Damascus University Journal for engineering sciences
Vo 40 number (4) -2025: 366-388

مجلة جامعة دمشق للعلوم الهندسية
المجلد40 العدد(4) – 2025: 366–388

دراسة تقييميّة على الأدوات المعتمدة على تحليل حركة المرور الشّبكي في كشف الاقتحامات

محمّد يامن أيمن حلّاق*[1] غسّان حسن شدّود[2] علي محمّد حمّاد[3]

*[1]. طالب ماجستير – قسم النّظم والشّبكات الحاسوبيّة – كليّة الهندسة المعلوماتيّة – جامعة دمشق
yamen.hallak@damascusuniversity.edu.sy

[2]. باحث في هيئة الطّاقة الذرّيّة السّوريّة . g.chaddoud@damascusuniversity.edu.sy.

[3]. مدرّس في قسم النّظم والشّبكات الحاسوبيّة – كليّة الهندسة المعلوماتيّة – جامعة دمشق
ali.hammad@damascusuniversity.edu.sy.

الملخص:

ازداد اهتمام مشغّلي الشّبكات ومزوّدي الخدمات الشّبكيّة بتحليل المرور الشّبكي بشكل كبير خلال العقد الماضي، ويُعدّ تحليل المرور الشّبكي وسيلة لفحص البيانات أثناء تدفّقها عبر الشّبكة. وبعبارة أخرى، فإنّ تحليل المرور الشّبكي هو إجراء تقوم به إدارة الشّبكة لفحص البيانات التي تخصّ مستخدمي الشّبكة بهدف تحليلها، ومعالجة الازدحام في الشّبكة، والقيام بعمليات إحصائيّة، ومراقبة سلوك المستخدمين، والكشف عن الهجمات. ودرءاً من الأعمال التّخريبيّة، فقد تمّ توظيف تحليل المرور الشّبكي في نظم الكشف عن الاقتحامات، والتي تعتمد على تحليل المرور الشّبكي أوّلاً، وثانياً البحث عن مؤشّرات وأدلّة الاقتحامات والتي قد يكون أحد هذه المؤشّرات هو البحث عن أنماط المرور الشّبكي الشّاذّة عن الأنماط المعروفة. إنّ الكشف عن مؤشّرات الهجوم ليس مهمّة سهلة، وتزداد صعوبة بوجود التّطبيقات الحديثة كتطبيقات التّواصل الاجتماعي التي تهتم بالخصوصيّة وتحافظ على سرّية البيانات المنقولة، ووجود الأقنية المخفيّة التي تسمح بتسريب البيانات والتي قد يكون من ضمنها برمجيّات خبيثة، والتي غالباً ما يتمّ تجاهلها من قبل أنظمة الكشف عن الاقتحام المعروفة. نتيجة لذلك تُعدّ دراسة أدوات تحليل المرور الشّبكي وتقييم فعالية وأداء أنظمة الكشف عن الاقتحامات مهمّان بالنسبة لمشغّلي الشّبكات كخطوة أولى في اختيار النّظام المناسب، وتالياً تطوير النّظام الموجود لديهم.

جرى في هذا البحث دراسة لبعض أدوات تحليل المرور الشّبكي مثل: TCPDump, Wireshark، وتقييماً لنظاميّ كشف الاقتحامات: Snort و Suricata من ناحية: كشف الهجمات المختلفة التي تستهدف سرّيّة وسلامة وإتاحيّة البيانات والموارد الحاسوبيّة بما فيها هجمات القنوات المخفيّة؛ توزيع الحمل على نُوى المعالجة؛ استنزاف الموارد الحاسوبيّة؛ قدرة نظم الكشف على استقبال الطّرود الشّبكيّة ومعالجتها، وإسقاط الطّرود والتّنبيهات؛ بالإضافة لدراسة إمكانيّة تحسين تلك الأنظمة.

الكلمات المفتاحيّة: تحليل المرور الشّبكي، نظم كشف الاقتحامات، القنوات المخفيّة.

تاريخ الايداع:2022/9/8
تاريخ القبول: 2022/12/13

Evaluation Study on Network Traffic Analysis-Based Tools in Intrusions Detection

Mohammad Yamen Ayman Hallak*[1] Ghassan Hassan Chaddoud[2] Ali Mohammed Hammad[3]

*[1]. Master Student – Department of Computer Systems and Networks – Faculty of Information Technology Engineering – Damascus University – yamen.hallak@damascusuniversity.edu.sy.

[2]. Researcher at Syrian Commission of Atomic Energy. g.chaddoud@damascusuniversity.edu.sy.

3. Lecturer at Department of Computer Systems and Networks – Faculty of Information Technology Engineering, Damascus University ali.hammad@damascusuniversity.edu.sy.

Received: 8/9/2022
Accepted: 13/12/2022

Abstract:

Network operators and network service providers are increasingly interested in network traffic analysis over the past decade, and network traffic analysis is a method of examining data as it flows through the network. In other words, network traffic analysis is a procedure carried out by the network administration to examine the data belonging to the network users for the purpose of analyzing it, dealing with network congestion, perform statistical operations, monitor user behavior and detect attacks. To prevent sabotage, network traffic analysis has been employed in intrusion detection systems, which rely on network traffic analysis first, and secondly, searching for indicators and evidence of intrusions, one of which may be the search for abnormal network traffic patterns for known patterns.

Detecting indicators of an attack is not an easy task, and it becomes more difficult with modern applications such as social media applications that care about privacy and maintain the confidentiality of transmitted data, and the presence of covert channels that allow data leakage, which may include malicious software, which is often ignored by well-known intrusion detection systems. As a result, studying network traffic analysis tools and evaluating the effectiveness and performance of intrusion detection systems are important for network operators as a first step in choosing the appropriate system, and subsequently developing their existing system.

In this research, a study of some network traffic analysis tools such as: TCPDump and Wireshark, and an evaluation of the two intrusion detection systems: Snort and Suricata in terms of: detect various attacks targeting the confidentiality, integrity and availability of data and computer resources including covert channels attacks; load balancing on processing cores; consumption of computer resources; the ability of detection systems to receive and process network packets, and drop network packets and alerts; in addition to studying the possibility of improving these systems.

Keywords: Network Traffic Analysis, Intrusion Detection Systems, Covert Channels.

المقدّمة:

إنّ تنامي الميّزات التي تقدّمها تقنيّات الاتصالات الحديثة من ناحية ازدياد السّعة (Capacity) والإتاحيّة (Availability)، جعلت سلوك مستخدم الشّبكة أكثر تعقيداً عن السّابق، ممّا جعل المستخدم يميل لاستخدام خدمات أكثر على الانترنت [1]، وبالتّالي فإنّ احتمال حدوث الانتهاكات أصبح أكبر، ممّا جعل الحاجة ملحّة لمراقبة وتحليل الأحداث على الشّبكة، حيث أنّ تحليل المرور الشّبكي من ناحية تحليل الطّرود (Packets) يعتبر تقنية أساسيّة في التّدقيق الجنائي (Forensic) للشبكة، ويمكن استخدام هذه التّقنية للعثور على آثار السّلوك الخبيث عبر الانترنت [2].

يمكن تحليل بيانات الشّبكة وفصل حركة مرور الشّبكة حسب النّوع، وذلك باستخدام برامج مصمّمة خصّيصاً لهذا الغرض، تسمّى بمحلّلات الطّرود (أو ملتقطي الطّرود (Packet Sniffers)، وأحياناً محلّلات الشّبكات (Networks Analyzers) [2]، بحيث أنّ تلك الطّرود الشّبكيّة تحوي معلومات هامّة جدّاً يمكن تحليلها لجمع أدلّة تخدم التّحقيق الجنائي للشبكة، من حيث اكتشاف حركة المرور الخبيثة وسلوكها على الشّبكة، بما في ذلك محاولات الاقتحام، وإساءة استخدام الشّبكة، وتحديد البرمجيّات الضّارّة بما فيها برمجيات الفدية (Ransomware) [2] . كما ذُكر في [3] أحد الفوائد الهامّة لأدوات التقاط الطّرود، والتي تتجلّى في استخدامها للكشف عن الاقتحامات (Intrusions) الحاصلة في الشّبكة، بحيث تقوم أنظمة كشف الاقتحام (Intrusion Detection Systems: IDS) بقراءة وتحليل كلّ الطّرود المارّة بالشّبكة، ومقارنتها مع أنواع محدّدة من الهجمات الشّبكية، وعند كشف هجمة معيّنة، يقوم نظام الكشف عن الاقتحام بإعلام مسؤول الشّبكة.

إنّ القيام بالفحص السّابق في الوقت المناسب يُعتبر أمراً ضروريّاً من أجل تمكين مسؤول الشّبكة من تقييم نتائج تحليل حركة المرور، واتّخاذ القرار المناسب بخصوص الإجراءات المضادّة المناسبة [4].

ومن النّاحية التّعريفيّة، وكما تمّ ذكره في [5]، فإنّ كشف الاقتحام هو الكشف عن الأفعال التي تحاول المساس بسريّة مورد أو سلامته أو إتاحيّته، وحسب [6] فإنّ عمليّات الاقتحام تحدث عادةً عن طريق المهاجمين الذين يصلون إلى الأنظمة من الانترنت، أو من خلال إساءة المستخدمين لصلاحيّات الوصول الممنوحة لهم، أو محاولتهم الحصول على امتيازات وصول أكبر غير مصرّح لهم بها.

وانتقالاً للقنوات المخفيّة التي كان لها شأن كبير في البحث العلمي نظراً لإمكانيّة استغلالها من قِبل المهاجمين في إحداث خرق في أنظمة حماية الشّبكة، وتسريب المعلومات من داخل مؤسّسة أو حاسوب ما إلى الخارج من خلال تلك القنوات، فتناولت الأبحاث طرق إنشاء القنوات المخفيّة، وكيفيّة استغلالها من قِبل المهاجمين، بالإضافة لسبل الوقاية منها.

إنّ أوّل مرّة تمّ فيها ذكر مفهوم القناة المخفيّة كان من خلال ملاحظة قالها لامبسون عام 1973 [7]، والذي اعتبر أنّ القنوات المخفيّة هي القنوات الغير مخصّصة لنقل المعلومات على الإطلاق، مثلها مثل عبء برامج الخدمة (Service Programs) على أداء النّظام [8]. منذ حينها لاقت القنوات المخفيّة اهتماماً كبيراً في مجال أمن المعلومات [9]، ليظهر تعريفاً آخر من قِبل وزارة الدّفاع الأمريكية عام 1985 والذي يصف القناة المخفيّة على أنّها قناة اتصال يمكن استغلالها من قِبل إجراء ما (Process) لنقل المعلومات بطريقة تنتهك

سياسة أمن النظام (10).

لقد تمّ تقسيم القنوات المخفيّة إلى نوعين اثنين أُشير لهما في [7] بالاستفادة من [10]، وهما: 1) قنوات التّخزين المخفيّة (Storage Covert Channel)، بحيث تقوم إحدى الإجراءات بالكتابة بشكل مباشر أو غير مباشر على أحد الأماكن، بينما يقوم الإجراء الآخر بالقراءة من ذات المكان [9]؛ 2) قنوات التّوقيت المخفيّة (Timing Covert Channel)، بحيث يتمّ فيها تعديل خصائص الوقت بغرض إخفاء المعلومات، ويمكن تحقيق ذلك من خلال إحداث تعديل في التأخير بين الطّرود [9]، أو عن طريق تعديل استخدام الموارد بمرور الوقت (مثل استخدام وحدة المعالجة المركزيّة) [11].

