مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٩) العدد (١٤) ٢٠٢٥
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (9) No. (14) 2025

تصميم وتنفيذ محرك استجابة ما بعد الكشف لهجمات الشبكات في بيئة SDNباستخدام وحدة التحكم Floodlight

أ. د. م. علي أحمد *

د. م. صادق برو **

م.نيرمين عقول***

(تاريخ الإيداع ٢٠٢٥/٧/٧ . قُبِل للنشر في ٢٠٢٥/١١/٩)

□ ملخّص □

تواجه الشبكات المعرفة برمجياً Software-Defined Networking (SDN) تحديات أمنية متزايدة نتيجة لطبيعتها المركزية وفصل طبقتي التحكم والبيانات، مما يجعل وحدة التحكم هدفاً رئيسياً للهجمات السيبرانية. في هذا البحث، تم اقتراح وتنفيذ محرك استجابة تفاعلي لما بعد الكشف، مدمج ضمن وحدة التحكم Floodlight، يهدف إلى تنفيذ استجابات تلقائية فور تلقي تنبيهات بهجوم من نظام كشف خارجي. يعتمد المحرك على تحليل التنبيهات المستلمة، ومقارنتها بسياسات استجابة محددة مسبقاً، ثم إصدار أوامر لتعديل جداول التدفق في المبدلات باستخدام بروتوكول OpenFlow. تشمل الاستجابات الممكنة كلاً من حظر الحزم أو عزل المصدر باستخدام Virtual Local Area Network (VLAN)، أو تحديد معدل مرور البيانات. تم اختبار النظام في بيئة افتراضية باستخدام Mininet، عبر محاكاة ثلاثة أنواع شائعة من الهجمات هي SYN Flood وUDP Flood و Port Scanning. تم تقييم أداء النظام من خلال معايير تشمل كلاً من زمن الاستجابة ومعدل تقليل الحزم الضارة واستهلاك الموارد ونسبة الإنذارات الكاذبة. أظهرت النتائج قدرة النظام على تنفيذ الاستجابة خلال أقل من ٦٧٠ ميلي ثانية، مع تقليل الحزم الضارة بنسبة تصل إلى ٩٤%، وباستهلاك لا يتجاوز ٣,٤% من موارد وحدة التحكم. كما أظهرت النتائج انخفاضاً في معدل الإنذارات الكاذبة، مما يعكس كفاءة وموثوقية منطق الاستجابة المدمج. يعكس هذا العمل أهمية نقل منطق الاستجابة من الأنظمة الخارجية إلى داخل وحدة التحكم نفسها، ويبرز دور السياسات القابلة للتخصيص في بناء أطر أمنية أكثر مرونة وتكيفاً. كما يفتح المجال أمام تطوير وحدات تحكم ذكية قادرة على اتخاذ قرارات أمنية ذاتية مدفوعة بالبيانات، مما يعزز مفهوم "الدفاع الذاتي" في بيئات SDN

الكلمات المفتاحية:الشبكات المعرفة برمجياً (SDN)، وحدة التحكم Floodlight، محرك استجابة تفاعلي، كشف الهجمات السيبرانية، الاستجابة بعد الكشف.

* أستاذ – قسم هندسة تكنولوجيا الاتصالات– كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سورية.

** مدرس – قسم هندسة تكنولوجيا الاتصالات– كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سورية.

*** طالبة دراسات عليا (دكتوراه) – قسم هندسة تكنولوجيا الاتصالات – كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سورية.

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٩) العدد (١٤) ٢٠٢٥
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (9) No. (14) 2025

Design and Implementation of a Post-Detection Response Engine for Network Attacks in SDN Environments Using the Floodlight Controller

Dr. Ali Ahmad*

Prof. Sadek Pro**

Eng. Nermin Akoul***

(Received 7/7/2025 . Accepted 9/11/2025)

□ ABSTRACT □

Software-Defined Networking (SDN) faces growing security challenges due to its centralized architecture and the separation of the control and data planes, which makes the controller a primary target for cyberattacks. This research proposes and implements an interactive post-detection response engine, integrated within the Floodlight controller, designed to automatically execute response actions upon receiving alerts from an external detection system. The engine analyzes alerts, compares them against predefined response policies, and issues appropriate flow table modification commands to the switches via the OpenFlow protocol. The possible actions include packet dropping, source isolation using Virtual Local Area Network (VLAN), or traffic rate limiting The system was tested in a virtual environment using Mininet, simulating three common attack types: SYN Flood, UDP Flood, and Port Scanning. The engine's performance was evaluated based on response time, malicious traffic mitigation rate, resource consumption, and false positive rate. Results demonstrated the system's capability to initiate a response within less than 670 milliseconds, reduce malicious traffic by up to 94%, and maintain resource consumption of the controller below 3.4%. Additionally, the system showed a low false positive rate, reflecting the reliability and efficiency of the embedded response logic. This work highlights the importance of moving response logic from external systems to the controller itself, and highlights the role of customizable policies in building more flexible and adaptive security frameworks. It also paves the way for the development of intelligent controllers capable of making data-driven, autonomous security decisions, enhancing the concept of "self-defense" in SDN environments.

Keywords: Software-Defined Networking (SDN), Floodlight Controller, Interactive Response Engine, Cyberattack Detection, Post-Detection Response.

* Professor - Department of Communications Technology Engineering - Faculty of Information and Communications Technology Engineering – Tartous University - Syria.

** Lecturer - Department of Communications Technology Engineering - Faculty of Information and Communications Technology Engineering – Tartous University - Syria.

* Postgraduate student (PhD) - Department of Communications Technology Engineering - Faculty of Information and Communications Technology Engineering - Tartous University - Syria.

1- المقدمة

مع تسارع التحوّل الرقمي في البنى التحتية الشبكية، باتت الحاجة ملحّة لاعتماد بنية مرنة وقابلة للبرمجة تُمكِّن من التحكم الذكي والتلقائي في تدفق البيانات. من هنا برزت الشبكات المعرفة برمجياً-Software) Defined Networking - SDN) كأحد أبرز الابتكارات التقنية في العقد الأخير، لما توفره من فصل واضح بين طبقة التحكم وطبقة البيانات، الأمر الذي يسمح بإدارة مركزية وتطوير تطبيقات شبكية متقدمة .ومع هذه القدرات، تنشأ تحديات أمنية جديدة تتعلق بالاعتمادية ومقاومة الهجمات، لا سيما أن تركيز منطق التحكم في وحدة مركزية يجعلها هدفاً للمهاجمين [1-3].

من أبرز التهديدات التي تواجه بنية SDN هي هجمات حجب الخدمة (Denial-of-Service (DoS، وهجمات مسح الطوبولوجيا Topology Discovery Attacks، وتزوير حزم Link Layer Discovery Protocol (LLDP)، والاستغلال الخبيث للرسائل المفتوحة مثل Packet-In [4]. ولطالما انصبت الجهود البحثية في السنوات الأخيرة على تطوير آليات الكشف المبكر عن هذه الهجمات، سواءً باستخدام أدوات تقليدية مثل Snort وBro/Zeek، أو عبر خوارزميات تعلم الآلة. ومع ذلك، فإن المرحلة التالية للكشف، أي مرحلة الاستجابة التلقائية ومعالجة التهديدات فور اكتشافها، لا تزال تعاني من نقص واضح في الدراسات التطبيقية، لا سيما في إطار وحدات التحكم مفتوحة المصدر مثل Floodlight [5-7].

في معظم الحالات، يتوقف النظام الأمني عند حد إرسال تنبيه بوجود تهديد دون أن يُتبع بإجراء عملي يمنع الضرر أو يعزل المصدر الضار،وهذا الفاصل الزمني بين الكشف والمعالجة يمثل ثغرة حرجة يمكن للمهاجمين استغلالها لتحقيق اختراقات متقدمة أو التسبب بانهيار الشبكة [8].

