مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (37) العدد (5) 2015
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (37) No. (5) 2015

دراسة ومقارنة أنظمة كشف الاختراقات المفتوحة المصدر

بشرى ديوب*

(تاريخ الإيداع 4 / 6 / 2015. قُبِل للنشر في 10/ 9 / 2015)

□ ملخّص □

ظهرت أنظمة كشف الاختراقات Intrusion Detection Systems IDS، من أجل زيادة وتطوير الأمن في الشبكات، وأصبحت فعالة لحماية الشبكات الداخلية internal networks من الهجمات الخارجية، واتخاذ الإجراءات المناسبة ضد المخترقين intruders. كما تستخدم أنظمة كشف الاختراق تقنيات من أجل جمع معلومات عن الهجوم، ومن الممكن استخدام هذه المعلومات كدليل ضد المهاجم. تستخدم أنظمة كشف الاختراقات طرائق مختلفة في عملية الكشف، فبعضها يستخدم التواقيع في الكشف signature based، وبعضها يكشف الشذوذ anomaly based، وغيرها من الطرائق.

يقارن هذا البحث التقنيات المستخدمة في أنظمة كشف الاختراق، ويركز على الأنظمة التي تستخدم التواقيع في عملية الكشف، وبالأخص النظامين snort و Bro، وهما من الأنظمة المفتوحة المصدر open source، ومقارنة الإنذارات التي يطلقها النظامان عند تطبيق أداة توليد الهجمات IDSWakeup.

الكلمات المفتاحية: أمن الشبكات، أنظمة كشف الاختراقات، الأنظمة المفتوحة المصدر، الأنظمة المعتمدة على الكشف بالتواقيع.

* ماجستير – قسم الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين–اللاذقية– سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (37) العدد (5) 2015
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (37) No. (5) 2015

A Study and Comparison of Open Source Intrusion Detection Systems Study

Bushra Dayoub*

(Received 4 / 6 / 2015. Accepted 10 / 9 / 2015)

□ ABSTRACT □

With the recent advances in the field of network security, a technique called Intrusion Detection System IDS is developed to further enhance and make network secure. It is a way by which we can protect our internal network from outside attack, and can take appropriate action if needed. Using intrusion detection methods, information can be collected from known types of attack and can be used to detect if someone is trying to attack the network. Many techniques are there to detect intrusion in a network like signature matching, anomaly based and others.

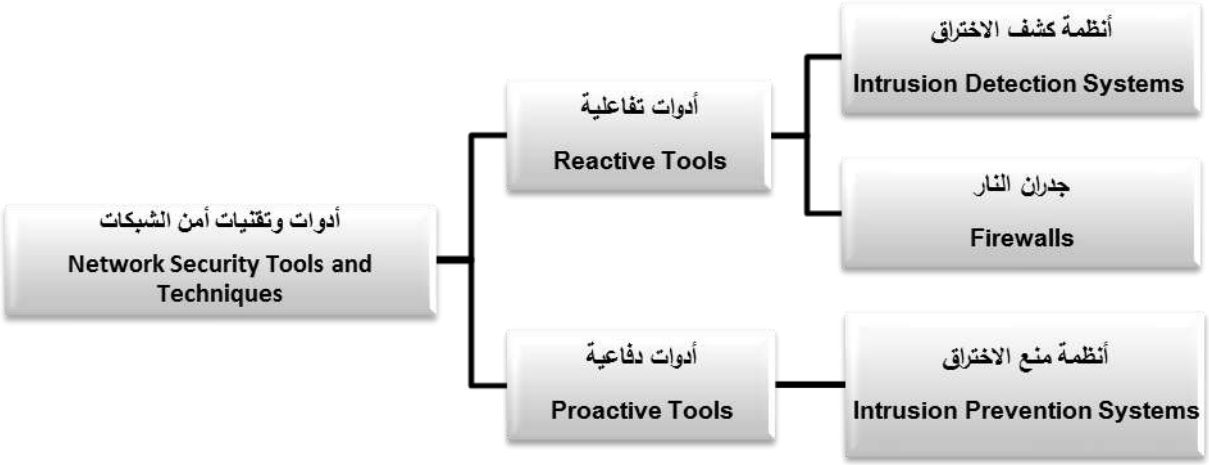
The work presented here studies and compares the techniques used by intrusion detection systems, and focuses on the signature matching technique. It discusses the open source, free intrusion detection system Snort. Another open source intrusion detection system Bro is also discussed. It compares these systems alarms against the open source tool IDSWakeup.

