

مشروع أعد لنيل شهادة الهندسة في تخصص أمن النظم والشبكات الحاسوبية (تخرج ١)

Design of IDS for detecting SQL injection in a SIEM system

تصميم نظام كشف تسلل لرصد هجمات SQL ضمن بيئة SIEM

إعداد :

أحمد عبدالرحمن خلوف

طلال محمد الحلو

إشراف :

د . وسيم جنيدى

الملخص :

فهرس المحتويات :

المحتويات

١٠	١.١ المقدمة :
١١	١.٢ الهدف من البحث :
١١	١.٣ التطبيقات العملية :
١٢	١.٤ التحديات :
١٣	الفصل الأول : الدراسات النظرية
١٤	٢.١ المقدمة :
١٥	٢.٢ مفاهيم الأمن السيبراني(من المعلومات) والبني الدفاعية
١٧	٢.٣ هجوم حقن (SQL injection)
١٧	٢.٣.١ تعريف ومبدأ العمل
١٧	٢.٣.٢ تصنیف أنماط حقن SQL
١٨	٢.٣.٣ آثار الهجوم والمخاطر الأمنية
١٨	٢.٣.٤ صعوبة الاكتشاف والدافع لوجود IDS متخصص
١٩	٢.٣.٥ مؤشرات الكشف (IDS) مفيدة (Indicators of Compromise)
١٩	٤.١ أنظمة كشف ومنع التسلل (IDS/IPS) :
١٩	٤.١.١ المفهوم العام
٢٠	٤.١.٢ الأنواع الأساسية لأنظمة الكشف
٢٠	٤.١.٣ آليات الكشف المستخدمة في IDS
٢١	٤.١.٤ المكونات الرئيسية لنظام IDS
٢١	٤.١.٥ مؤشرات أداء النظام والتحديات
٢٢	٤.٢ نظام إدارة معلومات وأحداث الأمن (SIEM)
٢٢	٤.٢.١ المفهوم العام
٢٢	٤.٢.٢ مكونات نظام SIEM
٢٣	٤.٢.٣ آلية عمل SIEM
٢٤	٤.٢.٤ فوائد استخدام SIEM
٢٤	٤.٦ التكامل بين أنظمة IDS و SIEM
٢٤	٤.٦.١ المفهوم العام للتكامل

٢٥	آلية التكامل الفني بين SIEM وIDS ٢.٦.٢
٢٦	الفوائد العملية للتكامل بين SIEM وIDS ٢.٦.٣
٢٧	التحديات المرتبطة بعملية التكامل ٢.٦.٤
٢٩	الفصل الثاني : بيئة العمل
٣٠	المقدمة : ٣.١
٣٠	الأدوات المستخدمة : ٣.٢
٣٠	Wazuh ٣.٢.١
٣٢	Suricata ٣.٢.٢
٣٣	DVWA ٣.٢.٣
٣٣	قاعدة البيانات MySQL (MySQL) ٣.٢.٤
٣٣	بيئة افتراضية (Virtualization) ٣.٢.٦
٣٤	سبب اختيار الأدوات عن منافسيها ٣.٢.٧
٣٥	تصنيف الأدوات ٣.٣
٣٥	Wazuh ٣.٣.١
٣٩	Suricata ٣.٣.٢
٤١	DVWA & MYSQL & APACHE ٣.٣.٣
٤٢	ربط الأدوات ٣.٤
٤٤	الفصل الثالث : الحل المقترن

فهرس الاشكال:

الشكل (١) : أداة تنزيل الشهادات واعداد الملف confi.yaml ٣٦
الشكل (٢) : تعديل قيمة host لنتيشر الى indexer ٣٧
الشكل (٣) : تحديد server host & openserch.host ٣٨
الشكل (٤) : تهيئة IDS ٤٠
الشكل (٥) : إضافة localfile في ملف ossec ٤٠

فهرس الجداول :

٣٤	جدول (١) : الفروقات بين أدوات SIEM
٣٥	جدول (٢) : الفروقات بين أدوات IDS

قائمة المصطلحات :

قائمة الرموز :

١.١ المقدمة :

في عصر التحول الرقمي المتتسارع، أصبحت الأنظمة الحاسوبية تُشكّل العمود الفقري لعمل المؤسسات في مختلف القطاعات. ومع تزايد الاعتماد على الخدمات الإلكترونية والبني التحتية الرقمية، ارتفعت أهمية حماية هذه الأنظمة من التهديدات السيبرانية. إن اختصاص «أمن النظم» يعني بضمان استمرارية عمل الأنظمة الحاسوبية، حماية البيانات من الوصول غير المصرح به، وتأمين الموارد الحساسة ضد الهجمات التي قد تؤدي إلى تعطيل الخدمة أو خسارة المعلومات. في هذا الإطار، تُعدّ أنظمة الكشف عن التسلل (IDS)، ونظم تحليل الأحداث الأمنية (SIEM) من الأدوات الحيوية التي تسهم في تعزيز منظومة الدفاع الأمني للمؤسسات.

من بين أنواع الهجمات السيبرانية التي تستهدف قواعد البيانات وتطبيقات الويب، يبرز هجوم SQL Injection (SQL Injection) كأحد الأخطى والأكثر شيوعاً. يتم هذا الهجوم حين يُدخل المهاجم تعليمات SQL خبيثة في مدخلات المستخدم أو في المتغيرات التي تُركب في استعلامات قاعدة البيانات، ما يسمح له بتجاوز الحماية، استخراج بيانات حساسة، تعديلها أو حذفها. تُشير الأدلة إلى أن ضعف التحقق من المدخلات (input validation)، والاستخدام غير المُحكم لاستعلامات ديناميكية، وعدم تفعيل مبدأ الأقل امتيازاً (least privilege) يزيدان من قابلية التطبيق للتعرض لهذا النوع من الهجمات وبالتالي، فإن التصدي لهجوم حقن SQL يمثل ضرورة أمنية ملحة لحماية البيانات وضمان استقرار الأنظمة. من بين أنواع الهجمات السيبرانية التي تستهدف قواعد البيانات وتطبيقات الويب، يبرز هجوم SQL Injection كأحد الأخطى والأكثر شيوعاً. يتم هذا الهجوم حين يُدخل المهاجم تعليمات SQL خبيثة في مدخلات المستخدم أو في المتغيرات التي تُركب في استعلامات قاعدة البيانات، ما يسمح له بتجاوز الحماية، استخراج بيانات حساسة، تعديلها أو حذفها. وبالتالي، فإن التصدي لهجوم حقن SQL يمثل ضرورة أمنية ملحة لحماية البيانات وضمان استقرار الأنظمة.

على الرغم من توافر أدوات تقنية متعددة لمراقبة الشبكات وتحليل الأحداث، إلا أن العديد من المؤسسات لا تزال تعاني من صعوبة في ربط تبيهات نشاط الشبكة المشتبه به مع الأحداث المخزنة في قواعد البيانات والتطبيقات. بعبارة أخرى: رغم وجود نظام SIEM قادر على جمع وتحليل السجلات، فإن الكشف استباقياً لهجمات الحقن ليس دائماً مهيأً بشكل كافٍ. كما أن تحليل النشاط في الزمن الحقيقي واتخاذ إجراءات تلقائية ما تزال تشکّل تحدياً في البيئات الحقيقية. من هنا تنشأ الحاجة إلى تصميم وحدة IDS مخصصة لاكتشاف حقن SQL داخل بيئة SIEM وتنقل الرابط بين تبيهات قاعدة البيانات ونظام تحليل الأحداث لضمان توقيت استجابة أسرع ودقة أعلى في الكشف.

١.٢ الهدف من البحث :

يهدف هذا البحث إلى تصميم وتنفيذ نظام كشف تسلل (SQL Injection) مخصصاً لاكتشاف (IDS) ضمن بيئة تحليلاً أحداث أمنية (SIEM) في الزمن الحقيقي، بحيث يعزز قدرة المؤسسة على اكتشاف محاولات التسلل والتلاعب بقاعدة البيانات، وربط التطبيقات المستخلصة من الشبكة والتطبيقات بقواعد بيانات المؤسسة لتحليل مركزي وتحقيق استجابة أوتوماتيكية.

من جهة، يعالج البحث الفجوة التي تفيد بأنه رغم وجود منصات IDS وأنظمة SIEM منفصلة، إلا أن الرابط بينها لا يزال محدوداً في كثير من الحالات، مما يقلل من قدرة الرصد الاستباقي للتهديدات. ومن جهة أخرى، يستند البحث إلى التطورات الحديثة في الكشف مثل استخدام تحليلاً لأنماط حركة البيانات الصادرة

(outbound traffic) لاكتشاف هجمات حقن SQL بدقة عالية، كما في دراسة قدمت نسبة دقة بلغت أكثر من ٩٨٪ باستخدام هذه الطريقة. [1]

كما يراعي البحث التطورات في أساليب الهجوم المعقدة والمتحيرة بسرعة، والتي تتجاوز الأساليب التقليدية القائمة على قواعد ثابتة فقط، ويقترح دمج تقنيات مثل التعلم الآلي أو تحليلاً لأنماط المتقدمة لزيادة فعالية الكشف وتقليل التطبيقات الكاذبة.

بالتالي، النتائج المنتظرة لهذا المشروع تشمل: تحسين دقة كشف هجمات حقن SQL ، تقليل زمن الاستجابة للحوادث، وتوفير بيئة اختبار قابلة للتطبيق العملي في سياقات المؤسسات.

١.٣ التطبيقات العملية :

يمكن لهذا النظام أن يطبق عملياً في عدة بيئة ومجالات تشغّل بنية تحتية تعتمد على قواعد بيانات وتطبيقات ويب، مثل المؤسسات المالية، ومرکز البيانات، وشركات التجارة الإلكترونية، أو حتى في المراكز الأمنية (SOC) على سبيل المثال:

في مؤسسة تستخدم قاعدة بيانات خلفية لتطبيق الويب، يمكن لـ IDS أن يراقب الاستعلامات الواردة والصادرة، ويرسل التبيهات إلى منصة SIEM عند رصد نمط يشير إلى حقن SQL.

يمكن أيضاً دمج النظام مباشرة مع لوحة عرض (Dashboard) في SIEM تعرض مؤشرات مثل: عدد محاولات الحقن خلال آخر ٢٤ ساعة، عناوين IP المكررة، أو الحقول التي تم استهدافها.

إضافة إلى ذلك، يمكن تنفيذ استجابة تلقائية، مثل: حظر عنوان IP المهاجم، تعطيل حساب المستخدم المشبوه، أو جلب تقرير تلقائي لمشرف الأمان.

أيضاً،

هذا النظام قابل للتوسيعة ليعمل في بيئة متعددة الأجهزة أو متعددة الشبكات (Distributed Deployment)، حيث تجمع بيانات IDS متعددة في منصة SIEM واحدة لتحليل مركزي، مما يعزّز قدرة المؤسسة على التعامل مع هجمات موزّعة أو تنسيقية. دراسة حديثة تشير إلى إمكانية استخدام حركة البيانات الصادرة (outbound traffic) كعنصر رئيسي في الكشف، ما يجعل التطبيق العملي أكثر فعالية حتى في بيئات الشركات.

٤.١ التحديات :

تنفيذ مثل هذا النظام يواجه عدة تحديات تقنية ومنهجية، ومنها:

١. التفوغ والتطور في أنماط هجمات SQL Injection: هجمات الحقن لا تقتصر على نمط محدد، بل تتضمن أنواعاً مثل Blind، Time-Based، In-Band، أو حتى حقن عبر بروتوكولات غير مباشرة، ما يجعل بناء قاعدة توقيع (signature) واحدة فعالة غير كافية.
٢. الدمج والتكامل مع منصة SIEM: ليس كافياً أن يكتشف الهجوم داخل IDS فقط، بل يجب أن يُنقل التبيه إلى منصة SIEM، تحلّ ارتباطه مع أحداث أخرى (مثل تسجيل دخول فاشر، تغييرات قاعدة البيانات، حركة مشبوهة) وتتخذ إجراءً. هذا يتطلّب تنسيقاً بين الأدوات، وضبط بروتوكولات الإرسال، وتوحيد تنسيق السجلات (logs).
٣. تقليل التبيهات الكاذبة وتحسين صيانة النظام: قواعد الكشف تحتاج إلى تحديث مستمر لمتابعة الهجمات الجديدة، كما أن التبيهات الكاذبة إن ارتفعت بفعل قرارات النظر غير الدقيقة، ستقود إلى تجاهل التبيهات الحقيقية. دراسة عن تصميم التوقيعات بـIDS أظهرت أن ضعف التوقيعات يُضعف كفاءة النظام بشدة.

