الجمهورية العربية السورية

المعهد العالي للعلوم التطبيقية والتكنولوجيا

قسم الاتصالات-ماجستير شبكات

نظام إدارة الثقة في الشبكات المعّرفة برمجياً

AN APPLICATION TRUST MANAGEMENT SYSTEM IN SOFTWARE DEFINED NETWORKS

قُدِّمت هذه الدراسة لنيل درجة الماجستير في هندسة الاتصالات

تقديم

م.محمد شعبان

إشراف

د.خلدون خرزم د.محمد الشّايطة

تشرين الأول 2019

الإهداء

الى ياسمين دمشق وغار حلب والى كل حبة تراب فيك..

الى الشمس المنيرة, الظلال الوفيرة, الأمن و الآمان و الاطمئنان, المتربعة بعلياء في قلبي و عقلي في كل زمان و مكان, أبجدية اللسان و الإنسان, المترامية من النهرين الى البحر و من اسكندرون الى قلبها الجولان

بلدي الحبيبة سوريه

الى شجر الزيتون الصامد منذ الأزل، ورائحة الليمون العابقه في الجولان، شقيقة دمشق و وصية السماء وأرض الأنبياء الى الغائبة الحاضرة.....

فلسطين حاضرة الشام

الى من دعمتني في مسيرتي التعليمية.

زوجتي حبيبتي

الى من بوجودها أصبح للحياة معنى ... الى طفلتي .....

اليكِ وحدكِ

كلمة شكر

الحَمدُ لله حتَّى يَبلُغَ الحَمدُ مُنْتَهَاه

أتقدم بالشكر إلى كل من ساهم في هذا البحثّ وأخص بالذكر:

الاستاذين المشرفين: **الدكتور محمد الشايطة والدكتور خلدون خرزم** لما قدماه من توجيهات، وإرشادات ساهمت في إغناء الأطروحة، وإشرافهما المستمر، وعملهما الدؤوب من أجل رفع سوية الأبحاث العلمية، لترقى الى مستوى الأبحاث العلمية المتقدمة.

كما أتقدم بالشكر لكل من ساهم بإنجاح التحصيل العلمي في المعهد العالي للبحوث العلمية.

أشكر إدارة المعهد، وعلى رأسها **الدكتور ماهر سليمان** لجهودهم الكريمة في دعم مسيرة الأبحاث العلمية، من خلال سعيهم الدؤوب، والتشجيع المستمر، وتقديم المناخ المناسب للطلاب، من أجل تقديم أبحاث متطورة تضاهي الأبحاث في أرقى الجامعات العلمية.

الخلاصة

إنّ الشبكات المعرّفة برمجياً SDN هي عبارة عن بنية شبكية جديدة توفر التحكم المركزي بكامل الشبكة. يعمل هذا المتحكم كنظام تشغيل يقوم بإرسال التعليمات وتطبيق التغييرات من خلال الواجهات التخاطبية بينه وبين الأجهزة المسؤول عنها، ويُدعى بالمتحكم. بالرغم من أنّ التحكم المركزي هو عبارة عن سمة مميزة جداً في الشبكات المعرّفة برمجياً، إلّا أنه يواجه تحديات خطيرة عديدة، وعلى جميع المستويات. يوجد اليوم العديد من الدراسات التي تهتم بمسائل حماية الشبكات المعرّفة برمجياً، ولعلّ أهم الدراسات تلك التي تعمل على تخفيف نقاط الضعف الخاصة بإمكانية حقن تطبيقات خبيثة ضمن الشبكة والقيام بمختلف الهجمات على تجهيزات الشبكة. تتنوع هذه التطبيقات وتختلف فيما بينها بطريقة تنفيذ الهجوم وبالضرر الذي تسبّبه في الشبكة [39]. لعلّ من أهم مشاكل طبقة التطبيقات في هذه الشبكات هي مسألة نقطة السقوط المنفردة single point of failure والتي تحدث في حال أصبح غير متاحاً نتيجة هجوم حجب الخدمة الموزع DDoS (Distributed Denial of Service Attack)، حيث من خلال زرع تطبيق خبيث ضمن الشبكة، يستطيع المهاجم القيام باستهداف مضيف محدّد أو شبكة فرعية في الشبكة أو استهداف المتحكم بحد ذاته، مما يؤدي إلى إحداث ضرر كبير في الشبكة، لذلك من هنا كانت نقطة انطلاقتنا في بحثنا، فقمنا باستخدام التحكم المركزي في الشبكات المعرّفة برمجياً لكشف الهجمات، واقتراح حل فعّال بحيث لا يشكل عبئ على موارد الشبكة لمواجهة مثل هكذا هجمات.

يعرض هذا البحث أهم نقاط ضعف طبقة التطبيقات، كما يعرض كيفية استطاعة هجمات DDoS استنزاف موارد المتحكم وتوفير حل لكشف مثل هكذا هجمات بالاعتماد على تغيرات الأنتروبية Entropy للعنوان المنطقي الخاص بالعقدة الهدف. يمكن لهذه الطريقة كشف هجمات DDoS بالاعتماد على أول بضع مئات من الطرود واردة من الطرف المهاجم.

فهرس المحتويات

فهرس الأشكال

فهرس الجداول

🕮 الاختصارات المستخدمة:

SDN	:	**S**oftware **D**efined **N**etwork
DDoS	:	**D**istributed **D**enial of **S**ervice Attack.
VLAN	:	**V**irtual **L**ocal **A**rea **N**etwork.
API	:	**A**pplication **P**rogramming **I**nterface.
ONF	:	**O**pen **N**etworking **F**oundation.
MAC	:	**M**edia **A**ccess **C**ontrol.
TLS	:	**T**ransport **L**ayer **S**ecurity.
TCAM	:	**T**ernary **C**ontent **A**ddressable **M**emory.
IDS	:	**I**ntrusion **D**etection **S**ystem
PKI	:	**P**ublic **K**ey **I**nfrastructure.
TCB	:	Trusted Computing Base.
VAVE	:	Virtual source Address Validation Edge.
DNS	:	**D**omain **N**ame **S**erver.
ICMP	:	**I**nternet **C**ontrol **M**anagement **P**rotocol
PDF	:	**P**robability **D**ensity **F**unction
CDF	:	**C**umulative **D**ensity **F**unction
NAT	:	Network Address Translation

❖ لا يضيع شيء ذو قيمة إذا صرفنا الوقت الكافي في إتقانه (آبراهام لينكولن).

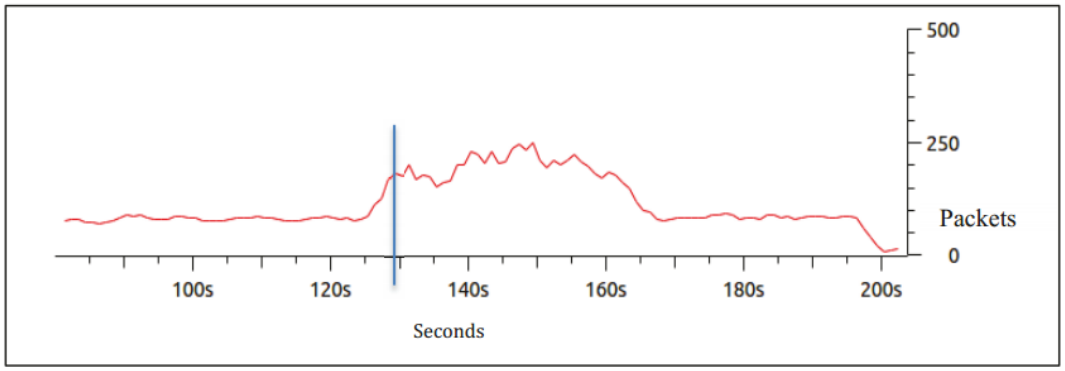
1.1 مقدمة:

تعدُّ فكرة الشبكات المعرّفة برمجياً ثورةً في مجال إدارة الشبكة. في هذه الشبكات، لا يقوم المبدّل switch بمعالجة الطرود الواردة إليه، إنّما يقوم بالبحث عن تطابق بين الطرد الوارد وإحدى المدخلات الموجودة ضمن جدول التسيير، فعندما لا يجد تطابقاً، يقوم بإرسالها إلى المتحكم controller. يعدُّ المتحكم بمثابة نظام التشغيل للشبكات المعرّفة برمجياً، حيث يقوم بمعالجة المعطيات وإصدار الإجراء اللازم اتخاذه من قبل المبدّل بشأن هذا الطرد، فإمّا التسيير forwarding أو التجاهل drop. بهذه الآلية، تكون الشبكات المعرّفة برمجياً قد فصَلت طبقة المعطيات عن طبقة التحكم.

يمكن أن تتكون الشبكات المعرّفة برمجياً من عدة متحكمات، وكلٌّ منها متصل مع شبكة من المبدّلات. يمكن النظر إلى كلِّ متحكم مع مبدّلاته على أنّهم شريحة من الشبكة. في حال فقدان الاتصال بين المبدّلات والمتحكم، سوف تفقد الشبكة القدرة على المعالجة.

إنّ أحد أهم الأسباب التي قد تؤدي إلى جعل المتحكم غير متاحاً هي زرع تطبيقات خبيثة ضمن الشبكة تقوم بشنِّ هجمات منع الخدمة الموزعة DDoS (Distributed Denial of Service Attack) عليه. يتم في مثل هذه الهجمات إرسال أعداد ضخمة من الطرود إلى مضيف host محدّد أو إلى مجموعة من المضيفين أو الأجهزة ضمن الشبكة. في حال تم انتحال العنوان المنطقي الخاص بالطرد الوارد "IP spoofing"، فإنّ المبدّل لن يستطيع إيجاد مطابقة مع جدول التسيير الخاص به، فيقوم بإرسال الطرد إلى المتحكم. يمكن أن يؤدي ورود مجموعة من الطرود الشرعية والطرود غير الشرعية (المنتحلة العناوين المنطقية والمستخدمة في هجوم DDoS) إلى تقييد موارد المتحكم واستنزافها نتيجة المعالجة المستمرة، ممّا يجعل المتحكم غير متاحاً وبالتالي فقدان بنية الشبكات المعرّفة برمجياً، وفي حال وجود متحكم احتياطي، فإنّه سيعاني من المشكلة ذاتها.[40]

إنّ الهدف الرئيسي من هذا البحث هو دراسة أثر وجود التطبيقات الخبيثة في الشبكة، وكشف هجمات منع الخدمة الموزعة DDoS التي قد تقوم بها في مراحل مبكّرة، مصطلح مبكّراً يعتمد على الشبكة ذاتها ويعتمد على قدرة الأجهزة على التحمّل وعلى خصائص حركة مرور البيانات traffic. على أية حال، في حال تمّ كشف الهجوم في الطرود القليلة الأولى (من رتبة بضع المئات)، يمكن حينها تخفيف وطأة الهجوم وإنقاذ المتحكم من أن يصبح ضحية لأعداد ضخمة من الطرود الخبيثة. يعرض الشكل(1-1) مثالاً بسيطاً عن هجمات DDoS على متحكم، حيث أنّ الوضع الطبيعي لطرود المعطيات يبلغ حوالي 100 طرد في الثانية، لكن عند بدء الهجوم، نلاحظ أنَّ المنحني يرتفع بشكل حادّ، فيبلغ تقريباً 250 طرد في الثانية. إنّ عملية المحاكاة هذه لهجمات DDoS تمّ توجيهها على متحكم SDN متصل مع 46 مضيف و 9 مبدّلات. استغرقت عملية الهجوم 40 ثانية وتمَّ إرسال 500 طرد بعناوين منطقية وهمية، وجميعها تستهدف مضيفاً محدّداً.



الشكل(1-1). مثال بسيط عن هجوم DDoS على متحكم.

إنَّ الهدف من هذا البحث، هو اكتشاف وجود تطبيق يقوم بهكذا هجمات ومن ثمّ كشف الهجوم عند بدايته، بالتالي إن نظرنا إلى الشكل(1-1) السابق، نجد أنَّ الخط الأزرق يشير إلى المكان الأمثل لكشف هذا الهجوم. في حال تمّ اكتشاف الهجوم عند هذه النقطة، فإنّ آليات التخفيف والوقاية من هذا الهجوم سوف يكون لديها الوقت الكافي لتأمين المتحكم قبل أن يتم إعمائه بهذا الهجوم ويصبح غير متاحاً. لتحقيق ذلك، ينبغي إيجاد

طريقة سريعة وفعّالة لإنقاذ الشبكة، وفي الوقت ذاته، ينبغي أن تكون هذه الطريقة خفيفة على الشبكة ولا تسبّب استنزاف قدرات المتحكم، وخاصة عند ذروة الهجوم.

إنّ جمع الإحصائيات هي إحدى المهام التي يقوم بها المتحكم. في دراستنا، سوف نستخدم هذه الواصفة attribute لإضافة مجموعة أخرى من الإحصائيات إلى المتحكم، وهي العناوين المنطقية للعقد الهدف. سوف نستخدم العنوان المنطقي للطرود،حيث أنّ الشبكات المعرّفة برمجياً تسمح بجمع معلومات الحقول الخاصة كافّة.

في الحل المطروح، يتم قياس عشوائية الطرود الواردة، حيث تعدّ الأنتروبية مقياساً جيّداً للعشوائية [11]. تقيس الأنتروبية احتمال وقوع حدث مع الأخذ بعين الاعتبار للعدد الكلّي للأحداث. مثال على ذلك، في شبكة مؤلفة من 64 مضيف، فإنّ جميع المضيفين ينبغي أن يكون لديهم احتمالات متقاربة لاستقبال طرود جديدة واردة، وهذا سيؤدي بشكل منطقي إلى وجود أنتروبية عالية. إنّ ورود طرد جديد، لا يوجد له تطابق مع جدول التسيير في المبدّل، يعني أنّه سيتم توجيهه إلى المتحكم. في حال بدأ مضيف أو عدة مضيفين باستقبال سلسلة متتالية من الطرود الواردة، سوف تتناقص العشوائية وبالتالي ستنخفض الأنتروبية. سنستخدم هذه الخاصية في بحثنا لكشف الهجوم في المراحل المبكّرة. بالاعتماد على الاختبارات التي أجريناها أثناء البحث، قمنا باختيار عتبة threshold معينة، وسيتم اعتبار القيم المنخفضة عنها على أنّها هجمات. تعدّ ميزة قابلية البرمجة في هذه الشبكات إحدى أهم السمات، ففي أية وقت تتغير فيه إعدادات الشبكة، يمكن ضبط قيمة العتبة، ويمكن إعادة ضبطها أثناء النقل الحي للمعطيات داخل الشبكة. بالاعتماد على الشبكة، يمكن أن يتم حساب الأنتروبية للعناوين المنطقية للعقدة الهدف، علامة الشبكات الافتراضية المحلّية VLAN tag، البوابة الهدف أو لأية حقل آخر. في حال كانت العشوائية أقل من العتبة، فيتم الإقرار بأنّ هجوماً يُشنُّ حالياً.

2.1 الهدف من البحث:

في بحثنا هذا، سنقوم بدراسة الشبكات المعرّفة برمجياً لإيجاد نقاط الضعف العامة لهذه الشبكات، وسنقوم بدراسة مفصّلة لنقاط الضعف الممكن استخدامها في طبقة التطبيقات لشنّ هجوم منع الخدمة DDoS. سنقوم بدراسة طرق مختلفة لكشف الهجمات DDoS لدى المتحكم. على أية حال، فإنّ بنية شبكات SDN تفرض قيوداً على نوع الحل والطريقة التي سوف يتم تنجيزه فيها، وهذه القيود هي:

- محدودية الموارد الخاصة بالمتحكم.

- الحاجة إلى كشف الهجوم قبل أن يصبح المتحكم خارج الخدمة بسبب العدد الضخم للطرود الخبيثة.

3.1 مساهمة البحث:

من خلال هذا البحث، نستطيع إيجاد نقطة الضعف في الشبكات المعرّفة برمجياً وكشف التطبيقات التي تحاول القيام بهجوم منع الخدمة الموزع DDoS. يساهم هذا البحث في إظهار كيفية أن هجوم DDoS يقّيد موارد المتحكم من خلال جعله يقوم بعمليات معالجة مستمرة لطرود خبيثة، وأيضاً طرح آلية فعّالة لكشف الهجوم DDoS. يمكننا تفنيد هذه المساهمة كمايلي:

أ. إظهار كيفية قيام الهجوم DDoS بإعماء المتحكم بشكل كامل في بنية الشبكات المعرّفة برمجياً.

ب. طرح آلية فعّالة وخفيفة لمواجهة هذه الهجمات بالاعتماد على الأنتروبية وذلك بغرض حماية المتحكم.

جـ. تنجيز وطرح الآلية في بيئة Mininet [2].

د. البرهان على نجاعة الحل من خلال عمليات المحاكاة.

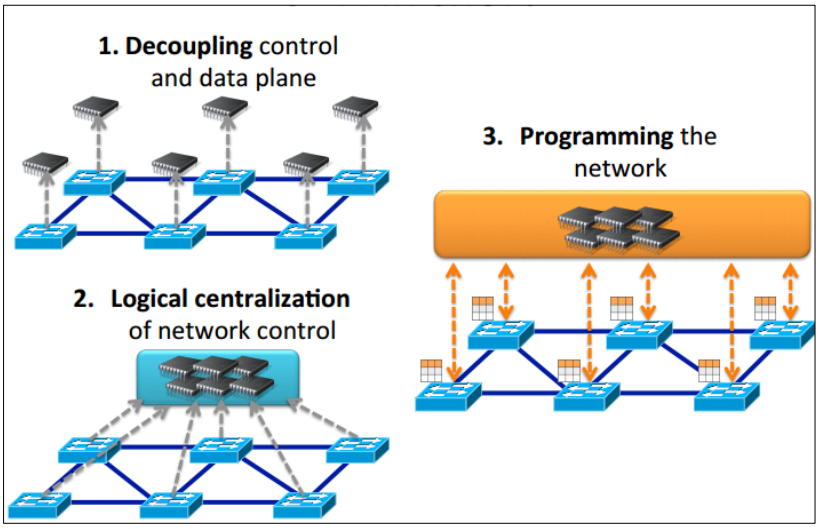
- الجودة تبدأ من الداخل ثم تحفر طريقها للخارج (بوب مواواد).

1.2 مقدمة عامة عن الشبكات المعرفة برمجياً SDN:

سوف نقوم فيما يلي بإعطاء لمحة عن الشبكات المعرّفة برمجياً SDN، وعن المتحكم أو البروتوكول OpenFlow والذي يُعتبر أساس هذه الشبكات.

1.1.2 السمات الأساسية لشبكات SDN:

جاءت فكرة الشبكات المعرّفة برمجياً SDN من أجل تسهيل عملية إدارة الشبكات ولتؤمّن مرونة في التحكم بكامل الشبكة دون الحاجة إلى الوصول إلى كامل الأجهزة الموجودة ضمن الشبكة، والتي عادة ما تكون جغرافياً متباعدة جداً. يوضح لنا الشكل (2-1) المفهوم العام لهذه الشبكات، حيث تتصف بثلاث سمات رئيسية: السمة الأولى هي فصل طبقة المعطيات عن طبقة التحكم، الثانية: مركزية التحكم بالأجهزة الموجودة ضمن الشبكة، والثالثة هي برمجة المتحكمات المركزية. أيْ أنّه أصبح لدينا ضمن هذه الشبكات جزء مسؤول عن إدراة الأجهزة يمكن تشبيهه بالدماغ (متحكمات خاصة) وأجهزة تقوم فقط بالاستجابة لأوامر المتحكمات ويمكن أن نشبّهها بالأذرع (مثل المبدّلات switches والموجهات routers..).



الشكل(2-1). سمات الشبكات المعرّفة برمجياً SDN.

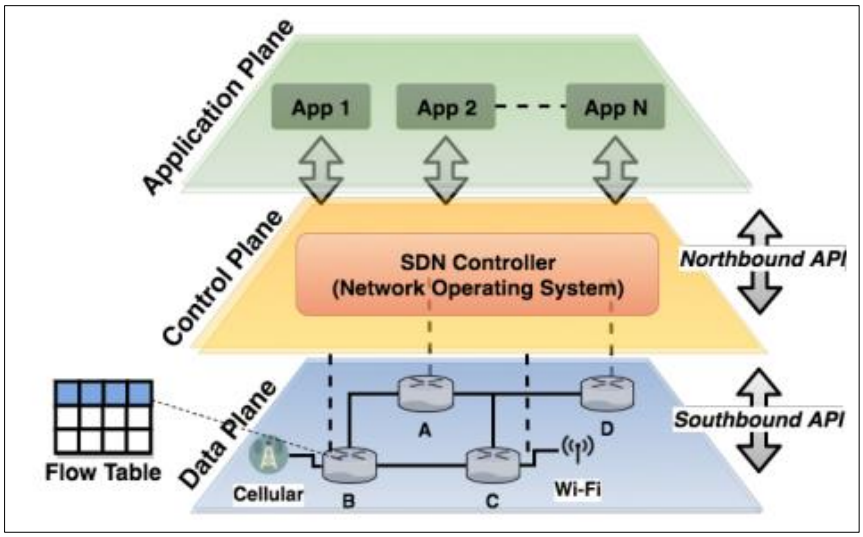
2.1.2 بنية شبكات SDN:

تمَّ تقسيم بنية شبكات SDN إلى ثلاث طبقات موضحة في الشكل (2-2):

1. طبقة التطبيقات Application Layer: تتكون هذه الطبقة من الخدمات والتطبيقات وأدوات المجانسة الخاصة بالشبكة والتي تُستخدم للتفاهم وتقديم الأوامر للمتحكمات، كما أنّها تفتح المجال أمام إمكانية التواصل مع الأجزاء التحتية لشبكات SDN.

2. طبقة التحكم Control Layer: تتكون هذه الطبقة من المتحكمات المركزية المفصولة عن البنية التحتية والتي تقوم بالتزويد بنظرة عامة، شاملة، ومركزية لكامل الشبكة. تَستخدم هذه الطبقة بروتوكولOpen Flow للتخاطب مع طبقة المعطيات (طبقة البنية التحتية).

3. طبقة المعطيات Data Layer أو البنية التحتية infrastructure لشبكة SDN:وهي الطبقة الدنيا ضمن الهيكلية العامة لشبكة SDN وتتكون من أجهزة الشبكة الوهمية والفيزيائية مثل المبدلات أو الموجهات، وكل هذه الأجهزة تدعم بروتوكول Open Flow لتنفيذ قواعد تسيير المعطيات.



الشكل(2-2) هيكلية شبكات SDN.[40]

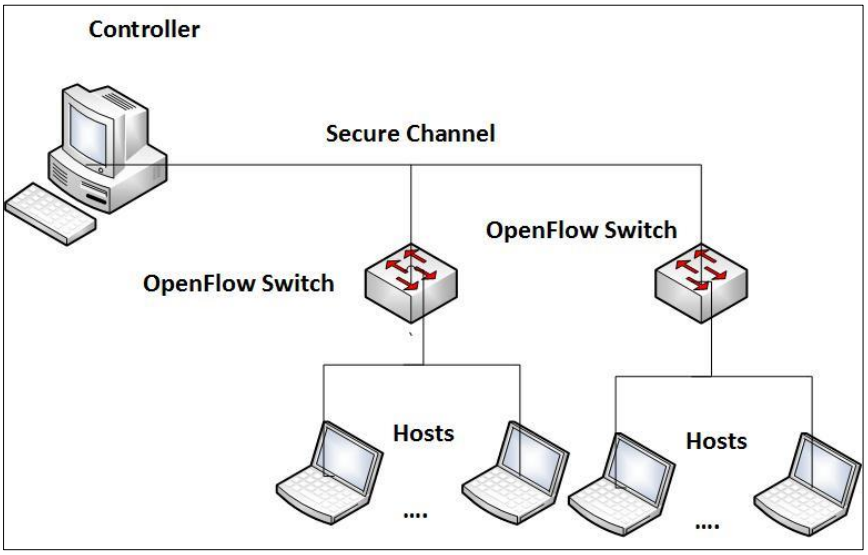
3.1.2 مصطلحات ومفاهيم عامة في شبكات SDN:

– واجهة برمجة التطبيقات الجنوبية Southbound API (Southbound Application Programming Interface): وهي التطبيقات التي تسمح للمتحكم بتحديد سلوك البنية التحتية لشبكة SDN, والتي هي طبقة تسيير المعطيات Forwarding Layer.[3]

– واجهة برمجة التطبيقات الشمالية Northbound API (**Northbound** Application Programming Interface): وهي التطبيقات التي تسمح لطبقة التطبيقات Application Layer الموجودة بالأعلى ضمن هيكلية بينة شبكة SDN . بأخذ نظرة عامة للشبكة وإدارة عمل المتحكمات والشبكة ككل.[3]

3.2 بروتوكول OpenFlow:

يُعتبر بروتوكول OpenFlow بمثابة العصب الرئيسي في الشبكات المعرّفة برمجياً، حيث يقوم بإدارة المبدّلات switches في الشبكة ويسمح للمكوّنات الخارجية مثل المتحكم controller بتعديل دفق المعطيات في الشبكة. في الآونة الأخيرة، بدأت العديد من الشركات المصنّعة بتصنيع المبدّلات، حيث تمتلك هذه المبدّلات جداول [4] تعرض ضمنها مسارات الدخول والخروج (ingress,egress) لجميع الطرود الخاصة بهذا المبدّل. يقوم بروتوكول OpenFlow بجعل هذه الجداول متاحةً أمام المتحكم. يستقبل المبدّل OpenFlow مدخلات الجدول الخاص به وعمليات التعديل من المتحكم وذلك من خلال قناة آمنة. يعرض لنا الشكل (2-3) نموذجاً مبسطاً لشبكة معرّفة برمجياً.



