تعتبر الشبكات المعرفة برمجيًّا، وهو ما يطلق عليها باللغة الإنجليزية (Software-Defined Networking)، واختصاراً (SDN) أنها طريقة للتحكم بالشبكة مركزيًا .وهي مقاربة يحاول من خلالها المهندسون إعادة ترتيب أجزاء وأدوار كل مكونات البنية التحتية (infrastructure) للشبكات ،التي لم تعدل منذ عام ١٩٨٠ حيث أن آخر تعديل كان هو الانتقال من NCP إلى TCP/IP الغني عن التعريف ومنذ ذلك الحين لم يطرأ أي تغيير على مستوى البنية التحتية للشبكات على الرغم من التقدم التقني الهائل في عالم ال IT(تقنية المعلومات) وبشكل خاص تقنية الـ VIRTUALIZATION(البنية الافتراضية) التي استطاعت إجراء محاكاة لجميع طبقات الشبكة لكنها وقفت عاجزة عند البنية التحتية.

تاريخ ال SDN :

من المعروف أن لكل تقنية تاريخها ولكن SDN تاريخه فقط حوالي 10 سنوات على ظهوره بهذا الشكل المتكامل ،أصل ال SDN يعود إلى مقال كتبه الطالب سابقا Martin Casado بعنوان Ethane: Taking Control of the Enterprise سنة 2004 حيث أظهر عيوب البنية التحتية لشبكات الشركات التي تعتمد على مبدأ management is distributed وقدم Martin وهو حاليا مؤسس لشركة Nicira التي تم بيعها مؤخراً ل VMware ،حل أسماه Ethane التي تعتمد على فصل Control plan فيزيائيا ومنطقيا بحيث يصبح L3 Switch مسؤولاً عن forwarding .

بعد مرور خمس سنوات قام هو وفريقه في جامعة ستانفورد باستكمال نظريته من جميع الجوانب ،حيث تمت تجربة SDNأي Ethane سابقاً على صعيد Stanford compus لتكون 2009 السنة التي غرفت بداية هذه التقنية ،وتم نشر مقال بعنوان OpenFlow : enabling innovation in campus networks الذي أصبح المرجع لكل من يريد الكتابة في هذا المجال ،يعرف المقال ببروتوكول جديد اسمه Open flow .

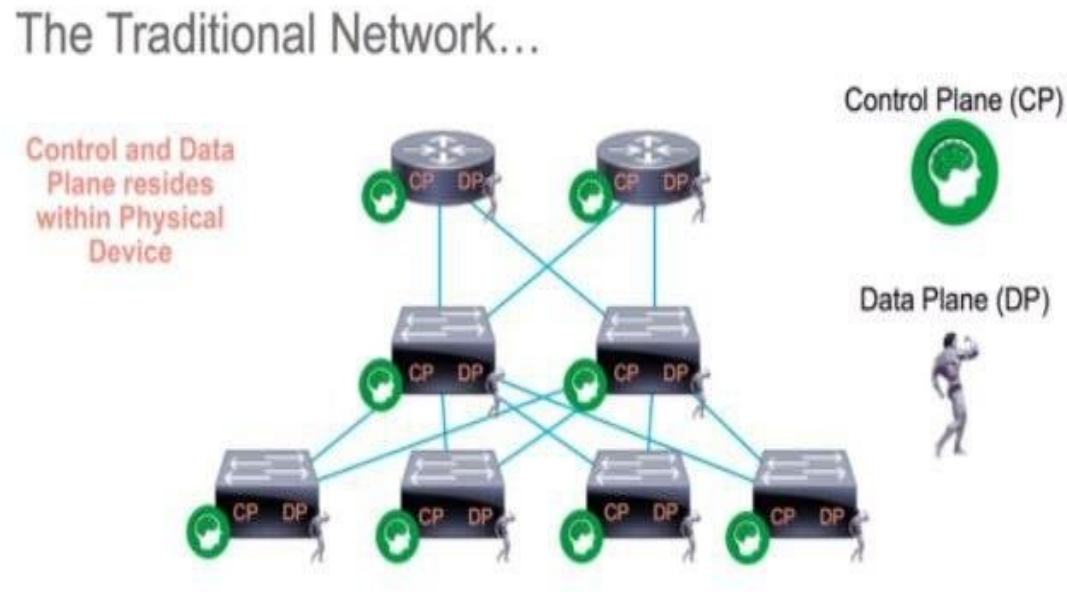
شهدت الاعوام القليلة المنصرمة عدد كبير من المشكلات التي واجهت هندسه الشبكات و تمكن العلماء من محاصرتها و حلها حلاً جذرياً لنأتي اليوم و نحل مشكله جديدة من المشكلات التي جسدت عقبة في قدرة هندسة الشبكات على الاستمرار

١- مشكله بطئ (سرعه نقل البيانات) : تم حلها من خلال fiber optics
٢- مشكله تخزين البيانات ذات الحجم الكبير : تم حل هذه المشكلة من خلال توفير سيرفرات عملاقه تستطيع تخزين أحجام ضخمه من البيانات بالإضافة إلى الحوسبة السحابية cloud computing

٣- مشكله عناوين الانترنت ip : كانت أحد المشاكل الرئيسية حيث أدى أنتشار التكنولوجيا إلى نفاذ عنوانين الاى بي IP نعم ، كانت العناوين قد أوشكت على النفاذ الا أن المجتمع التقني وجد حلاً نهائياً لهذه المشكلة بظهور IP من الاصدار السادس بدلاً من IP الخاص بالإصدار الرابع ، مما وفر مليارات ال IPS الاضافية التي نحتاج لعشرات السنوات كي ننفذ من استخدامها

٤- بقيت لنا مشكله واحده وهى البنية التحتية القوية ، فوجود بنية تحتية قويه للشبكة أمر غير متوفر بسهوله و إن توافر لنا بنية تحتية بمواصفات عالية فإن ذلك يكون مكلف جداً ،الامر الذى يجعلنا نتراجع عن شراء تلك البنية التحتية القوية

من هنا فكر العلماء ، في إيجاد بديل ، يوفر بنية تحتية قوية و بتكلفة منخفضة ، ليجدوا مبتغاهم في تقنيه جديدة تسمى softwere-defined networking SDN ، هذه التقنية تمثل مستقبل الجيل القادم من البنية التحتية لهندسه الشبكات ، **فـ Network devices المستخدمة حالياً مثل switch أو router تتكون من جزئين رئيسيين هما : control plane و data plane.**



الجزء الاول المسمى ب control planeهو المسئول عن أتخاذ القرارات و توجيه العمليات مثل توجيه الترافيك (الباندوس) إلى مسار معين في الشبكة و الجزء الثاني المسمى ب data planeهو المسئول عن تنفيذ القرارات التي يصدرها الجزء الاول هذا في الاجهزة التقليدية المستخدمة هذه الايام .

تقوم فكره SDN على الفصل بين الجزئين السابقين ، بحيث يظل الجزء الثاني في الاجهزة الاعتيادية كما هو ، بينما ينتقل الجزء الاول المسئول عن أتخاذ القرارات إلى سيرفرات ضخمه ،

تقوم تلك السيرفرات بدور control plane بصوره اكبر و أوسع و أدق و أسرع ، نظراً لقدراتها الفائقة ، مما يوفر عمليه control plane بصوره سريعة و أسعار رخيصة .

الفصل بين طبقتي التحكم والتنفيذ استدعى وجود بروتوكول ينظم التواصل بين الطبقتين، ولأن الشبكات المعرفة برمجياً تم اعتمادها من قبل منظمة ONF (Open Networking Forum) والتي تتكون من مجموعة من الشركات؛ كان لزاماً أن يتم الاتفاق على بروتوكول يتعامل مع طبقة التحكم وطبقة البنية التحتية.