لقد ذُكر في [7،11] العديد من البروتوكولات وحقولها التي من الممكن توظيفها لنقل البيانات بشكل مخفيّ، ونظراً للتعدّد الكبير في أشكال القنوات المخفيّة، ولأنّنا لسنا بصدد سرد احتمالات تشكّل القنوات المخفيّة بقدر اهتمامنا بطريقة تعاطي نظم كشف الاقتحامات مع تلك القنوات، فقد تمّ في هذه الدّراسة الاعتماد على إخفاء البيانات في الحقلين TTL و TOS من البروتوكول IP كمثال يعبّر عن قنوات التّخزين المخفيّة، أمّا بالنّسبة لقنوات التّوقيت المخفيّة، فقد تم التعرّض لأحد الأمثلة التي سنذكرها لاحقاً. إنّ مايثير القلق هو أنّ التطبيقات الحديثة وتطبيقات التّواصل الاجتماعي ومشاركة البيانات تستخدم في عملها تلك البروتوكولات [12]، ممّا يشكّل تهديداً نظراً للبيانات الخبيثة التي قد تحويها تلك القنوات، أو من خلال تسريب البيانات الشّخصيّة.

إنّ فرصة الخصم بإنشاء قناة مخفيّة تعتبر جيّدة طالما هناك طريقة للوصول إلى شبكة الانترنت [7]، كما أنّ الحجم الهائل للبيانات المنقولة عبر الانترنت بواسطة البروتوكولات TCP/IP يجعل تلك البروتوكولات بيئة حاضنة مناسبة للقنوات المخفيّة [9]. إنّ فهم القنوات المخفيّة بأنواعها وكذلك التّهديدات المحتملة لها هو أمر ضروري من أجل حماية الموارد الحسّاسة للشّبكة ضدّ هجمات القنوات المخفيّة [13]، وبالتّالي يجب الكشف عن تلك القنوات والتّعامل معها في الوقت المناسب.

إنّ الكشف المثالي عن الاقتحامات ليس هدفاً بسيطاً يمكن تحقيقه بسهولة، نظراً للتعقيد والتطوّر السّريع في كلٍّ من الهجمات والأنظمة [14]، بالإضافة لعدم ضرورة تطوّر مواصفات الأجهزة مع تطوّر الهجمات، ولظهور القنوات المخفيّة وهجماتها، وبالتّالي ينبغي الوصول لنتائج مقارنة كافية لنظاميّ الكشف المُقترحين في هذا البحث (Snort, Suricata) تؤمّن لنا القدرة على اختيار نظام كشف الاقتحامات المناسب لبُنية شبكيّة مطروحة.

ستشمل هذه الدّراسة معايير تقييم عديدة، وهي: كشف الهجمات المختلفة التي تستهدف سرّيّة وسلامة وإتاحيّة البيانات والموارد الحاسوبيّة بما فيها هجمات القنوات المخفيّة؛ دقّة التّنبيهات التي تُصدِرها نظم الكشف ومدى قربها من طبيعة الهجوم الحاصل؛ توزيع الحمل على نُوى المعالجة؛ استنزاف الموارد الحاسوبيّة بما فيها وحدة المعالجة المركزيّة (CPU) وذاكرة الوصول العشوائيّ (RAM)؛ قدرة نظم الكشف على استقبال الطّرود الشّبكيّة وتحليلها (معالجتها)، وإسقاط الطّرود والتّنبيهات؛ بالإضافة لدراسة إمكانيّة تحسين تلك الأنظمة من ناحية: الاستجابة للهجمات التي فشلت سابقاً في كشفها بما فيها هجمات القنوات المخفيّة؛ القدرة على استقبال الطّرود الشّبكيّة وتحليلها وتقليص عدد الطّرود التي يتمّ إسقاطها؛ بالإضاقة لتوزيع الحمل على نُوى المعالجة.

في تتمّة البحث، يُظهر القسم (2) دراسة مرجعيّة للأعمال المُشابهة. القسم (3) يسرد دراسة تقييميّة لنظم الكشف وفق

المعــايير السّــابقة وســرداً لأدوات تحليــل المــرور الشّــبكي المُســتخدمة. القســم (4) يــذكر مقترحــات لتحســين نظــاميّ الكشـف. وأخيـراً القسـم (5) يضـمّ نتـائج البحـث وخلاصــة النّتائج التي تمّ الوصول لها.

2. دراسة مرجعيّة:

تمّ دراسة عدّة أعمال مشابهة بغرض مقارنة مجال بحثها مع بحثنـا الحـاليّ، واستكشـاف المعـايير التـي تطرّقت إليهـا فـي دراستها المُقارِنـة، وإيجـاد مجـالات لـم تغطّيهـا لمناقشـتها فـي بحثنـا. فقد تـمّ فـي [15] عـرض مقارنـة مـابين نظـم كشـف الاقتحامات مفتوحة المصدر المعتمدة على الشّبكة (بما فيها نظـاميّ Snort و Suricata) وفـق معـايير محـدّدة، مثـل: السّرعة، المرونـة، دعـم تعدّديّـة نظـم التشـغيل، ودعـم تعدديّـة النّياسـب، إلّا أنّ الوضـوح لـم يكـن كافيـاً، والاختصـار كـان كبيراً في بعض الأفكار الهامّة التي طرحتها كأوجه مقارنة.

فـي حين ذُكـر فـي [16] مقارنـة مـا بـين النّظـامين Snort و Suricata وفـق عـدّة معـايير أهمّهـا: الدقّـة، القـدرة علــى استثمار البيئـة متعدّدة نُوى المعالجـة، وإسقاط التّنبيهات، إلّا أنّ الدّراسة لم تُناقش مقترحات لتحسين تلك الأنظمـة، كمـا أنّ التقييم اقتصر على هجمات محصورة بـ Microsoft.

بالانتقــال لـــ [17]، حيــث تــمّ ذكــر أنّ نظــاميّ Snort و Suricata غير مناسبين تماماً لأغراض الكشف عن القنوات المخفيّة، لأنّ بنيتهما تقوم علـى آليّـة كشف ثابتـة من خـلال مطابقة القواعد، بينما القنوات المخفيّة لا يمكن التّعامل معها مــن خــلال مطابقــة القواعــد فقــط (لتعــدّد أشــكالها وكثــرة احتمالاتهـا كمـا أسـلفنا سـابقاً)، ومـا ميّـز هـذا العمـل بشـكل صـريح من وجهـة نظـر بحثنـا كونـه يتنـاول القنوات المخفيّـة، إلا أنّ المقارنة كانت سطحيّة لحدّ ما وتفتقر للنتائج العمليّة.

بالمتابعــة فــي [18]، حيــث تــمّ تنــاول آليّــة مقترحــة لكشــف القنــوات المخفيّــة مــن خــلال تركيبــة مــن: جــدار النّــار (Firewall)، نظام كشف الاقتحامات، ونظام كشف القنوات المخفيّة (Covert Channel Detection System). حيث أنّ هذا العمل اقترح أنّ هنالك نظام مخصّص لكشف القنوات المخفيّة من خلال آليّات معيّنة يتمّ تعريفها عليه، بينما كان دور جدار النّار ينحصر ضمن ترشيح (Filtering) الطّرود الشّبكيّة حسب القواعد التي يتمّ تعريفها، ودور نظام كشف الاقتحامات يتجلّى في كشف الهجمات الحاصلة من خلال مطابقة القواعد. وبالتّالي فإنّ هذا العمل لم يُناقش قدرة نظم كشف الاقتحامات في حدّ ذاتها في كشف القنوات المخفيّة، كما أنّه لم يذكر أيّ نوع من نظم الكشف التي تمّ التّعامل معها.

أمّا بالنّسبة لـ [19] فقد تمّت مقارنة ما بين نظاميّ Snort و Suricata مـن ناحيــة البُنيــة، وتــمّ شــرح بُنيــة كــلّ منهمــا، بالإضـافة للمقارنـة وفقاً لبعض المعايير، مثل: دعم تعدّديّة نُــوى المعالجــة، والتوسّــعيّة، لكــن اقتصــرت مقارنــة الأداء واستنزاف الموارد على الأعمال السّابقة فقط، بالإضافة لأنّ المقارنات كانت مختصرة لحدّ ما.

بالانتقال لـ [20] حيث تمّ التطرّق أيضاً إلى نظاميّ الكشف Snort و Suricata والمقارنــة فيمــا بينهــا وفــق: الاســتجابة للهجمــات، كيفيّــة إعــداد تلــك الأنظمــة، دعــم التحــديثات ومخرجات التّنبيه، بغرض الوصول إلى توصية استخدام أحد تلك الأنظمة ضمن المؤسّسات الصّغيرة والمتوسّطة، إلا أنّ هـذه الدّراسـة لـم تُنـاقش معـايير مقارنـة هامّـة مثـل: الأداء، انشـغاليّة المـوارد، وغيرهـا. فـي حين ذُكِـرَ فـي [21] مقارنـة كمّيّــة مــا بــين نظــاميّ Snort و Suricata وفــق معــايير الأداء، التوسّعيّة، والتّعامل مع تعدّدية نُوى المعالجة، حيث تــمّ إخضــاع النّظــامين السّــابقين للمقارنــة ضــمن ظــروف إعـدادهما الافتراضـي، وضـمن ظـروف إعداد محسّـن لكـلّ منهما، إلا أنّ الدّراسة لم تتطرق لمعايير تحسين أخرى مثل تحسين استجابة نظم الكشف للهجمات.

أمّا بالنّسبة لـ [22]، فقد تمّ سرد مقارنة ما بين نظاميّ Snort و Suricata مــن ناحيــة الأداء، والدقّــة ضــمن بيئــة آلــة افتراضـيّة مشـغولة (Busy)، بحيـث كانـت المقارنـة تشـمل

النّسبة المئويّة لاستخدام وحدة المعالجة المركزيّة، الذّاكرة، والشّبكة، وذلك بعد إخضاع تلك الأنظمة لحركة مرور خبيثة، لكنّها لم تذكر معايير مقارنة أخرى مثل استجابة نظم الكشف لهجمات القنوات المخفيّة.

بالمتابعة في [23]، حيث تمّ التطرّق لنظام Snort وغيره من الأنظمة المعتمدة على القواعد في كشف الاقتحامات، بحيث كان الحديث حول تقييم الأنظمة المقترحة من ناحية تعريضها لحركة مرور شبكي تحوي هجمات ويب، إلّا أنّ هذه الدّراسة لم تتعرّض لنظام Suricata رغم أهميّته وتنافسيّته مع نظام Snort، كما أنّها لم تناقش هجمات محتملة أخرى سوى هجمات الويب. أمّا في [24] فقد جاء فيها تقييماً لنظاميّ Snort و Suricata من ناحية استجابتهما لبعض الهجمات المطبقّة، ومن ناحية دقّة التنبيهات التي تطلقها تلك الأنظمة. إلا أنّ هذه الدّراسة لم تُناقش هجمات القنوات المخفيّة، كما أنّها لم تتعرّض لمعايير مقارنة أخرى ولا لمناقشة تحسين تلك الأنظمة.

