند:

- افرادی که به پایگاه‌داده وارد یا از آن خارج می‌شوند.
- افرادی که خود را به جای مدیر پایگاه‌داده نمایش می‌دهند.

---

[2] Checksum [6]

- تلاش برای حمله‌ی تزریق SQL[۲۷]
- 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 پایگاه‌داده 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 پایگاه‌داده‌ی تغییریافته به‌دست‌آورد، می‌توان از طریق ابزارهایی که توسط سامانه پایگاه‌داده استفاده می‌شوند؛ همچون نهان‌گاه طرح اجرایی[۲۸] و رویدادنگاری تراکنش‌ها[۲۹] 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 پایگاه‌داده و برای بررسی‌ها و تحقیقات مختلف، مفید است. سایر منابع شامل فایل‌هایی هستند که تاریخچه‌ی مربوط به پایگاه‌داده را ذخیره می‌کنند. برخی از این فایل‌ها به طور اختصاصی در پایگاه‌داده کاربرد دارند، همچون فایل رویدادنگاری پایگاه‌داده و فایل‌های داده در حالی که سایر فایل‌ها همچون رویدادنگاری سرور وب و رویدادنگاری رویدادهای سیستمی یک سیستم‌عامل به طورخاص به سرور پایگاه‌داده اختصاص داده نمی‌شوند. در هنگام تصمیم‌گیری درمورد 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 پایگاه‌داده 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 پایگاه‌داده همچون بروزرسانی، درج و حذف داده‌ها ثبت می‌شوند. گاهی نیاز است که شواهدی برای بررسی سرقت داده‌ها شناسایی شود. سرقت داده‌ها، ممکن است به

---

2 SQL Injection
2 Execution plan cache
2 Transaction logs
7
8
9

نحوی باشد که داده‌ها را از پایگاه‌داده حذف نکند. بنابراین می‌بایست به دنبال شواهدی برای دسترسی به داده‌ها از طریق پایگاه‌داده 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[۳۰]میان شواهد استفاده می‌شود. به عنوان مثال، در پایگاه‌داده اوراکل، ممکن است شواهدی مبنی بر ورود مهاجم و ایجاد اتصال به پایگاه‌داده وجود داشته باشد. مهاجم ممکن است پرسمان SELECT را وارد کرده باشد و بهینه‌ساز[۳۱]ر اوراکل وارد عملی 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 درمورد 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. درصورتی 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

---

30 Correlation
31 Optimizer
32 Rotate

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 نشان‌دهنده‌ی تغییر غیرمجاز در این فایل‌ها و عدم اعتبار اطلاعات ذخیره شده، است.

```
ORA-00308: cannot open archived log 'C:\Users\Desktop\oracle\oradata\orcl\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
           vaild. 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 ارایه گردید.

جدول ۲: تهیه‌ی مستند از فرآیند جرم‌شناسی پایگاه‌داده

عنوان رویداد			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی			
منابع رویدادنگاری			
شیوه یا ابزار تحلیل			
امکان ترمیم			

۳ درج، حذف، تغییر 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، ویرایش و حذف سطر (های) موجود و همچنین مشاهده تمامی یا بخشی از جدول (ها) در پایگاه‌داده می‌شود، می‌پردازیم. رویدادهای فوق از جمله دستورات دستکاری داده (DML)[۳۳] در پایگاه‌داده به حساب می‌آیند. از آنجاییکه رویدادهای مورد بحث در این فصل از نقطه نظر جرم‌شناسی به هم نزدیک هستند، 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.

[3] Data Manipulation Language [3]

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. در پایگاه‌داده‌ی MySQL با استفاده از رویدادنگاری General query log، رویدادنگاری binary log و همچنین با استفاده از ممیزی می‌توان اطلاعات مربوط به رویداد حذف داده از جدول را جمع‌آوری کرد. از این‌رو، در ادامه به معرفی این منابع و نحوه آگاهی و تنظیم آنها می‌پردازیم.

استفاده از رویدادنگاری General query log: در رویدادنگاری General query log دو نوع اطلاعات ثبت می‌شود:

۱. اطلاعات مربوط به اتصال و قطع اتصال کلاینت‌ها
۲. عبارات SQL دریافت شده از کلاینت‌ها

مقصد خروجی general query log، می‌تواند یکی از مقادیر FILE، TABLE یا NONE باشد. درصورتی‌که FILE انتخاب شود، رویدادها در فایل مشخص‌شده در پارامتر general_log_file ثبت می‌شوند. با انتخاب TABLE، رویدادها در جدول ثبت می‌شوند و NONE باعث غیرفعال شدن general query log می‌شود.

به‌صورت پیش‌فرض general query log در MySQL غیرفعال است. به‌منظور شناسایی وضعیت general query log دستور زیر اجرا می‌شود (شکل ۴).

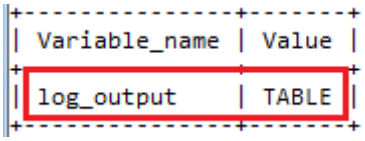
```
SHOW VARIABLES LIKE '%general_log%';
```

```
+------------------+-----------------------------------------------------------------------------------------+
| Variable_name    | Value                                                                                   |
+------------------+-----------------------------------------------------------------------------------------+
| general_log      | ON                                                                                      |
| general_log_file | C:\Users\TESTUSER\Desktop\mysql enterprise\mysql-advanced-5.7.21-winx64\data\TestUser-PC.log |
+------------------+-----------------------------------------------------------------------------------------+
```

شکل ۴: بررسی وضعیت general query log

همچنین با استفاده از دستور زیر می‌توان مسیر خروجی general query log را به دست آورد (شکل ۵).

```
SHOW VARIABLES LIKE '%log_output%';
```



شکل ۵: مقصد خروجی general query log

نکات تکمیلی: با استفاده از دستورات زیر می‌توان general query log را فعال و جدول را به عنوان خروجی آن انتخاب کرد.

```
SET global general_log = 1;
SET global log_output = 'TABLE';
```

در پایگاه‌داده‌ی MySQL می‌توان تنظیمات مورد نیاز را در فایلی به عنوان مثال با نام my.ini قرار داد تا هنگام شروع به کار سرویس mysqld، اطلاعات پیکربندی از آن فایل خوانده شود. امکان تعریف پارامترهای مختلف از جمله پارامترهای فوق در فایل my.ini وجود دارد:

```
general_log = on
log-output=TABLE
```

همچنین می‌توان پارامتر log_output را برای نوشتن اطلاعات بر روی فایل تنظیم کرد:

```
SET global general_log_file='/tmp/mysql.log';
SET global log_output = FILE;
```

جدول زیر، مراحل اشاره شده در بالارا به طور خلاصه نشان می‌دهد.

بررسی فعال بودن general query log		
SHOW VARIABLES LIKE '%general_log%';		
فعال کردن general query log (اعمال تنظیمات در فایل پیکربندی)		
۱	فعال کردن general query log	general_log = on
۲	ثبت رویدادها در جدول	log-output=TABLE
۳	ثبت رویدادها در فایل	general_log_file='/tmp/mysql.log' log-output=FILE
فعال کردن general query log (اعمال تنظیمات با دستورات)		
۱	فعال کردن general	SET global general_log = 1;

	query log	
۲	ثبت رویدادها در جدول	SET global log_output = 'TABLE';
۳	ثبت رویدادها در فایل	SET global general_log_file='/tmp/mysql.log'; SET global log_output = FILE;
برخی دستورات سودمند		
۱	یافتن مقصد خروجی general query log	SHOW VARIABLES LIKE '%log_output%';

استفاده از رویدادنگاری binary log: این رویدادنگاری شامل رویدادهایی است که تغییرات پایگاه‌داده را توصیف می‌کنند (مثل عملیات ایجاد جدول یا تغییر در داده‌های جدول). این رویدادنگاری همچنین شامل عباراتی است که پتانسیل ایجاد تغییر را دارند؛ همچون یک عبارت DELETE که با هیچ سطری منطبق نشود. Binary log برای عباراتی همچون SELECT یا SHOW که داده‌ها را تغییر نمی‌دهند، کاربرد ندارد. با استفاده از دستورات زیر، وضعیت فعال یا غیرفعال بودن binary log بررسی می‌شود (شکل ۶):