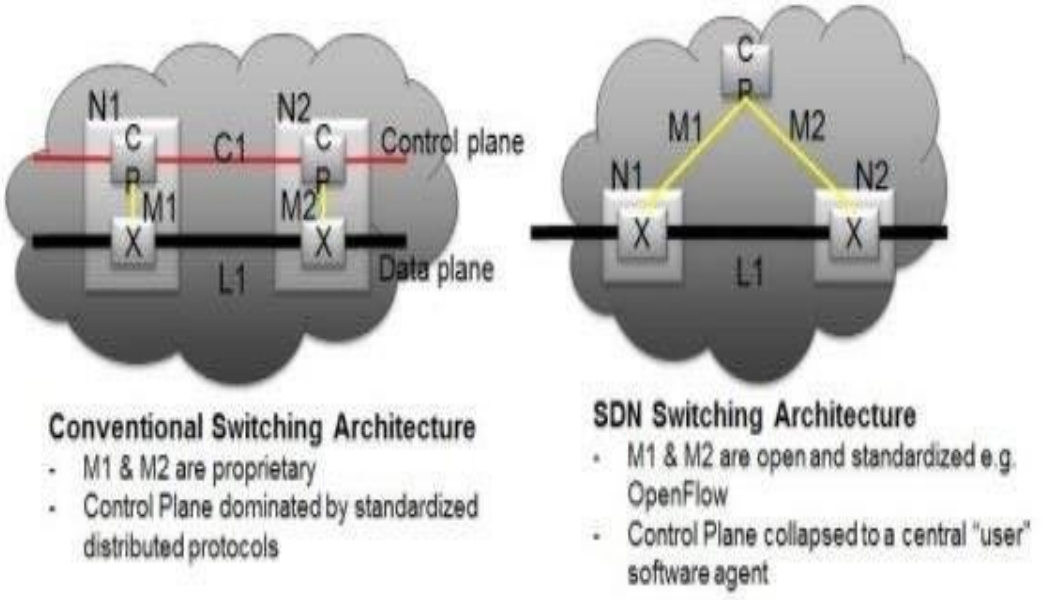
يقوم بروتوكول (OpenFlow) بتحديد مسار الرزم بناء على قواعد محددة مسبقاً بواسطة مهندس الشبكة. إضافة إلى ذلك، يحدد البروتوكول الوظيفة المناسبة (Action) كأن يقوم المُبَدّل (Switch) بتمرير رزمة البيانات (Forward)، أو تجاهلها (Drop).

ولأن الشبكات المعرفة برمجيًا هي نموذج حديث من الشبكات، فأبرز الصعوبات هي الانتقال إلى استخدامها بسبب وجود أجهزة تعمل في الوقت الحالي على النظام التقليدي. أيضاً انتقال رزم البيانات من المبدّلات إلى المتحكم ينشأ عنه بطء (Delay) بسبب المركزية في اتخاذ القرارات.

❖ مقارنة الشبكات التقليدية بال SDN :

بمقارنة الشبكة المعرفة برمجياً بالشبكات التقليدية: يمكن أن يتم التفريق بين النوعين في أن الشبكات المعرفة برمجيًا يمكن تهيئتها أثناء عمل الشبكة بشكل أسهل. إضافة إلى ذلك فإن ميزة المركزية في التحكم (والتي تعتبر من مقومات نجاح هذا النوع من الشبكات) تساعد على تقليل تكلفة تشغيل وإدارة الشبكة. أيضًا عند تصميم الشبكة، يستلزم التركيز على الخدمات التي يقدمها مركز التحكم بالشبكة، بصرف النظر عن أنواع ومصنّعي المبدلات المستخدمة في الشبكة؛ وذلك بسبب أن الشبكة تعمل على بروتوكول (OpenFlow) المتّفَق عليه من قِبَل المصنّعين.

أما في الشبكات التقليدية: فيصعب تهيئتها وقت عملها، وذلك يتطلب إيقاف الشبكة وقت التهيئة. أيضاً على صعيد التحكم، فالشبكات التقليدية يجب تهيئة كل جهاز على حدة. ومن ناحية اكتشاف الأخطاء، فهذه معضلة الشبكات التقليدية؛ حيث أن اكتشاف الأخطاء يتطلب وقتاً طويلاً. وتوضح الصورة التالية كيف أنه من الممكن أن تتشارك أكثرُ من شبكةٍ مركزَ التحكم (CONTROL PLANE) وتبين الفرق بين معمارية شبكات الـ SDN ومعمارية الشبكات التقليدية.



معمارية شبكات الـ SDN:

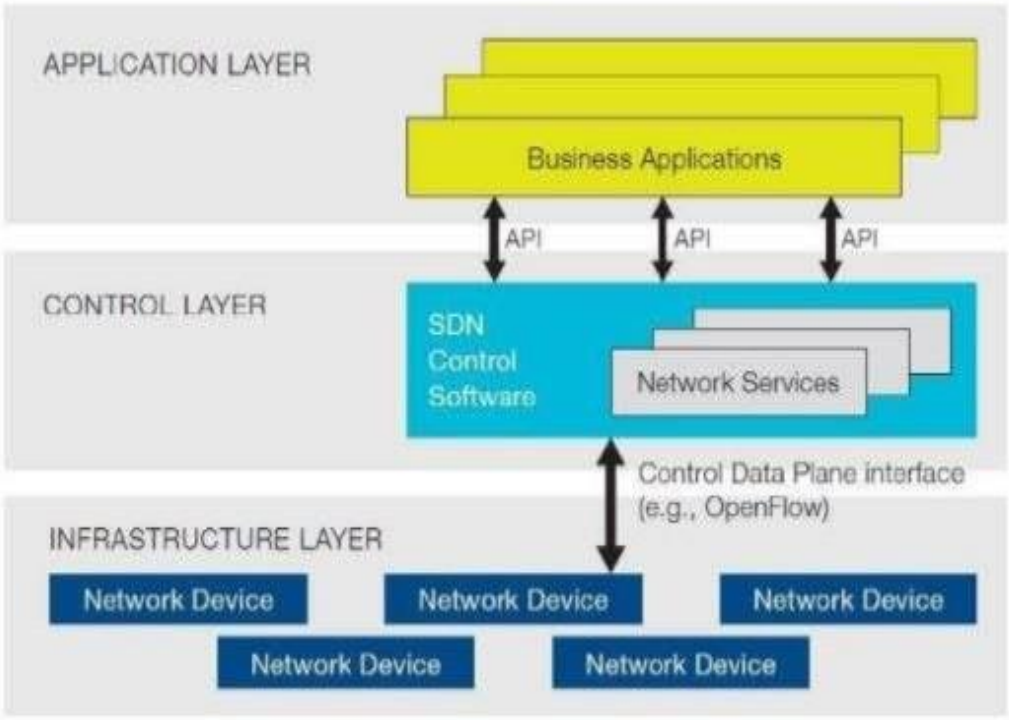
قامت منظمة الـ ONF (OPEN NETWORKING FOUNDATION) وهي منظمةٌ غير ربحية تدعمها بعض الشركات مثل غوغل فيسبوك ومايكروسفت وغيرها، تهدف إلى تطوير علم الشبكات عن طريق الـ SDN وبروتوكول الـ OPENFLOW الخاص بهذه التقنية، قامت بوضع معمارية شبكات الـ SDN إذ قامت بتقسيمها إلى ثلاث طبقات:

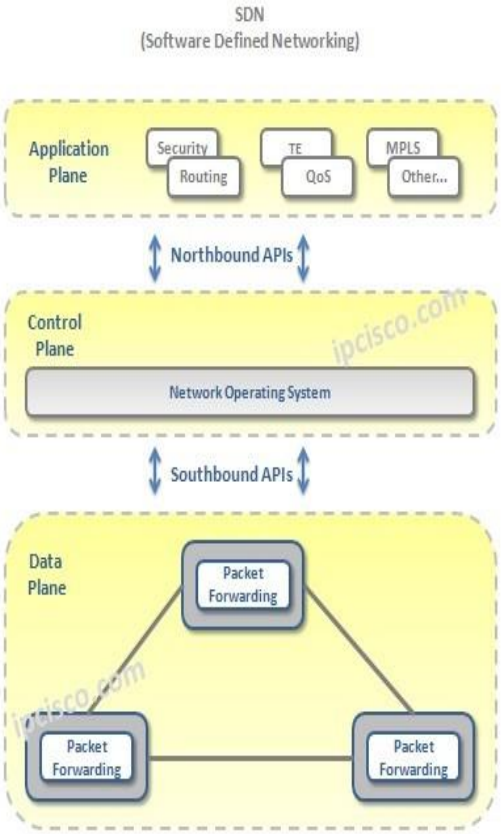
١- طبقة التطبيقات APPLICATION LAYER: وهي الطبقة الأولى في البنية المعمارية في شبكات الـ SDN وتتكون من الخدمات والتطبيقات التي تقدمها للمستخدم، وتتواصل مع الطبقة التي تليها عن طريق واجهة البرامج التطبيقية APIs (APPLICATION PROGRAMMING INTERFACES) وهي مشابهةٌ لطبقة التطبيقات في بنية الشبكات التقليدية.

٢- طبقة التحكم CONTROL LAYER: وهي الطبقة الثانية في شبكات الـ SDN وتمثل نقطة التحكم المركزية بالشبكة أو المتحكم (CONTROLLER)، إذ يقوم المتحكمُ بوظيفة التحكم والإدارة وإعطاء الأوامر لجميع أجهزة الشبكة من راوترات وسويتشات (SWITCHES&ROUTERS) ويعطيها صلاحية توجيه وتمرير البيانات فقط.

يُعتبر المتحكم هو أساس تقنية الـ SDN، إذ يقوم بوظيفة مركز التحكم في الشبكة (CONTROL PLANE) التي كانت تقوم بها أجهزة الشبكة من راوترات وسويتشات في الشبكات التقليدية بينما هو عبارة عن واجهة تطبيقية لكود برمجي في شبكات الـ SDN ويتم التواصل بينه وبين أجهزة الشبكة عن طريق بروتوكول خاص بتقنية الـ SDN هو بروتوكول الـ OPENFLOW.

٣- طبقة البنية التحتية INFRASTRUCTION LAYER: وهي الطبقة الثالثة والأخيرة في شبكات الـ SDN وتتكون من أجهزة الشبكة الوهمية والفيزيائية مثل المبدلات أو الموجهات (SWITCHES & ROUTERS) وكل هذه الأجهزة تدعم بروتوكول Open Flow لتنفيذ قواعد تسيير المعطيات.





لكي نتمكن من فهم ال SDN يمكن ان نقوم بتقسيم الشبكة الى ثلاثة اقسام :

١. Data plan switches

وهي الطبقة الدنيا من الراوترات التي تتصل بأجهزة المستخدمين النهائية وهي طبقة سريعة وبسيطة

- هذه الطبقة ستقوم فقط بتوجيه البيانات
- وهنا يكون flow table (routing table) الموجود داخل السويتشات (switches) منصب من قبل المتحكم controller .
- تتواصل هذه الطبقة مع المتحكم controller عن طريق بروتوكول open flow .

٢. Network control apps

تطبيقات التحكم بالشبكة والتي قد تكون

(routing –firewall-load balance...) وغيرها من الوظائف التي كانت تخصص للرواترات في الشبكات السابقة

- تعتبر هذه الطبقة هي دماغ التحكم التي تقوم بتنفيذ وظائف التحكم بواسط low level service (وهي API بتم توفيرها لها من قبل SDN controller)

- هذه الطبقة غير مقيدة مع المتحكم أو data plan ،أي يمكن تزويدها من قبل طرف ثالث والذي من الممكن أن يكون شركة أخرى مثلاً تقوم بتصميم هذه ال API وتنصيبها بشكل مباشر على الأجهزة

٣. المتحكم (SDN CONTROLLER).

: SDN Controller (network operating system)

* يعتبر المتحكم بمثابة نظام تشغيل للشبكة ، كانت كل الوظائف الموجودة في هذه الطبقة موجودة في الراوترات سابقاً، إلا أنه تم عزلها حاليا بطبقتين هما data plane و control plane .

* المتحكم يقوم بالمحافظة على المعلومات الخاصة بحالة الشبكة ، كما أنه يتفاعل مع تطبيقات الشبكة (network control application) من خلال ما يسمى ب

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

وكذلك يتفاعل مع ال switches الخاصة بالشبكة عبر ما يعرف ب

Southbound API (**واجهة برمجة التطبيقات الجنوبية : وهي التطبيقات التي تسمح للمتحكم بتحديد سلوك البنية التحتية لشبكة SDN. والتي هي طبقة تسيير المعطيات Layer Forwarding**).

* يتم تنفيذ المتحكم كنظام موزع في الشبكة لتحقيق أداء أفضل للشبكة ومقاومة الهجمات والأخطاء التي قد تحصل فيها .

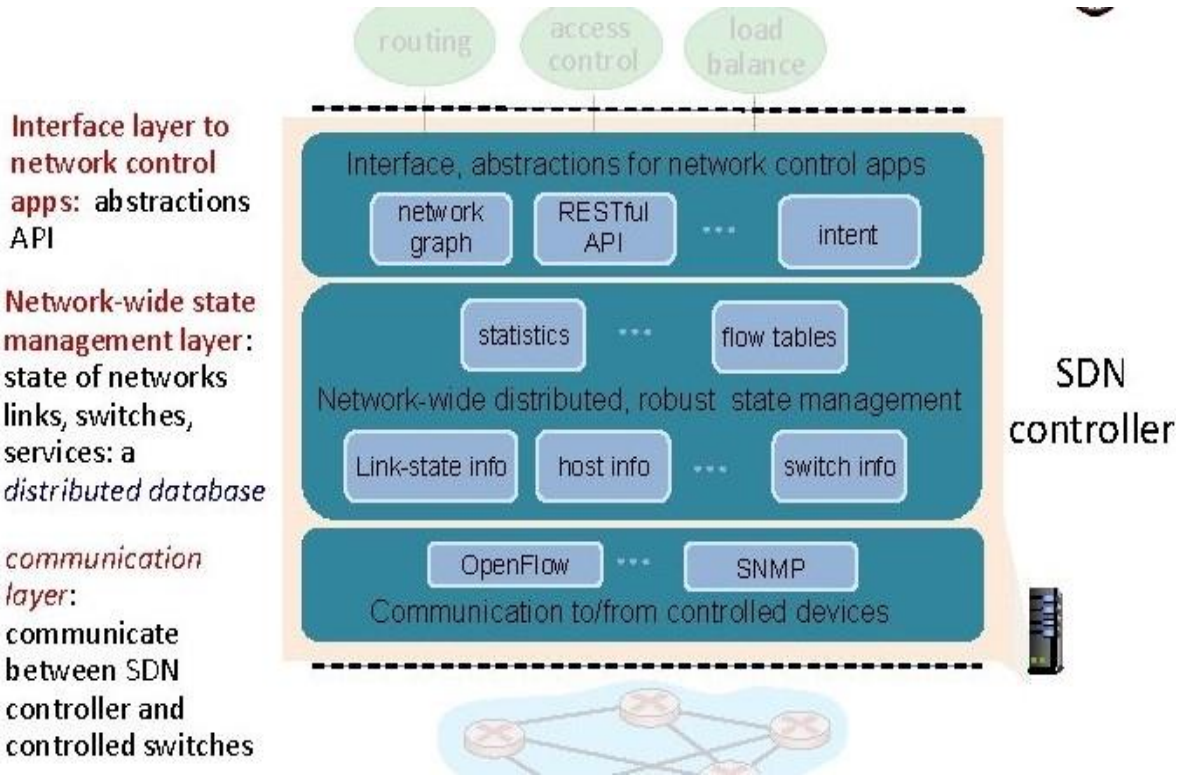
مكونات المتحكم (SDN controller components):