الفصل الأول : الدراسات النظرية

٢.١ المقدمة :

في العصر الرقمي الحديث أصبحت الأنظمة المعلوماتية تمثل العمود الفقري لمختلف المؤسسات الحكومية والخاصة. وتعتمد جميع عملياتها اليومية - من المعاملات المالية إلى إدارة البيانات الحساسة - على بنية تحتية متربطة من الخوادم والشبكات وقواعد البيانات. ومع ازدياد هذا الاعتماد، تصاعدت في المقابل التهديدات السيبرانية التي تستهدف استغلال الثغرات التقنية والبشرية لتحقيق مكاسب غير مشروعة أو تعطيل الخدمات الحيوية.

هنا يبرز مفهوم أمن النظم والمعلومات (Information and System Security) بوصفه علمًا يهدف إلى حماية أصول المعلومات والأنظمة التي تعالجها من أي خطر يمكن أن يؤثر على سريتها أو سلامتها أو توافرها. ويعرف المعهد القومي للمعايير والتكنولوجيا (NIST) أمن المعلومات بأنه: "حماية نظم المعلومات وبياناتها من الوصول أو الاستخدام أو الإفصاح أو التعديل أو التعطيل أو التدمير غير المصرح به".

يرتكز أمن المعلومات على ما يُعرف بـ مثلث CIA (Confidentiality – Integrity – Availability) ، وهو النموذج الذي يشكل الأساس لجميع السياسات الأمنية المعتمدة في المؤسسات:

١. السرية (Confidentiality): ضمان عدم وصول البيانات إلا للمستخدمين المصرح لهم بذلك.
٢. السلامة (Integrity): التأكد من عدم تعديل البيانات أو إتلافها أثناء التخزين أو النقل.
٣. التوافر (Availability): التأكد من بقاء الأنظمة والبيانات متاحة عند الحاجة إليها دون تعطيل.

ولتحقيق هذه الأهداف تعتمد المؤسسات على منهجية الدفاع المتعدد الطبقات (Defense in Depth) ، والتي تقوم على بناء طبقات حماية متتابعة تشمل جدران الحماية (Firewalls) ، أنظمة كشف التسلل (IDS/IPS) ، أنظمة إدارة الأحداث الأمنية (SIEM) ، وأنظمة التحكم في الوصول (Access Control) . والهدف من ذلك هو إلا يشكل اختراق طبقة واحدة انهياراً للنظام بالكامل.

من جانب آخر، تتتنوع التهديدات الأمنية ما بين برمجيات خبيثة (Malware) ، وهجمات الحرمان من الخدمة (DoS / DDoS) ، ومحاولات التصيد (Phishing) ، وحقن الأكواكاد في التطبيقات (Code Injection) بما فيها هجمات حقن SQL التي تُعد من أخطرها على الإطلاق. وتشير تقارير IBM X-Force Threat Intelligence إلى أنَّ أكثر من ٦٠ % من الثغرات المسجلة ترتبط بسوء تأمين تطبيقات الويب أو ضعف آليات التحقق من إدخال المستخدمين.

تُظهر هذه المعطيات أنَّ أمن النظم لم يعد خياراً إضافياً بل أصبح ضرورة استراتيجية لضمان استمرارية الأعمال وثقة المستخدمين. وللهذا طرحت المنظمات الدولية مثل ISO و OWASP أطرًا ومعايير إجرائية تهدف إلى إنشاء سياسات أمنية متكاملة، تشمل تحديد المخاطر (Risk Assessment) ، وتطبيق الضوابط التقنية (Incident Response) ، والتعامل مع الحوادث الأمنية. (Security Controls)

ضمن هذا السياق يأتي مشروعنا ، الذي يستند إلى أحد أهم مفاهيم الحماية وهو الكشف المبكر عن التسلل (Intrusion Detection) ، وذلك من خلال تصميم نظام IDS قادر على اكتشاف محاولات حرق SQL وتحليلها بالتكامل مع منصة SIEM لتقديم رؤية شاملة للأنشطة الأمنية داخل البيئة المستهدفة.

٢.٢ مفاهيم الأمن السيبراني(أمن المعلومات) والبنيّة الدفاعية

يُعدّ الأمن السيبراني(Cybersecurity) أو أمن المعلومات بصيغة أكبر الإطار الأشمل الذي يندرج ضمنه أمن النظم والمعلومات، ويعني بحماية الأنظمة الرقمية من الهجمات التي تستهدف سرقة المعلومات أو تعطيل العمليات أو إتلاف البنية التحتية التقنية. يُعرفه NIST بأنه "مجموعة من الأنشطة والضوابط المصممة لحماية نظم المعلومات من التهديدات الرقمية وضمان استمرارية العمليات". ويشمل هذا المجال تحليل المخاطر، والاكتشاف المبكر للثغرات، والاستجابة الفورية للحوادث، والتعافي من آثار الهجوم.

تقوم البنية الأمنية في المؤسسات عادةً على مبدأ يُعرف باسم الأمن الطيفي (Layered Security) أو ما يُسمى Defense in Depth، وهو مفهوم أساسي في تصميم الأنظمة الحديثة. هذا المبدأ يقوم على فكرة أن النظام الآمن يجب أن يحتوي على عدة طبقات من الحماية تعمل بشكل متكامل، بحيث يُعيق كل مستوى محاولات الاختراق قبل أن تصل إلى البيانات الحساسة. من هذه الطبقات:

- طبقة المحيطية: (Perimeter Security) وتشمل جدران الحماية (Firewalls) وأنظمة كشف ومنع التسلل.(IDS/IPS).
- طبقة التطبيقات: (Application Security) وتشمل تأمين كود التطبيق ضد الثغرات مثل-Cross-Site scripting (XSS).
- طبقة البيانات: (Data Security) ، وتشمل التشفير (Encryption) ، وإدارة الوصول (Access Management).
- طبقة المراقبة والتحليل: (Monitoring & Analytics) وتشمل أنظمة إدارة الأحداث الأمنية التي تقوم بدمج وتحليل البيانات الأمنية من جميع الطبقات السابقة.

يُضاف إلى ذلك مبدأ Zero Trust Architecture الذي يعتمد على "عدم الثقة الافتراضية" ، أي عدم افتراض أن أي كيان داخل الشبكة آمن تلقائياً، بل يجب التحقق من هوية وصلاحيات كل مستخدم أو تطبيق في كل خطوة.

وقد أصبحت هذه الفلسفة من ركائز الأمن المؤسسي الحديثة وفقاً لتوصيات NIST SP 800-207.

من جهة أخرى، تصنف الهجمات السيبرانية عادةً ضمن ثلاث فئات رئيسية وفق الهدف من الهجوم:

هجمات سرية Confidentiality Attacks مثل سرقة كلمات المرور أو البيانات الحساسة.

هجمات تكامل Integrity Attacks مثل تعديل البيانات أو التلاعب في قواعد البيانات.

هجمات التوافر Availability Attacks مثل الحرمان من الخدمة (DoS/DDoS).

ويُعتبر حقن SQL من أخطر أنواع هجمات التكامل، لأنّه يتيح للمهاجم تغيير منطق استعلامات قاعدة البيانات

عبر إدخال تعليمات SQL غير متوقعة، ما يؤدي إلى تسريب أو تعديل أو حذف بيانات أساسية.

لذلك فإن فهم طبيعة هذه الهجمات يعد أساسياً لتصميم آليات الكشف المناسبة.

أما من حيث مبادئ التصميم الآمن للأنظمة (Secure System Design Principles)، فهناك مجموعة من

القواعد التي وضعتها منظمة OWASP و Saltzer & Schroeder، وهي تشكّل الإطار النظري لأي نظام

آمن. من أهمها:

- مبدأ الأقل امتيازاً: (Least Privilege) إعطاء المستخدم أو العملية أقل قدر ممكّن من الصلاحيات اللازمة لأداء المهام.
- الدفاع بالعمق: (Defense in Depth) عدم الاعتماد على وسيلة حماية واحدة.
- الفصل بين الواجبات: (Separation of Duties) توزيع المهام الأمنية بين مكونات مختلفة لتقليل الأخطاء البشرية.
- الفشل الآمن: (Fail-Safe Defaults) في حال فشل النظام يجب أن يبقى في وضع آمن.
- قابلية المراجعة: (Accountability) ضرورة تسجيل الأحداث والأنشطة لمتابعة أي محاولة خرق.

تؤكد هذه المبادئ على أنّ الأمان ليس مجرد تركيب أدوات مثل SIEM أو IDS ، بل هو منهجية متكاملة تبدأ

من مرحلة التصميم البرمجي حتى التشغيل والمراقبة.

فالنظام الذي يفتقر لتصميم آمن سيكون عرضة للاختراق حتى لو استخدم أحدث الأدوات.

ولهذا يركّز مشروعنا على الجمع بين الكشف التقني عبر IDS والتحليل الذكي عبر SIEM في إطار متكامل،

ما يجسد عملياً مبدأ ”الدفاع المتعدد الطبقات“.

إن فهم هذه الأسس النظرية يمنح الباحث القدرة على تفسير سلوك الأنظمة الأمنية وتحليل التبيّنات الناتجة بدقة

أكبر. كما يساعد على ربط المكونات التقنية — مثل snort أو Wazuh — ضمن فلسفة أمنية متكاملة، وهو

ما سيُبني عليه التصميم العملي في الفصول التالية.

٢.٣ هجوم حقن SQL (SQL injection)

٢.٣.١ تعريف وبدأ العمل

حقن SQL (SQL Injection) هو أسلوب هجوم يستغل ضعف التحقق من مدخلات المستخدم في تطبيقات الويب التي تقوم بتكوين استعلامات (Queries) إلى قاعدة بيانات بطريقة ديناميكية. بدلاً من أن تتعامل المدخلات كبيانات فقط، يتم دمجها في جملة SQL بما يغير منطق الاستعلام، مما يمكن المهاجم من تنفيذ أوامر غير مرغوبة على قاعدة البيانات، مثل استخراج بيانات حساسة أو تعديلها أو حذفها أو تجاوز آليات المصادقة.

آلية الهجوم الأساسية تمر عبر ثلات مراحل عملية ضمن دورة الهجوم: (Attack Lifecycle)

١. الاستطلاع (Reconnaissance): المهاجم يحدد نقاط الإدخال (نماذج، وسائل HTTP، رؤوس cookies) التي تكون استعلامات لقاعدة البيانات.

٢. الاستغلال (Exploitation): إدخال سلسلة حقن (payload) تغير من بنية الاستعلام، مثل تحويل WHERE clause إلى دائمًا صحيح.

٣. الاستخراج/التحكم (Exfiltration/Control): استغلال النتائج لإظهار بيانات، تنفيذ أوامر تخريبية، أو الحصول على وصول أكبر داخل النظام.

٢.٣.٢ تصنيف أنماط حقن SQL

تختلف أساليب حقن SQL تبعاً لطريقة حصول المهاجم على استجابة من الخادم، ومن أشهر الأنواع:

In-Band SQLi (Union-based / Error-based) 

المهاجم يحصل على بيانات مباشرة من استجابة الخادم.

Union-based يستخدم UNION SELECT لمج نتائج استعلام خبيث مع استجابة التطبيق.