من هذا المنطلق، يهدف هذا البحث إلى سد هذه الفجوة من خلال تصميم وتنفيذ محرك استجابة ما بعد الكشف (Post-Detection Response Engine) يعمل ضمن وحدة التحكم Floodlight ، ويُتيح الاستجابة التلقائية والفورية للهجمات المكتشفة.

يتميز هذا المحرك ببنية معيارية تسمح بربطه بسهولة بأنظمة الكشف الخارجية، ويتيح تنفيذ إجراءات متعددة مثل حظر المصدر أو إعادة التوجيه أو العزل ضمن بيئة افتراضية مبنية على Mininet ،كما يعتمد المحرك على سياسات قابلة للتعديل ديناميكياً، ما يجعله قابلاً للتكيف مع أنواع متعددة من التهديدات ومستويات الخطورة.

1-1 أهمية البحث

تنبع أهمية هذا البحث من كونه يقدم حلاً عملياً قابلاً للتطبيق في بيئات SDN الفعلية، ويساهم في جانب المعالجة بعد الكشف. أبرز الفوائد تشمل:

١. تقليل الفاصل الزمني بين الكشف والمعالجة إلى أقل من ثانية واحدة.
٢. بنية معيارية قابلة للتعديل لدعم أنواع متعددة من الهجمات وطرق استجابة متنوعة.
٣. تنفيذ الاستجابة داخل وحدة Floodlight نفسها، ما يقلل من التأخير الناتج عن الاعتماد على أدوات خارجية.
٤. اعتماد بيئة Mininet مع أدوات اختبار حقيقية مثل hping3و nmap لتقييم الأداء بدقة.

2-1 مشكلة البحث

كيف يمكن تصميم وتنفيذ محرك استجابة ما بعد الكشف، يعتمد على السياسات ومُدمج داخل وحدة تحكم Floodlight، بحيث يوفّر استجابة فورية وفعّالة للتهديدات المكتشفة دون التأثير على أداء وحدة التحكم أو التسبب في إنذارات كاذبة؟

3-1 هدف البحث

اقتراح وتنفيذ محرك استجابة تفاعلي لما بعد الكشف مدمج داخل وحدة التحكم Floodlight، يهدف إلى اتخاذ إجراءات استجابة تلقائية فور تلقي تنبيه بوجود هجوم من نظام كشف خارجي. يعمل المحرك على تحليل التنبيهات المستلمة عبر تنسيق JSON، ومقارنتها بسياسات استجابة محددة مسبقاً، ثم يُصدر أوامر تعديل لجداول التدفق في المبدلات باستخدام بروتوكولOpenFlow ، بما يشمل إجراءات مثل الحظر، العزل باستخدام VLAN، أو الحد من الحزم المسموح بها.

2- الدراسات المرجعية

تناولت العديد من الدراسات السابقة الجوانب الأمنية في بيئة الشبكات المعرفة برمجياً (SDN)، حيث ركزت بدايةً على كشف هجمات DoS و DDoS باستخدام منهجيات متعددة مثل الإنتروبيا وحركة مرور البيانات المنخفضة والزمن، إلى جانب استخدام تقنيات تعلم الآلة [٩]، إلا أن هذه الدراسات توقفت عند حد الكشف دون استكمال الطريق نحو آليات المعالجة داخل وحدات التحكم. وفي سياق آخر، جرى التركيز على مشكلات اكتشاف الطوبولوجيا والتهديدات المرافقة لها [١٠]، دون تقديم حلول مباشرة للتعامل مع الهجمات بعد اكتشافها، بينما ظهرت محاولات لمناقشة تحديات المصادقة وتفويض التطبيقات، مع الإشارة إلى تأثير أحداث Packet-In على قدرة وحدة التحكم [١١]، لكن دون تقديم حلول ديناميكية تستجيب فورياً للهجمات داخل بيئة التحكم.

كما تم التطرق إلى أبعاد السلامة والسرية وتكامل البيانات بين وحدات التحكم، حيث طُرحت أطر عامة لضمان الأمان مثل FROSCO [12] ، لكنها بقيت ضمن مستوى التصميم النظري دون بناء أو اختبار محركات استجابة فعلية. وتواصلت الإشارة إلى وجود فجوات في التعامل مع تهديدات واجهة Northbound [١٣]، دون تجاوز مرحلة التصنيف إلى مرحلة الاستجابة القابلة للتنفيذ. في سياق آخر، أُظهرت قابلية وحدات التحكم مثل Floodlight للتعرض لهجمات DoS والتزوير، مع التوصية باستخدام وسائل حماية خارجية كـ middleboxesأو مصادقات إضافية [١٤]، غير أن ذلك لم يترافق مع تطوير نماذج حماية مدمجة داخل وحدة التحكم نفسها. كما سُلّط الضوء على هجمات طبقة التحكم مثل اختطاف موقع المضيف والتزييف، إلى جانب الإشارة إلى بعض الأطر مثل Floodlight-SE [15]، لكن دون تقديم آليات استجابة عملية ومدمجة داخل بيئة Floodlight.

وفي إطار التصنيف، استُعرضت تهديدات أمنية متعددة على مستوى الطبقات والواجهات المختلفة في SDN [16]، إلا أن هذه التصنيفات بقيت ضمن حدود التحليل النظري دون تقديم نماذج تطبيقية للاستجابة بعد الكشف.

من خلال ذلك، يتبين أن معظم هذه الأعمال ركزت على جانب الكشف أو التصنيف، بينما لم تحظَ آليات الاستجابة التلقائية بعد الكشف، خصوصاً داخل وحدات التحكم مثل Floodlight، باهتمام كافٍ .ومن هنا، يسعى هذا البحث إلى سد هذه الفجوة عبر تقديم نموذج تطبيقي جديد يُدمج مباشرة داخل وحدة التحكم، ليعمل

على اتخاذ قرارات لحظية لعزل أو حظر أو إعادة توجيه التدفقات الضارة فور استلام تنبيه أمني، مقدماً بذلك خطوة عملية نحو تحسين أمان بيئة SDN بصورة متكاملة.

3- الإطار النظري:

يرتكز الإطار النظري على مجموعة من المفاهيم والمكونات التقنية الأساسية التي تشكل قاعدة لفهم النظام المقترح وآلية عمله، وتشمل مفاهيم الشبكات المعرفة برمجياً و طبقاتها و آليات الكشف عن الهجمات، وأسس المعالجة التفاعلية داخل وحدة التحكم. يتناول هذا الإطار المفاهيم الخلفية العلمية التي تُبنى عليها فكرة محرك الاستجابة ما بعد الكشف، وذلك على النحو التالي:

٣-١ الشبكات المعرفة برمجياً SDN

شبكة SDN هو نمط معماري حديث لإدارة الشبكات يهدف إلى فصل طبقة التحكم (Control Plane)عن طبقة البيانات (Data Plane)، مما يتيح للمبرمجين التحكم بالشبكة ديناميكياً من خلال وحدة تحكم مركزية. توفر SDN قدرة عالية على البرمجة، التهيئة الآلية، وتطبيق السياسات الأمنية من خلال واجهات برمجية [17].

يتكوّن نموذج SDN عادة من ثلاث طبقات رئيسية :

١. **الطبقة السفلى :(Southbound Interface)** تنقل الأوامر من وحدة التحكم إلى معدات الشبكة مثل (OpenFlow switches).
٢. **طبقة التحكم :(Control Plane)** تحتوي على منطق اتخاذ القرار في الشبكة وتتمثل في وحدة التحكم.
٣. **الطبقة العليا :(Northbound Interface)** تمثل التطبيقات والخدمات التي تتفاعل مع وحدة التحكم.

