الجمهورية العربية السورية
المعهد العالي للعلوم التطبيقية والتكنولوجيا
قسم المعلوميات

تقرير بحث

أعد لنيل درجة الماجستير في نظم دعم اتخاذ القرار

تقليل التنبيهات الخاطئة في نظم كشف الاختراق

تقديم

م. نوار اسماعيل

إشراف

د. محمد جنيدي **د. غسان سابا**

29/12/2016

المحتويات

الفصل الأول: ملخص.....5
الفصل الثاني: مقدمة.....8
1.1 مفهوم الهجوم (الاختراق).....9
1.2تصنيف نظم كشف الاختراق 10
1.3 البنية العامة لأنظمة كشف الاختراق: 12
1.4 توضع نظام كشف الاختراق على الشبكة.....14
1.5 الأخطاء الإيجابية 14
1.6 مثالين عن نظم كشف الاختراق 17
1.6.1 نظام كشف الاختراق المفتوح المصدر Snort.....17
1.6.2 نظام كشف الاختراق المخصص لحاسب معين CSSE.....20
1.7 خاتمة.....22
الفصل الثاني: تقنيات تقليل الخطأ الإيجابي.....23
2.1 المستوى الأول: تحسين النظام نفسه.....25
2.2 المستوى الثاني: استثمار البيئة 25
2.3 المستوى الثالث: معالجة للتنبيهات 25
2.4 المستوى الرابع إقحام المحلل (مسؤول أمن الشبكة).....26
الفصل الثالث: نبذة عن تعلم الآلة.....27
3.1 نماذج التعلم 28
3.1.1 التعلم الموجه.....28
3.1.2 التعلم نصف الموجه 28
3.1.3 التعلم غير الموجه.....29
3.2 التصنيف Classification.....29

3.2.1 القواعد التنبؤية 30
3.2.2 أشجار القرار 31
3.2.3 آلات متجهات الدعم support vector machines 32
3.2.4 الشبكات العصبونية 32
3.2.5 تقييم المصنفات 35
3.3 العنقدة clustering 37
3.3.1 خرائط التنظيم الذاتية Self-Organizing Maps SOM 37
الفصل الرابع: مجموعة البيانات المستخدمة 39
4.1 مجموعات البيانات المتاحة 40
4.2 مجموعات البيانات المحاكية 40
4.3 مجموعات مصائد مخترقي الشبكات 41
4.4 مجموعة معطيات "هجمات فقط" DEFCON 9 CTF 41
4.5 مجموعات البيانات الحقيقية 41
4.6 مجموعة البيانات DARPA1998 42
الفصل الخامس: أعمال متعلقة 43
الفصل السادس: التنفيذ العملي 47
6.1 تحضير بيئة العمل 48
6.1.1 تنصيب المكتبات الضرورية لعمل snort 49
6.1.2 تنصيب snort 49
6.1.3 تنصيب مجموعة القواعد اللازمة لكشف الاختراقات 49
6.1.4 تنصيب Barnyard2 50
6.2 استخلاص البيانات الأولية 50
6.3 وسم التنبيهات 52

6.4 ربط البروتوكولات.....54
6.5 فلترة التنبيهات.....55
6.6 المعالجة الأولية للتنبيهات 56
6.7 بناء المصنف وإجراء التجربة.....59
6.8 استخدام النظام.....66
الفصل السابع: خاتمة وآفاق مستقبلية.....67

الفصل الأول: ملخص

مـع التطـور الهائـل والمتسـارع فـي طـرق الاتصـالات الرقميـة ومـع ظهـور بروتوكـولات التشـفير والتواقيـع الرقميـة، أصـبحت نظـم كشـف الاختـراق تعتبـر خـط الدفاع الأخير لتأمين منظومـة الشبكة الحاسـوبية ومواردهـا. ولكـن المشـكلة الأساسـية فـي نظـم كشـف الاختـراق التجاريـة الشـائعة هـذه الأيـام هـي توليـد أعـداد هائلـة مـن التنبيهات الخاطئة، مما يجعلها مهمة معقدة وشبه مستحيلة لمراقب أمن الشبكة للتحقق من صحة هذه التنبيهات وصولاً لاتخاذ الاجراءات المناسبة. من هنـا بـرزت الحاجـة الملحة لاستكشاف مساحة البحث هذه لإيجاد حل مناسب لهذه المشكلة.

في هذه الأطروحـة، اخترنـا هذه المسـألة كمسـاحة أساسـية فـي بحثنـا. اختبرنـا فعاليـة اسـتخدام الشـبكات العصـبونية لتقليـل عـدد التنبيهـات الخاطئـة مـن خـرج نظـام كشف الاختـراق. اختبـرت هذه الطريقـة مـع أحد أكثـر نظـم كشـف الاختـراق مفتوحـة المصـدر شـهرةً والمسـمى SNORT، والـذي قمنـا بإعـداده ليعمـل بـنمط NIDS mode مـع حزمة معطيات طرود الشبكة DARPA98.

A long with Cryptographic protocols and digital signatures, Intrusion Detection Systems(IDS) are considered to be the last line of defense to secure a network. But the main problem with todays most popular commercial IDSs(Intrusion Detection System) is the generation of huge amount of false positive alerts along with the true positive alerts, which is a cumbersome task for the operator to investigate in order to initiate proper responses. So, there is a great demand to explore this area of research and to find out a feasible solution.

In this thesis, we have chosen this problem as our main area of research. We have tested the effectiveness of using the Neural Networks NN in order to reduce the number of false alerts from an IDS output. NN was tested with output of one of the most popular network based open source IDS, named Snort, which was configured to IDS mode to look for DARPA 1998 network traffic dataset. Using NN reduces (more than 99%) the number of false alarms.

الفصل الثاني: مقدمة

في هذا القسم سوف نقدم المفاهيم الأساسية في مجال أمن الحواسيب والشبكات، وسنعرف إجرائية كشف الاختراق.

لعب أمن الحواسيب والشبكات دائماً دوراً هاماً في مجال الحوسبة الإلكترونية؛ إلا أن الزيادة الهائلة في عدد من الأجهزة المتصلة بالشبكة واستخدام الإنترنت في التسعينات جنبا إلى جنب مع ازدياد عدد من الأنشطة غير المرخصة والزيادة في تعقيد سلوك المهاجمين جعلت منها واحدة من أهم مشاكل الحوسبة.

باختصار، يتعامل أمن الكمبيوتر مع حماية البيانات وموارد الحوسبة ويترافق عادةً مع الخصائص الثلاثة التالية:

السرية **Confidentiality** منع أي كشف عن البيانات غير مصرح به مقصود أو غير مقصود. على سبيل المثال، يعتبر سلوك متسلل يستعلم عن قاعدة بيانات بطاقات الائتمان للعملاء أو الحصول على ملكية شفرة المصدر خرقاً للسرية. لاحظ أنه عادة مثل هذا الانتهاك لا رجعة فيه، ولا يمكن أن تحل مشاكله بسهولة.

يمكن أيضا أن يفهم على مصطلح السرية في سياق أوسع، فيه أيضا نقوم بعدم تسليم الخدمات للمستخدمين غير المصرح لهم، حتى وإن كان هذا من شأنه أن لا يخرب السرية في حد ذاته.

السلامة **Integrity** منع التعديل على البيانات غير المصرح به مقصود أو غير مقصود. على سبيل المثال، مهاجم يقوم بتشويه خادم الويب للشركة أو تعديل محتوى قاعدة البيانات للبنك لتحقيق مكاسب شخصية هو هجوم ضد سلامة البيانات. لاحظ أن السلامة عادة يمكن استعادتها أو احتواء الهجوم عليها، على سبيل المثال، من مصادر أخرى مثل نسخ احتياطية، على الرغم من أن هذه العملية قد تكون مكلفة، وقد تستغرق وقتا طويلا وليس بالإمكان دائما استعادتها كاملة.

التوافرية **Availability:** منع الحجب غير المصرح به لموارد الحوسبة. ومن الأمثلة على التوافرية هجوم حجب الخدمة (DoS)، الذي فيه يقوم المهاجم بحجب موارد الحوسبة بحيث يصبح من غير الممكن للمستخدمين المخولين استخدامها، يمكن أن يشمل أيضاً سرقة المعدات المادية.

اعتماداً على التعاريف الثلاثة السابقة، بإمكاننا وضع تعريف للهجوم كالتالي:

1.1 مفهوم الهجوم (الاختراق)

يمكن تعريف الاختراق أو الهجوم بأنه سلسلة من العمليات المترابطة التي يقوم بها خصم مشبوه، والتي تؤدي بالنتيجة إلى تخريب النظام الهدف أو الحصول على

معلومات معينة منه غير مخول بالحصول عليها. هذه العمليات تشكل انتهاكاً واضحاً للسياسة الأمنية للنظام الهدف. السياسة الأمنية لمنظومة معينة تعرف أي من العمليات تلحق الضرر بهذه المنظومة وتلك التي يجب منعها للحفاظ على خصوصية النظام.

إن عملية التعريف والاستجابة للعمليات المشبوهة التي تستهدف منظومة ما تسمى كشف الاختراق، هي مقاربة مكملة لأمان النظام الضابط للتحكم بالولوج وعمليات التشفير. تُستخدم نظم كشف الاختراق لإدارة نظم الحواسيب المتضمنة المنصات الحاسوبية والشبكات وهي تقوم برفع تنبيهات عندما يتم ضبط أنشطة هجومية أو مشبوهة على هذه النظم.

قُدِّم مفهوم نظم كشف الاختراق أولاً من قبل أندرسون (2). وباتت تحظى باهتمام متزايد على مدى العشرون عاماً الأخيرة. نظم كشف الاختراق تهدف إلى كشف الاختراقات، بمعنى، مجموعة الأعمال التي تهدف إلى تخريب تكاملية أو إتاحية أو موثوقية نظام حاسوبي معين. إن التزايد الهائل في عدد الحواسيب المتصلة عبر الشبكة ولاحقاً استخدام الانترنت في أغلب المؤسسات كوسيلة للاتصال وتبادل المعلومات قاد إلى تزايد كبير في هذه الأنشطة الغير شرعية، ليس فقط من قبل مهاجمين من خارج نطاق الشبكة ولكن أيضاً من قبل أشخاص من داخل الشبكة غير مصرح لهم بالوصول لمعلومات معينة، مثل الموظفين المحتالين أو الأشخاص التي تسيء استخدام الصلاحيات الممنوحة لهم لتحقيق مكاسب شخصية، ولكن، في الجهة المقابلة، فإن نشر أعداد كبيرة من نظم كشف الاختراق له آثار سلبية ومشاكل عديدة بدأت بالظهور (8).

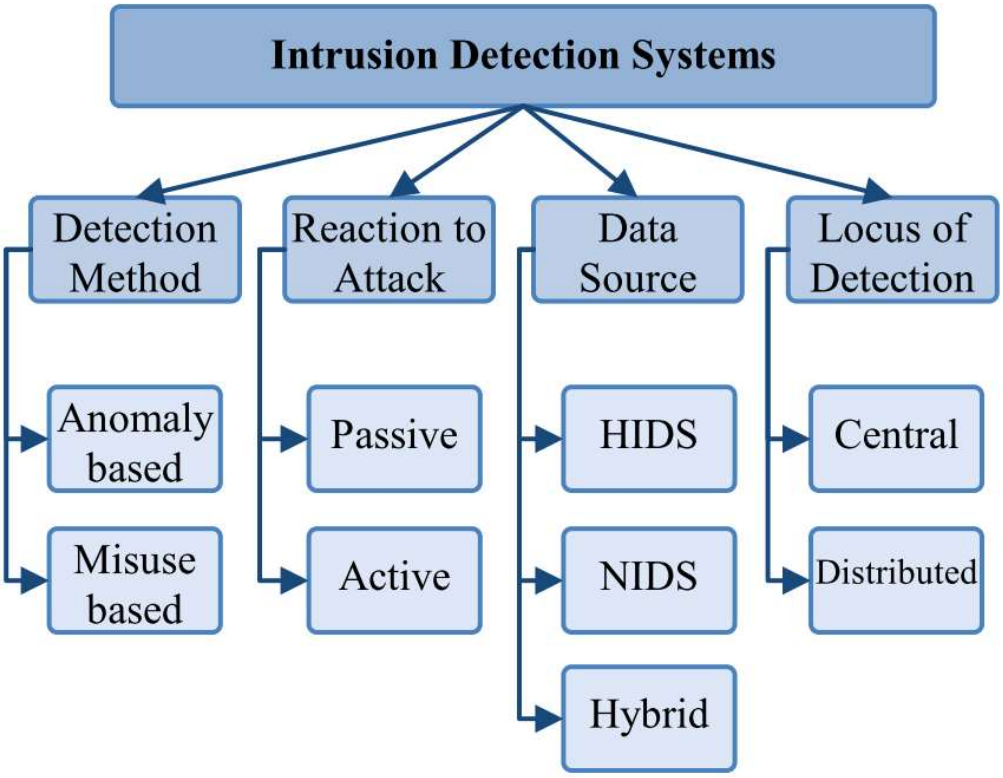
إن واحدة من أكثر المشاكل أهمية التي تواجه نظم كشف الاختراق اليوم هي ما يسمى **الأخطاء الايجابية** والتي تعني التنبيهات الخاطئة التي تصدرها وتتضمن قضايا متعلقة بأمن الشبكة وبحاجة إلى تدقيق وتحليل من قبل مسؤول الأمن في الشبكة. في الحقيقة تبين أن نسبة 99% من التنبيهات التي يصدرها نظام كشف الاختراق غير متعلقة بقضايا أمن الشبكة (8).

1.2 تصنيف نظم كشف الاختراق

يمكن تصنيف نظم كشف الاختراق من عدة وجهات نظر. أولاً من منظور طريقة التحقق (Detection Method)، يمكن تقسيم نظم كشف الاختراق إلى مجموعتين: تلك المعتمدة على النشاط الشاذ anomaly والمعتمدة على التوقيع signature based، بالنسبة لتلك المعتمدة على النشاط الشاذ فإنها تحاول تمييز الانزياحات عن النشاطات

الطبيعية وتبعاً لذلك يمكن تصنيف هذه النشاطات على أنها هجمات. في المقابل، فإن نظم كشف الاختراق المعتمدة على التوقيع تستخدم نماذج أو تعريفات معينة ومسبقة التعريف pre-defined patterns لهجمات معروفة وتقوم بمطابقتها مع النشاطات المشبوهة ليتم تصنيف هذه النشاطات على أنها هجمات، من مساوئ هذه النظم أنها بحاجة دائمة إلى تحديث قاعدة المعطيات الخاصة بها Patterns لكشف الهجمات الجديدة. أما سابقتها Anomaly based فمن مساوئها أنها تقوم بتصنيف الكثير من النشاطات الطبيعية على أنها هجمات وبالتالي فإنها تصدر العديد من التنبيهات الخاطئة False Alerts.

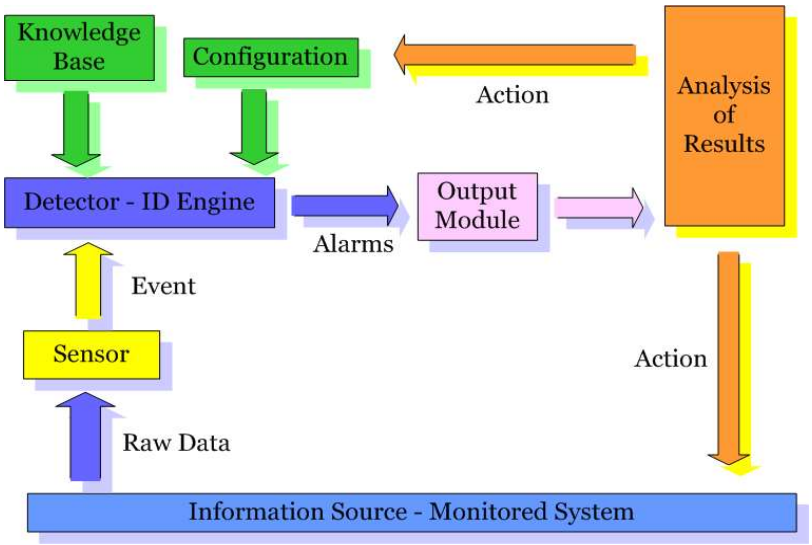
من الممكن أيضاً تصنيف نظم كشف الاختراق اعتماداً على مصدر المعلومات: من الممكن أن يكون نظام كشف الاختراق شبكي Network Based، أو نظم كشف اختراق لحاسب Host Based. إن نظم كشف الاختراق الشبكية تقوم بتحليل الأحداث أو النشاطات المرتبطة بالشبكة مثل حجم التراسل الجاري، عناوين الانترنت الجاري طلبها او الاستقبال منها، منافذ الخدمات..الخ. أما كشف اختراق لحاسب ما فإنها تقوم بتحليل أحداث أو نشاطات مثل الإجرائيات العاملة أو استدعاءات النظام، وبشكل أساسي تلك النشاطات المتعلقة بنظام التشغيل نفسه. ومن الممكن أيضاً تصنيف نظم كشف الاختراق من منظور ردة الفعل التي يقوم بها، حيث هناك ما يسمى نظم الاختراق السلبية Passive، والتي فيها تقوم حساسات الأمان بكشف الاختراق وإصدار تقارير بمعلومات هذه الاختراقات إلى واجهة المستخدم أو نافذة الأوامر، وفي المقابل هناك نظم كشف الاختراق النشطة أو Reactive وتسمى أحياناً نظم منع الاختراق IPS والتي تقوم بردة فعل اتوماتيكية عند حصول اختراق إما من خلال اعادة تهيئة الاتصال من جديد أو من خلال برمجة الجدار الناري Firewall لحظر هذا الاتصال. ويوجد تصنيف آخر لنظم كشف الاختراق يميزها من خلال تموضعها في الشبكة. يوضح الشكل 1.1 ماتم الاشارة اليه سابقاً من تصنيفات:



الشكل 1.1 تصنيف نظم كشف الاختراق

1.3 البنية العامة لأنظمة كشف الاختراق:

يوضح الشكل 1.2 البنية العامة والأقسام الوظيفية لنظم كشف الاختراق, سنقدم فيمايلي شرحاً مختصراً لكل قسم من الأقسام السابقة.



الشكل 1.2: البنية العامة لنظم كشف الاختراق.

- طبقة الحساس – Sensor :

يمثل الحساس الطبقة الأولى من نظم كشف الاختراق, ويقوم بتجميع المعطيات الضرورية لعمل النظام وتحويلها إلى شكل قابل للفهم من قبل طبقة الكشف. تختلف المصادر التي يقوم الحساس بتجميع المعطيات منها وتتنوع بحسب بنية النظام وآلية عمله, فقد تكون هذه المصادر هي الرزم المارة عبر الشبكة أو سجلات وملفات المراقبة لحاسب معين. يمكن في حالة شبكة كبيرة وموزعة استخدام أكثر من حساس لتغطية كامل الشبكة.

- طبقة الكشف – Detector:

تقوم طبقة الكشف بتحليل المعلومات الواردة من طبقة الحساس وتطبق عليها مجموعة من النماذج والمعايير المخزنة في قاعدة معطيات خاصة (Knowledge Base) بالإضافة إلى مجموعة القواعد والإعدادات (Configuration) التي يتم تزويد النظام بها, حيث تقوم بمقارنة السلوك الحالي للشبكة مع نماذج السلوك المخزنة في النظام, وفي حالة كشف اختراق فإنها تصدر إنذاراً إلى طبقة الخرج. ترتبط فعالية النظام وقدرته على الكشف الصحيح لمحاولات الاختراق بجودة النماذج التي يعرفها النظام وبالإعدادت التي يتم تزويده بها.