Blind SQLi (Boolean-based / Time-based) 

المهاجم لا يرى البيانات مباشرة؛ يعتمد على استجابات منطقية أو زمنية.

Boolean-based (content-based) يحقن شرطاً يؤدي إلى اختلاف الاستجابة (صفحة مختلفة أو وجود عنصر) للدلالة على قيمة بait أو بت.

Time-based يحقن استعلاماً يُجبر قاعدة البيانات على الانتظار مدة محددة (SLEEP())، وبناءً على استجابة الوقت يستنتج قيمة.

Out-of-Band (OOB) SQLi

يحدث عندما لا تُعاد النتائج مباشرة عبر القناة الحالية؛ يُستخدم قناة خارجية (مثل DNS أو HTTP callback) لجلب بيانات من الخادم. أقل شيوعاً لكنه فعال في بيئات محددة.

٢٠٣٠٣ آثار الهجوم والمخاطر الأمنية

هجمات حقن SQL قد تؤدي إلى نتائج خطيرة مؤسسيًا وتقنيًا، منها:

- انتهاك سرية البيانات (Data Breach): تسريب قواعد بيانات كاملة تشمل معلومات شخصية وحساسة.
- فقدان تكامل البيانات: تعديل أو حذف سجلات حرجية يؤدي إلى فقدان موثوقية النظام.
- تجاوز المصادقة (Authentication Bypass): الدخول بحسابات إدارية دون معرفة كلمات المرور.
- الإضرار بالتوافر (Denial of Service): استغلال أوامر ثقيلة تسبب بطء أو تعطل قاعدة البيانات.
- نفذ حدود النظام (Privilege Escalation) والتقلل الجانبي: في حالات متقدمة يمكن للمهاجم تنفيذ أوامر تؤدي إلى اختراق طبقات أخرى داخل الشبكة.

٢٠٣٠٤ صعوبة الاكتشاف والدلوافع لوجود IDS متخصص

حقن SQL يصعب اكتشافه أحياناً للأسباب التالية:

- تدخل مع الاستخدام الشرعي: بعض المدخلات الشرعية قد تشارك بسمات نحوية تشبه هجمات (مثلاً نصوص البحث).
- هجمات Slow–Rate و Blended (time-based blind) الهجمات البطيئة لا تولد كمية كبيرة من الأحداث في وقت قصير، لذا يمكنها التسلل دون إثارة إنذارات عابرة.
- تنويع القنوات: الهجوم قد يتم عبر REST APIs ، رؤوس HTTP ، أو حتى بيانات مخزنة داخل ملفات JSON ، ما يزيد من احتمالات التخفي.
- التعلم من قواعد التوقيع التقليدية: التوقيعات الثابتة قد تتجاوزها صيغ حقن جديدة أو تحويلات (encoding/obfuscation) تشغيل/ترميز.

لهذا تُصبح الحاجة إلى نظام كشف تسلل مخصص (IDS) قادر على رصد أنماط SQLi أمراً محورياً – سواء عبر قواعد توقع متقدمة تطابق أدق أشكال payloads ، أو عبر الكشف بالسلوك (anomaly detection) الذي يراقب أنماط الاستعلامات إلى قاعدة البيانات ويكشف الانحرافات.

٢.٣.٥ مؤشرات الكشف (Indicators of Compromise) مفيدة لـIDS

عند تصميم IDS لاكتشاف SQL injection ، من المهم مراقبة مؤشرات محددة مثل: وجود كلمات مفاتيحية في مدخلات المستخدم.(UNION, SELECT, --, /*, SLEEP, OR 1=1) زيادة مفاجئة في استجابات الأخطاء من قاعدة البيانات. استعلامات طويلة غير اعتيادية أو استدعاءات متكررة لـ information_schema. تغير في زمن الاستجابة (Time-based payloads). نشاطات خارجية مثل طلبات لإرسال استجابات عبر استدعاءات خارجية.(OOB)

٤. أنظمة كشف ومنع التسلل (IDS/IPS) :

٤.١ المفهوم العام

تُعد أنظمة كشف التسلل (IDS – Intrusion Detection Systems) من الركائز الأساسية في البنية الدفاعية لأي مؤسسة رقمية. يقصد بـ "التسلل" أي نشاط غير مصرح به يسعى إلى انتهاك سياسات الأمان أو استغلال الثغرات التقنية. وظيفة IDS هي مراقبة حركة الشبكة والأنشطة الداخلية وتحليلها لاكتشاف أي سلوك يُحتمل أن يكون خبيثاً، وإرسال تتبّيه إلى فرق الأمان عند حدوثه. تُكمل هذه الأنظمة عمل جدران الحماية (Firewalls) ، في بينما يتحكم الجدار الناري في حركة المرور المسموح بها أو المحظورة، يقوم IDS بمراقبة ما يحدث داخل هذا المرور لاكتشاف أي نشاط ضار قد يتجاوز السياسات المعلنة.

عند تطور الحاجة من الكشف إلى الاستجابة التلقائية ظهر نوع آخر يُعرف باسم نظام منع التسلل (Intrusion Prevention System – IPS) ، والذي لا يكتفي بإصدار تتبّيه بل يقوم أيضًا باتخاذ إجراء فوري مثل حظر الاتصال أو إسقاط الحزمة المشبوهة أو عزل المضيف.

٢.٤.٢ الأنواع الأساسية لأنظمة الكشف

تنقسم أنظمة IDS حسب موقع المراقبة إلى نوعين رئيسيين:

١. أنظمة كشف التسلل الشبكية (Network-based IDS – NIDS):

تُنصَب عادة على نقاط استراتيجية في الشبكة وتقوم بمراقبة حركة المرور المارة عبر الشبكة في الزمن الحقيقي.

أمثلة: Snort ، Suricata

تمتاز NIDS بقدرتها على مراقبة نطاق واسع لكنها محدودة في تحليل الأنشطة الداخلية على الأجهزة المضيفة.

٢. أنظمة كشف التسلل المستندة إلى المضيف (Host-based IDS – HIDS):

تُثبت على الأجهزة نفسها لمراقبة سلوكيات النظام مثل ملفات السجل (Logs) ، عمليات التشغيل، سلامة الملفات (File Integrity).

أمثلة: OSSEC ، Wazuh

تمتاز بقدرتها على اكتشاف التهديدات الداخلية أو تغييرات النظام، لكنها تتطلب موارد أكبر.

٢.٤.٣ آليات الكشف المستخدمة في IDS

تعمل أنظمة IDS وفق منهجين رئيسيين للكشف:

١. الكشف بالتوقيعات (Signature-based Detection):

يعتمد على قاعدة بيانات تحتوي على أنماط (Patterns) أو توقيع للهجمات المعروفة (مثلاً سلاسل نصية، تعليمات معينة، أو سلوك محدد).

عندما تتطابق حركة مرور أو طلب معين مع أحد هذه التوقيعات، يتم إطلاق تبيه.

- الميزة: دقة عالية في اكتشاف الهجمات المعروفة.

- العيوب: غير فعالة أمام هجمات جديدة أو معدلة. (Zero-Day Attacks)

٢. الكشف بالشذوذ (Anomaly-based Detection):

يقوم النظام أولاً بتعلم النمط الطبيعي لسلوك الشبكة أو التطبيق (Baseline Behavior)، ثم يُصدر تنبئها عند ملاحظة أي انحراف كبير عن هذا النمط.

يمكن أن يعتمد على خوارزميات إحصائية أو تعلم آلي.

- الميزة: إمكانية كشف هجمات غير معروفة مسبقاً.

- العيوب: احتمال مرتفع للإنذارات الكاذبة. (False Positives)

وفي بعض الأنظمة الحديثة يُدمج المنهجان ضمن ما يُعرف بـ Hybrid IDS لتحقيق توازن بين الدقة والتكيّف مع التهديدات الجديدة.

٤.٤.٤ المكونات الرئيسية لنظام IDS

يتكون أي نظام IDS فعال من مجموعة من الوحدات الأساسية:

١. وحدة الالتقط (Data Collection): تلتقط حركة المرور أو سجلات النظام من الشبكة أو المضيف.
٢. وحدة التحليل (Analysis Engine): تفسر البيانات باستخدام قواعد أو خوارزميات للكشف.
٣. وحدة قاعدة المعارف (Knowledge Base): تخزن تفاصيل الهجمات أو ملفات الأنماط.
٤. وحدة التبيه والتقارير (Alerting & Reporting): تُرسل تبليغات إلى مدير الأمان أو إلى نظام SIEM.
٥. واجهة الإدارة (Management Console): تُستخدم لتكوين النظام وإدارة القواعد ومراجعة التبليغات.

٤.٤.٥ مؤشرات أداء النظام والتحديات

أداء أنظمة IDS يقاس بعدها معايير:

- معدل الكشف الحقيقي (True Positive Rate).
- معدل الإنذارات الكاذبة (False Positive Rate).
- الزمن اللازم للكشف (Detection Latency).
- معدل المعالجة (Throughput).

التحديات الرئيسية تشمل:

- صعوبة مواكبة أنماط الهجمات الجديدة.
- استهلاك الموارد الحاسوبية عند تحليل كم كبير من البيانات.
- التعامل مع التشفير (Encrypted Traffic) الذي يعيق التحليل العميق للحزم (Deep Packet Inspection).
- الحاجة إلى تكامل مع أنظمة SIEM لتجميع وتحليل الأحداث من مصادر متعددة بشكل مركزي.

٢.٥ نظام إدارة معلومات وأحداث الأمن (SIEM)

٢.٥.١ المفهوم العام

يُعتبر نظام إدارة معلومات وأحداث الأمن (SIEM) أحد أهم أدوات التحليل والمراقبة في مراكز العمليات الأمنية (Security Operations Centers – SOC).

ظهر مفهوم SIEM نتيجة الحاجة إلى دمج وظائف نظمتين سابقين:

• SIM (Security Information Management):
يهم بتخزين وتحليل البيانات الأمنية التاريخية على المدى الطويل.

• SEM (Security Event Management):
يركز على تحليل الأحداث الفورية وإطلاق التبليغات في الزمن الحقيقي.

ومن هنا جاء نظام SIEM كحل متوازن يجمع بين تحليل البيانات التاريخية والتحليل الفوري، ليمثل فرق الأمن رؤية مركبة شاملة لجميع الأنشطة الأمنية في بيئة المؤسسة.

يعمل نظام SIEM على جمع وتحليل وربط الأحداث الأمنية الصادرة من مختلف الأجهزة والتطبيقات مثل الجدران النارية، أنظمة كشف التسلل IDS ، الخوادم، قواعد البيانات، والتطبيقات السحابية. من خلال ذلك، يمكن لـ SIEM تحديد الأنماط المريبة والتصرفات غير المعتادة وربطها ببعضها لتوليد إنذار دقيق يدل على حادث أمني فعلي.

٢.٥.٢ مكونات نظام SIEM

يتكون أي نظام SIEM فعال من مجموعة من المكونات المتربطة، يمكن تلخيصها فيما يلي:

١. جامع البيانات (Data Collectors / Agents):

مسؤول عن جمع السجلات (Logs) والأحداث من مصادر متعددة وإرسالها إلى النظام центральный للتحليل.

تُستخدم بروتوكولات مثل Syslog ، Winlogbeat ، Filebeat ، و API Integrations.

٢. قاعدة بيانات الأحداث (Event Repository):

تخزن جميع السجلات الواردة بطريقة منتظمة لتسهيل التحليل والاسترجاع.

يجب أن تكون هذه القاعدة قابلة للتتوسيع (Scalable) وتتوفر فهرسة سريعة.

٣. محرك التحليل (Correlation & Analytics Engine):

القلب الحقيقي للنظام، حيث يتم تحليل الأحداث باستخدام قواعد ارتباط (Correlation Rules) أو خوارزميات تعلم آلي لتحديد العلاقات بين الأحداث من مصادر مختلفة.

مثلاً: محاولة تسجيل دخول فاشلة تابعها استعلامات قاعدة بيانات غير طبيعية قد تشير إلى هجوم SQL Injection.