الشكل(2-3). شبكة بسيطة لمبدّلات OpenFlow.

عندما يصل طرد جديد إلى مبدّل OpenFlow، يقوم بالبحث ضمن جدول الدفق عن تطابق. في حال لم يجد هنالك تطابق، فإنَّ المبدّل يقوم بإرسال الطرد إلى المتحكم. يقوم المتحكم بمعالجة الطرد ويقوم بتعليم الطرد مع الحدث كمايلي:

- إضافة دفق جديد للطرود المشابهة الواردة.
- إهمال الطرود المشابهة.
- إشارة إلى الطرد مع رقم تعريف الرتل queue ID.

1.3.2 سمات بروتوكول OpenFlow:

تعتمد كل من عمليات البحث، المطابقة، التسيير والطلبات القادمة من المتحكم على سمات وخصائص بروتوكول OpenFlow المعلن عنها من قبل مؤسسة Open Networking Foundation. سنستخدم في عملنا النسخة 1.0. [5]

2.3.2 مبدّل OpenFlow:

يتكون المبدّل OpenFlow من جدول دفق أو مجموعة من هذه الجداول وقناة آمنة للاتصال مع المتحكم. يوجد ضمن كل جدول حقل مطابقة، عدّاد، ومجموعة من التعليمات لكل دخل. تشمل عملية المطابقة في المبدّل حقولاً مختلفة من ترويسات الطرود. يوضح الجدول(2-1) هذه الحقول التي يقوم المبدّل باستخدامها لإيجاد المطابقة ضمن جدوله. في الجدول، هنالك حقول بيانات وصفية metadata (السطر الثاني في الجدول(2-1)) تُعرّف على أنّها سجلّات تحمل معلومات الترويسة من جدول إلى آخر. عادةً ما يحتوي المبدّل على عدة جداول يتم استخدامها بشكل متفرع أو على التوازي. يتم نقل الطرد من جدول إلى آخر للمطابقة وحمل المعطيات الوصفية metadata، في حال تم إيجاد عملية مطابقة، يتم تحديث علامة البيانات الوصفية metadata وفقاً لذلك.

يتم فحص أية طرود تدخل المبدّل أمام جميع الجداول. في حال تمَّ إيجاد تطابق، يتم تطبيق الأجراء المناسب على ذلك الدخل ويتم تحديث العدّاد الخاص بهذا الدخل. يشمل العدّاد عدداً من المكونات في المبدّل مثل عدّادات دخل الدفق، الجدول، البوابة، الرتل وغيرها. مثال على ذلك، تشير الفترة الزمنية duration إلى كمية الوقت التي قضاها الدفق ضمن الجدول.

في حال لم يتم إيجاد تطابق، سيتم إرسال الطرد إلى المتحكم. في النسخة 1.3، في حال لم يوجد هنالك تطابق يتم إهمال الطرد. لذلك فإنّه وعلى اعتبار أنّ ترويسة الطرد لا تحتوي على أية من الحقول المذكورة في الجدول(2-1)، سيتم اعتبار الطرد غير شرعي أو غير صالح.

الجدول(2-1). حقول المطابقة في ترويسة الطرد.

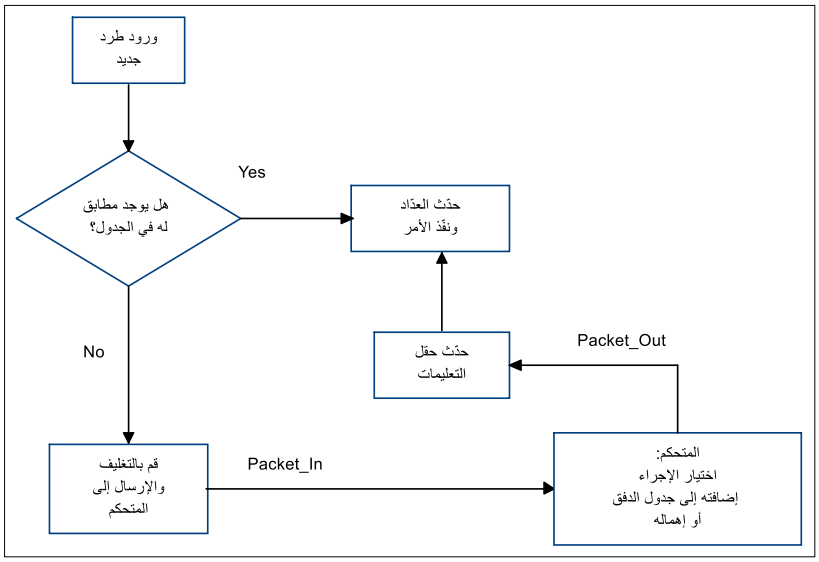
حقول الترويسة
Ingress port
Metadata
Ether Src
Ether Type
Vlan id
VLAN priority
MPLS label
MPLS Traffic class
IPv4 Src
IPv4 Dst
IPv4 proto/ARP opcode
IPv4 ToS bits
TCP/UDP/SCTP src port ICMP Type
TCP/UDP/SCTP dst ICMP code

يمكن إرسال الطرد كاملاً إلى المتحكم، أو يمكن للمبدّل تخزين الحمل payload وإرسال الترويسة فقط. عندما يتم إرسال الطرد إلى المتحكم، يتم تغليف الطرد وتعليمه على أنّه رسالة OFTP_PACKET_IN. وسيتم الإشارة إليه على أنه طرد وارد Packet_In.

بالأخذ بعين الاعتبار كل من عدد المبدّلات، الوقت أثناء اليوم، حجم الطرد، الأولوية وغيرها من المعاملات، يقوم المتحكم بمعالجة هذه الطرود وإرسال الإجابة مع الإجراء الواجب تطبيقه على هذا الطرد والطرود القادمة من المنبع نفسه. بالتالي، لا يبقَ على المبدّل إلا تنفيذ الأمر الوارد إليه من المتحكم بخصوص هذا الطرد الجديد.

يوجد العديد من الإجراءات التي يقوم المتحكم بإرسالها إلى المبدّل مثل: تسيير، إهمال، ضعه في الرتل، جودة الخدمة، تعديل علامة الـ VLAN، تعديل العنوان الفيزيائي MAC أو العنوان المنطقي. تعدّ إجراءات التسيير والإهمال أساسية، أما إجراءات تعديل الحقول والوضع ضمن رتل فإنها اختيارية. يتم اختيار الإجراء ضمن المتحكم ومن ثم يتم إرساله إلى المبدّل برسالة Packet_Out.

يعرض لنا الشكل(2-4) إجرائية دخل الدفق. في حال تمَّ وسم الطرد مع الإجراء "تجاهل drop"، يتم إضافة دخل الدفق، وبالتالي، أية طرد سوف يطابق هذا الدفق، سوف يتم إهماله. في حال لم يتم استقبال طرود مماثلة لهذا الدفق، تتم إزالة هذا الدفق من الجدول بعد فترة معينة time out.



الشكل(2-4). إجرائية دخل الدفق.

3.3.2 القناة الآمنة:

تعدّ هذه القناة هي الممر الوحيد لعمليات التخاطب بين طبقة المعطيات وطبقة التحكم في الشبكات المعرّفة برمجياً. يتم تأسيس الاتصال في هذه القناة إما ببروتوكول TLS (Transport Layer Security) أو TCP

connection بين المتحكم والمبدّل. في حال فقدان الاتصال، يحاول المبدّل الاتصال بالمتحكم الاحتياطي في حال وجوده. في حال كان باستطاعة المبدّل العمل في الشبكات المعرّفة برمجياً والشبكات غير المعرّفة برمجياً Non-SDN فإنّه يدعى بِـ Hybrid switch، لكن في هذه الحالة لن يقوم المبدّل باتّباع بروتوكول OpenFlow وستفقد الشبكة بنيتها المعرّفة برمجياً.

4.3.2 نظام تشغيل الشبكة Network Operating System:

إنّ نظم التشغيل في هذه الشبكات هي أنظمة موزعة تعمل على خلق نظرة عامة ومحدَّثة بشكل دائم للشبكة ككل، وتعمل في المخدمات أيّ المتحكمات في الشبكة. تَستخدم البروتوكولات المفتوحة من أجل:

1. جلب معلومات الحالة من عناصر التوجيه أو التسيير.
2. إعطاء الأوامر التحكمية إلى عناصر التسيير.

- المبدع كبير بنقاط ضعفه كما أن الماسة جميلة بشرخ صغير فيها أو ضمور في ضوئها وفي ذلك ما يميزها عن الماس الصناعي. (غادة السمان)

1.3 مقدمة عن المشاكل الأمنية التي تعاني منها شبكات SDN:

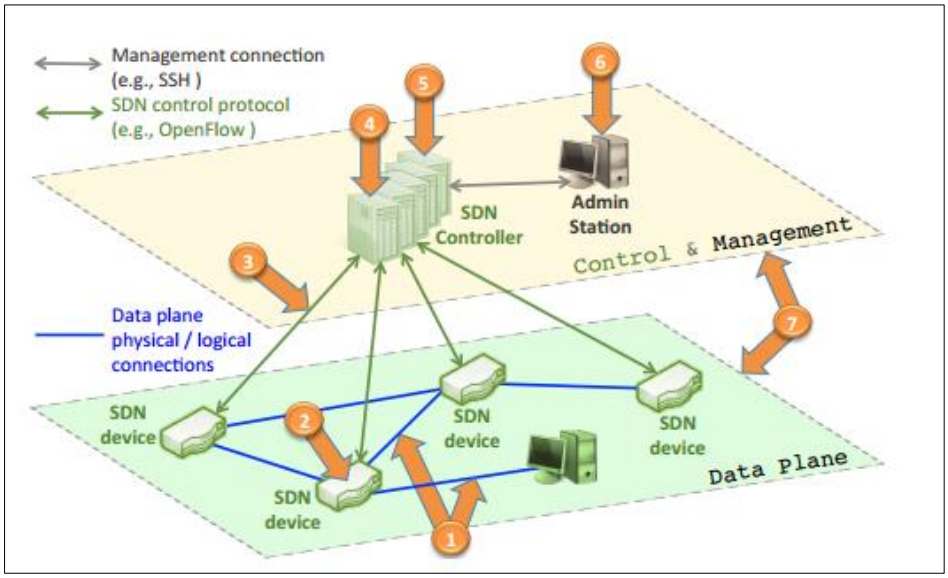
صحيحٌ أن عملية فصل طبقة المعطيات عن طبقة التحكم أدَّت إلى قفزة نوعية في عالم الشبكات، لكن أدّت هذه الخطوة إلى نشوء ثغرات جديدة لم تكن موجودة في الشبكات التقليدية، إلّا أنّه وخلال الفترة السابقة، بدأ الاهتمام بشكل أكبر وأصبح هنالك توجُّه حقيقي لمعالجة قضايا الأمن والوثوقية ضمن شبكات SDN، لذلك بدأت العديد من الأبحاث تتناول هذه الأخطار ونقاط الضعف والتهديدات التي نجمت عن هذه التقنية الجديدة، وبدأت تقدّم هذه الأبحاث بعض الحلول التي ينبغي أخذها بعين الاعتبار منذ الخطوة الأولى في بناء شبكة SDNوالتي قد تساهم بتفادي تلك التهديدات ومواجهتها وتخفيف خطرها.

تتمتع الشبكات التقليدية بمناعة طبيعية ضد الهجمات الشائعة نظراً لِـ: طبيعة أجهزتها المغلقة، تصميمها الثابت، تجانس البرمجيات والتحكم اللامركزي، فمثلاً إنِ استغلَّ أحد المهاجمين نقطة ضعف للأجهزة المصنعة من قبل شركة ما، فإنَّ الشبكة سوف تتأثر فقط في الجزء الذي يحوي على أجهزة تابعة لنفس الشركة، أما باقي الأجهزة فلن تتأثر على اعتبار أنها تتبع لشركات مصنّعة أخرى. أما في شبكات SDN فوجود بروتوكول Open Flow المشترك بين جميع الشركات سوف يزيد من خطورة التهديدات ونشر أعطال مشتركة بين جميع الشركات. إذاً، أحدثت شبكات SDN فكرة رائعة في عالم الشبكات لكن قامت بزيادة سطح الخطر والتهديدات، مما أوجب لزاماً مناقشة قضايا الأمن والوثوقية security and dependability والحلول الواجب أخذها بعين الاعتبار عند تصميم شبكة SDN.

2.3 التهديدات والأخطار التي تعاني منها شبكات SDN:

تمتلك شبكات SDN سمتين أساسيتين تجلعها مصدر جذب للمهاجمين والمخترقين ومصدر قلق لأصحاب هذه الشبكات: 1. برمجة الشبكة باستخدام برمجيات software 2. مركزية التحكم بالشبكة، بالتالي فإنَّ الوصول

لأحد المتحكمات يعني الوصول والتحكم بكامل الشبكة. سنورد فيما يلي أهم التهديدات التي قد تواجهنا في شبكات SDN والتي نجمعها في الشكل (3-1) مع الحلول البسيطة المقترحة: [6،40]



الشكل(3-1). أهم التهديدات في شبكات SDN.

1. حقن معطيات مزورة أو مزيفة: حيث يمكن مهاجمة المبدلات أو الموجهات من خلال وجود أجهزة معطلة في الشبكة أو من خلال مهاجم خبيث يستخدم أحد مكونات الشبكة (موجه، مبدل، مخدم..الخ) وذلك لإطلاق طرود بأعداد كبيرة من أجل تحقيق هجوم قطع الخدمة DoS (Denial of Service) والتي قد تكون مثلاً ضد المبدلات التي تعمل ببروتوكول Open Flow وذلك من أجل استهلاك جميع الذواكر TCAM (Ternary Content Addressable Memory) الموجودة ضمن المبدل. **الحل المقترح:** استخدام أنظمة كشف التسلل IDS (Intrusion Detection System) مدعومة بأنظمة معرفة السبب الحقيقي المسبِّب للمشكلة Runtime Root-Cause Analysis وذلك لكشف السلوك الغير طبيعي لعناصر الشبكة، بالإضافة إلى آليات من أجل التحكم الديناميكي بسلوك المبدل (مثلاً: وضع حَدّ معين لمعدل طلبات التحكم).

2. الهجوم على نقاط الضعف الخاصة بالمبدلات: حيث أنَّ الهجوم أو السيطرة على مبدّل واحد ذلك يعني إمكانية: تجاهل طرد ما، إعادة توجيه طرد ما إلى وجهة خاطئة، نسخ طرود معينة، أو إبطاء تسيير الطرود ضمن الشبكة أو حتى حقن معطيات أو طلبات وهمية في الشبكة وذلك لإسقاط المتحكمات أو المبدلات المجاورة. **الحل المقترح:** استخدام آليات من أجل إجراء عمليات المصادقة على البرامج مثل أنظمة إدارة الثقة الذاتية للمكونات البرمجية، أو استخدام آليات لمراقبة أو كشف السلوك الغير طبيعي لأجهزة الشبكة.

3. الهجوم على اتصالات طبقة التحكم: والذي قد يُستخدم لإجراء هجوم DoS أو لسرقة معطيات. حتى لو تم استخدام تقنية التعمية TLS/SSl من أجل هكذا اتصالات، إلا أنَّ خطوط طبقة التحكم تبقى مهدَّدة وخاصة وأن هنالك العديد من الأبحاث تشير إلى نقاط الضعف الخاصة بـ TLS/SSL حيث أنها تعتمد على بنية PKI (Public Key Infrastructure) لتبادل المفاتيح العامة. إنَّ أمن هذه الاتصالات يكون قوياً بقدر قوة أضعف خطوطها أو عناصرها، وقد يكون هذا الضعف ناجماً عن شهادات موقَّعة ذاتياً، أو جهة غير آمنة تمنح شهادات رقمية، أو تطبيقات ومكتبات ضعيفة، بالتالي هذا يفتح المجال أمام ما يسمى هجوم الرجل في الوسط man-in-the-middle. في حال نجح أحد المهاجمين بالسيطرة على اتصالات طبقة التحكم، فإنه يستطيع تجميع قوة كافية (وبحسب عدد المبدلات التي سوف تصبح تحت متناول يده) أنْ يشنَّ هجوم قطع خدمة موزع DDoS. **الحل المقترح:** هو تأمين الاتصال بين نسخ المتحكمات عن طريق تشفير هذه الطبقة. بالإضافة إلى ذلك يمكن استخدام آليات ديناميكة ومضمونة لربط الأجهزة وذلك لضمان الثقة بين أجهزة طبقة التحكم وأجهزة طبقة المعطيات.

4. الهجوم على نقاط الضعف في المتحكم: والتي ربما تكون أشدُّ التهديدات خطورة في شبكات SDN. إنّ خلل وحدة تحكم واحدة أو إصابتها بهجوم خبيث، يمكن أن يُسقط الشبكة بكاملها. كما أنّ استخدام نظام كشف التسلل IDS (Intrusion Detection System) قد لا يكون كافياً، لأنه من الصعب إيجاد التجميعة

الدقيقة للأحداث التي قد تؤدي إلى توليد سلوك معين، والشيء الأكثر أهمية هو معرفة أنَّ هذا السلوك هو سلوك خبيث. بشكل مشابه أيضاً، يمكن للتطبيقات الخبيثة في الشبكة أن تفعل ما يحلو لها في الشبكة على اعتبار أن المتحكمات فقط هي التي تزود الشبكة بالتجريدات التي تُترجم إلى أوامر تحكمية إلى البنية التحتية. **الحل المقترح:** يمكن استخدام العديد من التقانات مثل التكرار أو النسخ (لكشف، إزالة، أو إخفاء السلوك الغير طبيعي)، توظيف مسألة التنوع (للمتحكمات، البروتوكولات، لغات برمجة، أو النسخ برمجية إلخ)، وإعادة التهيئة أو الاسترداد (التحديث الدوري للنظام للوصول إلى الحالة الموثوقة والسليمة). أيضاً، من المهم تأمين كل العناصر الحساسة داخل المتحكم (مفاتيح التشفير مثلاً). علاوة على ذلك، إنَّ استخدام سياسات أمنية تضمن التطبيق الصحيح لسلوك تلك التقانات، وتُقيِّد الواجهات التي يمكن استخدامها من قبل التطبيقات وماهي القواعد والأوامر التي تستطيع هذه التطبيقات توليدها لبرمجة الشبكة.

5. عدم وجود آليات لضمان الثقة بين المتحكمات وتطبيقات الإدارة: حيث على غرار التهديد رقم 3، تفتقر المتحكمات والتطبيقات إلى القدرة على إقامة علاقات ثقة. يكمن الفرق الرئيسي عن التهديد المشار إليه في الطريقة التي تُنشأ فيها الشهادة، حيث أنَّ التقنيات المستخدمة للمصادقة على أجهزة الشبكة تختلف عن تلك المستخدمة للتطبيقات. **الحل المقترح:** استخدام آليات لإدارة الثقة ذاتياً تضمن الثقة بالتطبيقات طوال فترة عملها.

6. الهجوم على نقاط ضعف محطات الإدارة: والتي عادةً ما تكون موجودة ضمن الشبكات التقليدية، تُستخدم هنا أيضاً في شبكات SDN للنفاذ إلى المتحكم بالشبكة، لكن الفرق في أنه إذا ما تعرض جهاز أو محطة إدارة واحدة فقط للخطر، فإن هذا الخطر سوف يزداد بشكل دراماتيكي في شبكات SDN، حيث سوف يكون من السهل إعادة برمجة الشبكة وذلك من مكان واحد. **الحل المقترح:** استخدام البروتوكولات التي تتطلب التحقق

المزدوج (مثال على ذلك، طلب النفاذ إلى المتحكم يتطلب تفويض من قبل شخصين اثنين). أيضاً استخدام آليات استرداد مضمونة لضمان حالة موثوقة بعد إعادة التشغيل.

7. عدم وجود مصادر موثوقة من أجل التوصيف و التعافي: والتي قد تسمح بفهم سبب المشكلة التي تم كشفها و معالجتها للعودة بسرعة إلى الوضع الآمن. من أجل التحقيق والتثبت حول حادث ما، نحن بحاجة إلى معلومات موثوقة من جميع الأجزاء والمجالات المكونة للشبكة. علاوة على ذلك، هذه المعلومات سوف تكون مفيدة فقط إذا كانت مضمونة الوثوقية، وبشكل مشابه، التعافي يتطلب مراقبة آمنة وموثوقة للنظام لضمان الاسترداد السريع والصحيح لعناصر الشبكة إلى الحالة التي كانت تعمل فيها. **الحل المقترح:** استخدام آليات التسجيل والتتبع، حيث أنَّ هنالك حاجة لها في مستوى المعطيات والمستوى التحكمي، ومع ذلك حتى تكون فعَّالة فإنها يجب أن لا تُمحى أو أن تكون غير قابلة للتغيير. علاوة على ذلك يجب تخزين السجلات ضمن البيئات النائية والآمنة.

3.3 الأمن والوثوقية ضمن شبكات SDN:

يناقش المؤلفون في [6،40] بعض المفاهيم الأساسية عن الأمن والوثوقية ويقترحون بعض الآليات والتقنيات التي يجب أخذها بعين الاعتبار عند تصميم منصة آمنة وموثوقة للتحكم ضمن شبكات SDN.

1.3.3 أساسيات:

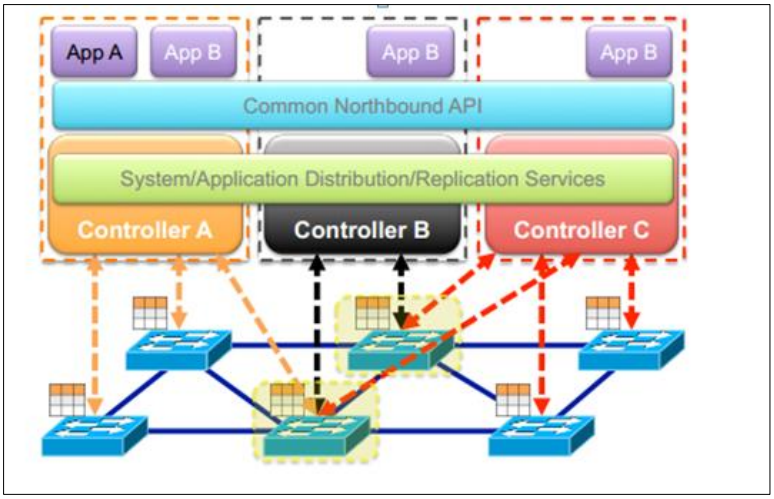
لا يوجد حتى اليوم بحسب رأي المؤلفين أي متحكم SDN يراعي قضايا الأمن والوثوقية، إما من خلال استخدام تقنيات تحقق بسيطة أو حتى نسخ معطيات التحكم بين المتحكمات المنسوخة، فمثلاً: لا يوجد آليات تُستخدم لضمان ارتباط متحكم-مبدل موثوق، أو آليات لكشف، تصحيح، أو إخفاء أعطال أحد أجزاء الشبكة، أو آليات للتأكد من سلامة وسرية المعطيات المتناقلة بين المتحكمات. من منظور الأمن والوثوقية، إنَّ أحد أهم

المفاتيح لضمان نظام قوي للغاية هو التسامح أو تحمل الخطأ وأعباء التسلل الغير شرعي إلى النظام. أهم نموذجان للخطأ هما نموذج التحطم ونموذج بيزنطة. أيضاً يعتبر إنشاء بنى تحمل التسلل (Intrusion-tolerant architecture) خطوة في طريق بناء نماذج الأمنية الذاتية، حيث أنَّ أنظمة تحمل التسلل تبقى تعمل بشكل صحيح وتبقى قادرة على ضمان الخواص مثل السرية، السلامة، التوافر بالرغم من غياب أجزاء مخترقة أو معطلة بسبب الهجوم الناجح.