- طبقة الخرج – Output Module:

تتلقى هذه الطبقة الإنذارات من طبقة الكشف وتقوم بإعلام مدير النظام بحدوث اختراق وذلك بطرق متنوعة, فقد يتم إظهار المعلومات السابقة على الحاسوب الخاص بمدير النظام, أو يتم تخزينها ضمن قاعدة معطيات خاصة, أو يتم إرسالها على شكل بريد إلكتروني, أو يتم إرسال تنبيه (SNMP Trap) وغير ذلك من الطرق التي تكفل وصول المعلومات الخاصة بالاختراق إلى مدير النظام.

- قسم تحليل النتائج – Analysis of Results:

يتألف هذا القسم من مجموعة من المحللين الذين يقومون بدراسة وتحليل النتائج بالإضافة إلى تقييم أداء النظام, وفي ضوء النتائج التي يحصل عليها فريق المحللين يتم اتخاذ الإجراءات الكفيلة بتحسين نظام الحماية الخاص بالشبكة المعنية, فقد يتم تعديل قواعد جدار النار (Firewall Policies) أو تعديل قواعد ونماذج الكشف المستخدمة.

1.4 توضع نظام كشف الاختراق على الشبكة

يختلف المكان المناسب الذي يجب وضع نظام كشف الاختراق فيه بحسب نوع هذا النظام, ففي حالة النظم التي تعمل على حاسب مضيف (HIDS) يجب وضع النظام على كل الحواسب – المراد حمايتها – حتى يكون بمقدوره مراقبة وتحليل سجلات وملفات المراقبة.

أما نظم كشف الاختراق الشبكية (NIDS) فتختلف أماكن توضُّعها بحسب البنية الفيزيائية للشبكة وبحسب الاختراقات المراد الكشف عنها. فإذا كان المطلوب هو حماية الشبكة من الاختراقات الخارجية عندها يجب وضع النظام على منافذ الشبكة مع العالم الخارجي (الشبكات الخارجية المتصلة مع الشبكة المراد حمايتها), إما مع الموجه (Router) أو مع جدار النار (Firewall).

أما إذا كنا نريد حماية الشبكة من الاختراقات الداخلية فيجب وضع نظام كشف الاختراق على كل مقطع شبكي، أو فقط على المقاطع الحساسة.

1.5 الأخطاء الإيجابية

إن مشكلة الخطأ الإيجابي تعني أن التنبيهات التي يطلقها نظام كشف الاختراق عند ملاحظة نشاط مشبوه من الممكن أن يطلقها أو يولدها عندما يكون ذلك النشاط نشاطاً طبيعياً (بمعنى أن نظام كشف الاختراق في هذه الحالة يصدر تنبيهاً خاطئاً). إن التعامل مع التنبيهات من قبل نظام كشف الاختراق ليس بالأمر السهل، تقوم حساسات الأمان المستقلة بتوليد التنبيهات وإرسالها إلى الجزء المختص بتحليل التنبيهات الذي يقوم بتحليل طبيعة الهجوم ويحاول التقليل من الخطأ في التنبيهات. على الرغم من ذلك، يكون هناك الكثير من التنبيهات الخاطئة اعتماداً على الطريقة التي تتصرف بها حساسات الأمان. نورد فيما يلي بعض المصطلحات المتعلقة بمفهوم الهجوم والخطأ في مجال كشف الاختراق:

- **الخطأ السلبي** **False Negative FN** : عندما يكون هناك هجوم، ولا يصدر تنبيه أو إنذار بحدوث اختراق، فإن هذا يسمى **بالخطأ السلبي** **FN**، ويمكن أن يوصف الخطأ السلبي بأنه عدم قدرة نظام كشف الاختراق على كشف الاختراق.
- **الخطأ الإيجابي** **False Positive FP:** عندما لا يكون هناك أية هجوم ويصدر نظام كشف الاختراق إنذاراً بوجود هجوم.

إن نظم كشف الاختراق المعتمدة على النشاط الشاذ تصدر الكثير من الخطأ الإيجابي والقليل من الخطأ السلبي، بينما تلك المعتمدة على التوقيع تغفل الهجمات

الجديدة الغير محدثة تواقيعها ما يسبب الكثير من الأخطاء السلبية والقليل من الخطأ الإيجابي.

- **الصواب السلبي TN**: عندما لا يكون هناك أي هجوم ، ولا يصدر النظام أية تنبيهات.
- **الصواب الايجابيTP**: عندما يكون هناك هجوم ويصدر الـ IDS تنبيهاً بذلك.

إن الصواب الإيجابي والصواب السلبي هي أنشطة طبيعية وصحيحة يقوم بها نظام كشف الاختراق وهي بعكس الـ FN و FP الخطأ السلبي والإيجابي.

- False Positives Rate FPR أو ما يعرف **بمعدل الخطأ الايجابي**. يعرف بأنها عدد النشاطات الطبيعية أو العادية التي يقوم الـ IDS بتصنيفها على أساس أنها هجمات مقسوماً على عدد الأنشطة الطبيعية الكلية. إن معدل خطأ سلبي False FPR Negative Rate عالٍ يعني أداءً منخفضاً للـ IDS سيعرض النظام لخطر العديد من الاختراقات، لذلك للحصول على IDS فعال فإنه ينبغي أن تكون معدلات الخطأ الإيجابي FP والسلبي FN أصغر ما يمكن ومعدلات الصواب الإيجابي TN والإيجابي TP أكبر ما يمكن.

إن تقليل الخطأ الإيجابي تعد مسألة هامة في مجال الـ IDS ولكنها ليست كافية. إن بعض تقنيات تقليل الخطأ الإيجابي ستتسبب بإغفال الهجمات الحقيقية، لذلك تعتبر التقنيات الفعالة في تقليل الخطأ الإيجابي هي تلك التي تزيد دقة نظام كشف الاختراق أو تحافظ على الدقة القائمة. إن واحدة من أكثر المتطلبات أهمية في نظام كشف الاختراق هي أنه يجب أن يكون **فعال**، بمعنى، أن يقوم باكتشاف نسبة مهمة من الهجمات مع الحفاظ-في الوقت نفسه- على معدل التنبيهات الايجابية الخاطئة في مستوىً مقبول. وضوحاً، هذه المتطلبات يشوبها بعض التناقض، أي أنه كلما زادت نسبة كشف نظام كشف الاختراق للهجمات الجارية كلما زاد وبشكل ملحوظ معدل إصداره للتنبيهات الخاطئة.

إن بناء نظام كشف اختراق فعال يولد أعداد قليلة من التنبيهات الايجابية الخاطئة يعد مسألة صعبة للغاية. أسباب هذه الصعوبة تتضمن:

- **محدودية زمن التنفيذ**: في الكثير من الأحيان، تختلف أنشطة الهجمات الحقيقية على الشبكة قليلاً جداً عن الأنشطة الطبيعية، وفي بعض الأحيان فإن سياق هذه الأنشطة هو الدليل الوحيد الذي يحدد فيما إذا كانت هذه أنشطة هجمات أم لا. ونظراً لمتطلبات الوقت الحقيقي القاسية فإن نظام كشف الاختراق لا يستطيع تحليل جميع الأنشطة الجارية إلى المدى المطلوب(19).

- **خصوصية تواقيع الكشف:** إن كتابة تواقيع توصف أنماط الأنشطة الهجومية على الشبكة لنظم كشف الاختراق المعتمدة على التواقيع، تعتبر مسألة صعبة أيضاً (15). في بعض الأحيان، يكون من الصعب اعتماد ميزان واضح بين زيادة في التواقيع المحددة (الغير قادرة على اكتشاف هجمات جديدة) وبين زيادة في التواقيع العمومية (والتي ممكن أن توصف أنشطة طبيعية على أنها هجمات).
- **محدودية البيئة المعتمدة:** إن بعض الأنشطة التي يمكن أن تصنف هجمات في بيئة ما، ممكن أن تعتبر أنشطة طبيعية في بيئة أخرى. على سبيل المثال، إن إجراء مسح للحواسب الموجودة على الشبكة يعتبر نشاط غير شرعي ما لم يكون الشخص الذي يقوم بهذه العملية مخولاً للقيام بها. إن العديد من نظم كشف الاختراق الشائعة بنيت على معايير توصف أنشطة طبيعية على أنها غير مشروعة.
- **مغالطة المعدل القاعدي:** من وجهة نظر إحصائية بحتة، فإن حتى نظم كشف الاختراق ذات النسبة القليلة جداً من معدلات الخطأ الايجابي، لا تنتج معدلات كشف مرغوبة، وذلك لأن الهجمات الحقيقية على الشبكة هي أنشطة نادرة الحدوث.

باتباع مثال طرحه Axelsson (3)، بافتراض لدينا نظام كشف اختراق يحلل 1000000 طرد في اليوم تحتوي هذه الطرود على 20 طرد هجومي، هذا يعطي احتمالية الهجوم بـ$2*10^{-5}$. مع معرفة معدل كشق الحساس ($P(A \mid I)$) (احتمال إصدار تنبيه مع وجود هجوم). ومعدل الخطأ الايجابي فيه ($P(A \mid \neg I)$) (احتمال إصدار تنبيه مع عدم وجود هجوم)، نستطيع حساب معدل الكشف ($P(I \mid A)$) (احتمال وجود هجوم عند إصدار تنبيه) باستخدام النظرية البايزية، أو بكلمة أخرى احتمال أن يكون التنبيه الصادر يتضمن هجوم:

$$P(I \mid A) = \frac{P(I) \cdot P(A \mid I)}{P(I) \cdot P(A \mid I) + P(\neg I) \cdot P(A \mid \neg I)} = 0.66$$

باستخدام احتمال الهجوم أعلاه $P(I) = 2*10^{-5}$ وبافتراض معدل اكتشاف عال (غير حقيقي) $P(A \mid I) = 1.0$ ومعدل خطأ ايجابي منخفض جداً: $P(A \mid \neg I) = 10^{-5}$ نحصل على $P(I \mid A) = 0.66$. وهذا يعني أن أكثر من ثلث التنبيهات هي غير متعلقة بأنشطة هجومية.

1.6 مثالين عن نظم كشف الاختراق

في هذا القسم سوف نعرض اثنين من نظم كشف الاختراق: SNORT، الذي يعتبر واحد من أكثر نظم كشف الاختراق الشبكية مفتوحة المصدر شهرةً، والذي كثيرا ما يستخدم من قبل من الباحثين والمتدربين، وCSSE، نظام كشف اختراق موجه لحاسب معين متخصص في الدفاع ضد هجمات الحقن في التطبيقات.

1.6.1 نظام كشف الاختراق المفتوح المصدر Snort

يعد Snort نظام كشف اختراق شبكي يستخدم كمصدر لتدقيق طرود الشبكة الخام الملتقطة من خلال واجهة التقاط الطرود Pcap. ومثل كل نظم كشف الاختراق الشبكية، عند تشغيل Snort في بيئة شبكية تبدلية (Switched)، فإنه يجب وصله على البوابة من المبدل التي تُعكَس عليها (تمرر إليها) كل طرود الشبكة.

مبدأ تشغيل Snort بسيط جدا (21): هو يطبق قائمة مرتبة من القواعد (وتسمى تواقيع) على كل طرد يتلقاه. القاعدة الأولى التي تنطبق على الطرد تولد تنبيه. التوقيعات في Snort هي مجموعة من الشروط متعلقة بحقول حزم IP / TCP / UDP فضلا عن حمولة الحزمة. لمطابقة حمولة الطرود يوفر Snort مجموعة من التوابع تعطي لغة تعابير قوية، من بينها التعابير المنتظمة (مثلاً احتواء حمولة طرد ما على محارف معينة بصرف النظر عن ما قبلها وما بعدها) وتحليلات الحالة (أي أنه أثناء التوقع يكون بالإمكان الإشارة إلى الطرود الملتقطة والمحللة سابقاً) (18).

ومع ذلك، إذا تم تنفيذه (أي Snort) بهذه الطريقة سيكون فقط قادراً على كشف موثوق للهجمات في أدنى طبقة شبكة (بروتوكول IP)، في حين أن معظم الهجمات الحالية تحدث في الطبقات العليا: النقل (TCP / UDP) أو طبقات التطبيقات (على سبيل المثال ، HTTP، RPC). ولمعالجة ذلك، يستخدم Snort ما يسمى preprocessors التي تؤدي إعادة تجميع لحزم الشبكة والبروتوكولات عالية المستوى. يتم تجميعها نتيجة لهذا وتجهيزها في طرد افتراضي خاص تطبق عليه لائحة التوقيعات. على الرغم من أنها بسيطة من الناحية النظرية، لا ينبغي التقليل من التعقيد الحقيقي لمثل هذه المعالجة. لتوضيح هذا بمثال: **الكشف عن هجوم يستهدف مخدم الويب**، يتطلب ذلك من preprocessors تنفيذ العمليات التالية:

1. إعادة تجميع الطرود وتنظيمها على مستوى طبقة الـ IP.
2. محاكاة الحالة في البروتوكول TCP، وإعادة تجميع القناة.
3. التجميع والتنظيم وفك الترميز على مستوى بروتوكول الـ HTTP.

من الواضح أن ليس كل التنبيهات يمكن أن تتولد بشكل مستقل عن الحالة وفقط من خلال تطبيق لائحة قواعد على الطرد. يتم التعامل مع تلك الحالات، التي من الضروري فيها الحفاظ على الحالة، من خلال بعض الإضافات المتخصصة (مثل، كشف ARP-spoof أو كشف مسح البوابات).

```
Input: a real-time sequence of packets
Result: a sequence of alerts
1 parse the configuration files and rulesets
2 while (P=receivePacket()) != NULL do
3   |   for R ϵ getPreProcessorList() do
    |   /* may create a virtual packet */
4   |   (P,A) <- applyPreprocessor(R,P);
5   |   if A != NULL then
        |   /* report alerts generated by
        |   preprocessors */
6       |   reportAlert(A);
7       |   end
8   |   end
9   for S ϵ getOrderedSignatureList() do
10  |   if (A=signatureMatch(S,P)) != NULL then
11      |   reportAlert(A);
        |   /* report only the first match */
12      |   break;
13      |   end
14  |   end
15 end
```

خوارزمية الكشف في Snort (21)

لتوضيح هذا، نبين فيما يلي اثنين من التوقيعات (تم اختيارهم من بين أكثر من 4000 توقيع المتاحة حاليا في Snort). التوقيع الأول هو توقيع إساءة استخدام نموذجي، الكشف عن محاولة استغلال في الأداة AWStats (أداة مفتوحة المصدر لتحليل بيانات الويب وإصدار التقارير، ومناسبة لتحليل البيانات من خدمات الإنترنت مثل الويب، والبريد، والفيديو):

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
     (msg:"WEB-CGI awstats.pl configdir command
execution attempt";
flow:to_server,established; uricontent:"/awstats.pl?";
nocase;
uricontent:"configdir="; nocase;
pcre:"/awstats.pl?[^\r\n]*configdir=\x7C/Ui";
reference:bugtraq,12298; sid:3813; rev:2;)
```

توقيع إساءة استخدام الأداة AWstats في snort

لتفسير التوقيع، يتم تطبيقه على حركة مرور TCP الصادرة من الشبكة الخارجية والمرسلة إلى خوادم HTTP التي تعمل على البوابات المعتادة لخدمة الويب (من أشهرها {..80,81,3128,8000,8008,8080 etc}) وهي مسندة إلى المتحول HTTP_port في أي ملف إعدادات snort معياري . يتم تطبيق هذا التوقيع على التدفق المرسل إلى المخدم والذي يحتوي على سلسلة /awstats.pl وconfigdir = .

التوقيع هو عبارة عن تعبير منتظم يحتوي على السلسلة الأولى، ولا تليها محارف سطر جديد ويليها التعبير الثاني ويليه الرمز "|" (الرمز C7 في الترميز العالمي الموحد). في الواقع هذا المحرف الأخير من الممكن جدا أن يستغل نقطة ضعف، والسماح بتنفيذ تعليمة معينة (على سبيل المثال، /awstats.pl?configdir=|/bin/ls|). هذا التوقيع كله مهم لوضع محرف القناة هذا في سياقه السليم. يشير التوقيع إلى نقطة ضعف معروفة ب #12298 في قاعدة بيانات Bugtraq الصادرة عن معهد ماساشوستس للتكنولوجيا. نلاحظ أن Snort لا يمكنه في هذه المرحلة تحديد ما إذا كان الهجوم قد نجح أم لا. على سبيل المثال، إذا كان موقع Snort محمياً من خلال Snort نفسه، فإن أي مستخدم يحاول الوصول إلى العنوان http://www.snort.org/awstats?configdir=|/bin/ls| من شأنه أن يؤدي إلى إصدار تنبيه.

يمثل الحقل sid معرف القاعدة وهو رقم مميز للتوقيع والحقل rev عدد المراجعات التي تمت على القاعدة (التوقيع) .

إن إصدار هذه التنبيهات مفيداً من ناحية أن يكون المسؤول عن الموقع مهتماً في معرفة ما إذا كان شخص ما يحاول أن يجد ثغرة في ذلك.

توقيع آخر من نوع مختلف لا يكشف عن استغلال لنقاط الضعف المعروف، بل يقوم بالكشف عن الأنشطة التي ربما تكون مشبوهة. وبالتالي فإنه يسمى توقيع توقع القصد:

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433
(msg:"MS-SQL sp_adduser - database user creation";
flow:to_server,established;
content:"s|00|p|00|_|00|a|00|d|00|d|00|u|00|s|00|e|00|r|
00|"; nocase;
sid:685; rev:5;)
```

توقيع توقع قصد في snort

تقدح هذه القاعدة إذا حاول جهاز خارجي تنفيذ الإجراء المخزن sp_adduser في مخدم قاعدة بيانات MSSQL، مما يخلق سجل جديد في قاعدة بيانات المستخدمين. مع أنه لا يوجد هناك شيء خاطئ باستدعاء هذا الإجراء المخزن في حد ذاته، فقد لوحظ أن العديد من الهجمات التي شنت ضد خوادم SQL تستخدم هذه الإجراءات المخزنة للتحايل على آليات أمن قاعدة البيانات. إن مثل هذه التوقيعات تؤدي إلى إصدار تنبيهات ايجابية خاطئة كثيرة حيث أنه في كل مرة يستخدم مسؤول قاعدة البيانات (مثلاً) هذا الإجراء سوف يقوم snort بإصدار تنبيه.

1.6.2 نظام كشف الاختراق المخصص لحاسب معين CSSE

في السنوات الأخيرة شهدنا زيادة مطردة في الثغرات الأمنية على مستوى التطبيق، أي نقاط الضعف التي تؤثر على التطبيقات بدلا من نظام التشغيل أو أنظمة الكمبيوتر الوسيطة. ومن بين نقاط الضعف على مستوى التطبيق، تعد تهديدات التحقق من صحة المدخلات هي الفئة الأبرز في هذا المجال وتستحق اهتماماً خاصاً (14). بالإضافة لذلك (كما المثال في Snort أعلاه) إن هجمات استغلال نقاط الضعف هذه عادة يصعب اكتشافها من قبل نظام كشف الاختراق الشبكي.