يقسم المتحكم الى ثلاث اقسام كالتالي :

طبقة الواجهة لتطبيقات التحكم في الشبكة: طبقة البرمجيات (API)

طبقة إدارة الحالة على مستوى الشبكة: حالة ارتباطات الشبكات والمفاتيح والخدمات

طبقة اتصال: التواصل بين وحدة تحكم SDN والمفاتيح الخاضعة للرقابة



أنواع المتحكمات :

يتألف التحقيق الأساسي لمعيار OpenFlow من تحقيق مبدل ومتحكم، يقوم المتحكم بإدارة عدد من المبدلات، هذا يسمح للمبدلات أن تتعلم أي منافذ للمبدلات وعناوين MAC تتصل بها وتقوم بتحميل القواعد لاحقاً تبادل البيانات بين هذه المبدلات، يتم كتابة المتحكم الخاص بلغة C++.

يعتبر المتحكم هو العصب المركزي للشبكة ونظام التشغيل المركزي يتوضع في هذا المتحكم، فهو المسؤول عن إدارة جداول التدفق والاتصال بين التطبيقات والأجهزة الشبكية وذلك باستخدام بروتوكول OF.

تصنف المتحكمات ضمن نوعين

١ -مفتوحة المصدر وبالتالي نسخة وحيدة من المتحكم.

٢ -تجارية وبالتالي قد تكون موزعة ومكونة من عدة نسخ.

تعتبر المتحكمات المفتوحة المصدر متاحة للباحثين وبالتالي توفر نسخة واحدة من المتحكم لكن مع إمكانية توفير عدد من الواجهات APIs لمنصات العمل لكي يتم إنجاز مهام محددة.، والاختلاف بين هذه المتحكمات هي بطريقة ولغة البرمجة المستخدمة لكتابه وتحقيق هذا المتحكم.

ونبين أهم المتحكمات ومنها:

1- المتحكم من نوع NOX

توم تطوير NOX في جامعة ستانفورد و تحقيقه بلغة C++ ،ولغة python ، كما تمت كتابته بطريقة سهلة بحيث يمكن الإضافة عليه و توسيعه من خلال مكونات صغيرة تكتب ب c++ و Python أيضاً. استخدم هذا المتحكم في هذه الجامعة ليجعل الباحثين قادرين بسهولة على اختبار أبحاثهم على الشبكات الحقيقية وبطريقة آمنه.

2-المتحكم Beacon

يتميز هذا المتحكم والذي يتم تحقيقه بلغة JAVA بالميزات التالية:

١- الاستقرار (Stability :)فمنذ تحقيقه عام ٢٠١٠ تم استخدامه بشكل واسع من قبل الكثير من الباحثين والمشاريع.

٢ -مفتوح المصدر وهو موثق بشهادة GPL v2.

٣ -أحد أسرع متحكمات SDN.

٤ -سريع التحميل والتشغيل وكونه يستخدم لغة JAVA فإنه يتميز بسهولة تنقيح الأخطاء واكتشافها

3-المتحكم Maestro

أيضاً كتب هذا المتحكم بلغة الجافا وطور ضمن جامعة Rice(أمريكا(، حيث يعتمد هذا المتحكم على البرمجة التفرعية قدر الإمكان للحصول على أداء أفضل، النسخة الحالية من هذا المتحكم هي النسخة الثانية v 0.2 يدعم Maestro حالياً فقط بناء الشبكات مع خوارزمية تعلم التبديل learning switch

4- تريما Trema

هي وحدة تحكم OpenFlowقابلة للتوسيع واسمها الأصلي هيليوس. تم بناؤه بواسطة NEC باللغتين Rubyوc وقد كانت محط اهتمام الباحثين بشكل أساسي.

5-NEC ProgrammableFlow

وحدة التحكم NEC الحائزة على جوائز من OpenFlow ™ .

NEC ProgrammableFlow كانت أول وحدة تحكم متوفرة تجاريًا في السوق ، تم تقديمها في Interop في مايو ٢٠١١ وتقدم في الإنتاج اليوم شبكات آمنة افتراضية ومتعددة المستأجرين. يوفر PF6800 ، القائم على السياسة ، والطوبولوجيا المادية المستقلة ، توجيهًا آليًا وقابلًا للتكيف مع تحكم متعدد المسارات والازدحام.

6- وحدة تحكم Lumina SDN

تم إطلاق Lumina SDN Controller 7.1.0في عام 2017 بواسطة Lumina، والتي تدعم النيتروجين(منصة opendaylight) يمكن للمستخدمين اختيار البروتوكولات والخدمات من خلال مساعدة هذه المنصات.

:POX -7

يعد من المتحكمات الهامة والشهيرة في SDN وكيف يحقق ذكاء الشبكة وإمكانية برمجته لينفذ أكثر من تطبيق مما يحقق مفهوم SDNومن ثم قياس إنتاجية وتأخير هذا المتحكم وبالتالي تتشكل لدينا معرفة كاملة حول مكونات تقنية SDN مما يساعدنا في اقتراح وتنفيذ عدة تطبيقات مختلفة.

:Plexxi -8

تعتمد أنظمة Plexxi على مفهوم شبكات التقارب ، وتقدم نوعًا مختلفًا قليلاً من وحدة التحكم - وهي خوارزمية تحسين إعادة توجيه الملكية ونظام التوزيع.

تتمثل الوظيفة الأساسية لوحدة التحكم في Plexxi في جمع معلومات حول الارتباطات ديناميكيًا من الأنظمة الخارجية أو بشكل ثابت عبر السياسات التي تم إنشاؤها يدويًا ثم ترجمة معلومات التقارب هذه إلى طبولوجيا إعادة التوجيه داخل شبكة Plexxi.

9- Cisco OnePK:

وحدة التحكم Cisco OnePK هي وحدة تحكم تجارية تجسد مفهوم الإطار من خلال دمج العديد من المكونات الإضافية لبروتوكول southbound ، بما في ذلك مكون إضافي لبروتوكول southbound غير عادي ، وهو Cisco OnePK API.

البنية عبارة عن إطار عمل OSGI قائم على Java يستخدم نموذج تخزين حالة في الذاكرة ويوفر واجهة REST ثنائية الاتجاه (مصادق عليها). يتم دعم التجميع باستخدام أدوات التنظيم والمعاملات Infinispan و JBoss .

10- Ryu:

Ryu هو إطار عمل مفتوح المصدر قائم على المكونات (مدعوم من NTT Labs) ويتم تنفيذه بالكامل في Python. تدعم خدمة المراسلة Ryu المكونات المطورة بلغات أخرى.