١-١-٣الفرق الرئيسي بين SDN والشبكات التقليدية هو البنية التحتية:

شبكات SDN تعتمد على البرمجيات، في حين أن الشبكات التقليدية تعتمد على الأجهزة. نظراً لأن مستوى التحكم يعتمد على البرمجيات، فإن SDN أكثر مرونة من الشبكات التقليدية. فهو يسمح للمسؤولين بالتحكم في الشبكة، وتغيير إعدادات التكوين، وتوفير الموارد، وزيادة سعة الشبكة، كل ذلك من واجهة مستخدم مركزية، دون الحاجة إلى المزيد من الأجهزة. هناك أيضاً اختلافات أمنية بين SDN والشبكات التقليدية. بفضل الرؤية الأكبر والقدرة على تحديد المسارات الآمنة، توفر SDN أماناً أفضل بعدة طرق. ومع ذلك، نظراً لأن الشبكات المعرفة بالبرمجيات تستخدم وحدة تحكم مركزية، فإن تأمين وحدة التحكم يعد أمراً ضرورياً للحفاظ على شبكة آمنة.

في شبكات SDN، يتم فصل البرمجيات عن الأجهزة، حيث يُنقل جزء التحكم المسؤول عن تحديد مسار حركة البيانات إلى البرمجيات، بينما يبقى الجزء المسؤول عن توجيه البيانات فعلياً داخل الأجهزة. يتيح هذا لمسؤولي الشبكات إمكانية برمجة الشبكة والتحكم الكامل فيها بسهولة.

هناك ثلاثة أجزاء لبنية SDN النموذجية:

١. التطبيقات، التي تنقل طلبات الموارد أو المعلومات حول الشبكة ككل.

٢. وحدات التحكم، والتي تستخدم المعلومات من التطبيقات لتحديد كيفية توجيه حزمة البيانات

٣. أجهزة الشبكة، التي تتلقى معلومات من وحدة التحكم حول مكان نقل البيانات.

٣-٢ وحدة التحكم Floodlight

Floodlight هي وحدة تحكم مفتوحة المصدر ومبنية بلغةJava ، تدعم بروتوكولOpenFlow، وتتميز بسهولة التوسعة وإضافة الوحدات الجديدة إليها. تُستخدم بشكل واسع في الأبحاث والتطبيقات الصناعية لتطوير نظم SDN تفاعلية وآمنة [18].

في هذا البحث، يتم تطوير محرك استجابة ما بعد الكشف كجزء مدمج داخل Floodlight، مما يمنحه القدرة على التفاعل مع الهجمات المكتشفة عبر تعديلات فورية في جدول التدفق (Flow Table).

٣-٣ الكشف عن الهجمات فيSDN

الكشف عن الهجمات في بيئة SDN يعتمد على تحليل حركة البيانات الواردة إلى وحدة التحكم عبر رسائل Packet-In. وتُستخدم عدة أساليب للكشف، من بينها [19]:

١. تحليل الإنتروبيا(Entropy-based)

٢. مراقبة الزمن بين الحزم(Time-based)

٣. الكشف القائم على التعلم الآلي(Machine Learning-based detection)

٤. أنظمة كشف التسلل مثل Snort وZeek

٣-٤ الاستجابة ما بعد الكشف (Post-Detection Response)

يمثل هذا المفهوم أساس أنظمة معالجة الهجمات، وتتمثل الفكرة في الانتقال من مرحلة الكشف إلى مرحلة المعالجة، بحيث يتم اتخاذ إجراءات تلقائية مباشرة داخل وحدة التحكم و تشمل [20]:

١. **عزل المصدر الضار :(Quarantine)** باستخدام VLANs أو تغيير مسار التدفّق.

٢. **حظر الحزم :(Flow Drop)** تعديل قواعد OpenFlow لحظر IP محدد.

٣. **القوائم السوداء :(Blacklisting)** تسجيل مصادر التهديد لمنعها مستقبلاً.

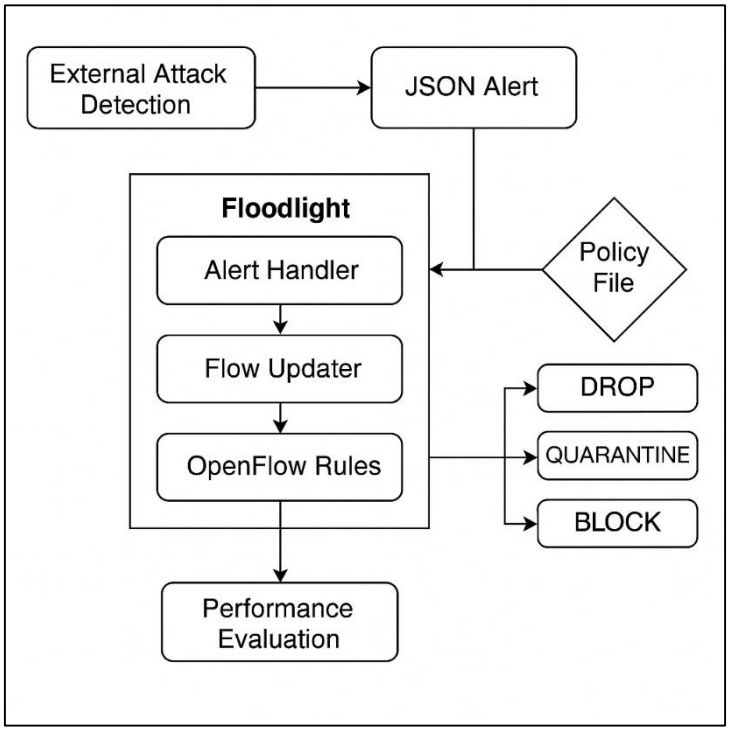
٤. **التفاعل مع السياسات :(Policy-Driven Reactions)** تنفيذ إجراءات تعتمد على نوع الهجوم، ومصدره وخطورته.

4- الأدوات والمنهجية

يركز هذا القسم على الأدوات البرمجية والبيئة التجريبية التي تم استخدامها لتصميم و تنفيذ، وتقييم محرك الاستجابة ما بعد الكشف داخل وحدة التحكم Floodlight، بالإضافة إلى المنهجية المتبعة لاختبار فاعلية المحرك ضد هجمات متعددة. يعرض الشكل (١) تسلسل المعالجة التفاعلية لمحرك الاستجابة ما بعد الكشف، المدمج ضمن وحدة التحكمFloodlight . تبدأ العملية عند استقبال تنبيه أمني صادر عن نظام كشف خارجي، حيث يتم تحويل التنبيه إلى صيغة JSON وإرساله إلى وحدة التحكم. يتم تحليل هذا التنبيه ضمن وحدة مخصصة تُعرف باسم Alert Handler، والتي تتفاعل مع ملف السياسات (Policy File) لتحديد الإجراء المناسب بناءً على نوع الهجوم ومصدره.

بمجرد تحديد الاستجابة الملائمة، تنتقل المعالجة إلى وحدة Flow Updater المسؤولة عن تعديل قواعد التدفق (OpenFlow Rules) على مستوى المبدلات، بحيث يتم تنفيذ أحد الإجراءات التالية: DROP لحظر الحزم القادمة من المصدر الضار، QUARANTINE لعزل المصدر ضمن نطاق VLAN محدد، أو BLOCKلتطبيق قيود مرورية أو حظر جزئي.

بعد تنفيذ الإجراء الأمني، يتم توجيه النتائج إلى وحدة Performance Evaluation لقياس كفاءة المحرك وفق معايير محددة، مثل زمن الاستجابة و نسبة تقليل الحزم الضارة، واستهلاك موارد النظام. يوضح الشكل كذلك تكامل النظام مع ملف السياسات الخارجي الذي يحدد منطق الاستجابة، مما يعزز من مرونة النظام وقابليته للتخصيص.



الشكل (١): المنهجية المتبعة في تصميم واختبار فاعلية المحرك ضد هجمات متعددة.