```
SELECT @@log_bin;
SHOW VARIABLES LIKE '%log_bin%';
```

```
+---------------------------------+-----------------------------------------------------------------------------------+
| Variable_name                   | Value                                                                             |
+---------------------------------+-----------------------------------------------------------------------------------+
| log_bin                         | ON                                                                                |
| log_bin_basename                | C:\Users\TESTUSER\Desktop\mysql enterprise\mysql-advanced-5.7.21-winx64\binary-log\binary-log       |
| log_bin_index                   | C:\Users\TESTUSER\Desktop\mysql enterprise\mysql-advanced-5.7.21-winx64\binary-log\binary-log.index |
| log_bin_trust_function_creators | OFF                                                                               |
| log_bin_use_v1_row_events       | OFF                                                                               |
| sql_log_bin                     | ON                                                                                |
+---------------------------------+-----------------------------------------------------------------------------------+
```

شکل ۶: بررسی وضعیت binary log

همچنین با استفاده از دستور زیر ساختار ذخیره‌سازی اطلاعات در فایل‌های binary log مشخص می‌شود (شکل ۷).

```
SHOW VARIABLES like '%binlog_format%';
```

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| binlog_format | ROW   |
+---------------+-------+
```

شکل ۷: ساختار ذخیره‌سازی اطلاعات در فایل‌های binary log

نکات تکمیلی: به منظور فعال‌سازی binary log در فایل پیکربندی my.ini پارامتر log-bin به همراه مسیر ذخیره‌سازی فایل‌های رویدادنگاری، مقداردهی می‌شوند.

```
log-bin="MYSQL_HOME/binary-log/binary-log"
```

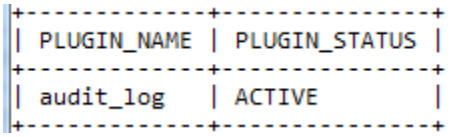
```
server-id=master-01
binlog-format=ROW
```

جدول زیر، مراحل اشاره شده در بالارا به طور خلاصه نشان می‌دهد.

بررسی فعال بودن binary log		
SELECT @@log_bin; SHOW VARIABLES LIKE '%log_bin%';		
فعال کردن binary log **(اعمال تنظیمات در فایل پیکربندی)**		
log-bin="MYSQL_HOME/binary-log/binary-log"	فعال کردن binary log	۱
binlog-format=ROW binlog-format=STATEMENT	تنظیم ساختار ذخیره‌سازی اطلاعات	۲
server-id=master-01	تنظیم شناسه‌ی یکتا برای سرور	۳
برخی دستورات سودمند		
SHOW VARIABLES like '%binlog_format%';	یافتن ساختار ذخیره‌سازی اطلاعات در فایل‌های binary log	۱

استفاده از ممیزی: نسخه‌ی MySQL Enterprise شامل MySQL Enterprise Audit است که به‌صورت پلاگینی با نام audit_log پیاده‌سازی شده است. با استفاده از MySQL Enterprise Audit می‌توان بر اتصالات و پرسمان‌هایی که بر روی سرور MySQL اجرا می‌شوند، نظارت و رخدادها را ثبت کرد. زمانی که پلاگین نصب شود، پلاگین این امکان را برای سرور MySQL فراهم می‌کند که فایل ثبتی حاوی رکوردهای فعالیت‌های سرور تولید شود. محتوای فایل ثبت شامل اتصال و قطع اتصال کلاینت از سرور و فعالیت‌هایی است که در حین اتصال، کلاینت انجام می‌دهد. فایل ثبت پس از نصب پلاگین در دایرکتوری Data با نام audit.log ایجاد می‌شود. این فایل دارای فرمت XML است و محتوای آن، رمز شده نیست. این فایل می‌تواند شامل اطلاعات حساس همچون متن عبارات SQL باشد. برای امنیت بیشتر این فایل باید در دایرکتوری نوشته شود که تنها برای سرور MySQL و کاربران مجاز قابل‌دسترسی باشد. با استفاده از دستور زیر می‌توان وضعیت نصب پلاگین audit_log را بررسی کرد (شکل ۸):

```
SELECT PLUGIN_NAME, PLUGIN_STATUS  FROM INFORMATION_SCHEMA.PLUGINS
                                    WHERE PLUGIN_NAME LIKE 'audit%';
```

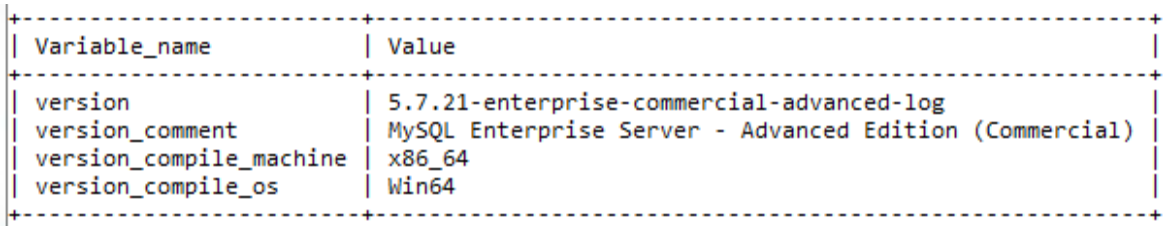


PLUGIN_NAME	PLUGIN_STATUS
audit_log	ACTIVE

شکل ۸: بررسی وضعیت نصب پلاگین audit_log

اطلاعات تکمیلی: به‌منظور نصب پلاگین ابتدا باید از موجود بودن نسخه‌ی MySQL Enterprise اطمینان حاصل شود (شکل ۹).

```
INSTALL PLUGIN audit_log SONAME 'audit_log.dll';
```



Variable_name	Value
version	5.7.21-enterprise-commercial-advanced-log
version_comment	MySQL Enterprise Server - Advanced Edition (Commercial)
version_compile_machine	x86_64
version_compile_os	Win64

شکل ۹: بررسی نسخه‌ی MySQL

به‌منظور بارگذاری پلاگین و ثبت آن در جدول سیستمی mysql.plugins با هدف بارگذاری پلاگین در راه‌اندازی‌های بعدی سرور، دستور زیر اجرا می‌شود:

جدول زیر، مراحل اشاره شده در بالارا به طور خلاصه نشان می‌دهد.

بررسی وضعیت نصب پلاگین audit_log		
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'audit%';		
نصب پلاگین audit_log		
SHOW VARIABLES LIKE 'version%';	اطمینان از وجود نسخه‌ی MySQL Enterprise	۱
INSTALL PLUGIN audit_log SONAME 'audit_log.dll';	بارگذاری پلاگین و ثبت آن در جدول سیستمی mysql.plugins	۲

۳-۳ استخراج و تجزیه و تحلیل اطلاعات

در این بخش نحوه‌ی استخراج اطلاعات برای هر یک از منابع، 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، شناسایی نماید. در ادامه برای هر یک از منابع حاوی شواهد برای رویدادهای سامانه، نحوه استخراج و تجزیه و تحلیل شواهد تشریح می‌گردد.