٤. وحدة التبيه والاستجابة: (Alerting & Response Module)
تُصدر تنبیهات فوریة عند اكتشاف تهديد، ويمكن أن تُنفذ إجراءات تلقائية مثل حظر عنوان IP ، تعطيل مستخدم، إرسال بريد فوري إلى الفريق الأمني.)
٥. لوحات المراقبة والتقارير: (Dashboards & Reporting)
تتيح عرض الأحداث في شكل رسومي وتحليلي لفهم الاتجاهات الأمنية، وتقدم تقارير دورية للإدارة.
٦. نظام إدارة المستخدمين والصلاحيات:
لضمان أن الوصول إلى بيانات SIEM يتم فقط من قبل الموظفين المصرح لهم، وفق مبدأ الأقل امتيازاً.(Least Privilege).

٢.٥.٣ آلية عمل SIEM

- تمر دورة العمل داخل نظام SIEM بعدة مراحل متابعة كما هو موضح أدناه:
١. جمع البيانات: (Data Collection)
استلام السجلات من الأجهزة المختلفة عبر الوكاء (Agents) أو بروتوكولات النقل.
 ٢. تطبيق البيانات: (Normalization)
تحويل السجلات من تنسيقات مختلفة إلى نموذج موحد (Common Event Format – CEF) ليسهل تحليلها.
 ٣. التحليل والترابط: (Correlation & Analysis)
تطبيق قواعد منطقية لتحديد تسلسل أحداث غير طبيعي أو مريب.
مثال: “إذا تم تسجيل دخول غير ناجح أكثر من ٥ مرات في دقيقة، تلاه تنفيذ استعلام SQL غريب → إصدار تنبية.”
 ٤. التخزين والفهرسة: (Storage & Indexing)
حفظ البيانات وتحليلها لاحقاً في التحقيقات. (Forensics)
 ٥. التبيه والاستجابة: (Alerting & Incident Response)
إرسال تنبیهات عبر البريد الإلكتروني، أو إلى أنظمة الاستجابة التلقائية. (SOAR)

٤.٥.٢ فوائد استخدام SIEM

١. الرؤية المركزية للأمن (Centralized Visibility):
يمنح الإدارة الأمنية رؤية موحدة لجميع الأحداث داخل الشبكة.
٢. تحسين سرعة الاستجابة (Incident Response):
بفضل التبيهات الآلية ودمج الأحداث المتعددة في سياق واحد، يمكن للفريق الأمني الاستجابة بسرعة للحوادث.
٣. الكشف عن التهديدات المركبة (Complex Threats):
من خلال correlation rules يمكن كشف الهجمات متعددة المراحل مثل SQL Injection من خلال correlation rules بسرعة بيانات.
٤. تحليل الاتجاهات (Trend Analysis):
يساعد في اكتشاف أنماط تكرار الهجمات لتطوير سياسات دفاعية جديدة.
٥. الامتثال للمعايير (Compliance):
أنظمة SIEM تساعد المؤسسات على الامتثال لمعايير مثل ISO 27001 و PCI DSS و GDPR من خلال التوثيق الدقيق للأحداث.

٤.٦ التكامل بين أنظمة IDS و SIEM

٤.٦.١ المفهوم العام للتكميل

يُعد التكامل بين أنظمة كشف التسلل (IDS) وأنظمة إدارة معلومات وأحداث الأمن (SIEM) أحد أهم ممارسات الأمن السيبراني الحديثة، إذ يهدف إلى الانتقال من الكشف المنعزل إلى المراقبة والتحليل الشامل متعدد الطبقات.

بينما يعمل نظام IDS على اكتشاف الأنشطة المشبوهة في الشبكة أو المضيف بشكل فوري، فإن SIEM يمثل الطبقة التحليلية التي تجمع وتربط هذه التبيهات مع أحداث أخرى من مصادر مختلفة. بذلك يتحول الحدث من مجرد "إنذار محلي" إلى "حادثة أمنية موثقة" يمكن تتبعها وتحليلها ضمن سياقها الكامل.

في بيئه متكاملة، يصبح IDS بمثابة مستشعر أمني (Sensor)، بينما يعتبر SIEM العقل التحليلي المركزي (Analytical Brain).

هذا التكامل يمكن المؤسسة من الانتقال من الكشف السلبي (Passive Detection) إلى الاستجابة الاستباقية (Proactive Response).

٢.٦ آليّة التكامل الفني بين SIEM وIDS

تتم عملية الدمج عبر سلسلة من الخطوات التقنية المدروسة، تختلف حسب الأدوات المستخدمة ولكنها تتبع غالباً نفس المفهوم العام:

١. توليد الحدث: (Event Generation)

يقوم IDS بتحليل حركة المرور ورصد نشاط مريب (مثل محاولة SQL Injection) ، ثم يُنشئ سجلاً تفصيلياً (Alert Log) يحتوي على بيانات الحدث:

- عنوان الـ IP المصدر.
- المنفذ المستهدف.
- نوع التوقيع الذي تم مطابقته (Signature ID).
- التاريخ والوقت.
- الرسالة التوضيحية للهجوم.

٢. نقل الحدث إلى: (Event Forwarding)

يتم إرسال هذه السجلات إلى خادم SIEM باستخدام بروتوكولات قياسية مثل Syslog أو JSON API.

في الأنظمة الحديثة مثل Suricata + Wazuh ، يتم ذلك آلياً عبر وحدة تكامل مخصصة تتيح إرسال التطبيقات لحظة وقوعها.

٣. تطبيق البيانات: (Normalization)

عند وصول الحدث إلى SIEM ، يتم تحويله إلى تنسيق موحد (Unified Event Schema) ، مما يسمح بدمج الأحداث القادمة من مصادر مختلفة (مثل IDS و Firewall و Database Logs) في قاعدة بيانات واحدة.

٤. التحليل والترابط: (Correlation Analysis)

يُطبق SIEM قواعد ترابط (Correlation Rules) للبحث عن علاقات منطقية بين الأحداث. مثل عملي:

- حدث من IDS يشير إلى محاولة SQL Injection من IP محدد.
- حدث من Web Server يُظهر استعلامات غير اعتيادية في نفس الدقيقة من نفس الـ IP.
- النتيجة SIEM: يُصدر تبيّناً مركّباً يشير إلى "هجوم مؤكّد".

٥. الاستجابة والتبيه (Alerting & Response):

يتم إرسال تبليغ موحد للفريق الأمني، ويمكن تفعيل استجابة تلقائية مثل حظر العنوان المهاجم عبر الجدار الناري (Firewall Integration) أو تعطيل الجلسة النشطة.

٢٦.٣ الفوائد العملية للتكامل بين SIEM وIDS

١. تحسين دقة الكشف وتقليل الإنذارات الكاذبة (False Positives):

عندما يدمج SIEM تنبؤات IDS مع مصادر أخرى، يمكنه التحقق من مدى مصدقتها قبل إطلاق إنذار حقيقي.

مثلاً، إذا كشف IDS محاولة SQL Injection لكن لم تُسجل أي استعلامات غريبة في قاعدة البيانات، قد يعتبر SIEM الحدث غير مؤكّد ويقلل الإزعاج للفريق الأمني.

٢. رؤية موحدة للأحداث (Unified Visibility):

في بيئة متعددة الأنظمة، قد يصعب على المهندسين تتبع مصدر الهجوم. التكامل يسمح بعرض جميع الأحداث ذات الصلة في لوحة واحدة، مما يسهل عملية التحليل والتحقيق.

٣. التحليل الزمني والتسلسلي (Temporal Correlation):

يمكن لـ SIEM أن يربط بين سلسلة من الأحداث على مدار الوقت (مثل محاولات تسجيل دخول متكررة ثم حقن SQL لتحديد هجمات متقدمة).

٤. الاستجابة التلقائية (Automated Incident Response):

بعض منصات SIEM مثل Splunk أو Wazuh يمكنها تنفيذ إجراءات تلقائية فور استقبال تبليغ من IDS ، مثل إرسال أوامر إلى الجدار الناري لحظر المهاجم.

٥. دعم الامتثال والتدقير (Compliance & Auditing):

الدمج بين النظم يسهل عملية إعداد تقارير امتثال لمعايير أمنية مثل ISO 27001 أو PCI-DSS و GDPR عبر توثيق كل الأحداث بشكل مركزي ومنظم.

٢٦.٤ التحديات المرتبطة بعملية التكامل

رغم الفوائد الكبيرة، إلا أن الدمج بين SIEM و IDS يواجه بعض العقبات الفنية والتنظيمية، مثل:

- اختلاف تنسيدات السجلات (Log Formats) تحتاج المؤسسات إلى آليات تطبيق فعالة لتوحيد البيانات من مصادر متعددة.

- الكلم الهائل من البيانات (Data Volume) كل من IDS و SIEM يولد آلاف الأحداث يومياً، مما يتطلب خوادم قوية وسعة تخزين كبيرة.
- ضبط القواعد (Rule Tuning) أي خطأ في قواعد الترابط قد يؤدي إلى تتبّعها زائدة أو فقدان أحداث مهمة.
- الزمن الحقيقي (Real-time Processing) التكامل الفعال يتطلّب معالجة سريعة دون تأخير في نقل وتحليل الأحداث.
- التوافق البرمجي: قد تختلف واجهات الربط (APIs) بين الأنظمة المختلفة، مما يتطلّب كتابة أدوات وسيلة (Middleware) أحياناً.

٢.٧ الخلاصة :

استعرض هذا الفصل الأسس النظرية الازمة لفهم وتصميم نظام كشف تسلل مخصص لاكتشاف هجمات SQL Injection وربطه بمنصة SIEM للتحليل المركزي. بدأنا بتاكيد أهمية أمن النظم والمعلومات كمدخل استراتيجي لحماية الأصول الرقمية، ومراجعة نموذج المثلث الأساسي (CIA) ومبادئ التصميم الآمن كالحد من الامتيازات والدفاع المتعدد الطبقات و Zero Trust. ثم انتقلنا إلى تحليل طبيعة هجمات تطبيقات الويب، مع تفصيل آليات وطرق تنفيذ حقن SQL (Union-based, Error-based, Blind, Time-based)، مع إبراز صعوبة اكتشاف بعض الأنماط البطيئة أو المشفرة وال الحاجة إلى مؤشرات كشف متخصصة.

بعد ذلك عرضنا بشكل مفصل وظائف وأدوار أنظمة كشف ومنع التسلل (IDS/IPS)، وشرحنا فروق NIDS مقابل HIDS، ومنهجيات الكشف (Signature vs Anomaly) ومزايا كل منها وقيودهما. بيّنا أيضاً مكونات IDS الأساسية (التقطات البيانات، محرك التحليل، قاعدة التواقيع، وحدة التتبّع) ومعايير قياس الأداء مثل معدل الكشف والזמן اللازم للتحقق ومؤشرات الإنذارات الكاذبة. هذا العلم يوفر الأساس التقني لكتابة قواعد كشف فعالة، وفهم متطلبات الأداء والموارد لنشر IDS في بيئه حقيقية.

ثم عرض الفصل دور نظم SIEM كمنصة مركبة تجمع سجلات البنية التحتية، وتقوم بتطبيعها وربطها وتحليلها عبر قواعد ارتباط وخوارزميات ذكية، وتدعم الاستجابة الآلية والإجراءات التصحيحية. بيّنا كيف يعالج SIEM مشكلات التشتت في السجلات ويخفض الإنذارات الكاذبة عبر correlation، كما يدعم الامتنال والتدقيق وتحليل الاتجاهات. وربطنا بينهما في قسم مخصص تناول آليات التكامل (T) توليد الأحداث، نقلها عبر بروتوكولات مثل JSON/Syslog، التطبيّع، الترابط، والاستجابة التلقائية، مع توضيح فوائد التكامل وأبرز التحديات التقنية مثل حجم البيانات، تطبيّع الصيغ، وضبط قواعد الترابط.