نتحدث فيما يلي عن الحلول والمقترحات لبناء منصة تحكم موثوقة وآمنة ضمن شبكات SDN:

2.3.3 منصة التحكم الموثوقة والآمنة:

يقدّم المؤلفون في [6] التصميم العام المقترح لمنصة تحكم بشبكة SDN آمنة وموثوقة، حيث يعرض الشكل (3-2) نظرة مبسطة لبنية التصميم، ثم لاحقاً سوف نعرض بعض الآليات التي تم اقتراحها لمعالجة التهديدات الآنفة الذكر.



الشكل(3-2). منصة التحكم الآمنة والموثوقة لشبكة SDN.

سوف سنسرد فيمايلي أهم الطرق التي تم اقتراحها من أجل الحصول على منصة تحكم بشبكة SDN آمنة وموثوقة:

1. التكرار أو النسخ المتماثل Replication:

تُعدُّ واحدة من أهم الآليات المستخدمة في تأمين الوثوقية في شبكات SDN، كما نلاحظ في الشكل (3-2) تم نسخ المتحكم إلى ثلاثة متحكمات، ويجب أن تُنسخ التطبيقات أيضاً. إلى جانب تكرار المتحكم في الشكل، نلاحظ تكرار التطبيق B أيضاً في جميع نسخ المتحكمات. إنّ إجرائية الخلط هذه تضمن تحمّل الأعطال في الجزأين الصلب والبرمجي سواءً كانت الأعطال مقصودة أو عرضية، ويتيح استخدام النسخ إمكانية إخفاء الأعطال وعزل التطبيقات أو المتحكمات الخبيثة أو المعطلة. أكثر من ذلك، في حال تقسيم الشبكة، يبقى التطبيق B ومع وجود خوارزميات اتساق مناسبة قادراً على برمجة جميع المبدلات على عكس التطبيق A.

2. التنوع Diversity:

يعدّ التنوع طريقة أخرى من أجل إكساب المتانة والقوة إلى الأنظمة ذات الوثوقية والأمن، المبدأ الأساسي خلف هذه التقنية هو تجنب الأخطاء المشتركة (مثل نقاط الضعف البرمجية أو خلل برمجي). مثال على ذلك، من المعروف أن أنظمة التشغيل من العائلات المختلفة تمتلك العديد من نقاط الضعف الغير مشتركة، هذا يعني أن التنوع في أنظمة التشغيل يحدُّ من التأثير الكلي للهجمات عليها لأن هنالك هجمات تؤثر على نقاط ضعف معينة في نظام تشغيل ما دون أن تؤثر على نظام تشغيل ما آخر. في شبكات SDN يمكن تشغيل التطبيق نفسه لإدارة متحكمات مختلفة مثل تطبيقات Northbound API.

3. آليات التعافي الذاتي Self-Healing mechanisms:

يمكن باستخدام الاسترداد التفاعلي أو الاستباقي (proactive or reactive recovery) أن يتم جلب النظام إلى طور العمل الصحيح وذلك باستبدال الأجزاء المتضررة والمحافظة عليها تعمل بالشكل السليم أكثر وقت ممكن. عندما يتم استبدال الأجزاء، فإنه من المهم أن يتم استبدالها بإصدارات جديدة ومتنوعة كلما كان ذلك ممكناً.

بمعنى آخر، يجب أن نتوخى التنوع في إجرائية الاسترداد وأن نعزِّز الدفاع ضد الهجمات التي تستهدف نقاط ضعف معينة للنظام.

4. ربط الأجهزة ديناميكياً Dynamic device associated:

إذا كان لدينا مبدل مرتبط مع متحكم وحيد controller، أمكننا القول أنَّ التحكم بهذا المبدّل غير متسامح أو لا يتحمل الأعطال، لأنه وببساطة إذا سقط (تعطّل) المتحكم فإن المبدّل سوف يسقط حتماً وبالتالي يصبح بحاجة إلى الارتباط بمتحكم آخر، لهذا السبب، يجب أن يمتلك المبدّل القدرة على الارتباط بشكل ديناميكي بعدة متحكمات وبطريقة آمنة (مثلاً: باستخدام التشفير وذلك لكشف المتحكمات الخبيثة والتحقق منها ولمواجهة أحد الأخطار الشهيرة وهو هجوم الرجل في الوسط). سيغدو المبدّل المتّصل بعدة متحكمات قادراً على التسامح مع الأخطاء بشكل آلي. يضمن هذا الإجراء مزايا أخرى وهي زيادة دفق المستوى التحكمي (العديد من المتحكمات يمكن استخدامها في توزيع الحمل(load balancing) وتقليل تأخير التحكم باستخدام المتحكم ذو الاستجابة الأسرع.

5. الثقة بين الأجهزة والمتحكمات:

تُعد مسألة بناء الثقة بين الأجهزة والمتحكمات أمراً مهماً جداً من أجل وثوقية المستوى التحكمي بشكل كامل، حيث ينبغي أن يتم السماح لأجهزة الشبكة بالارتباط بشكل ديناميكي بالمتحكمات لكن بشكل لا يسبّب تخفيض مستوى العلاقات الموثوقة. يمكن تطبيق مقاربة بسيطة وهي آلية التحقق من قائمة معروفة لأجهزة موثوقة (قائمة بيضاء)، محفوظة داخل المتحكم. إنّ الخيار الآخر هو الثقة بجميع المبدلات بشكل عام حتى الوصول إلى حالة ينبغي فيها التحقق من مدى وثوقية مبدّل ما (نتيجة لسلوك غير طبيعي نجم عنه)، عندها يتم فحص الوثوقية الخاصة به بشكل دقيق. يمكن إرسال تقرير عن سلوك غير طبيعي أو سلوك خبيث من خلال المبدّلات أو المتحكمات بالاعتماد على خوارزميات مُعدَّة لكشف الأعطال. في حال تم فقدان الوثوقية بأحد المبدلات أو

المتحكمات وأصبحت تحت عتبة معينة، يتم فوراً تجميد هذا المبدل أو إجراء حجر صحي عليه بشكل أوتوماتيكي من قبل جميع الأجهزة والمتحكمات.

6. الثقة بين التطبيقات وبرمجيات المتحكمات:

بما أنَّ البرمجيات تعاني بحد ذاتها من مسائل: الاستخدام الطويل الأمد، الأعطال، الخلل والهجمات، كل هذه المسائل وغيرها أوجبت إيجاد نموذج ثقة ديناميكي. يستعرض المؤلفون في [6] الجدوى الممكنة لاستخدام نموذج يدعم إدارة الثقة الذاتية في أنظمة البرمجيات. يستخدم المؤلفون هنا فكرة شاملة للثقة تسمح للطرف طالب الثقة أن يقيمِّ مقدار الثقة بالجهاز المطلوب الثقة به من خلال مراقبة سلوكه وقياس موصفات الجودة مثل: التوافر، الوثوقية، السلامة، الأمان، الصيانة، والسرية. النموذج المقترح أيضاً يمكن أن يطبق لمراقبة وضمان الوثوقية في العلاقات بين كيانات النظام.

7. المجالات الأمنية:

المجالات الأمنية المعزولة هي نوع شائع من التقانات المستخدمة في مختلف أنواع الأنظمة. مثال على ذلك، في أنظمة التشغيل، لايُسمح للتطبيقات التي بمستوى المستخدم، الولوج إلى الأنظمة الفرعية في مستوى النواة. يمكن تحقيق العزل في منصات التحكم بشبكات SDN عن طريق استخدام تقنيات مشابهة للصناديق الرملية sandboxes أو الفصل الافتراضي virtualization. تتيح هذه التقنيات تصميم نماذج عزل قوية من خلال التعريف الجيد للواجهات interfaces التي تسمح بأقل عدد ممكن من الاتصالات والعمليات بين المجالات الأمنية المختلفة.

8. المكونات الآمنة:

تعدُّ المكونات الآمنة واحدة من أهم اللبنات الأساسية في بناء نظام آمن وموثوق. يمكن استخدام العناصر الأمنية لتوفير قواعد الحوسبة الموثوقة TCB **(** Trusted Computing Base **)** لضمان خواص أمنية مثل السرية.

9. التحديث والترميم السريع والموثوق للبرمجيات :

على اعتبار أنه لايوجد أي برنامج خالٍ من العيوب أو نقاط الضعف، فإنَّ عملية التحديث والتصحيح السريعة والموثوقة للبرمجيات مهمة جداً لتقليل حجم نقاط الضعف. لذلك يجب توزيع منصات التحكم مع آليات تقوم بالتحديثات بطريقة سلسلة وآمنة.

للتلخيص، لدينا الجدول (3-1) الذي نعرض فيه الحلول المقترحة والتهديدات التي ممكن أن تحلها [6]:

الجدول (3-1). الآليات المقترحة مع التهديدات التي تواجهها.[6]

الآلية المقترحة	التهديدات
التكرار	1،4،5،7
التنوع	3،4،6
التعافي الذاتي	2،4،6
الربط الديناميكي	3،4
الثقة بين الأجهزة والمتحكمات	1،2،3
الثقة بين التطبيقات والمتحكمات	4،5
المجالات الأمنية	4،5
المكونات الآمنة	4،5،7
التحديث والترميم السريع والموثوق للبرمجيات	2،4،6

ويجُمل المؤلفون في[6]بالقول بأنه للحصول على نظام آمن وموثوق في شبكات SDN فإنه ينبغي توخي السمات الثلاث التالية: التنوع، النسخ المتكرر، والارتباط الديناميكي للأجهزة.

كما ذكرنا سابقاً، هنالك العديد من الأبحاث التي بدأت تُعنى بمسألة الأمن والوثوقية بشبكات SDN، يتناول البحث [7] مثلاً، بعض المشاكل التي جاءت مع اختراع فكرة SDN ويتحدث عن نقاط هامة أخرى مثل مراعاة أمن نظام التشغيل operating system، حيث أنَّ التغيير الذي قد يطرأ على نظام تشغيل الشبكة سوف يؤدي إلى تبديل أو تدمير مكونات الشبكة أو تدمير كامل نظام التشغيل لعناصر الشبكة مثل المتحكمات والمسيرات، ويقترح ضمان أمن نظام التشغيل عن طريق استخدام أنظمة TCB. يؤدي أيضاً التلاعب ببرمجيات الشبكة إلى تخريب عمل مكونات الشبكة ويمكن ضمان أمنها باستخدام TCB أيضاً. إنَّ خلل نظام التشغيل أو البرمجيات برأي المؤلف سوف يؤدي إلى خلل في كامل طبقات شبكة SDN، ويصنِّف المؤلفون الأخطاء بشكل عام إلى أخطاء برمجية وأخطاء صلبة، وأنه يمكن تفادي كلاهما بتطبيق تقنيات مضمونة، لكن الفرق بينهما أنَّ الأخطاء البرمجية قد تؤثر على كامل طبقات الشبكة أما الأخطاء الصلبة فتؤثر فقط على طبقتي المعطيات والتحكم.

ويذهب المؤلفون أبعد من ذلك ليصنِّفوا الهجمات ويحدِّدوا تماماً الطبقات التي تؤثر عليها، فهجوم قطع الخدمة DoS يؤثر على الطبقات الثلاثة، أما الهجوم من جانب القناة side channel attack وهو الهجوم الذي يقوم المهاجم فيه بإدخال طرد ذو حجم معين ليستكشف جدول الدفق flow rules لدى المبدّل وذلك عن طريق تحليل الزمن الذي استغرقه المبدل لمعرفة القاعدة التي سوف يطبّقها على هذا الطرد، فإنه يؤثر فقط على طبقة المعطيات. فيمايلي نعرض بعض الهجمات التي تناولتها الباحثون في [7] وآليات تجنّبها:

يطرح هنا المؤلف مسألة حماية المتحكم لأنه يعدُّ عصب الشبكة، حيث يستطيع المهاجم أن يسرق عنوان المتحكم IP Spoofing ويجعل الطرود تتدفق إليه بدلاً من المتحكم الرئيسي، فيجب حماية المتحكم من التهديدات الخارجية أو الفيزيائية، ويجب ضمان أمن نظام التشغيل بأن لا يكون له باب خلفي أو أن لايترك بوابات أو خدمات أو حتى بروتوكولات مفتوحة.

أيضاً يتحدث المؤلف عن الهجوم على المعطيات المتدفقة في الشبكة، حيث يطرح مسألة حماية المعطيات عن طريق التشفير وعن طريق استخدام آليات للتحقق وإعطاء الصلاحيات وذلك لتجنّب الهجوم side channel، ويؤكد أيضاً على ضرورة ضمان أمن جميع مكونات الشبكة لأنَّ استغلال نقطة ضعف أياً منها، قد يؤدي إلى الوصول إلى الأجزاء الرئيسية من الشبكة مثل المتحكم، ويقترح أن تتم حمايتها فيزيائياً، مع استخدام IPS و IDS و Firewall.

أيضاً يتحدث المؤلف عن الهجوم على طبقة التطبيقات، فالهجوم على Southbound API قد يؤدي إلى قطع الخدمة في سائر الشبكة، فيجب تأمين حماية لهذه التطبيقات وحماية جميع الاتصالات التي تتم بين جميع التطبيقات عن طريق استخدام TLS، واستخدام ترامیز آمنة وذلك عن طريق وجود توقيع رقمي خاص بها.

في ورقة البحث [8] يقترح المؤلفون حماية الاتصالات بين المبدل والمتحكم باستخدام TLS أيضاً مع التحقق المتبادل لشهادتي الطرفين من خلال مفتاح خاص يولَّد من جهة ثالثة موثوقة (root certificate)، و يتحدث أيضاً عن أنَّ هنالك العديد من الشركات تتجنّب مسألة تضمين TLS في شبكتها نظراً لتكلفتها وصعوبتها، فبدايةً يجب توليد شهادة المتحكم، ثم شهادة المبدّل، ثم توقيع الشهادات من طرف ثالث عن طريق مفتاح خاص، ومن ثمَّ أخيراً إدخال المفاتيح الصحيحة والشهادات إلى جميع الأجهزة في الشبكة، إذاً وفي حال غياب تقانة TLS لا يوجد طريقة ليتحقّق المتحكم فيها من المبدّل. يوجد مثلاً نموذج Flow Visor والذي يمثل proxy بين المتحكم والمبدّل، حيث يقوم هذا النموذج بتقسيم الشبكات إلى شرائح، حيث يكون كل متحكم مسؤول عن جزء من الشبكة، لذلك إن قام مهاجم بإصدار الأمر بتجاهل جميع طرود UDP فإنه وبنموذج Flow visor فقط جزء من الشبكة سوف يتأثر ولن تتأثر الأجزاء الأخرى لأن flow visor سوف يعيد كتابة القاعدة بناءً على شرائح الشبكة.

أيضاً يقترح المؤلفون وجود checksum لكل القواعد الموجودة ضمن جدول التوجيه في المبدّلات يُرسَل بشكل دوري إلى المتحكم، الذي يقوم بحسابها في كل مرة ليعرف ما إذا تمّ تبديل القواعد أو تمّ التلاعب بها.

يوجد العديد من التصاميم للمتحكمات الهادفة إلى تجاوز نقاط الضعف وتأمين هذه المتحكمات، ومنها المتحكمات FlowDayLight والتي لاقت العديد من نقاط الضعف نسرد بعضها: **1**- المتحكم يأمر المبدّل بإرسال الطرود multicast إليه في كل مرة، دون أن يتم تسجيل قاعدة التسيير لطرود من هذا النوع في جدول توجيه المبدّل. **2**- يتم ترحيل جميع الطرود الحاملة لعنوان فيزيائي غير معروف إلى المتحكم. **3**- لا يوجد آلية حماية ضد MAC Spoofing، لذلك يستطيع المهاجم ملء جدول التوجيه لدى المبدل بعدد غير محدود من الطرود التي تحمل عناوين فيزيائية عشوائية. أيضاً يستطيع ملء ذواكر المتحكمات بمثل هذه الطرود العشوائية، لذلك يجب أن يوجد آليات لمنع MAC Spoofing ووضع محددات لعدد الطرود التي تحمل عناوين غير معروفة ووضع قواعد من أجل مسألة الـ multicast.

4.3 نقاط ضعف طبقة التطبيقات في الشبكات المعرّفة برمجياً والتحسينات المضافة إليها:

في[39]، يتحدّث المؤلفون عن مدى خطورة نقاط الضعف الموجودة في هذه الشبكات، وعن كيفية استغلال اتصالات التطبيقات الجنوبية لإرسال معلومات حسّاسة في الشبكة إلى طرف ثالث. أهم المخاطر التي يحذّر منها الباحثون:أ. الولوج غير المصرّح به، ب. استدعاء التوابع بشكل غير شرعي، ت. حقن قواعد خبيثة، ث. استنزاف الموراد، والرجل في الوسط. يقترح المؤلفون متحكّماً يدعى بِـ ControllerSEPA، حيث بالاستفادة من RESTful API، يستطيع الباحثون مواجهة التطبيقات الخبيثة في الشبكات المعرّفة برمجياً. يمكن لهذا المتحكم توفير نظام تحقق AAA (Authentcation, Authorization, Accounting)، معالجة القواعد المتضاربة، عزل تطبيقات OpenFlow، التحكم الحُبيبي بالنفاذ والتشفير. يعرض الباحثون في [41] كيفية

استخدام التطبيقات في الهجوم على المتحكمات OpenDayLight، ONOS، وFloodLight، حيث قاموا بإجراء العديد من السيناريوهات لعرض الحالات التي يمكن من خلالها استغلال نقاط الضعف في طبقة التطبيقات الخاصة بهذه المتحكّمات، ويبرهنون على أنّه وباستخدام برنامج خبيث صغير، يمكن القضاء على بيئة الشبكات المعرّفة برمجياً بالكامل. بينما في [42]، يقدّم الباحثون آلية أمنية لكشف النشاطات الخبيثة في الشبكات المعرّفة برمجياً، حيث يعتمد الباحثون على استغلال مركزية التحكم وقابلة البرمجة في هذه الشبكات لإجراء مراقبة دورية لمعدّلات استخدام وحدة المعالجة المركزية لمختلف التطبيقات، وإحصائيات المعلومات المتداولة بين المضيفين، حيث يقوم الباحثون بالمقارنة بين الاستدعاءات الاعتيادية لتوابع النظام، مع الاستدعاءات التي تتم في الحالات غير الطبيعة، سواءً أوامر القراءة أو الكتابة أو التعديل أو غيرها من التوابع المستخدمة في هذه الشبكات.

في [43]، يقدّم الباحثون نظاماً جديداً يُسمّى INDAGO، الذي يقوم بعمليات تحليل إحصائية لتطبيقات الشبكات المعرّفة برمجياً كما يقوم بنمذجة سلوكها، حيث يقوم بإجراء عمليات كشف بشكل أوتوماتيكي، كما يقوم باستخدام منهجيات تعلّم الآلة لتحقيق غرض التحليل والكشف.

في[44]، يطرح الباحثون منهجية جديدة في اكتشاف التعديلات الخفية التي تقوم بها التطبيقات الخبيثة في الشبكات المعرفة برمجياً وتقوم بمنعها من هكذا محاولات قبل أن تبدأ بها، حيث أنّهم قادرون على إيقاف التطبيق الخبيث قبل أن يقوم بإجراء هجومه. يتم تحقيق ذلك من خلال مقارنة حالة الشبكة الفعلية، والتي تشمل كل من التطبيقات الضارة والحميدة، مع حالة الشبكة التي يتم توفيرها بواسطة متحكم مشبوه. تمّ استخدام نوذج أوّلي للمنهجية في بيئة بايثون، حيث تمّ استخدام مكتبة مفتوحة المصدر للتعامل مع واصفات بروتوكول التطبيقات الجنوبية.

تطرح ورقة البحث [9] نقاط ضعف ومشاكل أخرى تتعلق بطبقة التطبيقات في شبكات SDN نعرض بعضاً منها كمايلي:

1. نقاط الضعف في طبقة التطبيقات:

أ. التطبيقات المتداخلة **nested application**: وتعدّ هذه من التحديات الصعبة والخطيرة لموضوع الأمن بشبكات SDN، حيث إن كانت تعمل التطبيقات بطريقة تداخل الحلقات service chain interference، وتوقفت أحد المخدمات أو دخلت في حالة حلقة لا نهائية، فهذا يعني انقطاع كامل الحلقة وتوقفها.

ب. تمتلك التطبيقات القدرة على الوصول إلى الذاكرة الداخلية لشبكات **SDN**: حيث أنّ التطبيقات تمتلك إمكانية الوصول إلى الموارد الداخلية في الشبكة، وبالتالي فإنّ المتحكمات تتشارك الذواكر الداخلية مع التطبيقات، إذاً تستطيع التطبيقات الوصول والتلاعب بقواعد المعطيات الموجودة داخل الذواكر الداخلية لشبكات SDN.

ج. التطبيقات مسؤولة عن بعض رسائل التحكم: إنَّ رسائل التحكم مسؤولة عن تأمين الاتصال بين طبقتي التطبيقات والمعطيات، بالتالي يمكن أن يستغل أحد التطبيقات هذه الرسائل لتغيير محتوى جدول التوجيه مثلاً، وبالتالي تخريب عمل الشبكة، أو يمكن أن يقوم تطبيق خبيث بإيقاف عمل كامل الشبكة وذلك بمسح جداول التوجيه بالمبدّلات.

د. استهلاك موارد الشبكة: يمكن أن تتشارك التطبيقات الخبيثة من أجل تعطيل الشبكة كاملةً وذلك عن طريق استهلاك الذواكر أو وحدة المعالجة المركزية CPU أو يمكن أن تقوم هذه التطبيقات بتنفيذ أمر خروج من النظام لتجاهل نسخ المتحكمات، وهذا ما نهتم به في بحثنا هذا.

2. نقاط الضعف لدى التطبيقات Northbound API: تعاني هذه التطبيقات من عدم وجود معيار واحد لها جميعاً، بالتالي نحن مضطرين للعمل مع عدة أنماط، وكل منها له ما له وعليه ما عليه من نقاط القوة والضعف، وأيضاً مازالت تعاني هذه التطبيقات من ضعف التصميم حيث يمكن استغلال هذه التطبيقات بسهولة للتلاعب بمحتويات الشبكة.

3. نقاط الضعف بطبقة التحكم حيث لدينا هنا عدة نقاط ضعف، منها:

أ. مشكلة الطرد الوارد **packet-in**: يسمى الطرد الوارد والذي لا يوجد قاعدة خاصة به لتوجيهه ضمن المبدّل يسمى بِ packet-in، بالتالي وحسب بروتوكول Open Flow فإنه يتم توجيه هذا الطرد إلى المتحكم، وعلى اعتبار أنه لا يوجد مبدّلات موثوقة فيمكن أن يستغلّ أحد المهاجمين نقطة الضعف هذه لإرسال الكثير من الطرود packet-in وإسقاط المتحكم. يمكن أيضاً باستخدام طرود مزيفة أن يتم تسميم نظرة المتحكم للشبكة وذلك بإرسال طرود ARP عشوائية لعناوين فيزيائية غير موجودة. يمكن أيضاً تسميم اكتشاف طوبولوجيا الشبكة وذلك عن طريق التلاعب بخدمة اكتشاف الخط.

ب. التضاربات في إعدادات المتحكمات: حيث يوجد لدينا ضمن المجال الواحد عدة متحكمات وعدة بُنى open flow بالتالي من الممكن أن لا يكون هنالك تزامن بعمل هذه المكونات مع بعضها.