رویدادنگاری General query log: در صورت انتخاب مقدار TABLE برای پارامتر log_output می‌توان رویدادهای ثبت شده را با پرسمان زیر مشاهده کرد (شکل ۱۰):

```
SELECT * FROM mysql.general_log;
```

```
+----------------------------+-----------------------------------+-----------+-----------+--------------+-----------------------------------
-----------------------------------------------------------------+
| event_time                 | user_host                         | thread_id | server_id | command_type | argument
                                                                 |
+----------------------------+-----------------------------------+-----------+-----------+--------------+-----------------------------------
-----------------------------------------------------------------+
| 2018-09-20 11:37:27.773457 | [root] @ localhost [127.0.0.1]    |         5 |         0 | Connect      | root@localhost on  using SSL/TLS
                                                                 |
| 2018-09-20 11:37:27.774457 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | select @@version_comment limit 1
                                                                 |
| 2018-09-20 11:38:50.315178 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | select @@version
                                                                 |
| 2018-09-20 12:36:20.344508 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | SHOW VARIABLES LIKE '%general_log
'                                                                |
| 2018-09-20 12:36:52.666357 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | SHOW VARIABLES LIKE '%log_output%
                                                                 |
| 2018-09-20 12:42:23.376273 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | SELECT @@log_bin
                                                                 |
| 2018-09-20 12:42:32.975822 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | SHOW VARIABLES LIKE '%log_bin%'
                                                                 |
| 2018-09-20 13:24:56.174284 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | create table test (a int, b int)
                                                                 |
| 2018-09-20 13:25:05.421813 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | insert into test values (1,2)
                                                                 |
| 2018-09-20 13:25:09.582051 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | insert into test values (3,4)
                                                                 |
| 2018-09-20 13:25:12.998247 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | insert into test values (5,6)
                                                                 |
| 2018-09-20 13:25:17.293492 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | select * from test
                                                                 |
| 2018-09-20 13:25:25.693973 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | delete from test where a = 1
                                                                 |
| 2018-09-20 13:25:28.302122 | root[root] @ localhost [127.0.0.1] |         5 |         0 | Query        | select * from test
                                                                 |
```

شکل ۱۰: خروجی general query log در جدول

همچنین نمونه‌ای از خروجی ذخیره شده در فایل، در شکل ۱۱ نشان داده شده‌است.

```
Time                 Id Command    Argument
2018-09-20T10:04:04.791663Z      3 Connect   root@localhost on dbtest using SSL/TLS
2018-09-20T10:04:04.792663Z      3 Query SELECT * FROM mysql.general_log
2018-09-20T10:04:21.464617Z      3 Query select * from test
2018-09-20T10:04:27.144942Z      3 Query delete from test where a = 3
```

شکل ۱۱: خروجی general query log در فایل

همانطور که در شکل ۱۰ و شکل ۱۱ دیده می‌شود، با انتخاب TABLE به عنوان خروجی general log output، می‌توان از اطلاعات بیشتری در مورد پرسمان اجراشده بهره‌مند شد.

مشاهده‌ی رویدادهای general query log در جدول
SELECT * FROM mysql.general_log;

استخراج و تحلیل اطلاعات موجود در رویـدادنگاری binary log: بـا اسـتفاده از دسـتور زیـر می‌تـوان فهرستی از نام‌های فایل‌های binary log به دست آورد (شکل ۱۲).

```
SHOW BINARY LOGS;
```

```
+-------------------+-----------+
| Log_name          | File_size |
+-------------------+-----------+
| binary-log.000001 |       177 |
| binary-log.000002 |      1794 |
| binary-log.000003 |       418 |
+-------------------+-----------+
```

شکل ۱۲: فهرست فایل‌های binary log

همچنین برای مشاهده‌ی نام فایل جاری binary log، دستور زیر اجرا می‌شود (شکل ۱۳).

```
SHOW MASTER STATUS;
```

```
+-------------------+----------+--------------+------------------+-------------------+
| File              | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-------------------+----------+--------------+------------------+-------------------+
| binary-log.000003 |      418 |              |                  |                   |
+-------------------+----------+--------------+------------------+-------------------+
```

شکل ۱۳: فایل جاری binary log

در صورتی که مقدار پارامتر binlog-format برابر STATEMENT باشد، با اجرای دستور زیر، امکان مشاهده‌ی عبارت‌های مربوط به هر پرسمان اجرایی به صورت کامل وجود دارد (شکل ۱۴).

```
SHOW BINLOG EVENTS IN 'binary-log.000011';
```

```
+-------------------+-----+---------------+-----------+-------------+-----------------------------------------------
| Log_name          | Pos | Event_type    | Server_id | End_log_pos | Info
+-------------------+-----+---------------+-----------+-------------+-----------------------------------------------
| binary-log.000004 |   4 | Format_desc   |         0 |         123 | Server ver: 5.7.21-enterprise-commercial-advance
| binary-log.000004 | 123 | Previous_gtids|         0 |         154 |
| binary-log.000004 | 154 | Anonymous_Gtid|         0 |         219 | SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
| binary-log.000004 | 219 | Query         |         0 |         302 | BEGIN
| binary-log.000004 | 302 | Query         |         0 |         408 | use `dbtest`; delete from test where a = 5
| binary-log.000004 | 408 | Xid           |         0 |         439 | COMMIT /* xid=5 */
+-------------------+-----+---------------+-----------+-------------+-----------------------------------------------
```

شکل ۱۴: خروجی binary log با ساختار STATEMENT

ولی در صورتی که مقدار این پارامتر برابر ROW باشد، خروجی حاصل از دستور فوق به صورت شکل ۱۵ خواهد بود که در آن عبارت حذف به طور کامل نشان داده نشده‌است.

Log_name	Pos	Event_type	Server_id	End_log_pos	Info
binary-log.000003	4	Format_desc	0	123	Server ver: 5.7.21-enterprise-commercial-ad
binary-log.000003	123	Previous_gtids	0	154	
binary-log.000003	154	Anonymous_Gtid	0	219	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
binary-log.000003	219	Query	0	293	BEGIN
binary-log.000003	293	Table_map	0	343	table_id: 108 (dbtest.test)
binary-log.000003	343	Delete_rows	0	387	table_id: 108 flags: STMT_END_F
binary-log.000003	387	Xid	0	418	COMMIT /* xid=4 */

شکل ۱۵: خروجی binary log با ساختار ROW

همانطور که در شکل ۱۴ و شکل ۱۵ دیده می‌شود، با استفاده از رویدادهای ثبت شده در binary log نمی‌توان نام کاربری را به دست آورد. در حقیقت فایل‌های binary log به منظور ثبت تمامی تغییرات پایگاه‌داده با هدف تکرار[۳۴] و بازیابی[۳۵] ایجاد می‌شوند و در نتیجه اطلاعات اضافه به غیر از تغییرات اعمال شده بر روی داده‌ها را در خود ذخیره نمی‌کنند.

مشاهده‌ی محتوای فایل binary log		
SHOW BINLOG EVENTS IN '<FILE_NAME>';		
برخی دستورات سودمند		
SHOW BINARY LOGS;	فهرست فایل‌های binary log	۱
SHOW MASTER STATUS;	مشاهده‌ی نام فایل جاری binary log	۲

استخراج و تحلیل اطلاعات با استفاده از ممیزی: برای این منظور، فایل audit.log در دایرکتوری Data ایجاد می‌شود. نمونه‌ای از این فایل در شکل ۱۶ نشان داده شده‌است.

[3] Replication
[3] Recovery
[4]
[5]

```
<AUDIT_RECORD>
<TIMESTAMP>2018-09-20T10:23:47 UTC</TIMESTAMP>
<RECORD_ID>6_2018-09-20T10:23:24</RECORD_ID>
<NAME>Query</NAME>
<CONNECTION_ID>3</CONNECTION_ID>
<STATUS>0</STATUS>
<STATUS_CODE>0</STATUS_CODE>
<USER>root[root] @ localhost [127.0.0.1]</USER>
<OS_LOGIN/>
<HOST>localhost</HOST>
<IP>127.0.0.1</IP>
<COMMAND_CLASS>delete</COMMAND_CLASS>
<SQLTEXT>delete from test where a = 5</SQLTEXT>
</AUDIT_RECORD>
```

شکل ۱۶: فایل audit.log

همانطور که در شکل ۱۶ دیده می‌شود، با استفاده از اطلاعات ذخیره شده در فایل ممیزی، می‌توان پرسمان اجرایی، زمان و کاربر اجراکننده‌ی آن را استخراج کرد. همچنین وضعیت اجرای پرسمان (اجرای موفق یا اجرای با خطا) نیز در این فایل مشخص می‌شود.