Keywords: Network security, Intrusion Detection System, Open Source, Snort, Bro, IDSWakeup, Signature-based IDS

* **Master, Department of Computers and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria**

مقدمة

أصبحت مسألة أمن الشبكات من أهم الأمور التي تشغل بال المؤسسات والشركات، وخاصة مع ازدياد معرفة ودهاء المخترقين والمهاجمين الذين تمكنوا من تنفيذ العديد من المحاولات الناجحة لتعطيل شبكات مؤسسات ضخمة، وإيقاف مواقع ويب مهمة. هناك العديد من التقنيات والأدوات المستخدمة في أمن الشبكات، ومن الممكن تصنيفها إلى أدوات تفاعلية Reactive، وأدوات دفاعية Proactive (الشكل 1)، الأدوات التفاعلية نقوم بتوليد نوع من الاستجابة أو الإنذار بعد الكشف عن الأفعال المريبة، وتعتبر أنظمة كشف الاختراق IDS وجدار النار firewall من هذه الأدوات، والأدوات الدفاعية تقوم بعمليات منع بالزمن الحقيقي لمحاولات الاختراق التي يتم الكشف عنها.



الشكل (1) موقع أنظمة كشف الاختراق من أدوات وتقنيات أمن الشبكات

أهمية البحث وأهدافه

تتعرض شبكات المؤسسات ومواقع الويب للكثير من الهجمات ومحاولات الاختراق، بهدف التخريب أو الحصول على المعلومات، وتلعب أنظمة كشف الاختراقات دوراً مهماً في عملية كشف وتتبع المخترقين، وتزداد أهميتها مع ازدياد خبرة وتطور الهجمات والمهاجمين. يهدف البحث إلى:

- توضيح أنواع أنظمة كشف الاختراق وتصنيفاتها والمقارنة بين هذه الأنواع.
- دراسة ومقارنة نظامي الكشف مفتوحي المصدر Bro و Snort من حيث البنية والخصائص.
- إعداد بيئة لاختبار الإنذارت التي يطلقها النظامان عند استخدام أداة توليد هجمات مفتوحة المصدر هي IDSwakeup.

أنظمة كشف الاختراق Intrusion Detection systems

تستخدم أنظمة برمجية أو عتادية لإدارة مجموعة أجهزة حواسيب أو شبكة معينة، تقوم هذه الأنظمة بجمع معلومات من أجزاء مختلفة من الشبكة ، من أجل استخدام هذه لمعلومات للكشف عن الأفعال المريبة ، والتي قد تكون اختراقات (هجمات من خارج الشركة)، أو سوء استخدام misuse (هجمات من داخل الشركة أو المؤسسة) [1].

يعد الكشف عن الاختراق الهدف الأساسي لأنظمة كشف الاختراق، أثناء الاختراق وليس بعد انتهائه، وإنذار الشخص المسؤول عن أمن الشبكة، بـإرسال بريد إلكتروني أو بإطلاق إنذار من نوع ما . الهدف الثاني لأنظمة كشف

الاختراق هو جمع المعطيات من النظام، وتسجيل كل الأحداث الهامة، وتحديد مصدر الهجوم، وهذه المعطيات تستخدم لأهداف قانونية كدليل أو إثبات ضد المهاجم.