4. نقاط الضعف في واجهة التطبيقات Southbound API: حيث يمكن تنفيذ هجمات الرجل في الوسط هنا لتبديل محتوى رسائل التحكم بين طبقة التحكم وطبقة المعطيات وتخريب محتوى جدول التوجيه، ويمكن أن تكون عرضة للتنصت سواءً الفعّال أو غير الفعّال وذلك عن طريق التجسس على قناة التحكم حيث يمكن للمتحكم تعلُّم طوبولوجيا الشبكة من خلال التجسس على رسائل التحكم. يمكن أيضاً أن تكون عرضة

لهجمات مستوى الـ TCP حيث لا يؤمّن استخدام TLS حماية لمستوى TCP. أيضاً هنا لايوجد معيار موحّد لهذه الواجهات.

5. نقاط ضعف طبقة المعطيات: واحدة من أهم المشاكل التي تعاني منها هي عدم معرفة الأصل الحقيقي للمعطيات أو جدول التوجيه. أيضاً، تعدُّ الطاقة الاستيعابية المحدودة للمبدّلات من المسائل الهامة، حيث لا يمكن للمبدّل تحمل عدد كبير من جداول الدفق، بالتالي يمكن القضاء على المبدّل من خلال مهاجم مشبع بالموارد.

تتناول ورقة البحث [10،46] أخطار أخرى في شبكات SDN مثل الانتحال spoofing، وهجمات أخرى نستعرضها كما يلي:

1. ARP Spoofing: حيث تكمن الخطورة هنا، أنْ يدعي مثلاً أحد المبدّلات الخبيثة أنه صاحب العنوان المنطقي الموثوق، فيضع عنوانه الفيزيائي بدلاً من العنوان الفيزيائي الخاص بالمبدل الموثوق، بالتالي يتم تحويل كل الطرود إليه. يقترح المؤلف Matias وآخرون نموذج يقوم فيه المتحكم في كل مرة بالتحقق من جدول المقابلة بين العنوان المنطقي والعنوان الفيزيائي للأجهزة الموثوقة وذلك لكشف عمليات spoofing، وفي حال اكتشف المتحكم أنَّ هنالك عنوان فيزيائي لجهاز غير موثوق يقوم بتجاهله. في بروتوكول open flow يمكن إجراء هجوم ARP spoofing في حال لم نكن نستخدم SSL.

2. IP Spoofing: من المعروف أن أحد أهم المخاطر في الشبكات هو انتحال العناوين المنطقية، لذلك أحد النماذج المستخدمة في شبكات SDN والمطبقة في بروتوكول Open Flow، هو نموذج VAVE (Virtual source Address Validation Edge) والمضمَّن في المتحكمات، حيث يقوم المتحكم بالتحقق من العناوين الخاصة بالطرود الخارجية والغير مسجَّلة ضمن جدول التوجيه، حيث يكون لكل جدول توجيه قائمة بيضاء (موثوقة) لمقارنة العناوين مع قواعد معينة إما للتجاهل أو للمعالجة أو للتسيير. ينبغي على متحكم شبكة SDN

أن يكون قادراً على عزل شبكته المحلية عن الشبكة الخارجية كما في حال شبكات NAT (Network Address Translation) حيث يمكن أن يكون لدى المتحكم جدول لترجمة العناوين الخارجية إلى عناوين داخلية.

3. البعثرة Tamparing: وهو التعديل الغير شرعي لمعلومات الشبكة لتدميرها وتخريبها، مثل الطوبولوجيا وجداول التوجيه، حيث يمكن أن يقوم المهاجم بتعديل جدول التوجيه أو قواعد جدار الحماية Firewall من أجل السماح لبعض الأشخاص الغير مصرّح لهم بدخول الشبكة. لتفادي مشكلة البعثرة، يجب أن يقوم المتحكم بفحص دوري لطرق التشفير والاتصالات. إنّ المشكلة الأساسية باستخدام TLS هي أنه إجراء اختياري في بروتوكول open flow وأنّ الكثير من الشركات التي تقوم بصنع المتحكمات لا تدعم TLS.

4. اختلاس المعلومات:

الهدف من هكذا هجمات هو ليس التخريب أو الإساءة إلى معلومات الشبكة، إنّما تهدف إلى سرقة المعلومات بالدرجة الأولى والاستفادة منها. من الطرق المتبَّعة لمواجهة هذا النوع من الهجمات هو استخدام مضادات لأدوات المسح scanning والتي من المعروف أنها تُستخدم عند الخطوة الأولى في هكذا هجوم، وهنالك العديد من التنجيزات المستخدمة في بروتوكول Open Flow لمواجهة أدوات المسح هذه. أيضاً من الممكن استخدام القوائم البيضاء والسوداء لترشيح التدفقات الشبكية، حيث تعتمد الشبكات التقليدية على العناوين المنطقية والفيزيائية للفلترة ويمكن تعميمها في بنية Open Flow لإنشاء قوائم بيضاء وسوداء في جداول التوجيه بناءً على تلك العناوين.

5. قطع الخدمة DoS:

هنالك العديد من الاقتراحات والنماذج التي تُعنى بالحماية من هجوم قطع الخدمة، فمنها من يعتمد على مقارنة وقياس دوري لحجم الطرود الواردة، ووضع عتبة معينة إن تمَّ تجاوزها يتم اتخاذ القرار بأن هنالك هجوم قطع خدمة

وشيك. أيضاً منها من يستخدم نماذج ثابتة للمقارنة معها، أي نماذج لهجمات سابقة، فيعلم المتحكم من شكل التدفق أن هنالك هجوم قطع خدمة قادم.[45]

يقترح المؤلف هنا استخدام Firewall لكن بسمات مختلفة قليلاً عن تلك الموجودة في الشبكات التقليدية حيث أن المتحكم هنا هو المسؤول الرئيسي عن تسيير الطرود.

5.3 مناقشة سريعة للحلول المقدّمة في الأبحاث السابقة:

- بدايةً بالنسبة للبحث[6]، يقترح المؤلفون **7** أنواع للتهديدات الممكنة في شبكات SDN ولكل تهديد يقترحون حلّاً ممكناً، **أولاً:** بالنسبة للتهديد الأول الخاص بالطرود الزائفة، تحدّث المؤلفون عن إمكانية استخدام أنظمة تحليل وأنظمة لمعرفة المسبب الحقيقي للحدث، ولكن أرى أنّ هذا الحل غير ناجح إلا في الشبكات الصغيرة والمحلية، أما في الشبكات الكبيرة جداً، ففي حال استطاع المهاجم الوصول لعدة أجهزة متناثرة في الشبكة واستغلالها لإرسال طرود وهمية وزائفة، فإنّ هذه الأنظمة لن تجدي نفعاً في معرفة مسبّب الهجوم، ونحن نعلم وجود العديد من التطبيقات والفيروسات التي تستطيع تنفيذ عملية remote access. **ثانياً:** لم يفنّد المؤلفون في التهديد الثاني ماهي نقاط الضعف التي يرون أن استخدام أنظمة لإدارة الثقة بين أجهزة الشبكة قد تفيد في تفاديها، لكن أقترح أن يتم وضع المبدّل switch في أماكن بعيدة المنال عن أيدي المخربين، وأن لا يُسمح لأي شخص بالوصول إليهم إلا بعد أخذه لتصريح من قبل ثلاثة أشخاص من نفس فريق العمل، وأن يكون عدد الواجهات interfaces محدود بحسب الحاجة وبحسب الدراسة المفضية إلى تركيب هذا المبدل، وفي حال أردنا التوسع بالشبكة يتم تركيب واجهات حسب الحاجة فقط حتى لا يَستغل أحد وجود واجهة فارغة ويوصل جهازه فيها ليقوم بالتخريب. **ثالثاً:** مهاجمة اتصالات طبقة التحكم، يتحدث المؤلفون عن إمكانية استخدام تقنيات تستدعي أن يحصل المبدل فيها على درجة من الوثوقية ثم بعد ذلك يقوم بالعمل والتواصل مع طبقة التحكم، أرى أن هذا

سوف يفضي إلى بطئ في عملية التسيير. **رابعاً:** في حال استخدمنا طريقة النسخ، فإنه أرى أنه عند نجاح المهاجم باختراق المتحكم الأول، فسوف يعي تماماً ما هي نقاط ضعف المتحكمات الأخرى المثيلة. يمكننا أيضاً هنا كحل إضافي عدم السماح لأحد للولوج إلى المتحكم إلا بتصريح من قبل عدة اختصاصيين من نفس المجال. **خامساً:** عدم وجود آلية لضمان الثقة بين التطبيق والمتحكم، لذلك يمكن إنشاء مؤسسة ومنظمة تقوم بإعطاء شهادات دورية للتطبيقات التي تلج إلى المتحكمات، ويتم تحديث هذه الشهادات بشكل دائم ومتجدد، حيث يقوم المتحكم بدايةً بالتأكد من صحة شهادة التطبيق ثم بعد ذلك يستجيب لأوامره. **سادساً:** أرى أنه من الممكن أن يكون هنالك اتفاق بين هؤلاء المصرّحين والمصرَّح لهم، لذلك من الأفضل أن يتم وضع أناس لا يعرفون بعضهم البعض، ويتم اختيارهم في كل مرة بشكل عشوائي للتصريح لبعضهم وأن يكون هنالك ملف log لتسجيل حركة الولوج إلى المتحكمات ومن هم الأشخاص الذين قاموا بإعطاء التصريحات. **سابعاً:** يقترح المؤلف من أجل التهديد السابع وجود نظام تعقب و تسجيل!!! لكن في حال وقوع الشبكة بالكامل، ربما لن يفيد التعقّب والتسجيل، لذلك أرى أنّه يجب أن يكون هنالك نظام مراقبة عام لدى المحطة الأساسية لمراقبة كل ما يجري من الشبكة وأنظمة تنبؤ بالأفعال الغير اعتيادية، وأن يكون هنالك قواعد معطيات تحتوي على أنماط هجمات معروفة وتقليدية لمقارنة سلوك الشبكة معها بشكل دائم لمعرفة المشكلة قبل وقوعها.

بالنسبة لطرحهم الخاص ببناء منصة تحكم آمنة وموثوقة، فإن اقتراحهم بخصوص الربط الديناميكي للمبدّل مع المتحكمات الأخرى، فمن وجهة نظرنا أنه عند استغلال أحد المبدّلات للوصول إلى المتحكم المرتبط معها وتعطيله، سوف ينتقل المبدل إلى متحكم آخر وبالتالي إصابة المتحكم الجديد بعطل (إلا في حال كان المتحكم الآخر ليس نسخة عن المتحكم المصاب)، بالتالي نحن بحاجة إلى تأمين المبدّلات أولاً بعد ذلك نستطيع استخدام هذا الاقتراح، وينبغي أن يكون هنالك آلية موثوقة للتعريف عن المبدّلات لدى المتحكمات قبل ارتباطهم، أو أن يكون هنالك إجرائية فحص للأمان يقوم بها المتحكم لكل مبدل يرغب بالارتباط معه.

بالنسبة للحل الذي ينص على بناء الثقة بين المتحكم والأجهزة: ما هي الآليات التي يمكن استخدامها لمثل هكذا أمر؟؟ وما مدى تأثيرها على جودة وسرعة الشبكة على اعتبار أن كل جهاز جديد بحاجة للدخول للشبكة سوف يكون هنالك حاجة إلى التحقق منه ونسبه إلى القائمة البيضاء أو وضعه بطور التجميد أو حتى رميه خارجاً.

الحل الآخر الذي تمّ اقتراحه هو إعطاء الثقة لكامل المبدّلات ثم سؤال أحد المبدّلات عن الوثوقية الخاصة بها في حال كان مشكوك بها، لكن في هذه الحالة يكون المهاجم لربما نفّذ هجمته ثم لم يعد يهتم بعد ذلك لا لإعطاء دليل وثوقية ولا حتى للعودة للشبكة من جديد.

- أما في ورقة البحث [8] فعندما اقترح المؤلف أن يكون هنالك checksum لجدول التوجيه. أرى أن الـ checksum يمكن تبديله بسهولة حيث حتى لو تم التلاعب بالقواعد، فيستطيع المهاجم أن يحافظ على نفس checksum لذلك أقترح أن يتم إضافة عملية التهشير hash function لضمان عدم التلاعب هنا. هنالك العديد من النماذج المطروحة لبناء المتحكمات، وأيضاً عند اقتراحه العمل ببروتوكول TLS فإن هذا سوف يعرض الاتصالات لمشكلة هجوم الرجل في الوسط man-in-the-middle .
- أما في ورقة البحث [9] يطرح المؤلفون عدة نقاط ضعف جديدة لكن دون تحديد الأساليب الممكنة لمواجهتها لكن سوف أناقش بعضاً منها. بالنسبة للتطبيقات المتداخلة، برأيي يجب أن يتم كسر أي حلقة وجعل التطبيقات تعمل بشكل منفصل دائماً، أو أن يكون لكل تطبيق وقت محدد ويتم توزيع الموارد بشكل متساوي بين التطبيقات في كل حيز زمني slot time. بالنسبة لقدرة التطبيقات على الوصول إلى الذاكرة الداخلية، فيجب أن يكون هنالك إجرائية تسمح بداية بالتحقق من وثوقية هذا التطبيق ثم بعد ذلك يعُطى الصلاحية للنفاذ إلى الذاكرة. مشكلة الطرد packet-in أقترح أن يكون هنالك أنظمة مراقبة دائمة مع أنظمة معرفة السبب الحقيقي وذلك لتبيان المنبع الذي تصدر منه هذه الطرود، وأيضاً إجرائيات لوضع عتبة معينة لعدد الطرود الواردة والتي

ليس لها قاعدة ضمن جدول التوجيه، وأن يكون المبدّل متصل مع أكثر من متحكم، فلا يرسل هذه الطرود في كل مرة إلى نفس المتحكم، إنما يقوم بعملية موازنة للحمل load balancing. أما نقاط الضعف الخاصة بطبقة المعطيات، فأقترح أن يكون هنالك إجرائية تسمح بوسم كل مبدّل برقم تسلسلي ما، ويكون هذا الرقم مشفراً بطريقة التهشير، حيث في كل مرة يتأكد المتحكم من صاحب هذا الدفق، ثم بعد ذلك يعالجه.

6.3 الخلاصة:

يوجد الكثير من الأبحاث التي بدأت بالتوجه نحو مسألتي الأمن والوثوقية في شبكات SDN. يجب أن تؤخذ هذه الدراسات بعين الاعتبار، سواءً من قِبل الشركات المصنعة أو الخبراء والباحثين المعنيين بإدارة شبكات SDN، ورأينا الكثير من الحلول المقترحة كما في[6]، حيث فنَد المؤلفون آليات بالاسم لمواجهة مشكلات محددة ومعينة، واقترحوا مسألة النسخ المتكرر للمتحكمات، ومسألة التنوع وهذه نقطة جيدة هامة، تُفضي مع مسألة النسخ إلى نظام أكثر متانة وقوة.

- عندما يتمدد العقل لاستيعاب فكرة جديدة لا يعود أبداً إلى حجمه الطبيعي (أوليفر وندل هولمز).

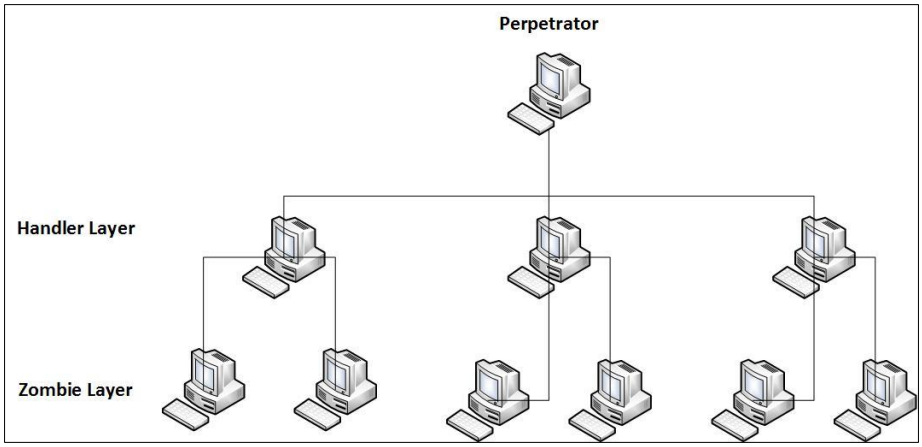
1.4 مقدمة:

تُعتبَر هجمات منع الخدمة الموزّع DDoS محاولة خبيثة لاستنزاف موارد حاسب أو شبكة من الحواسب عن طريق إرسال أعداد ضخمة من الطرود، يمكن للمهاجم زرع العديد من التطبيقات الخبيثة في الشبكة للقيام بمثل هذه الهجمات، حيث يكون لدى المهاجمين هدفين أساسيين:

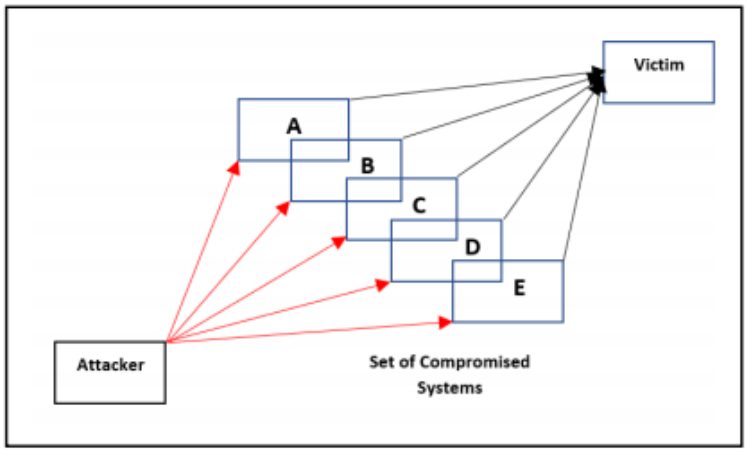
أ. استخدام كامل عرض الحزمة Bandwidth.

ب. استنزاف الموارد [12].

يبدأ الهجوم من خلال زرع المهاجم لرماز برمجي code خبيث في أجهزة الحاسوب والتي يشار إليها باسم Botnet. عند بدء الهجوم، يتم تشغيل هذه الرمازات أو البرمجيات، ويتم توجيه سيل من الطرود نحو الضحية. يوجد نوع معقّد أكثر لهذه الهجمات، حيث يقوم المهاجم باستخدام عدد من الحواسب المصابة بالفايروس، والتي يتم تسميتها handler، للتحكم بعدد ضخم من الحواسب التي يتم تسميتها بـ Zombie. تعدّ هذه الحواسب المضيفة Zombie مسوؤلة عن توليد حركة مرور المهاجم [12]. باستخدام الـ botnets، يكون المهاجم أكثر تركيزاً ويحافظ على نفسه متخفياً عن برمجة الكشف. إنّ هجوم منع الخدمة الموزع DDoS يعدّ الأكثر شهرةً في مجال التحميل الزائد للشبكة أو قطع الخدمة في الشبكة. يعرض لنا الشكل(4-1) مثال بسيط عن مكونات هجوم منع الخدمة، و يعرض الشكل(4-2) آلية مبسطة لهجوم منع الخدمة.



الشكل(4-1). مكوّنات هجوم منع الخدمة.



الشكل(4-2). آلية مبسّطة لهجوم منع الخدمة.

2.4 أنواع الهجمات DDoS:

يتطلّب البدء بأية هجوم النفاذ إلى الشبكات الفرعية للضحايا لاستخدامها كـ zombies. يقوم المهاجم بإجراء عمليات مسح لإيجاد الحواسب الضعيفة في الشبكة، وتكون عملية المسح هذه عشوائية أو بالاعتماد على قائمة معينة، أو عملية مسح للشبكة المحلية الفرعية، أو من خلال خوارزمية قام بتصميمها القرصان نفسه [13].

يمكن تصنيف الهجمات بناءً على التطبيق، المضيف، الموارد، الشبكة، وهجمات البنية التحتية [13]:

1. التطبيق: يتم استهداف التطبيق في المضيف لمنع استخدامه.

2. المضيف: يتم جعله غير متاحاً.

3. الموارد: يتم من خلال سحق المخدّم وإشغاله بمعالجة دفق ضخم من الطلبات المزيّفة.

4. الشبكة: يتم استنزاف عرض حزمة الشبكة من خلال إرسال أحجام ضخمة من المعطيات إلى الشبكة.

5. البنية التحتية: يتم استهداف مخدم DNS (Domain Name Server) بشكل متزامن من عدة أماكن.

ظهرت العديد من هجمات منع الخدمة في الآونة الأخيرة، ويظهر النمط المتّبع في هذه الهجمات أنواع معينة من الطرق للبدء بالهجمات [14-15] نذكر منها:

أ. فيضان طرود UDP: وهو نوع من الهجمات حيث يقوم المهاجم بإرسال عدد ضخم من الطرود إلى بوابات عشوائية لدى جهاز الضحية مما يستدعي قيام الضحية بالبحث عن التطبيقات الموافقة لهذه البوابات. يقوم جهاز الضحية بإرسال رسالة الهدف غير متاح Destination Unreachable كإجابة على الطرود الواردة. كلّما ازداد عدد الطرود الواردة، كلما ازداد زمن تأخير الاستجابة وبالتالي سيصبح الجهاز غير متاحاً.

ب. فيضان طرود التزامن SYN: يقوم المهاجم بتأسيس اتصال TCP لاستهداف جهاز الضحية. يتم إرسال عدة طرود SYN إلى الضحية لكن دونما إرسال طرود ACK للرد على الضحية مما يسبب إشغال موارد الجهاز الضحية وجعله يستخدم حجوم ذاكرة إضافية وهو ينتظر الطرود ACK من جهاز المهاجم.

جـ. هجوم عكس الـ DNS: يقوم المهاجم بتزييف العنوان المنطقي الخاص بمخدم الـ DNS، ووضع العنوان المنطقي الخاص بالضحية، بالتالي توجيه طلبات DNS إلى الضحية، وبالتأكيد لن يستطيع جهاز الضحية الصمود طويلاً أما العدد الضخم لتلك الطلبات.

د. فيضان طرود ICMP (Internet Control Management Protocol): وهو نوع آخر من الهجمات التي تستنزف موارد الضحية من خلال إرسال عدد ضخم من طرود ping (echo request)، والتي تقيّد المخدم وتجعله بحالة إرسال مستمرة لطرود (echo replies).

إنّ المعامل المشترك لدى جميع الهجمات السابقة هو ضخ عدد كبير جداً من الطرود في شبكة الضحية واستنزاف الموارد.

3.4 الكشف عن السلوك غير الطبيعي لحركة البيانات لتخفيف هجوم منع الخدمة:

إنّ المعامل المشترك بين أنواع هجمات منع الخدمة المختلفة هو العدد غير الطبيعي للطرود المرسلة إلى الضحية. في الحالة الطبيعية، يكون هنالك نمط معين لنشاط الشبكة ومعدل استهلاك مقبول لعرض الحزمة. في حال حدوث دفق فجائي لحركة المرور، تأخير زمني، ومعدل استخدام غير طبيعي لوحدة المعالجة المركزية CPU (Central Processing Unit)، أو حدوث هبوط في أداء الشبكة، فإنّ ذلك يعتبر أمراً غير طبيعياً. إنّ أية إجرائية أو خوارزمية مضادة لهجمات منع الخدمة، سوف تبحث عن هذه الأحداث الغير طبيعية [16]. قد يتم توجيه الهجوم نحو طبقة الشبكة ممايسبّب الاختناقات ضمنها، أو إلى طبقة التطبيقات مما يسبب استنزاف موارد وحدة المعالجة المركزية. إنّ عملية إدراك نوع المعطيات وسماتها في الشبكة تعدّ الخطوة الأولى في سبيل اكتشاف الحركات الغير طبيعية للبيانات. إنّ هذه السمات قد تكون، معلومات ترويسة الطرد، التأخير الزمني، حجم الطرد، البروتوكول..إلخ.

4.4 أهم الآليات المُستخدمة في اكتشاف انحراف الشبكة عن الوضع الطبيعي:

في الشبكات التقليدية (IP networks)، يوجد محدودية في عرض الحزمة، محدودية في القدرة على المعالجة وحمل المعطيات. عندما تخضع بعض واصفات الشبكة لعمليات تحليل إحصائي، فإنّ لكل واصفة نمط محدّد سوف يظهر. كلما كان زمن ظهور النمط أطول، كلما كان هذا النمط أكثر متانة.

إن جمع المعطيات، الترشيح وعمليات المعالجة لكشف السلوك الغير طبيعي للشبكة تم استخدامهم من قبل العديد من التقنيات. إن التحليل الإحصائي وتعلّم الآلة هما أهم الطرق المستخدمة في كشف السلوك الغير طبيعي للشبكة.