أجريت مراجعة مستهدفة للدراسات السابقة التي تناولت كشف SQL Injection ودمج IDS مع SIEM، وخلص الفصل إلى استنتاجين عمليين: الأول أن الاعتماد على توقيعات بسيطة وحدها غير كافٍ لمحابهة تطور

أساليب الحقن، والثاني أن الدمج الفعال مع SIEM يعزز الدقة ويقلل الضوضاء التحليلية، خصوصاً عند إكماله بتقنيات تحليل سلوكي أو تعلم آلي. تلك النتائج تدعم الفرضية المركزية للمشروع: أن تصميم IDS مخصص لاكتشاف أنماط SQLi وربطه بشكل صحيح بنصبة SIEM سيؤدي إلى تحسين ملحوظ في معدلات الكشف وسرعة الاستجابة مقارنة بالحلول المنعزلة.

ختاماً، يؤسس هذا الفصل لمرحلة التصميم والتنفيذ التي ستتركز عليها الفصول التالية. سنتناول في الفصل الثاني متطلبات النظام ، بنية الاختبار (testbed) المقترحة، واختيار الأدوات (مثل IDS ك Suricata/ Snort و Wazuh/ELK كسيروفيس SIEM) مع تبرير تقني لاختيار كل مكون. بعد ذلك سننتقل إلى فصل يشرح كتابة قواعد الكشف، آليات تطبيق السجلات، آليات الربط (log forwarding) إلى SIEM ، وختارات الأداء والحالات التجريبية لهجمات SQLi المتوعة.

الفصل الثاني : بيئة العمل

٣.١ المقدمة :

هدف هذا الفصل إلى توثيق خطوات تجهيز بيئة العمل التي تم الاعتماد عليها لتنفيذ مشروع كشف محاولات حقن قواعد البيانات (SQL Injection) ضمن بيئة مخبرية، مع ضمان أن تكون المكونات متراقبة وقابلة لإنتاج سجلات (Logs) قابلة للتحليل من طرف نظام إدارة أحداث الأمان. ويُعدّ إعداد البيئة خطوة تأسيسية لأن دقة النتائج اللاحقة تعتمد مباشرة على سلامة التنصيب، وضبط الإعدادات، وتوافق المخرجات بين الأدوات. يرتكز هذا الفصل حصرياً على: عرض الأدوات المستخدمة، مبررات اختيارها، آلية تنصيبها، ثم طريقة ربطها معًا بحيث يتم توليد أحداث HTTP وتتبعها وتحليلها مركزياً.

٣.٢ الأدوات المستخدمة :

اعتمدت بيئة العمل في هذا المشروع على منظومة أدوات متكاملة تعمل كسلسلة متراقبة تبدأ من توليد حركة HTTP داخل بيئة اختبار، مروراً بالكشف الشبكي عن أنماط SQLi ، وانتهاءً بالتجميع والتحليل المركزي للأحداث. وفيما يلي شرح الأدوات المستخدمة مع تضمين مكونات Wazuh الأساسية التي تشَكِّل البنية التشغيلية للمنصة:

Wazuh ٣.٢.١

تم استخدام Wazuh بوصفه منصة SIEM مفتوحة المصدر لالتقط السجلات وتحليلها وتوليد التتبعات وفق قواعد معرفة. ويقوم Wazuh ضمن المشروع على عدة مكونات أساسية، لكل منها دور محدد في دورة حياة الحدث: (Log/Event Pipeline)

• Wazuh Manager

هو المكون المركزي المسؤول عن استقبال البيانات القادمة من الوكلاء ومصادر السجلات، ثم إجراء عمليات التحليل والتقطيع والتصنيف وتطبيق قواعد الكشف. داخل الـ Manager تُعمل وظائف محورية مثل:

- فكّ وترميز الرسائل وتقسيمها (Decoding/Parsing) اعتماداً على الـ Decoders.
- تطبيق قواعد الكشف (Rules Engine) لتوليد تنبؤ أو رفع مستوى الخطورة عند تحقق شروط معينة.
- إدارة الوكلاء (Agent Management) من حيث التسجيل (Registration) والتحقق من الاتصال وتحديث الإعدادات.

Wazuh agent •

هو عميل مراقبة يتم تثبيته على النظام الذي تتنج فيه السجلات (مثل جهاز تشغيل Suricata أو الخادم الذي يقرأ منه eve.json). مهمته:

- مراقبة ملفات السجلات المحددة في الإعدادات.
- إرسال الأحداث بشكل آمن ومنتظم إلى Wazuh Manager.
- تمكين جمع معلومات إضافية على مستوى المضيف عند الحاجة (Host-based Visibility).

Wazuh Indexer •

يمثل طبقة التخزين والفهرسة (Indexing Layer) التي تحفظ الأحداث بعد معالجتها، وتتيح البحث السريع والاستعلام عنها. وجود الـ Indexer ضروري لتمكين:

- حفظ كميات كبيرة من الأحداث بكفاءة.
- تنفيذ عمليات بحث وتحليل لاحقة بشكل سريع ومرن.

Wazuh Dashboard •

واجهة العرض والتحكم التي تُستخدم لمتابعة التبيهات والأحداث والرسوم البيانية والتحليلات وتتيح :

- مراقبة تدفق السجلات والتبيهات بشكل بصري.
- استعراض نتائج القواعد والبحث في الأحداث المؤرشفة داخل الـ Indexer.
- تسهيل التحقق من نجاح التكامل بين الأدوات عبر تتبع وصول تبيهات Suricata وتصنيفها.

Decoder/Rules •

رغم أنها ليست "أدوات" مستقلة، إلا أنها مكونات تشغيلية داخل Wazuh لا يمكن إهمالها في وصف المنظومة:

- Decoders مسؤولة عن استخراج الحقول من السجلات الخام مثل JSON الخاص بـ Suricata وتحويلها إلى حقول قابلة للاستخدام في القواعد.
- Rules تمثل منطق الكشف الذي يحدد متى يتم اعتبار الحدث مؤشراً لهجوم أو سلوك غير طبيعي، وكيف يتم تقييم شدته.

بهذا التكوين، يعمل Wazuh كنظام متكامل يبدأ باستقبال سجل قادم من مصدر خارجي (Suricata)، ثم يفككه ويفهمه (Decoders)، وبعد ذلك يطبق منطق الكشف (Rules)، ويخرج النتائج (Indexer)، ويعرضها (Dashboard) للمراقبة.

Suricata ٣.٢.٢

تم اعتماد suricata في هذا المشروع بوصفها نظام كشف تسلل شبكي (Network Intrusion Detection System – NIDS)، أي أنها تعمل على المراقبة والتحليل وإطلاق التبيهات دون تنفيذ إجراءات منع تلقائية. وظيفة Suricata هنا تدور حول رصد حركة المرور الشبكية المرتبطة بتطبيق الويب داخل بيئه الاختبار، وتحليلها لاكتشاف أنماط هجمات SQL Injection، ثم تسجيل نتائج الكشف ضمن سجل أحداث قابل للاستهلاك من Wazuh.

- آلية عمل IDS كـ Suricata في سياق المشروع
- تعتمد IDS كـ Suricata على مبدأ أساسى وهو فحص حركة المرور (Traffic Inspection) ومقارنتها بمجموعة من التواقيع (Signatures/Rules). وعند تحقق شروط توقيع معين، تقوم Suricata بإنشاء حدث تبیه (Alert Event) بدل منع الاتصال أو إسقاط الحزم.

ضمن هذا المشروع، تمر العملية عملياً بالمراحل التالية:

- النقطة حركة المرور (Traffic Capture)
تقوم Suricata بمراقبة واجهة الشبكة داخل بيئه العمل للالتقط تدفق الاتصالات المتوجهة نحو خادم الويب الذي يستضيف DVWA. هذا الالتقط يتيح فحص طلبات HTTP الفعلية الناتجة عن تفاعل المستخدم مع التطبيق، بما في ذلك الطلبات التي قد تحتوي مدخلات خبيثة. تحليل بروتوكول HTTP واستخراج الأجزاء ذات الدلالة الأمنية
- مطابقة حركة المرور مع قواعد الكشف (Rule Matching)
تطبق Suricata القواعد المعرفة للكشف عن SQLi عبر مطابقة أنماط نصية (مثل كلمات دلالية أو سلوك) وعند تطابق الطلب مع قاعدة معينة، يتم إنتاج تبیه يحمل هوية القاعدة ومعلوماتها.
- توليد سجل تبیه بصيغة JSON داخل eve.json
الناتج التشغيلي الأكثر أهمية في المشروع هو أن Suricata تسجّل التبيهات ضمن ملف eve.json

حيث يظهر الحدث عادة كنوع event_type:"alert" ويتضمن معلومات تشخيصية مفيدة لمرحلة التجميع والتحليل في Wazuh ، مثل:

رقم التوقيع الذي تم تفعيله (يمثل القاعدة التي اكتشفت النمط).
وصف القاعدة (رسالة التبیه).

بيانات المصدر والوجهة التي تساعد على تتبع سياق src_ip / dest_ip / ports / proto الهجوم.

هذا التسجيل بصيغة JSON هو ما يجعل التكامل مع Wazuh Agent عملياً، لأن Wazuh يستطيع قراءة هذا الملف وتمريره إلى Wazuh Manager لمعالجته.

DVWA ٣.٢.٣

تم استخدام DVWA كتطبيق ويب تدريسي لتوليد سيناريوهات SQLi ضمن سياق حقيقي. يتيح ذلك:

- إنشاء طلبات HTTP تحتوي مؤشرات SQLi في URI أو جسم الطلب (Request RAW-URI أو Body).
- اختبار قدرة Suricata على الالتفاف، ثم قدرة Wazuh على الاستقبال والتحليل وإظهار التبليغ.

٣.٢.٤ قاعدة البيانات (MySQL)

تعمل قاعدة البيانات كخلفية تشغيل DVWA ، مما يجعل سيناريوهات SQLi مرتبطة بمعمارية Web App ، ما يجعل قاعدة البيانات ضروري لأن:

- التطبيق يعتمد على استعلامات SQL فعلية.
- إدخالات SQLi تتفاعل ضمن سياق قريب من بيئات الإنتاج من حيث المنطق العام للتطبيق.

٣.٢.٥ خادم الويب وبيئة التشغيل

تم تشغيل DVWA على PHP مع Apache ضمن نظام لينكس، وذلك من أجل:

- استقبال وتتنفيذ طلبات HTTP/GET/POST الخاصة بالتطبيق.
- توفير بيئة تشغيل متكاملة تضمن أن التفاعلات داخل DVWA تنتج حركة مرور قابلة للرصد والتحليل Suricata.

٣.٢.٦ بيئة افتراضية (Virtualization)

تم تنفيذ البيئة ضمن آلة افتراضية لدعم:

- العزل الأمني أثناء تشغيل تطبيق ضعيف أمنياً وإجراء اختبارات هجومية.
- التحكم بالشبكات والواجهات (Interfaces) الازمة لالتفاف الحركة بواسطة Suricata

- سهولة إعادة التجربة وتكرار نفس سيناريوهات الاختبار.