٤-١ الأدوات المستخدمة

اعتمد هذا البحث في تنفيذه التجريبي على مجموعة من الأدوات المفتوحة المصدر التي تم تنصيبها وتكوينها ضمن بيئة تشغيل موحّدة تعتمد على نظام Ubuntu بالإصدار ٢٠,٠٤ .LTS تم إنشاء البيئة الافتراضية للشبكة باستخدام محاكي Mininet الإصدار d٦٢,٣,٠، والذي يوفر بيئة مرنة لإنشاء شبكات SDN افتراضية تضم مضيفين ومبدلات افتراضية، ويتم التحكم بها من خلال وحدة تحكم خارجية. أُتيح من خلال Mininet تنفيذ هجمات موجهة بين المضيفين في سيناريوهات تحاكي حالات هجوم حقيقية، مما وفر بيئة مثالية لاختبار نظام الاستجابة المقترح.

أما وحدة التحكم المستخدمة فهي Floodlight الإصدار ١,٢، وهي وحدة تحكم مفتوحة المصدر مبنية بلغة Java وتدعم بروتوكولOpenFlow. تمت إضافة وحدة مخصصة إلى الكود الأساسي للوحدة تحت مسمى PostDetectionResponseEngine.java، بحيث تتفاعل هذه الوحدة مع تنبيهات الكشف القادمة من مصادر

خارجية وتقوم بتعديل قواعد التدفق (Flow Rules) على مستوى المبدلات وفقاً لسياسات الاستجابة المحددة مسبقاً.

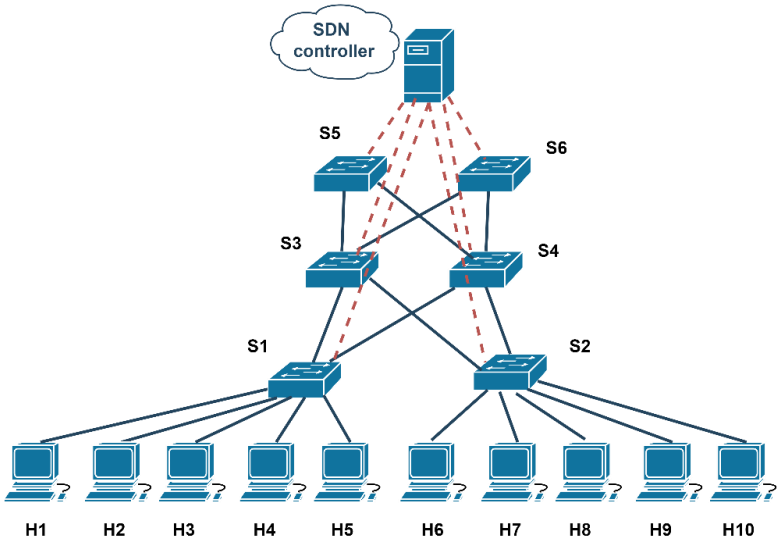
تم استخدام مجموعة من أدوات توليد الهجمات بغرض اختبار فعالية محرك الاستجابة في ظروف هجوم واقعية، شملت أداة hping3 لمحاكاة هجومSYN Flood ، وأداة nmap لتنفيذ مسح للمنافذ (Port Scanning)، بالإضافة إلى أداة nping لتوليد هجمات UDP Flood. ولرصد سلوك الشبكة والتحقق من فعالية الاستجابة، تم استخدام أداة Wireshark لالتقاط الحزم وتحليلها، بالإضافة إلى أدوات مراقبة النظام مثل top و htopلرصد استهلاك وحدة المعالجة المركزية والذاكرة أثناء حدوث الهجمات وتنفيذ الاستجابة.

٤-٢ إعداد بيئة التجربة

تم تصميم بيئة تجريبية افتراضية (الشكل ٢) تحاكي بنية شبكة معرفّة برمجياً (SDN) لاختبار أداء وفعالية محرك الاستجابة ما بعد الكشف المقترح، وذلك باستخدام محاكي Mininet كوحدة أساسية لتوليد العناصر الشبكية وإدارتها. تضمنت البيئة ستة مبدلات افتراضية (Open vSwitch) تمثل نقاط التوجيه الرئيسية في الشبكة، وعشر مضيفين افتراضيين تم توزيعهم بحيث تلعب بعض هذه المضيفات دور المهاجم، بينما يعمل المضيفين الأخرين كأهداف محتملة للهجمات.

تم توصيل وحدة التحكم من نوع Floodlight إلى هذه البيئة من خلال قناة تحكم خارجية مستقلة، لتُمثل مركز القرار الشبكي وتتولى إدارة جداول التدفق على مستوى المبدلات. وقد تم بناء هذه البيئة على نحو يتيح محاكاة سيناريوهات هجوم متنوعة من حيث المصدروالهدف، ونوع التهديد، بهدف اختبار الاستجابة الآلية للمحرك في ظروف مختلفة.

اشتملت التجارب الأساسية على ثلاثة سيناريوهات هجمات رئيسية: تم في الأول توليد هجوم حجب خدمة من نوع SYN Flood انطلاقاً من أحد المضيفين (h1) مستهدفاً مضيفاً آخر(h3)، بينما تم في السيناريو الثاني تنفيذ هجوم UDP Flood بين h2 وh4. أما السيناريو الثالث فقد تضمن تنفيذ مسح منافذ شبكي (Port Scanning)عبر أداة Nmap من المضيف h5 نحوh6 ، وذلك لمحاكاة هجمات استكشافية تهدف إلى جمع معلومات عن الأهداف.



الشكل (٢): شكل توضيحي مختصر عن بنية الشبكة المستخدمة.

تسمح هذه البيئة التجريبية بالتحكم الكامل في ظروف الهجوم ومراقبة أداء وحدة التحكم والاستجابة الناتجة، بما يشمل زمن الاستجابة وفعالية إجراءات الحظر أو العزل، وتأثير الحمل الناتج عن الهجمات على موارد النظام. كما توفر البيئة مرونة في توسيع نطاق التجربة لاحقاً لتشمل أنواعاً إضافية من الهجمات أو سياسات استجابة أكثر تعقيداً.

٤-٣ منهجية التنفيذ

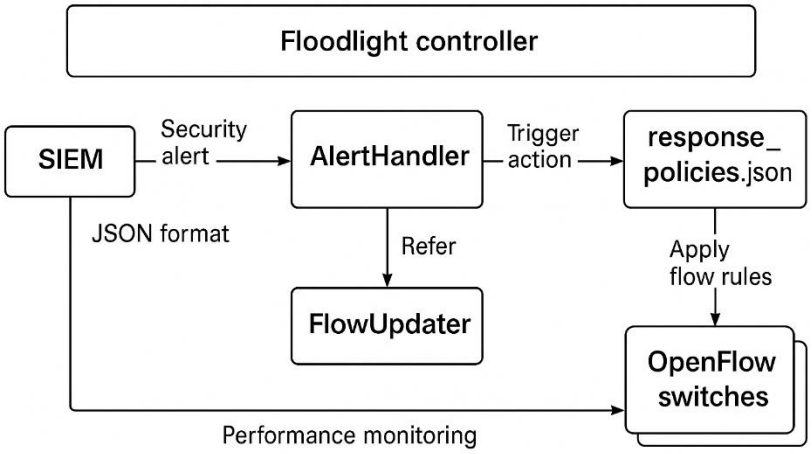
تستند منهجية تنفيذ النظام المقترح الموضحة بالشكل (٣) إلى تصميم دورة معالجة متكاملة تبدأ من استقبال التنبيه الأمني الناتج عن نظام كشف خارجي Snort. تم استخدام نظام الكشف Snort نظراً لقدرته على تحليل الحزم داخل الشبكة في الزمن الحقيقي، وإصدار تنبيهات أمنية دقيقة تتضمن نوع التهديد وعنوان المصدر. تم ربط Snort بأحد المضيفين داخل بيئة Mininet ليعمل كمراقب للشبكة، ويقوم بإرسال التنبيهات بصيغة JSONإلى وحدة التحكم Floodlight عبر واجهة REST مخصصة. يعمل هذا النظام الخارجي على تحليل حركة البيانات داخل الشبكة عبر مراقبة الحزم الواردة إلى وحدة التحكم، واستكشاف الأنماط التي تدل على وجود تهديد محتمل، مثل معدلات تدفق غير طبيعية أو محاولات مسح للمنافذ. وعند اكتشاف تهديد، يتم إصدار تنبيه يحتوي على معلومات مثل نوع الهجوم (على سبيل المثال SYN Flood أوPort Scan)، وعنوان IP الخاص بالمصدر، ووقت حدوث الكشف. يُرسل هذا التنبيه إلى وحدة التحكم باستخدام صيغة JSON (JavaScript Object Notation) ، هو تنسيق نصي مفتوح المصدر لتمثيل البيانات المنظمة بناءً على بناء جملة كائن JavaScript، وهي طريقة منظمة لتمثيل البيانات على شكل أزواج (اسم–قيمة) تُستخدم لتسهيل تبادل المعلومات بين الأنظمة، وذلك عبر واجهة اتصال موحدة.