1.4.4 التحليل الإحصائي:

تمَّ اقتراح آليات مثل حساب الأنتروبية وآلية Chi-Square، لكشف التغيرات في حركة بيانات الشبكة. تَستخدم الأنتروبية ترويسات الطرود كرموز معلومات مستقلة مع احتمال فريد لوقع الحدث. إنها طريقة شائعة لكشف هجوم منع الخدمة [18-17-19-20]. باختيار نافذة زمنية بطول معين لنقل الطرود، وتحريك نافذة التسيير، سوف يظهر نمط مع احتمالات محدّدة لكل نوع من ترويسات الطرود. بالتالي تؤدي التغييرات الجذرية في هذه الترويسات عن النمط المعين إلى نشوء إنذار في النظام يشير إلى وجود حركة غير طبيعية في الشبكة. في حال تمّ توقّع نمط معيّن من أنماط التسلل، وتمت معرفة نوع ترويسة الطرد، فإنّ نموذج Chi-Square يعدّ الأفضل للاستخدام.

1.1.4.4 خوارزمية Chi-Square :

أُستخدمت هذه الخوارزمية بدايةً كأحد أهم الاختبارات التي تخضع لها السلاسل العشوائية وذلك لمعرفة جودة العشوائية ضمنها، وتُعتبر هذه الخوارزمية أفضل خوارزمية من حيث إجراء الاختبارات الإحصائية على السلاسل العشوائية. [21]

يُعرّف توزيع Chi-Square والذي يُرمز له بالرمز χ^2 علىأنه توزيع مجموع مربعات عدة متحولات عشوائية، ويُعطى بالعلاقة (1-4) وذلك من أجل $z_1^2, \dots \dots, z_k^2$ متحول عشوائي:

$$\chi_k^2 = \sum_{i=0}^{k} Z_i^2 \qquad (1\text{-}4)$$

حيث يُمثل k عدد درجات الحرية والذي يُعد من أهم المعاملات في هذا التوزيع فهو يلعب دوراً كبيراً في تحديد خصائص هذا التوزيع، ويُعرّف بأنه عدد المتحولات العشوائية جميعها ناقص واحد : $k = a - 1$ حيث a هو عدد المتحولات العشوائية التي يجري اختبارها.

وفيما يلي نبين أهم الخصائص لتوزع χ^2:

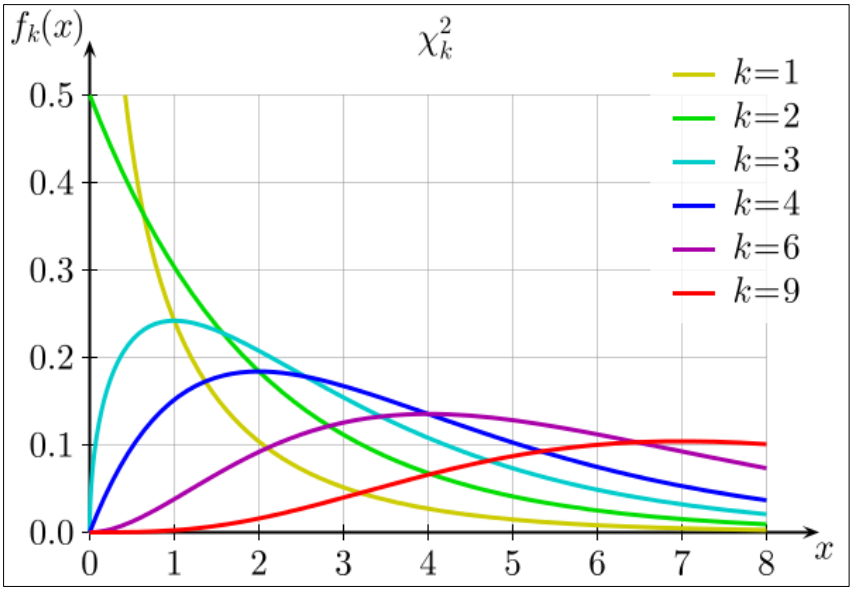
- ✓ تابع الكثافة الاحتمالي PDF (Probability Density Function) :

يقوم هذا التابع بإعطاء توصيف تقريبي لاحتمال أن يأخذ متحول عشوائي قيمة معينة ما. توضح لنا العلاقة (2-4) الصيغة الرياضية لتابع الكثافة الاحتمالية:

$$f(x;k) = \frac{x^{\left(\frac{k}{2}\right)-1} e^{\frac{-x}{2}}}{2^{\frac{k}{2}}\,\Gamma\left(\frac{k}{2}\right)} \; ; x \geq 0 \qquad (2\text{-}4)$$

حيث تمثل x قيم توزع χ^2 الناتجة من العلاقة (1-4)، k عدد درجات الحرية. أما التابع $\Gamma(.)$فهو تابع غاما غير التام: $\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t}.dt$.

يرتبط يوضح الشكل(3-4) تابع الكثافة الاحتمالية من أجل قيم مختلفة لعدد درجات الحرية k.



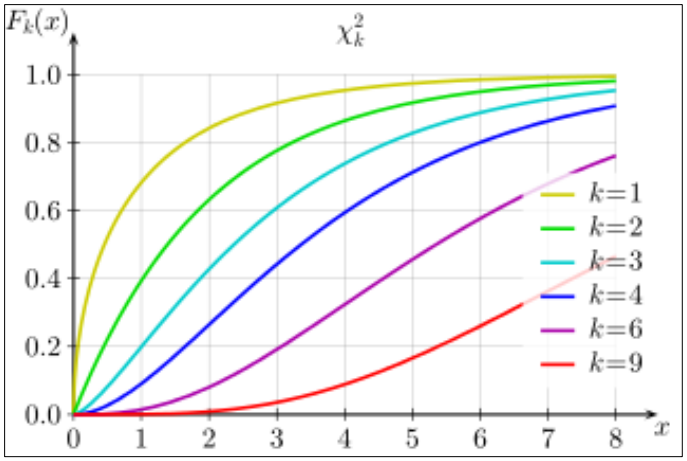
الشكل (3-4) : تابع الكثافة الاحتمالي PDF وتغيره مع درجة الحرية (k).

✓ تابع التوزيع التراكمي CDF (Cumulative Density Function) :

يحدّد هذا التابع احتمال أن تكون القيمة الحقيقية لمتحول عشوائي ما تقع تحت قيمة x، توضح لنا العلاقة (3-4) الصيغة الرياضية لتابع التوزيع التراكمي والذي من خلاله نستطيع حساب قيمة p-value التي سوف نأتي على ذكرها لاحقاً.

$$f(x;k) = \frac{\gamma(\frac{k}{2}, \frac{x}{2})}{\Gamma(\frac{k}{2})} \qquad (3\text{-}4)$$

حيث x هو قيمة توزع χ^2 و $\gamma(.)$ هو تابع غاما غير التام السفلي: $\gamma(s,z) = \int_0^z t^{s-1}e^{-t}.dt$



الشكل (4-4) : تابع التوزيع التراكمي CDF وتغيره مع درجة الحرية (k).

نلاحظ من العلاقة (3-4) اعتماد تابع التوزيع التراكمي على عدد درجات الحرية، ويتضح ذلك بشكل أفضل من خلال الشكل (4-4).

- ✓ المتوسط، التشتت، الانزياح (Skewness):

يُعرّف المتوسط لتوزيع χ^2 بعدد درجات الحرية k، أما التشتت يساوي 2k، أي أن العلاقة طردية بين درجات الحرية وتشتت التوزع، فكلما ازدادت درجات الحرية يزداد التشتت. الانزياح هو ميل منحني باتجاه اليمين أو اليسار، فإذا كانت قيمته سالبة فهذا يعني أن الذيل الأيسر لمنحني تابع الكثافة الاحتمالي سوف يكون أطول وأكثر تسطحاً من الذيل اليميني والعكس صحيح. في توزيع χ^2 يُعطى الانزياح بالعلاقة $\sqrt{\frac{8}{k}}$ أي كلما ازدادت درجات الحرية فإن قيمة الالتواء في تابع الكثافة الاحتمالية سوف تنقص وبالتالي سوف يكون الجانب الأيمن من منحني تابع الكثافة الاحتمالي أطول من الأيسر وأكثر تسطحاً منه.

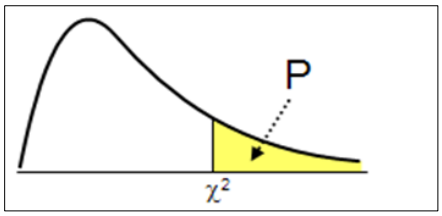
من أجل عدد درجات حرية كبير نلاحظ أن التشتت سوف يكون كبير، بينما الانزياح سوف ينعدم تقريباً، وهذا واضح من الشكل (3-4)، فعند زيادة عدد درجات الحرية يميل تابع الكثافة الاحتمالي لأن يكون متناظر حول المتوسط (عدد درجات الحرية).

✓ القيمة p-value :

تُعرّف قيمة p-value على أنها احتمال وقوع نتيجة اختبار عينة ما في منطقة الذيل من منحني توزيع chi-square. وبما أننا عرّفنا سابقاً تابع التوزيع التراكمي على أنه احتمال أن تكون القيمة الحقيقية لمتحول عشوائي ما تقع تحت قيمة X. إذاً بطرح قيمة CDF من الواحد نحصل على قيمة p-value والتي يمكن أن نعبّر عنها بالعلاقة التالية:

$$P = Pr(\chi^2_{k-1} \geq \chi^2_{PoV}) = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2_{PoV}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}} dx \qquad (4\text{-}4)$$

يبين الشكل (4-5) منطقة p-value من أجل قيمة χ^2 ما. كلما اقتربت نتيجة χ^2 من منطقة الذيل فهذا يعني أن السلسلة التي جرى عليها الاختبار ليست عشوائية بشكل كامل. وهذا يكافئ الحصول على قيمة للمتحول p-value قريبة من الواحد، والعكس صحيح فكلما كانت قيمة p-value قريبة من الصفر كلما دلّ ذلك على عشوائية جيدة للمعطيات المدروسة. (في حال كان اختبار χ^2 مستخدماً لقياس الفرق بيني توزيعين، فإن p-value قريبة من الصفر معناه أن التوزيعين قريبين من بعضهما).



الشكل (4-5) : منطقة p-value من أجل قيمة χ^2 ما.

نعرض في الجدول(4-1) مجموعة من القيم لتوزع χ^2 ذلك من أجل عدد درجات حرية والقيم p-value المقابلة لها.

الجدول (4-1) : قيم توزع χ^2 من أجل عدد درجات حرية مختلفة (k).

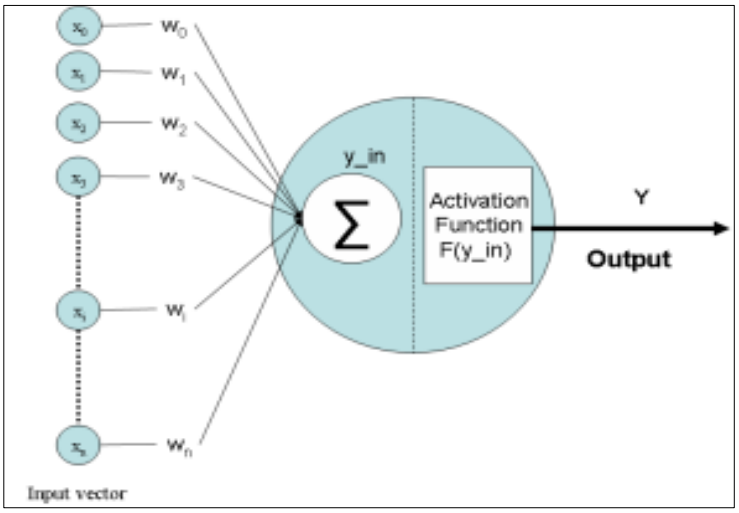
K	χ^2 value										
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.86	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
P value	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001
	Non-significant								**Significant**		

نلاحظ من الجدول السابق أن قيمة توزع Chi-Square تزداد مع ازدياد عدد درجات الحرية وأن قيمة p-value تنقص مع ازدياد قيم Chi-Square. لنفترض أن قيمة χ^2 من أجل درجة حرية df = 3 لسلسلة عشوائية ما كانت 20.3، من الجدول نجد أن احتمال أن تكون قيمة التوزع أكبر من 17.73 هي 5%، بالتالي نستطيع القول بأن هذه السلسلة عشوائية باحتمال 95%.

2.1.4.4 تعلّم الآلة:

تعدُّ هذه الآلة طريقة أخرى لتأمين الدفاع اللازم للشبكات أمام عمليات الاختراق والتسلل، فبدلاً من تثبيت خوارزمية معينة لاكتشاف السلوك الغير طبيعي للمعطيات، يتم إعداد خوارزمية متكيّفة تتدرّب باستمرار بناءً على التحديثات التي تحصل في الشبكة وعلى الأحداث التي تطرأ لتكون قادرة على مواجهة أية عملية تسلل. كمثال

على هذه الخوارزميات الشبكات العصبونية [22]. إنَّ الشبكة العصبية الصنعية (Artificial Neural Networks)هي عبارة عن نظام لمعالجة البيانات بشكل يحاكي ويشابه الطريقة التي تقوم بها الشبكات العصبية الطبيعية للإنسان أو للكائن الحي. تحتوي الشبكة العصبية (Neural Network) على عدد كبير من العناصر الصغيرة لمعالجة المعلومات تسمى الخلية العصبية أو العصبون (Neuron). إنها شبكات بعناصر حساب عصبونية عالية الوصلات الداخلية فيما بينها، ولها المقدرة على الاستجابة لإشارة الدخل والتعلم لتتلاءم مع الوسط المحيط وتعطي الخرج المناسب.يبيّن الشكل التالي (4-6) كيفية محاكاة الخلية العصبية الطبيعية:



الشكل (4-6): محاكاة الخلية العصبية الطبيعية.

حيث:

تمثل (Xn……. X3, X2, X1) شعاع إشارة الدخل.

يمثل (wn……. w3, w2, w1) شعاع وزن الشبكة.

يمثل F(y_in) تابع التنشيط Activation Function.

وتختلف الشبكات العصبونية حسب المحددات التالية:

- نموذج الوصل بين العصبونات وهذا يسمى بالبنية Architecture.

- طريقة تعيين الأوزان المرافقة للوصلات وهذا يسمى بالتدريب Training أو التعليم Learning.
- تابع التنشيط Activation function.

5.4 أثر هجوم منع الخدمة على المتحكم OpenFlow:

كما ذكرنا سابقاً فإنّ المتحكم يعدّ العصب الرئيسي في الشبكات المعرفة برمجياً، ويتم التواصل بين المتحكم والمبدّلات عن طريق قناة آمنة، بمجرد أن ينقطع الاتصال في هذه القناة، تفقد الشبكة البنية المعرّفة برمجياً، لكن ينبغي أن تكون المبدّلات قادرة على التكيف مع الوضع الجديد والعودة إلى العمل كما في الشبكات التقليدية. إذاً يمكن لهجوم منع الخدمة أن يخرّب الشبكة ككل في حال إسقاطه للمتحكم، أو يمكن له أن يملأ جدول التسيير الخاص بالمبدّل.

1.5.4 التصدي لهجوم منع الخدمة في الشبكات المعرّفة برمجياً:

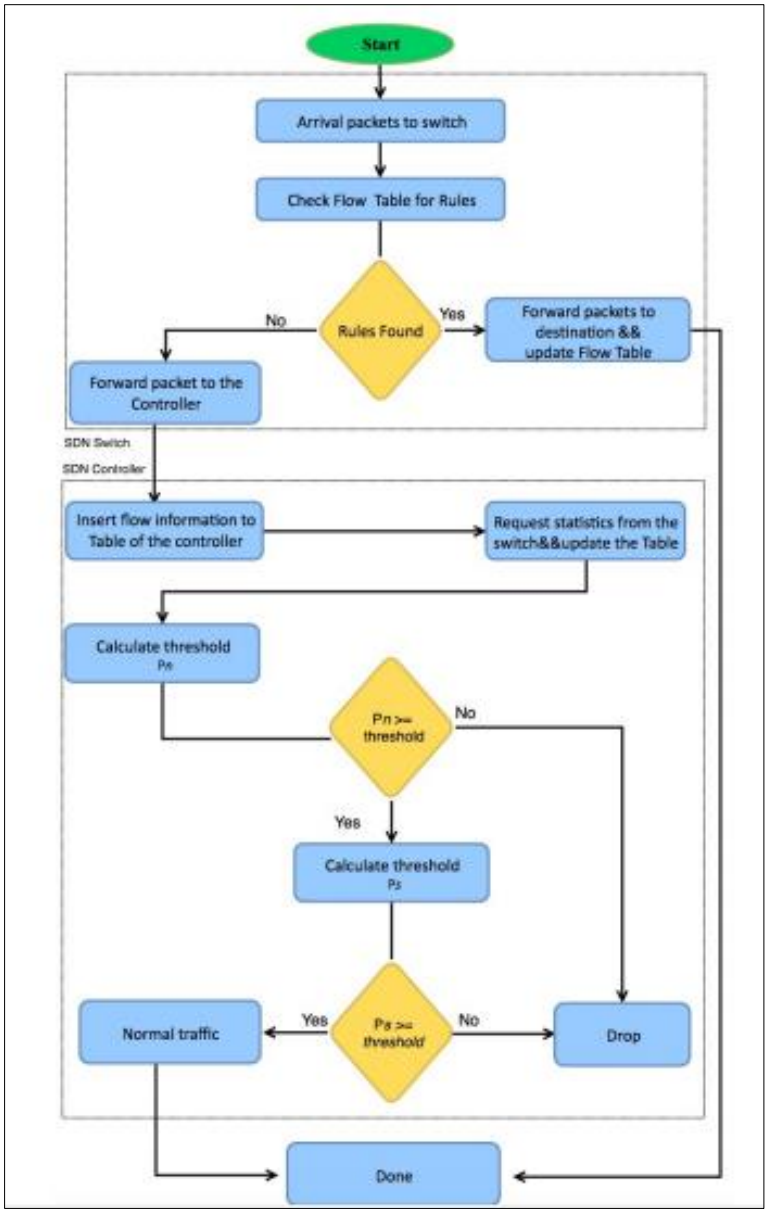
قامت الشركات المصنّعة للتجهيزات الشبكية بدراسة معمّقة وطويلة لأنواع التهديدات التي قد تصيب الشبكات التقليدية. لكن الشبكات المعرّفة برمجياً تعدّ بنية جديدة ومازالت حديثة العهد والدراسات.

يدرس الباحثون في [23] كيفية استخدام الشبكات المعرّفة برمجياً بشكل فعّال لإيقاف هجمات منع الخدمة، حيث يقترحون تطبيقاً لحجب مثل هكذا هجمات والذي يتم تشغيله على المتحكم، وبحسب الباحثين فإنهم نجحوا في تصميم بنية فعالة لحجب botnet-mounted والتي لا تُبدي أية معلومات إحصائية غير طبيعية. أيضاً في [24] تمّ تحليل عتبة الهجمات على طبقتي المعطيات والتحكم، حيث قدّم الباحثون منصة عمل مرنة وقابلة للتطوير تدعى FLOODGUARD قادرة على التصدي لمثل هكذا هجمات. تتكون المنصة من وحدتين تسميّان محلّل التدفق الاستباقي ومحرّك الطرود. تقوم الوحدة الأولى بتنجيز منهجية تشتقّ مدخلات التدفق الاستباقية في

الزمن الحقيقي من خلال تتبع التطبيقات الديناميكي. بينما يعدّ محرك الطرود مسؤولاً عن التهجير، التخزين المؤقت ومعالجة الطرود التي ليس لها جدول توجيه عن طريق استخدام عمليات جدولة محدّدة. يقترح الباحثون في [25] البنية SPHINX، يعدّ هذا المتحكم قادراً على كشف الهجمات المعروفة وغير المعروفة في طوبولوجيا الشبكة وطبقة تسيير المعطيات في الشبكات المعرفة بالبرمجيات. يقوم SPHINX بتحليل رسائل التحكم ببروتوكول OPENFLOW ليتعلّم بشكل مستمر عن سلوك الشبكة وبناء بيان للتدفق لكل حركة مرور في الشبكة.

يقدم الباحثون في [26] بنية متماسكة تعتمد على العديد من القواعد لتصنيف الطرود بشكل فعّال في الشبكة واكتشاف ما إذا كان المبدّل خبيثاً أم لا. يوضّح لنا الشكل (4-7) التالي البنية المقترحة. ينشئ الباحثون جدولاً يدعى D في المتحكم openflow ويتم استخدام هذ الجدول لتخزين العناوين المنطقية لجميع الطرود الواردة. كل عنوان منطقي له عدّاد Pn، يحمل رقم الطرد في التدفق. يحتوي الجدول D أيضاً على عدّاد Ps والذي يمثل حجم الطرود المرسلة في التدفق. بينما يدعى العدّاد الثالث بِ Du ويعرض الزمن الذي قضاه دخل التدفق في جدول التسيير. كلما ورد طرد جديد، يقوم المبدّل بمحاولة مطابقته مع أحد المدخلات التي لديه، في حال وجد تطابقاً يتم تسيير الطرد إلى البوابة المناسبة ويتم تحديث المحدّدات (العدّادات) الموافقة لهذا الدخل. في حال لم يكن هنالك تطابقاً يقوم المبدّل بإرسال رسالة Packet_in إلى المتحكم وينتظر دخلاً جديداً. عند وصول الطرد packet_in إلى المتحكم، فإنه يسأل المبدّل عن المعلومات الموافقة لهذا الطلب ويحدّث الجدول D، لذلك عند ورود مثل هكذا طلب، يفترض النظام وجود هجوم DDoS، لذلك يقوم بتعديل قيم المعاملات hard_timeout. باستخدام الجدول D والمعامل Pn (عدد الطرود الواردة من العنوان المنطقي) واستنتاج عتبة متوسط الطرود، فإن كانت قيمة Pn أقل من العتبة المحسوبة، يوقن النظام بأنه مازال تحت تأثير هجوم DDoS ولا يقوم بأية عمل. أما في حال كانت Pn أكبر أو تساوي العتبة، ينتقل النظام إلى الخطوة التالية، حيث يقوم المتحكم بإنشاء عتبة، لكن هذه

المرة بالنسبة لمتوسط البايتات وليس الطرود. في حال كان المعامل Ps أكبر أو يساوي العتبة، يقرر النظام بأن حركة المعطيات طبيعية، يوضح لنا الشكل (4-7) هذه الخوارزمية.



الشكل(4-7). المخطط التدفقي للخوارزمية المقترحة في [26].

يقترح الباحثون في [27] بنية تعاونية للتصدّي لهجمات DDoS باستخدام الشبكات المعرّفة برمجياً، حيث صمّم الباحثون بروتوكول C-to-C (Control to Control) آمن يسمح للمتحكمات في الشبكات المعرّفة برمجياً

بالاندماج في أنظمة التحكم الذاتي AS (Autonomous System) لتأمين الاتصال وتناقل معلومات الهجوم فيما بينهم. يمّكن هذا البروتوكول إنشاء آلية إعلام وإنذار فعّالة أثناء وجود هجوم مستمر وعملية ترشيح فعّالة للبيانات القريبة من مصدر الهجوم، ونتيجة لذلك توفير وقت ثمين جداً أمام موارد الشبكة. قام الباحثون بتجربة العديد من المنهجيات وطرق الربط سواءً الخطية أو المركزية أو حتى التشعبية. بينما قام الباحثون في [28] بتصميم وتنجيز نظام كشف تسلل خاص بشبكات SDN (SDN-IDS) يقوم بشكل تفاعلي باكتشاف الهجوم منذ لحظة نشوئه ويضمن العمل الطبيعي لعموم الشبكة. يقوم نظام كشف التسلل المقترح بالكشف عن العديد من هجمات منع الخدمة، وفي حال اكتشاف وجود هجوم، يقوم النظام بإبلاغ المتحكم الرئيسي. بينما قام الباحثون في [29] بتصميم بنية للكشف عن هجوم منع الخدمة في الزمن الحقيقي وذلك بالاستفادة من المحلل sFlow. أما في [30]، يقدّم الباحثون ما يسمّى بـ "درع الشبكات المعرّفة برمجياً" "SDN-Guard"، وهو عبارة عن بنية جديدة فعّالة في حماية الشبكات المعرّفة برمجياً من هجمات منع الخدمة عن طريق:1. إعادة توجيه الطرود المشبوه بها أو الخبيثة، 2. ضبط زمن انتهاء مهلة مكوث الدفق في جدول التسيير، 3. تجميع قواعد الدفق.