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 روشی به غیر از روش‌های مربوط به تهیه‌ی نسخه‌های پشتیبان را برای بازیابی داده‌های از دست رفته مورد بحث و بررسی قرار دهیم. در این روش از فایل‌های رویدادنگاری binary log برای ترمیم و بازیابی داده‌های از دست رفته استفاده می‌شود. با توجه به آنکه فایل‌های binary log اعمال DML اجرا شده بر روی یک جدول را ذخیره می‌کنند، با استفاده از محتوای این فایل‌ها می‌توان داده‌های از دست رفته را به دست آورد. ابتدا، با استفاده از دستوراتی مشابه دستور زیر، فایل‌های binary log به عبارت‌های SQL تبدیل می‌شوند.

```
mysqlbinlog.exe MYSQL_HOME\binary-log\binary-log.000008 > query_log.sql
```

خروجی دستور بالا، در شکل ۱۷ نشان داده شده‌است.

```
/*!*/;
# at 1635
#180220 23:51:44 server id 0  end_log_pos 1746 CRC32 0x190041cc     Query   thread_id=
SET TIMESTAMP=1519158104/*!*/;
insert into test2 values (3, 'c')
/*!*/;
# at 1746
#180220 23:51:44 server id 0  end_log_pos 1777 CRC32 0x00290d0d     Xid = 27
COMMIT/*!*/;
# at 1777
#180220 23:52:02 server id 0  end_log_pos 1842 CRC32 0xd1f695f7     Anonymous_GTID  la
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 1842
#180220 23:52:02 server id 0  end_log_pos 1925 CRC32 0xfdffc1bd     Query   thread_id=
SET TIMESTAMP=1519158122/*!*/;
BEGIN
/*!*/;
# at 1925
#180220 23:52:02 server id 0  end_log_pos 2030 CRC32 0x1c9007f2     Query   thread_id=
SET TIMESTAMP=1519158122/*!*/;
delete from test2 where a=3
```

شکل ۱۷: تبدیل فایل binary log به عبارت‌های SQL

همانطور که در شکل ۱۷ دیده می‌شود، می‌توان در فایل‌های binary log به دنبال عبارت‌های درج معادل عبارت حذف گشت و آن‌ها را مجدداً اجرا کرد. همچنین می‌توان یک نمونه‌ی جدید از MySQL را راه‌اندازی کرد و فایل‌های SQL ایجاد شده با استفاده از دستور فوق را از زمان ایجاد جدول تا پیش از دستور حذف، در نمونه‌ی جدید اجرا کرد. بدین ترتیب کل جدول با داده‌هایی که پیش از عمل حذف غیرمجاز/ناخواسته در آن وجود داشته‌اند، بازیابی می‌شوند. به عنوان مثال، پس از ایجاد فایل query_log.sql مشابه شکل ۱۷، خط مربوط به پرسمان حذف را پاک کرده و سپس دستور زیر اجرا می‌شود تا جدول و محتوای آن بازیابی شوند. در اینجا فرض بر آن است که پرسمان ایجاد جدول و درج محتوای آن، همگی در یک فایل binary log ذخیره شده‌اند.

```
mysql.exe -u root < query_log.sql
```

همچنین در صورتی که مقدار پارامتر binlog-format برابر ROW باشد و هنگام تبدیل محتوای فایل binary log به عبارات SQL از دستور زیر استفاده شود، جزئیات بیشتری در فایل حاوی عبارات SQL نمایش داده می‌شود.

```
mysqlbinlog.exe -v MYSQL_HOME\binary-log\binary-log.000008 > query_log08.sql
```

```
# at 1186
#180221 12:30:21 server id 0  end_log_pos 1229 CRC32 0x2e61d03a     Delete_rows: table id 96 flags: STMT_END_F

BINLOG '
JTWNWhMAAAAANQAAAKIEAAAAAGAAAAAAAAAEABmRidGVzdAAFdGVzdDIAAgMPAgoAA/Wyv98=
JTWNWiAAAAAAKwAAAM0EAAAAAGAAAAAAAAAEAAgAC//wBAAAAAmNjOtBhLg==
'/*!*/;
### DELETE FROM `dbtest`.`test2`
### WHERE
###   @1=1
###   @2='cc'
# at 1229
#180221 12:30:21 server id 0  end_log_pos 1260 CRC32 0xefc55c32     Xid = 9
COMMIT/*!*/;
```

شکل ۱۸: تبدیل فایل binary log به عبارت‌های SQL

همانطور که در شکل ۱۸ دیده می‌شود، سطر حذف شده در ستون اول مقدار ۱ و در ستون دوم مقدار cc داشته است. بنابراین می‌توان پرسمان درج را با محتوای ۱ و cc برای بازیابی سطر اجرا کرد. در صورتی که پرسمان حذف بدون عبارت WHERE اجرا شود یا پرسمان حذف چندین سطر را شامل شود، خروجی در قالب شکل ۱۹ خواهد بود؛ یعنی تمامی سطرهای حذف شده، مشخص می‌شوند.

```
# at 1726
#180221 14:05:56 server id 0  end_log_pos 1785 CRC32 0x1de0b2d3     Delete_rows: table id 96 flags: STMT_END_F

BINLOG '
jEuNWhMAAAAANQAAAL4GAAAAAGAAAAAAAAAEABmRidGVzdAAFdGVzdDIAAgMPAgoAA/9Q94k=
jEuNWiAAAAAAOwAAAPkGAAAAAGAAAAAAAAAEAAgAC//wCAAAAAmNj/AEAAAACYWH8AwAAAAJkZNOy
4B0=
'/*!*/;
### DELETE FROM `dbtest`.`test2`
### WHERE
###   @1=2
###   @2='cc'
### DELETE FROM `dbtest`.`test2`
### WHERE
###   @1=1
###   @2='aa'
### DELETE FROM `dbtest`.`test2`
### WHERE
###   @1=3
###   @2='dd'
```

شکل ۱۹: تبدیل فایل binary log به عبارت‌های SQL – حذف داده‌های جدول

خروجی برای درج داده‌ها نیز در شکل ۲۰ به تصویر کشیده شده است.

```
#180221 14:05:44 server id 0  end_log_pos 1503 CRC32 0xa96f1a7c     Write_rows: table id 96

BINLOG '
gEuNWhMAAAAANQAAAKwFAAAAAGAAAAAAAAAEABmRidGVzdAAFdGVzdDIAAgMPAgoAA0B44eM=
gEuNWh4AAAAAMwAAAN8FAAAAAGAAAAAAAAAEAAgAC//wBAAAAAmFh/AMAAAACZGR8Gm+p
'/*!*/;
### INSERT INTO `dbtest`.`test2`
### SET
###   @1=1
###   @2='aa'
### INSERT INTO `dbtest`.`test2`
### SET
###   @1=3
###   @2='dd'
# at 1503
```

شکل ۲۰: تبدیل فایل binary log به عبارت‌های SQL – درج داده در جدول

به طور مشابه می‌توان داده‌های به‌روزرسانی شده در یک جدول را به دست آورد (شکل ۲۱).

```
#180221 12:17:51 server id 0  end_log_pos 963 CRC32 0x0cba3dd7  Update_rows: table id 96

BINLOG '
NzKNWhMAAAAANQAAAH8DAAAAAGAAAAAAAAAEABmRidGVzdAAFdGVzdDIAAgMPAgoAAyYMdFk=
NzKNWh8AAAAARAAAAMMDAAAAAGAAAAAAAAAEAAgAC///8AQAAAAJhYfwBAAAAAmNj/AIAAAACYWH8
AgAAAAJjY9c9ugw=
'/*!*/;
### UPDATE `dbtest`.`test2`
### WHERE
###   @1=1
###   @2='aa'
### SET
###   @1=1
###   @2='cc'
### UPDATE `dbtest`.`test2`
### WHERE
###   @1=2
###   @2='aa'
### SET
###   @1=2
###   @2='cc'
# at 963
```

شکل ۲۱: تبدیل فایل binary log به عبارت‌های SQL – به‌روزرسانی داده در جدول

جدول زیر، مراحل اشاره شده در بالارا برای ترمیم به طور خلاصه نشان می‌دهد.