تتضمن المكونات دعم بروتوكول OpenFlow السلكي (حتى الإصدار ١.٣ من OF-wire بما في ذلك ملحقات Nicira) ، وإدارة الأحداث ، والرسائل ، وإدارة الحالة داخل الذاكرة ، وإدارة التطبيقات ، وخدمات البنية التحتية وسلسلة من المكتبات القابلة لإعادة الاستخدام (على سبيل المثال ، مكتبة NETCONF، sFlow / مكتبة Netflow).

بروتوكول OpenFlow

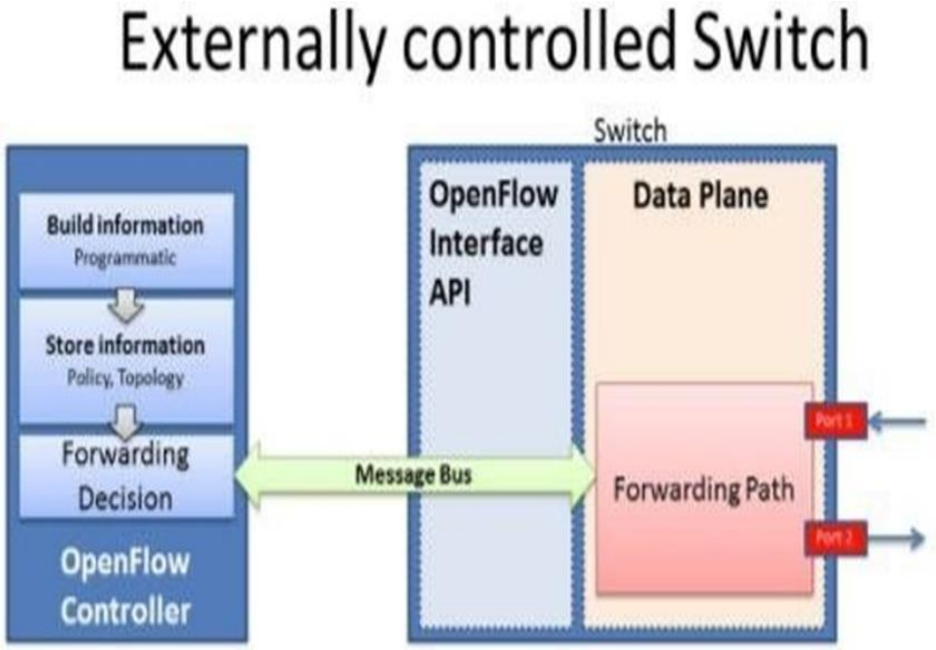
وهو بروتوكول التواصل بين طبقة التحكم (CONTROL LAYER) وطبقة البنية التحتية (INFRASTRUCTION LAYER) التي تليها، و يمكن بروتوكول **OpenFlow** أجهزة التحكم the control plane من التحكم عن بعد لتحديد مسار حزم الشبكة من خلال شبكة من أجهزة التوزيع Switches ينصح بتنصيب اثنين على الأقل من وحدات التحكم حيث الأول يعد أساسياً والثاني كجهاز احتياطي في حال تعثر الأول.

تعتبر **مؤسسة الشبكات المفتوحة** Open Network Foundation ، منظمة يقودها المستخدمون لتطوير وتبني الشبكات المعرفة بالبرمجيات، المسؤول عن وضع وإدارة المعايير لبروتوكول OpenFlow.

فقد تم اعتماد النسخة الأولى منه OPENFLOW V1.1 عام **2011** ثم تابعت منظمة الـ OFN تطويره حتى تم إطلاق النسخة الأخيرة منه OPENFLOW V1.5 في **2014**.

يُعتبر هذا البروتوكول التطبيقَ العملي لفكرة شبكات الـ SDN، ويرى عدد من الباحثين بأن شبكات الـ SDN وبروتوكول الـ OPENFLOW مفهوم واحد وهو من عائلة الـ TCP ويستخدِم البورت **6653**.

يعمل هذا البروتوكول في الطبقة الثانية في معمارية شبكات الـ SDN أما حسب توزيع الطبقات في الشبكات التقليدية نستطيع القول أنه يعمل في الطبقة الثالثة، وفي الحالتين تكمن وظيفته في نقل أوامر التوجيه وتحديد المسارات من طبقة التحكم (CONTROL PLANE CP) إلى طبقة البنية التحتية (DATA PLANE DP) وأيضاً نقل متطلبات البنية التحتية (DP) إلى مركز القيادة CP.

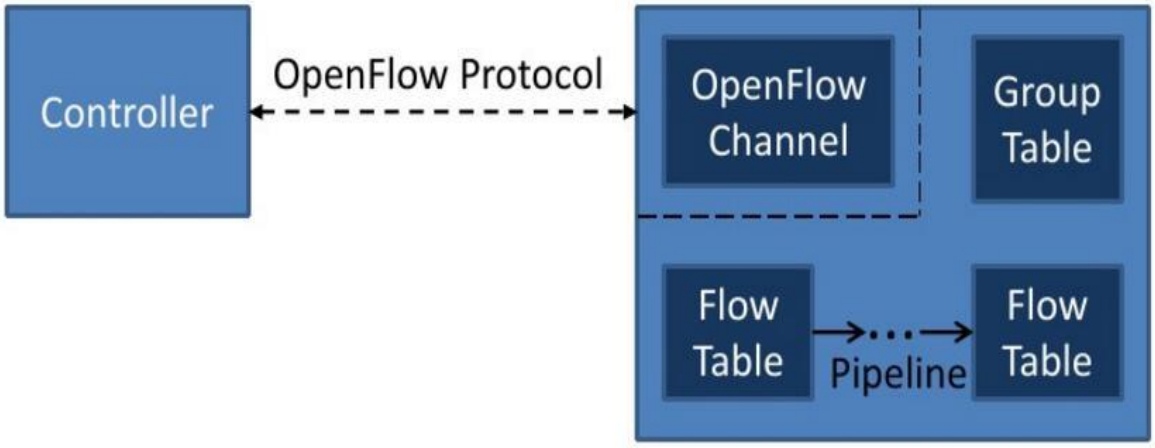


تحليل بروتوكولOpenFlow :

فكرة بروتوكول OpenFlow ، حتى لو كان جهاز الشبكة يحتفظ بـ FlowTable ويعالج الحزم فقط من خلال FlowTable ، فإن إنشاء FlowTable نفسه وصيانته وتسليمه يتم تنفيذه بالكامل بواسطة وحدة التحكم الخارجية.

بالإضافة إلى ذلك ، يقوم مفتاح OpenFlow بتحويل الرسائل التي يتم التحكم فيها بالكامل بواسطة المحول / الموجه في الشبكة التقليدية إلى البيانات التي يكملها المحول وجهاز التحكم.

عملية إعادة التوجيه ، وذلك لتحقيق فصل التحكم في توجيه البيانات وإعادة توجيهها. تقوم وحدة التحكم بتشغيل جدول التدفق في مفتاح OpenFlow من خلال واجهة محددة مسبقًا لتحقيق الغرض من إعادة توجيه البيانات.