داخل وحدة التحكم Floodlight، يتم اعتراض التنبيه الأمني بواسطة وحدة متخصصة تُعرف باسم AlertHandler، وهي مسؤولة عن استخراج المعلومات من التنبيه ومقارنتها مع السياسات المحددة مسبقاً في ملف التكوين response_policies.json. يمثل هذا الملف قاعدة معرفية تتضمن مجموعة من القواعد المصممة مسبقاً، تُحدد نوع الإجراء الذي يجب اتخاذه عند كل نوع من أنواع الهجمات. على سبيل المثال، يمكن تحديد أنه في حالة الهجوم من نوع SYN_FLOOD يتم تطبيق إجراء "DROP" لمدة ٦٠ ثانية، وفي حالة PORT_SCANيتم عزل المصدر باستخدام VLAN رقم ٣٠٠، بينما في حالة UDP_FLOOD يتم تطبيق إجراء "BLOCK" إذا تجاوز عدد الحزم ٥٠٠ حزمة في فترة قصيرة. تُستخدم تقنية VLAN (الشبكات المحلية الافتراضية) لعزل حركة مرور مصدر التهديد عن باقي مكونات الشبكة دون فصل مادي. في هذا السياق، يتم نقل الجهاز المهاجم إلى VLAN منفصلة كإجراء عزل (Quarantine)، مما يمنع تواصله مع الأهداف الأخرى ويقلل من تأثير الهجوم.

تقوم وحدة FlowUpdater بترجمة نتيجة القرار إلى أوامر OpenFlow تُرسل إلى المبدلات، لتعديل قواعد التوجيه (Flow Rules) وتنفيذ الإجراء المقرر مثل حظر المصدر و عزله، أو تقليل تدفقه. يتم تطبيق هذه الأوامر بشكل فوري داخل الشبكة، بما يتيح الاستجابة الآنية للتهديد دون الحاجة لتدخل بشري. وتخزين السياسات في ملف خارجي يتيح تعديلها حسب الحاجة دون إعادة تشغيل وحدة التحكم.

أخيراً، تبدأ مرحلة التقييم التجريبي التي تشمل تسجيل زمن الاستجابة منذ لحظة استقبال التنبيه وحتى تنفيذ الإجراء الأمني، إلى جانب مراقبة استهلاك موارد وحدة التحكم (CPU)، وقياس فعالية المحرك من خلال حساب

نسبة الحزم الضارة التي تم منعها أو تقليلها. تساهم هذه النتائج في تأكيد قدرة النظام على توفير استجابة أمنية ذكية تعتمد على البيانات، وتعزز من مفهوم "الدفاع الذاتي" في بيئة SDN.



الشكل (٣): محرك الاستجابة ما بعد الكشف لهجمات الشبكات.

٤-٤ معايير تقييم الأداء

تم اعتماد مجموعة من المعايير الكمية والنوعية لتقييم أداء محرك الاستجابة ما بعد الكشف المقترح، وذلك بهدف قياس كفاءته التشغيلية، وفعاليته في التصدي للهجمات، ومدى تأثيره على موارد النظام. أولى هذه المعايير هي زمن الاستجابة، والذي يُقصد به الفترة الزمنية الفاصلة بين لحظة استلام وحدة التحكم للتنبيه الأمني من نظام الكشف، وبين لحظة تنفيذ إجراء الحظر أو العزل الفعلي على مستوى المبدّل. يمثل هذا المؤشر عنصراً حاسماً في قدرة النظام على احتواء الهجمات في الوقت الفعل و هذا موضح بالعلاقة.

$$T_{response} = t_{action} - t_{alert} \qquad (1)$$

حيث $T_{response}$ زمن الاستجابة، t_{alert} هو توقيت استقبال التنبيه، وt_{action} هو توقيت تنفيذ الإجراء.

كما تم قياس نسبة تقليل التهديد(R)، والتي تُعبّر عن الفرق بين عدد الحزم الضارة التي وصلت إلى الهدف قبل تنفيذ الاستجابة، وعددها بعد تطبيق الإجراء الأمني، مما يوضح مدى فعالية المحرك في الحد من الأثر التشغيلي للهجمات. ولتقييم دقة النظام في تنفيذ الاستجابة، تم احتساب نسبة الإنذارات الكاذبة، والمتمثلة في عدد الحالات التي تم فيها تفعيل إجراء أمني ضد مضيف لم يصدر عنه سلوك عدائي، مما يساهم في تحديد مدى موثوقية السياسات المُطبقة وخوارزميات الكشف المرتبطة بها كما يلي:

$$R_{mitigation} = \frac{P_{before} - P_{after}}{P_{before}} \times 100 \qquad (2)$$

حيث $R_{mitigation}$ نسبة تقليل التهديد، P_{before} هو عدد الحزم الضارة قبل تنفيذ الاستجابة، وP_{after} هو عددها بعد التنفيذ.

أخيراً، شمل التقييم تحليل استهلاك الموارد، وذلك بقياس معدل استخدام وحدة المعالجة المركزية والذاكرة الخاصة بوحدة التحكم أثناء مراحل الكشف والاستجابة. ويهدف هذا المؤشر إلى التأكد من أن النظام لا يضيف عبئاً غير مقبول على بيئة التشغيل، خاصة في أوقات الذروة أو عند حدوث هجمات متعددة في الوقت نفسه. تم جمع البيانات الخاصة بهذه المعايير باستخدام أدوات مراقبة النظام والتحليل الشبكي، وتم تحليلها ضمن بيئة ثابتة تضمن توحيد ظروف القياس لجميع السيناريوهات.وهذا موضح بالعلاقة :

$$FPR = \frac{N_{false\_positives}}{N_{\text{total\_alerts}}} \times 100 \qquad (3)$$

حيث FPR هو نسبة الإنذارات الخاطئة (False positive rate) $N_{false\_positives}$ هو عدد الإنذارات التي استهدفت مصادر غير ضارة، و $N_{\text{total\_alerts}}$ هو العدد الكلي للتنبيهات المُعالجة.