ذخیره‌سازی فایل‌های binary log در قالب عبارات SQL
MYSQL_HOME\bin\mysqlbinlog.exe -v BINARY_LOG_PATH\FILE_NAME > query_log.sql
اجرای فایل حاوی عبارات SQL
MYSQL_HOME\bin\mysql.exe -u <USER_NAME> < query_log.sql

۳-۵ ارائه‌ی مستندات

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

حذف غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐	رویدادنگاری binary log ☐	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	استفاده از محتوای فایل‌های binary log ☐		
توضیحات			

درج غیرمجاز در جدول			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐	رویدادنگاری binary log ☐	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	حذف سطر(های) درج شده ☐		
توضیحات			

مشاهده غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐		

شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.
امکان ترمیم	فاقد امکان ترمیم است.
توضیحات	

بروزرسانی غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐	رویدادنگاری binary log ☐	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	استفاده از محتوای فایل‌های binary log ☐		
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 سطر (های) موجود، تغییر سطر(های) موجود یا مشاهده تمامی یا بخشی از جدول (ها) در پایگاه می‌گردند، پرداختیم. برای این منظور، پس از شناسایی جرم، سه رویکرد به نام‌های رویدادنگاری general query log، رویدادنگاری binary log و ممیزی با پلاگین audit_log را برای جمع‌آوری شواهد و اطلاعات مربوط به جرم معرفی کردیم. در ادامه نیز تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و تهیه مستندات در این رابطه را مورد بحث و بررسی قرار دادیم.

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)[3] 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. از این‌رو، در ادامه به معرفی منابع مربوط به شواهد رویداد حذف غیرمجاز یک جدول و نحوه آگاهی و تنظیم پارامترهای مربوط به آن می‌پردازیم. در پایگاه‌داده‌ی PostgreSQL با استفاده از رویدادنگاری General query log، رویدادنگاری binary log و همچنین با استفاده از ممیزی می‌توان اطلاعات مربوط به رویداد تغییر در شمای پایگاه‌داده را جمع‌آوری کرد که در ادامه به تشریح هر یک از آنها می‌پردازیم.

استفاده از رویدادنگاری General query log: در رویدادنگاری General query log دو نوع اطلاعات ثبت می‌شوند:

- اطلاعات مربوط به اتصال و قطع اتصال کلاینت‌ها
- عبارات SQL دریافت شده از کلاینت‌ها

مسیر خروجی general query log، می‌تواند یکی از مقادیر FILE، TABLE یا NONE باشد. درصورتی‌که FILE انتخاب شود، رویدادها در فایل مشخص‌شده در پارامتر general_log_file ثبت می‌شوند. با انتخاب TABLE، رویدادها در جدول ثبت می‌شوند و NONE باعث غیرفعال شدن general query log می‌شود. به‌صورت پیش‌فرض general query log در MySQL غیرفعال است. به‌منظور شناسایی وضعیت general query log از دستور زیر بهره می‌گیریم.

---

[3] Data Definition Language [6]

```
SHOW VARIABLES LIKE '%general_log%';
```

همچنین با استفاده از دستور زیر می‌توان مقصد خروجی general query log را به دست آورد.

```
SHOW VARIABLES LIKE '%log_output%';
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای جمع‌آوری اطلاعات و شواهد با استفاده از رویدادنگاری General query log را نشان می‌دهد.

بررسی فعال بودن general query log		
SHOW VARIABLES LIKE '%general_log%';		
فعال کردن general query log **(اعمال تنظیمات در فایل پیکربندی)**		
۱	فعال کردن general query log	general_log = on
۲	ثبت رویدادها در جدول	log-output=TABLE
۳	ثبت رویدادها در فایل	general_log_file='/tmp/mysql.log' log-output=FILE
فعال کردن general query log **(اعمال تنظیمات با دستورات)**		
۱	فعال کردن general query log	SET global general_log = 1;
۲	ثبت رویدادها در جدول	SET global log_output = 'TABLE';
۳	ثبت رویدادها در فایل	SET global general_log_file='/tmp/mysql.log'; SET global log_output = FILE;
برخی دستورات سودمند		
۱	یافتن مقصد خروجی general query log	SHOW VARIABLES LIKE '%log_output%';

استفاده از رویدادنگاری binary log: این رویدادنگاری شامل رویدادهایی هستند که تغییرات پایگاه‌داده را توصیف می‌کنند. عملیات ایجاد جدول یا تغییر در داده‌های جدو را می‌توان نمونه‌هایی از این رویدادها به شمار آورد. این رویدادنگاری همچنین شامل عباراتی است که پتانسیل ایجاد تغییر را دارند؛ همچون یک عبارت DELETE که با هیچ سطری منطبق نشود. Binary log برای عباراتی همچون SELECT یا SHOW که داده‌ها را تغییر نمی‌دهند، کاربرد ندارد. با استفاده از دستورات زیر می‌توان از وضعیت binary log مطلع شد.

```
SELECT @@log_bin;
SHOW VARIABLES LIKE '%log_bin%';
```

همچنین با استفاده از دستور زیر ساختار ذخیره‌سازی اطلاعات در فایل‌های binary log مشخص می‌شود.

```
SHOW VARIABLES like '%binlog_format%';
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای جمع‌آوری اطلاعات و شواهد با استفاده از رویدادنگاری Binary log را نشان می‌دهد.

بررسی فعال بودن binary log		
SELECT @@log_bin; SHOW VARIABLES LIKE '%log_bin%';		
فعال کردن binary log **(اعمال تنظیمات در فایل پیکربندی)**		
۱	فعال کردن binary log	log-bin="MYSQL_HOME/binary-log/binary-log"
۲	تنظیم ساختار ذخیره‌سازی اطلاعات	binlog-format=ROW binlog-format=STATEMENT
۳	تنظیم شناسه‌ی یکتا برای سرور	server-id=master-01
برخی دستورات سودمند		
۱	یافتن ساختار ذخیره‌سازی اطلاعات در فایل‌های binary log	SHOW VARIABLES like '%binlog_format%';

استفاده از ممیزی: نسخه‌ی MySQL Enterprise شامل MySQL Enterprise Audit است که به‌صورت پلاگینی با نام audit_log پیاده‌سازی شده‌است. با استفاده از MySQL Enterprise Audit می‌توان بر اتصالات و پرسمان‌هایی که بر روی سرور MySQL اجرا می‌شوند، نظارت و رخدادها را ثبت کرد. زمانی که پلاگین نصب شود، پلاگین این امکان را برای سرور MySQL فراهم می‌کند که فایل ثبتی حاوی رکوردهای فعالیت‌های سرور تولید شود. محتوای فایل ثبت شامل اتصال و قطع اتصال کلاینت از سرور و فعالیت‌هایی است که در حین اتصال، کلاینت انجام می‌دهد. فایل ثبت پس از نصب پلاگین در دایرکتوری Data با نام audit.log ایجاد می‌شود. این فایل دارای فرمت XML است و محتوای آن، رمز شده نیست. این فایل می‌تواند شامل اطلاعات حساس همچون متن عبارات SQL باشد. برای امنیت بیشتر این فایل باید در

دایرکتوری نوشته شود که تنها برای سرور MySQL و کاربران مجاز قابل‌دسترسی باشد. با استفاده از دستور زیر می‌توان وضعیت نصب پلاگین audit_log را بررسی کرد:

```
SELECT PLUGIN_NAME, PLUGIN_STATUS   FROM INFORMATION_SCHEMA.PLUGINS
                                    WHERE PLUGIN_NAME LIKE 'audit%';
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای جمع‌آوری اطلاعات و شواهد با استفاده از ممیزی را نشان می‌دهد.

بررسی وضعیت نصب پلاگین audit_log		
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'audit%';		
نصب پلاگین audit_log		
SHOW VARIABLES LIKE 'version%';	اطمینان از وجود نسخه‌ی MySQL Enterprise	۱
INSTALL PLUGIN audit_log SONAME 'audit_log.dll';	بارگذاری پلاگین و ثبت آن در جدول سیستمی mysql.plugins	۲

۴-۳ 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 شناسایی نماید. در ادامه برای هر یک از منابع حاوی شواهد برای رویدادهای سامانه، نحوه استخراج و تجزیه و تحلیل تشریح می‌گردد.

رویدادنگاری General query log: در صورت انتخاب مقدار TABLE برای پارامتر log_output می‌توان رویدادهای ثبت شده را با پرسمان زیر مشاهده کرد.