- الاستمتاع بالعمل يضفي عليه المثالية. (أرسطو)

1.5 مقدّمة:

في الحل المطروح، يتم قياس عشوائية الطرود الواردة، حيث تعدّ الأنتروبية مقياساً جيّداً للعشوائية [11]. تقيس الأنتروبية احتمال وقوع حدث مع الأخذ بعين الاعتبار للعدد الكلّي للأحداث. مثال على ذلك، في شبكة مؤلفة من 64 مضيف، فإنّ جميع المضيفين ينبغي أن يكون لديهم احتمالات متقاربة لاستقبال طرود جديدة واردة، وهذا سيؤدي بشكل منطقي إلى وجود أنتروبية عالية. إنّ ورود طرد جديد، لا يوجد له تطابق مع جدول التسيير في المبدّل، يعني أنّه سيتم توجيهه إلى المتحكم. في حال بدأ مضيف أو عدة مضيفين باستقبال سلسلة متتالية من الطرود الواردة، سوف تتناقص العشوائية وبالتالي ستنخفض الأنتروبية. سنستخدم هذه الخاصية في بحثنا لكشف الهجوم في المراحل المبكّرة. بالاعتماد على الاختبارات التي أجريناها أثناء البحث، قمنا باختيار عتبة threshold معينة، وسيتم اعتبار القيم المنخفضة عنها على أنّها هجمات. تعدّ ميزة قابلية البرمجة في هذه الشبكات إحدى أهم السمات، ففي أية وقت تتغير فيه إعدادات الشبكة، يمكن ضبط قيمة العتبة، ويمكن إعادة ضبطها أثناء النقل الحي للمعطيات داخل الشبكة. بالاعتماد على الشبكة، يمكن أن يتم حساب الأنتروبية للعناوين المنطقية للعقدة الهدف، علامة الشبكات الافتراضية المحلّية VLAN tag، البوابة الهدف أو لأية حقل آخر. في حال كانت العشوائية أقل من العتبة، فيتم الإقرار بأنّ هجوماً يُشنُّ حالياً.

2.5 الأنتروبية لكشف هجوم منع الخدمة:

تعدّ الأنتروبية طريقة فعّالة للكشف عن هجمات منع الخدمة. يعدّ معاملي حجم النافذة والعتبة المكونين الأساسيين في الطرق التي تستخدم الأنتروبية لكشف هكذا هجمات. يعتمد حجم النافذة إما على الفترة الزمنية أو على عدد الطرود. يتم حساب الأنتروبية خلال النافذة وذلك من خلال قياس مقدار الشك في الطرود الواردة. لكشف الهجوم، نحتاج إلى عتبة، ففي حال كانت قيمة الأنتروبية المحسوبة أقل من العتبة الموضوعة، هذا يعني أن

هجوماً قد أُكتشِف. في [18]، يقترح الباحثون طريقة مع عرض نافذة زمنية (مقداره 0.1 ثانية) 3 مستويات للعتبة. تهتمُّ هذه الطريقة في تجنّب الإنذارات الخاطئة في الشبكة، لكن وباعتراف الباحثين أنفسهم، فإنّ هذه الطريقة تستهلك الكثير من الوقت والموارد. بينما يقترح آخرون في [19] طريقة أسرع في حساب الأنتروبية من خلال جعل عملية الحساب تقتصر على أنواع الطرد وحجومها في الشبكة. تستخدم هذه الطريقة أيضاً الفترات الزمنية، أما من أجل العتبة، فيقوم الباحثون بتشغيل مجموعة قواعد معطيات لإيجاد العتبة المناسبة وهي عبارة عن مجموعة من الانحرافات المعيارية standard deviation لقيم الأنتروبية.

3.5 مقياس العشوائية:

إنَّ السّبب الكامن وراء استخدام الأنتروبية هو قدرتها على قياس العشوائية في الشبكة، فالعشوائية العالية تعني أنتروبية عالية والعكس صحيح.

ليكن لدينا المتحول Ws وهو عبارة عن مجموعة من المعطيات بـ i عنصر، و e هو حدث في هذه المجموعة، عندها يكون احتمال حدوث e في Ws كما موضح في المعادلة (5-1). لقياس الأنتروبية التي يُرمز لها بِـ H، نقوم بحساب احتمالات جميع العناصر في المجموعة ونجمعها كما في المعادلة(5-3).

$$Ws = \{e_1, e_2, e_3 \dots \dots \dots, e_i\} \dots \dots \quad (1-5)$$

$$P_j = \frac{e_j}{i} \dots \dots \quad (2-5)$$

$$H = -\sum_{i=1}^{n} P_i \log(P_i) \dots \dots \quad (3-5)$$

سوف تكون قيمة الأنتروبية عظمى إذا كانت جميع العناصر متساوية الاحتمالات. في حال ظهور عنصر أكثر من الآخرين، سوف تكون قيمة الأنتروبية أقل. يدعى حجم Ws حجم النافذة. في حال وجود دفق مستمر من

المعطيات الواردة (في حالتنا المعطيات هي ترويسة الطرود) فسوف تنقسم إلى مجموعات متساوية تدعى نوافذ. في النافذة، يتمُّ احتساب كل عنصر مع عدد تكراراته. على سبيل المثال، إذا كان حجم النافذة 64، وجميع العناصر ظهرت مرة واحدة فقط، عندها تكون الأنتروبية 1.8، لكن في حال ظهور أحد العناصر 10 مرات، فسوف تكون الأنتروبية 1.64. سوف نستخدم هذه الخاصية لحساب العشوائية في متحكم SDN.

كما ذكرنا سابقاً، عند ورد الطرود إلى المتحكم، يكون العنوان المنطقي الخاص بالمصدر دائماً جديد، لهذا السبب تّم إرساله إلى المتحكم. أيضاً، الحقيقة الأخرى المعروفة هي أنّ الطرد الجديد القادم إلى المتحكم لديه هدف موجود ضمن الشبكة التي يسيطر عليها المتحكم، بمعرفة أن الطرد جديد، وأنّ الهدف موجود ضمن الشبكة، فإنّنا نستطيع تحديد مستوى العشوائية بحساب الأنتروبية اعتماداً على حجم النافذة. إنّ حجم النافذة هو عدد الطرود الجديدة الواردة والتي تُستخدم في حساب الأنتروبية. في هذه الحالة، تكون الأنتروبية بقيمتها العظمى في حال كان كل طرد موجه نحو مضيف وحيد محدد في الشبكة، وتكون القيمة الدنيا للأنتروبية في حال كانت جميع الطرود في النافذة موجهة نحو مضيف واحد.

4.5 استخدام الأنتروبية في الشبكات التقليدية لكشف هجوم منع الخدمة الموزّع:

قبل أن نخوض في الحل المقترح، لنلقي نظرة سريعة على آلية استخدام الأنتروبية في الشبكات التقليدية لاكتشاف هجوم منع الخدمة.

يطرح الباحثون في [1] طريقة إحصائية تعتمد على حسابات الأنتروبية، حيث يقومون بحساب الأنتروبية ضمن نوافذ صغيرة الحجم. تفترض الدراسة حجم النافذة 50 طرداً لجمع الإحصائيات. في هذه الطريقة، تمَّ تجريب أحجام مختلفة للنافذة للحصول على قياسات أنتروبية أمثلية. يعرض الجدول (5-1) النتائج التي تم الحصول عليها عند تجربة أحجام مختلفة للنوافذ.

الجدول(5-1). قيم الأنتروبية عند تغيير حجم النافذة.

حجم النافذة	H_N	H_A	$\|H_N - H_A\|$	S_N	S_A	z
5	1.36	1.98	0.62	0.79	0.48	1.29
10	1.89	2.72	0.83	0.98	0.56	1.49
50	3.11	4.22	1.11	1.35	0.65	1.7
100	3.59	4.73	1.15	1.39	0.64	1.8
500	4.54	5.51	0.96	1.05	0.40	2.4
1000	4.88	5.67	0.79	0.78	0.32	2.48
5000	5.50	5.92	0.42	0.31	0.13	3.25

من الجدول السابق، W هي حجم النافذة، H_N هي الأنتروبية في الشروط الطبيعية، H_A هي الأنتروبية أثناء وجود هجوم، S_N و S_A هما الانحراف المعياري للأنتروبية لحركة المعطيات في شروط طبيعية وعند وجود هجوم على الترتيب. أما z فهو معامل اختبار الأهمية، أو بمعنى آخر هو اختبار لصحة فرضية بين متوسطين لمتحولين مختلفين. عندما يكون أكثر من 1.64، تكون الفرضية صحيحة. في الجدول(5-1)، يمكن أن نلاحظ أنه من أجل حجم نافذة 50، وz هي 1.7 ، يمكن حساب قيمة z من خلال المعادلة:

$$z = \frac{|\overline{H_N} - \overline{H_A}|}{\sqrt{\sigma_n^2/n} + \sqrt{\sigma_r^2/r}} \dots\dots\dots (4-5)$$

إنّ σ_n و σ_r هما كما S_N و S_A. أما n فهي مجموعة من طرود البيانات العادية (قيمة n غير معطاة)، أما r فهي 25. لاختبار الفرضية، فقد تمّ إجراء اختبار من طرف واحد مع مجال ثقة 5%. توضح المعادلة(5-5) الشكل المتخذ في هذا الاختبار حيث $\bar{x}$ هي متوسط المجموعة، μ_0 متوسط العينات، σ الانحراف المعياري و n عدّاد العينات.

$$\frac{\bar{x} - \mu_0}{\sigma\ /\sqrt{n}} \dots\dots \qquad (5-5)$$

بدلاً من إجراء هذا الاختبار، قمنا باختيار العتبة بشكل تجريبي. بما أنّ القيمة العظمى والدنيا للأنتروبية تكون ثابتة في المتحكم، إذاً لدينا الحرية لاختيار العتبة عن طريق نمذدة الهجمات على المتحكم. لقد قمنا بتشغيل العديد من الهجمات لاختيار العتبة من خلال اختلاف قيم H_N و H_A. إن كانت قيمة الأنتروبية أقل من العتبة، فهذا يعني أنّ هجوماً يشنُّ الآن.

بالنظر إلى الجدول(6-1)، نلاحظ أنَّ أنتروبية وضع الهجوم أعلى منها في الوضع الطبيعي. في [1]، تعتمد الأنتروبية على بوابة الهدف والعنوان المنطقي للعقدة المصدر. إنها أعلى بسبب أنّ طرود الهجوم لديها عناوين منطقية للمصدر مختلفة ومعظمها تكون مزيفة أو منتحلة. في الوضع الطبيعي لحركة البيانات، يتم تأسيس الاتصال بين المصدر والهدف وتكون الطرود متشابهة العناوين المنطقية الخاصة بالمصدر. بما أنّ هذه الطرود لديها العدد الأكبر في الشبكة، سوف تكون أنتروبية الوضع الطبيعي أقل من تلك في وضع الهجوم والتي تساهم العناوين المنطقية الجديدة للمنبع في زيادتها. على أية حال، في شبكات SDN، تحاول الطرود تأسيس اتصال نحو الهدف من خلال جلب قاعدة التسيير من المتحكم. بالتالي، فإنَّ امتلاك عناوين منبع مختلفة لن يساهم في اكتشاف هجمات DDoS.

لقد قمنا باختيار حجم نافذة قدره 100. إنّ السبب الأساسي وراء اختيار هذه القيمة هو المحدودية في عدد الاتصالات الجديدة الآتية إلى كل مضيف في الشبكة. في الشبكات المعرفة برمجياً، بمجرّد تأسيس الاتصال، لن يتم تمرير الطرود من قبل المتحكم مالم يكن هنالك طلب جديد. إنّ السبب الآخر هو لتسهيل الحسابات والعبئ على الحاسب نتيجة قلة الموارد المتاحة، وأيضاً سبب آخر لاختيارنا هذه القيمة هو تجنب احتمالات الإنذارات الكاذبة. في شبكات SDN، يكون عدد المضيفين والمبدلات المتصلة بالمتحكم معروفاً، أيضاً حجم النافذة، وقيمة الأنتروبية العظمى للعناوين المنطقية الخاصة بالهدف معروفين، لذلك فالوضع الطبيعي أن يستهدف كل طرد مضيفاً محدّداً، فما علينا فعله إلا أن نقوم بحساب الطرود التي تستهدف مضيفاً معيناً أو شبكة فرعية محدّدة.

5.5 الحل المقترح:

كما رأينا سابقاً، فإنّ الوظيفة الأساسية للمتحكم، هي جمع الإحصائيات من جميع المبدّلات لكشف التدفقات غير الفعّالة، حيث تتمّ إزالة هذه التدفقات بعد زمن timeout معين. لذلك قمنا بإضافة مجموعة أخرى من الإحصائيات إلى المتحكم، حيث قمنا بإضافة أنتروبية العنوان المنطقي الخاص بالهدف في المتحكم. يقوم هذا التابع بتحديد ما إذا كان معدل التوجيه نحو هدف معين أكثر من المعدّل الطبيعي. كما ذكرنا سابقاً، فإن حجم النافذة تم اختياره ليكون 100. بالتالي يبقى علينا اختيار العتبة المناسبة والذي سوف نناقشه في جزء المحاكاة والنتائج.

في التابع الجديد، كل 100 طراد وارد سوف يتم احتساب الأنتروبية الخاصة بهم. في حال كانت الأنتروبية المحسوبة أقل من العتبة وبقيت ثابتة لفترة معينة أي لخمس قياسات متتالية، عندها يتم الإقرار بأن هجوماً يحدث. على اعتبار أننا ننتظر حوالي 100 طرد فقط لنعلم بوجود هجوم، فهذا الأمر يعطي إمكانية الإنذار المبكر عن الهجوم.

مع نافذة بطول 100 طرد، وشبكة فيها 50 مضيف أو أكثر، تكون الأنتروبية العظمى عندما يتم توزيع كل من هذه الطرود الـ 100 بالتساوي على جميع المضيفين. عندما يتم يحدث الهجوم، فإنّ عدد الطرود الذاهبة إلى نفس الهدف، أو الشبكة الفرعية، يكون أعلى، مما يجعل الهدف غير متاح أمام الطرود الشرعية، وهذا غالباً ما يكون الهدف الأساسي من الهجوم.

إنّ إحدى السمات الجيدة في الحل المقترح هي الحرية في اختبار المتحكم مع معدلات هجوم مختلفة لمعايرة العتبة. ليكن I هو العنوان المنطقي لجميع المضيفين المتصلين في الشبكة، بالنظر إلى المعادلة(5-7)، W هو النافذة التي تحتوي الطرود الجديدة، عنوان الهدف المنطقي x، وعدد مرات تكراره y يكن لدينا:

$$I = \{x_1, x_2, x_3 \dots \dots \dots, x_N\} \dots \dots \quad (6-5)$$

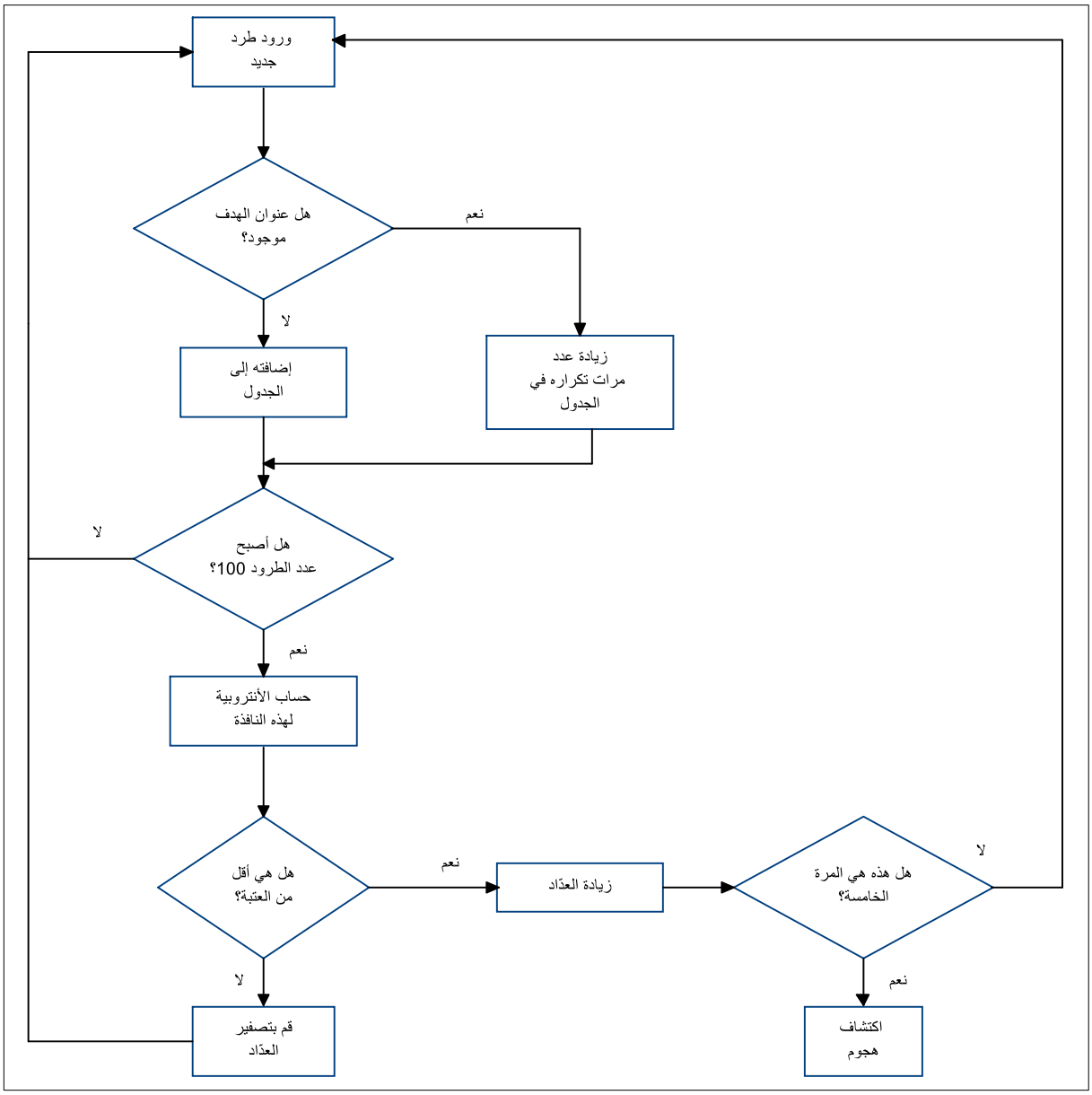
$$W = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots\dots\} \dots\dots \quad (7-5)$$

بالتالي سوف تكون الأنتروبية بقيمتها العظمى في حال كان كل عنوان منطقي للهدف وحيداً:

$$\forall x \epsilon W: H_{MAX} \Longrightarrow y = 1 \dots\dots \quad (8-5)$$

في حال لم يتم استيفاء الشرط السابق، فهذا يعني أنَّ هنالك بعض العناوين المنطقية قد ظهرت أكثر من مرة. إذاً لا يتم اعتبار أن هنالك هجوم إلا في حالة كانت الأنتروبية تحت العتبة المحددة، وكانت هذه القيمة مستمرة لِـ 5 حسابات متتالية لتجنب الإنذارات الكاذبة.

يبين لنا الشكل(5-1) المخطط التدفقي لآلية الكشف المقترحة، عند ورود طرد جديد مع عنوان منطقي جديد، أولاً، نقوم بتفحص العنوان المنطقية الهدف لمعرفة ما إذا كان العنوان المنطقي الهدف موجود نسخة منها في نافذتنا. في حال كان كذلك، نزيد العدّاد الخاص بهذا العنوان، وإلا سوف تتم إضافته كعنوان جديد. في الخطوة التالية، نقوم بتفحص ما إن أصبح لدينا 100 طرد، إن كان كذلك، نقوم بحساب الأنتروبية الخاصة بهذه النافذة. أخيراً، نقوم بمقارنة الأنتروبية مع العتبة، إن كانت أعلى منها، نقوم بالرجوع خطوة إلى الوراء بانتظار طرود جديدة، وهنا يكون عدّاد الخمس حسابات المتتالية بقيمة صفر. في حال كانت الأنتروبية أعلى من العتبة سوف نقوم بزيادة عدّاد الخمس حسابات المتتالية، فإنّ أصبح العدّاد على القيمة 5، هذا يعني أنّ هجوماً قد بدأ.



الشكل(5-1). الخوارزمية المقترحة للكشف عن الهجوم DDoS.

❖ لا يزال المرء عالماً ما دام في طلب العلم، فإذا ظن أنه قد علم فقد بدأ جهله. (ابن قتيبة)

1.6 مقدمة:

سوف نعرض في هذا الفصل كيفية تنجيزنا للجزء العملي الخاص بتحصيل قيم الأنتروبية والنتائج العملية التي حصلنا عليها، وكيفية اكتشاف هجمات منع الخدمة الموزّعة والعمل على التصدّي لها وتخفيف أثرها على أداء الشبكات المعرّفة برمجياً، ومناقشة تلك النتائج.

كما ذكرنا سابقاً، إنّ موارد المتحكمات في الشبكات المعرّفة برمجياً محدودة، لذلك فإنّ معظم الخوارزميات التي يتم طرحها لتلافي الثغرات الأمنية في الشبكات المعرّفة برمجياً تسعى إلى اكتشاف الهجوم عند بدايته أو في بضعِ المئات الأولى من الطرود، لتتمكن من إنقاذ المتحكم قبل أن يسقط نتيجة الأعداد الضخمة من الطرود المزيّفة.

2.6 إعدادات المحاكاة:

1.2.6 المتحكم:

بدايةً نقوم باختيار المتحكم، طبعاً يوجد العديد من المتحكمات المتاحة، لكننا سوف نقوم باختيار المتحكم POX [31]، نظراً لاستخدامه بشكل كبير في التجارب العملية، فهو يمتاز بالسرعة، ويمكن التعديل والبناء عليه، ويعتبر تحديثاً على المتحكم NOX [32] وكلاهما يعمل بلغة البرمجة Python. يعمل المتحكم POX في بيئة Linux، Mac OS وويندوز. يوجد أيضاً متحكم FloodLight [33] ويتم برمجته بلغة جافا، أيضاً يوجد المتحكم Beacon [34] ويبرمج بلغة جافا ويمتاز بمعدل تدفق كبير وتأخير زمني منخفض نسبياً. يعدّ المتحكم OpenDayLight [35] النسخة الأحدث من متحكمات OpenFlow.

2.2.6 محاكي الشبكة:

قمنا باستخدام المحاكي Mininet [36]، حيث يعدّ هذا المحاكي من الأدوات التي يتم استخدامها بشكل كبير في الأبحاث والدراسات التي تعتمد على الشبكات المعرّفة برمجياً. يمكن تنصيب هذا المحاكي على

حاسب شخصي محمول أو عادي. يوفر Mininet العديد من السمات والخدمات مثل جداول ARP، التعامل مع البيئة الافتراضية، كما يمكن نمذجة شبكات كبيرة واختبارها.

يمكن لمحاكي Mininet تشغيل مجموعة من المضيفين، المبدّلات، الموجّهات، وخطوط الاتصال على نواة linux واحدة. يقوم هذا المحاكي باستخدام بيئة افتراضية خفيفة لجعل النظام ككلّ يبدو وكأنه شبكة متكاملة. يقوم المضيف في هذه البيئة بأخذ سلوك الجهاز الحقيقي، يمكن الاتصال معه عن طريق SSH، وتشغيل البرامج. يمكن لهذه البرامج إرسال طرود إلى واجهات fast Ethernet حقيقة، مع سرعة خط معطاة ومع تأخير زمني محدّد.