تعريف و تصنيف أنظمة كشف الاختراق

يمكن تعريف **كشف الاختراق** [2]: بأنه عملية معرفة و تحديد الاستخدام المؤذي أو غير المشروع، سوء الاستخدام، تجاوز الحد في استخدام أنظمة الحواسب.

يمكن تصنيف أنظمة الكشف [3]:

- **حسب مصدر المعلومات التي تراقبها و تحللها إلى :**
  - أنظمة كشف شبكية Network-Based IDS: يراقب النظام مقطع كامل من الشبكة.
  - أنظمة كشف على المضيف Host-Based IDS: تراقب هذه الأنظمة المضيف الموضوعة عليه.

يقارن الجدول (1) المنهجين من حيث الإيجابيات والسلبيات، من الواضح أن التقنيتين مكملتان لبعضهما البعض ومن الجيد استخدام مزيج من الاثنين من أجل رفع احتمالية كشف الاختراقات.

الجدول (1) مقارنة أنظمة أنظمة كشف شبكية Network IDS مع أنظمة كشف على المضيف Host IDS

	أنظمة كشف على المضيف Host-Based IDS	أنظمة كشف شبكية Network-Based IDS
الحسنات	• يستطيع تحليل ما يقوم به التطبيق. • يمكنه التأكد من نجاح أو فشل الاختراق. • يكشف الاختراقات التي لا تحتاج وجود شبكة. • لا نحتاج إلى عتاديات hardware إضافية. • لا تتأثر في حال الشبكات الموصولة بمبدلات switched networks.	• لا تؤثر على أداء أجهزة الشبكة. • غير مرئية أبداً من قبل أجهزة الشبكة. • يمكنها مراقبة أكثر من مضيف في الوقت نفسه. • أكثر مقاومة لمحاولات التلاعب. • يمكنها كشف الاختراقات الشبكية التي من غير الممكن كشفها من وجهة نظر جهاز واحد.
السيئات	• من الصعب إدارتها لأنه يجب جمع و إدارة المعلومات على كل جهاز. • يجب تنصيبها على كل جهاز على الشبكة. • تؤثر على أداء النظام من الممكن التلاعب به.	• يجب أن تلم بكمية كبيرة من التفاصيل. • يجب أن تكون سريعة جداً • صعبة التركيب و الإعداد. • مشكلة قنوات الاتصال المشفرة.

- **حسب طريقة التحليل و الكشف إلى:**
  - أنظمة كشف الشذوذ Anomaly-Based: تعتمد على تعريف ما هو السلوك الطبيعي أو المسموح به في النظام و من ثم الإعلام عن أي فعل أو حدث يقع خارج حدود المسموح أو الطبيعي.
  - أنظمة الكشف المعتمدة على التواقيع Signature-Based IDS، أو أنظمة كشف سوء الاستخدام Misuse-Based: النموذج الذي كُتب لتوصيف السلوك السيئ هو الفرضية الأساسية، بعد ذلك يقوم النظام بمقارنة تسلسل المعلومات مع هذا النموذج ليقرر ما هو جيد و ما هو خبيث.

يقارن الجدول (2) المنهجين، تحتاج الأنظمة المعتمدة على التواقيع لصيانة وتعديل بشكل دائم للتعرف على الاختراقات الجديدة، في حين تستطيع أنظمة كشف الشذوذ الكشف عن الاختراقات الجديدة.

الجدول (2) مقارنة أنظمة Signature-Based IDS مع Anomaly Based IDS

	أنظمة الكشف المعتمدة على التواقيع Signature-Based IDS	أنظمة كشف الشذوذ Anomaly-Based IDS
الحسنات	• معدل كشف أعلى و إنذارات خاطئة أقل. • لا تحتاج إلى طور تعلم. • من الصعب تجاوزها. • تزود بمعلومات أكثر عن الاختراق.	• لا تحتاج إلى صيانة مستمرة • يمكنها كشف الاختراقات الجديدة.
السيئات	• تحتاج إلى تعديل دائم. • لا تستطيع كشف الاختراقات الجديدة.	• تصدر إنذارات خاطئة أكثر. • لا تعطي معلومات عن الاختراق. • من الصعب كتابة نموذج دقيق. • من السهل تخطيها.