```
SELECT * FROM mysql.general_log;
```

```
2018-09-20 13:24:56.174284 | root[root] @ localhost [127.0.0.1] |       5 |       0 | Query       | create table test (a int, b int)
| 2018-09-20 13:25:05.421813 | root[root] @ localhost [127.0.0.1] |       5 |       0 | Query       | insert into test values (1,2)
| 2018-09-20 13:25:09.582051 | root[root] @ localhost [127.0.0.1] |       5 |       0 | Query       | insert into test values (3,4)
```

شکل ۲۲: خروجی general query log

همانطور که در شکل ۲۲ دیده می‌شود، پرسمان اجرایی به همراه زمان و کاربر اجراکننده‌ی آن در general query log ثبت می‌شوند.

مشاهده‌ی رویدادهای general query log در جدول
SELECT * FROM mysql.general_log;

رویدادنگاری binary log: با استفاده از دستور زیر می‌توان فهرستی از نام فایل‌های binary log به دست آورد.

```
SHOW BINARY LOGS;
```

همچنین برای مشاهده‌ی نام فایل جاری binary log دستور زیر اجرا می‌شود.

```
SHOW MASTER STATUS;
```

در صورتی که مقدار پارامتر binlog-format برابر STATEMENT باشد، با اجرای دستور زیر، امکان مشاهده‌ی عبارت‌های پرسمان‌های اجرایی به صورت کامل وجود دارد (شکل ۲۳).

```
SHOW BINLOG EVENTS IN 'binary-log.0000۰1';
```

Log_name	Pos	Event_type	Server_id	End_log_pos	Info
binary-log.000004	4	Format_desc	0	123	Server ver: 5.7.21-enterprise-commercial-advanced-log, Bin
binary-log.000004	123	Previous_gtids	0	154	
binary-log.000004	154	Anonymous_Gtid	0	219	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
binary-log.000004	219	Query	0	302	BEGIN
binary-log.000004	302	Query	0	408	use `dbtest`; delete from test where a = 5
binary-log.000004	408	Xid	0	439	COMMIT /* xid=5 */
binary-log.000004	439	Anonymous_Gtid	0	504	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
binary-log.000004	504	Query	0	618	use `dbtest`; create table test_tbl (a int, b int)
binary-log.000004	618	Anonymous_Gtid	0	683	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
binary-log.000004	683	Query	0	804	use `dbtest`; DROP TABLE `test` /* generated by server */

شکل ۲۳: خروجی binary log

همانطور که در شکل ۲۳ دیده می‌شود، با استفاده از رویدادهای ثبت شده در binary log نمی‌توان نام کاربری را به دست آورد. در حقیقت فایل‌های binary log به منظور ثبت تمامی تغییرات پایگاه‌داده با هدف تکرار[۳۷] و بازیابی[۳۸] ایجاد می‌شوند و در نتیجه اطلاعات اضافه به غیر از تغییرات اعمال شده بر روی داده‌ها را در خود ذخیره نمی‌کنند.

مشاهده‌ی محتوای فایل binary log		
SHOW BINLOG EVENTS IN '<FILE_NAME>';		
برخی دستورات سودمند		
SHOW BINARY LOGS;	فهرست فایل‌های binary log	۱
SHOW MASTER STATUS;	مشاهده‌ی نام فایل جاری binary log	۲

استخراج و تحلیل اطلاعات با استفاده از ممیزی: به صورت پیش‌فرض در صورت فعال بودن ممیزی، فایل audit.log در دایرکتوری Data ایجاد می‌شود. نمونه‌ای از این فایل در شکل ۲۴ نشان داده شده‌است.

```
<AUDIT_RECORD>
 <TIMESTAMP>2018-09-20T10:40:47 UTC</TIMESTAMP>
 <RECORD_ID>14_2018-09-20T10:23:24</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>3</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [127.0.0.1]</USER>
 <OS_LOGIN/>
 <HOST>localhost</HOST>
 <IP>127.0.0.1</IP>
 <COMMAND_CLASS>drop table</COMMAND_CLASS>
 <SQLTEXT>drop table test</SQLTEXT>
</AUDIT_RECORD>
<AUDIT_RECORD>
```

شکل ۲۴: خروجی ممیزی

همانطور که در شکل ۲۴ دیده می‌شود با استفاده از اطلاعات ذخیره شده در فایل ممیزی می‌توان پرسمان اجرایی، زمان و کاربر اجراکننده‌ی آن را استخراج کرد. همچنین وضعیت اجرای پرسمان (اجرای موفق یا ناموفق) نیز در این فایل مشخص می‌شود.

---

3 Replication
3 Recovery
7
8

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. در این بخش روشی علاوه بر تهیه‌ی فایل‌های پشتیبان برای بازیابی تغییرات اعمال شده توسط پرسمان‌های DDL تشریح می‌گردد. با توجه به آنکه فایل‌های binary log اعمال DDL را ذخیره می‌کنند با استفاده از محتوای این فایل‌ها می‌توان داده‌های از دست رفته را به دست آورد.

ابتدا، با استفاده از دستوراتی مشابه دستور زیر، فایل‌های binary log به عبارت‌های SQL تبدیل می‌شوند.

```
mysqlbinlog.exe MYSQL_HOME\binary-log\binary-log.000008 > query_log.sql
```

خروجی دستور بالا، در شکل ۲۵ نشان داده شده‌است.

```
'/*!*/;
### INSERT INTO `dbtest`.`test2`
### SET
###   @1=1
### INSERT INTO `dbtest`.`test2`
### SET
###   @1=2
### INSERT INTO `dbtest`.`test2`
### SET
###   @1=30
# at 750
#180227 22:02:44 server id 0  end_log_pos 781 CRC32 0x912564f6  Xid = 12
COMMIT/*!*/;
# at 781
#180227 22:03:00 server id 0  end_log_pos 846 CRC32 0x06c063a1  Anonymous_G
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 846
#180227 22:03:00 server id 0  end_log_pos 968 CRC32 0x7d723be0  Query   thr
SET TIMESTAMP=1519756380/*!*/;
DROP TABLE `test2` /* generated by server */
/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
```

شکل ۲۵: تبدیل binary log به عبارات SQL

همانطور که در شکل ۲۵ دیده می‌شود، می‌توان در فایل‌های binary log عبارت‌های درج را یافت و آن‌ها را اجرا کرد. همچنین می‌توان یک نمونه‌ی جدید از MySQL را راه‌اندازی کرد و فایل‌های SQL ایجاد شده با استفاده از دستور بالا را، از زمان ایجاد جدول تا پیش از دستور DROP یا TRUNCATE، در نمونه‌ی جدید اجرا کرد. بدین ترتیب کل جدول با داده‌هایی که پیش از عمل DDL غیرمجاز/ناخواسته در آن وجود

داشته‌اند، بازیابی می‌شوند. به عنوان مثال، پس از ایجاد فایل query_log.sql مشابه شکل ۲۵، خط مربوط به پرسمان DROP یا TRUNCATE را پاک کرده و سپس دستور زیر اجرا می‌شود تا جدول و محتوای آن بازیابی شود. در اینجا فرض شده است که پرسمان ایجاد جدول و درج محتوای آن، همگی در یک فایل binary log ذخیره شده‌اند.