في هجمات الحقن يوفر المهاجم دخل منحرف يحمل محتوى نحوي يغير دلالات تعبير في التطبيق. تكون النتائج معتمدة على التطبيق، وتؤدي عادة إلى تسرب معلومات، أو تجاوز للصلاحيات أو تنفيذ أوامر تعسفية.

Context-Sensitive String Evaluation (CSSE) هو نظام كشف ومنع اختراق (21) لحاسب ما مخصص لهجمات الحقن يستخدم بيئة مجهزة (مثل PHP أو Java)، بما يمكنه من الوصول إلى كل السياق اللازم لكشف، وبصورة أهم، لمنع حدوث مثل هذا الاختراق.

بتحليل سريع لتهديدات الحقن، نجد أن خاصية عامة فيها هي استخدام التمثيلات النصية للتعبيرات الناتجة من إدخالات معينة من المستخدم. في هجوم

الحقن، يمكن أن يؤثر دخل منحرف للمستخدم على سياق العمل، مما يؤدي إلى تغيير دلالات التعابير الناتجة. وسوف نشير إلى هذه العملية بخلط قنوات التحكم والبيانات.

إن تهديدات الحقن ليست نتيجة لاستخدام التمثيلات النصية نفسها، ولكن الطريقة التي يتم فيها بناء هذه التمثيلات. وعادة ما يتم سلسلة المتغيرات التي يولدها المستخدم في تمثيل نصي باستخدام عمليات السلاسل النصية (ربط السلاسل، الأقواس، علامات الترقيم). وبالتالي إن التسلسل المخصص لدخل المستخدم لخلق تمثيل نصي يعتبر هو السبب الجذر لهجمات الحقن.

عملية CSSE ان المنصة المدعومة بـ CSSE تضمن أن التعابير الناتجة من دخل المستخدم مقاومة لهجمات حقن عن طريق تطبيق الضوابط المناسبة تلقائيا على أجزاء التعبيرات المزودة من المستخدم. مضافاً إليها جزء المطور. CSSE يحقق ويميز بين الأجزاء المدخلة فيما إذا كانت من قبل مستخدم أو مقدمة من قبل مطور ، وثانياً: تحديد الضوابط المناسبة التي يتعين القيام بها على الأجزاء التي يدخلها المستخدم.

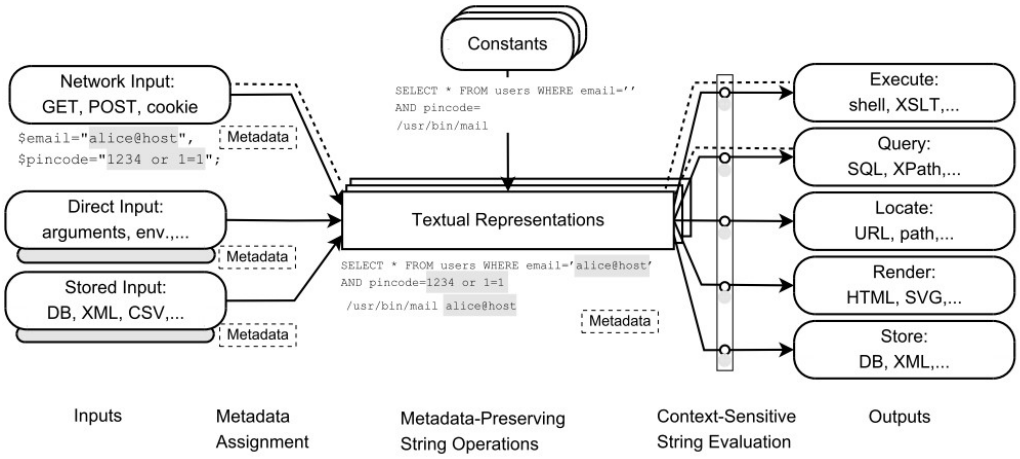
ويتحقق الشرط الأول عن طريق نظام تتبع يضيف البيانات الفوقية (أو معلومات التعريف) (metadata) لجميع أجزاء السلسلة في تطبيق ما من أجل تتبع أصل الأجزاء. الافتراض الأساسي هو أن أجزاء السلسلة القادمة من المطور موثوقة، عكس تلك القادمة من المدخلات التي يقدمها المستخدم، وبالتالي لا يمكن أن تحمل محتوى نحوي. معلومات التعريف تسمح لنا أن نميز بين دخل المطور ودخل المستخدم من التعبيرات الناتجة في أي مرحلة من مراحل إنشائها.

ويتحقق الشرط الثاني بتأجيل الاختبارات اللازمة لمرحلة متأخرة جداً، وهي لحظة استدعاء الواجهة البرمجية للتطبيق لتمرير التعبير الناتج للتطبيق. عند هذه النقطة، CSSE يعرف السياق الكامل للإجراء. معلومات تعريف التعبير النصي توفر الجزء الأول من السياق، والتي توصف أجزاء من تعبير الخرج التي تتطلب تدقيق. والجزء الثاني من السياق من خلال فحص الاستدعاء المعترَض لواجهة التطبيق البرمجية، والذي يحدد أي تحقق مطلوب سيتم تنفيذه. يقوم بعده CSSE باستخدام معلومات السياق هذه للتحقق من الأجزاء غير الآمنة للمحتوى النحوي. وفقاً للوضع الذي يستخدم CSSE فيه، فإنه يمكن إصدار تنبيه (كشف اختراق)، ومنع تنفيذ مضمون خطير (أي كشف الاختراق ومنعه بنفس الوقت).

تحقيق CSSE إن CSSE حالياً متاح كنموذج اختبار بحثي في مجال نظم كشف الاختراق لمنصات الـ PHP. تم تقييم النموذج الأولي مع نسخة قديمة ضعيفة الأمان من php، مع تطبيق ويب معروف مع نقاط ضعف أمنية معروفة والتحقق من أنه

قادر على كشف ومنع جميع هجمات حقن SQL المعروفة. ولأن CSSE كان مصمم دون معرفة هذه الهجمات، فإننا نتوقع أنه سيعطي أداءً جيداً مع التنفيذ الكامل مع التطبيقات الأخرى.

وعلاوة على ذلك فإن CSSE يتطلب زمن تنفيذ معقول وذاكرة معقولة(21). لوحظ في تجارب اختبار الـ CSSE أن كلفة زمن التنفيذ نادرا ما تجاوزت 10%، في حين ارتفع استهلاك الذاكرة بنسبة تقل عن 2%. هذا يظهر أن CSSE هو نظام كشف اختراق جيد للكشف عن هجمات الحقن.



الشكل 1.3 : عملية CSSE

1.7 خاتمة

في هذا القسم أعطينا مقدمة قصيرة لأمن الكمبيوتر في العموم وكشف الاختراق على وجه الخصوص. قدمنا مفاهيم الاختراقات وأنظمة كشف الاختراقات.

قدمنا أيضا مثالين على نظم كشف الاختراق بمزيد من التفاصيل: Snort، وهو نظام كشف اختراق شبكي مفتوح المصدر، وسوف نستخدمه في هذه الدراسة، وCSSE، نظام كشف اختراق لحاسب معين على درجة عالية من التخصص والقوة للكشف عن هجمات الحقن. لا نستخدم CSSE مباشرة في هذه الدراسة ولكنه يعطي مثالا على نظام كشف اختراق متخصص مع معدلات أخطاء إيجابية منخفضة للغاية ويظهر اتجاهات تطور نظم كشف الاختراق.

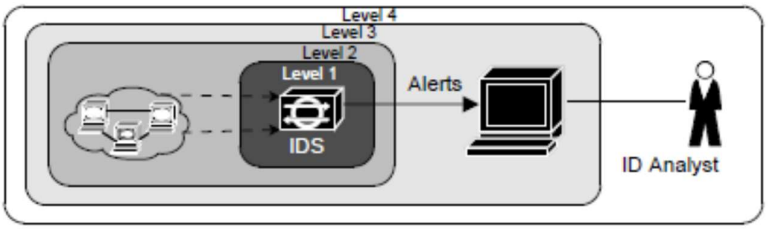
الفصل الثاني: تقنيات تقليل الخطأ الإيجابي

تم طرح العديد من الطرق لتقليل الخطأ الايجابي، وكل هذه الطرق يتم تصنيفها في إطار بندين، الأولى هي تلك الطرق التي تقوم بالمعالجة في مرحلة كشف الاختراق ونسميها تقنيات الكشف، أما الثانية فهي تلك التي تعمل على التنبيهات الصادرة بعد مرحلة الكشف ونسميها تقنيات معالجة التنبيهات.

الأبحاث المتعلقة بالمقاربة الأولى تطرح إعدادات مختلفة لنظام كشف الاختراق والعديد من طرق الكشف لتقليل الخطأ الايجابي مع توفير المزيد من الدقة في عملية الكشف. ولما كان من شبه المستحيل تجنب الخطأ الايجابي في نظم كشف الاختراق المعتمدة على النشاط الشاذ، فإن الهدف الرئيسي لمعظم الأبحاث هو زيادة معدل كشف النظام ودقته، من الواضح أن ذلك سيؤدي بالضرورة إلى تخفيف معدلات الخطأ، تلجأ أبحاث هذه الطريقة في الغالب إلى استخدام تقنيات التنقيب في المعطيات لتحسين معدلات الكشف وتقليل معدلات التنبيهات الخاطئة.

في المقابل، وبغرض تخفيف معدلات الخطأ الايجابي، تقوم المقاربة الثانية بإعدادات مختلفة لنظام كشف الاختراق من وجهة نظر معالجة التنبيهات المولدة أصلاً. معالجة التنبيهات تعد الحل الرئيسي لالتقاط التنبيهات ومعالجتها وتخفيف معدلات التنبيهات الخاطئة. خلال هذه الطريقة تعد تقنيات التنقيب في المعطيات أيضاً من أكثر التقنيات المستخدمة لتقليل التنبيهات ومعدلات الخطأ الإيجابي(1) . معالجة التنبيهات ممكن أيضاً أن تستخدم لأغراض أخرى مثل تقليل عدد التنبيهات بشكل عام والتنبيهات الخاطئة بشكل خاص.

يوضح الشكل 2.1 المستويات التي تصنف تقنيات تقليل الخطأ الإيجابي كما أشار إليها (Pietraszek T. , 2006) (18):



الشكل2.1: مستويات تقليل الخطأ الايجابي في نظم كشف الاختراق.

2.1 المستوى الأول: تحسين النظام نفسه

في البداية انصبّت معظم الجهود على بناء حساسات أفضل لنظم كشف الاختراق، بمعنى، حساسات تقوم بكشف هجمات أكثر أو حساسات مع معدلات خطأ ايجابي أقل. الحساسات تطورت من محركات مطابقة أنماط بسيطة جداً إلى أجهزة استشعار متخصصة و تفهم بروتوكولات النقل الأساسية وبعض تقنيات التشويش المحتملة (مثال: التجزيء على مستوى طبقة النقل TCP أو على مستوى طبقة الشبكة IP). و مع العدد المتزايد الهائل لنقاط الضعف على مستوى التطبيق، بدأت نظم كشف الاختراق بفاعلية بفهم بروتوكولات مختلفة على مستوى التطبيق (مثال على ذلك: HTTP,RPC). وفي الوقت نفسه، أصبحت لغات كتابة تواقيع الهجمات قوية على نحو متزايد، وتدعم التعابير الطبيعية (مثال SNORT) .

وعلى النقيض من بناء نظم كشف اختراق عامة، لجميع الأغراض، فإن نظم كشف الاختراق المتخصصة تقوم بالتركيز على أنواع محددة من الهجمات وتعد بمعدلات خطأ ايجابي منخفضة.

على سبيل المثال قدم (Sekar & Guang, 1999) (22) نظام كشف اختراق شبكي متخصص بهجمات المسح الشبكي الاستطلاعية وهجمات توقيف الخدمة (Denial of Service Attacks). طرح (Pietraszek، 2005) (17) نظام كشف اختراق متخصص بكشف هجمات حقن الـ sql على مستوى التطبيق. يجب أن ترفق نظم كشف الاختراق المحددة التوجه هذه بنظم كشف اختراق إضافية لتغطية الهجمات الأخرى على الشبكة، ما يستدعي الحاجة لبناء وخلق شبكة متخصصة لنظم كشف الاختراق الإضافية هذه.

2.2 المستوى الثاني: استثمار البيئة

إن نظم كشف الاختراق لديها رؤية محدودة عن البيئة الفاعلة عليها، وفي كثير من الأحيان، ليست لديها القدرة على التمييز بين الهجمات والأنشطة الطبيعية بدقة. باستخدام معلومات حول البيئة الفاعلة عليها (مثل معلومات عن نظم التشغيل) فإن نظم كشف الاختراق باستطاعتها فهم البيئة بشكل أفضل وتخفيض معدلات الخطأ الإيجابي.

2.3 المستوى الثالث: معالجة للتنبيهات

المقصود هنا استخدام التنبيهات المولدة من نظام كشف الاختراق كدخل ومحاولة تحسين جودة هذه التنبيهات ومعالجتها. وهذا يشمل أنظمة تستخدم استخراج البيانات data mining وما يسمى أنظمة ارتباط التنبيهات Alert Correlation Systems. على سبيل

المثال، أظهر K.julisch (8) كيف يمكن استخدام تحليل الأسباب الأصل للتنبيهات لاكتشاف مجموعة كبيرة من التنبيهات الايجابية الخاطئة وحذف ما يقارب 70% من التنبيهات الايجابية الخاطئة مستقبلاً. علاوة على ذلك، إن ارتباط التنبيهات، بالإضافة إلى التنبيهات الايجابية الخاطئة، تعنون مشكلة حقيقية لنظم كشف الاختراق، والمسماة بالتكرارية في سلسلة التنبيهات.

2.4 **المستوى الرابع إقحام المحلل (مسؤول أمن الشبكة)**

إن عدد قليل من الأنظمة الفاعلة في المستويات السابقة تستفيد من فرضية أن التنبيهات الصادرة من نظام كشف الاختراق سوف يتم تمريرها إلى متخصص التحليل ومسؤول الأمان على الشبكة ليتم تحليلها في زمن التنفيذ أو في زمن قصير (مع تأخير بسيط عن زمن التنفيذ).

في هذا البحث يتركز عملنا في المستوى الثالث ونقدم نموذجاً جديداً في استخدام الشبكات العصبونية للتقليل من عدد التنبيهات الايجابية الخاطئة في كشف الاختراق. وسنقوم ببناء مصنف للتنبيهات الصادرة عن نظام كشف الاختراق (أو مصنف للاختصار) لمساعدة المحلل (الإنسان) في تصنيف التنبيهات، الفكرة هي كالتالي: يقوم المصنف بتصنيف التنبيهات إلى صفوف معرفة مسبقاً ويقدم هذه التصنيفات إلى محلل كشف الاختراق. والمصنف الناتج يستخدم مجدداً لتصنيف التنبيهات الجديدة. تكرار هذه الإجرائية سيؤدي بالضرورة إلى تحسين دقة التصنيف للمصنف. يتطلب ذلك منا أن تستخدم خوارزمية التصنيف تمثيل معرفة صريح وواضح، مما يمكن المحلل من استخدامها عملياً والتحقق من صحتها.

الفصل الثالث: نبذة عن تعلم الآلة

إن تعلم الآلة (ML) يختص ببناء أنظمة تعمل على تحسين أداءها تلقائياً باستخدام خبرة مقدمة من المستثمر. إن التصنيف، المستخدم في هذه الأطروحة هو واحدة من المهام القياسية في تعلم الآلة.

تقنيات تعلم الآلة هي التقنيات التي تتعلم من ذوي الخبرة. ويهدف هذا التعريف العام جداً إلى تغطية معظم المهام التي تحلها مشاكل تعلم الآلة. هناك ثلاثة جوانب هامة من خبرة التدريب التي تؤثر بشكل كبير على اختيار طريقة التعلم.

أولا، مسألة ما إذا كان المتعلم يتلقى ردود الفعل المباشر أو غير المباشر فيما يتعلق بالاختيارات. على سبيل المثال، قد يُعلَم نظام تصنيف التنبيهات إلى تنبيهات الصواب والخطأ بالضبط ما هو التصنيف الصحيح. على العكس، في لعبة الداما، البرنامج اللاعب، يمكن أن يعطى فقط مجموعة من تسلسل حركاته ونتائج المباريات التي لعبت، وهي مهمة تعلم أكثر صعوبة. ثانياً، السؤال إلى أي درجة يمكن للمتعلم التحكم في سلسلة الأمثلة التدريبية يتم إعطاؤه إياها. ثالثا، السؤال عن مدى جودة توزيع أمثلة التدريب بحيث تمثل توزيع أمثلة النظام الحقيقية.

في هذه الأطروحة نتعامل مع أنظمة التعلم التي تتلقى ردود الفعل المباشر لاختيارات تم اعتمادها. في مثل هذا الإعداد، محور عملية التعلم هو مفهوم الحالات الموسومة، وهذا هو أزواج (i,c_i)، حيث i تمثل حالة مدروسة وc_i هو الصف الذي تنتمي إليه. اعتماداً على توافر حالات موصوفة نميز بين أربعة أنواع رئيسية للتعلم: التعلم الموجه، والتعلم نصف الموجه، التعلم النشط والتعلم غير الموجه. في التعلم الموجه، تستخدم مجموعة من الأمثلة الموسومة.

3.1 نماذج التعلم

3.1.1 التعلم الموجه

في نموذج التعلم الموجه، تستخدم طريقة تعلم الآلة مجموعة أمثلة من الأزواج (حالة- صف الحالة المعروف) لبناء المصنف. يستخدم هذا المصنف c في وقت لاحق لتصنيف الحالات الجديدة. هذا النوع من التعلم هو الأكثر شيوعاً.

3.1.2 التعلم نصف الموجه

في التعلم نصف الموجه، تستخدم طريقة تعلم الآلة مجموعتين للتعلم: مجموعة موسومة (معلمة)، مجموعة أزواج (حالة-صف الحالة معروف) ومجموعة حالات

غير معروف الصف الخاص بها. ومن الناحية المثالية، أداء مثل هذا المصنف هو أفضل من المصنف السابق الذي تم إنشاؤه باستخدام الحالات المعروفة فقط.

3.1.3 التعلم غير الموجه

وأخيراً، في التعلم غير الموجه تستخدم طريقة تعلم الآلة فقط مجموعة من الحالات غير معروفة التصنيف (غير موسومة) بهدف للكشف عن أنماط مثيرة للاهتمام في البيانات. الأمثلة على التعلم غير الموجه تتضمن التنقيب عن قواعد الربط والتحقق من المعطيات الشاذة.

افتراضات شائعة عادة ما تفترض أساليب تعلم الآلة الافتراضين التاليين، التي تبسيط تقنيات التعلم الآلي:

تمثيلات الواصفة-قيمة عادة ما تكون العينات ممثلة بصفوف في فضاء متعدد الأبعاد. الواصفات الشائعة هي واصفات تصنيفية (منفصلة وغير مرتبة، على سبيل المثال، أسماء البروتوكولات: "HTTP"، "TCP"، "UDP")، والواصفات العددية (على سبيل المثال، أرقام البوابات، عناوين الانترنت).

التوزيع المستقل والمماثل الاستدلال الإحصائي يفترض عادة أن الحالات مستقلة عن بعضها البعض، ووزعت بشكل مماثل. وهذا يعني أن كل حالة تنتمي لمجموعة العينات وضعت مع بعضها وفق توزيع احتمالي غير معروف، ولكن ثابت، والتصنيفات المعطاة للحالات كذلك الأمر غير معروفة ولكن ثابتة.