أنظمة الكشف المعتمدة على التواقيع Signature-based IDS

تعرف أيضاً هذه الأنظمة باسم أنظمة كشف سوء الاستخدام misuse detection، وهذه الأنظمة تعتمد على المعرفة knowledge، هناك العديد من الطرق المستخدمة في تصميم أنظمة الكشف المعتمدة على التواقيع، وهي: الأنظمة المعتمدة على مخطط الحالة state modeling system، الأنظمة الخبيرة expert system، الأنظمة المعتمدة على مخططات بتري الملونة colored Petri nets، الأنظمة المعتمدة على القواعد rule based systems.

الأنظمة المعتمدة على القواعد Rule-based systems

تفتقر هذه الأنظمة إلى قوة الأنظمة الخبيرة، ولكنها تتميز بوضوح نماذج الاختراق، كل اختراق يتم توصيفه بقاعدة واحدة تأخذ الشكل التالي

$(\text{condition}_1 \Lambda \text{condition}_2 \Lambda \ldots \text{condition}_x) \rightarrow \text{actions}$

يعتبر النظام أن هناك اختراق عندما تكون نتيجة تقييم كل الشروط true، وعندها يجب القيام بالأفعال المحددة بـ actions، ويتم تفعيل القاعدة من أجل الرد على هذا الاختراق [3].

نظام Snort

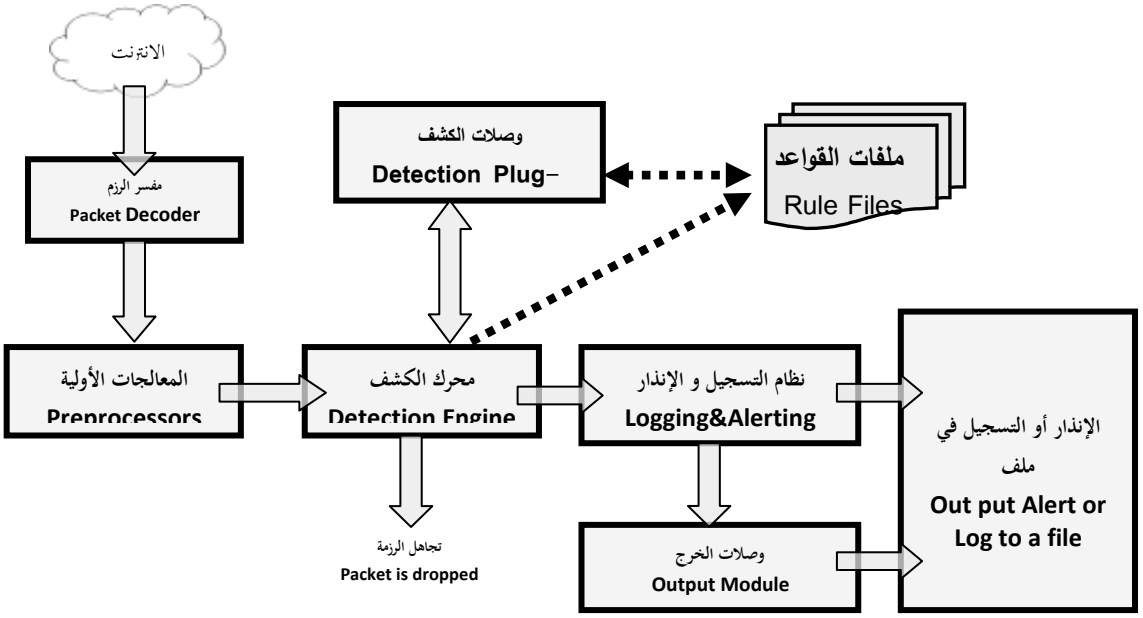
يعتبر هذا النظام من الأنظمة الشبكية التي تؤمن الوقاية من الاختراق Network Intrusion Prevention System (NIPS) و نظام كشف للاختراقات الشبكية Network Intrusion Detection System (NIDS) قادر على تحليل وتسجيل الرزم المتبادلة **[4]**.