```
mysql.exe -u root < query_log.sql
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای ترمیم رویداد مربوط به تغییر غیرمجاز شمای پایگاه‌داده را نشان می‌دهد.

ذخیره‌سازی فایل‌های binary log در قالب عبارات SQL
MYSQL_HOME\bin\mysqlbinlog.exe -v BINARY_LOG_PATH\FILE_NAME > query_log.sql
اجرای فایل حاوی عبارات SQL
MYSQL_HOME\bin\mysql.exe -u <USER_NAME> < query_log.sql

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 ۳: تهیه‌ی مستند از فرآیند جرم‌شناسی رویداد تغییر غیرمجاز شمای پایگاه‌داده

تغییر غیرمجاز شمای پایگاه‌داده			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐	رویدادنگاری binary log ☐	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	استفاده از محتوای فایل‌های binary log ☐		
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، سه رویکرد با نام‌های رویدادنگاری general query log، رویدادنگاری binary log و ممیزی با پلاگین audit_log برای جمع‌آوری شواهد و اطلاعات مربوط به جرم معرفی گردید. برای هر یک از این رویکردها، تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و در پایان تهیه مستندات تشریح گردید.

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 یا کلمه‌ی عبور یک نام کاربری 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 برای رویداد تلاش به منظور ورود به پایگاه‌داده را تعیین نماییم. این منابع از طریق پیکربندی رویدادنگاری General query log و همچنین ممیزی مشخص می‌شوند. در ادامه هر یک از گام‌های مربوط به مرحله‌ی جمع‌آوری اطلاعات و شواهد تشریح می‌شوند.

رویدادنگاری General query log: در رویدادنگاری General query log دو نوع اطلاعات ثبت می‌شوند:

- اطلاعات مربوط به اتصال و قطع اتصال کلاینت‌ها
- عبارات SQL دریافت شده از کلاینت‌ها

بنابراین با استفاده از رویدادهای ثبت شده در general log می‌توان اطلاعاتی در مورد تلاش برای یافتن نام کاربری یا تلاش برای یافتن رمز عبور یک نام کاربری مشخص را شناسایی کرد. به‌صورت پیش‌فرض general query log در MySQL غیرفعال است. به‌منظور شناسایی وضعیت general query log از دستور زیر استفاده می‌شود.

```
SHOW VARIABLES LIKE '%general_log%';
```

همچنین با استفاده از دستور زیر می‌توان مقصد خروجی general query log را به دست آورد.

```
SHOW VARIABLES LIKE '%log_output%';
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای جمع‌آوری اطلاعات و شواهد با استفاده از رویدادنگاری General query log را برای رویداد تلاش به منظور ورود غیرمجاز به پایگاه‌داده نشان می‌دهد.

بررسی فعال بودن general query log		
SHOW VARIABLES LIKE '%general_log%';		
فعال کردن general query log **(اعمال تنظیمات در فایل پیکربندی)**		
general_log = on	فعال کردن general query log	۱
log-output=TABLE	ثبت رویدادها در جدول	۲
general_log_file='/tmp/mysql.log' log-output=FILE	ثبت رویدادها در فایل	۳
فعال کردن general query log **(اعمال تنظیمات با دستورات)**		
SET global general_log = 1;	فعال کردن general query log	۱
SET global log_output = 'TABLE';	ثبت رویدادها در جدول	۲
SET global general_log_file='/tmp/mysql.log'; SET global log_output = FILE;	ثبت رویدادها در فایل	۳
برخی دستورات سودمند		
SHOW VARIABLES LIKE '%log_output%';	یافتن مقصد خروجی general query log	۱

ممیزی: نسخه‌ی MySQL Enterprise شامل MySQL Enterprise Audit است که به‌صورت پلاگینی با نام audit_log پیاده‌سازی شده‌است. با استفاده از MySQL Enterprise Audit می‌توان بر اتصالات و

پرسمان‌هایی که بر روی سرور MySQL اجرا می‌شوند، نظارت و رخدادها را ثبت کرد. زمانی که پلاگین نصب شود، پلاگین این امکان را برای سرور MySQL فراهم می‌کند که فایل ثبتی حاوی رکوردهای فعالیت‌های سرور تولید شود. محتوای فایل ثبت شامل اتصال و قطع اتصال کلاینت از سرور و فعالیت‌هایی است که در حین اتصال، کلاینت انجام می‌دهد. فایل ثبت پس از نصب پلاگین در دایرکتوری Data با نام audit.log ایجاد می‌شود. این فایل دارای فرمت XML است و محتوای آن رمز شده نیست. این فایل می‌تواند شامل اطلاعات حساس همچون متن عبارات SQL باشد. برای امنیت بیشتر این فایل باید در دایرکتوری نوشته شود که تنها برای سرور MySQL و کاربران مجاز قابل‌دسترسی باشد. با استفاده از دستور زیر می‌توان وضعیت نصب پلاگین audit_log را بررسی کرد:

```
SELECT PLUGIN_NAME, PLUGIN_STATUS  FROM INFORMATION_SCHEMA.PLUGINS
                                   WHERE PLUGIN_NAME LIKE 'audit%';
```

جدول زیر، خلاصه‌ای از گام‌های مورد نیاز برای جمع‌آوری اطلاعات و شواهد با استفاده از ممیزی را برای رویداد تلاش برای ورود غیرمجاز به پایگاه‌داده نشان می‌دهد.

بررسی وضعیت نصب پلاگین audit_log		
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'audit%';		
نصب پلاگین audit_log		
SHOW VARIABLES LIKE 'version%';	اطمینان از وجود نسخه‌ی MySQL Enterprise	۱
INSTALL PLUGIN audit_log SONAME 'audit_log.dll';	بارگذاری پلاگین و ثبت آن در جدول سیستمی mysql.plugins	۲

۵-۳ استخراج و تجزیه و تحلیل اطلاعات

با استفاده از اطلاعات جمع‌آوری شده, تحلیل‌گر باید تلاش‌های ناموفق برای ورود به پایگاه‌داده را بررسی و در صورت مشاهده‌ی تلاش‌های متوالی در فواصل زمانی کوتاه آن را به عنوان یک جرم تلقی نماید. استخراج و تجزیه و تحلیل اطلاعات برای رویداد ورود غیرمجاز به پایگاه‌داده از دو منبع رویدادنگاری General query log و ممیزی امکان‌پذیر است که در ادامه به تشریح هر یک از آنها می‌پردازیم.

رویدادنگاری General query log: در صورت انتخاب مقدار TABLE برای پارامتر log_output می‌توان رویدادهای ثبت شده را با پرسمان زیر مشاهده کرد (شکل ۲۶):