مفتاح OpenFlow ، يشتمل على قناة آمنة وجدول تدفق متعدد المستويات وجدول مجموعة. من خلال القناة الآمنة ، يمكن لمفتاح OpenFlow إنشاء اتصال بناءً على بروتوكول OpenFlow مع وحدة التحكم ؛ يتم استخدام جدول التدفق لمطابقة الحزم التي يتلقاها مفتاح OpenFlow ؛ يتم استخدام جدول المجموعة لتحديد الإجراءات التي يحتاجها جدول التدفق

ينفذ بروتوكول OpenFlow بشكل أساسي التحكم في التوجيه بين وحدة التحكم والمحول من خلال معالجة أنواع مختلفة من الرسائل. في الوقت الحالي ، يدعم بروتوكول OpenFlow بشكل أساسي ثلاثة أنواع من الرسائل ، وهي عبارة عن وحدة تحكم إلى مفتاح ، ومتماثلة (رسالة متماثلة) وغير متزامنة (نوع رسالة غير متزامن). يتوافق كل نوع من أنواع الرسائل مع مجموعة متنوعة من الأحداث ، مثل حدث PacketIn الأكثر شيوعًا لدينا في أنواع الرسائل غير المتزامنة .

الرسائل الشائعة الاستخدام هي رسائل الترحيب والرسائل المميزة ورسائل الصدى و Packet_in و Packet_out و Flow_mod وما إلى ذلك

➢ سمات بروتوكول OpenFlow:

تعتمد كل من عمليات البحث، المطابقة، التسيير والطلبات القادمة من المتحكم على سمات وخصائص بروتوكول OpenFlow المعلن عنها من قبل مؤسسة Foundation Networking Open.

➢ القناة الآمنة:

تعد هذه القناة هي الممر الوحيد لعمليات التخاطب بين طبقة المعطيات وطبقة التحكم في الشبكات المعرفة برمجياً .

يتم تأسيس الاتصال في هذه القناة إما بروتوكول TIS (Security Layer Transport) أو TCP connection بين المتحكم والمبدل. في حال فقدان الاتصال، يحاول المبدل الاتصال بالمتحكم الاحتياط في حال وجوده .

في حال كان باستطاعة المبدل العمل في الشبكات المعرفة برمجياً والشبكات غير المعرفة برمجياً Non-SDN فإنه يدعى ب switch Hybrid، لكن في هذه الحالة لن يقوم المبدل باتباع بروتوكول OpenFlow وستفقد الشبكة بنيتها المعرفة برمجياً

تطبيقات SDN(SDN Applications):

تعتبر تقنية شبكات SDN تقنية واعدة في حل المشاكل والتحديثات التي تعاني منها خدمات الحوسبة السحابية حيث تستخدم بشكل واسع في:

١ - أبحاث الانترنت: تعد شبكة الانترنت شبكة في تطور مستمر، فإنه من الصعوبة أن يتم تحديث الأفكار الجديدة التي تحاول حل المشاكل التي تتعرض لها الشبكة، وبالتالي بسبب مركزية التحكم في تقنية SDN فإن ذلك يسهل اختبار الأفكار الجديدة قبل أن يتم تطبيقها على الشبكات في الواقع

٢ - توازن الحمل بتطبيقات المخدمات: لزيادة التوافرية والتوسعية في الشبكات الضخمة فإن ذلك يتطلب

توازناً بالحمل، وهذا يتم حالياً باستخدام عدد من المخدمات، و باستخدام SDN يتم توزيع الحمل بين هذه المخدمات ولكن هنا تظهر مشكلة التوسعية التي يجب كتابة تطبيقات خاصة بها في هذه الحالة

٣- شبكات مراكز البيانات: تعد شبكات مراكز البيانات مكونة من مخزن مركزي و الذي قد يكون فيزيائياً

أو افتراضياً، ويقوم بتشغيل العديد من المخدمات المضيفة والأجهزة الشبكية التي تعالج الطلبات

لمضيف آخر أو لشبكة انترنت عامة. حيث تستخدم بشكل واسع في الشبكات الكبيرة ،Google

Amazon ,Yahoo, Facebook وغيرها من الشركات والتي تتطلب وجود مراكز بيانات ضخمة

ومعقدة لكي تستقبل الطلبات وتعالجها بالسرعة القصوى، وبالتالي تعد SDN مناسبة جداً لهذه

المراكز.

٤ ـ التقنيات الافتراضية

منظمات كبيرة مثل Microsoft, Citrix ، تقوم بنشر مخدمات بتقنيات Virtualization و بالإضافة لذلك ترغب منظمات أخرى الآن بتقديم ميزات جديدة لبنيتهم التحتية التي تستخدم مفاهيم تقنية تعتمد على الافتراضية والتجريد، وبالتالي يجب أن يكون هناك تكامل وثيق بين المكونات الفيزيائية للبنية التحتية و المخدمات الافتراضية للشبكات

أدوات ولغات البرمجة فيSDN (SDN Tools and Languages)

هناك العديد من اللغات والأدوات التي استخدمت لمراقبة وتحقيق SDN ،الأبحاث الأولى والتمهيدية لهذه التقنية ركزت على منصات التشكيل وذلك من أجل تحقيق متحكمات الشبكة كنظام موزع من أجل إدارة الشبكة بشكل مرن والأبحاث اللاحقة اتجهت نحو تحقيق أدوات التنقيح الشبكية مثل veriflow] والذي يكون قادر على اكتشاف الأخطاء في تطبيقات الشبكة من خلال قواعد التوجيه ومنعها من التأثير على أداء الشبكة.

قام الباحثون بتطوير أداة لإختبار تطبيقات شبكة SDN وهي Mininet ،والتي هي عبارة عن برنامج يوفر بيئة لكتابة أي فكرة في SDN بلغة برمجية وفي حال كان هذا التحقيق البرمجي صحيحاً يتم نشر هذا التحقيق للاستخدام العام، لكن لازالت هذه الأداة تعاني من ضعف الأداء عند التحميل العالي.

تم تقديم لغة برمجية عالية المستوى هي frenetic والتي تتألف من لغة استعلامات بالاعتماد على

تعابير SQL ولغة المعالجة الانسيابية)Language Processing Stream ،(ولغة توصيفيه لمعالجة تدفق الرزمة وتوجيهها عبر العقد في الشبكة. تسهل Frenetic من عمل المبرمج من خلال إنتاج قواعد لتوجيه الرزم ضمن مستوى عال من التجريد.

تقوم هذه اللغه بتوصيف بعض من مهام البروتوكول OpenFlow والتي تكون عرضه لعدم التوافق بين تحميل القواعد في المبدل والسماح لبقية المهام بمعالجتها. بالإضافة إلى نقص التزامن بين زمن وصول الرزمة وزمن تحميل القاعدة. حيث تتألف من مستويين للتجريد: المعاملات على مستوى المصدر level-Source التي تتعامل مع حركة المرور في الشبكة، والنظام في الزمن الحقيقي المسؤول عن تحميل القاعدة الأمنية ضمن المبدل.

بالإضافة للغة frenetic التي يمكن برمجة شبكة OpenFlow فيها، تم تقديم عدد من لغات البرمجة عالية المستوى مثل Pocera ولغة Nettle ،والتي تعتمد على البرمجة التفاعلية الوظيفية (functional programming reactive)التي من شأنها تسهيل عملية الإدارة وتدعم الشبكات المقادة بالأحداث

نقاط ضعف الشبكات المعرفة برمجيا SDN:

١ـ مقدمة عن المشاكل الأمنية التي تعان منها شبكات SDN:

صحيح أن عملية فصل طبقة المعطيات ع طبقة التحكم أدت إلى قفزة نوعية في عالم الشبكات، لكن أدت هذه الخطوة إلى نشوء ثغرات جديدة لم تطن موجودة في الشبكات التقليدية، إلا أنه وخلال الفترة السابقة، بدأ الاهتمام بشكل أكبر وأصبح هنالك توجه حقيقي لمعالجة قضايا الأمن ولوثوقيه ضمن شبكات SDN، لذلك بدأت العديد من الأبحاث تتناول هذه الأخطار ونقاط الضعف والتهديدات التي نجمت عن هذه التقية الجديدة، وبدأت تقدم هذه الأبحاث بعض الحلول التي ينبغي أخذها بعين الاعتبار منذ الخطوة الأولى في بناء شبكة SDN، والتي قد تساهم بتفادي تلك التهديدات ومواجهتها وتخفيف خطرها.