يقوم Snort بعملية تحليل للبروتوكولات، وعمليات مطابقة وبحث matching\searching في محتويات الرزم packet payload، وتم استخدامه بشكل أساسي من أجل الكشف (دون الرد) عن عدد كبير من الهجمات وعمليات السبر scan.

البنية الداخلية Internal Structure

يوضح الشكل (2) المكونات الجزئية التي يمكن تقسيم Snort وفقها، كل مكون يقوم بالتالي **[5]**:

- **التقاط الرزم Packet Capture:** هذا الجزء مبني بشكل أساسي على الحزمة البرمجية libpcap
- **مفسر الرزم Packet Decoder:** يحول الرزم التي تم التقاطها إلى بنية معطيات مناسبة وتعرف البروتوكولات المستخدمة في طبقة الاتصال الأولى. ومن ثم تنتقل إلى بروتوكولات الطبقة التي بعدها، فتفكك رموز IP، ثم TCP أو UDP (حسب البروتوكول المستخدم) من أجل الحصول على المعلومات المفيدة مثل أرقام المنافذ وعناوين الاتصال. ويمكن إطلاق إنذارات في هذه المرحلة في حال الكشف عن ترويسات رزم غير نظامية (مثل حقول ترويسة بأطوال كبيرة).
- **المعالجات الأولية Preprocessors:** من الممكن تشبيهها بمرشحات filters من نوع خاص، التي تقوم بتعريف بعض الأمور التي يجب فحصها لاحقاً في الجزء التالي والذي هو محرك البحث (Detection Engine)، مثل محاولات الاتصال المريبة إلى بعض منافذ TCP/IP Ports أو في حال تم إرسال عدد كبير من رزم TCP SYN (وهي رزم تكون قيمة بت SYN فيها مساوية للقيمة 1، وتُرسل عند بدء تأسيس اتصال TCP فقط) خلال فترة قصيرة (تعتبر مسح للمنافذ port scan).
- **ملفات القواعد Rule Files:** ملفات نصية تحتوي على قائمة القواعد التي تمت كتابتها بسياق محدد لتحديد تواقيع الاختراقات، من خلال نوع البروتوكول، عناوين IP، أرقام المنافذ، محتوى الرزمة ..
- **وصلات الكشف Detection Plug-ins:** هي أجزاء يتم الإشارة إليها من خلال تعريفها في ملفات القواعد. تستخدم لكي تقوم بتعريف النماذج patterns في كل مرة يتم فيها تقييم القاعدة.
- **محرك الكشف Detection Engine:** يقوم باستخدام وصلات الكشف، لمطابقة الرزم مع التواقيع التي تم تحميلها في الذاكرة عند بدء تشغيل Snort.
- **وصلات الخرج Output Plug-ins:** هي الأجزاء التي تمكن من تقديم التحذيرات وفق شكل محدد (إنذار alert أو سجل log) لكي يتمكن المستخدم من الوصول إليها بأي طريقة (عن طريق شاشة عرض، أو من ملف، أو من قاعدة بيانات) .



الشكل (2) مكونات نظام snort

بنية قواعد Snort

تتألف القاعدة من جزأين أساسيين: الرأس head و الخيارات options [6]، الشكل (3).

الشكل (3) بنية قواعد Snort

- **الرأس:** يتكون من الحقول التالية (بالترتيب):

○ الفعل Action : يحدد هذا الحقل الفعل الذي سيتخذه النظام عند تحقق هذه القاعدة، فهي تؤثر إذاً على خرج النظام.