3.2 التصنيف Classification

المصنفات بالتعريف هي وظيفة إسناد علامة صف أو فئة من مجموعة محدودة من العلامات للعنصر الجاري تصنيفه. على سبيل المثال، لدينا معلومات عن مقالة إخبارية، المصنف قد يحدد الموضوع الذي يتعامل معه (اقتصادي، سياسي، الخ..). لدينا صورة لحرف أبجدي، المصنف قد يقرر أي حرف من الأبجدية. المصنفات يمكن بناؤها تلقائياً أو ممكن أن يتم بناءها من قبل الخبراء البشر. اعتمادا على بنيتها، المصنفات المبنية تلقائياً يمكن أن تكون قابلة للتفسير من قبل البشر (على سبيل المثال، قواعد التصنيف، أشجار القرار) أو لا تكون (على سبيل المثال، آلات متجهات الدعم support vector machines، والشبكات العصبونية Neural Networks). وهذه الأخيرة غالباً ما يشار إليها باسم مصنفات الصندوق الأسود. في هذه الدراسة نستخدم النوع الثاني.

من الطرق الموجودة لبناء المصنفات تلقائياً استخدام تقنيات التعلم الموجه، والتي عادة ما يتم تطبيقها على مرحلتين: مرحلة التعلم ومرحلة الاختبار. في مرحلة التعلم، يعطى المصنف مجموعة من الحالات مع علامات الصف الصحيحة، مما يسمح له بتعديل هيكله الداخلي. في مرحلة الاختبار، يتم تقديم المصنف مع حالات لم يسبق له التعامل معها، يقوم المصنف بتوقع الصف الذي تنتمي له هذه الحالات. مرحلة الاختبار هذه تسمح للمستخدم بتقييم أداء المصنف.

هناك العديد من مسائل التصنيف الثنائية، بمعنى أن المصنف يسند علامة صف واحدة من صفين ممكنين. في معظم هذه الحالات يقوم المصنف باختبارات الحالات اعتماداً على خصائص معينة (على سبيل المثال، البريد الإلكتروني هل هو بريد الإلكتروني شرعي أو لا؛ كون حدث أمني هجوم أو لا). دون فقدان العمومية، سوف نشير إلى الوسوم (العلامات) المعطاة من قبل المصنف الثنائي بأنها إيجابية، "+" (عند وجود الخاصة التي نسأل عنها) وسلبية، "-" (على خلاف ذلك). قد يبدو التصنيف الثنائي مقيداً قليلاً، ولكنه مفهوم بشكل جيد ومفيد في كثير من الحالات. نعرض في ما يلي أمثلة عن أهم التقنيات المستخدمة في مجال التصنيف:

3.2.1 القواعد التنبؤية

القواعد التنبؤية هي أنماط من النموذج اذا <مجموعة من الشروط> عندها <استنتاج>. الشروط الفردية في مجموعة الشروط هي الاختبارات المتعلقة بقيم المعاملات الفردية. بالنسبة للقواعد التنبؤية، فإن الاستنتاج يعطي التنبؤ لقيمة الصف المستهدف (متغير). القواعد التنبؤية يمكن أن تكون مرتبة أو غير مرتبة. تعتبر القواعد غير المرتبة مستقلة والعديد منهم قد تنطبق على عينة جديدة بحاجة للتصنيف. وهناك حاجة إلى آلية لتسوية الخلاف الحاصل عندما توصي قاعدتين إلى صفوف مختلفة لنفس العينة. مثل هذا القرار يستند عادة على تغطية القاعدة أو دقتها. تشكل القواعد المرتبة ما يسمى قائمة القرار. تعبر القواعد في تلك القائمة من أعلى إلى أسفل القائمة. يتم استخدام القاعدة الأولى التي ينطبق على العينة لتوقع قيمة الصف خاصتها. عادة ما يكون هناك قاعدة افتراضية يتم اتخاذها في حالة لم تنطبق أي قاعدة أخرى على العينة الجاري تصنيفها.

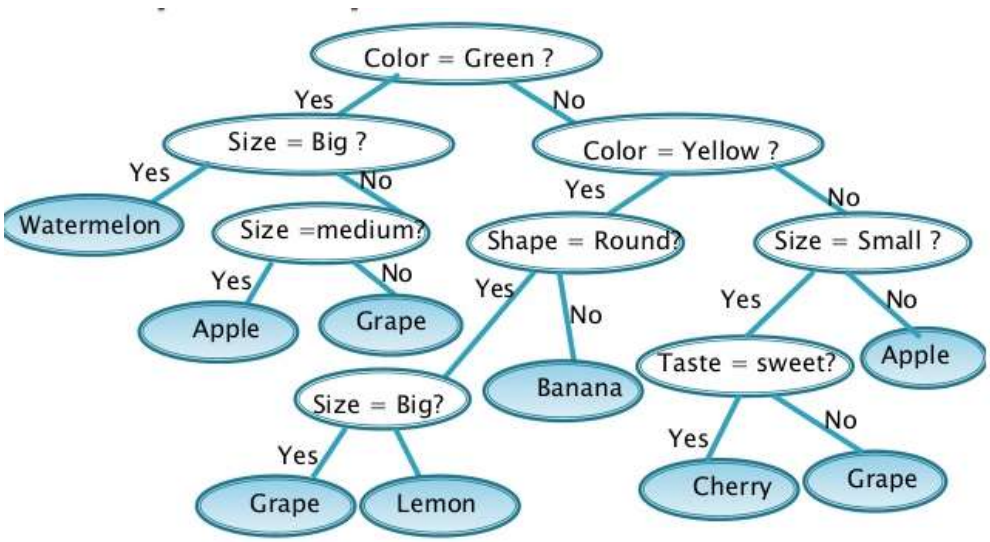
(physician-fee-freeze = y) and (synfuels-corporation-cutback = n) =>
Class=republican (138.0/3.0)
(physician-fee-freeze = y) and (export-administration-act-south-africa = y) =>
Class=republican (19.0/3.0)

متعلمي القواعد هي برامج لتعلم القواعد التنبؤية. تعتبر CN2 (16) واحدة من الخوارزميات الأولى في هذا المجال وأشهرها. الخوارزمية تنفذ منهجية تغطية على الأمثلة. تقوم الخوارزمية تكرارياً بإيجاد أفضل القواعد وفقا لبعض المعايير. القاعدة الأفضل تشمل بعض الأمثلة ذات التصنيف الإيجابية ولا تغطي أي من الأمثلة السلبية. ثم يتم إضافة القاعدة إلى الفرضية وتتم إزالة الأمثلة التي تشملها من المجموعة. انتهاء العملية تكون عندما لا يكون هناك مزيد من الأمثلة المشمولة.

3.2.2 أشجار القرار

التعلم باستخدام شجرة القرار هو واحد من الأساليب المستخدمة على نطاق واسع وتعتبر طريقة عملية للاستدلال الاستقرائي للتعلم عندما تكون العينات منفصلة بقيم دالة الهدف (12). التعلم باستخدام شجرة القرار قوي لبيانات يصحبها ضجيج. أشجار القرار يمكن أن يتم تمثيلها باستخدام مجموعة من قواعد إذا-افعل المقروءة والمفهومة من قبل الإنسان.

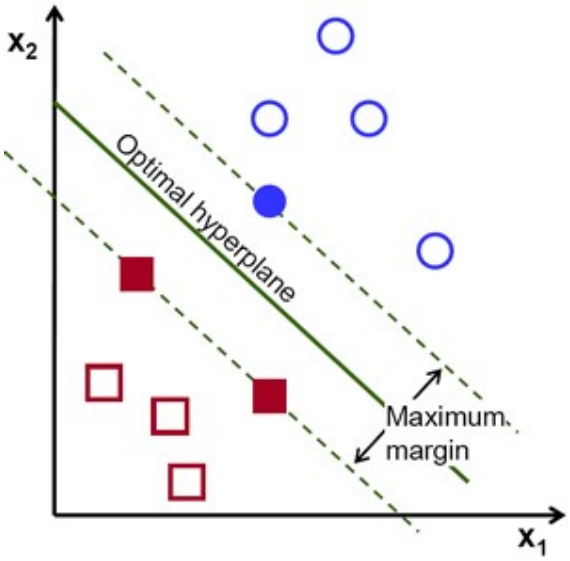
أشجار القرار تصنف الحالات عن طريق فرزهم من الجذر إلى بعض عقد الاوراق، والتي تنص على تصنيف عينة ما. كل عقدة في الشجرة تختبر بعض السمات للعينة وكل فرع متحدر من العقدة يحدد كل القيم الممكنة لهذه السمة. الأوراق تحتوي على قيمة الصف الذي تحاول شجرة القرار تقريبه. وبعبارة أخرى، تمثل شجرة القرار تباينات القيود على القيم سمات العينات، حيث كل مسار من الجذر إلى ورقة يتوافق مع قيمة واحدة من مجموعات السمات.



الشكل 3.1: مثال بسيط عن شجرة قرار

3.2.3 آلات متجهات الدعم support vector machines

تستخدم آلات متجهات الدعم SVMs ما يسمى ب "خدعة نواة" لتطبيق أساليب التصنيف الخطية على مشاكل التصنيف غير الخطية. SVMs تحاول فصل فئتين من نقاط البيانات في الفضاء متعدد الأبعاد باستخدام الفائق ذات الهامش الأقصى Hyperplane، أي، الفائق الذي لديه الحد الأقصى من المسافة إلى أقرب نقطة بيانات من كل الطبقات.



الشكل 3.2: الفائق ذات الهامش الأقصى في آلات متجهات الدعم

تعتبر مسألة تعليم SVMs مفهومة بشكل جيد. وهي مفيدة بشكل خاص لمجالات التطبيق مع عدد كبير من الأبعاد، مثل تصنيفات النصوص، والتعرف على الصور، المعلوماتية الحيوية أو التطبيقات الطبية. العيب في هذه الأساليب هو أن النماذج ليست مفهومة من قبل البشر.

3.2.4 الشبكات العصبونية

الشبكات العصبونية الاصطناعية Artificial Neural Network أو ما يدعى أيضاً بالشبكات العصبونية المحاكية simulated neural network أو : SNN مجموعة مترابطة من عصبونات افتراضية تنشئها برامج حاسوبية لتشابه عمل العصبون البيولوجي تستخدم النموذج الرياضي لمعالجة المعلومات بناء على الطريقة الاتصالية في الحوسبة .تتألف الشبكات العصبونية بشكل عام من عناصر معالجة بسيطة تقوم بعمل بسيط لكن السلوك الكلي للشبكة يتحدد من خلال

الاتصالات بين مختلف هذه العناصر التي تدعى هنا بالعصبونات ومؤشرات هذه العناصر element parameters. الإيحاء الأول بفكرة الشبكات العصبونية أتى من آلية عمل العصبونات الدماغية التي يمكن تشبيهها بشبكات بيولوجية كهربائية لمعالجة المعلومات الواردة إلى الدماغ. في هذه الشبكات اقترح Donald Hepp (7) أن المشبك العصبي يلعب دورا أساسياً في توجيه عملية المعالجة وهذا ما دفع للتفكير في فكرة الاتصالية والشبكات العصبونية الاصطناعية. تتألف الشبكات العصبونية الاصطناعية من عقد أو ما ذكرنا انه عصبونات neurons أو وحدات معالجة processing elements ، متصلة معا لتشكل شبكة من العقد، وكل اتصال بين هذه العقد يملك مجموعة من القيم تدعى الأوزان تسهم في تحديد القيم الناتجة عن كل عنصر معالجة بناء على القيم الداخلة لهذا العنصر.

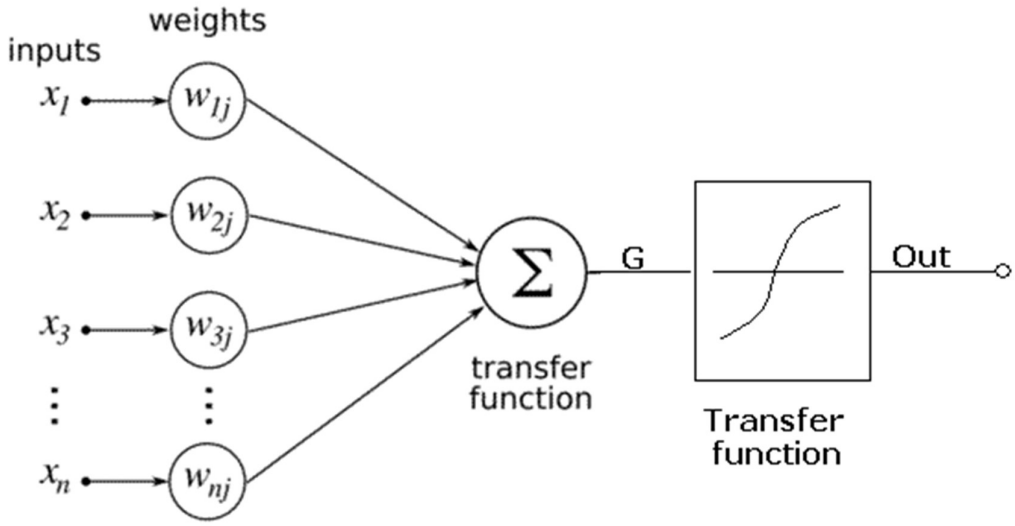
توفر الشبكات العصبونية طريقة عامة وعملية لتعلم قيم حقيقية، قيم منفصلة، متجهات. الشبكات العصبونية يمكن تطبيقها على نفس الفصول من المشاكل كما في شجرة القرارات، وقد أظهرت تحقيق دقة مماثلة.

أهم عيوب الشبكات العصبونية الاصطناعية هي أن هيكلها يجب أن يكون مناسبة لتطبيق معين (على سبيل المثال، من الصعب تصميم الشبكة العصبونية حيث يكون عدد ونوع المدخلات غير معروف) ويتطلب خبرة كبيرة في التدريب بشكل صحيح. أيضاً، تستغرق الشبكات العصبية وقت طويل للتعلم وبنيتها ليس قابلة للتفسير من قبل البشر. سنعرض في ما يلي شرح مفصل عن الشبكات العصبونية واستخدامنا لها في هذه الدراسة.

3.2.4.1 آلية عمل العصبون الاصطناعي

بشكل عام يمكننا أن نقول أن كل شبكة عصبية ترتب بشكل طبقات من الخلايا الاصطناعية : طبقة داخلية وطبقة خارجية وطبقات بينهم أو مخفية تتواجد بين طبقتي الدخل وطبقة الخارج. كل خلية في إحدى هذه الطبقات يتصل بكافة العصبونات الموجودة في الطبقة التي تليه وكافة العصبونات في الطبقة التي تسبقه. في البداية وجدوا إن الخلايا العصبية تقوم بعملية جمع لإشارات بمعني انه يوجد وصلتين لخلية عصبية مثلا وكل وصلة عليها إشارة تكون النتيجة هي محصلة الإشارات بالجمع العادي ومن ثم وجدوا أنّ كل عصبون يستطيع أن يقوم بعمل تكبير أو تصغير فتم إضافة عامل اسمه معامل الوزن (Weighting Factor) بمعني انه إذا كان هناك خلية مثلا ولها مدخلان فيتم ضرب الإشارة الأولي في المعامل الخاص

بالعصبون هذا وكذا للمدخل الثاني ومن ثم يتم جمعهم وعلي ذلك تم بناء النظام الهندسي للخلايا الصناعية.



الشكل 3.3 شكل مبسط لآلية عمل الشبكة العصبونية

3.2.4.2 طريقة معالجة المعلومات

كل اتصال بين عصبون وآخر يتميز بارتباطه بقيمة تدعى الوزن (Weighting) وهي تشكل مدى أهمية الارتباط بين هذين العنصرين، يقوم العصبون بضرب كل قيمة دخل واردة من عصبونات الطبقة السابقة بأوزان الاتصالات مع هذه العصبونات، من ثم جمع نواتج الضرب جميعا، ثم إخضاع النتيجة لتابع تحويل يختلف حسب نوع العصبون، ناتج تابع التحويل يعتبر خرج العصبون الذي ينقل إلى عصبونات الطبقة اللاحقة.

تم استيحاء آلية عمل العصبون الاصطناعي من عصبونات الدماغ : ففي العصبونات الحيوية، يمكن ان ننسب لكل مشبك اتصال قادم قيمة تدعى وزن المشبك weightتساعد هذه القيمة في نمذجة المشبك (عن طريق تحديد قيمته وأهميته) فالوزن يحدد قوة هذا المشبك وأثره في العصبون. يضرب وزن كل مشبك بالدخل القادم، ومن ثم تجمع نواتج الضرب لكل المشابك القادمة. عادة ما تكون العصبونات البيولوجية تابعة لقاعدة قيمة العتبة 'threshold value' فإذا كان المجموع الموزون weighted Sumلقيم الدخل أكبر من قيمة معينة تدعى العتبةthreshold يتفعّل العصبون مرسلاً إشارة كهربائية وتصل هذه الإشارة عن طريق تفرعات المحور إلى كل المشابك الخارجة outgoing synapses التي تتصل بعصبونات أخرى في الدماغ.

الشبكات العصبونية النموذجية تحاول أن تقلد هذا السلوك، فكل عقدة عصبونية تتلقى مجموعة من المدخلات عن طريق اتصالاتها بالعصبونات القبلية وكل عقدة لها تابع تفعيل activation function أو تابع تحويلtransfer function ، يحدد للعقدة متى وكيف تعمل أي لحظة وقيمة الخرج التي يجب أن تعطيها تماما كما العصبون البيولوجي. أبسط توابع التحويل هو تابع قيمة العتبة الذي يعمل العصبون على أساسه : معطيا قيمة 1 إذا كان المجموع الموزون لقيم الداخلي أكبر من قيمة معينة تدعى العتبة و 0 إذا كان المجموع الموزون اقل من العتبة. لكن توابع التحويل يمكن لها ان تأخذ أشكالا أخرى أكثر تعقيدا أهمها التابع الأسي, ولا تخلو شبكة من بعض عقد عصبية تملك تابع تحويل أسي، بشكل عام معظم توابع التحويل تحول قيمة المجموع الموزون لقيم الدخل إلى قيمة وحيدة محصورة في المجال [0,1].

من أهم أنواع الشبكات العصبونية : الشبكة العصبونية أمامية التغذية وهي مجموعة عقد عصبونية مرتبة بشكل طبقات. ترتبط هذه العصبونات مع بعضها عادة بحيث يرتبط كل عصبون في طبقة ما بجميع العصبونات في الطبقة التالية (لا ترتبط عصبونات نفس الطبقة مع بعضها).

الشكل النموذجي لهذه الشبكات هو ثلاث طبقات عصبونية على الأقل تدعى طبقة دخلinput layer ، طبقة مخفية hidden layer ، طبقة خرج output layer . طبقة الدخل لا تقوم بأي عملية معالجة فهي ببساطة مكان تغذية الشبكة بشعاع البيانات، تقوم طبقة الدخل بعد ذلك بتغذية (نقل المعلومات) الطبقة المخفية ومن ثم تقوم الطبقة المخفية بتغذية طبقة الخرج. المعالجة الفعلية للبيانات Data تتم في الطبقة المخفية وطبقة الخرج.