1.2.2.6 ميزات المحاكي Mininet:

لقد قمنا باختيار المحاكي Mininet لعدة أسباب، نذكر منها:

1. سرعته: فعملية تشغيل شبكة كاملة لا تستغرق بضعة ثواني.
2. بإمكاننا إنشاء الطوبولوجيا التي نرغب بها، شبكة بمبدّل واحد أو شبكة ضخمة، مركز بيانات أو غيرها من الطوبولوجيات.
3. بإمكاننا تشغيل برامج حقيقية، نستطيع تشغيل أية برنامج يعمل على Linux، مخدمات web، أو أدوات مثل Wireshark.
4. يمكننا خصخصة عمليات تسيير الطرود على اعتبار أنّ المبدّلات تدعم بروتوكول OpenFlow.
5. يمكن تشغيل Mininet على الحاسب المحمول، على مخدّم، داخل آلة افتراضية Virtual machine، صندوق لينوكس، أو ضمن سحابة cloud.
6. يمكننا مشاركة وإعادة نسخ النتائج بسهولة.
7. سهل الاستخدام.

3.2.6 توليد الطرود:

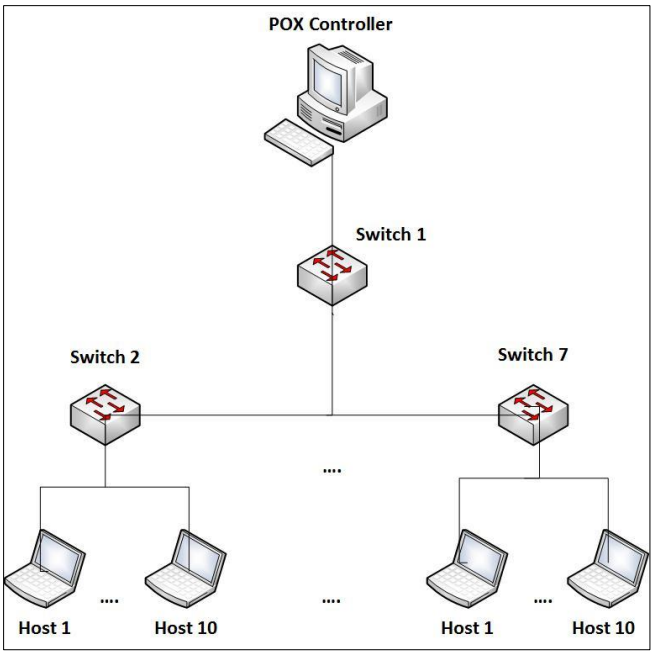
نقوم بتوليد الطرود باستخدام برنامج Scapy [37]، حيث يعتبر أداة فعالة لتوليد المعطيات، المسح، التجسس، تزوير المعطيات. نستخدم برنامج Scapy لتوليد طرود UDP وانتحال العنوان المنطقي للمصدر.

نستخدم لغة البرمجة Python لبرمجة POX، حيث قمنا بكتابة الرماز الذي يقوم بتوليد عناوين منطقية عشوائية للمصدر والمضيفين بلغة Python.

4.2.6 إعداد الشبكة:

تمّ إجراء التجارب على حاسب محمول من نوع DELL، يمتاز بمعالج corei5، بسرعة 2.7 GHz، وذاكرة عشوائية حجمها 8 غيغابايت، مع كرت شبكة يدعم السرعات 10/100/1000 Mbitps. نظام التشغيل هو Linux Ubuntu 12.04 ونسخة الـ mininet هي 2.0.0، تدعم هذه النسخة نسخة 1.0 من البروتوكول OPENFLOW.

باستخدام mininet، قمنا بإنشاء شجرة شبكة بعمق 2 مع 7 مبدّلات 60 مضيف. يظهر لنا الشكل (6-1) هذه الشبكة. قمنا باستخدام مبدّل برمجي OVS (Open Virtual Switch) [38]، حيث يعمل هذه المبدّل على بشكل عتادي صلب وبشكل برمجي. من أجل بحثنا، لا يوجد اختلاف بين المبدّل OpenFlow والمبدّل OVS. كلاهما يقوم بالعمل نفسه ويدعمهما mininet.



الشكل(6-1). الشبكة التجريبية مع 7 مبدّلات 60 مضيف.

5.2.6 اختيار العتبة:

بعد إعداد جميع المعاملات لكامل الشبكة، نحتاج إلى العتبة لكشف هجوم منع الخدمة. تفترض آليتنا أنه إن كانت الأنتروبية أقل من العتبة، وبعد إجراء 5 قياسات متتالية، عندها يتم الإقرار بوجود هجوم.

لإيجاد مجال لعتبة أمثلية، نقوم بتشغيل سلسلة من التجارب لمشاهدة تأثير الهجوم على الأنتروبية. تغطي التجارب عملية الهجوم على مضيف واحد أو شبكة فرعية مكونة من 6 مضيفين. لمقارنة معدلات مختلفة من الطرود الواردة، نقوم بالتحكم بمعدّل حركة البيانات في الوضع الطبيعي وأثناء الهجوم لزيادة أو إنقاص شدة الهجوم DDoS في المتحكم. تُظهر لنا المعادلة(6-9) المعدّل R لطرود الهجوم الواردة على الطرود الطبيعية الواردة، حيث P_a و P_n هي رقم طرود الهجوم وحجم تبادل البيانات الطبيعي على الترتيب.

$$R = \frac{P_a}{P_n} \times 100\% \ \ldots\ldots \qquad (9-6)$$

قمنا بتشغيل هجوم بمعدّل 25% على مضيف وحيد 25 مرة لإيجاد العتبة المناسبة. إنّ هذه العتبة هي الأنتروبية الأعلى في جميع الحالات والتي ستمكّن المتحكم من كشف أية هجوم مع تزييف 25% من الطرود الواردة أو

أكثر. نسمّي هذا المعدّل بمعدّل الهجوم 25%. يظهر الجدول (6-1) العتبات والمقارنات مع معدّل حركة مرور البيانات الطبيعي. اخترنا وضع قيمة العتبة 1.51، لجلب هذه القيمة قمنا بمايلي:

أ. حساب القيمة الدنيا التي يمكن لأنتروبية حركة مرور البيانات الطبيعية أن تصل إليها والتي تبلغ 1.6669.

ب. حساب القيمة العظمى التي يمكن لأنتروبية الهجوم أن تصل إليها والتي تبلغ 1.505.

ج. إيجاد الاختلاف بين القيمتين وتبلغ قيمته 0.1619، أي هنالك انخفاض بمقدار 10%.

بالرغم من أنّ الحسابات السابقة تشير إلى أن العتبة يجب أن تكون 1.505، إلا أنه وبعد تشغيل المحاكاة 25 مرة، وجدنا أن القيمة 1.51 هي العتبة الأنسب والتي يمكن أن يستطيع المتحكم من خلالها الإقرار بأنّه في حال كانت قيم الأنتروبية أقل منها فإنّ هذا يعني وجود هجوم حالياً.يوضّح *الجدول(6-4)* باقي العتبات المختارة.

الجدول(6-1). حساب قيمة العتبة.

	وضع طبيعي بدون هجوم	**هجوم بمعدّل 25%**
الأنتروبية الوسطى	1.67	1.5
الانحراف المعياري	0.008	0.013
مجال الثقة	0.0031	0.005
القيمة العظمى مع مجال الثقة	1.6731	1.505
القيمة الدنيا مع مجال الثقة	1.6669	1.495
الفرق بين قيمة الأنتروبيات	0.1619	
العتبة	1.51	

3.6 النتائج العملية:

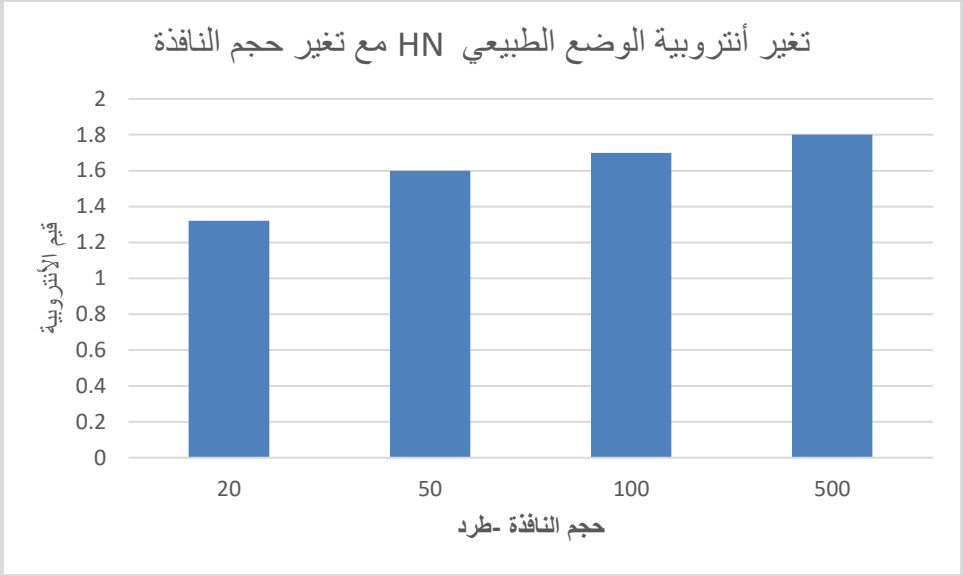
قمنا باختبار الأنتروبية مع 3 أطوال مختلفة للنافذة. يبيّن لنا الجدول (6-2) التالي، والأشكال (6-2) و(6-4) اختلاف الأنتروبية وعدد طرود الهجوم مع كل نافذة. يعبّر المعامل $H_N - H_A$ عن الاختلاف في الأنتروبية بين حالة عدم وجود هجوم وحالة وجوده.

مع نافذة بطول 20، يكون اختلاف الأنتروبية أقل من 10% مما يصعّب مسألة اختيار العتبة. من جهة أخرى، فإنَّ النافذة التي طولها 500 طرد، لا تقدم اختلافاً أفضل للأنتروبيات وتأخذ زمناً أكثر من النافذة ذات الطول 50 في حساب الأنتروبية. الاختلاف هو 0.2 بالنسبة للنافذة بطول 500 طرد حيث تنخفض 9% من أنتروبية الوضع الطبيعي. أما في النافذة ذات الطول 50 طرد، فالاختلاف 0.2 حيث تنخفض 9% من أنتروبية الوضع الطبيعي.

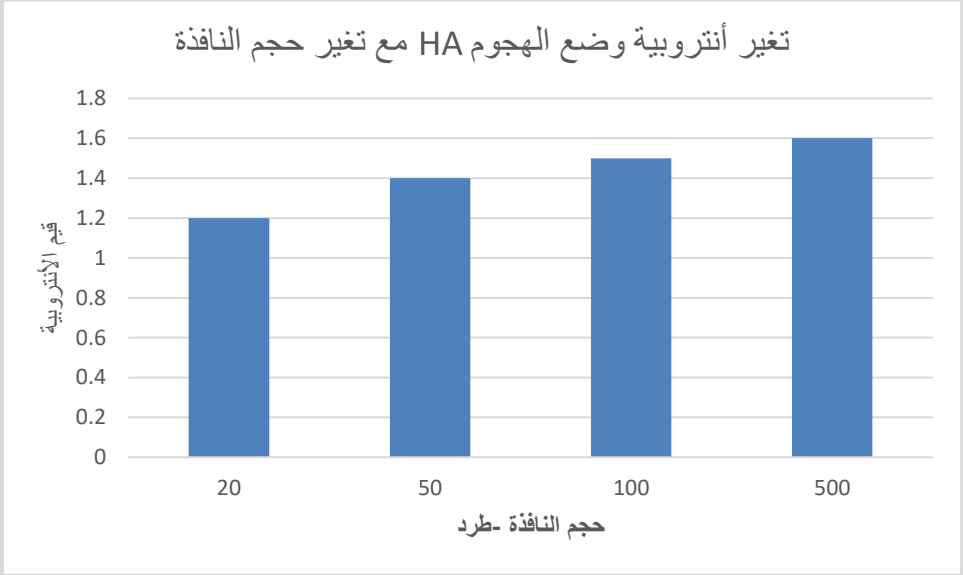
كما ذكرنا سابقاً قمنا باختيار طول النافذة 100. الجدير بالذكر هو ملاحظتنا عدم اختلاف معدل استخدام الذاكرة مع تغيير حجم النافذة، أمّا معدّل استخدام وحدة المعالجة المركزية فقد كان يزداد مع ازدياد حجم النافذة.

الجدول(6-2). مقارنة بين الأنتروبيات من أجل النوافذ الخمسة.

حجم النافذة	H_N	H_A	H_N - H_A
20	1.32	1.2	0.12
50	1.6	1.4	0.2
100	1.7	1.5	0.2
500	1.8	1.6	0.2



الشكل(6-2). تغير أنتروبية الوضع الطبيعي HN مع تغير حجم النافذة.



الشكل(6-3). تغير أنتروبية وضع الهجوم HA مع تغير حجم النافذة.

تغطي التجارب 5 حالات من الهجمات وحركة المرور الطبيعية للبيانات. تعمل 3 هجمات مختلفة الشدة على مضيف واحد، أو شدتين مختلفتين على 6 مضيفين من نفس الشبكة الفرعية أو تابعين للمبدّل ذاته. بينما في وضع حركة المرور الطبيعية، يتم توليد المعطيات بشكل عشوائي إلى جميع المضيفين. نشغّل ترافيك الهجوم من مصدر واحد، طبعاً نقوم بتشغيل الهجوم بشكل يدوي بعد انقضاء زمن معين من المحاكاة.

في Mininet، نقوم باختيار العناوين المنطقية لجميع المضيفين من الشبكة 192.168.0.1/24. في حالة الهجوم من مضيف وحيد، نقوم باختيار مضيف عشوائي على إحدى المبدّلات، بينما يبقى الوضع طبيعياً بالنسبة لباقي المضيفين. أيضاً في الهجوم على 6 مضيفين، نقوم باختيار 6 مضيفين بشكل عشوائي. يعرض لنا الجدول(6-3) إعدادات ترافيك الهجوم. سوف تكون الطرود جميعها من نوع UDP، وطبعاً البوابة الهدف 80، ونوع الهجوم DDoS. بشكل عام، في المتحكم OpenFlow، لا يتم إرسال الحمل، إنما يتم إرسال ترويسة الطرد فحسب.

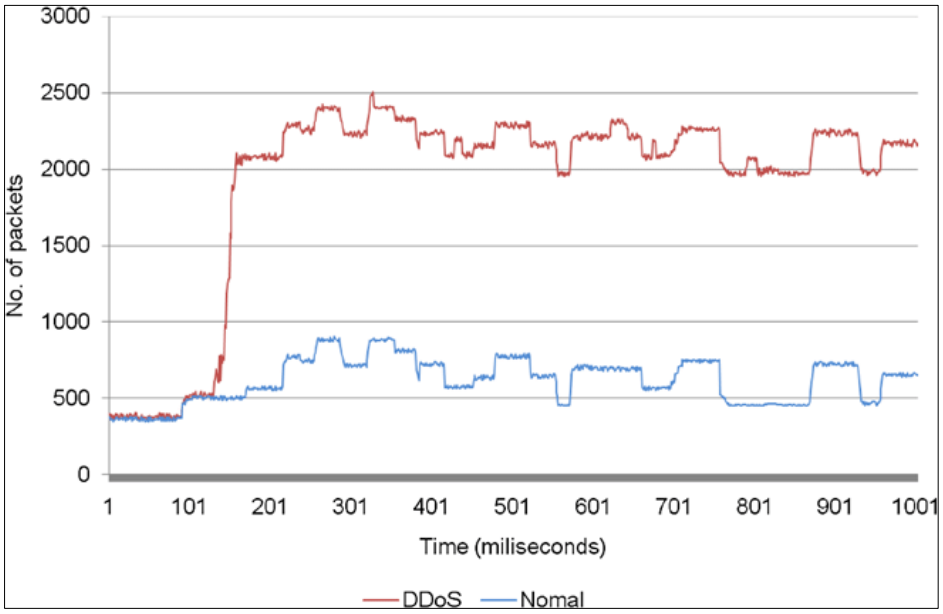
الجدول(6-3). إعدادات حركة المرور البيانات عند الهجوم.

البروتوكول	البوابة	الحمل	نوع الهجوم
UDP	80	لا يوجد	DDoS

في حالة الاختبار الثلاث للهجوم على مضيف وحيد، تكون كثافة الهجوم 25%، 50% ،75%. في جميع الاختبارات، نستخدم برنامجين Scapy، الأول يولّد حركة بيانات طبيعية، والآخر يولّد حركة مرور مهاجمة. يؤدي هذا التصميم إلى خسارة من 23 إلى 27 طرد من أصل 100 في حالة كان معدّل الهجوم 25% على مضيف وحيد. بينما في حالة معدّل هجوم 50% يكون الفقد حوالي 51 طرد، أما في حالة 75% فيكون هنالك 78 طرد من أصل 100.

في حالة الهجوم على شبكة فرعية مكونة من 6 مضيفين، سنكتفي بإجراء هجوم بمعدّل 50%، حيث سوف يستقبل كل مضيف من 7 إلى 9 طرود، وبمعدّل 75% حيث سوف يستقبل كل مضيف 11 أو13 طرد. بشكل عام هذه النسب بالنسبة لأعداد الطرود المزيفة ليست دقيقة تماماً، فأحياناً وعند اختيار المعدّل 25 % يكون عدد الطرود المزيفة 27، والتي تشكل 27% من 100 طرد، لكن هذا الأمر ليس بيدنا، إنما من بنية Mininet.

في الواقع، هجمات DDoS تكون بكثافة أعلى بكثير، حيث يتم توليد حركة مرور أضخم أعلى بكثير من حركة مرور البيانات الطبيعية. يعرض لنا الشكل(6-4) هجوماً يصل فيه عدد الطرود إلى 2000 في 100 ميلي ثانية، بينما في الوضع الطبيعي يكون حوالي 500 طرد تقريباً.



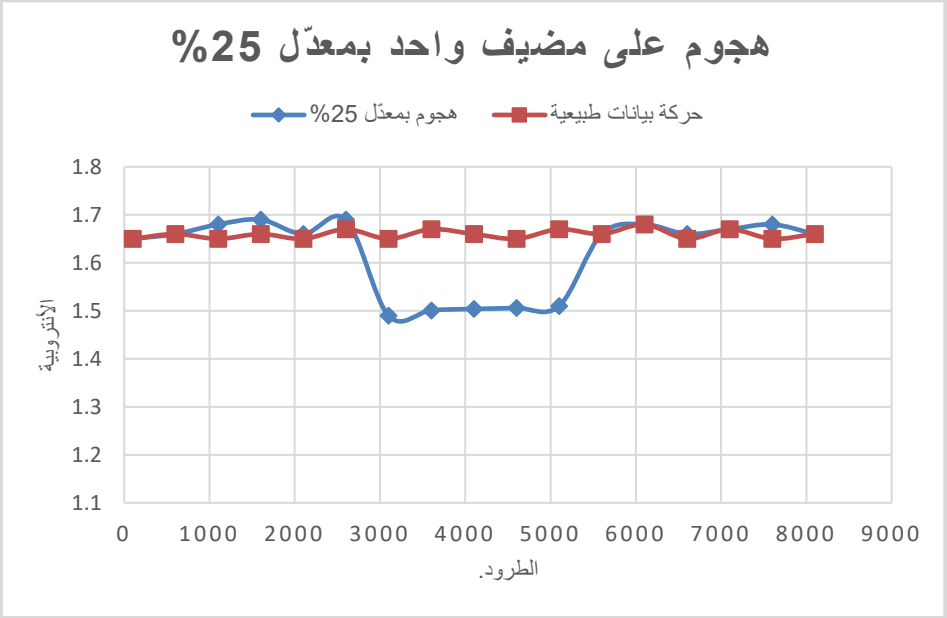
الشكل(6-4). توضيح الزيادة الفجائية في عدد الطرود المرسلة في حالة الهجوم.

1.3.6 الهجوم على مضيف وحيد:

جميع الأشكال البيانية التي سوف نعرضها لاحقاً، هي لـ 10 عمليات تشغيل مع توليد 8000 طرد في الاختبار الواحد. تعرض النقاط على المحور العمودي طول النافذة (100 طرد)، بينما يمثل المحور الأفقي قيم الأنتروبية. بيانات الشكل الواحد هي متوسط قيم الـ 10 عمليات تشغيل.

نعرض في الشكل (6-5) تغيرات الأنتروبية عند معدّل هجوم 25%. نلاحظ أنّ أول 6 قيم أنتروبية أقل من العتبة 1.51. في الجدول(6-1)، نلاحظ أنّ أقل نقطة في مجال الثقة للوضع الطبيعي كانت 1.6669 وأعلى قيمة للوضع الهجومي كانت 1.505. الفرق بين هاتين القيمتين 0.1619، يبين انخفاض الأنتروبية 9% والذي يعدّ أكبر بـ 32

مرة مجال الثقة في حالة هجوم بمعدّل 25% والذي كانت قيمته 0.005. بالتالي نلاحظ مدى سهولة هذه الطريقة وقدرتها على كشف هجوم فيه 25% طرود مزيفة أو أكثر.

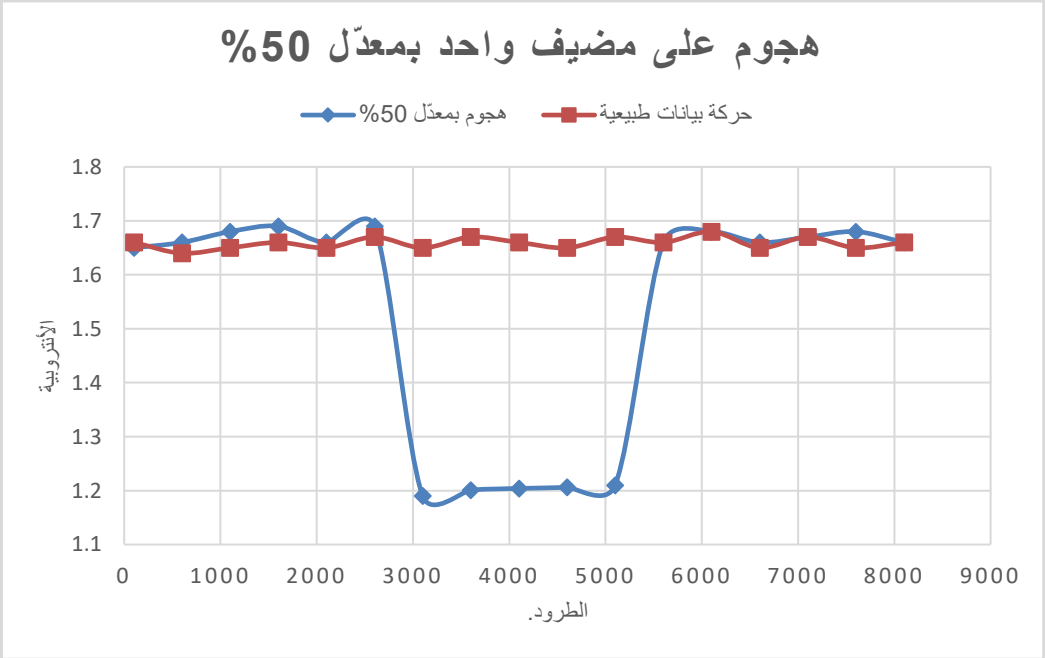


الشكل(6-5). نتيجة الهجوم بمعدّل 25% على مضيف وحيد.

يوضّح لنا الشكل (6-6-أ) والشكل (6-6-ب) نتيجة الهجوم بمعدّل 25% على مضيف واحد في المرة العاشرة والحادية عشر من مرات التكرار التي تحدّثنا عنها سابقاً، ونلاحظ تقارب النتائج مع الشكل (6-5) الذي يمثل وسطي المرات.

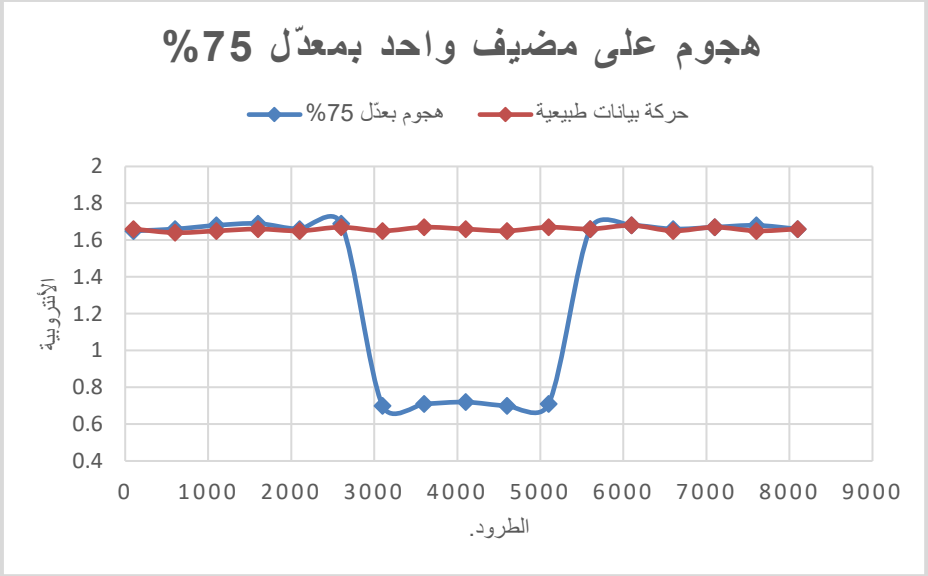
الشكل(6-6). نتيجة الهجوم بمعدّل 25% على مضيف واحد في المرة العاشرة(أ) والحادية عشر (ب).