من خلال الاطلاع على الأعمال السّابقة التي تصبّ في سياق بحثنا، نلاحظ أنّ بعض الدّراسات اقتصرت في مقارنتها على معايير مقارنة محدّدة دون الخوض بمعايير أخرى وبشكل أخصّ القنوات المخفيّة، وحتّى في الدّراسات التي تطرّقت لدور نظم كشف الاقتحامات في كشف القنوات المخفيّة، فقد كانت إمّا مختصرة وتفتقر للبرهان العمليّ الواضح، وإمّا تقترح الاعتماد على أنظمة مُساعدة. ومن جهة أُخرى، فإنّ الدّراسات السّابقة التي قمنا بسردها قد تناولت إصدارات قديمة من نظم الكشف، في حين أنّ هنالك تحسينات قد طرأت على الإصدرات الحديثة لابُدّ من تقييمها. إنّ تلك الجوانب التي لم تتناولها الدّراسات السّابقة في مجال بحثها، تجعل الفرصة أمامنا مواتية لاستغلالها والتطرّق إليها ومناقشتها وتوظيفها في هدف البحث، للوصول إلى دراسة تتميّز عن سابقاتها بالمواضيع التي ناقشتها، بالإضافة لجمع عدّة معايير مع بعضها في دراسة واحدة.

3. دراسة تقييميّة:

سيتمّ في هذا القسم تقديم مقارنة من خلال اختبارات عمليّة لنظاميّ الكشف Snort و Suricata وفق المعايير التي تمّ ذكرها سابقاً، وسنبدأ هذا القسم بتعريف مختصر لكلّ نظام على حدا.

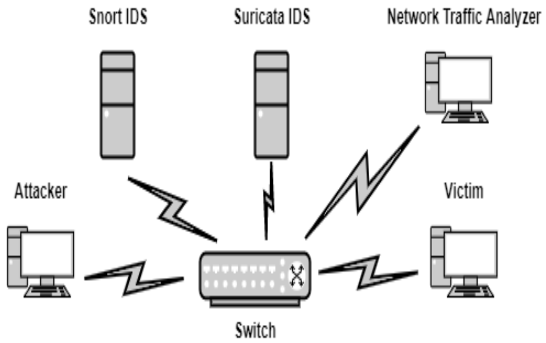
بالنّسبة لنظام Snort، فهو نظام كشف ومنع الاقتحامات مفتوح المصدر، قائم على تحليل حركة المرور الشّبكي في الزّمن الحقيقي والبحث ضمنها عن الهجمات، مع إتاحة إمكانيّة تسجيل الطّرود الشّبكيّة. يعتمد في عمله على تعريف سلسلة من القواعد (تواقيع – Signatures) التي تعرّف نشاط الشّبكة الضّار من أجل مطابقة حركة المرور الشّبكي معها لاحقاً، وكشف الاقتحامات من خلال تنبيه المستخدمين [25]. قمنا في هذه الدّراسة بتنصيب نظام Snortذو الإصدار الثّالث، والذي يقدّم حسب موقع نظام Snortالكثير من التطلّعات منها: دعم تعدديّة النّياسب (Multi-Threading) في معالجة الطّرود الشّبكيّة، والإقلاع السّريع [25].

أمّا بالنّسبة لنظام Suricata، فهو نظام كشف ومنع الاقتحامات ومراقِب للشّبكة مفتوح المصدر، يتعامل مع مجموعة من التّواقيع لمطابقتها مع التّهديدات المعروفة، وانتهاكات السّياسات الأمنية، والسّلوك الضّار. يمتاز بالأداء العالي، بحيث نسخة واحدة منه قادرة على فحص عدّة غيغا بت من حركة المرور الشّبكي، بالإضافة لكونه يدعم تعدديّة النّياسب، ولإمكانيّة إعداده بخصوص أوضاع التّشغيل التي سيعمل عليها (Run Modes) [26] والتي سنتكلّم عنها لاحقاً خلال مناقشة تحسين الأنظمة. قمنا في هذه الدّراسة بتنصيب نظام Suricata ذو الإصدار السّادس، والذي تناول حسب موقع نظام Suricata تحسينات في الأداء، والمتانة، والثّبات [26].

وبدءاً بتقييم نظاميّ الكشف، سنتطرّق لناحية استجابتها للهجمات المختلفة، بحيث لم يكن الهدف هنا تقييم أكبر عدد

من الهجمات المحتملة، بقدر تناول أشهر الهجمات التي تتناول سرّيّة وسلامة و إتاحيّة البيانات والموارد الحاسوبيّة (بما فيها هجمات القنوات المخفيّة).

يُظهر الشّكل (1) طبولوجيا الشّبكة المقترحة التي تمّ التجريب عليها، والتي تتألّف من آلات افتراضيّة (Virtual Machines) تُعبّر عن نظاميّ الكشف، والمهاجم، والضّحيّة، ومحلّل حركة المرور الشّبكي[1]. مع الإشارة إلى أنّه من الممكن أن يكون لدينا أكثر من مهاجم، وأكثر من ضحيّة.



الشّكل(1) طبولوجيا الشّبكة المقترحة

1.3. تقييم الاستجابة للهجمات الحاسوبيّة:

يُظهر الجدول (1) تقييماً لنظاميّ الكشف Snort و Suricata من ناحية الاستجابة لبعض الهجمات المقترحة التي تستهدف سرّيّة وسلامة وإتاحيّة البيانات والموارد الحاسوبيّة. وبطريقة مشابهة لحدّ ما لـ [20] تمّ إعطاء علامات التّقييم لكل نظام على حدا وفق الآتي:

- **الخانة الأولى:** علامة للكشف.
- **الخانة الثّانية:** علامة لقاء دقّة التّنبيه الذي يُصدره وقربه من طبيعة الهجوم.
- **الخانة الثّالثة:** علامة عند تفوّقه على النّظام الآخر في عدد التّنبيهات التي أطلقها.

❖ **ملاحظات حول تنفيذ الهجمات:**

تمّ تنفيذ هجمات: استغلال ثغرة MS12_020[2]، التحكّم عن بُعد بحاسوب الضّحيّة[3]، Slowloris[4] (حجب الخدمة عن مخدّم الويب) من خلال بيئة Metasploit[5]. وهجمات: مسح الشّبكة، TCP Scan[6]، UDP Scan[7] تمّت باستخدام الأداة Nmap[8]. واختراق بيانات تسجيل الدّخول تمّ باستخدام الأداة Hydra[9]. في حين هجمات: Code Injection[10]، PHP Object Injection[11]، XSS[12] من خلال بيئة الاختبار btslab[13]. وأخيراً هجوم SQL Injection[14] تمّ باستخدام الأداة SQLmap[15] على نظام Kali Linux[16] وبواسطة البيئة btslab.

---

[1] يمكن الاطلاع على مواصفات الآلات الافتراضيّة التي تمّ التّجريب عليها، وعلى تحليل المرور الشّبكي، وعلى قواعد الكشف التي تمّ تعريفها، بالإضافة للبرنامج الذي تمّت كتابته لتجريب القنوات المخفيّة من خلال الرابط: https://github.com/YamenHallak/Evaluation-Study-on-Network-Traffic-Analysis-Based-Tools-in-Intrusions-Detection

[2] https://docs.microsoft.com/en-us/securityupdates/securitybulletins/2012/ms12-020

[3] Hacking Windows 7 Using Metasploit Backdoor and Post Exploitation - Blog | Securium Solutions www.securiumsolutions.com/blog/hacking-windows-7-using-metasploit-backdoor-and-post-exploitation

[4] Hansen, R.: Slowloris HTTP DoS (2009). ha.ckers.org/slowloris/ (October 22, 2014).

[5] https://www.metasploit.com/

[6] TCP SYN (Stealth) Scan (-sS) | Nmap Network Scanning|www.nmap.org/book/synscan.html.

[7] UDP Scan (-sU) | Nmap Network Scanning|www.nmap.org/book/scan-methods-udp-scan.html.

[8] https://nmap.org/

[9] https://github.com/vanhauser-thc/thc-hydra

[10] https://owasp.org/www-community/attacks/Code_Injection

[11] https://owasp.org/www-community/vulnerabilities/PHP_Object_Injection

[12] https://owasp.org/www-community/attacks/xss/

[13] https://github.com/CSPF-Founder/btslab

[14] https://owasp.org/www-community/attacks/SQL_Injection

[15] https://sqlmap.org/

[16] https://www.kali.org/

تُظهر النّتائج في الجدول (1) كيف تفوّق نظام Suricata على نظام Snort في استجابته لبعض الاقتحامات، ويعود احتماله الأكبر لأنّ عدد قواعد الكشف في نظام Suricata أكبر ممّا هو عليه الحال في نظام Snort، حيث أنّ عدد القواعد في نظام Snort كان حوالي 16,000 قاعدة، في حين كان عدد القواعد في نظام Suricata حوالي 23,000 قاعدة.

وانتقالاً لتقييم استجابة نظم كشف الاقتحامات للقنوات المخفيّة، كان لابدّ من التطرّق لأدوات التقاط طرود الشّبكة التي تمّ الاستفادة منهما في تحليل حركة المرور الشّبكي، وفي تحديد سمات الهجمات.

هناك العديد من أدوات التقاط الطّرود (Sniffers) المتاحة مجّاناً، وتعتبر TCPDump [27] أداة شائعة الاستخدام تسمح بالنّظر في طرود الشّبكة وإجراء بعض التّحليلات الإحصائيّة لملفّات التّتبع (Trace Files)، وتمّ في [28] ذكر شعبيّة الأداة TCPDump التي يمكن استخدامها كواحدة من أخفّ وأفضل أداوت التقاط حركة المرور الشّبكي، والتي توفّر الكثير من الخيارات التي تساعد في تفصيل البيانات التي تمّ التقاطها. في حين يضيف Wireshark [29] واجهة مستخدم رسوميّة سهلة الاستخدام إلى TCPDump، وتتضمّن العديد من التّوقيعات المروريّة (Traffic Signature) التي يمكن استخدامها في التّحديد الدّقيق للتطبيقات المستندة إلى الحمولة (Payload-Based Application) [1].

ورغم شهرة الأداة Wireshark، إلّا أنّه حسب المقارنة التي تمّ القيام بها في [28]، فقد تفوّقت الأداة TCPDump على الأداة Wireshark من ناحية الاستخدام القليل للذّاكرة والطّاقة وقدرة المعالجة، في حين كانت الغلبة لل Wireshark في قدرته على تحليل الطّرود الملتقطة وفي سرعة الالتقاط. في دراستنا هذه، تمّ استخدام كلا الأداتين السّابقتين لمساعدتنا في تحليل المرور الشّبكي.