عندما يكون هناك عدد كاف من العصبونات، تكون الشبكة قادرة على التدرب trainingللقيام بأشياء مفيدة بالاستعانة بخوارزميات التدريبtraining algorithm ، تعتبر الشبكات أمامية التغذية مهمة جدا خاصة في استخدامات التصنيف الذكي والتمييز الذكي لبيانات غير مألوفة مسبقا وهو سبب اختيارنا لها في بناء مصنف التنبيهات الخاص بنا.

3.2.5 تقييم المصنفات

يوصف أداء مصنف k صف عادة باستخدام مصفوفة c بالأبعاد k*k، والمعروفة باسم مصفوفة الالتباس. الصفوف في c تمثل السمات الفعلية والأعمدة تمثل السمات المعطاة من قبل المصنف. يمثل العنصر c_{ij} عدد الحالات من ذات التصنيف الفعلي i صنفت من قبل النظام على أساس أنها من الصف j. من أجل مصنف ثنائي Cb يطلق

على عناصر المصفوفة ايجابيات حقيقية (TP)، سلبيات كاذبة (FP)، ايجابيات كاذبة (FP) والسلبيات حقيقية (TN) كما هو مبين في الجدول التالي. مجموع TP وFN هو مساو لعدد من الحالات الإيجابية (P). وبالمثل،ش فإن عدد الحالات السلبية (N) يساوي FP + TN.

C / A	+	-
+	0	C12
-	C21	0

صفوفة الالتباس

C / A	+	-
+	TP	FN
-	FP	TN

مصفوفة التكلفة

الشكل 3.4 مصفوفات الاتباس والتكلفة في تقييم المصنف الثنائي

العديد من مشاكل التصنيف في العالم الحقيقي هي غير متناظرة، مما يعني أن بعض الحالات المصنفة بشكل خاطئ هي أكثر "كلفة" من حالات أخرى. وهذا يمكن أن ينمذج عبر إعداد كلفة من خلال تقديم ما يسمى مصفوفة التكلفة Co متطابقة من حيث الصفوف والأعمدة مع مصفوفة الالتباس. القيمة Co_{ij} يمثل تكلفة تصنيف حالة ما من الصف i إلى صف j. في معظم الأحيان تكون كلفة التصنيف الصحيح هو صفر، أي $Co_{ii}=0$. في هذه الحالات، في المصنفات الثنائية، لا يوجد سوى قيمتين اثنتين في المصفوفة: c_{21} (تمثل تكلفة خطأ تصنيف تنبيه كاذب على أنه حقيقي) و c_{12} (تكلفة تكلفة خطأ تصنيف تنبيه صحيح على أنه كاذب). في الواقع، مثل مصفوفة التكلفة هذه تكون درجة حريتها هي واحد ، والنسبة بين القيمتين، تدعى نسبة التكلفة (CR). بالنسبة لكشف التسلل، تكون قيمة CR أصغر من واحد، من الأكثر بديهية أن نستخدم عكس هذه النسبة ICR، وهي معرفة على النحو التالي:

$$CR = c_{21}/c_{12} \qquad ICR = c_{12}/c_{21}$$

المصنفات في إعداد التكلفة يمكن وصفها من قبل التكلفة rc، وهو المجموع المثقل من التصنيفات الخاطئة مقسوما على عدد العينات المصنفة:

$$rc = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} C_{i,j} CO_{i,j}}{\sum_{i=1}^{k}\sum_{j=1}^{k} C_{i,j}}$$

والتي في الحالة الثنوية تصبح كمايلي:

$$rc = \frac{FN.C_{12} + FP.C_{21}}{TP + FN + FP + TN} = \frac{FN.C_{12} + FP.C_{21}}{N + T}$$

من الواضح أن المصنف الأمثل هو الذي تكون فيه القيمة rc مساوية للصفر وهو غير ممكن عملياً، لذلك فإنه علينا اختيار المصنف الذي يعطي أقل قيمة لـ rc.

في مجال تعلم الآلة يتم تقييم أداء المصنف عادة على مجموعة اختبار، وهي مستقلة عن مجموعة التدريب. في الحالات التي ينتج فيها أسلوب التعلم مجموعة من المصنفات نختار الأمثل، وعادة ما يتم استخدام ثلاث مجموعات مستقلة: مجموعة التدريب، لبناء المصنف. مجموعة الاختبار، لاختيار النموذج الأمثل. ومجموعة التحقق من الصحة، لتقييم المصنف. لأن مجموعة الاختبار والتحقق من الصحة مجموعات مستقلة عن بعضها البعض، ولأن مجموعة التحقق من الصحة لم يتم استخدامها في تحديد معالم المصنف، فإنه يمكن استخدامها لتقدير غير متحيز لأداء المصنف.

3.3 العنقدة clustering

العنقدة هي عملية تمييز وفرز أنواع مماثلة من الأشياء في مجموعات مماثلة، في مثل هذه الطريقة تكون الأغراض من نفس العنقود أكثر مشابهة لبعضها البعض من الأغراض من العناقيد الأخرى. يتم قياس التشابه بين العناقيد وفقا لمعلمة، تدعى"المسافة" قياس. انها خطوة مهمة في أي تجمع هو اختيار مقياس المسافة،والتي ستحدد كيفية حساب التشابه من عنصرين. ذلك يؤثر على ما سوف يكون شكل مجموعات، حيث أن بعض العناصر قد تكون قريبة من بعضها البعض وفقا لتابع مسافة معين، وبعيدة وفقاً لآخر. كمثال على العنقدة نذكر في ما يلي موجز عن خرائط التنظيم الذاتية كمثال عن التعلم غير الموجه.

3.3.1 خرائط التنظيم الذاتية Self-Organizing Maps SOM

خريطة التنظيم الذاتي (SOM) هي نوع من الشبكات العصبية الاصطناعية (ANN) التي يتم تدريب باستخدام التعلم غير الموجه لإنتاج فضاء قليل الأبعاد (عادة ثنائي الأبعاد)، وتمثل بيانات الإدخال بشكل منفصل بما يسمى خريطة. خرائط التنظيم الذاتي تختلف عن الشبكات العصبية الاصطناعية الأخرى من حيث تطبيق التعلم

التنافسي فيما يعني تصحيح خطأ التعلم، وبمعنى أنها تستخدم تابع جيران neighborhood funcion للحفاظ على الخصائص الطوبوغرافية لفضاء الدخل.

وهذا يجعل SOM مفيدة لعرض رؤية قليلة الأبعاد لبيانات عديدة الأبعاد، متجانسة ومتناسبة مع تلك البيانات المتعددة الأبعاد. تسمى أحيانا خريطة أو شبكة kohenen (10).

الفصـل الرابـع: مجموعـة البيانـات المستخدمة

في هذا الفصل نناقش مجموعات البيانات التي يمكن استخدامها في بناء وتقييم نظامنا وخصائصها. وهذا يعطي القارئ فهم أفضل لمجموعات البيانات المتاحة.

4.1 مجموعات البيانات المتاحة

ان نقص وجود مجموعات بيانات تمثيلية متاحة يعيق البحوث في مجال كشف الاختراق ويزيد من صعوبة المقارنة بين مختلف أنظمة كشف الاختراق والخوارزميات المستخدمة . ففي حين أنه من السهل أن تولد مجموعة كبيرة من تنبيهات كشف الاختراق (على سبيل المثال، عن طريق تشغيل نظام كشف اختراق في شبكة خاصة أو معرضة للإنترنت)، ان مثل هذه الطريق يشكل مشكلتين رئيسيتين: أولاً، ان مثل مجموعات البيانات هذه غير موسومة (أي أنه ليس من الواضح أي من الهجمات هي أخطاء ايجابية وأي منها هي صوابات سلبية)، ووضع العلامات الخاصة بهذه المسألة يشكل تحديا كبيرا. إلا إذا قام به العديد من المحللين المستقلين، ومثل هذا الوسم يمكن التشكيك فيه وخاضع للمناقشة. المشكلة الثانية هي ذات طبيعة مختلفة. في معظم الحالات التي تجمع فيها البيانات في نظم كشف الاختراق تكون ذات طبيعة سرية. تتضمن هذه البيانات بطبيعتها معلومات حول طوبولوجيا الشبكة، والحواسب الفاعلة وغيرها من المعلومات السرية (على سبيل المثال، محتوى رسائل البريد الإلكتروني، المواقع التي تمت زيارتها
أو حتى كلمات المرور في نص واضح). وبالتالي، فإن الوصول إلى هذه البيانات محدود للغاية ولا يمكن مشاركتها مع الآخرين.

4.2 مجموعات البيانات المحاكية

كانت هناك محاولة لتوفير مجموعة بيانات (DARPA 1998, DARPA 1999) متاحة للجمهور تم توليدها في بيئة محاكاة، ولكن كان فيها العديد من العيوب سواء في بيئة المحاكاة أو في إجراءات التقييم (6). مع ذلك، هذه المجموعات هي مجموعات البيانات الوحيدة -كما يدعي (Pietraszek,2006) (18) - المتاحة والمفيدة في تقييم نظم كشف الاختراق. في الواقع كان هناك العديد من الأبحاث باستخدام مجموعات البيانات هذه سواء في كشف الاختراق أو في مجال تعلم الآلة .

4.3 مجموعات مصائد مخترقي الشبكات

هنالك نهج آخر مستخدم وهو استخدام البيانات التي تم جمعها من قبل ما يسمى مصائد مخترقي الشبكات(4)، وهي أنظمة تنشر خدمات شبكية وهمية (أو حقيقية) بغرض التعرض للهجوم. أي وصول لهذه الخدمات تتم مراقبته والبيانات التي بتم جمعها يمكن أن تستخدم في وقت لاحق لاستخراج البيانات وتحليلها، من أجل فهم أنماط وتقنيات الاختراق الجديدة. البيانات بالتعريف تكون غير طبيعية ومتضمنة هجمات أو محاولات هجوم ووصول إلى أنظمة طبيعية بطرق غير مشروعة في الاتصال. وبالتالي، فإن البيانات هذه عادة لا تحتوي على معلومات حساسة ويمكن مشاركتها بسهولة. ولكن نظرا لطبيعتها، فإن بيانات مصيدة مخترقي الشبكات تكون أكثر فائدة للكشف عن الهجمات الآلية مثل الديدان أو الحواسب المصابة بالأخطار، وليس المهاجمين من البشر.

4.4 مجموعة معطيات "هجمات فقط" DEFCON 9 CTF

هي مجموعة بيانات أخرى تستخدم عادة في تقييم كشف الاختراق (23). DEFCON هو مؤتمر سنوي القرصنة، ويشمل طروحات ومسابقة لكشف ومكافحة القرصنة. تسجل كل حركات المرور الشبكي التي ولدت خلال المنافسة وتصبح متاحة. مجموعة البيانات هذه يمكن استخدامها لاختبار اختراقات أنظمة الكشف، على الرغم من أن لها عدد قليل من عناوين IP، وبسبب تركيزها العالي غير العادي على الاختراقات، وبالتالي عدم وجود حركة مرور خلفية يجعلها مناسبة لتقييم مصنفات التنبيهات في نظم كشف الاختراق.

4.5 مجموعات البيانات الحقيقية

وأخيراً، هناك مصدراً آخر من مصادر البيانات هو التنبيهات الصادرة من نظم كشف الاختراق الحقيقية المنتشرة في بيئات الشركات. وبما أن نظم كشف الاختراق تكون مفيدة فقط عندما يتم مراجعة التنبيهات التي تصدرها، يمكننا أن نفترض أن معظم تنبيهات نظم كشف الاختراق هي في الواقع منقحة. خلال التنقيح يحاول المحلل فهم السبب الجذر للتنبيه، ويحدد إذا كانت طبيعي أو خبيث، وإذا كان هناك أي إجراء لا بد من اتخاذه. واعتماداً على البيئة، يمكن أن يكون التنقيح أكثر أو أقل رسمية: في بعض الحالات يوسم كل تنبيه بعلامة لتحديد الأسباب الجذر له. وفي حالات أخرى يتم التحقيق فقط في الهجمات الناجحة، مما يجعل هذا التصنيف ضمنياً. ومن الواضح أن السيناريو الأول هو مفيد أكثر بالنسبة لنا، لأن هناك مجموعة

متنوعة من الأسباب الجذر وتوزيعها يكون أكثر توازناً. في السيناريو الثاني، الهجمات الناجحة هي أحداث تكون عادة نادرة جداً، مما يجعل التصنيف مسألة صعبة جداً.

إن ميزة استخدام مجموعات البيانات الحقيقية هي في الحقيقة أنها بيئات حقيقية يمكن استخدام النظام فيها. من ناحية أخرى، فإن البيانات الحقيقية لديها مشكلة هي أن تصنيف التنبيهات يكون ذاتياً وأحيانا غير مكتمل أو غير صحيح.

اعتمدنا استخدام مجموعة المعطيات DARPA1998 (سيتم التحدث عنها لاحقاً في هذا الفصل) انطلاقاً من معرفتنا أنها الأفضل وهي المرجع الوحيد المتاح للعموم، والتي يمكن استخدامها من قبل باحثين مستقلين لإجراء الاختبارات في مجال تقييم نظم كشف الاختراق وتصنيف التنبيهات.

4.6 **مجموعة البيانات** DARPA1998

مجموعة البيانات DARPA 1998 هي مجموعة بيانات اصطناعية تم جمعها في شبكة كومبيوتر محاكية لشبكة كومبيوتر في قاعدة عسكرية وهمية. تم توصيل الشبكة إلى الخارج من خلال موجه router. تم تعيين الموجه بسياسة مفتوحة، أي بحيث لا يمنع أي اتصالات. تم تشغيل المحاكاة لمدة سبعة أسابيع وأسفرت عن خمسة أسابيع من بيانات التدريب وأسبوعين من بيانات الاختبار. توجد جداول الهجوم الحقيقية لتوصيف الهجمات التي وقعت في كلا الفترتين. تتكون مجموعة بيانات darpa98 من مجموعة ملفات من حركة المرور الشبكي (ملفات tcpdump) لنظم كشف الاختراق الشبكية؛ وبيانات تدقيق لنظم كشف الاختراق لحاسب وحيد. سنعرض محتويات حزمة المعطيات هذه التي تهمنا بالتفصيل عند الحديث عن التنفيذ العملي وإجراء التجربة.

الفصل الخامس: أعمال متعلقة

في هذا الفصل سوف نظهر كيف استخدمت حلول مماثلة في جميع هذه المجالات. في النظم التي تستخدم تقنيات تعلم الآلة عادة ما تستخدم حالات موصفة صادرة عن مشغل أو مخبر ما. ومع ذلك، فقط مجموعة فرعية من تلك التطبيقات تقر أن البيانات مطلوبة تدريجيا وتستفيد من هذه الحقيقة في التعلم أوعملية التقييم. في طريقة معالجة التنبيهات، مع وجود استثناءات قليلة، فإنها تعالج مشكلة الأخطاء الايجابية في المستوى الثالث. وتشمل هذه النظم نظم ارتباط التنبيهات وتقنيات التنقيب في البيانات.

في عام 2000، قام (Clifton & Gengo) (5) باستخدام تقنيات التنقيب في البيانات لتحديد تسلسل التنبيهات المحتملة الناتجة من السلوك العادي، مما يمكن (عند بناء المرشحات) من حذف تلك التنبيهات. قاموا أيضاً بالبحث عن عن تسلسل التنبيهات المتكررة، من أجل استخدام هذه المعرفة لخلق مرشحات لتنبيهات نظم كشف الاختراق.

أظهر K. Julisch, 2001 (9) أن التنبيهات يجب أن تدار من خلال تحديد وحل أسبابها الجذرية. وقدم طريقة عنقدة التنبيهات الصادرة كوسيلة تدعم اكتشاف الأسباب الجذر لهذه التنبيهات. نمذج Julisch التنبيهات كمجموعة من السمات سجلات التنبيه على أنها مجموعة من التنبيهات. وتم إنشاء تصنيفات لكل سمة معينة . واستند في وصف التشابه بين التنبيهات على التصنيفات المحددة. وبالتالي، يتم تجميعها معاً في عنقود واحد. للقيام بذلك، استخدم خوارزميات التنقيب في البيانات لتنفيذ عملية العنقدة تلك. ونتيجة لذلك، تمكن من الحصول على تنبيهات عمومية وأتاح له اكتشاف الأسباب الجذر لها. وبإزالة تلك الأسباب، أظهر Julisch أن عدد التنبيهات الخاطئة انخفض أكثر من 90%.

(Pietraszek. T. , 2006) (19) اقترح متعلم تكيفي Adaptive Learner Alerts Classifier ALAC لتصنيف تنبيهات نظام كشف الاختراق للحد من التنبيهات الايجابية الكاذبة. وبيّن أن المعرفة الخلفية يمكن أن تكون مفيدة لتصنيف التنبيهات. النظام المقترح كان مصنف للتنبيهات معتمد على ردود فعل المحلل (مسؤول كشف التسلل) وتقنيات تعلم الآلة. ان مسألة تصنيف تنبيهات نظام كشف الاختراق تعتبر مسألة صعبة من مسائل تعلم الآلة. وقد تم تصميم ALAC للعمل في وضعين: وضع المزكي recommender، والذي فيه توسم جميع التنبيهات وتمرر الى المحلل، ووضع الوكيل agent، والذي فيه يتم معالجة بعض التنبيهات تلقائياً. في وضع المزكي، حيث يتعلم بتكيف التصنيف من المحلل، تم الحصول على تنبيهات خاطئة سلبية وخاطئة

إيجابية. بالمقابل في وضع الوكيل، تتم معالجة بعض التنبيهات بشكل مستقل (على سبيل المثال، يتم تجاهل التنبيهات الخاطئة بثقة عالية).

في 2006 قام Law & Kwok (11) باستخدام مصنِّف K-Nearest Neighbour KNN لفلترة التنبيهات الخاطئة التي يطلقها الـ IDS، نقدم هنا توضيح بسيط حول آلية عمل KNN-Classifier، يقوم هذا المصنف باستخدام المسافة الاقليدية لإيجاد التشابه بين نقطتي بيانات، إن ناتج التشابه النهائي لنقطة بيانات جاري تصنيفها يساوي معدل المسافات الاقليدية عن العدد κ من النقاط الطبيعية الأقرب لهذه النقطة. إذا كان ناتج التشابه لنقطة ما أكبر من عتبة معينة معرفة سابقاً فإن هذه النقطة يتم تصنيفها بأنها نقطة شاذة وفيما عدا ذلك سيتم تصنيفها بأنها نقطة طبيعية. نفس المبدأ تم اعتماده من قبل Kwok و Law في نظام كشف الاختراق، قاموا أولاً بنمذجة عينات تنبيهات الـ IDS في الحالة الطبيعية عندما لا يكون هناك أي اختراق، وبعدها قاموا باكتشاف النشاطات الشاذة عن طريق تصنيف التنبيهات القادمة من قناة التنبيهات باستخدام المصنف KNN. وكانوا قادرين على تقليل ما يصل الى 93% من التنبيهات الخاطئة باستخدام مجموعة المعطيات DARPA 1998، والتي تم استخدامها كعلامة عامة لتقييم ولاختبار أداء نظم كشف الاختراق.

في عام 2009 اقترح R. Vaarandi (20) طريقة تنقيب في المعطيات لتصنيف التنبيهات في الوقت الحقيقي لتمييز التنبيهات الكاذبة المهمة عن الأحداث ذات الأهمية المنخفضة الكاذبة. ويدعي أنه على عكس النهج التقليدية في استخراج البيانات، فإن هذه الطريقة مؤتمتة بالكامل وقادرة على التكيف مع تغيرات البيئة دون تدخل بشري.