5-النتائج والمناقشة

تم تنفيذ مجموعة من التجارب على بيئة SDN افتراضية باستخدام Mininet ووحدة التحكمFloodlight ، بهدف تقييم كفاءة وفاعلية محرك الاستجابة ما بعد الكشف في التصدي لثلاثة أنواع رئيسية من الهجمات: هجوم SYN Flood، هجوم UDP Flood، وهجوم مسح المنافذ (Port Scanning). تُعد هجمات مسح المنافذ (Port Scanning) هجمات استكشافية تهدف إلى تحديد المنافذ المفتوحة على الخوادم لاستغلالها لاحقاً، حيث يقوم المهاجم بإرسال طلبات سريعة إلى مجموعة واسعة من المنافذ لتحليل استجابات النظام. أما هجمات إغراق الخوادم باستخدام بروتوكول المستخدم (User Datagram Protocol Flood - UDP Flood) وإغراق الخوادم باستخدام حزم المصافحة الأولية(Synchronize Flood - SYN Flood) ، فهي تهدف إلى إغراق الخادم بعدد كبير جداً من الحزم خلال فترة زمنية قصيرة، بهدف استنزاف موارد النظام وتعطيل قدرته على تقديم الخدمات. ويُطلق على SYN Flood هذا الاسم بسبب استغلال حزم SYN في بروتوكول التحكم بالنقل Transmission) Control Protocol – TCP) ضمن عملية المصافحة الثلاثية، بينما يعتمد UDP Flood على إرسال أعداد ضخمة من حزم UDP نحو الخادم دون انتظار الرد، مما يؤدي إلى زيادة الحمل على الشبكة والخادم بشكل كبير.

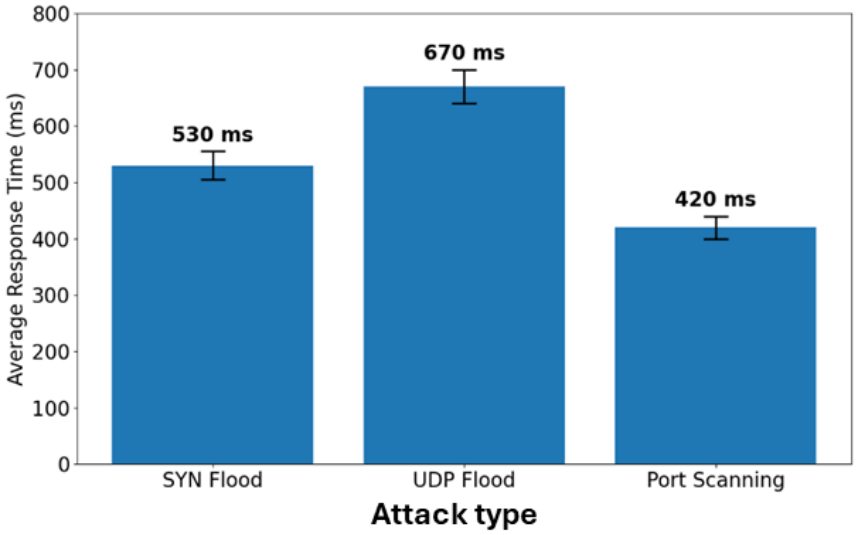
وللحصول على النتائج، تم توليد هذه الهجمات باستخدام أدوات توليد حركة هجومية (مثل hping3 وnmap)، بينما قامت وحدة التحكم بتسجيل التنبيهات والإجراءات المتخذة في سجلات (logs) بصيغة JSON، وتم قياس مؤشرات الأداء، باستخدام أدوات التحليل Wireshark و iperf لمراقبة حركة المرور ومعدلات الحظر بدقة. تم تنفيذ المحاكاة على حاسب بمعالج معالج (CPU) من نوع Intel Core i7 بسرعة ٢,٨ GHz مع ٨ أنوية، لضمان وضوح الظروف التي تم تحتها قياس استهلاك الموارد أثناء التجارب.

٥-١ نتائج زمن الاستجابة

تم حساب متوسط زمن الاستجابة (الشكل ٤) منذ لحظة استقبال التنبيه من وحدة الكشف وحتى تطبيق التعديل على جدول التدفق في المبدّل المستهدف. أظهرت النتائج الأولية لتجارب الأداء أن محرك الاستجابة ما بعد الكشف يتمتع بزمن استجابة فعّال في جميع السيناريوهات التجريبية التي تم تنفيذها ضمن بيئة SDN الافتراضية. تم حساب متوسط زمن الاستجابة بدءاً من لحظة استلام التنبيه الأمني الصادر عن وحدة الكشف، وحتى لحظة تطبيق الإجراء الاستباقي داخل المبدّل المستهدف من خلال تعديل قواعد OpenFlow.

في حالة هجوم حجب الخدمة من نوع SYN Flood، بلغ متوسط زمن الاستجابة حوالي ٥٣٠ مللي ثانية، بينما استغرق النظام حوالي ٦٧٠ مللي ثانية للاستجابة لهجوم UDP Flood، ويُعزى ذلك إلى كثافة الحزم العالية وسرعتها في هذا النوع من الهجمات. أما في سيناريو هجوم Port Scanning، فقد حقق النظام أفضل أداء بزمن استجابة قدره ٤٢٠ مللي ثانية، مما يعكس كفاءته في التعامل مع الهجمات الاستكشافية (Reconnaissance Attacks) التي تكون عادة ذات معدل تدفق أقل مقارنة بهجمات Flooding Attacks.

يشير ذلك إلى أن النظام يُظهر أفضل أداء استجابة عند التعامل مع هجمات Port Scanning، بينما يكون زمن الاستجابة أطول في حالة UDP Flood بسبب طبيعة الهجوم وكثافة تدفق الحزم.



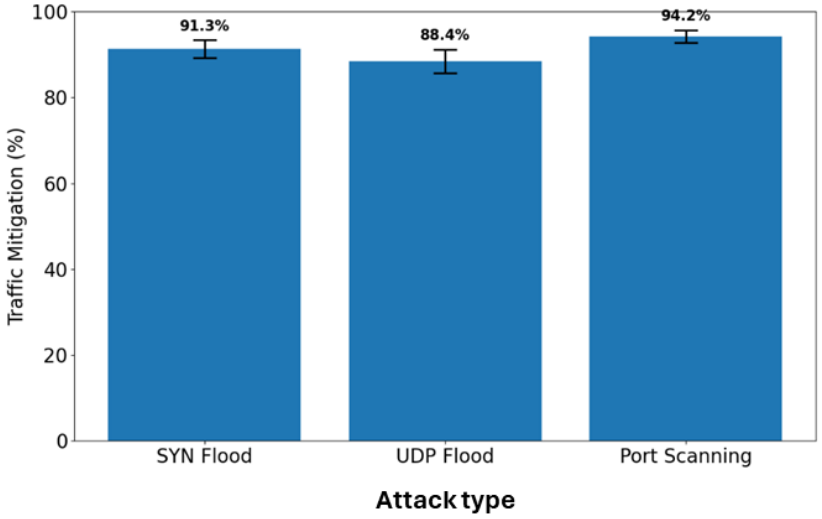
الشكل (4): علاقة نوع الهجوم مع متوسط زمن الاستجابة (مللي ثانية).

٥-٢ فعالية تقليل الهجوم (Traffic Mitigation)

تم تقييم فعالية محرك الاستجابة من خلال قياس قدرته على تقليل عدد الحزم الضارة التي تصل إلى أهدافها داخل الشبكة بعد تفعيل آلية المعالجة (الشكل ٥). تم تقييم فعالية محرك الاستجابة من خلال قياس قدرته على تقليل عدد الحزم الضارة التي تصل إلى أهدافها داخل الشبكة بعد تنفيذ آلية المعالجة. وكما يظهر في الشكل (٥)، اعتمدت المنهجية على مقارنة عدد الحزم الضارة المصنّفة التي استهدفت الهدف قبل تنفيذ الاستجابة، مع عدد الحزم الضارة التي وصلت إلى الهدف بعد تدخل المحرك، بما يسمح بتحديد نسبة النجاح في احتواء التهديدات ومنع انتشارها ضمن بيئة SDN الافتراضية. يُقصد بالحزم الضارة هنا تلك التي تم تصنيفها مسبقاً من قبل نظام الكشف الخارجي استناداً إلى نوع الهجوم (مثل SYN المكرر أو حزم الاغراق)، ولا تشمل جميع الحزم التي تدخل إلى الهدف، بل فقط الحزم المحددة سلفاً كحزم تهديد.