٣٠٢٧ سبب اختيار الأدوات عن منافسيها

المحاور	Wazuh	Splunk ES	IBM QRadar
نموذج النشر	محلي/سحابي	محلي/سحابي	محلي/سحابي
جمع البيانات	Logs Agents + قراءة ملفات JSON (مثل) وإرسالها للمدير للتحليل	Forwarders، غني من ناحية مصادر البيانات	تجميع أحداث (Events) وتدفقات (Flows) من مصادر متعددة
منهجية الكشف	قواعد + فك ترميز /Decoders + تتبیهات؛ مناسب لربط سجلات أدوات خارجية (مثلاً IDS)	Risk- Based Alerting التنبیهات الكاذبة	على Correlation مستوى الأحداث/التدفقات Offenses لإنتاج
التخزين والفهرسة والعرض	Indexer للتخزين والفهرسة + Dashboard للعرض	قدرات بحث/تحليلات قوية مع لوحة متقدمة (Splunk)	قدرات إدارة حوادث “Offenses” وواجهات تحقيق تجاري
الترخيص والتكلفة	مفتوح المصدر (مع خيارات خدمة سحابة)	تجاري وكلفته ترتفع مع حجم البيانات	تجاري

جدول (١) : الفروقات بين أدوات SIEM

Snort	Suricata	المحور
(Rules) قواعد DS/IPS على قواعد (Rules)	(Rules) قواعد DS/IPS على قواعد (Rules)	طبيعة الأداة
Rule-based signature قواعد detection تحدد أنماط نشاط Alerts خبيث وتوارد	Rule-based signature detection تحدد أنماط نشاط خبيث وتوارد Alerts	أسلوب الكشف
يعتبر جيد مثل Suricata كأداء لكن لا يدعم Multi-threaded	وقابلية تفتيش عالية Multi-threaded الأداء؛ مناسب لرفع Throughput	الأداء والمعمارية
يعتمد على تحليل/مسابقات معالجة قواعد	Automatic Protocol Detection وتطبيق منطق الكشف/التسجيل حتى على منافذ غير قياسية	فهم البروتوكول
يدعم alerts/logs اصعب في التكامل مع wazuh	يوحد alerts/metadata في JSON ؛ ممتاز SIEM للتكامل مع	المخرجات والتكامل

جدول (٢) : الفروقات بين أدوات IDS

٣.٣ تنصيب الأدوات

توضح هذه الفقرة خطوات تنصيب الأدوات الأساسية التي بُنيت عليها بيئة المشروع، بدءاً من منصة Wazuh ثم Suricata، وبعدها إعداد بيئة الاختبار المتمثلة ب Apache DVWA يركّز هذا القسم على جانب "التنصيب والتجهيز الأولي"

٣.٣.١ Wazuh تنصيب

هناك طريقتين لتنصيب أداة Wazuh وهما (Installation guide-Quickstart) تم اختيار طريقة Installation guide لأنها تناسب طبيعة المشروع

: Wazuh indexer

يعتمد Wazuh في مكوناته المركزية على اتصال مشفر بين طبقات المنظومة، لذلك يبدأ دليل التنصيب الرسمي بمرحلة إنشاء شهادات TLS ثم الانتقال لتنصيب الـ Indexer وضبطه.

أ) المرحلة الأولى: إنشاء الشهادات (Certificate creation)

١. تنزيل أداة إنشاء الشهادات وملف الإعداد config.yml

٢. تعديل config.yml لإدخال أسماء العقد (Indexer/Server/Dashboard) وعنوانين

IP/Hostnames الخاصة بها، بحيث تُستخدم نفس القيم لاحقاً أثناء ضبط كل مكون .

٣. توليد الشهادات عبر تشغيل سكريبت الشهادات، ثم ضغط المخرجات ضمن ملف واحد -wazuh-certificates.tar ونقله إلى جميع العقد .

```
# curl -s0 https://packages.wazuh.com/4.14/wazuh-certs-tool.sh
# curl -s0 https://packages.wazuh.com/4.14/config.yml

nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "<indexer-node-ip>"
    - name: node-2
      ip: "<indexer-node-ip>"
    - name: node-3
      ip: "<indexer-node-ip>"

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: "<wazuh-manager-ip>"
```

الشكل (١) : أداة تنزيل الشهادات واعداد الملف config.yaml

ب) المرحلة الثانية: تنصيب عقد الـ **Indexer** وضبطها

بعد تجهيز الشهادات، يتم تنصيب الحزم المطلوبة ثم إضافة مستودع Wazuh وتنصيب حزمة **wazuh-indexer**.
بعدها يُضبط ملف إعدادات الـ **Indexer** (opensearch.yml) لتحديد عنوان الشبكة

نقاط الضبط الأساسية كما في الدليل:

- (config.yml) لعنوان العقدة (ويُفضل مطابقته لما تم وضعه في network.host
- config.yml لاسم العقدة مثل كما عُرف في node.name
- إعدادات الكلاستر مثل discovery.seed_hosts و cluster.initial_master_nodes عند استخدام أكثر من عقدة .

ج) المرحلة الثالثة: نشر الشهادات وتهيئة **cluster** وتشغيل الخدمة

يتم استخراج الشهادات الخاصة بعقدة الـ **Indexer** إلى المسار المخصص للشهادات ثم تعيين الأذونات والملكية كما في الدليل، وبعدها تشغيل خدمة الـ **Indexer** وأخيراً تُجرى خطوة **Cluster initialization** عبر تشغيل سكريبت التهيئة الأمنية indexer-security-init.sh لتحميل الشهادات وتفعيل الإعدادات الأمنية .

ثانياً تنصيب Wazuh server :

يعرف الدليل الرسمي Wazuh Server باعتباره مكوناً مركزاً يضم **Wazuh Manager** لتحليل بيانات الوكلاء وإنتاج التنبؤات، إضافةً إلى **Filebeat** لنقل التنبؤات والأحداث المؤرشفة إلى Wazuh Indexer يتم تنصيب هذه الطبقة عبر: إضافة المستودع، تثبيت wazuh-manager، تثبيت Filebeat وضبطه، نشر الشهادات، ثم تشغيل الخدمات .

أ) إضافة مستودع Wazuh

تبدأ العملية بإضافة مستودع Wazuh أو تخطيها إذا كانت نفس العقدة تحتوي مسبقاً على المستودع نتيجة تنصيب مكون سابق

ب) تثبيت Wazuh Manager

يتم تثبيت الحزمة wazuh-manager باستخدام مدير الحزم المناسب للنظام .

ج) تثبيت Filebeat ثم تنزيل ملف إعداد جاهز وضبطه

يوجه الدليل إلى تثبيت Filebeat ثم تنزيل ملف إعداد مسبق من Wazuh ، وبعدها تعديل قيمة hosts لتشير إلى عقدة لا Indexer المعتمدة

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
hosts: ["10.0.0.1:9200"]
protocol: https
username: ${username}
password: ${password}
```

الشكل (٢) : تعديل قيمة host لتشير إلى indexer

د) نشر الشهادات وربط Filebeat/Manager بالـ Indexer

يتم نقل ملف wazuh-certificates.tar إلى عقدة السيرفر ثم نشر الشهادات وفق اسم العقدة المحدد في config.yml

بالإضافة إلى ذلك، يوضح الدليل خطوة حفظ بيانات اعتماد لا Indexer في **Wazuh manager** وذكر إمكانية تخطيتها إذا لم تكن بعض القدرات مطلوبة باستخدام wazuh-keystore

ه) تشغيل الخدمات والتحقق

بعد اكتمال الخطوات السابقة، يتم تشغيل filebeat ثم تشغيل wazuh-manager والتحقق من حالتهما .

ثالثاً تنصيب Wazuh Dashboard :

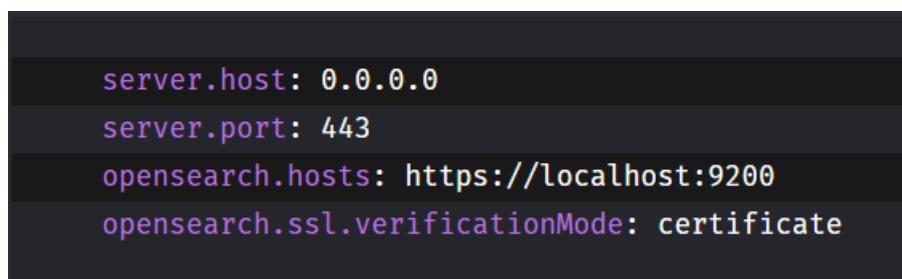
تم تنصيب Wazuh Dashboard كواجهة ويب لعرض وتقييم تنبیهات السيرفر والأحداث المؤرشفة. يوضح الدليل مراحل: تثبيت الاعتمادات، إضافة المستودع، تثبيت الحزمة، ضبط ملف opensearch.hosts خاصة (server.host و opensearch_dashboards.yml) ، نشر الشهادات، تشغيل الخدمة، ثم ضبط ملف wazuh.yml لربط الـ Wazuh Server بـ Dashboard.

أ) تثبيت الحزمة وضبط الاتصال بالـ Indexer

يتم تثبيت wazuh-dashboard ثم ضبط:

• server.host للسماح بالاتصال بالـ

• opensearch.hosts لتحديد عنوان/عناوين الـ Indexer التي سيتصل بها الـ Dashboard



```
server.host: 0.0.0.0
server.port: 443
opensearch.hosts: https://localhost:9200
opensearch.ssl.verificationMode: certificate
```

الشكل (٣) : تحديد server host & openserch.host

ج) نشر الشهادات ثم تشغيل الخدمة وربط الـ Dashboard بالـ Server

يشترط الدليل وجود wazuh-certificates.tar في Dashboard لنشر شهادات الـ

ث/ ثم تشغيل الخدمة. بعد ذلك يتم تعديل الملف:
/etc/wazuh-dashboard/certs
/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml

لتحديد عنوان Wazuh Server عادةً عبر منفذ API

كما يوضح الدليل طريقة الدخول الأولى إلى الواجهة عبر HTTPS باستخدام بيانات الاعتماد الافتراضية في سيناريو الـ step-by-step

رابعاً تنصيب Wazuh agent :

يوضح الدليل الرسمي أن تنصيب وكيل Wazuh على Linux يمكن تنفيذه باستخدام **Deployment Manager** لتسهيل عملية التثبيت والتسجيل والتهيئة الأولية، حيث يتم تحديد عنوان الوكيل عبر متغير مثل `WAZUH_MANAGER` أثناء التثبيت، ثم تشغيل خدمة الوكيل.

خطوات التنصيب وفق الدليل

١. تنفيذ أمر التثبيت باستخدام مدير الحزم، مع تعين قيمة `WAZUH_MANAGER` بعنوان مدير Wazuh (IP/Hostname).

٢. تفعيل خدمة `wazuh-agent` وتشغيلها.

٣.٣.٢ تنصيب Suricata

اعتمد المشروع على تنصيب Suricata بوصفها نظام كشف اختراق شبكي (IDS) لإجراء فحص لحركة المرور وإنتاج تنبؤات يتم تسجيلها بصيغة JSON داخل ملف `eve.json`، بحيث تكون هذه المخرجات قابلة للتجميع والمعالجة لاحقاً ضمن منصة Wazuh. تم تنفيذ خطوات التنصيب والإعداد وفق الدليل الرسمي الذي يوضح آلية إعداد Suricata على Ubuntu 22.04.

(أ) تثبيت Suricata على جهاز المراقبة (Endpoint)

يبدأ الدليل بإضافة مستودع Suricata المستقر ثم تحديث قائمة الحزم وتنصيب Suricata.

(ب) تحميل وتفعيل مجموعة القواعد (Ruleset)

يوضح الدليل تحميل حزمة قواعد Emerging Threats الخاصة بـ Suricata ثم استخراجها وإنشاء مسار قواعد مخصص ونقل ملفات القواعد إليه.

(ج) تهيئة ملف suricata.yaml الرئيسي

بعد تثبيت الأداة والقواعد، يتم ضبط المتغيرات الأساسية داخل الملف `/etc/suricata/suricata.yaml` بما يضمن:

- تعريف نطاق الشبكة الداخلية (`HOME_NET`) بشكل صحيح.
- تحديد مسار القواعد الافتراضي (`default-rule-path`).
- تحميل ملفات القواعد (`rule-files`).
- تفعيل الإحصاءات العامة (اختياري ضمن الدليل).

- تحديد واجهة الشبكة المراد مراقبتها ضمن af-packet مثل enp0s3

```

HOME_NET: "<UBUNTU_IP>"
EXTERNAL_NET: "any"

default-rule-path: /etc/suricata/rules
rule-files:
- "*.rules"

stats:
enabled: yes

af-packet:
- interface: enp0s3

```

الشكل (٤) : تهيئة IDS

(د) إعادة تشغيل Suricata لتطبيق الإعدادات

بعد حفظ التعديلات في suricata.yaml، يتم إعادة تشغيل خدمة Suricata لتطبيق الإعدادات الجديدة

(هـ) تمكين Wazuh Agent من قراءة سجل Suricata eve.json

لكي تصل تطبيقات Wazuh إلى منصة Suricata ، يوضح الدليل إضافة مقطع داخل ملف إعداد وكيل Wazuh:

/var/ossec/etc/ossec.conf

حيث يقرأ الوكيل ملف سجل Suricata:

/var/log/suricata/eve.json

وبصيغة json.