تتمتع الشبكات التقليدية بمناعة طبيعية ضد الهجمات الشائعة نظراً لطبيعة أجهزتها المغلقة، تصميمها الثابت، تجانس البرمجيات والتحكم اللامركزي، فمثلاً إن استغل أحد المهاجمين نقطة ضعف للأجهزة المصنعة من قبل شركة ما، فإن الشبكة سوف تتأثر فقط في الجزء الذي يحوي عل أجهزة تابعة لنفس الشركة، أما باق الأجهزة فلن تتأثر على اعتبار أنها تتبع لشركات مصنعة أخرى. أما في شبكات SDN فوجود بروتوكول Flow Open 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 لزاما مناقشة قضايا الأمن والوثوقية and security والحلول الواجب أخذها بعين الاعتبار عند تصميم شبكة .

٢- التهديدات والأخطار التي تعان منها شبكات SDN:

تمتلك شبكات SDN سمتين أساسيتين تجعلها مصدر جذب للمهاجمين والمخترقين ومصدر قلق لأصحاب هذه الشبكات:

١. برمجة الشبكة باستخدام برمجيات software

٢. مركزية التحكم بالشبكة، بالتالي فإن الوصول لأحد المتحكمات يعني الوصول والتحكم بكامل الشبكة.

سنورد فيما يلي أهم التهديدات التي قد تواجهنا في شبكات SDN مع الحلول البسيطة المقترحة:

١. 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 كيرة من أجل تحقيق هجوم قطع الخدمة Dos والتي قد تكون مثلا ضد المبدلات الى تعمل ببروتوكول OpenFlow وذلك من أجل استهلاك جميع الذواكر TCAM(Ternary Memory Addressable content)الموجودة ضمن المبدل.

الحل المقترح: استخدام أنظمة كشف التسلل IDS مدعومة بأنظمة معرفة السبب الحقيقي للمشكلة وذلك لكشف السلوك الغير طبيعي لعناصر الشبكة، بالإضافة إل آليات من أجل التحكم الديناميكي بسلوك المبدل (مثلا: وضع حد معين لمعدل طلبات التحكم).

٢. الهجوم على نقاط الضعف الخاصة بالمبدلات: حيث أن الهجوم أو السيطرة عل مبل واحد ذلك يعني إمكانية؛ تجاهل طرد ما، إعادة توجيه طرد ما إلى وجهة خاطئة، نسخ طرود معينة، أو إبطاء تسيير الطرود ضمن الشبكة او حتى حقن معطيات أو طلبات وهمية في الشبكة وذلك لإسقاط المتحكمات 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.

٣. الهجوم على اتصالات طبقة التحكم: والذي قد يستخدم لإجراء هجوم Dos أو لسرقة معطيات. حتى لو تم استخدام تقنية التعمية TLS/SSI من أجل هكذا اتصالات، إلا أن

خطوط طبقة التحكم تبقى مهددة وخاصة وأن هنالك العديد من الأبحاث تشير إلى نقاط الضعف الخاصة بTLS/SSL حيث أنها تعتمد على بنية (PKI(Public Key infrastructure

لتبادل المفاتيح العامة. إن أمن هذه الاتصالات يكون قويا بقدر قوة أضعف خطوطها أو عناصرها، وقد يكون هذا الضعف ناجما عن شهادات موقعة ذاتيا، 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) أن يشن هجوم قطع خدمة موزع DDoS

الحل المقترح: هر تأمين الاتصال بين نسخ المتحكمات عن طريق تشفير هذه الطبقة. بالإضافة إلى ذلك يمكن استخدام آليات ديناميكية ومضمونة لربط الأجهزة وذلك لضمان الثقة بين أجهزة طبقة التحكم وأجهزة طبقة المعطيات.

٤. عدم وجود آليات لضمان الثقة بين المتحكمات وتطبيقات الإدارة: حيث على غرار التهديد رقم ٣، تفتقر المتحكمات والتطبيقات إلى القدرة على إقامة علاقات ثقة. يكمن الفرق الرئيسي عن التهديد المشار إليه في الطريقة التي تتشأ فيها الشهادة، حيث أن التقنيات المستخدمة للمصادقة على أجهة الشبكة تختلف عن تلك المستخدمة للتطبيقات.

الحل المقترح: استخدام آليات لإدارة الثقة ذاتيا تضمن الثقة بالتطبيقات طوال فترة عملها.

٥. الهجوم على نقاط ضعف محطات الإدارة: والتي عادة ما تكون موجودة ضمن الشبكات التقليدية، تستخدم هنا أيضا 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 التحكم يتطلب تفويض من قبل شخصين اثنين). أيضاً استخدام آليات استرداد مضمونة لضمان حالة موثوقة بعد إعادة التشغيل.

٦. عدم وجود مصادر موثوقة من أجل التوصيف و التعافي: والتي قد تسمح بفخم سبب المشكلة التي تم كشفها ومعالجتها للعودة بسرعة إلى الوضع الآمن. من أجل التحقيق والتثبت حول

حادث ما، نحن بحاجة إلى معلومات موثوقة من جميع الأجزاء والمجالات المكونة للشبكة. علامة عل 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 الى الحالة التي كانت تعمل بها

الحل المقترح: استخدام آليات التسجيل والتتبع، حيث أن هنالك حاجة لها في مستوى المعطيات والمستوى التحكمي ، ومع ذلك حتى تكون فعالة فإنها يجب أن لا تمحى أو أن تكون غير قابلة للتغيير. علاوة على ذلك يجب تخزين السجلات ضمن البيئات النائية والآمنة.

٧. الهجوم على نقاط الضعف في المتحكم: والتي ربما تكون أشد التهديدات خطورة في شبكات SDN. إن خلل وحدة تحكم واحدة أو إصابتها بهجوم خبيث، يمكن أن يسقط الشبكة بكاملها. كما أن استخدام نظام كشف التسلل IDS قد لا يكون كافيا، لأنه من الصعب إيجاد التجميعية الدقيقة للأحداث التي قد تؤدي إلى توليد سلوك معين، والشيء الأكثر أهمية هو معرفة أن هذا السلوك هو سلوك خبيث. بشكل مشابه أيضا، يمكن للتطبيقات الخبيثة في الشبكة أن تفعل ما يحلو لها في الشبكة على اعتبار أن المتحكمات فقط هي التي تزود الشبكة بالتجريدات التي تترجم إلى أوامر تحكمية إلى البنية التحتية

الحل المقترح: استخدام العديد من التقانات مثل التكرار أو النسخ (لكشف، إزالة، أو إخفاء السلوك الغير طبيعي) ، توظيف مسألة التنوع (للمتحكمات، البروتوكولات، لغات برمجة ..إلخ)، وإعادة التهيئة أو الاسترداد (التحديث الدوري للنظام للوصول إلى الحالة الموثوقة والسليمة). أيضا، من المهم تأمين كل العناصر الحساسة داخل المتحكم (مفاتيح التشفير مثلا). علاوة على ذلك، إن استخدام سياسات أمنية تضمن التطبيق الصحيح لسلوك تلك التقانات، وتقيد الواجهات التي يمكن استخدامها من قبل التطبيقات وماهي القواعد والأوامر التي تستطيع هذه التطبيقات توليدها لبرمجة الشبكة.