وبالعودة للقنوات المخفيّة، وعلى وجه التّحديد قنوات التّخزين المخفيّة، فقد تمّ اختيار مكتبة Winpcap[1] لما توفّره من المرونة الكبيرة في البرمجة وتخصيص التّطبيق بما يخدم البحث والاستغناء عن الأدوات الجاهزة ومحدوديّة التّعامل معها، بحيث تمّ تمثيل الطّرد الشّبكي وإخفاء البيانات ضمن بعض حقول البروتوكول IP (وهما: TTL, TOS) لتشكيل قناة تخزين مخفيّة، والتأكّد من ذلك من خلال تحليل المرور الشّبكي. وكمثال على ذلك، فيما يلي تحليل للمرور الشّبكي يُظهر كيف تمّ حقن قيمة محرفيّة (Character) تمثّل الحرف "A" ضمن حقل TTL من ترويسة البروتوكول IP (وبطريقة مشابهة يمكن إخفاء محرف في الحقل TOS):

[1] https://www.winpcap.org/

الجدول(1) – تقييم استجابة نظم الكشف للهجمات المقترحة

الهجوم	نظام الكشف	الكشف	دقّة التنبيه	تفوّق على النظام الآخر	المحصّلة
استغلال ثغرة MS12_020 في نظام Windows	Snort	0	0	0	0
	Suricata	1	1	1	3
مسح (استكشاف) الشّبكة	Snort	1	1	#	2
	Suricata	1	1	#	2
اختراق بيانات تسجيل الدّخول	Snort	0	0	0	0
	Suricata	1	1	1	3
التحكّم عن بُعد بحاسوب الضحيّة	Snort	0	0	0	0
	Suricata	1	1	1	3
TCP Scan	Snort	1	1	#	2
	Suricata	1	0	#	1
UDP Scan	Snort	1	1	#	2
	Suricata	1	0	#	1
SQL Injection	Snort	1	1	1	3
	Suricata	1	1	0	2
XSS					
1- حقن مقطع برمجي في حقل مخصّص لدخل المستخدم	Snort	0	0	0	0
	Suricata	1	1	1	3
2- حقن قيمة في رابط صفحة الويب	Snort	0	0	0	0
	Suricata	0	0	0	0
Code Injection	Snort	0	0	0	0
	Suricata	1	1	1	3
PHP Object Injection	Snort	0	0	0	0
	Suricata	0	0	0	0
حجب الخدمة عن مخدّم الويب	Snort	0	0	0	0
	Suricata	1	0	1	2
المحصّلة	Snort				9
	Suricata				23

ملاحظة: عدد التنبيهات يعتمد على الحالة، مثال: بالنسبة لهجوم الشبكة، فهناك أجهز فيها عدد اكبر من المناف\ المفتوحة، فتكون عدد تنبيهاتها أكثر.

20:01:29.404562 IP 192.168.18.173.5000 > 192.168.18.171.80: Flags [.], ack 0, win 64240, length 0
0x0000: 4504 0028 0000 40004106 9323 c0a8 12ad E..(..@.A..#....
0x0010: c0a8 12ab 1388 0050 0000 0000 0000 0000 ....P........
0x0020: 5010 faf0 fa62 0000 0000 0000 0000 P....b........

أمّا بالنسبة لقنوات التّوقيت المخفيّة، فقد تمّت الاستعانة بالمثال المذكور في [13]، بحيث تمّ التطرّق لخوارزميّة بسيطة تقوم بتسريب المعلومات من خلال استغلال ساعة نظام التّشغيل، وترسل بتّات البيانات التي نرغب بتسريبها بناءً على خصائص السّاعة كالتّالي:

```
for each current _bit in data
while (true)
milliseconds := current _time();
if (current _bit == (milliseconds / INT) mod 2)
invoke _receiver();
break;
```

بحيث هنا يتمّ تنبيه المستقبِل لكي يقرأ السّاعة عندما يتوافق بت السّاعة مع بت البيانات التي نريد تسريبها. أيضاً تمّ التأكد من خلال تحليل المرور الشّبكي كيف تمّ إعلام المستقبِل بالبيانات التي تمّ تسريبها كما تمّ إرسالها بالضّبط، وفيما يلي مثالاً تمّ تطبيقه بغرض إرسال البيانات التّالية من المرسِل إلى المستقبِل:{1,0,0,1}.

عند المستقبِل من خلال تحليل المرور الشّبكي نلاحظ الخرج النّاتج أدناه، بحيث عندما تكون ثواني السّاعة ذات طبيعة زوجيّة، فهذا يعني أنّ هناك بت قيمته (0)، وعندما تكون ثواني السّاعة ذات طبيعة فرديّة، فهذا يعني أنّ هناك بت قيمته (1):

02:19:59.745656 IP w7-2.5000 > WIN-OS32J7UJUTG.localdomain.80: Flags [.], ack 0, win 64240, length 0
02:20:00.777036 IP w7-2.5000 > WIN-OS32J7UJUTG.localdomain.80: Flags [.], ack 1, win 64240, length 0
02:20:02.839291 IP w7-2.5000 > WIN-OS32J7UJUTG.localdomain.80: Flags [.], ack 1, win 64240, length 0
02:20:03.840075 IP w7-2.5000 > WIN-OS32J7UJUTG.localdomain.80: Flags [.], ack 1, win 64240, length 0

وبالتّالي نلاحظ أنّ بتّات البيانات التي تمّ استقبالها هي: {1,0,0,1}، أي كما تمّ إرسالها بالضّبط.

بالتّجريب العملي، يظهر لدينا فشل نظم كشف الاقتحامات السّابقة في كشف إخفاء البيانات ضمن الحقلين TTL و TOS الخاصّين بالبروتوكول IP، وكذلك في كشف قنوات التّوقيت المخفيّة ذات الشّكل المقترح سابقاً، رغم نجاح تلك القنوات في تسريب البيانات، وبالتّالي كانت المحصّلة هنا هي (0) لكلا النّظامين بالنّسبة لاستجابتهما لهجمات القنوات المخفيّة السّابقة.

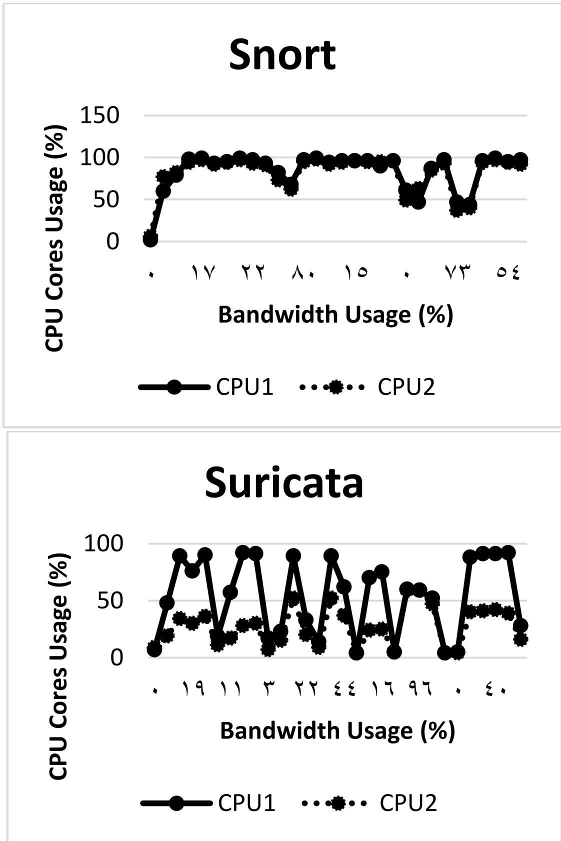
2.3. تقييم استنزاف الموارد الحاسوبيّة:

بمتابعة تقييم نظاميّ الكشف، كان من الجدير تقييم فعاليّة نظاميّ الكشف في توزيع الحمل على نُوى المعالجة من خلال إخضاعهما إلى حركة مرور شبكي متفاوتة (تمّ توليدها بالاعتماد على البرنامج Packet Sender[1])، وتسجيل النّسب المئويّة التي يستنزفها كلّ نظام كشف لنُوى المعالجة.

يُظهر الشّكل (2) كيفيّة استنزاف نُوى المعالجة وتوزيع الحمل عليها في كلّ من نظاميّ الكشف، ويتّضح لنا أنّ توزيع الحمل على نواتيّ المعالجة في نظام Snort كان أكثر تجانساً ممّا عليه الحال في نظام Suricata، بحيث نلاحظ التّجانس في توزيع الحمل في كثير من النّقاط، إلّا أنّ استنزاف نواتيّ المعالجة كان في كثير من الأحيان ~100%. في حين كان استنزاف نواتيّ المعالجة في نظام

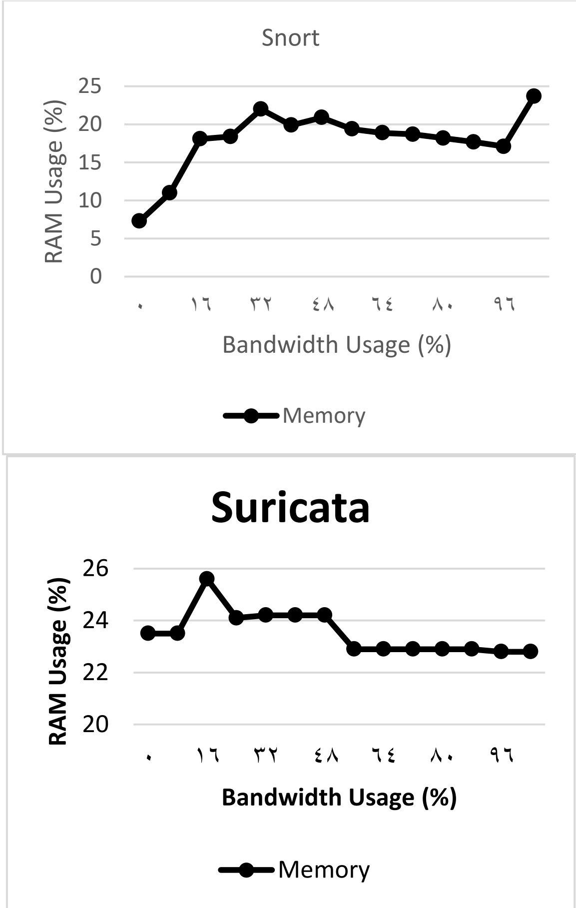
[1] https://packetsender.com/

Suricata أقلّ مما هو عليه الحال في نظام Snort، في حين عبء المعالجة الأكبر كان من نصيب النّواة الأولى.
يأتي تفوّق نظام Snort من ناحية تجانس توزيع الحمل على نواتيّ المعالجة، من خلال التّحسينات التي طرأت على نظام Snort في الإصدار الثّالث منه، والتي أصبحت تدعم تعدّديّة التّياسب في معالجة (تحليل) الطّرود الشّبكيّة المتدفّقة والتي يقوم باستقبالها، إلّا أنّ تلك التّحسينات قد زادت استنزاف نواتيّ المعالجة بشكل ملحوظ مقارنة بنظام Suricata.



الشّكل(2) استنزاف نُوى المعالجة وتوزيع الحمل عليها

وبمتابعة تقييم استنزاف نظاميّ الكشف للموارد، كان من الحري بنا مقارنة استنزاف نظاميّ الكشف للذّاكرة بغية التحقّق فيما إذا كانت الموارد كافية، وهل يتمّ استنزافها بشكل مناسب أم لا. يُظهر الشّكل (3) نسبة استنزاف ذاكرة الوصول العشوائي حسب تدفّق المرور الشّبكي.
من الواضح كيف كان استنزاف نظام Snort للذّاكرة يرتفع بشكل تدريجي عند زيادة تدفّق المرور الشّبكي من 0%~ إلى 32%~، وفيما بعد تذبذب استنزاف الذّاكرة بشكل بسيط زيادةً ونقصاناً. في حين نظام Suricata قد كان استنزافه للذاكرة أكثر من نظام Snort في كثير من النّقاط وبشكل أخصّ عند تدفّق مرور شبكي بسيط من 0%~ إلى 32%~. يمكن تبرير ذلك بسبب التّحسينات التي تمّت على نظام Snort والتي تقوم على دعم تعدديّة التّياسب في معالجة الطّرود، مما يُحرّر مساحة أكبر من الذّاكرة بغرض معالجة الطّرود [25]. وعلى الرّغم من تفوّق نظام Snort على نظام Suricata في قلّة استنزاف الذّاكرة، إلّا أنّ كلا النّظامين لم يستنزفا أكثر من 26% من إجمالي سعة الذّاكرة.