```

<ossec_config>
  <localfile>
    <log_format>json</log_format>
    <location>/var/log/suricata/eve.json</location>
  </localfile>
</ossec_config>

```

الشكل (٥) : إضافة localfile في ملف ossec

(و) إعادة تشغيل Wazuh Agent لتطبيق إعداد جمع السجلات

بعد تعديل إعدادات الوكيل، يتم إعادة تشغيل خدمة Wazuh Agent لتفعيل مراقبة ملف eve.json

٣.٣.٣ تنصيب DVWA & MYSQL & APACHE

في هذا الجزء تم تجهيز بيئة تطبيق ويب ضعيف أمنياً (DVWA) ليعمل فوق خادم ويب (Apache) مع ربطه بقاعدة بيانات MySQL ، بهدف توفير بيئة اختبار تولد طلبات HTTP واقعية يمكن استخدامها لاحقاً في سيناريوهات الكشف والتحليل. تم تنفيذ الخطوات على نظام Linux

(أ) تنصيب PHP (Web Server Stack) و Apache

تم تنصيب Apache كخادم ويب و PHP كبيئة تنفيذ لتطبيق DVWA بعد التثبيت تم تفعيل الخدمة والتأكد من أنها تعمل وتسجّب لطلبات HTTP ، لأن ذلك شرط أساسى قبل نشر ملفات التطبيق.

(ب) تنصيب وتجهيز قاعدة بيانات

تم تنصيب MySQL كDatabase backend لتطبيق DVWA ، ثم تم إنشاء Database ومستخدم مخصص للتطبيق وتحديد صلاحياته، وبعدها تم الاعتماد على هذه البيانات في إعدادات DVWA وجود MySQL (Database name / username / password / host / port) بشكل جاهز ومهماً يضمن أن DVWA سيُعمل بخصائصه المعتمدة على SQL بشكل صحيح.

(ج) نشر DVWA على Apache وربطه بقاعدة البيانات

بعد تجهيز Apache و MySQL، تم نشر ملفات DVWA ضمن مسار الويب الخاص بـ Apache ليصبح التطبيق قابلاً للوصول عبر المتصفح. ثم تم ضبط ملفات الإعداد الخاصة بـ DVWA لإدخال بيانات الاتصال بـ MySQL بعد ذلك تم التحقق من جاهزية التطبيق عبر صفحة Setup Check والتأكد من:

- توفر وحدات PHP المطلوبة مثل mysqli pdo_mysql.
- صلاحيات الكتابة للمجلدات التي يتطلبها التطبيق مثل config و uploads.
- نجاح الاتصال بقاعدة البيانات.

٤. ربط الأدوات

بعد إتمام تنصيب جميع الأدوات، تم تنفيذ عملية الربط بينها وفق سلسلة عمل متكاملة تضمن انتقال البيانات من نقطة توليدتها وحتى ظهورها كتبية قابل للتحليل داخل منصة Wazuh تبدأ السلسلة من طبقة التطبيق، حيث تم تشغيل DVWA على خادم الويب Apache وربطه بقاعدة البيانات MySQL بحيث يعمل التطبيق بصورة طبيعية ويولد طلبات HTTP فعلية

تمثل هذه الخطوة أساس بيئة الاختبار لأنها تضمن أن التفاعلات داخل DVWA تنتج حركة مرور واقعية مرتبطة باستعلامات SQL ، ما يسمح لاحقاً برصد مؤشرات SQL Injection في أماكن مختلفة من الطلب مثل URI و RAW-URI و HTTP Request Body

نقطة التسليم الأولى (Handover Point) في المنظومة كانت هنا : حركة HTTP الناتجة عن DVWA/Apache هي المدخل الذي تعتمد عليه طبقة الكشف الشبكي. بعد ذلك تم إعداد Suricata للعمل بوصفها IDS فقط ضمن المختبر، وتم ضبطها لانتقاد حركة المرور عبر واجهة الشبكة المعتمدة en0s3 ثم تحليل بروتوكول HTTP ومطابقة المحتوى مع قواعد الكشف المفعّلة.

عند تحقق المطابقة تقوم Suricata بتوليد حدث تتبّيه وتسجّله ضمن ملف السجل القياسي /var/log/suricata/eve.json بصيغة JSON؛ ويحتوي هذا الحدث على عناصر أساسية تم الاعتماد عليها في المشروع، أهمها نوع الحدث event_type:"alert" ومعرف التوقيع signature_id ووصف التوقيع signature، إضافة إلى معلومات المصدر والوجهة (IP/Ports) التي تحفظ سياق التتبّيه.

نقطة التسليم الثانية كانت : ملف eve.json باعتباره واجهة التكامل الرسمية بين Suricata ومنصة Wazuh ولإيصال هذه التبيّهات إلى منصة التحليل المركزية، تم ربط Suricata مع Wazuh Agent عبر Wazuh Manager من خلال تفعيل جمع السجلات من ملف eve.json مباشرة، بحيث يقوم الوكيل بمراقبة الملف وقراءة كل حدث جديد وإرساله إلى Wazuh Manager بشكل مستمر.

بعد وصول الأحداث إلى جهة الإدارة يقوم Wazuh Manager بمعالجتها عبر آليات التحليل Suricata (Decoding/Parsing) ثم تطبيق قواعد Wazuh التي تم إعدادها لتحديد أن الحدث صادر عن وأنه يرتبط بتواقيع SQLi المعتمدة، وبعد ذلك تُرسل النتائج إلى Wazuh Indexer لغرض التخزين والفهرسة.

نقطة التسليم الثالثة كانت : نقل الأحداث المعالجة من Wazuh Manager إلى Wazuh Indexer لضمان حفظها وإتاحتها للبحث والتحليل. أخيراً، يتم عرض البيانات والتحقق منها عبر Wazuh Dashboard باستخدام البحث داخل الواجهة للتتأكد من وصول أحداث Suricata وتصنيفها بالشكل الصحيح، وبذلك تكتمل السلسلة التشغيلية من طرف إلى طرف (DVWA/Apache/MySQL) : يولد حركة HTTP ، و (Suricata) يكتشف ويسجل التتبّيه، و (Wazuh Agent) يجمع السجل، و (Wazuh Manager) يحل ويصنّف، ثم (Indexer/Dashboard) يخزن ويعرض النتائج للتحليل.

أهم التعليمات/الإعدادات التي حققت الربط بين المكونات

- تحديد واجهة الالتقاط في Suricata: ضبط الواجهة ضمن قسم af-packet في ملف suricata.yaml إلى interface: en0s3 لضمان التقاط حركة HTTP المتوجه إلى خادم DVWA؛ أي خطأ في اسم الواجهة يؤدي إلى عدم التقاط الحركة وبالتالي عدم توليد أي تنبيهات.
- تفعيل سجل EVE JSON بصيغة Suricata وتحديد موقعه :الاعتماد على الملف alert كمصدر أساسي للتنبيهات الشبكية، لأنه يسجل أحداث Wazuh بالحقول اللازمة للتحليل داخل.
- تعريف جمع السجلات في Wazuh Agent: إضافة كتلة localfile ضمن ملف إعداد الوكيل : مع /var/ossec/etc/ossec.conf
 - log_format = json
 - Location = /var/log/suricata/eve.json
- إعادة تشغيل الخدمات بعد أي تعديل :إعادة تشغيل Suricata بعد تعديل suricata.yaml ، وإعادة تشغيل wazuh-agent بعد تعديل ossec.conf للتأكد من تطبيق الإعدادات فعلياً على مستوى التشغيل وليس فقط حفظها في الملفات.
- التحقق من وصول الأحداث عبر Wazuh Dashboard: إجراء تحقق عملي عبر واجهة Dashboard باستخدام البحث/الاستعلام للتأكد من ظهور أحداث Suricata ، وهو ما يثبت نجاح الربط بين طبقة الكشف الشبكي ومنصة التجميع والتحليل المركزية.

الفصل الثالث : الحل المقترن

٣.١ مقدمة الفصل

يقدم هذا الفصل الحل المقترن لكشف هجمات SQL Injection ضمن بيئه ويب مخبرية عبر تصميم طبقي يدمج بين طبقة توليد الحركة (DVWA/Apache/MySQL) ، وطبقة كشف شبكي تعمل ك IDS لإنتاج تنبیهات توقعیة، ثم طبقة تجميع وتحليل مركبة (Wazuh) لتحويل هذه التنبیهات إلى أحداث مصنفة وقابلة للبحث والعرض. يرکز الحل على نوعین رئیسین من هجمات SQLi وفق التصنيف المعتمد في المشروع:

Boolean Exploitation و Time Delay مثل Blind/Inferential SQL Injection .١

و (Heavy Query

Comments و Tautology و UNION و Error-based) مثل In-band SQL Injection .٢

و (Piggybacked

ولا يتضمن هذا الحل هجمات Out-of-Band SQL Injection لأنها خارج نطاق المشروع الحالي.

يتكون الحل من خمس طبقات متراكمة، لكل طبقة مسؤولية محددة ومدخلات ومخروطات واضحة:

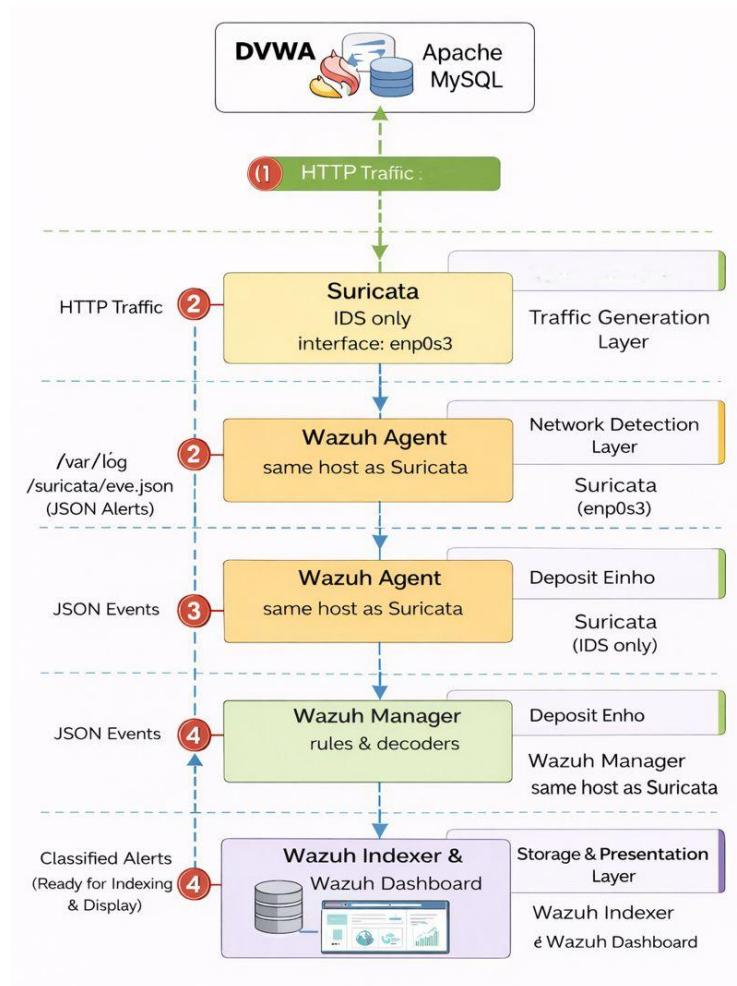
١. MySQL بـ Traffic Generation Layer توليد طلبات HTTP واقعية عبر Apache على DVWA

٢. Network Detection Layer مراقبة حركة HTTP واكتشاف أنماط SQLi باستخدام Suricata

٣. Log Collection Layer جمع سجل تنبیهات (eve.json) عبر Wazuh Agent وSuricata الموجود على نفس جهاز

٤. Analytics & Classification Layer تحليل أحداث Suricata داخل Wazuh Manager وتصنيفها وفق نوع SQLi وموقعه في الطلب.(URI/RAW-URI/BODY)

٥. Storage & Presentation Layer فهرسة الأحداث داخل Wazuh Indexer وعرضها والتحقق منها عبر Wazuh Dashboard



الشكل () : طبقات الحل

٣.٢ طبقة توليد الحركة Traffic Generation Layer

المكونات : DVWA + Apache + MySQL

الهدف: إنتاج حركة HTTP تمثل سيناريوهات إدخال واقعية يمكن أن تحتوي مؤشرات SQLi ضمن أجزاء الطلب المختلفة.

تعتمد هذه الطبقة على تشغيل DVWA كWeb application يعمل عبر Apache ويستخدم MySQL كDatabase backend تكمن أهمية وجود قاعدة البيانات في أن مدخلات DVWA تمر بمنطق يعتمد على SQL، وبالتالي تصبح هجمات SQLi ضمن المدخلات "ذات أثر" داخل مسار التطبيق، ما يجعل الحركة المتولدة أقرب لسلوك تطبيقات الويب الواقعية. ضمن المشروع، بحيث يمكن إعادة توليد نفس الأنماط خلال الاختبارات والتأكد من قدرة طبقة الكشف على التقاطها.

على مستوى سطح الهجوم (Attack Surface) ، يمكن أن تظهر مؤشرات SQLi ضمن:

- GET في URI / Query Parameters
- Encoding في حال استخدام RAW-URI
- POST في طلبات HTTP Request Body

Handover Point 1:

الخرج الأساسي من هذه الطبقة هو Apache HTTP Traffic المتوجه إلى Apache ، وهو المدخل المباشر لطبقة Suricata التي تقوم بالالتقط والتحليل.

٣.٣ طبقة الكشف الشبكي Network Detection Layer

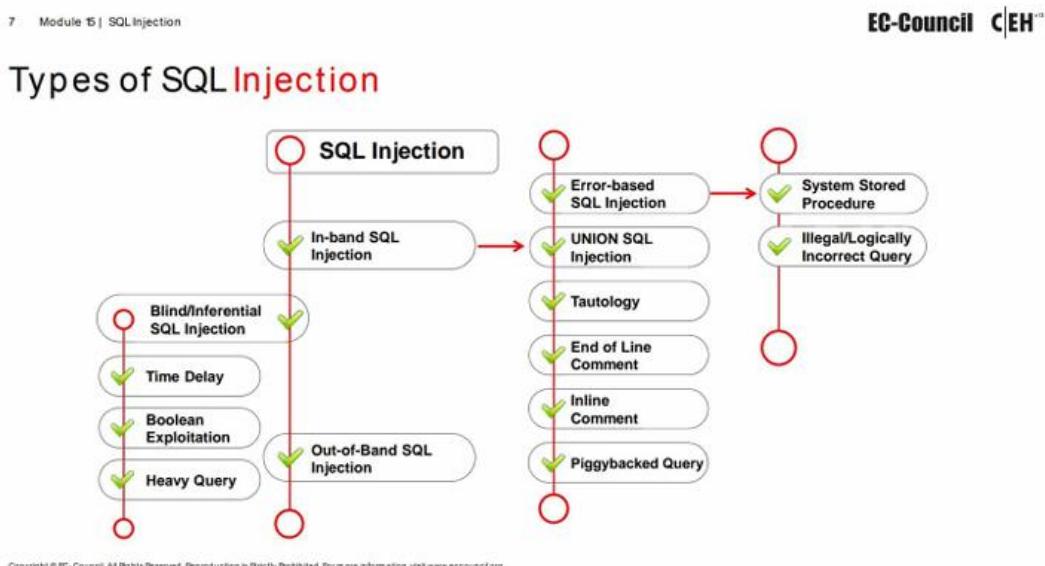
الهدف : كشف أنماط SQLi على مستوى الشبكة عبر قواعد توقيعية، وتوليد Alerts بصيغة JSON قابلة للتجميع.

في هذه الطبقة تعمل Suricata ك IDS فقط، أي أن دورها يقتصر على المراقبة والكشف والتسجيل دون منع الحركة. تم ضبط الالتقط على الواجهة en0s3 لضمان رصد حركة HTTP التي تولّدتها طبقة DVWA/Apache. بعد الالتقط، تقوم Suricata بتحليل HTTP واستخراج عناصر قابلة للتفتيش مثل URI و RAW-URI و Body، ثم تطبق قواعد SQLi المخصصة التي تم إعدادها في المشروع.

٣.٣.١ SQLi في المشروع

اعتمد المشروع على قواعد SQL مخصصة، وتم تنظيمها بحيث تحقق هدفين:

١. التغطية حسب التصنيف المعتمد:



الشكل () : تصنيف sqli

وأغراض الاتساق والتحليل اللاحق داخل Wazuh ، تم ترقيم هذه القواعد ضمن نطاق Signature IDs = 2001xxx (SID) بحيث يصبح تمييز تعيينات SQLi وتمريرها إلى مرحلة التصنيف أسهل وأكثر دقة.

٣.٣.٢ مخرجات Suricata ونقطة التسلیم

عند تحقق قاعدة معينة، تقوم Suricata بإنشاء Alert وتسجيله في الملف /var/log/suricata/eve.json بصيغة JSON.

Handover Point 2:

ملف eve.json يمثل قناة التسلیم الرسمیة بین Suricata وطبقه التحصیع فی Wazuh

٣.٤ طبقة التجميع Log Collection Layer

المكونات Wazuh Agent : على نفس جهاز Suricata .
الهدف : تحويل سجل Suricata المحلي إلى تدفق أحداث مُرسل إلى Wazuh Manager بصيغة قابلة للمعالجة.

تؤدي هذه الطبقة دوراً حاسماً لأنها تمثل نقطة الانتقال من "تبثة محلي على جهاز IDS إلى حدث مركزي داخل SIEM ، يوجد Wazuh Agent على نفس الجهاز الذي تعمل عليه Suricata ، وهذا يمنح ميزتين عمليتين: لا توجد حاجة لنقل ملفات عبر الشبكة أو مشاركة Storage ويقل احتمال فقد السجلات لأن القراءة تتم مباشرة من الملف المحلي.

تم تفعيل جمع السجلات عبر إضافة تعريف قراءة ملف eve.json ضمن إعدادات الوكيل، مع ضبط log_format = json لضممان التعامل مع الأحداث كبنية حقول، لا كنص خام location = /var/log/suricata/eve.json.

Handover Point 3:

لحظة وصول الحدث من Wazuh Agent إلى Wazuh Manager هي الحد الفاصل بين طبقة التجميع وطبقة التحليل والتصنيف.

٣.٥ طبقة التحليل والتصنيف Analytics & Classification Layer

المكونات Wazuh Manager (Rules/Decoding) .
الهدف : تحويل Alerts القادمة من Suricata إلى تبيهات مصنفة ضمن سياق SQLi الخاص بالمشروع.

تُعد هذه الطبقة "القيمة المضافة" للحل المقترن، لأن Suricata بمفردها تنتج Alert توقيعي، لكن Wazuh يضيف بعدها تحليلياً وإدارياً عبر :

- استخراج الحقول من JSON وربطها بمصدرها (Suricata).
- تطبيق قواعد محلية للتعرف على Alert كحدث SQLi ضمن نطاق SIDs الخاص بالمشروع (2001xxx).
- تصنيف التبثة وفق نوع SQLi وموضعه في الطلب (URI/RAW-URI/BODY) عندما يكون ذلك ممكناً عبر معلومات التوقيع أو الـ SID.
- رفع مستوى الخطورة (Level) وإضافة مجموعات (Groups/Tags) تجعل البحث داخل Dashboard أكثر تنظيماً وتمكن التقرير مخرجات واضحة.

عملياً، يعتمد التصنيف في Wazuh على حقيقةتين أساسيتين من تصميم المشروع:

١. أحداث Suricata تصل إلى Wazuh كـJSON ، وبالتالي يمكن استخراج `signature_id` و `signature` بدقة.

٢. قواعد مميزة بـ SID range 2001xxx ، ما يتاح له Suricata تعريف قواعد "شاملة" تلقط SQLi دون الاعتماد على نصوص قد تتغير، ثم قواعد أدق لكل نوع/جزء عند الحاجة.

Handover Point 4:

بعد تطبيق قواعد التحليل والتصنيف، ترسل الأحداث إلى Wazuh Indexer لتخزينها وفهرستها، وتصبح جاهزة للعرض داخل Dashboard

: مثال على قاعدة من WAZUH