في سيناريو هجوم SYN Flood ، أظهرت النتائج أن المحرك تمكن من تقليل ما نسبته ٩١,٣% من الحزم الضارة، ما يعكس قدرة فعّالة على حظر أو إسقاط التدفقات التي تحتوي على حزم SYN المكررة التي تستهدف استنفاد موارد نظام الهدف. وفي حالة UDP Flood، والتي تُعد من الهجمات الأكثر تطلباً من حيث الكثافة والسرعة، سجلت نسبة تقليل الحزم ٨٨,٤%، وهي نسبة جيدة تشير إلى أن المحرك استطاع التعامل مع الهجوم بالرغم من تدفقه غير المنضبط.

أما في هجوم مسح المنافذ (Port Scanning)، فقد حقق النظام أعلى مستوى من الفعالية بنسبة ٩٤,٢%، وهو ما يمكن تفسيره بأن حركة هذا النوع من الهجمات غالباً ما تكون أقل حدة وتُكتشف بشكل أسرع عبر أنظمة التنبيه، مما يتيح الاستجابة المبكرة قبل اكتمال عملية المسح.



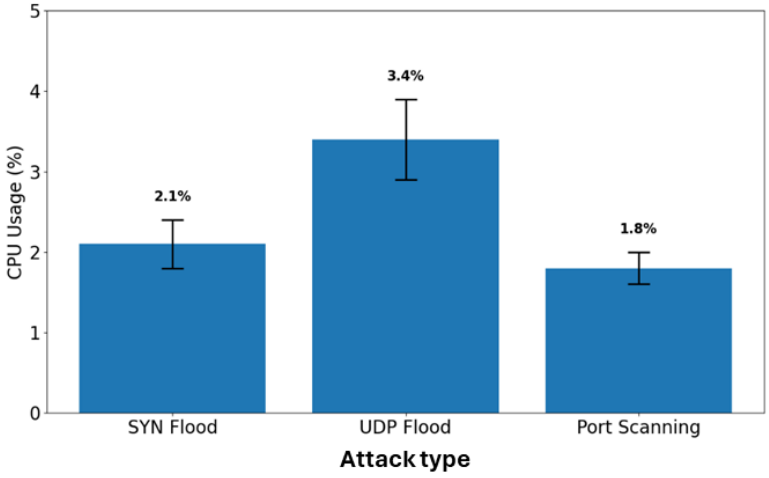
الشكل (٥): نسبة تقليل الحزم الضارة (%).

٥-٣ تحليل استهلاك الموارد

تم تحليل الأثر التشغيلي لمحرك الاستجابة المقترح من خلال مراقبة موارد وحدة التحكم أثناء تنفيذ سيناريوهات الهجوم والاستجابة، وذلك باستخدام أداة مراقبة النظام top ضمن بيئة Ubuntu. وقد تركز التقييم على استهلاك وحدة المعالجة المركزية (CPU) في الشكل (٦)، بوصفها مؤشراً حساساً لمدى كفاءة التنفيذ وفعالية أداء المهام في الزمن الحقيقي، دون الإخلال باستقرار وحدة التحكم أو النظام الشبكي ككل.

أظهرت النتائج أن هجوم SYN Flood تسبب بزيادة معتدلة في استهلاك المعالج، حيث بلغ متوسط الاستخدام حوالي ٢,١% أثناء فترة الاستجابة. أما في حالة هجوم UDP Flood، فقد سُجل أعلى مستوى لاستهلاك الموارد بنسبة ٣,٤%، وهو أمر متوقع نظراً للكثافة العالية والتدفق السريع لهذا النوع من الهجمات، والذي يتطلب معالجة أكبر عدد من التنبيهات وتعديلات متكررة في جداول التدفق.

في المقابل، كان استهلاك وحدة المعالجة في حالة هجوم Port Scanning الأقل، إذ لم يتجاوز ١,٨%، ما يعكس انخفاض حجم الحزم الناتجة عن هذا النوع من الهجمات وطبيعتها الاستكشافية محدودة التأثير.



الشكل (٦): علاقة نوع الهجوم مع نسبة استهلاك وحدة المعالجة المركزية (CPU %).

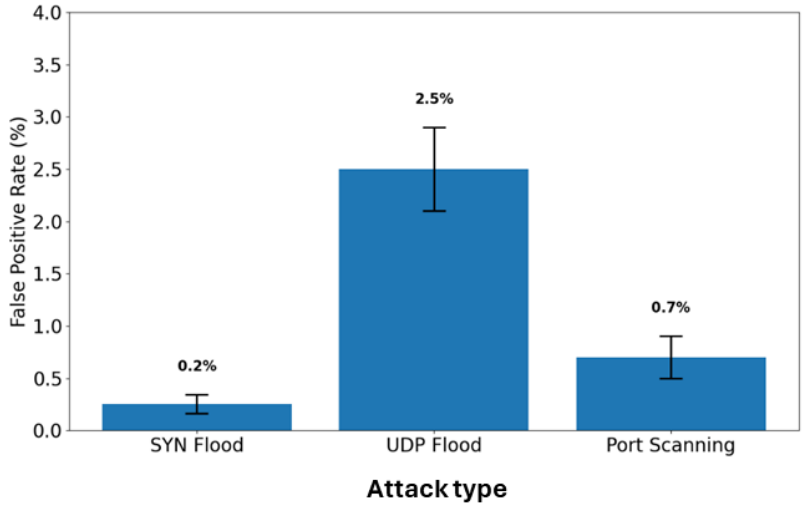
٥-٤ نسبة الإنذارات الكاذبة

ضمن إطار تحليل الأداء الشامل لمحرك الاستجابة ما بعد الكشف، تم تحليل نسبة الإنذارات الكاذبة باعتبارها مؤشراً مهماً لدقة النظام في تطبيق الاستجابة الأمنية بعد التحقق من صحة التهديد. يُقصد بالإنذار الكاذب في هذا السياق كل حالة يتم فيها تفعيل إجراء أمني (مثل الحظر أو العزل) بناءً على تنبيه صادر عن نظام الكشف، في حين يُثبت لاحقاً، من خلال التحقق اليدوي أو مطابقة حركة المصدر مع أنماط الهجوم المعروفة، أن هذا المصدر لا يُظهر سلوكاً عدائياً فعلياً.

تم اعتماد هذا التصنيف بناءً على مقارنة قائمة التنبيهات الصادرة من Snort مع سجل التدفقات (Flow Logs) المسجّلة داخل الشبكة. فإذا تم تطبيق إجراء أمني بحق مضيف لم يصدر عنه حزم مطابقة لأنماط التهديد المصنّفة (مثل SYN متكرر أو حزم غير مكتملة أو سلوكيات مسح منفذ منتظمة)، يتم اعتباره إنذاراً كاذباً. تم التحقق من هذه المقارنة باستخدام قواعد التحقق اليدوي وأداة التحليل الشبكي Wireshark ، للتأكد من أن التنبيه كان مبنياً على تهديد فعلي.

أظهرت النتائج أن المحرك لم يُسجل أي إنذار كاذب خلال تنفيذ هجوم SYN Flood ، بنسبة ٠,٢%، مما يعكس دقة عالية في كشف الأنماط التنبؤية لهذا النوع من الهجمات. أما في حالة UDP Flood ، فقد بلغت نسبة الإنذارات الكاذبة ٢,٥%، وهي النسبة الأعلى بين السيناريوهات الثلاثة، ويُعزى ذلك إلى احتمال تداخل بعض الحزم الشرعية المرتفعة المعدل مع أنماط الإغراق. في حالة Port Scanning ، كانت نسبة الإنذارات الكاذبة محدودة عند ٠,٧% فقط، وهو ما يُظهر تميزاً في دقة الكشف لحركة المسح الاستكشافية التي تُكتشف غالباً قبل اكتمال عملية المسح.