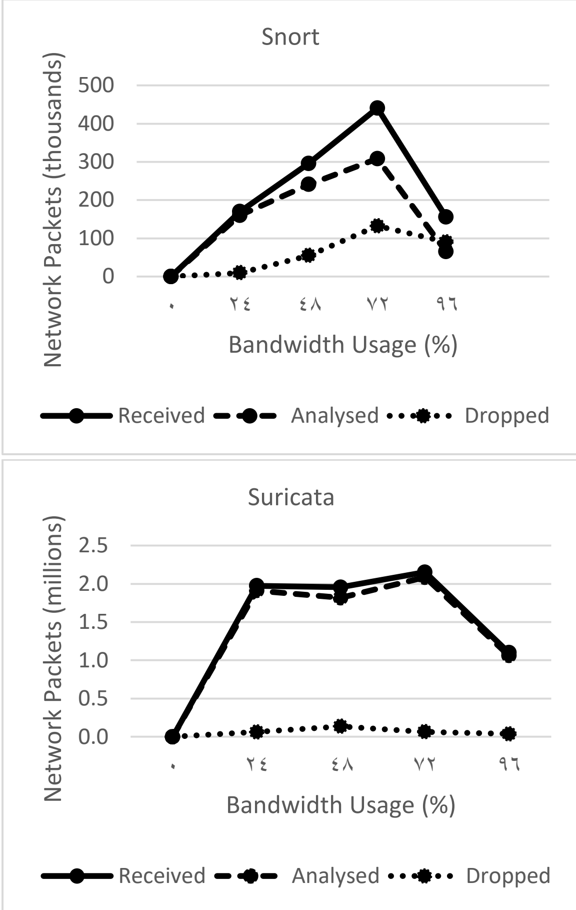
الشّكل(3) استنزاف ذاكرة الوصول العشوائي تبعاً لتدفّق المرور الشّبكي

3.3. تقييم قدرة نظم الكشف على استقبال الطّرود الشّبكيّة وتحليلها، وإسقاط الطرود والتنبيهات:

من الجوانب الهامّة جدّاً التي يمكن تناولها في تقييم أنظمة الكشف، بحيث يجب من النّاحية النّظريّة المثاليّة على نظام الكشف أن يقوم باستقبال كافّة الطّرود الشّبكيّة التي تمرّ على مقطع الشّبكة الذي يقوم نظام الكشف بمراقبته، ومعالجة تلك الطّرود التي قام باستقبالها بحثاً عن علامات اقتحام، مع تلافي إسقاط أيّ طرد شبكي نظراً لأنّه من الممكن أن يشكّل هذا الطّرد إقتحام وبالتّالي تهديداً للموارد. إلّا أنّ هذا الأمر كما اتضح من النّاحية العمليّة لبحثنا أنّه غير محقّق، لكن يمكن أن يكون نظام كشف أفضل من الآخر في معيار التقييم هذا.

لقد تمّ الاختبار بإخضاع نظاميّ الكشف لأحجام تدفّقات مرور شبكي متزايدة، وتسجيل معدّلات استقبال الطّرود الشّبكيّة، ومعدّل معالجة تلك الطّرود، ومعدّل إهمالها (إسقاطها) من قِبل النّظام المدروس.

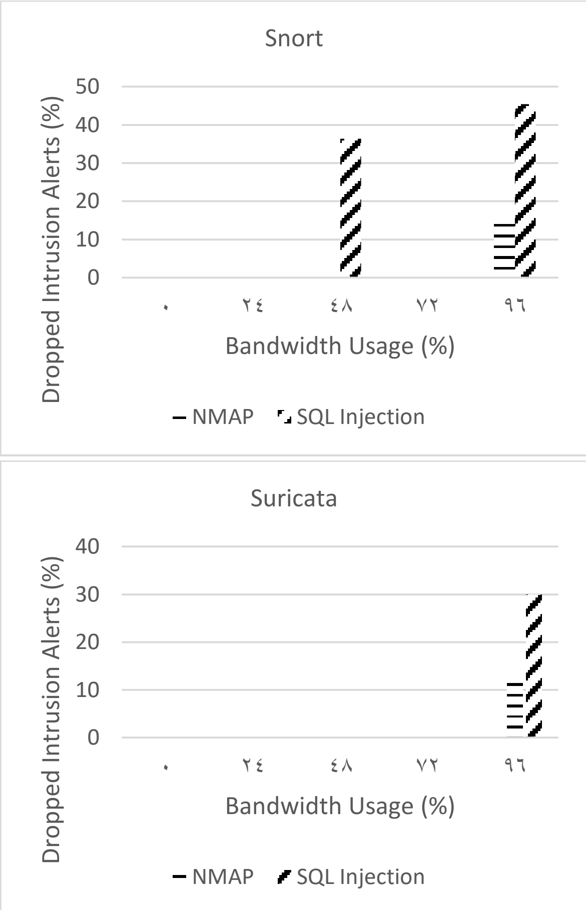
يوضّح الشّكل (4) الحالة المذكورة، فمن الواضح كيف تفوّق نظام Suricata على نظام Snort من خلال ملاحظة تجانسه الكبير جدّاً بين الطّرود التي استقبلها والطّرود التي قام بتحليلها، كما تفوّق نظام Suricata أيضاً من خلال انخفاض معدّل الطّرود التي قام بإسقاطها والتي تشكّل حوالي 7% من إجمالي عدد الطّرود التي قام باستقبالها، مقارنة بنظام Snort الذي أسقط ما يُقارب 58% من إجمالي الطّرود التي قام باستقبالها. ومن الجدير بذكره هنا، أنّ عدد الطّرود الشّبكيّة التي استقبلها نظام Snort كانت تتراوح بين (400 – 500) ألف طرد شبكي، بينما عدد الطّرود الشّبكيّة التي استقبلها نظام Suricata كانت تتراوح بين (2 – 2.5) مليون طرد شبكي، وهنا يظهر التفوّق الكبير جدّاً لنظام Suricata على نظام Snort في قدرته على استقبال وتحليل الطّرود الشّبكيّة.



الشّكل(4) قدرة نظم الكشف على استقبال الطّرود الشّبكيّة وتحليلها وإسقاط الطّرود

ونظراً لأنّه من الممكن أن تكون الطّرود التي يتمّ إسقاطها من قِبل نظام الكشف قد تشكّل هجمات، فمن الضّروريَ تقييم نظاميّ الكشف من هذه النّاحية، وذلك لأنّ المهاجم من المحتمل أن يقوم بإغراق الشّبكة بتدفّق مرور شبكي كبير بالتّزامن مع قيامه بتنفيذ الهجوم ليتجاوز تنبيهات نظام الكشف. يُظهر لنا الشّكل (5) هجمتان نجح نظاميّ الكشف سابقاً في كشفهما، والنّسبة المئويّة لإسقاط الطّرود التي تشكّل تلك الهجمات (بمعنى آخر: التّنبيهات) وذلك وفقاً للنسبة المئويّة لتدفّق المرور الشّبكي.

يتّضح من الشّكل (5) كيف بدأ نظام Snort بإسقاط الطّرود التي تشكّل هجمات عندما أصبح حجم تدفّق المرور الشّبكي ~48%، في حين بدأ نظام الكشف Suricata بإسقاط الطّرود التي تشكّل هجمات عندما أصبح حجم تدفّق المرور الشّبكي ~96%. ومن الجدير بذكره هنا، أنّ الشّكل (4) أظهر أنّ نظام Suricata قد بدأ بإسقاط الطّرود الشّبكيّة (بشكل عام وليس شرطاً أن تشكّل تلك الطّرود هجمات) عند تدفّق مرور شبكي ~24%، في حين الشّكل (5) أظهر أنّ نظام Suricata قد أسقط الطّرود (التي تشكّل هجمات) عند تدفّق مرور شبكي ~96%.



الشّكل(5) إسقاط الطّرود الشّبكيّة التي تشكّل هجمات

بملاحظة الشّكلين (5,4)، يتضح لنا كيف تفوّق نظام Suricata على نظام Snort بشكل كبير في قدرته على استقبال عدد أكبر من الطّرود الشّبكيّة، وهذا يعود لتصميمه القائم على تعدّديّة النّياسب، ولأداءه العالي الذي يُمكّن نسخة واحدة منه من استقبال عدّة غيغا بت من حركة المرور الشّبكي [26]، في حين التّحسينات التي وعدت بها Snort،

والتي تقوم كما ذكرنا على دعم تعدديّة التّياسب في معالجة الطّرود مما يُحرّر مساحة أكبر من الذّاكرة بغرض معالجة الطّرود، لم تُشكّل منافسة قويّة مع نظام Suricata وفق النّتائج التي وصلنا لها.

ومن جانب آخر، فإنّ نظام Suricata يدعم تعدّدية التّياسب على مستوى تحليل الطّرود وكشف الهجمات [30]، وليس فقط على مستوى إقلاع النّظام ومعالجة (تحليل) الطّرود حسب التّحسينات التي وعدت بها Snort للإصدار الثالث [25]، ممّا جعل نظام Suricata يتفوّق على نظام Snort في الانخفاض الواضح في نسبة إسقاط الطّرود التي تشكّل هجمات.

4. مقترحات لتحسين نظم الكشف:

1.4. تحسين نظم الكشف من ناحية عدم استجابتها للهجمات المطبّقة:

بعد فشل نظم الكشف السّابقة في كشف بعض الهجمات التي تمّ تطبيقها، سيتمّ التطرّق لآليّة تعريف قواعد كشف جديدة لضمّها لمجموعة قواعد الكشف الخاصّة بكلّ نظام لتمكين النّظم تلك من كشف الهجمات التي فشلت سابقاً في التعرّف عليها. سيتمّ تحليل المرور الشّبكي أثناء تطبيق كلّ هجمة على حدا، واستخلاص أبرز السّمات التي تميّز كل هجمة من ناحية المرور الشّبكي، ومن ثمّ إسقاط تلك السّمات كقواعد كشف يتمّ تعريفها ضمن نظم كشف الاقتحامات السّابقة.