في عام 2013 قدم (Mokarian, Faraahi, & Delavar) (13) نشرة تفصيلية عن الأبحاث الصادرة بين عامي 2000 و 2011 في مجال تقليل الأخطاء الإيجابية لنظام كشف الاختراق. وصنفوا الأبحاث إلى قسمين: الأول في مجال تحسين الكشف، والثاني في مجال معالجة وفلترة التنبيهات الصادرة، وقدموا إحصائية للأبحاث الجارية كل مع مجموعة البيانات المستخدمة، الشكل التالي يمثل لمحة عن هذه الأبحاث:

	Researches (2000-2011)	False Positive Reduction Techniques	KDD CUP 99	DARPA 1998	DARPA1999	DARPA 2000	Real World	Results	
Detection Techniques	[15]	SVM	*					1.00%	False Positive Rate
	[15]	C4.5	*					1.44%	
	[19]	Decision Tree Classification ,Rule-based Classification	*					3.2%	
	[20]	Decision tree Classification , Bayesian Clustering	*					N/A	
	[22]	Self-Organizing Map , K-means Clustering	*				*	0.91-2.43%	
Alert Processing Techniques	[23]	Sequential Association Mining						NA	False Positive Reduction Ratio
	[25], [26]	Clustering (Attribute Oriented Induction)					*	75%, 87%	
	[10],[27],[28]	Machine-Learning (ALAC), Clustering (CLARAty)			*		*	30%, 50%	
	[29]	Quality Parameters , Normalization				*		98.03%	
	[30]	Multi-Level Clustering (Fuzzy Cognitive Modeling)				*		N/A	
	[31]	Clustering (based on xml distance measure)		*				N/A	
	[32] , [33]	Classification , Clustering			*		*	37%	
	[34],[35],[36]	Clustering , root cause analysis		*	*		*	82%,93%,74%	
	[5], [37]	Classification (Frequent Itemset Mining) , Clustering					*	81-99%,43.31%	
	[8]	Statistical Filtering			*			75%	
	[38]	Classification (Pattern Mining)			*			36%	
	[40]	Clustering , GHSOM					*	15% - 4.7%	
	[7]	Self-Organizing Map , K-means Clustering			*		*	90%,87%,50%	
	[13]	Rule-based Classification	*					N/A	
	[39]	Fuzzy Alert Aggregation			*			N/A	

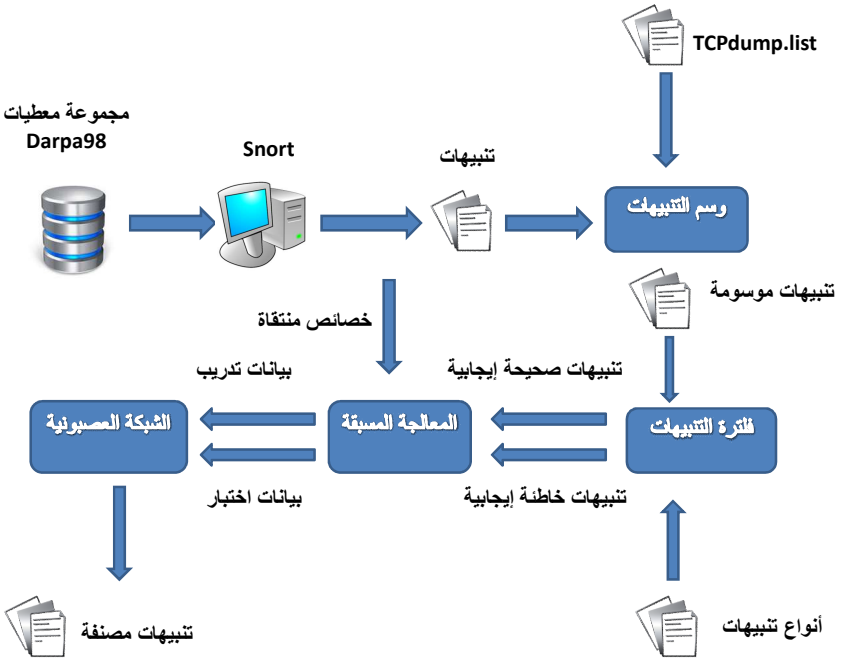
الشكل 5.1 نشرة عن الأبحاث الصادرة في مجال تقليل الأخطاء الإيجابية الواردة في مقالة (Mokarian، Faraahi، Delavar) (13).

اعتماد استخدام الشبكات العصبونية

في الحقيقة لم نتمكن من الإطلاع على أية تفاصيل فيما يخص الشق العملي والتنفيذي في الأبحاث التي اطلعنا عليها في هذا المجال، وانجذبنا بشكل حدسي إلى التصنيف والتنقيب في الأنماط وإدراكها، لم نشهد استخدام الشبكات العصبونية في هذا المجال على الرغم من بساطتها وملائمتها للمسألة (كمسألة تصنيف وتنقيب في الأنماط). في الحقيقة استخدم شيء مشابه للشبكات العصبونية وهي خرائط التنظيم الذاتية ولكن تم استخدامها من منظور آخر أتينا على ذكره سابقاً وهو عنقدة التنبيهات وليس تصنيفها. لذلك قررنا اختبار استخدام الشبكات العصبونية مع مجموعة المعطيات Darpa98 في هذه الدراسة.

الفصل السادس: التنفيذ العملي

يمثل الشكل التالي مخططاً للنظام المقترح، يعتمد النظام بشكل أساسي على التنبيهات المولدة من قبل نظام كشف الاختراق. لتوليد التنبيهات، استخدمنا snort مع مجموعة المعطيات darpa98. snort، كما ذكرنا سابقاً، نظام كشف اختراق مفتوح المصدر يأخذ مجموعة جزئية من مجموعة المعطيات darpa98 كدخل له ليقوم بإصدار التنبيهات المناسبة لها. هذه التنبيهات تعتبر دخل النظام المقترح.



الشكل 6.1 مخطط النظام

6.1 تحضير بيئة العمل

إن الهدف الفعلي من هذه المرحلة هو استخلاص البيانات اللازمة لبناء المصنف الخاص بنا، حيث أننا في هذا المشروع لا نهدف إلى تحسين آلية عمل نظام كشف الاختراق وإنما فقط إلى استخدام خرج هذا النظام كدخل للنظام الخاص بنا. لتحضير بيئة العمل قمنا باختيار SNORT الذي قدمنا شرح موجز عنه في الفقرة 2.1.2 كونه نظام كشف اختراق مفتوح المصدر وهو الأكثر شيوعاً والأكثر استخداماً في تقييم واختبار نظم كشف الاختراق. قمنا بتنصيب النسخة SNORT2.9 (متوفرة على الموقع www.snort.com) على نظام التشغيل ubuntu 14 وفي مايلي الخطوات الأساسية لعملية التنصيب:

6.1.1 تنصيب المكتبات الضرورية لعمل snort

هناك عدد من المكونات الضرورية لتشغيل **snort**، نعرض أهمها فيما يلي:

- المكتبة **PCAP** وهي المسؤولة عن التقاط الطرود الشبكية التي سيقوم **snort** لاحقاً بتحليلها وإصدار التنبيهات الخاصة بكل منها.
- المكتبة **PCRE** وهي المسؤولة عن تفسير التعابير النظامية التي تكتب باستخدامها أغلب قواعد الكشف الخاصة بـ **snort.**
- المكتبة **libdnet** وهي المسؤولة عن أغلب العمليات الشبكية منخفضة المستوى مثل تفسر عناوين الانترنت وقد تكون منصبة أصلاً ولكن يجب التأكد من موافقة النسخة الفاعلة مع نسخة **snort** الجاري تنصيبها.
- المكتبة **DAQ** هي المسؤولة عن استدعاءات المكتبة **PCAP** للاستحواذ على الطرود الشبكية.

6.1.2 تنصيب snort

بعد تنصيب هذه المكتبات نحن جاهزين لتنصيب snort. وبعد التنصيب نقوم بإنشاء ملفات الإعدادات الضرورية وأهمها snort.conf , snort.conf, threshold.conf,gen-msg. map,attribute_table.dtd, Unicode.map.

إن الإعداد الأهم الواجب ذكره في هذه المرحلة هو إعداد المتحول HOME_NET في ملف الاعداد snort.conf والذي سنقوم بإسناد قيمة الشبكة الداخلية في مجموعة المعطيات darpa98 وهي 172.16.0.0 عبر تعديل السطر الخاص بهذا المتحول في الملف snort.conf ليصبح كالتالي:

ipvar HOME_NET 172.16.0.0/16 وهي الشبكة الداخلية في مجموعة المعطيات darpa98.

6.1.3 تنصيب مجموعة القواعد اللازمة لكشف الاختراقات

لتنصيب مجموع قواعد الكشف، كنا أمام خيارين:

- **استخدام الأداة Pulled Pork** : Pulled Pork هو سكريبت مكتوبة بلغة Perl لتنزيل وتنصيب وإعداد ملفات قواعد snort الخاصة بكشف الاختراق،
- **تنزيل القواعد وتنصيبها يدوياً** من دون اللجوء إلى استخدام Pulled Pork.

في الحقيقة لجأنا إلى استخدام الأداة Pulled Pork لسهولة التعامل معها حيث أن هدفنا ليس التعامل والإلمام بكل تفاصيل عمل snort ولكن استخدامه للحصول على البيانات الأولية لبناء المصنف الخاص بنا.

فيما يلي أهم الخطوات المطلوبة لتنفيذ عملية التنصيب:

- تنزيل وتنصيب Pulled Pork.
- اعداد Pulled Pork لتنزيل القواعد أتوماتيكياً، عن طريق تعديل الملف pulledpork.conf وإدخال الـرمز الخاص بالمستخدم الذي حصلنا عليه عند اتمام عملية التسجيل في الموقع www.snort.com والمسمى oinkcode.
- تشغيل Pulled Pork بالتعليمة sudo /usr/local/bin/pulledpork.pl -c ./etc/snort/pulledpork.conf –l.

6.1.4 تنصيب Barnyard2

من المهم جداً كتابة الأحداث التي يبلغ عنها snort بشكل مقروء من المحلل البشري، سواء على الواجهة النصية أو إلى قواعد بيانات، أو إلى ملفات نصية. بالشكل المثالي نحن بحاجة إلى كتابة أحداث snort إلى مخدم قواعد بيانات لنستطيع قراءتها والبحث فيها وفلترتها. فيما يلي أهم الخطوات التي قمنا بها في هذه الخطوة:

- تنصيب مخدم قواعد بيانات MySql.
- خلق قاعدة بيانات خاصة بـ snort.
- تنصيب Barnyard2 وهو مفسر مفتوح المصدر لملفات خرج snort الثنائية، سوف نقوم بإعداد snort ليقوم بكتابة خرجه (الأحداث)كملفات ثنائية إلى مجلد الخرج، ومن ثم نقوم بإعداد Barnyard2 لقراءة هذه الأحداث بشكل متزامن وكتابتها إلى قاعدة المعطيات.

6.2 استخلاص البيانات الأولية

سوف نقوم بتمرير مجموعة من بيانات darpa98 إلى نظام كشف الاختراق للحصول على التنبيهات، اخترنا أن نختبر النشاط الشاذ الخارجي من بيانات darpa98، لذلك قمنا بتمرير معطيات المجموعة outside.tcpdump فقط إلى snort. يوضح الشكل 6.2 بعض عينات المجموعة outside.tcpdump من يوم الإثنين من الإسبوع الأول لمعطيات تدريب darpa98. مأخوذة من برنامج التقاط الطرود wireshark. قمنا بإعداد snort ليعمل بنمط NIDS mode وقمنا بخلق 19 جدول في قاعدة البيانات لتخزين التنبيهات الناتجة من تمرير حزمة الطرود inside.tcpdump إلى snort عبر التعليمة :

Snort –r outside.tcpdump –c - /etc/snort/snort.conf

No.	Time	Source	Destination	Protocol	Info
397217	5455.843481	172.16.114.148	196.227.33.189	TCP	ftp-data > 8824 [ACK] Seq=1 Ack=1 Win=32120 Len=0
397218	5455.843657	172.16.114.148	196.227.33.189	FTP	Response: 150 Opening BINARY mode data connection for committee
397219	5455.847091	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397220	5455.848307	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397221	5455.854575	196.227.33.189	172.16.114.148	TCP	8824 > ftp-data [ACK] Seq=1 Ack=2921 Win=32736 Len=0
397222	5455.854676	196.227.33.189	172.16.114.148	TCP	8585 > ftp [ACK] Seq=901 Ack=2574 Win=32120 Len=0
397223	5455.857321	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397224	5455.858151	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 956 bytes
397225	5455.858267	172.16.114.148	196.227.33.189	TCP	ftp-data > 8824 [FIN, ACK] Seq=5337 Ack=1 Win=32120 Len=0
397226	5455.858348	172.16.114.148	196.227.33.189	FTP	Response: 226 Transfer complete.
397227	5455.858431	196.227.33.189	172.16.114.148	TCP	8824 > ftp-data [ACK] Seq=1 Ack=5338 Win=30660 Len=0
397228	5455.866071	196.227.33.189	172.16.114.148	TCP	8824 > ftp-data [FIN, ACK] Seq=1 Ack=5338 Win=32120 Len=0
397229	5455.866434	172.16.114.148	196.227.33.189	TCP	ftp-data > 8824 [ACK] Seq=5338 Ack=2 Win=32120 Len=0
397230	5455.866523	196.227.33.189	172.16.114.148	FTP	Request: PORT 196,227,33,189,34,122
397231	5455.867066	172.16.114.148	196.227.33.189	FTP	Response: 200 PORT command successful.
397232	5455.869761	196.227.33.189	172.16.114.148	FTP	Request: RETR err2data.c
397233	5455.871677	172.16.114.148	196.227.33.189	TCP	ftp-data > 8826 [SYN] Seq=0 Win=512 Len=0 MSS=1460
397234	5455.871899	196.227.33.189	172.16.114.148	TCP	8826 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=32736 Len=0 MSS=1460
397235	5455.872283	172.16.114.148	196.227.33.189	TCP	ftp-data > 8826 [ACK] Seq=1 Ack=1 Win=32120 Len=0
397236	5455.872471	172.16.114.148	196.227.33.189	FTP	Response: 150 Opening BINARY mode data connection for err2data.
397237	5455.882685	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397238	5455.883951	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397239	5455.884570	196.227.33.189	172.16.114.148	TCP	8826 > ftp-data [ACK] Seq=1 Ack=2921 Win=32736 Len=0
397240	5455.884656	196.227.33.189	172.16.114.148	TCP	8585 > ftp [ACK] Seq=946 Ack=2698 Win=32120 Len=0
397241	5455.887389	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1460 bytes
397242	5455.888345	172.16.114.148	196.227.33.189	FTP-DATA	FTP Data: 1178 bytes
397243	5455.888444	172.16.114.148	196.227.33.189	TCP	ftp-data > 8826 [FIN, ACK] Seq=5559 Ack=1 Win=32120 Len=0
397244	5455.888524	172.16.114.148	196.227.33.189	FTP	Response: 226 Transfer complete.
397245	5455.888609	196.227.33.189	172.16.114.148	TCP	8826 > ftp-data [ACK] Seq=1 Ack=5560 Win=30660 Len=0
397246	5455.888742	196.227.33.189	172.16.114.148	TCP	8826 > ftp-data [FIN, ACK] Seq=1 Ack=5560 Win=32120 Len=0
397247	5455.889118	172.16.114.148	196.227.33.189	TCP	ftp-data > 8826 [ACK] Seq=5560 Ack=2 Win=32120 Len=0

الشكل 6.2: لقطة من مجموعة بيانات darpa98 ممرّرة إلى برنامج التقاط تحليل الطرود wireshark.

بعد تمرير حزمة الطرود الى نظام كشف الاختراق snort فإن snort سوف يقوم بإصدار التنبيهات المناسبة بالرجوع إلى قواعد الكشف الموجودة لديه. وسوف يقوم بتخزين هذه التنبيهات في قاعدة البيانات التي أنشأناها سابقاً في مرحلة تنصيب snort. سنقوم بتعديل ملف إعدادات barnyard لنجعل snort يخزن التنبيهات في ملف csv. فيمايلي لقطة من التنبيهات المخزنة في الملف output.csv:

timestam	sig_gener	sig_id	sig_rev	msg	proto	src	srcport	dst	dstport	ethsrc	ethdst	ethlen	tcpflags	tcpseq	tcpack	tcplen	tcpwindo	ttl	tos	id	dgmlen	iplen
06/01-04:(	138	5	1	sensitive_	TCP	194.27.251	1111	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0xDE0C22	0xAD9596A		0x7FB8	64	0	947	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	194.27.251	1111	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***AP***	0xDE0C24	0xAD9596A		0x7FB8	64	0	948	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	194.27.251	1111	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x156	***AP***	0xDE0C26	0xAD9596A		0x7FB8	64	0	949	328	73732
06/01-04:(	139	1	1	sensitive_data: sens		195.73.151.50		172.16.114.50		0:C0:4F:A	0:0:C:4:41	0x22						63	0	191	20	20480
06/01-04:(	139	1	1	sensitive_data: sens		195.73.151.50		172.16.113.50		0:C0:4F:A	0:0:C:4:41	0x46						59	0	210	56	57344
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CB3	0xC5C113F		0x7FB8	64	0	2195	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***AP***	0x1D9CB5	0xC5C113F		0x7FB8	64	0	2196	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CB7	0xC5C113F		0x7FB8	64	0	2197	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CB9	0xC5C113F		0x7FB8	64	0	2198	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CBB	0xC5C113F		0x7FB8	64	0	2199	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CBD	0xC5C113F		0x7FB8	64	0	2200	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CC0	0xC5C113F		0x7FB8	64	0	2201	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CC2	0xC5C113F		0x7FB8	64	0	2202	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CC4	0xC5C113F		0x7FB8	64	0	2203	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CC6	0xC5C113F		0x7FB8	64	0	2204	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CC8	0xC5C113F		0x7FB8	64	0	2205	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CCA	0xC5C113F		0x7FB8	64	0	2206	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CCC	0xC5C113F		0x7FB8	64	0	2207	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CCE	0xC5C113F		0x7FB8	64	0	2208	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CD0	0xC5C113F		0x7FB8	64	0	2211	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CD2	0xC5C113F		0x7FB8	64	0	2212	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CD4	0xC5C113F		0x7FB8	64	0	2213	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CD7	0xC5C113F		0x7FB8	64	0	2214	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CD9	0xC5C113F		0x7FB8	64	0	2215	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CDB	0xC5C113F		0x7FB8	64	0	2216	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CDD	0xC5C113F		0x7FB8	64	0	2217	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CDF	0xC5C113F		0x7FB8	64	0	2218	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CE1	0xC5C113F		0x7FB8	64	0	2219	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CE3	0xC5C113F		0x7FB8	64	0	2220	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CE5	0xC5C113F		0x7FB8	64	0	2221	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CE7	0xC5C113F		0x7FB8	64	0	2222	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CE9	0xC5C113F		0x7FB8	64	0	2223	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CEC	0xC5C113F		0x7FB8	64	0	2224	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CEE	0xC5C113F		0x7FB8	64	0	2225	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CF0	0xC5C113F		0x7FB8	64	0	2226	576	65544
06/01-04:(	138	5	1	sensitive_	TCP	195.73.151	2062	172.16.113	25	0:C0:4F:A	0:0:C:4:41	0x24E	***A****	0x1D9CF2	0xC5C113F		0x7FB8	64	0	2227	576	65544

الشكل 6.3 لقطة من التنبيهات الصادرة عن snort بعد تخزينها في ملف csv

6.3 وسم التنبيهات

في مجموعة البيانات darpa98 هناك بعض الملفات التي تحتوي على معلومات عن الهجمات في طرود المجموعة عن كل يوم تجريب. هذه الملفات هي ملفات نصية ذات لاحقة list. فيما يلي بعض محتوى ملف list. وهو يحتوي على البارامترات التالية لكل اتصال:

- Timestamp لحظة الاتصال.
- مدة الاتصال.
- خدمة الاتصال.
- عناوين الـ IP المصدر والهدف.
- البوابات المصدر والهدف.
- خانة تكون بالقيمة 1 إذا كان الاتصال يحتوي على هجوم و 0 إذا كان اتصال طبيعي.
- نوع الهجوم، وإذا لم يكن هناك هجوم تكون قيمة الخانة -.