تُعد هذه النسب ضمن الحدود المقبولة في نظم كشف التسلل [1,2]، وقد تم تقديرها بناءً على مراجعة التدفقات الفعلية ومقارنتها بالقوائم السوداء الناتجة عن وحدة التحكم، ما يوفر أداة لتقدير (FPR)، وهي مؤشر يستخدم عادة في تقييم الأنظمة الأمنية الذكية.



الشكل (٧): علاقة نوع الهجوم مع نسبة الإنذارات الكاذبة (%).

6-الاستنتاجات و التوصيات

٦-١الاستنتاجات

بناءً على ما تم عرضه وتحليله، يمكن تلخيص أبرز الاستنتاجات كما يلي:

١. أظهرت نتائج التجارب أن النظام تمكن من تقليل حركة المرور الضارة بنسبة 91.3% في SYN Flood، 88.4%في UDP Flood ، و 94.2% في Port Scanning، مع الحفاظ على استخدام منخفض للمعالج بلغ 2.1%، ٣,٤%، و١,٨% على التوالي .كما حقق النظام معدلات إنذار كاذب منخفضة بلغت 0.2% فيSYN Flood ، ٢,٥% في UDP Flood ، و٠,٧% فيPort Scanning ، بينما بلغ متوسط زمن الاستجابة 530 مللي ثانية فيSYN Flood ، و٦٧٠ مللي ثانية في UDP Flood ، و٤٢٠ مللي ثانية في Port Scanning.

٢. إن تنفيذ وحدة استجابة مدمجة ضمن وحدة التحكم Floodlight يعزز بشكل مباشر من قدرة الشبكة على التصدي للهجمات دون تدخل يدوي أو تأخير.

٣. النظام المقترح أثبت كفاءة عالية في تقليل التهديدات، واستجابة شبه لحظية، واستهلاك محدود للموارد، مما يجعله مناسباً للنشر في بيئات SDN الحقيقية.

٤. قابلية تخصيص سياسات الاستجابة عبر ملفات خارجية يزيد من مرونة النظام ويقلل من الحاجة لتعديلات عميقة في البنية الداخلية لوحدة التحكم.

٥. فعالية المحرك ترتبط ارتباطاً مباشراً بجودة ودقة نظام الكشف المستخدم، مما يفتح الباب أمام تحسينات مستقبلية قائمة على الذكاء الاصطناعي أو التحليل السلوكي.

٦-٢ التوصيات المستقبلية

استناداً إلى نتائج البحث والملاحظات التجريبية، يُوصى بالآتي لتطوير المحرك المقترح وتوسيع تطبيقاته:

١. دمج وحدات كشف داخلية قائمة على التعلم الآلي لتحسين دقة التنبيهات وتقليل معدلات الإنذارات الكاذبة.

٢. توسيع النظام ليشمل معالجات استجابة متعددة المستويات تشمل الطبقة التطبيقية وواجهات المستخدم.

٣. دعم محركات استجابة موزعة في وحدات تحكم متعددة (Multi-Controller Systems) لتعزيز الاستجابة في البيئات الواسعة النطاق.

٤. تطوير واجهات مرئية (Dashboard) لمراقبة أداء النظام وتحليل الحوادث بشكل لحظي.

٥. اختبار المحرك في بيئات هجينة (Hybrid SDN) أو شبكات حقيقية للتحقق من قدرته على التكيف مع المتغيرات الواقعية والتحديات التشغيلية.

7- المراجع

1. Bajenaid, A., Khemakhem, M., Eassa, F. E., Bourennani, F., Qurashi, J. M., Alsulami, A. A., & Alturki, B. (2025). *Towards Robust SDN Security: A Comparative Analysis of Oversampling Techniques with ML and DL Classifiers*. Electronics (2079-9292*), 14*(5).

2. Farooq, M. S., Riaz, S., & Alvi, A. (2023). *Security and privacy issues in software-defined networking (SDN): A systematic literature review*. Electronics, *12*(14), 3077.

3. Wu, Y. J., Hwang, P. C., Hwang, W. S., & Cheng, M. H. (2020). *Artificial intelligence enabled routing in software defined networking*. Applied Sciences, 10(18), 6564.

4. Arevalo-Herrera, J., Camargo Mendoza, J., Martínez Torre, J. I., Zona-Ortiz, T., & Ramirez, J. M. (2025*). Assessing SDN Controller Vulnerabilities: A Survey on Attack Typologies, Detection Mechanisms, Controller Selection, and Dataset Application in Machine Learning*. Wireless Personal Communications, 1-37.

5. Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., & Hong, W. C. (2021). *Internet of things: Evolution, concerns and security challenges*. Sensors, *21*(5), 1809.

6. Alabdulatif, A., & Thilakarathne, N. N. (2024). *A novel cloud-enabled cyber threat hunting platform for evaluating the cyber risks associated with smart health ecosystem*s. Applied Sciences, *14*(20), 9567.

7. Faria, J. (2021). *Designing Network Security Tools for Home Users* (Doctoral dissertation, Doctoral dissertation, Worcester Polytechnic Institute, Worcester, USA, 2021. Available online: h [ps://digital. wpi. edu/downloads/pv63g338d (accessed on 9 January 2025)).

8. Dawood, M., Tu, S., Xiao, C., Alasmary, H., Waqas, M., & Rehman, S. U. (2023). *Cyberattacks and security of cloud computing: a complete guideline*. Symmetry, *15*(11), 1981.

9. Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y. W., & Sanjalawe, Y. K. (2020*). Detection techniques of distributed denial of service attacks on software-defined networking controller–a review*. IEEE Access, *8*, 143985-143995.

10. Khan, S., Gani, A., Wahab, A. W. A., Guizani, M., & Khan, M. K. (2016). *Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art.* IEEE Communications Surveys & Tutorials, 19(1), 303-324.

11. Ahmad, I., Namal, S., Ylianttila, M., & Gurtov, A. (2015). *Security in software defined networks: A survey.* IEEE Communications Surveys & Tutorials, 17(4), 2317-2346.

12. Xie, J., Guo, D., Hu, Z., Qu, T., & Lv, P. (2015). *Control plane of software defined networks: A survey.* Computer communications, *67*, 1-10.

13. Rauf, B., Abbas, H., Usman, M., Zia, T. A., Iqbal, W., Abbas, Y., & Afzal, H. (2021). *Application threats to exploit northbound interface vulnerabilities in software defined network*s. ACM Computing Surveys *(CSUR), 54*(6), 1-36.

14. Ochoa-Aday, L., Cervelló-Pastor, C., & Fernández-Fernández, A. (2020). *Self-healing and SDN: bridging the gap*. Digital Communications and Networks, 6(3), 354-368.

15. Rahouti, M., Xiong, K., Xin, Y., Jagatheesaperumal, S. K., Ayyash, M., & Shaheed, M. (2022*). SDN security review: Threat taxonomy, implications, and open challenge*s. Ieee Access, *10*, 45820-45854.

16. Jimenez, M. B., Fernandez, D., Rivadeneira, J. E., Bellido, L., & Cardenas, A. (2021). *A survey of the main security issues and solutions for the SDN architectur*e. IEEE Access, *9*, 122016-122038.

17. Hussain, M., Shah, N., Amin, R., Alshamrani, S. S., Alotaibi, A., & Raza, S. M. (2022). *Software-defined networking: Categories, analysis, and future directions*. Sensors, *22*(15), 5551.

18. Braun, W., & Menth, M. (2014*). Software-defined networking using OpenFlow: Protocols, applications and architectural design choices*. Future Internet, 6(2), 302-336.

19. Reddy, S., & Shyam, G. K. (2022). *A machine learning based attack detection and mitigation using a secure SaaS framework*. Journal of King Saud University-Computer and Information Sciences, *34*(7), 4047-4061.

20. Wijesekara, P. A. D. S. N., & Gunawardena, S. (2023, August*). A comprehensive survey on knowledge-defined networking*. In Telecom (Vol. 4, No. 3, pp. 477-596). MDPI.