```
SELECT * FROM mysql.general_log;
```

2018-09-21 09:55:57.632918	root[root] @ localhost [127.0.0.1]	5	0	Connect	Access denied for user 'root'@'localhost' (using password: YES)
2018-09-21 09:56:00.200065	[root] @ localhost [127.0.0.1]	6	0	Connect	root@localhost on using SSL/TLS
2018-09-21 09:56:00.201065	root[root] @ localhost [127.0.0.1]	6	0	Connect	Access denied for user 'root'@'localhost' (using password: YES)
2018-09-21 09:56:02.403191	[root] @ localhost [127.0.0.1]	7	0	Connect	root@localhost on using SSL/TLS
2018-09-21 09:56:02.405191	root[root] @ localhost [127.0.0.1]	7	0	Connect	Access denied for user 'root'@'localhost' (using password: YES)
2018-09-21 09:56:04.497311	[root] @ localhost [127.0.0.1]	8	0	Connect	root@localhost on using SSL/TLS
2018-09-21 09:56:04.498311	root[root] @ localhost [127.0.0.1]	8	0	Connect	Access denied for user 'root'@'localhost' (using password: YES)
2018-09-21 09:56:06.528427	[root] @ localhost [127.0.0.1]	9	0	Connect	root@localhost on using SSL/TLS
2018-09-21 09:56:06.529427	root[root] @ localhost [127.0.0.1]	9	0	Connect	Access denied for user 'root'@'localhost' (using password: YES)
2018-09-21 09:56:29.542743	[admin] @ localhost [127.0.0.1]	10	0	Connect	admin@localhost on using SSL/TLS
2018-09-21 09:56:29.543743	[admin] @ localhost [127.0.0.1]	10	0	Connect	Access denied for user 'admin'@'localhost' (using password: YES)
2018-09-21 09:56:36.966168	[test] @ localhost [127.0.0.1]	11	0	Connect	test@localhost on using SSL/TLS
2018-09-21 09:56:36.967168	[test] @ localhost [127.0.0.1]	11	0	Connect	Access denied for user 'test'@'localhost' (using password: YES)

شکل ۲۶: خروجی general query log

همانطور که در شکل ۲۶ دیده می‌شود، چندین بار با استفاده از نام کاربری root تلاش برای ورود انجام شده‌است که همگی با شکست و خطای access denied پایان یافته‌اند. همچنین تلاش‌هایی برای ورود با نام‌های کاربری مختلف از جمله admin و test نیز صورت گرفته است که همگی در general log ثبت شده‌اند.

مشاهده‌ی رویدادهای general query log **در جدول**
SELECT * FROM mysql.general_log;

ممیزی: به صورت پیش‌فرض در صورت فعال بودن ممیزی، فایل audit.log در دایرکتوری Data ایجاد می‌شود. نمونه‌ای از این فایل در شکل ۲۷ نشان داده شده‌است.

```
<AUDIT_RECORD>
 <TIMESTAMP>2018-09-21T06:26:06 UTC</TIMESTAMP>
 <RECORD_ID>9_2018-09-21T06:25:28</RECORD_ID>
 <NAME>Connect</NAME>
 <CONNECTION_ID>9</CONNECTION_ID>
 <STATUS>1045</STATUS>
 <STATUS_CODE>1</STATUS_CODE>
 <USER>root</USER>
 <OS_LOGIN/>
 <HOST>localhost</HOST>
 <IP>127.0.0.1</IP>
 <COMMAND_CLASS>connect</COMMAND_CLASS>
 <CONNECTION_TYPE>SSL/TLS</CONNECTION_TYPE>
 <PRIV_USER>root</PRIV_USER>
 <PROXY_USER/>
 <DB/>
</AUDIT_RECORD>
```

شکل ۲۷: خروجی ممیزی

همانطور که در شکل ۲۷ دیده می‌شود، با استفاده از اطلاعات ذخیره شده در فایل ممیزی، می‌توان زمان اتصال، نام کاربری و موفقیت یا عدم موفقیت در ورود را به دست آورد.

۵-۴ ترمیم

در صورتی که رویدادهای ثبت‌شده نشان‌دهنده‌ی تلاش برای ورود به پایگاه‌داده باشند، بنابر صلاحدید مدیر پایگاه‌داده می‌توان حساب کاربری را برای مدتی با دستور زیر قفل کرد (شکل ۲۸). بدین ترتیب به نوعی حمله‌ی حدس رمزعبور دشوار و غیرممکن می‌شود.

```
ALTER USER 'admin'@'localhost' ACCOUNT LOCK;
```

```
C:\Users\TestUser\Desktop\mysql enterprise\mysql-advanced-5.7.21-winx64\bin>mysql.exe -u admin --port 3307 --password
Enter password: *******
ERROR 3118 (HY000): Access denied for user 'admin'@'localhost'. Account is locked.
```

شکل ۲۸: قفل کردن یک نام کاربری

همچنین با دستور زیر می‌توان حساب کاربری را از حالت قفل خارج کرد.

```
ALTER USER 'admin'@'localhost' ACCOUNT UNLOCK;
```

پایگاه‌داده MySQL راه‌حل پیش‌فرضی به منظور قفل کردن حساب کاربر پس از تعداد مشخصی تلاش ناموفق برای ورود را ندارد، هرچند می‌توان این نوع عملکرد را با تریگر[۳۹] پیاده‌سازی کرد.

---

[3] Trigger [9]

قفل کردن حساب کاربری به صورت دستی		
ALTER USER '<USER_NAME>'@'<SERVER_ADDRESS>' ACCOUNT LOCK;		
برخی دستورات سودمند		
ALTER USER '<USER_NAME>'@'<SERVER_ADDRESS>' ACCOUNT UNLOCK;	خارج کردن حساب کاربری از حالت قفل	۱

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 ۴: 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	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	ممیزی با پلاگین audit_log ☐		
منابع رویدادنگاری	رویدادنگاری general query log ☐		
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	قفل کردن نام کاربری به صورت دستی ☐		
توضیحات			

۵-۶ 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 رویکردی با نام‌های رویدادنگاری general query log و ممیزی با پلاگین audit_log برای جمع‌آوری شواهد و اطلاعات مربوط به جرم معرفی گردید. در پایان نیز برای هر یک از رویکردهای معرفی شده، تنظیمات مربوط به فعال‌سازی، استخراج اطلاعات و شواهد، تحلیل شواهد، ترمیم و تهیه مستندات تشریح گردید.

۶ خلاصه مطالب

جرم‌شناسی پایگاه‌داده 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 زمانی آغاز می‌شود که از مأمور ممیزی، کشف چگونگی وقوع نقض امنیتی و شخص مجرم درخواست شود. جرم‌شناسی پایگاه‌داده، چالش‌ها و مسائل زیادی به همراه دارد که آن را تبدیل به یک موضوع پیچیده کرده است.

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

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

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

برای هر یک از رویدادهای غیرمجاز در سامانه پایگاه‌داده به هنگام بررسی جرم، جداولی برای ارائه مستندات مورد نیاز 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 زیر، اطلاعات مربوط به تمامی رویدادهای غیرمجاز تشریح شده در مستند حاضر را نشان می‌دهد.

جدول ۵: تهیه‌ی مستند از فرآیند جرم‌شناسی در پایگاه‌داده‌ی MySQL

حذف غیرمجاز محتوای جدول			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐

شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log	☐ رویدادنگاری binary log	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	☐ استفاده از محتوای فایل‌های binary log		
توضیحات			
درج غیرمجاز در جدول			
وقوع جرم	☐ عدم وقوع	☐ وقوع خصمانه	☐ وقوع ناخواسته
شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log	☐ رویدادنگاری binary log	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	☐ حذف سطر(های) درج شده		
توضیحات			
مشاهده غیرمجاز محتوای جدول			
وقوع جرم	☐ عدم وقوع	☐ وقوع خصمانه	☐ وقوع ناخواسته
شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log		
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	فاقد امکان ترمیم است.		
توضیحات			
بروزرسانی غیرمجاز محتوای جدول			
وقوع جرم	☐ عدم وقوع	☐ وقوع خصمانه	☐ وقوع ناخواسته
شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log	☐ رویدادنگاری binary log	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		

امکان ترمیم	☐ استفاده از محتوای فایل‌های binary log		
توضیحات			
تغییر غیرمجاز شمای پایگاه‌داده			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log	☐ رویدادنگاری binary log	
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	☐ استفاده از محتوای فایل‌های binary log		
توضیحات			
تلاش برای یافتن نام کاربری یا کلمه‌ی عبور			
وقوع جرم	عدم وقوع ☐	وقوع خصمانه ☐	وقوع ناخواسته ☐
شیوه ممیزی	☐ ممیزی با پلاگین audit_log		
منابع رویدادنگاری	☐ رویدادنگاری general query log		
شیوه یا ابزار تحلیل	فاقد شیوه یا ابزار تحلیل است.		
امکان ترمیم	☐ قفل کردن نام کاربری به صورت دستی		
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]. https://dev.mysql.com/doc/refman/5.7/en/

[20]. http://kedar.nitty-witty.com/blog/restore-dropped-mysql-database-from-binary-logs