٣- الأمن والوثوقية ضمن شبكات SDN:

يناقش المؤلفون بعض الفاهم الأساسية عن الأمن والوثوقية ويقترحون بعض الآليات والتقيات التي يجب أخذها بعين الاعتبار عند تصميم منصة أمنة وموثوقة للتحكم ضمن شبكات SDN

٣-١ : 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،

فمثلا: لا يوجد آليات تستخدم 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، إن 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. أيضا يعتر إنشاء بنى تحمل التسلل -architecture lntrusion tolerant خطوة في طريق بناء نماذج الأمنية الذاتية، حيث أن 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 موثوقة وآمنة ضمن شبكات SDN:

٣-٢ : منصة التحكم الموثوقة والآمنة:

سوف نسرد أهم الطرق للحصول على منصة تحكم بشبكة SDN آمنة موثوقة :

١- التكرار أو النسخ المتماثل Replication:

تعد واحدة من أهم الآليات المستخدمة في تأمين الموثوقية في شبكات SDN، نسخ المتحكم إلى ثلاثة متحكمات، ويجب أن ننسخ التطبيقات أيضا. إلى جانب تكرار المتحكم إن هذه الإجرائية تضمن تحمل الأعطال في الجزأين الصلب والبرمجي سواء كانت الأعطال مقصودة أو عرضية، ويتيح استخدام النسخ إمكانية إخفاء الأعطال وعزل التطبيقات أو المتحكمات الخبيثة أو المعطلة.

٢- التنوع Diversity:

يعد 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 يحد من التأثير الكلي للهجمات عليها لأن هنالك هجمات تؤثر على نقاط ضعف معينة في نظام تشغيل ما دون أن تؤثر على نظام تشغيل ما أخر. في شبكات SDN يمكن تشغيل التطبيق نفسه لإدارة متحكمات مختلفة مثل تطبيقات API Northbound.

٣- آليات التعافي الذاتي mechanisms Self-Healing:

يمكن باستخدام الاسترداد التفاعلي أو الاستباقي (recovery reactive •OI proactive) 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 ممكنا، بمعنى أخر، يجب أن نتوخى التنوع في إجرائية الاسترداد وأن نعزز الدفاع ضد الهجمات التي تستهدف نقاط ضعف معين في النظام .

٤- ربط الأجهزة ديناميكيا Dynamic devices associated :

إذا كان لدينا مبدل مرتبط مع متحكم وحيد controller، أمكننا القول أن التحكم بهذا المبدل غير متسامح أو لا يتحمل الأعطال، لأنه وببساطة إذا سقط (تعطل) المتحكم فإن المبدل سوف يسقط حتما وبالتالي يصبح بحاجة إلى الارتباط بمتحكم أخر، لهذا السبب، يجب أن يمتلك المبدل القدرة على الارتباط بشكل ديناميكي بعدة متحكمات وبطريقة أمنة (مثلا: باستخدام التشفير وذلك لكشف المتحكمات الخبيئة والتحقق منها ولمواجهة أحد الأخطار الشهيرة وهو هجوم الرجل في الوسط. سيغدو المبدل المتصل بعدة متحكمات قادرا على التسامح مع الأخطاء بشكل ألي. يضمن هذا الإجراء مزايا أخرى وهي زيادة دفق المستوى التحكمي العديد من المتحكمات يمكن استخدامها في توزيع الحمل وتقليل تأخير التحكم باستخدام المتحكم ذو الاستجابة الأسرع.

٥- الثقة بين الأجهزة والمتحكمات:

تعد 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 يسبب تخفيض مستوى العلاقات الموثوقة. يمكن تطبيق مقاربة بسيطة وهي آلية التحقق من قائمة معرفة لأجهزة موثوقة (قائمة بيضاء)، محفوظة داخل المتحكم. إن الخيار الآخر هو الثقة بجميع المبدلات بشكل عام حتى الوصول إلى حالة ينغي فيها التحقق من مدى وثوقية مبدل 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 المبدلات أو المتحكمات بالاعتماد على خوارزميات معدة لكشف الأعطال. في حال تم فقدان الوثوقية بأحد المبدلات أو المتحكمات وأصبحت تحت عتبة معينة، يتم فورا تجميد هذا المبدل أو إجراء حجر صحي عليه بشكل أوتوماتيكي من قبل جميع الأجهزة والمتحكمات.

٦- الثقة بين التطبيقات وبرمجيات المتحكمات:

بما أن البرمجيات تعاني بحد ذاتها من مسائل: الاستخدام الطويل الأمد، الأعطال، الخلل والهجمات، كل هذه المسائل وغيرها أوجبت إيجاد نموذج ثقة ديناميكي. يدعم إدارة الثقة الذاتية في أنظمة البرمجيات

٧- المكونات الآمنة:

تعد المكونات الآمنة واحدة من أهم اللبنات الأساسية في بناء نظام آمن وموثوق. يمكن استخدام العناصر الأمنية لتوفير قواعد الحوسبة الموثوقة TCB لضمان خواص أمنية مثل السرية.

٨- المجالات الأمية:

المجالات الأمنية المعزولة هي نوع شائع من التقائات المستخدمة في مختلف أنواع الأنظمة. مثال على ذلك، في أنظمة التشغيل، لا يسمح 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

٩- التحديث والترميم السريع والموثوق للبرمجيات :

على اعتبار أنه لا يوجد أي برنامج خال من العيوب أو نقاط الضعف، فإن عملية التحديث والتصحيح السريعة والموثوقة للبرمجيات مهمة جدا لتقليل حجم نقاط الضعف. لذلك يجب توزيع منصات التحكم مع آليات تقوم بالتحديثات بطريقة سلسلة وآمنة.

بعض فوائد ال SDN :

١. أهم ما يميز شبكات الـ SDN عن الشبكات التقليدية أنها شبكات قابلة للبرمجة ومفتوحة المصدر، وهذا يعني من وجهة نظر المصممين ومهندسي الشبكات أن تقنية الـ SDN ساهمت في إيقاظ مستوى الإبداع على المستوى الفردي بحيث أصبح يستطيع المهندس أو المصمم إظهار إبداعاته على تصميمه لبنيته الشبكية بالإضافة لمستوى أمان أعلى وتلبية احتياجات أكثر، بمعنى أنَّ كل شركة تستطيع أن تبرمج شبكتها بما يتناسب مع أعمالها، مما يوفر الوقت والمال ويزيد من وثوقية الشبكة ويخفض معدل حصول الأخطاء وتحسين من أداء الشبكة ككل

٢. استخدام تقنية SDN يوفر المال ؛لأنك لن تحتاج إلى شراء switches و routers مغلقه البنية ، بل تقوم بشراء أجهزة شبيه ب routers و switches لتقوم فقط بالتنفيذ و تكون متاحه للدمج و التعديل عكس الراوترات القديمة التي تأتى مغلقة من الشركة المصنعة ولا تستطيع دمجها أو تعديلها بصوره كبيرة

٣. استخدام تقنية SDN يجعلنا نحصل على أداء أعلى من المعتاد حالياً و بصوره أرخص مادياً كثيراً مما هو عليه الان

٤. يتيح استخدام تقنية SDN سهوله تقسيم الشبكات و برمجتها و التحكم فيها و مراقبتها لإننا لا نتعامل مع كل switch وrouter كحاله منفصله بل نتمكن من خلال الوحدة المركزية (السيرفرات الجديدة) من التحكم في كل الراوترات و السويتشات في وقت واحد