فيما يلي بعض محتوى ملف tcpdump.list من يوم الإثنين من الاسبوع الثاني من معطيات التدريب darpa98:

```
9532  06/18/1988 15:10:18 00:00:01 http 30867 80 172.016.116.044
137.245.085.134 0 -
9534  06/18/1988 15:10:26 00:00:01 tcpmux 55384 1 010.020.030.040
172.016.112.050 1 neptune
```

يشكل الملف list. مع ملف التنبيهات الصادر عن snort دخل مرحلة وسم التنبيهات إما بأنها تنبيهات خاطئة أو بالهجوم الذي يدل عليه كل تنبيه، فيما يلي موجز عن خوارزمية وسم التنبيهات:

الدخل: ملف التنبيهات AlertListFile، الملف المرفق tcpdum.list.

الخوارزمية:

- أنشئ لائحة هجومات فارغة AttackList.
- أنشئ لائحة تنبيهات فارغة AlertList.
- من أجل كل سطر من ملف tcpdump.list:
  - إذا كان السطر موسوماً بأنه هجوم أضف السطر إلى لائحة الهجومات AttackList.
- من أجل كل سطر من ملف AlertListFile:
  - أنشئ مفتاح من القيم التالية: عنوان الانترنت المصدر، البوابة المصدر، عنوان الانترنت الهدف، البوابة الهدف، الخدمة.
  - إذا وجد المفتاح في لائحة الهجومات، قم بوسم السطر بالهجوم الموافق، من لائحة الهجمات. وإلا قم بوسم السطر على أنه خطأ إيجابي.
  - أضف السطر إلى لائحة التنبيهات AlertList.

الخرج: لائحة التنبيهات

بعد تنفيذ خوارزمية وسم التنبيهات السابقة سنحصل على لائحة من التنبيهات الصادرة من snort وهي موسومة إما على أنها خطأ إيجابي (normal_alert)، أو باسم الهجوم الذي حصلنا عليه من ملفات list. الموافقة. فيما يلي لقطة من ملف التنبيهات بعد وسمها:

dst	dstport	ethsrc	ethdst	ethlen	tcpflags	tcpseq	tcpack	tcplen	tcpwindo	ttl	tos	id	dgmlen	iplen	icmptype	icmpcode	icmpid	icmpseq	
207.200.7	80	0:0:C:4:41	0:60:97:DE	0x154	***AP***	0xD9E0F5	0xAFDA7151		0x7D78	63	0	5460	326	20					normal_alert
207.46.13	80	0:0:C:4:41	0:60:97:DE	0x18F	***AP***	0x9CEFA4	0x815BFF20		0x7D78	63	0	55255	385	20					normal_alert
172.16.11	21	0:C0:4F:A	0:0:C:4:41	0x46	***AP***	0xD57109	0xC25230E2		0x7D78	64	16	3451	56	20					normal_alert
197.218.1	11457	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x5A21E6	0x700DCDD2		0x2238	254	0	30323	55	20					normal_alert
172.16.11	21	0:C0:4F:A	0:0:C:4:41	0x46	***AP***	0xC0AA85	0xCE14110D		0x7D78	64	16	1579	56	20					normal_alert
135.13.21	20542	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x9656FF	0x57A7C808		0x2238	254	0	3940	55	20					normal_alert
135.13.21	4160	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x1F2CDF	0x9DD7D465		0x2238	254	0	22619	55	20					normal_alert
194.7.248.	22682	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0xF131EC	0x7BE2E456		0x7FE0	63	16	40956	55	20					normal_alert
194.27.25	20553	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x340401	0x9C03AABB		0x7FE0	63	16	8275	55	20					normal_alert
207.25.71.	80	0:0:C:4:41	0:60:97:DE	0x119	***AP***	0xCFDC03	0xF566DB98		0x7D78	63	0	37604	267	20					normal_alert
172.16.11	12384	0:60:97:DE	0:0:C:4:41	0x41F	***AP***	0x188697	0x7995C064		0x7FE0	64	0	32079	1041	20					normal_alert
194.27.25	25318	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x936684	0x2FCFA90D		0x7FE0	63	16	60825	55	20					normal_alert
135.13.21	4238	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0x265EC8	0xB78D0696		0x2238	254	0	35686	55	20					normal_alert
172.16.11	21	0:C0:4F:A	0:0:C:4:41	0x46	***AP***	0x584875	0xC7E53A2		0x7D78	64	16	1893	56	20					normal_alert
209.143.1	80	0:0:C:4:41	0:60:97:DE	0x134	***AP***	0x570E37	0xCCDA715		0x7D78	63	0	14595	294	20					normal_alert
194.7.248.	26669	0:0:C:4:41	0:C0:4F:A	0x45	***AP***	0xC0F00F	0xC19AD449		0x7FE0	63	16	13237	55	20					normal_alert
197.218.1	21	0:0:C:4:41	0:C0:4F:A	0x46	***AP***	0x28A221	0x757D1652		0x7D78	63	16	14731	56	20					normal_alert
209.143.1	80	0:0:C:4:41	0:60:97:DE	0x125	***AP***	0x402B7A	0x3D937E0		0x7D78	63	0	2995	279	20					normal_alert
197.218.1	21	0:0:C:4:41	0:C0:4F:A	0x46	***AP***	0xC34C6B	0xE1613142		0x7C00	63	16	11622	56	20					normal_alert
205.181.1	80	0:0:C:4:41	0:60:97:DE	0x143	***AP***	0x15EE61	0x134777E5		0x7D78	63	0	61215	309	20					normal_alert
197.218.1	21	0:0:C:4:41	0:C0:4F:A	0x46	***AP***	0x7A01A9	0x1E935C79		0x7D78	63	16	63761	56	20					normal_alert
172.16.11	21	0:C0:4F:A	0:0:C:4:41	0x46	***AP***	0x8B00D6	0x52D4F17A		0x7D78	64	16	54827	56	20					normal_alert

الشكل 6.4 لقطة من ملف التنبيهات بعد وسمها.

6.4 ربط البروتوكولات

سنقوم في هذه المرحلة بإرجاع الأنواع المختلفة للبروتوكولات الموجودة في ملف الـ list. من مجموعة المعطيات darpa98 إلى واحد من البروتوكولات الأساسية (tcp,udp,icmp). الهدف من هذه المرحلة هو حتى يمكننا مقارنة الحقلين في الملف list. وملف التنبيهات الصادر من snort حيث أن snort يملأ خانة البروتوكول في التنبيه بواحد من البروتوكولات الثلاثة المذكورة. سنقوم بمسح ملفات الـ list. في مجموعة المعطيات وخلق خريطة لربط هذه البروتوكولات وفق القاعدة التالية:

If protocol contains “/i” then protocol is icmp.
If protocol contains “/u” then protocol is udp
Else protocol is tcp

فيمايلي لقطة من الملف الناتج من ربط البروتوكولات المختلفة من ملف list. من مجموعة المعطيات darp98:

imap tcp
444/u udp
1/u udp
urp/i icmp
4/u udp

6.5 فلترة التنبيهات

استناداً إلى ما قدمه كل من [Brugger&Chow] (24) في معرض تقييمهم لمجموعة المعطيات darpa98 باستخدام نظام كشف الاختراق snort، فإن snort لا يقوم بكشف جميع الاختراقات الموجودة في هذه المجموعة بنفس الدقة التي يقوم بالكشف عن بعضها.

سوف نقوم بفلترة التنبيهات الملتقطة من snort إلى هذه الأنواع المحددة من الهجمات حيث أننا لا نسعى في بناء المصنف إلى كشف الهجمات التي لايقوم snort بكشفها ولكن إلى تقليل الأخطاء الإيجابية في التنبيهات المضمونة التي يصدرها.

فيمايلي أهم الهجمات التي يقوم snort بالكشف عنها بدقة مقبولة وتوصيفاتها كما أشير إليها في موقع https://www.ll.mit.edu/ideval/docs/attackDB.html:

- Back وهو هجوم حجب خدمة denial of service، هذا الهجوم يستهدف مخدم الويب Apache، يقوم المهاجم بإرسال طلبات http بعناوين تحتوي على العديد من "/" وعندما يحاول المخدم معالجة هذه الطلبات سوف يؤدي ذلك إلى بطء تخديمه وأكثر من ذلك يصبح غير قادر على معالجة الطلبات الأخرى.
- Land هو هجوم حجب خدمة، وهو فعال ضد بعض تحقيقات المكدس tcp/ip، يحدث هذا الهجوم عندما يقوم المهاجم بإرسال طرد syn يكون فيه عنوان المصدر هو نفسه عنوان الهدف.
- Pod أو ما يعرف بـ ping of death، وهو هجوم حجب خدمة شائع يؤثر على العديد من أنظمة التشغيل القديمة، ورغم أنه أصبح هجوم شائع جداً إلا أنه ثبت أن بعض الأنظمة تتصرف بشكل مغاير عند التعرض لمثل هذا الهجوم، مثل التخريب أو الجمود أو إعادة الاقلاع.
- Phf وهو هجوم يستخدم الواجهات البرمجية للتطبيق cgi بشكل سيء لتنفيذ تعليمات على مستوى الصلاحيات في مخدم الويب. أي برنامج نصي cgi يستخدم التابع escape_shell_cmd() يمكن أن يكون تهديد لهجوم.
- Rootkit هو مجموعة من البريمجات، عادة غير مشروعة، مخصصة للوصول إلى موارد الحاسب أو إلى برامجه بطريقة غير مشروعة.
- Imap يستهدف هذا الهجوم مخدمات imap في نظام التشغيل redhat linux ويعتمد على مفيض المكدس stack overflow بشكل يسمح للمهاجمين بتنفيذ عمليات وتعليمات بصلاحيات مدير النظام.

- Dict هجوم القاموس dictionary، وهو ما يسمح للمهاجمين بالوصول إلى موارد الحاسب بعد تنفيذ سلسلة عمليات تخمين لاسم المستخدم وكلمة السر. وهو ممكن أن يستهدف العديد من الخدمات، مثل البريد، الويب، ftp..الخ.
- Nmap هو نوع من أنواع المسح للبوابات الفاعلة في النظام لمعرفة الخدمات الجارية ومحاولة الوصول إليها.

في مرحلة فلترة التنبيهات سوف نقوم باختزال التنبيهات الصادرة عن snort إلى التنبيهات التي تبلغ عن الهجمات السابقة فقط بغية الحصول على أفضل أداء للمصنف. في هذه المرحلة أيضاً سوف نقوم بانتقاء خصائص التنبيهات التي سوف نبني المصنف على أساسها وهي كالتالي:

- Event signature وهو المعرف الخاص بالقاعدة المصدرة للتنبيه.
- Signature revision وهو عدد مميز لمراجعات القاعدة المصدرة للتنبيه ويجب أن يستخدم مع المعرف السابق.
- Protocol وهو بروتوكول الطرد الذي أصدر بسببه snort ذلك التنبيه.
- عنوان IP المصدر.
- البوابة المصدر.
- عنوان IP الهدف.
- البوابة الهدف.
- طول طرد الـ IP.

وفي حالة تنبيهين لهما نفس قيم الخصائص السابقة فإننا سوف نختار أحد هذين التنبيهين وتنبين التجربة أن هذه المرحلة لا تقوم بحذف تنبيهين بخصائص متشابهة ولكن مع نوع هجوم مختلف.

حصلنا في نهاية هذه المرحلة على 8644 تنبيه صحيح إيجابي سيشكلون نواة بيانات الاختبار والتدريب للمصنف الهدف.

6.6 المعالجة الأولية للتنبيهات

إن بعض خصائص التنبيهات من المرحلة السابقة تكون عددية وبعضها الآخر تكون سلاسل نصية، في هذه المرحلة نقوم بتحويل الخصائص ذات القيم النصية إلى قيم عددية. وفي هذه المرحلة أيضاً سوف نقوم بتقييس مجال القيم بين القيمتين [0,1]، إذاً هذه هي المرحلة الأخير لتهيئة المعطيات لبناء المصنف، أن دخل هذه المرحلة هو

لائحة من التنبيهات الخاطئة والصحيحة وخرجها هو معطيات التدريب والاختبار للمصنف.

باستخدام المعايير التالية نقوم بتحويل القيم النصية لعنوان الانترنت والبروتوكول إلى معطيات عددية:

- $IP = X_1.X_2.X_3.X_4$
    - $IP_Val = (((X_1*256)+X_2)*256+X_3)*256+X_4$
- Protocol_Val =
    - 0 if no protocol detected.
    - 4 if icmp.
    - 10 if TCP.
    - 17 if UDP.

إن تقييس القيم إلى المجال [0,1] Unit Range UR من الممكن أن يفقد المعطيات دقتها، لذلك سوف نتبع طريقة ثانية لتقييس القيم إلى المجال [0.1,0.9] Improved Unit Range IUR.

- $UR = \frac{X-Xmin}{Xmax-Xmin}$
- $IUR = 0.1 + \frac{X-Xmin}{Xmax-Xmin} + 0.8$

نعرض فيمايلي لقطات من ملفات التنبيهات قبل وبعد معالجتها بالطريقتين المذكورتين اعلاه.

1	sig_id	sig_rev	protocol	src_ip	src_port	dest_ip	dest_port	dgmlen	attack_type
2	1019	6	10	2265463990	4193	2886758962	80	1500	back
3	1019	6	10	2265463990	5173	2886758962	80	1500	back
4	1019	6	10	2265463990	4167	2886758962	80	560	back
5	1019	6	10	2265463990	4924	2886758962	80	560	back
6	1019	6	10	2265463990	4250	2886758962	80	1500	back
7	1019	6	10	2265463990	3841	2886758962	80	1500	back
8	1019	6	10	2265463990	4924	2886758962	80	1500	back
9	1019	6	10	2265463990	4192	2886758962	80	1500	back
10	1019	6	10	2265463990	4163	2886758962	80	1500	back
11	1019	6	10	2265463990	4196	2886758962	80	560	back
12	1019	6	10	2265463990	4200	2886758962	80	1500	back
13	1019	6	10	2265463990	4921	2886758962	80	560	back
14	1019	6	10	2265463990	4176	2886758962	80	560	back
15	1019	6	10	2265463990	4668	2886758962	80	560	back
16	1019	6	10	2265463990	4387	2886758962	80	560	back
17	1019	6	10	2265463990	4188	2886758962	80	1500	back
18	1019	6	10	2265463990	4221	2886758962	80	1500	back
19	1019	6	10	2265463990	4167	2886758962	80	1500	back
20	1019	6	10	2265463990	4186	2886758962	80	1500	back

الشكل 6.5: لقطة من ملف التنبيهات قبل المعالجة المسبقة

sig_id	sig_rev	protocol	src_ip	src_port	dest_ip	dest_port	dgmlen	attack_type
0.432455395	0.333333333	0.588235294	0.643474878	0.084325477	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.104034269	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.08380259	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.099026627	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.085471804	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.0772464	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.099026627	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.084305366	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.083722146	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.08438581	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.084466254	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.098966294	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.083983589	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.093878208	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.088227013	0.821272494	0.001608881	0.361413043	back
0.432455395	0.333333333	0.588235294	0.643474878	0.084224922	0.821272494	0.001608881	1	back
0.432455395	0.333333333	0.588235294	0.643474878	0.084888585	0.821272494	0.001608881	1	back

الشكل 6.6: لقطة من ملف التنبيهات بعد المعالجة المسبقة بطريقة UR

sig_id	sig_rev	protocol	src_ip	src_port	dest_ip	dest_port	dgmlen	attack_type
0.445964316	0.366666667	0.570588235	0.614779902	0.167460381	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.183227415	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167042072	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.179221302	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.168377443	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.16179712	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.179221302	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167444292	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.166977717	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167508648	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167573003	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.179173035	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167186872	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.175102566	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.17058161	0.757017995	0.101287105	0.389130435	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167379937	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167910868	0.757017995	0.101287105	0.9	back
0.445964316	0.366666667	0.570588235	0.614779902	0.167042072	0.757017995	0.101287105	0.9	back

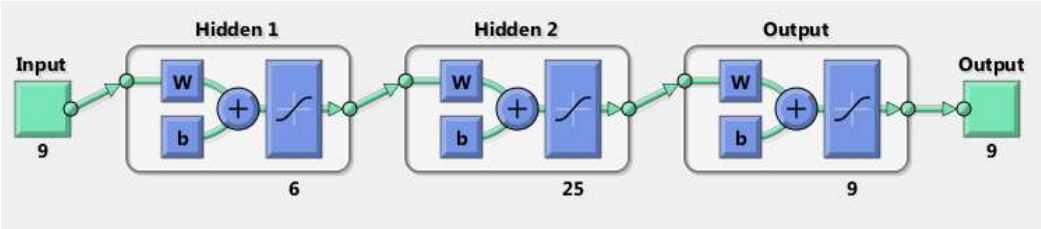
الشكل 6.7: لقطة من ملف التنبيهات بعد المعالجة المسبقة بطريقة IUR

ان المعطيات السابقة تشكل البيانات الأولية اللازمة لنا لبناء واختبار المصنف. سنقوم بقسم هذه المعطيات إلى قسمين متساويين. مجموعة بيانات تدريب المصنف ومجموعة بيانات اختباره.

6.7 بناء المصنف وإجراء التجربة

قررنا اعتماد الشبكات العصبونية في بناء المصنف المطلوب من أجل تصنيف التنبيهات الخاطئة. في الحقيقة سوف نقوم ببناء مصنف يقوم بتصنيف مجموعة التنبيهات الصادرة إلى عدة صفوف، صفوف بأسماء الهجمات التي تم شرحها سابقاً، وقمنا بإضافة صف جديد يمثل تنبيهات الخطأ الإيجابي باسم صف normal_alert.

قمنا باستخدام أداة الشبكات العصبونية nnstart الموجودة في بيئة ماتلاب MATLAB2013، وقمنا ببناء شبكة عصبونية من طبقتين. ولاختيار عدد العصبونات الأمثل في كل طبقة قمنا بأداء اختبار أمثلة على عدد العصبونات بين المجال [1,25] للحصول على الأداء الأفضل. حصلنا على أداء أمثل بعدد عصبونات 6 للطبقة المخفية الأولى و25 للطبقة المخفية الثانية. وفي حالة البيانات المعالجة بطريقة IUR كان عدد العصبونات الأمثل هو 23 للطبقة الأولى و22 للثانية. يبين الشكل التالي بنية الشبكة العصبونية المستخدمة في البيانات المعالجة بطريقة UR (وهي نفسها بالنسبة للبيانات المعالجة بطريقة IUR ولكن اختلف عنها بعدد العصبونات في كل طبقة):



الشكل 6.8 بنية الشبكة العصبونية المستخدمة في حالة البيانات المعالجة بطريقة UR.