```
<!-- Piggybacked/Stacked -->
<rule id="910017" level="12">
  <if_sid>910001</if_sid>
  <field name="alert.signature_id">^20017[0-9][0-9]$\</field>
  <description>SQLi IN-BAND - Piggybacked/stacked statement indicators</description>
  <group>sql, inband, piggybacked</group>
</rule>

<!-- Correlation: multiple SQLi alerts from same src_ip -->
<rule id="910050" level="15" frequency="5" timeframe="120">
  <if_matched_sid>910001</if_matched_sid>
  <same_field>src_ip</same_field>
  <description>SQL Injection attack campaign detected (multiple attempts)</description>
  <group>sql, correlation, critical</group>
```

الشكل () : مثال على ربط Wazuh

٣.٦ طبقة التخزين والعرض Storage & Presentation Layer

المكونات : Wazuh Indexer + Wazuh Dashboard

الهدف : حفظ الأحداث المصنفة وإتاحة البحث والتحقق والتوثيق.

في هذه الطبقة يتم تخزين الأحداث داخل Indexer وفهرستها، ما يتيح إجراء عمليات بحث مبنية على الحقول (Field-based search) مثل البحث بـ `signature_id` أومجموعات القواعد. ثم تُستخدم Wazuh Dashboard للتحقق النهائي من أن: تبيهات Suricata وصلت فعلاً. التصنيف الخاص بـ SQLi تم تطبيقه. يمكن استخراج أدلة التقرير (Screenshots/Queries) بسهولة.

هذه الطبقة ليست للعرض فقط، بل هي جزء من آلية ضمان الجودة، لأن نجاح البحث وإظهار الأحداث يثبت عملياً صحة الرابط من طرف إلى طرف.

٣.٨ ملخص نقاط التسليم Handover Points في الحل المقترن

لإظهار التصميم بصورة علمية داخل التقرير، يلخص الحل المقترن نقاط التسليم كما يلي:

١. Suricata مدخل DVWA/Apache → HTTP Traffic

٢. Wazuh Agent مدخل Suricata → /var/log/suricata/eve.json JSON Alerts

٣. Wazuh Agent بدء التحليل والتصنيف → Wazuh Manager

٤. Wazuh Manager → Wazuh Indexer → Wazuh Dashboard بحث/عرض/تخزين

٣.٩ خلاصة الفصل

يبين هذا الفصل أن الحل المقترن ليس مجرد استخدام أدوات أمنية منفصلة، بل منظومة طبقية مصممة لالتقاط هجمات SQLi وفق تصنيف محدد ثم نقل نتائج الكشف عبر نقطة تسليم موحدة (eve.json) إلى منصة Wazuh لتصنيفها وعرضها مركزاً. كما أن استخدام نطاق SIDs مخصص (2001xxx) ضمن قواعد Suricata مكّن من بناء مسار تصنيف واضح داخل Wazuh ، ما يسهل التوثيق والبحث وإنتاج مخرجات منظمة يمكن الاعتماد عليها في فصل الاختبارات والنتائج.