يُظهر الجدول (2) الهجمات التي فشلت نظم الكشف السّابقة في التعرّف عليها، وأبرز سمات تلك الهجمات من ناحية المرور الشّبكي، وأحد الأمثلة عن القواعد التي تمّ تعريفها على نظام الكشف للتعرّف لاحقاً على الهجمة، ونتيجة تطبيق القواعد الجديدة على عمل نظم الكشف. إنّ الشّكل العام لقاعدة التّنبيه في نظام الكشف يكون كالتّالي:

Alert <Protocol> <Source IP Address> <Source Port Number> <Direction: From Source to Destination> <Destination IP Address> <Destination Port Number>

كما أنّه من الممكن إضافة بعض الخيارات، مثل: sid، لإعطاء رقم معرّف للقاعدة؛ msg، لإضافة رسالة مرافقة لهذا التّنبيه؛ flags، لمطابقة ال flags الخاصّة بالبروتوكول TCP؛ content، لمطابقة حمل طرد شبكي؛ pcre، وهي اختصاراً لـ Perl Compatible Regular Expression والتي من خلالها نستطيع كتابة التّعابير النّظاميّة (Regular Expressions) بذات الطّريقة المتّبعة في لغة Perl [31]؛ ip _ proto، لفحص القيمة الموجودة في الحقل Protocol من البروتوكول IP؛ ttl، لاختبار قيمة الحقل TTL من البروتوكول IP؛ tos، لاختبار قيمة الحقل TOS من البروتوكول IP؛ dsize، لاختبار طول الطّرد.

الجدول(2) تحسين نظم الكشف من ناحية عدم استجابتها للهجمات المطبّقة

الهجوم	احد السّمات على الاقل	قواعد الكشف الجديدة (واحدة على الاقل)	النتيجة
هجمات فشل نظام Snort سابقا في كشفها			
استغلال ثغرة MS12_020 في نظام Windows	ترويسة طرد بروتوكول TCP، المنفذ 3389، ال flags الخاصّة بالبروتوكول TCP	alert tcp any any -> any 3389 (msg:"RDP MS12_020 Possible Exploit!"; sid:9911991; flags:S,A;)	نجاح
اختراق بيانات تسجيل الدّخول	محتوى الطّرد فيه كلمة USER أو PASS ويليها اسم المستخدم أو كلمة المرور تباعاً	alert tcp any any -> any 21 (msg:"Passing User Name or Password in Clear Text!! Maybe Someone Using FTP or 'Hydra' Tool Based on FTP Protocol!"; pcre:"/\b(USER|PASS)\b .*/"; sid:1000016;)	نجاح
التحكّم عن بُعد بحاسوب الضّحيّة	العديد من الطرود المتبادلة مابين جهاز الضّحيّة والمهاجم، كمثال عليها (تمّ وضع نقاط بسبب الحجم الكبير لحمولة الطّرد): ....69 73 21 54 68.....	alert tcp any any -> any any (msg:"MSFVENOM Traffic Detected_1!"; ....|"; sid:1000025;) content:"|21 54 68 69 73	نجاح
Code Injection	حمولة الطّرد: PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000	alert tcp any any -> any any (msg:"PHP Information-Disclosure - PHP Credits"; content:"PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000"; sid:1000035;)	نجاح
حجب الخدمة عن مخدّم الويب	عدد كبير من الطرود تمرّ عبر المنفذ 80، ولها أشكال flags مختلفة للبروتوكول TCP	alert tcp any any -> any 80 (msg: "Slowloris Possible Attack! Flags: ACK"; flags:A; sid:1000038;)	نجاح
XSS – الهجوم الأوّل	من خلال تحليل المرور الشبكي، نلاحظ التّالي: 0x01f0: 312e 7068 703f 6b65 7977 6f72 643d 2533 1.php?keyword=%3 0x0200: 4373 6372 6970 7425 3345 2b61 6c65 7274 Cscript%3E+alert	alert tcp any any -> any any (msg:"Cross Site Scripting (XSS) Attack - Script on Input Filed"; content:"<script>";content:"</script>"; sid:1000030;)	نجاح
هجمات فشل كلا النظامين في التعرف عليها			
الهجوم	احد السّمات على الاقل	قواعد الكشف الجديدة (واحدة على الاقل)	النتيجة
XSS – الهجوم الثّاني	من خلال تحليل المرور الشبكي، نلاحظ التّالي: 0x0050: 732f 7873 7332 2e70 6870 3f63 6c72 3d25 s/xss2.php?clr=% 0x0060: 3233 4141 4126 4368 616e 6765 3d43 6861 23AAA&Change=Cha 0x0070: 6e67 6520 4854 5450 2f31 2e31 0d0a 486f nge.HTTP/1.1..Ho	alert tcp any any -> any any (msg:"Cross Site Scripting (XSS) Attack - URL Modify"; pcre:"/clr=[%][0-9a-fA-F]{5}[&]Change=Change/s"; sid:1000031;)	نجح كلا النّظامين
PHP Object Injection	عند تحليل المرور الشبكي اثناء التعامل مع الرّابط التّالي: http://localhost/btslab/vulnerability/injection/obj.php?array=a:3:{i:0;s:6:"Delphi";i:1;s:1:"C";i:2;s:4:"Perl";} يمكن تمييز الطّرد التّالي (تمّ تحديد الجزء الهام وتعويض باقي الأجزاء بنقاط بسبب كبر الحمولة): 0x0000: 4500 01fa 896d 4000 4006 0912 c0a8 1281 E....m@.@....... .... 0x0060: 6172 7261 793d 613a 333a 7b69 3a30 3b73 array=a:3:{i:0;s 0x0070: 3a36 3a25 3232 4465 6c70 6869 2532 323b :6:%22Delphi%22;	alert tcp any any -> any any (msg:"PHP Object Injection Possible Attack!"; pcre:"/.+=[a-z]+\:[0-9]+\:\{([a-zA-Z0-9]+\:[0-9]+(\:%22[a-zA-Z0-9]+%22)?\;)+\}/"; sid:1000036;)	نجح كلا النّظامين
كشف القناة المخفيّة المتمثّلة بالحقل TTL من البروتوكول IP	عند إرسال الطرد التالي الذي يحوي قيمة حرف A في الحقل TTL 0x0000: 4504 0028 0000 4000 4106 9323 c0a8 12ad E..(..@.A..#....	alert ip any any -> any any (msg: "IP Time To Live Field Contain Unfamiliar Data... May Be Contain Covert Channel!"; ip_proto:tcp; ttl:65; sid:9988998;)	نجح كلا النّظامين
كشف القناة المخفيّة المتمثّلة بالحقل TOS من البروتوكول IP	عند إرسال الطرد التالي الذي يحوي قيمة حرف A في الحقل TOS x0000: 4541 0028 0000 4000 0 4006 93e6 c0a8 12ad ....@.@..).EA	alert ip any any -> any any (msg: "IP Type of Service Field Contain Unfamiliar Data... May Be Contain Covert Channel!"; ip_proto:tcp; tos:65; sid:9988990;)	نجح كلا النّظامين
كشف قناة التوقيت المخفيّة	بملاحظة تحليل المرور الشبكي، ومن خلال الطّرد المرسَل من البرنامج الذي تمّ كتابته باستخدام Winpcap، كان طول طرد ال IP هو 20 بايت.	بما انّ الإرسال يتمّ بمثابة طرد موحّد، فوضعنا القاعدة التّالية: alert ip any any -> any any (msg:"Short IP Packet Detected"; dsize:<21;)	نجح كلا النّظامين

❖ ملاحظات هامّة حول كشف قنوات التّخزين المخفيّة:

إنّ اختبار الحقل TTL أكثر من مرّة هو أمر ممكن في نظام Suricata، لكن نفي هذا الحقل (اختبار عدم المساواة مع قيمة معيّنة) غير ممكن، مثال:

alert ip any any -> any any (msg:"IP TTL Field May Be Contain Covert Channel!"; ttl:65; ttl:66; ttl:67; ip_proto:6; sid:2101321;)

كما أنّ اختبار الحقل TOS أكثر من مرّة هو أمر ممكن في نظام Suricata، بالإضافة لإمكانيّة استخدام النفي، مثال:

alert ip any any -> any any (msg:"IP TOS Field May Be Contain Covert Channel!"; tos:!0; tos:!1; tos:!2; tos:!4; tos:!8; ip_proto:6; sid:2101322;)

في حين يمكن استخدام النّفي أثناء اختبار الحقلين TOS, TTL في نظام Snort، لكن في حالتنا لا تتحقّق الإفادة إلّا في حال كان المرور الشّبكي قد تمّ فيه ضبط قِيم تلك الحقول بشكل تام على قِيم معيّنة ثابتة (مثال: TTL=64, TOS=0).

alert ip any any -> any any (msg:"IP TTL Field May Be Contain Covert Channel!"; ttl:!64; ip_proto:6; sid:2101321;)
alert ip any any -> any any (msg:"IP TOS Field May Be Contain Covert Channel!"; tos:!0; ip_proto:6; sid:2101322;)

إلّا أنّه من غير الممكن في نظام Snort اختبار أكثر من قيمة واحدة للحقل TTL أو الحقل TOS في كلّ قاعدة مفردة.

يتضح لدينا هنا أنّ نظام Suricata كان أكثر شموليّة لحدّ ما من نظام Snort في تعريف قواعد لكشف قنوات التّخزين المخفيّة المُقترحة، بحيث يمكن في نظام Suricata تعريف قاعدة واحدة لضبط عدّة قِيم لحقليّ TTL, TOS، بينما في نظام Snort نحتاج لعدد من القواعد يتناسب مع عدد القِيم المحتملة التي من الممكن أن تكون موجودة في الحقلين السّابقين والتي تمثّل تسريباً للمعطيات.

باختصار، فإنّ تحسين نظم الكشف من ناحية عدم استجابتها للهجمات المطبّقة قد أحرز تقدّماً في كلا نظاميّ الكشف، كما أنّ تعريف قواعد الكشف الجديدة في كلا النّظامين كان متشابهاً وسهلاً في كثير من الأحيان، إلّا أنّ الصّعوبة كانت في فهم كيفيّة صياغة التّعابير النظاميّة في بعض القواعد.

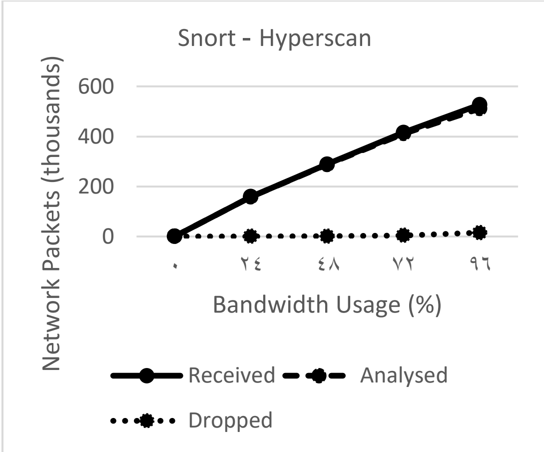
2.4. تحسين نظام الكشف Snort من ناحية قدرته على استقبال الطّرود الشّبكيّة وتحليلها، وتقليص عدد الطّرود التي يتمّ إسقاطها:

حسب ما تمّ إيضاحه في السّابق كيف أنّ نظام Suricata قد تفوّق على نظام Snort في قدرته على تحليل الطّرود الشّبكيّة التي يستقبلها، وكذلك في انخفاض عدد الطّرود الشّبكيّة التي يتمّ إسقاطها، فكان الاتّجاه نحو مناقشة تحسين نظام Snort من هذه النّاحية، ليتمّ استخدام تقنيّة ال Hyperscan، والتي هي مكتبة مفتوحة المصدر، عالية الأداء، تقوم بمطابقة التّعابير النّظاميّة، وكذلك قواعد PCRE، وعلى الرّغم من أنّها تعمل على إبطاء عمليّة الإقلاع (وحسب تجربتنا ليست أبطأ من إقلاع نظام Suricata)، إلّا أنّها تعمل على زيادة سرعة مطابقة القواعد [32].