أما في حالة معدّل 50%، أو 75% (الشكل (6-7)، (6-8)) نلاحظ أنّ نافذة الهجوم أعمق وأضيق، مما يظهر انخفاض أكبر في الأنتروبية. في هجوم المضيف الواحد، يتم إرسال 500 طرد كطرود هجوم، فعندما يزيد معدل الهجوم، ويبقى عدد الطرود المولدة ثابتاً، تزداد النسبة المئوية للهجوم في النافذة. في حالة هجوم بمعدّل 75% يكون الانخفاض ضيّقاً جداً، لذلك نزيد عدد الطرود إلى 1000 لنحصل على رؤية أوضح لانخفاض الأنتروبية.

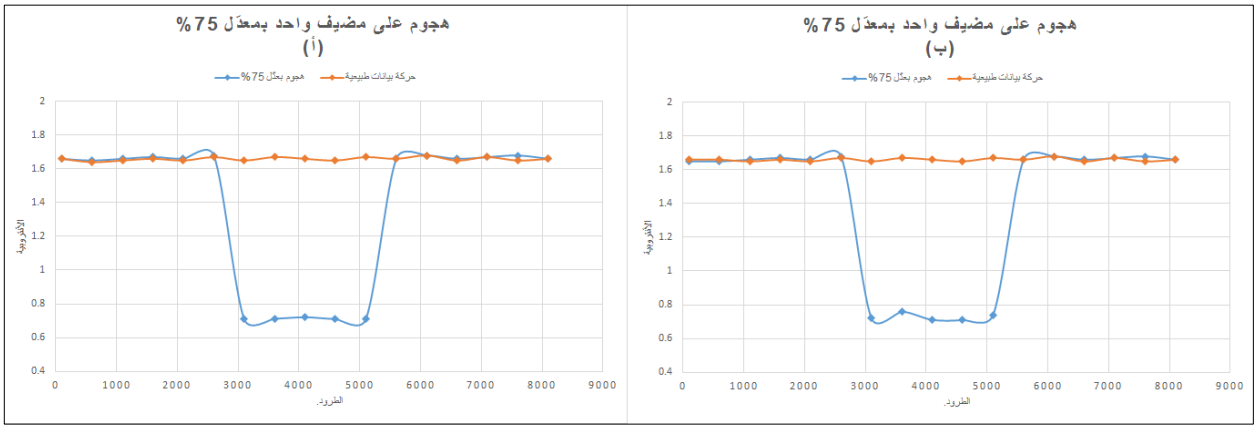


الشكل(6-7). قيم الأنتروبية في حالة هجوم بمعدّل 50%.

أما الشكل (6-8) فيوضح نتائج الهجوم بمعدّل 75% في حالة مضيف واحد، وذلك في المرات الخامسة والسادسة (أ،ب)، أيضاً نلاحظ كيفية اقترابها من المعدّل الوسطي.



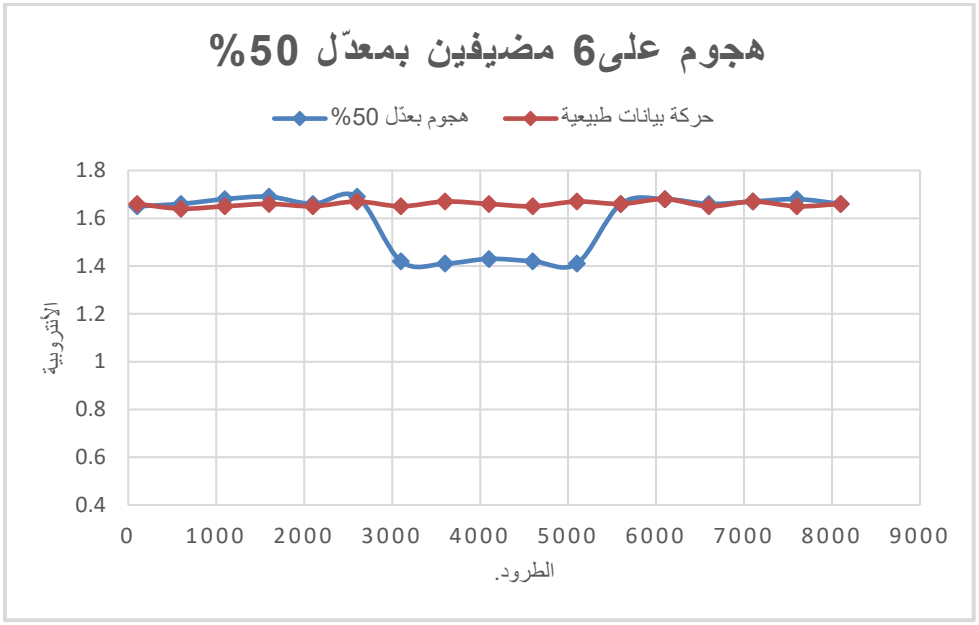
الشكل(6-8). قيم الأنتروبية في حالة هجوم بمعدّل 75%.



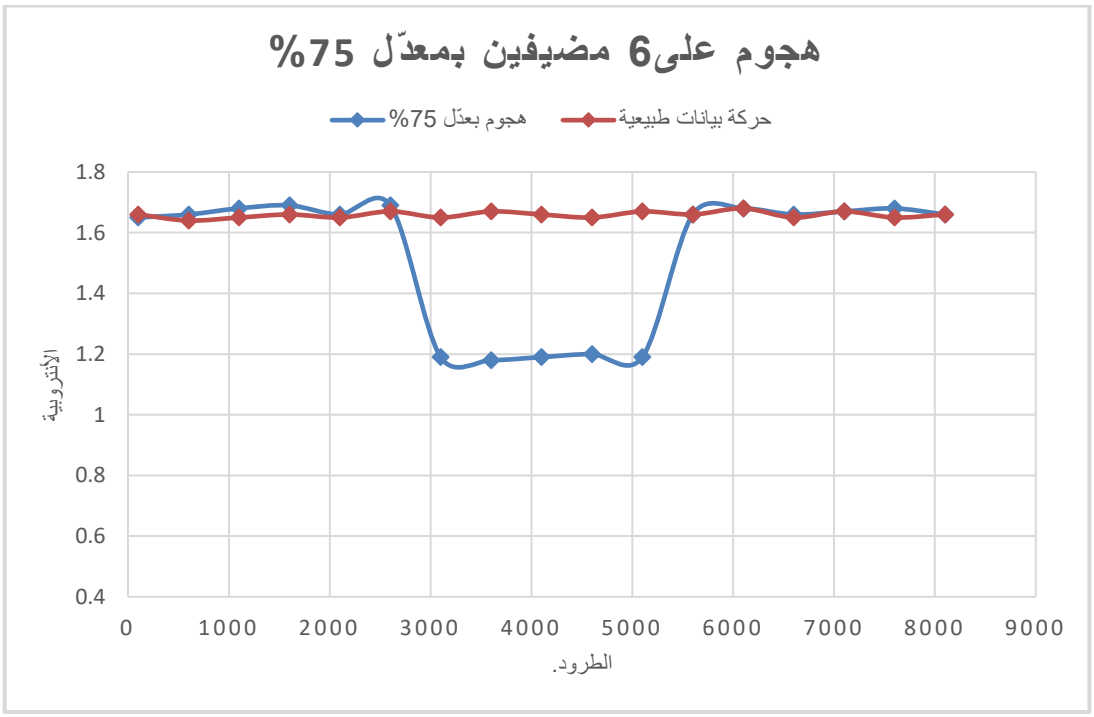
الشكل(6-9). نتيجة الهجوم بمعدّل 75% على مضيف واحد في المرة العاشرة(أ) والحادية عشر (ب).

2.3.6 الهجوم على شبكة فرعية:

أيضاً تعرض لنا الأشكال(6-10)،(6-11) نتيجة الهجوم على 6 مضيفين، ويلخص لنا الجدول(6-4) الاختلاف في قيم أنتروبيات الهجوم والعتبات. بالنسبة لمعدّل هجوم 25% فقد كان مجال الثقة 0.005.

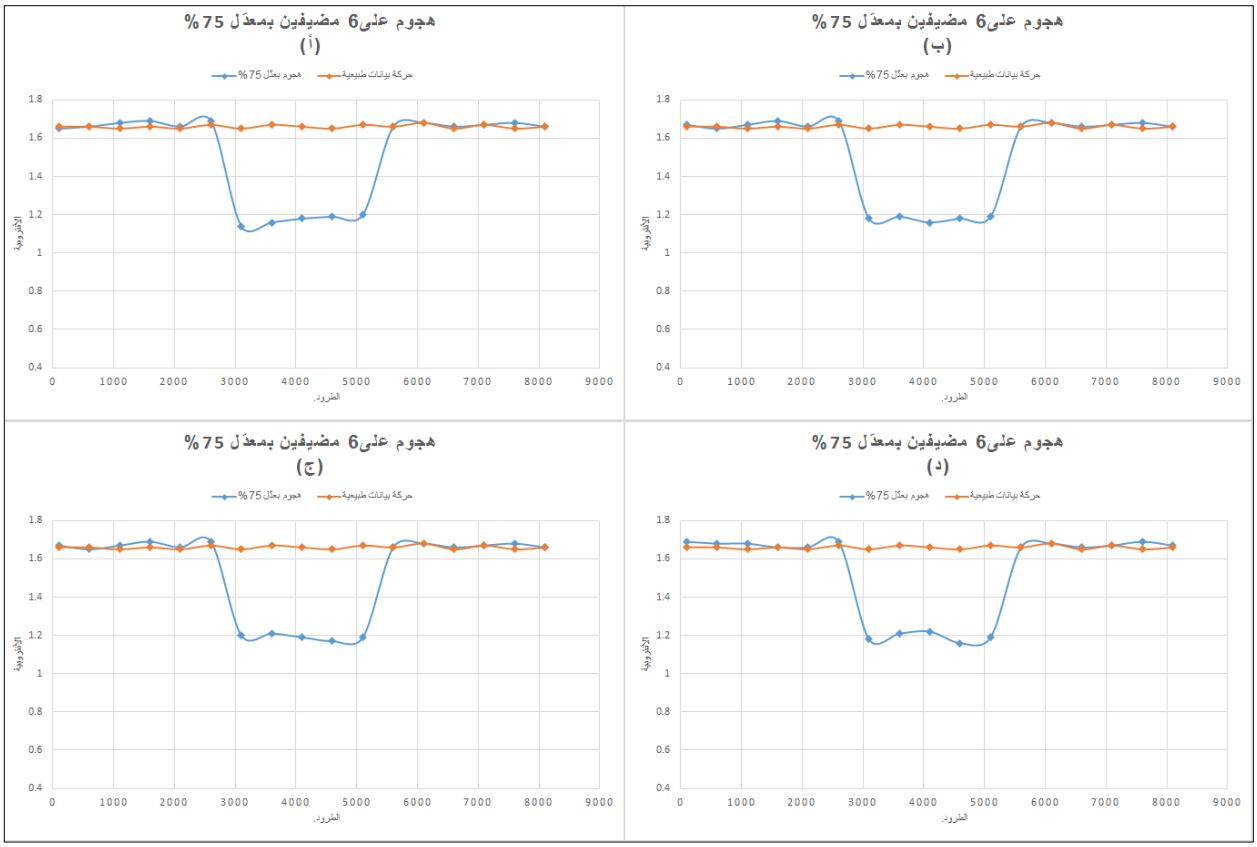


الشكل(6-10). قيم الأنتروبية في حالة الهجوم على 6 مضيفين في حالة معدّل هجوم 50%.



الشكل(6-11). قيم الأنتروبية في حالة الهجوم على 6 مضيفين في حالة معدّل هجوم 75%.

يوضّح الشكل(6-12)، نتائج الهجوم بمعدّل 75% على 6 مضيفين، في مرات مختلفة حيث نلاحظ أن النتائج متقاربة مع الأشكال السابقة.



يوضّح الشكل(6-12)، نتائج الهجوم بمعدّل 75% على 6 مضيفين، في مرات مختلفة.

الجدول(6-4). قيم الأنتروبيات في الاختبارات العملية.

نوع الهجوم	الأنتروبية الوسطى	العتبة–أنتروبية الهجوم
هجوم على مضيف بمعدّل 25%	1.5	0.01
هجوم على مضيف بمعدّل 50%	1.26	0.25
هجوم على مضيف بمعدّل 75%	0.76	0.75
هجوم على 6 مضيفين بمعدّل 50%	1.4	0.11
هجوم على 6 مضيفين بمعدّل 75%	1.18	0.33

بينما يوضح لنا الجدول(6-5) عدد الطرود المستقبلة من قبل كل مضيف في حالة المعدّلات الثلاثة، وفي حالة الهجوم على مضيف واحد أو شبكة فرعية.

الجدول(6-5). عدد الطرود الواردة إلى كل مضيف في الاختبارات.

الاختبار	وسطي الطرود المستقبلة لدى كل مضيف
الوضع الطبيعي بدون هجوم	1.56
هجوم على مضيف بمعدّل 25%	25
هجوم على مضيف بمعدّل 50%	50
هجوم على مضيف بمعدّل 75%	75
هجوم على 6 مضيفين بمعدّل 50%	8.3
هجوم على 6 مضيفين بمعدّل 75%	12.5

يوضّح الجدول (6-6) نسبة نجاحنا في معرفة حدوث هجمات في السيناريوهات المنفّذة، نلاحظ من الجدول التالي مدى فعالية الطريقة المقترحة في تحديد وجود هجوم فعلي، فكما ذكرنا سابقاً، تقوم الخوارزمية بالإقرار بوجود هجوم في حال انخفضت قيمة الأنتروبية خمس مرات متتالية عن العتبة المحدّدة.

الجدول(6-6). نسبة نجاح كشف الهجوم.

الاختبار	نسبة نجاح الخوارزمية في الكشف
هجوم على مضيف بمعدّل 25%	%95
هجوم على مضيف بمعدّل 50%	%96
هجوم على مضيف بمعدّل 75%	%97
هجوم على 6 مضيفين بمعدّل 50%	%95
هجوم على 6 مضيفين بمعدّل 75%	%97

الخاتمة و آفاق التطوير

لقد كان الهدف الأساسي في بحثنا هو طرح طريقة جديدة لمواجهة خطر التطبيقات التي قد تقوم بشنّ هجمات منع الخدمة الموزّع، وحماية المتحكم من هذه الهجمات، حيث قمنا بطرح حلّ بسيط يستطيع كشف حالات هجوم منع الخدمة الموزع بعد بضع مئات من الطرود الواردة. قمنا بالاستفادة من الميزات التي يوفرها المتحكم في شبكات SDN، ومن الإحصائيات التي يوفرها، واستفدنا من بيئة Mininet التي تتيح الكثير من الأدوات للمراقبة واستخلاص المعلومات.

بتطبيق مبدأ حساب الأنتروبية، كنا قادرين على كشف عمليات هجوم منع الخدمة الموزع، ومن خلال التكرارات المستمرة لعمليات المحاكاة، وأخذ المعدّل الوسطي لتلك النتائج، استطعنا التأكد من أنّ نسبة نجاح كشف الهجوم كانت أكثر من 95%، ففي حالة الهجوم على مضيف واحد، كنا قادرين على كشف الهجمات بشكل فعّال جداً وناجح. كذلك أيضاً بالنسبة للهجوم على 6 مضيفين.

قمنا أيضاً في هذا البحث بتغطية أهم الثغرات الأمنية التي تعاني منها شبكات SDN، وقمنا بِـ:

- إيضاح كيفية حدوث هجوم منع الخدمة الموزع وكيفية استنزافه لقدرات المتحكم.
- اقتراح حلّ بسيط للتصدّي لمثل هكذا هجمات بالاعتماد على قيم الأنتروبية.
- تنجيز الحل المقترح على بيئة Mininet واستخدام متحكم POX.
- البرهان على نجاح الحل المقترح.

آفاق التطوير

يمكن العمل مستقبلاً على دراسة الوضع في حالة تم إجراء هجوم منع خدمة على كامل الشبكة، أي عندما يستهدف الهجوم كل مضيف في الشبكة. يعالج البحث الذي قمنا به حالة الهجوم في شبكة لديها متحكم واحد، يمكن العمل مستقبلاً على إجراء دراسة على شبكة تحتوي أكثر من متحكم وكيفية تعاونهما على تخفيف الهجوم وتوزيع الحمل بينهما.

المراجع

[1] T. Nakashima, T. Sueyoshi S. Oshima, "Early DoS/DDoS Detection Method using Short-term Statistics," in International Conference on Complex, Intelligent and Software Intensive Systems, 2010, pp. 168-173.

[2] Erickson, David. "The beacon openflow controller." *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013.

[3] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76.

[4] Banerjee, Subhasis, and Kalapriya Kannan. "Tag-in-tag: Efficient flow table management in sdn switches." *10th International Conference on Network and Service Management (CNSM) and Workshop*. IEEE, 2014.

[5] Software-Defined Networking: The New Norm for Networks, ONF White Paper April 13, 2018.

[6] Kreutz, Diego, Fernando Ramos, and Paulo Verissimo. "Towards secure and dependable software-defined networks." *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, (2013).

[7] Akhunzada, Adnan, et al. "Securing software defined networks: taxonomy, requirements, and open issues." IEEE Communications Magazine 53.4 (2015): 36-44.

[8] Deb, Raktim, and Sudipta Roy. "Dynamic vulnerability assessments of software-defined networks." *Innovations in Systems and Software Engineering* (2019): 1-7.

[9] Akhunzada, Adnan, et al. "Secure and dependable software defined networks." Journal of Network and Computer Applications 61 (2016): 199-221.

[10] Izzat Alsmadi, “Security of Software Defined Networks: A survey”. Published by Elsevier Ltd (2015).

[11] Abdurohman, Maman, Dani Prasetiawan, and Fazmah Arif Yulianto. "Improving Distributed Denial of Service (DDOS) Detection using Entropy Method in Software Defined Network (SDN)." *ComTech: Computer, Mathematics and Engineering Applications* 8.4 (2017): 215-221.

[12] Douligeris, Christos, and Aikaterini Mitrokotsa. "DDoS attacks and defense mechanisms: classification and state-of-the-art." *Computer Networks* 44.5 (2004): 643-666.

[13] Rajesh wari.S, Malathi.K, Regina.B. “A Survey on Characterization of Defense Mechanisms in DDOS Attacks ". International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3, Issue-2, December 2013.

[14] Keijer, Julik. *Automated DDoS mitigation based on known attacks using a Web Application Firewall*. BS thesis. University of Twente, 2019.

[15] Mohammadi, Reza, et al. "SYN-Guard: An effective counter for SYN flooding attack in software-defined networking." *International Journal of Communication Systems* (2019): e4061.

[16] C. Ji M. Thottan, "Anomaly Detection in IP Networks," IEEE Transaction on Signal Processing, vol. 51, no. 8, pp. 2291 -2204, Aug 2003.

[17] Khalaf, Bashar Ahmed, et al. "Comprehensive Review of Artificial Intelligence and Statistical Approaches in Distributed Denial of Service Attack and Defense Methods." *IEEE Access* 7 (2019): 51691-51713.

[18] Z. Qin, L. Ou, J. Liu, A. X. Liu J. Zhang, "An Advanced Entropy-Based DDoS Detection Scheme," in International Conference on Information, Networking and Automation, 2010, pp. 67-71.

[19] I. Ra G. No, "An efficient and reliable DDoS attack detection using fast entropy computation method," in International Symposium on Communication and Information technology, 2009, pp. 1223-1228.

[20] Y. Chen X. Ma, "DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy," IEEE Communications Letters, vol. PP, no. 99, pp. 1 -4, 2013.

[21] C. Stanley. “Pairs of Values and the Chi-squared Attack”, Master’s thesis, Department of Mathematics, Iowa State University, 2005.

[22] Ham, Fredric M., and Ivica Kostanic. *Principles of neurocomputing for science and engineering*. McGraw-Hill Higher Education, 2000.

[23] Lim, Sharon, et al. "A SDN-oriented DDoS blocking scheme for botnet-based attacks." *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2014.

[24] Wang, Haopei, Lei Xu, and Guofei Gu. "Floodguard: A dos attack prevention extension in software-defined networks." *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2015.

[25] Dhawan, Mohan, et al. "SPHINX: Detecting Security Attacks in Software-Defined Networks." *NDSS*. Vol. 15. 2015.

[26] Gkountis, Christos, et al. "Lightweight algorithm for protecting SDN controller against DDoS attacks." *2017 10th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2017.

[27] Hameed, Sufian, and Hassan Ahmed Khan. "SDN based collaborative scheme for mitigation of DDoS attacks." *Future Internet* 10.3 (2018): 23.

[28] Manso, Pedro, José Moura, and Carlos Serrão. "SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks." *Information* 10.3 (2019): 106.

[29] Lawal, Babatunde Hafis, and A. T. Nuray. "Real-time detection and mitigation of distributed denial of service (DDoS) attacks in software defined networking (SDN)." *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018.

[30] Dridi, L., and M. Zhani. "SDN-Guard: DoS Attacks Mitigation in SDN Networks-IEEE Conference Publication." *Ieeexp/ore. ieee. org*. 2017.

[31] M. Murphy. (2019, Sep) POX. [Online]. https://noxrepo.github.io/pox-doc/html.

[32] T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker N. Gude, "NOX: Towards an Operating System for Networks," Computer CommunicationReview, vol. 38, no. 3, pp. 105-110, Jul 2008

[33] Dover, Jeremy M. "A denial of service attack against the Open Floodlight SDN controller." *Dover Networks LCC, Edgewater, MD, USA* (2013).

[34] Jarschel, Michael, et al. "A flexible OpenFlow-controller benchmark." *2012 European Workshop on Software Defined Networking*. IEEE, 2012.

[35] Medved, Jan, et al. "Opendaylight: Towards a model-driven sdn controller architecture." *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, 2014.

[36] (2018, Sep) Mininet. [Online]. http://mininet.org/

[37] P Biondi. “Scapy Documentation”. Release 2.4.2-dev, Apr 29, 2019.

[38] (2018, Jan) Open Vswitch. [Online]. http://openvswitch.org/

[39] Y Tseng, Z Zhang, F Na¨ıt-Abdesselam. “ControllerSEPA: A Security-Enhancing SDN Controller Plug-in for OpenFlow Applications”. International Conference on Parallel and Distributed Computing, Applications and Technologies, 2016.

[40] A Shaghaghi, M A Kaafar, R Buyya, S Jha. “Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions”. Cluster Computing Journal in the second half of 2018.

[41] S Lee, C Yoon, S Shin. “The Smaller, the Shrewder: A Simple Malicious Application Can Kill an Entire SDN Environment”. *SDN-NFVSec’16, March 11 2016.*

[42] Ch Mansour, D Chasaki. “Design of an SDN Security Mechanism to Detect Malicious Activities”. IEEE, 2018.

[43] C Lee, C Yoon, S Shin, S K Cha. “INDAGO: A New Framework For Detecting Malicious SDN Applications”. International Conference on Network Protocols, 2018.

[44] C Röpke, T Holz. “Preventing Malicious SDN Applications From Hiding Adverse Network Manipulations”. ISBN 978-1-4503-2138-9., 2018.

[45] Sahoo, Kshira Sagar, et al. "Toward secure software-defined networks against distributed denial of service attack." *The Journal of Supercomputing* (2019): 1-46.

[46] Rehmani, Mubashir Husain, et al. "Software Defined Networks based Smart Grid Communication: A Comprehensive Survey." *IEEE Communications Surveys & Tutorials* (2019).