لتدريب الشبكة العصبونية أخذنا نسبة 70% من التنبيهات المعالجة والمفلترة أي 6053 من التنبيهات الصحيحة الإيجابية، وأضفنا إليها 2443 تنبيه خاطئ كاذب، ليصبح مجموع عينات التدريب بين تنبيهات صحيحة ايجابية وموسومة وتنبيهات خاطئة كاذبة 8469

نحن بحاجة إلى معطيات دخل تكون منفصلة عن الصفوف التي تنتمي إليها، هذه الصفوف تمثل المعطيات الهدف التي سيتم تدريب الشبكة العصبونية عليها. يمثل الشكل التالي لقطة من معطيات التدريب المعالجة بطريقة UR مأخوذة من بيئة ماتلاب (لن نعرض البيانات الخاصة بطريقة IUR لأنها مشابهة لها):

0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.10546215...	0.82127249...	0.00160888...	0.36141304...	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.10526104...	0.82127249...	0.00160888...	0.36141304...	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.08515002...	0.82127249...	0.00160888...	0.36141304...	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.08814656...	0.82127249...	0.00160888...	1	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.09092188...	0.82127249...	0.00160888...	0.36141304...	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.10532137...	0.82127249...	0.00160888...	1	back
0.43245539...	0.33333333...	0.58823529...	0.64347487...	0.08370203...	0.82127249...	0.00160888...	0.36141304...	back
0.12659303...	0.26666666...	0.23529411...	0.82127249...	0	0.99094297...	0	0.37771739...	pod
0.17204757...	0.33333333...	0.58823529...	0.82127242...	0.00046255...	0.82127242...	0.00046255...	0.00815217...	land
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.56558201...	0.01970108...	dict
0.25106202...	0.53333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.34379776...	0.01630434...	dict
0	0	0.23529411...	0.64358008...	0	0.82127280...	0	1	pod
0.25106202...	0.53333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.37778537...	0.01630434...	dict
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.02091545...	0.02241847...	dict
0.25191163...	0.4	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.36384844...	0.01970108...	dict
0.25191163...	0.4	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.42021961...	0.01970108...	dict
0.25191163...	0.4	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.17619258...	0.01970108...	dict
0.25106202...	0.53333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.57980049...	0.01630434...	dict
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.36427077...	0.01970108...	dict
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.09576864...	0.01970108...	dict
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.12388383...	0.01970108...	dict
0.25191163...	0.4	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.09144477...	0.01970108...	dict
0.25191163...	0.4	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.33148982...	0.01970108...	dict
0.47068819...	0.33333333...	0.58823529...	0.82127249...	0.00046255...	0.98769842...	0.61809186...	0.01970108...	dict

الشكل 6.9 مصفوفة التدريب (مستثنى منها العمود الذي يشير لنوع الهجوم)

سنقوم بخلق المعطيات الهدف استناداً إلى ملف معطيات التدريب وستكون بينته عبارة عن مصفوفة من أصفار وواحدات. إن قيمة سطر من مصفوفة الهدف يمثل الصف الذي ينتمي إليه السطر المقابل من مصفوفة التدريب، تكون قيم هذا السطر جميعها أصفاراً باستثناء القيمة التي تمثل ترتيب الصف الذي ينتمي اليه هذا السطر. سنعرض فيما يلي كيفية ترتيب هذه الصفوف الذي اعتمدناه عند خلق ملف الدالة الهدف:

```
 if (AttackType == "nmap")
    MatrixRow = "1 0 0 0 0 0 0 0 0";
if (AttackType == "back")
    MatrixRow = "0 1 0 0 0 0 0 0 0";
if (AttackType == "land")
    MatrixRow = "0 0 1 0 0 0 0 0 0";
if (AttackType == "imap")
    MatrixRow = "0 0 0 1 0 0 0 0 0";
if (AttackType == "pod")
    MatrixRow = "0 0 0 0 1 0 0 0 0";
if (AttackType == "phf")
    MatrixRow = "0 0 0 0 0 1 0 0 0";
if (AttackType == "dict")
    MatrixRow = "0 0 0 0 0 0 1 0 0";
if (AttackType == "rootkit")
    MatrixRow = "0 0 0 0 0 0 0 1 0";
if (AttackType == "normal_alert")
    MatrixRow = "0 0 0 0 0 0 0 0 1";
```

وبالتالي ستكون صيغة ملف المعطيات الهدف لتدريب الشبكة العصبونية هو كالتالي (جزء من الملف):

```
010000000
000010000
001000000
000000100
000010000
000000001
```

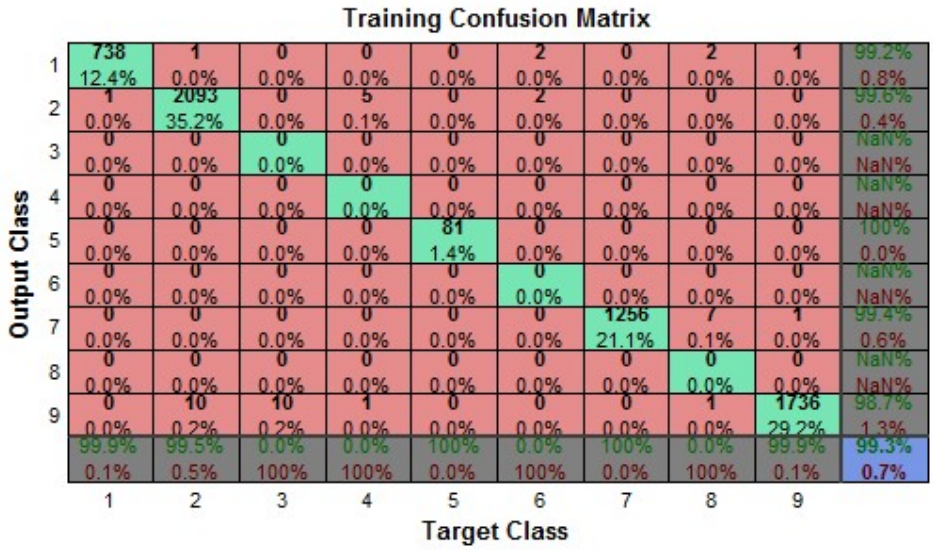
من الواضح واستناداً أن السطر الأول من العينة يدل على أن السطر المقابل له من ملف التدريب يمثل تنبيه عن هجوم من نوع back والسطر الثالث يمثل تنبيه عن هجوم من نوع land أما السطر الأخير فيشير وضوحاً إلى أن السطر المقابل له في مجموعة معطيات التدريب يمثل تنبيه خاطئ إيجابي ونصنفه في الصف normal_alert.

عند تدريب الشبكات متعددة الطبقات، اقتضت الممارسة العملية أولاً تقسيم البيانات إلى ثلاث مجموعات فرعية. المجموعة الفرعية الأولى هي مجموعة التدريب، والتي تستخدم لحساب التدرج وتحديث أوزان شبكة والانحيازات.

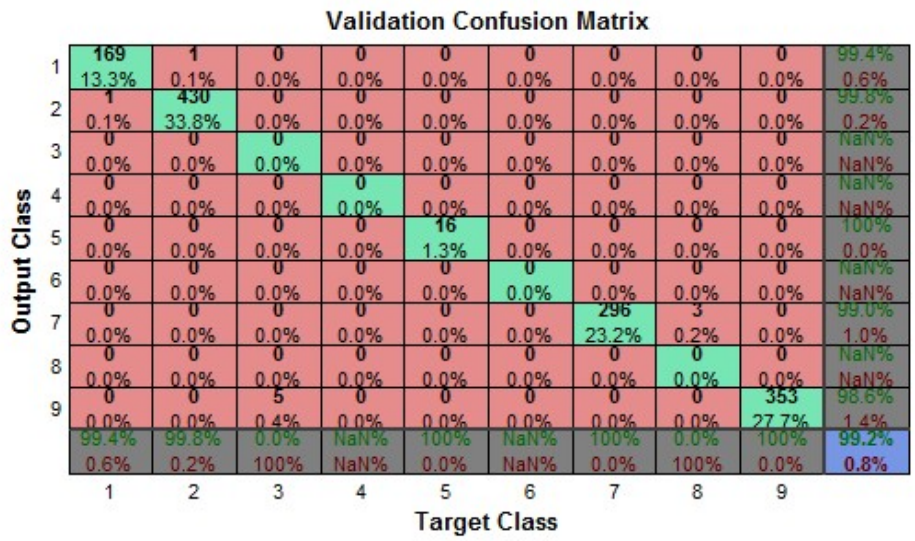
المجموعة الفرعية الثانية هي مجموعة التحقق من الصحة. يتم فيها رصد الخطأ والتحكم به خلال عملية التدريب. خطأ التحقق ينخفض عادة خلال المرحلة الأولى من التدريب، كما الخطأ في مجموعة التدريب. ومع ذلك، عندما تبدأ الشبكة بتصنيف خاطئ للبيانات، فإن الخطأ في التحقق من الصحة عادة ما يبدأ في الارتفاع. عندها يتم حفظ أوزان الشبكة والانحيازات في الحد الأدنى من الخطأ لممجموعة التحقق.

المجموعة الفرعية الثالثة هي لاختبار الشبكة العصبونية. مع ذلك سنقوم بإجراء اختبار منفصل للشبكة العصبونية بجزء من مجموعة التنبيهات التي حصلنا عليها.

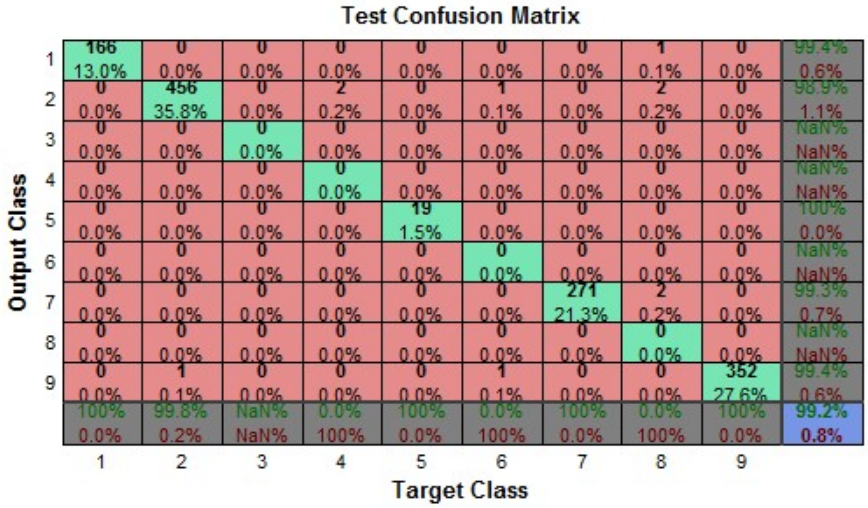
سنعرض فيما يلي مصفوفات الالتباس بعد تدريب الشبكة العصبونية على البيانات المعالجة بطريقة UR (التدريب، التحقق، الاختبار على الترتيب) للشبكة العصبونية بعد عملية التدريب ب 85 عملية تكرارية على نسبة 70% من معطيات الدخل، والتحقق على نسبة 15% منها والاختبار الذاتي على نسبة 15% منها.



الشكل 6.10 مصفوفة الالتباس لمرحلة تدريب الشبكة العصبونية



الشكل 6.11 مصفوفة الالتباس في مرحلة التحقق



الشكل 6.12 مصفوفة الالتباس في مرحلة الاختبار.

بنظرة سريعة على مصفوفات الالتباس الثلاثة السابقة نلاحظ عدم الدقة في تصنيف العينات التي تنتمي للصفوف3(land)و5(imap)و6(phf)و8(rootkit). ممكن أن نعزو ذلك إلى قلة عينات التدريب التي تنتمي فعلياً لهذه الصفوف.

في مصفوفة الالتباس في مرحلة الاختبار نجد أنّ نسبة تصنيف التنبيهات الايجابية الخاطئة (الصف 9 normal_alert) بشكل صحيح هي 99.7%.

كما ذكرنا سنقوم بإجراء اختبار منفصل للشبكة العصبونية باستخدام نسبة 30% من البيانات التي حصلنا عليها بعد معالجة التنبيهات أي 2591 تنبيه صحيح ايجابي وقمنا بإضافة 5259 تنبيه خاطئ إيجابي وتم تمرير مجموعة الاختبار هذه إلى الشبكة العصبونية لتعطينا مصفوفة النتيجة، قمنا باستخدام #c بكتابة برنامج اختبار بسيط ليعطينا بالتفصيل نتائج هذا الاختبار.

سنعتمد المعايير التالية في اختبار أداء المصنف لتصنيف التنبيهات الايجابية الخاطئة:

- ✓ خطأ التصنيف: هو عدد التنبيهات الكلي المصنفة بشكل خاطئ.
- ✓ معدل خطأ التصنيف: هو النسبة المئوية لعدد التنبيهات الكلي المصنفة بشكل خاطئ.
- ✓ خطأ التصنيف التدريبي = النسبة المئوية لعدد التنبيهات المصنفة بشكل خاطئ في مرحلة تدريب الشبكة العصبونية إلى عدد التنبيهات المستعملة في التدريب.

- ✓ معدل خطأ التصنيف التدريبي = النسبة المئوية للمعدل السابق.
- ✓ دقة التصنيف = 100 - معدل خطأ التصنيف.
- ✓ خطأ تصنيف الخطأ الايجابي = عدد التنبيهات الايجابية الخاطئة المصنفة بشكل خاطئ.
- ✓ معدل خطأ تصنيف الخطأ الإيجابي = النسبة المئوية للمعدل السابق من عدد التنبيهات الايجابية الخاطئة الكلي.
- ✓ نسبة تقليل الخطأ الإيجابي = 100- معدل الخطأ الايجابي.

	UR	IUR
خطأ التصنيف (الإجمالي)	87	41
معدل خطأ التصنيف (%)	1.1	0.52
معدل خطأ التصنيف التدريبي (%)	2	0.8
دقة التصنيف (%)	98.9	99.48
خطأ تصنيف الخطأ الايجابي	19	21
معدل خطأ تصنيف الخطأ الإيجابي (%)	0.36	0.39
نسبة تقليل الخطأ الإيجابي (%)	99.63	99.60

الجدول 6.1 أداء المصنف في تصنيف التنبيهات الإيجابية الخاطئة باستخدام طريقتي المعالجة UR وIUR.

وفي ما يتعلق بأداء المصنف في تصنيف الهجمات المذكورة سابقاً حصلنا على النتائج التالية:

	back	pod	Phf	land	nmap	Imap	Rootkit	dict
العدد الاجمالي للهجمات	1283	49	3	4	461	3	7	781
عدد التصنيفات الخاطئة	7	0	3	4	0	3	7	0
النسبة المئوية	0.54	0	100	100	0	100	100	0

الجدول 6.2 نتائج تصنيف الهجمات المعالجة باستخدام UR

dict	Rootkit	Imap	Nmap	land	Phf	Pod	Back	
781	7	3	461	4	3	49	1283	العدد الاجمالي للهجمات
0	7	3	0	4	3	1	2	عدد التصنيفات الخاطئة
0	100	100	0	100	100	2.04	0.15	النسبة المئوية

الجدول 6.3 نتائج تصنيف الهجمات المعالجة باستخدام IUR

6.8 استخدام النظام

قيمنا ببناء تطبيق Console Application نقوم فيها بتمرير ملف التنبيهات بالصيغة القياسية الصادر من branyard2 في بيئة ubuntu كدخل للبرنامج. ليقوم التطبيق بالاتصال ببيئة ماتلاب وتمرير التنبيهات إلى الشبكة العصبونية المختارة (UR أو IUR) وإرجاع نتائج تصنيف التنبيهات وإلحاقها بملف التنبيهات الدخل كملف جديد.

الفصـــل الســـابع: خاتمـــة وآفـــاق مستقبلية

في هذا البحث قمنا ببناء نظام تصنيف تنبيهات معتمد على الشبكات العصبونية قادر على تصنيف تنبيهات معينة صادرة عن نظام كشف الاختراق snort بدقة عالية وتمييز التنبيهات الإيجابية الخاطئة من التنبيهات الصحيحة السلبية وبالتالي توفير جهد كبير على مسؤول الأمان في الشبكة للتحقق من صحة جميع التنبيهات الصادرة.

يشكل هذا البحث خطوة كبيرة في مجال دراسة ارتباط التنبيهات الايجابية الخاطئة وبالتالي ممكن أن يشكل نواة بحث أكبر في مجال ارتباط التنبيهات والبحث عن السبب الجذر لإصداره والعمل على إزالته. على سبيل المثال تتبع معرفات التوقيعات ومراجعاتها (خصوصاً مع الحجم الكبير والمتزايد من تواقيع الهجمات المحدثة على مدار الساعة) وتعديلها بشكل تلقائي يشكل مساحة بحث جديدة في هذا المجال.

المراجع.

[1] Al-Mamory, S., & Zhang, H. (2007). A survey on IDS alerts processing techniques.

[2] Anderson, J. P. (1980). Computer security threat monitoring and surveillance.

[3]Axelsson, S. (1999). The base-rate fallacy and its implications for the intrusion detection.

[4]Bellowin, S. M. (1992). There be dragons. In Proceedings of the 3rd USENIX Security Symposium.

[5]C. Clifton, G. G. (2000). Developing custom intrusion detection filters using data mining

[6] Chan, M. V. (2003). An analysis of the 1999 /Lincolin Laboratory evaluation data.

[7] Hebb, D. (1949). The Organization of Behavior.

[8] K. Julisch. (2003). Using Root Cause Analysis to Handle Intrusion Detection Alarms.

[9] K. Julisch. (2001). “Mining alarm clusters to improve alarm handling efficiency”,

[10] Kohonen, T. (1997). Self-Organized Maps.

[11] Law, K. H., & Kwok, L. F. (2006). IDS False Alarm Filtering Using KNN Classifier.

[12] Mitchel, T. M. (1997). Machine Learning.

[13] Mokarian, A., Faraahi, A., & Delavar, A. G. (n.d.). False Positives Reduction Techniques in Intrusion Detection A-Review 2013.

[14] NIST. (2004). ICAT Metabase. Web page at http://icat.nist.gov/.

[15] Paxson, V. (1998). A system for detecting network intruders in real-time.

[16] Peter Clark, T. N. (1989). The CN2 induction algorithm. Machine Learning.

[17] Pietraszek. (2005). Defending against injection attacks.

[18] Pietraszek, T. (2006). ALERT CLASSIFICATION TO REDUCE FALSE

POSITIVIES IN IDS.
[19] Ptacek, T. H., & Newsham, T. N. (1998). Insertion, evasion and denial of service.
[20] R. Vaarandi. (2009). “Real-time classification of IDS alerts with data mining techniques”.
[21] Roesch, M. (2005). SNORT. The Open Source Network Intrusion System.
[22]Sekar, R., & Guang, Y. (1999). A high-performance network intrusion detection system.
[23] Tangent., D. (2001). DEF CON 9. Web page at http://www.defcon.org/html/defcon-9/defcon-9-post.html, .
[24] S Terry Brugger and Jedidiah Chow, " An Assessment of the DARPA IDS Evaluation Dataset Using Snort".