يُظهر الشّكل (6) تقييماً لنظام الكشف Snort بعد تطبيق خاصيّة ال Hyperscan لناحية قدرته على تحليل الطّرود الشّبكيّة التي يقوم باستقبالها، وللطرود الشّبكيّة التي يقوم بإسقاطها، ويتّضح لنا كيف أنّ عدد الطّرود الشّبكيّة التي يتم تحليلها متجانسة لحدّ كبير جدّاً مع عدد الطّرود التي يتمّ استقبالها، كما أنّ نسبة الطّرود الشّبكيّة التي يتم إسقاطها كان قليل جدّاً ولاتجاوز نسبة 3%.

بالمقارنة مع تقييم نظام Snort بدون خاصيّة ال Hyperscan، نلاحظ أنّ تقنيّة ال Hyperscan قد حسّنت

بشكل واضح قدرة النّظام على استقبال الطّرود الشّبكيّة ومعالجتها وتخفيض نسبة الطّرود التي يتم إسقاطها. يُظهر الجدول (3) ملخّصاً لأبرز أوجه المقارنة ما بين الحالتين.



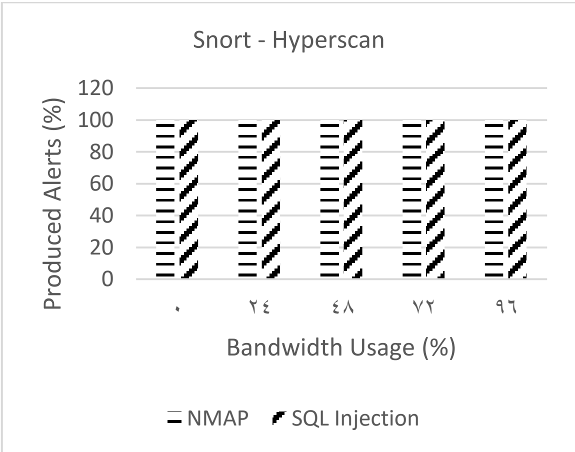
الشّكل(6) تقييم نظام الكشف Snort مع خاصيّة Hyperscan من ناحية استقبال الطّرود الشّبكيّة وتحليلها وعدد الطّرود التي يتم أسقاطها

الجدول(3) مقارنة نظام Snort في حال استخدام خاصيّة Hyperscan وفي حال عدم استخدامها

	بدون Hyperscan	مع Hyperscan
نسبة الطّرود الشّبكيّة التي يتمّ معالجتها	تنحدر من 100% إلى 42%~	من 100% إلى 97%~
نسبة الطّرود الشّبكيّة التي يتمّ إسقاطها	من 0% إلى 58%~	من 0% إلى 3%~
استقبال الطّرود الشّبكيّة	تضعف قدرته بشكل ملحوظ بعد بلوغ تدفّق المرور الشّبكي نسبة 72%	يستمرّ في استقبال الطّرود الشّبكيّة حتّى عند بلوغ تدفق المرور الشّبكي نسبة 96%

وبما أنّه قد تمّ لمس تحسّن ملحوظ في نسبة الطّرود الشّبكيّة التي يتم تحليلها، وانخفاض واضح في نسبة الطّرود الشّبكيّة التي يتم إسقاطها، فكان من الجدير بتقييمه هو نسبة إسقاط الطّرود الشّبكيّة التي تشكّل هجمات عند استخدام هذه الخاصيّة. وبعد التّجريب، تمّ الوصول لنتائج عمليّة خلصنا من خلالها إلى نتيجة 0% كنسبة معبّرة عن إسقاط الطّرود الشّبكيّة التي تشكّل هجمات في كلا الهجمتين التي تمّ اختبارهما. وبمعنى آخر، فإن النّسبة المئويّة المعبّرة عن عدد الطّرود الشّبكيّة التي تشكّل هجمات والتي تمّ التنبيه عنها هي 100% لكلا الهجمتين السابقتين. يظهر الشّكل (7) هجمتان نجح نظام Snort في كشفهما سابقاً، والنّسبة المئويّة للتنبيهات عن الطّرود التي تشكّل هجمات وذلك وفقاً للنسبة المئويّة لتدفّق المرور الشّبكي.

هنا يتضح لنا التحسّن الكبير مقارنة بحالة عدم استخدام خاصيّة Hyperscan، فقبل استخدام هذه الخاصيّة وكما استنتجنا من الشّكل (5) يتضح أنّ نسبة إسقاط الطّرود المعبّرة عن الهجمات تتراوح ما بين (15 – 45)%، أمّا عند استخدام هذه الخاصّية وبالاستنتاج من الشّكل (7)، فقد انحدرت نسبة إسقاط الطّرود المعبّرة عن الهجمات إلى 0%.

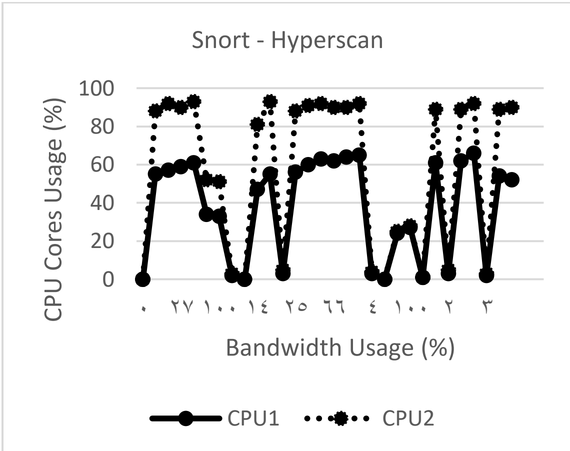


الشّكل(7) نسبة التّنبيه عن الطّرود التي تشكّل هجمات في نظام Snort مع الخاصيّة Hyperscan

في متابعة لما سبق، ورغم التّحسينات السّابقة التي قدّمتها خاصيّة Hyperscan، إلّا أنّ تلك الخاصيّة لم تتفوّق على نظام Suricata في قدرته على استقبال الطّرود الشّبكيّة، حيث أنّ عدد الطّرود الشّبكيّة التي استقبلها نظام Snort مع خاصيّة Hyperscan كان يتراوح بين (500 – 600) ألف

طرد شبكي، بينما عدد الطّرود الشّبكيّة التي استقبلها نظام Suricata كما ذكرنا سابقا كانت تتراوح بين (2 – 2.5) مليون طرد شبكي، إلّا أنّ التفوّق على نظام Suricata كان من حيث تقليص عدد الطّرود التي تمّ إسقاطها (بما فيها تلك الطّرود التي تشكّل هجمات). ومن جهة أُخرى، وعند تقييم نظام Snort مع خاصيّة Hyperscan من ناحية استنزاف نُوى المعالجة وتوزيع الحمل عليها، كانت النّتائج تُشير كما في الشّكل (8) إلى أنّ التّجانس في توزيع الحمل كان أقلّ ممّا هو عليه الحال قبل استخدام الخاصيّة Hyperscan، كما أنّ استنزاف نواتيّ المعالجة كان أقلّ، وعبء المعالجة الأكبر كان من نصيب النّواة الثّانية.

يمكن القول باختصار، أنّ خاصيّة Hyperscan قد حسّنت قدرة نظام الكشف Snort على استقبال الطّرود الشّبكيّة، وتحليلها، وتقليص نسبة الطّرود الشّبكيّة التي يتمّ إسقاطها بما فيها الطّرود المعبّرة عن الهجمات، وذلك لأنّها تقوم على تسريع مطابقة حركة المرور الشّبكي مع قواعد الكشف من (2 – 3) مرّات [32]. إلّا أنّها لم تقدّم وعود من ناحية تحسين استنزاف نُوى المعالجة وتوزيع الحمل عليها، وهذا ماتمّ تأكيده من خلال تقييماتنا السّابقة.



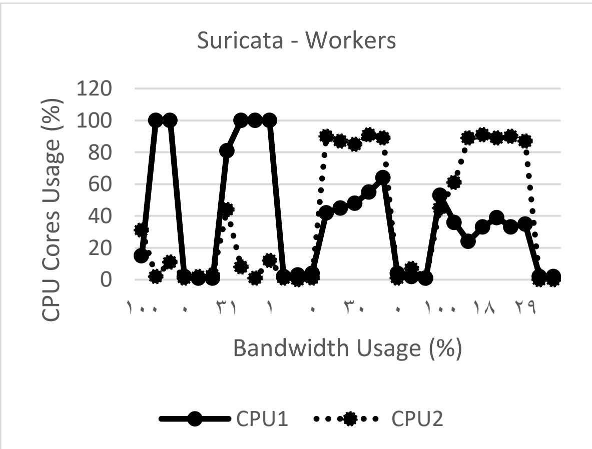
الشّكل(8) تقييم نظام الكشف Snort مع الخاصيّة Hyperscan من ناحية استنزاف نُوى المعالجة وتوزيع الحمل عليها

3.4. تحسين نظام الكشف Suricata من ناحية توزيع الحمل على نُوى المعالجة:

كما تمّ توضيحه سابقاً، فقد كان نظام Suricata يفتقر لتحقيق التّجانس المثالي في توزيع الحمل على نُوى المعالجة، لذلك برزت الحاجة لدينا لمناقشة تحسين هذا النّظام من هذه النّاحية، وبما أنّ نظام Suricata هو نظام متعدّد النّياسب، وبما أنّ تشكيل توزيع النّياسب في التقاط الطّرود الشّبكيّة له أثر ينعكس على أداء نظام الكشف، فقد وفّر نظام Suricata عدّة أوضاع تشغيل تختلف في طريقة تشكيلها للنياسب [33].

1.3.4. تقييم نظام Suricata عند استخدام وضع التّشغيل Workers:

في هذا الوضع تقوم بطاقة الشّبكة (NIC: Network Interface Card) بالتّأكّد من أنّ الطّرود الشّبكيّة يتمّ موازنتها بشكل ملائم على نياسب المعالجة، وهو الوضع الذي يُوصف بأنه يقوم بالأداء الأفضل حسب [33].



الشّكل(9) تقييم نظام الكشف Suricata مع وضع تشغيل Workers من ناحية استنزاف نُوى المعالجة وتوزيع الحمل عليها

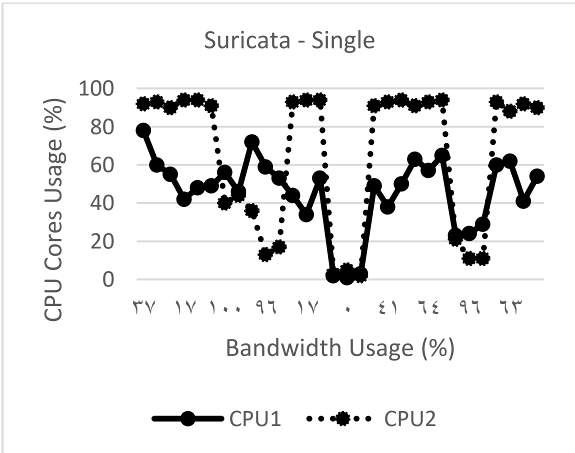
بالتقييم تمّ استخلاص الشّكل (9)، بحيث نلاحظ كيف أنّ توزيع الحمل على نواتيّ المعالجة لم يكن أمثليّاً أيضاً، حتّى أنّه كان أقلّ تجانساً من وضع التّشغيل العاديّ، إلّا أنّ عبء المعالجة كان من نصيب النّواة الأولى تارةً، وتارةً أخرى من

نصيب النّواة الثانية، في حين كانت النّواة الأولى يقع عليها عبء المعالجة الأكبر في وضع التشغيل العادي.

2.3.4. تقييم نظام Suricata عند استخدام وضع التشغيل Single:

يُشبه الوضع Workers، لكن مع فرق أنّ عدد النّياسب هو 1 [33]، وبالتّقييم تمّ استخلاص الشّكل (10)، بحيث يظهر لنا فقر التّجانس أيضاً وبشكل أسوء حتّى من الوضع Workers، كما أنّ عبء المعالجة الأكبر كان من نصيب النّواة الثانية.

إنّ تحسين نظام Suricata من ناحية توزيع الحمل على نُوى المعالجة لم يترك أثراً ملموساً في ظروف بحثنا، ويمكن تبرير ذلك بأنّ الآلات الافتراضيّة التي نقوم بالتّجريب عليها لها مواصفات تشغيليّة متواضعة، في حين من المؤكّد أنّه سيتمّ لمس التحسّن عند إتاحة عدد أكبر من نّوى المعالجة والنّياسب بسبب طبيعة تصميم النّظام القائمة على تعدديّة النّياسب، وهذا مايتّفق مع العديد من الدّراسات السّابقة [16,21] والتي تؤكّد تحسّن أداء نظام Suricata عند إتاحة الموارد بشكل أكبر.



الشّكل(10) تقييم نظام الكشف Suricata مع وضع تشغيل Single من ناحية استنزاف نُوى المعالجة وتوزيع الحمل عليها

5. خُلاصة البحث:

من خلال مناقشة نتائج التّقييمات التي وصلنا لها، نجد أنّ نظام Suricata قد تفوّق على نظام Snort في استجابته لبعض الهجمات المُطبّقة، وذلك لاحتواءه على عدد أكبر من التّواقيع، وكذلك تفوّق من حيث قدرته على استقبال ومعالجة عدد أكبر من الطّرود الشّبكية، وفي انخفاض نسبة عدد الطّرود والتّنبيهات التي يقوم بإسقاطها، حيث أنّ التّحسينات التي وعد بها نظام Snort من ناحية دعمه لتعدديّة النّياسب لتحرير مساحة أكبر من الذّاكرة لمعالجة الطّرود، لم تكن على مستوى المنافسة مع نظام Suricata. وحتّى عند تفوّق نظام Snort في توزيع الحمل على نُوى المعالجة، فقد كان هنالك ضريبة على الأداء جعلت استنزاف نُوى المعالجة في كثير من الأحيان 100%~، كما أنّ تفوّق نظام Snort في قلّة استنزافه للذّاكرة لم يُشكّل تلك العلامة الفارقة والتي تميّزه بشكل كبير عن منافسه Suricata.

إنّ تحسين نظاميّ الكشف من ناحية الاستجابة للهجمات المُطبّقة قد أثمر في كلا النّظامين. كما أنّ تحسين قدرة نظام Snort على تقليص نسبة الطّرود والتّنبيهات التي يتمّ إسقاطها جعلته يتفوّق على نظام Suricata، إلّا أنّه ورغم التحسّن في زيادة عدد الطّرود الشّبكيّة التي يستقبلها ويعالجها، فلم يستطع مُجاراة نظام Suricata. في حين مقترحات تحسين توزيع الحمل على نُوى المعالجة في نظام Suricata لم تترك نتائج مُرضية في ظروف بحثنا، وذلك نظراً لإمكانية موارد الآلات الافتراضيّة التي تمّ التّجريب عليها.

يمكن القول أنّه في حال كانت الآلة التي يتمّ تنصيب نظام الكشف عليها هي آلة محدودة الموارد، بالإضافة لأنّ الشّبكة ليست ذات سرعة عالية وتدفّق كبير، كما أنّ أشكال القنوات المخفيّة محدود، فإنّ نظام الكشف Snort هو المفضّل في هذه الحالة. أمّا في حال كان لدينا آلة ذات موارد أعلى، وتقبع على شبكة ذات تدفّق كبير، ولها احتمالات أكبر من أشكال القنوات المخفيّة، فإنّ نظام Suricata يكون هو

الخيار الأفضل. ويبقى تحليل المرور الشّبكي باستخدام أحد الأدوات التي تمّ ذكرها أو غيرها، هو جوهر الكشف عن القنوات المخفيّة والهجمات النّاشئة، بحيث كانت أدوات التّحليل تلك هي الذّراع الأيمن لنظم الكشف، فمن خلالها تمّ التعرّف على أشكال القنوات المخفيّة وعلى الهجمات التي لم تستطع نظم الكشف التعرّف عليها.

يترك هذا البحث الباب مفتوحاً أمام تقييم نظم الكشف الأخرى، كما أنّه من المفيد جدّاً تقييم نظم الكشف من ناحية معايير أخرى لم يتناولها هذا البحث، بالإضافة لتطوير القواعد التي تمّ تعريفها لكشف الهجمات التي لم تتعرّف عليها نظم الكشف من جهة، ومن جهة أخرى تطوير مجموعة جديدة من القواعد التي تغطّي عدد أكبر من أشكال القنوات المخفيّة – رغم قناعتنا بأنّ نظم الكشف السّابقة لا تُعتبر الحلّ الأمثل لمكافحة القنوات المخفيّة، بسبب التعدّد الكبير لأشكال تلك القنوات، ولاعتماد نظم الكشف السّابقة على مطابقة القواعد (التّواقيع) في عملها.

التمويل: هذا البحث ممول من جامعة دمشق وفق رقم التمويل(501100020595).

References:

[1] Callado, A., Kamienski, C., Szabo, G., Gero, B. P., Kelner, J., Fernandes, S., & Sadok, D. (2009). A Survey on Internet Traffic Identification. IEEE Communications Surveys & Tutorials, 11(3), 37–52. https://doi.org/10.1109/surv.2009.090304

[2] Sikos, L. F. (2020). Packet analysis for network forensics: A comprehensive survey. Forensic Science International: Digital Investigation, 32, 200892. https://doi.org/10.1016/j.fsidi.2019.200892

[3] Qadeer, M. A., Iqbal, A., Zahid, M., & Siddiqui, M. R. (2010). Network Traffic Analysis and Intrusion Detection Using Packet Sniffer. IEEE. Xplore. https://doi.org/10.1109/ICCSN.2010.104

[4] Munz, G., & Carle, G. (2007). Real-time Analysis of Flow Data for Network Attack Detection. 2007 10th IFIP/IEEE International Symposium on Integrated Network Management. doi:10.1109/inm.2007.374774.

[5] P, W. (2020). A Survey of Intrusion Detection System. International Journal of Informatics and Computation, 1(1), 1–10. https://doi.org/10.35842/ijicom.v1i1.7

[6] Lazarevic, L., Vipin, K., Jaideep, S. (2005). Intrusion Detection: A Survey. In Kumar, V. et al. (eds.) Managing Cyper Threats Issues, Approaches, and Challenges vol. 5, pp.19-78.

[7] Smeets, M., & Koot, M. (2006), Research Report: Covert Channels (University of Amsterdam, Amsterdam).

[8] Lampson, B. (1973). A note to the confinement problem. Communications of the ACM, 16(10), 613–615

[9] Vishnoi, V. (2018), AN OFFLINE AND EFFICIENT STORAGE COVERT CHANNEL DETECTION MECHANISM. 10.1729/Journal.22251.

[10] US DoD: Trusted Computer System Evaluation Criteria, 1985, http://csrc.nist.gov/publications/history/dod85.pdf

[11] Zander, S., Armitage, G., & Branch, P. (2007). A survey of covert channels and countermeasures in computer network protocols. IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2007.4317620.

[12] Selvi, J. (2012), Covert Channel Over Social Network, SANS Institute, https://www.sans.org/reading-room/whitepapers/threats/covert-channels-social-networks-33960

[13] Okhravi, H., Bak, S., & King, S.T. (2010). Design, implementation and evaluation of covert channel attacks. 2010 IEEE International Conference on Technologies for Homeland Security (HST), 481-487. 10.1109/THS.2010.5654967.

[14] Sabahi, F. & Movaghar, Ali. (2008). Intrusion Detection: A Survey. Systems and Networks Communication, International Conference on. 23-26. 10.1109/ICSNC.2008.44.

[15] Tayyebi, Y., & Bhilare, D.S. (2018). A Comparative Study of Open Source Network Based Intrusion Detection Systems. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 9 (2), 2018, 23-26.

[16] Day, D.J., & Burns, B.M. (2011). A performance analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines. ICDS 2011.

[17] Gunadi, H., Zander, S. (2017). Comparison of IDS Suitability for Covert Channels Detection. Murdoch University IT NSRG Technical Report 20170818A.

[18] HAMMOUDA, S., MAALEJ, L., & TRABELSI, Z. (2008). Towards Optimized TCP/IP Covert Channels Detection, IDS and Firewall Integration. 1-5. 10.1109/NTMS.2008.ECP.101.

[19] Fekolkin, R. (2015). Intrusion Detection and Prevention Systems: Overview of Snort and Suricata. Internet Security, A7011N, Luleå University of Technology, jan 06, 2015 pp 1-4.

[20] Hänninen, M. (2019). Open source intrusion detection systems evaluation for small and medium-sized enterprise environments.

[21] White, J.S., Fitzsimmons, T.D., & Matthews, J.N. (2013). Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata. Proceedings of SPIE, The International Society for Optical Engineering. 8757. 10.1117/12.2015616.

[22] Albin, E. (2011). A Comparative Analysis of the Snort and Suricata Intrusion-Detection Systems. Master's Thesis, Naval Postgraduate School, Monterey CA.

[23] Díaz-Verdejo, J., Muñoz-Calle, J., Estepa Alonso, A., Estepa Alonso, R., & Madinabeitia, G. (2022). On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks. Applied Sciences, 12(2), 852. https://doi.org/10.3390/app12020852.

[24] Bada, G. K., Nabare, W. K., & Quansah, D. K. K. (2020). Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro Intrusion Detection Systems in Perspective. International Journal of Computer Applications, 176(40), 39–44. https://doi.org/10.5120/ijca202092051 3

[25] Snort - Network Intrusion Detection & Prevention System | www.snort.org.

[26] Home – Suricata | www.suricata.io.

[27] TCPDUMP public repository | www.tcpdump.org .

[28] Goyal, P., & Goyal, A. (2017). Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark. 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN), 77-81.

[29] Wireshark | Go Deep, www.wireshark.org.

[30] Ghafir, Ibrahim & Prenosil, Vaclav & Svoboda, Jakub & Hammoudeh, Mohammad. (2016). A Survey on Network Security Monitoring Systems. 77-82. 10.1109/W-FiCloud.2016.30.

[31] PCRE - Perl Compatible Regular Expressions, https://www.pcre.org/

[32] Snort Blog: Improve Snort 3 performance with Hyperscan, blog.snort.org/2020/09/snort-3-hyperscan-.html Runmodes —Suricata 6.0.0. documentation.

[33] suricata.readthedocs.io/en/suricata-6.0.0/performance/runmodes.htm| l