○ البروتوكول protocol: يستخدم لتطبيق القاعدة على بروتوكول محدد (IP, TCP, UDP..).

○ عنوان IP للمصدر و الوجهة Source and destination IP : يحددان الطريق الذي تسلكه الرزم التي يجب أن تطبق عليها القاعدة.

○ أرقام المنافذ للمصدر والوجهة source and destination ports : وهذين الحقلين يستخدمان للبروتوكولات TCP,UDP .

○ الاتجاه direction: رمز للتعبير عن جهة الرزمة التي ستطبق القاعدة عليها.

- **خيارات القاعدة :** من الممكن تصنيف خيارات القواعد في Snort إلى أربع أنواع أساسية [6]:

○ خيارات عامة General Rule Options: تعطينا معلومات عن القاعدة ولكن لا تؤثر في عملية المطابقة، مثل msg و sid و classtype.

○خيارات الكشف في المحتوى Payload Detection Rule Options: تبحث عن معطيات داخل محتوى الرزمة، مثل content و offset وdistance

○خيارات الكشف التي لا تنظر في المحتوى Non-Payload Detection Rule Options: تبحث عن بيانات غير موجودة في المحتوى، مثل flow أوisdataat أو ack أو ttl .

○خيارات الكشف الخلفية Post-Detection Rule Options: تحدد ما يجب إطلاقه بعد عملية المطابقة، مثل session أو resp أو react أو tag .

ميزات Snort

تلخص أهم مزايا نظام Snort بالنقاط التالية [7]:

○من الممكن تنصيب Snort على أي حاسب على الشبكة.

○يزود Snort بمجموعة تواقيع مجربة وموثقة بشكل جيد.

○قابلية الحمل Portable: من الممكن تنصيبه على أنظمة تشغيل مختلفة (Linux, Windows, MacOS , XSolaris, BSD, IRIX, Tru64, HP-UX).

○من الممكن إعداد Snort ليعمل كنظام منع اختراقات Intrusion Prevention System (IPS).

سيئات Snort

على الرغم من كون Snort أشهر أنظمة الكشف والمنع IDS/IPS فإن لديه بعض السيئات هي [7]:

- قواعد المعطيات الخاصة بالتواقيع ضخمة جداً، على سبيل المثال، يحوي Snort على أكثر من 1000 توقيع لرزم http، وبالتالي هناك عملية معالجة طويلة لمطابقة هذه الرزم.
- تعتبر مراقبة الرزم في الشبكات الكبيرة مهمة مكلفة.
- يفشل في كشف الرزم المقطعة fragmented packets في شبكات السرعات العالية (أكثر من 5Gbps) [8].

نظام Bro

كتب النظام الأصلي من قبل Vern Paxson، وهو نظام مفتوح المصدر [9]، يعمل على نظام Unix، وهو نظام كشف اختراق شبكي (Network Intrusion Detection System) NIDS، يقوم بمراقبة الرزم على الشبكة بحثاً عن الأفعال المريبة، ويكشف الاختراق عن طريق تطبيق تحليل لغوي أو إعراب parsing على الرزم من أجل استخراج دلالات طبقة التطبيقات ومن ثم تطبيق محلل مرتبط بالأحداث، يقوم هذا المحلل بمقارنة الفعل مع نماذج تحدد الأحداث المزعجة.

يملك Bro لغة خاصة policy language تمكن المستخدم من إضافة عمليات جديدة للنظام، وذلك حسب التغير الذي قد يحدث في سياسة الموقع الذي يستخدم النظام، أو في حال تم الكشف عن اختراقات جديدة. في حال تم الكشف عن حدث مريب، فمن الممكن الاختيار بين إطلاق إنذار في الزمن الحقيقي،أو تسجيل الحدث في ملفات السجل. تم بناء النظام من أجل العمل ومواكبة الشبكات ذات السرعات العالية جداً (Gbps)، ومن أجل الكشف عن عدد كبير جداً من الاختراقات، باستخدام تقنيات ترشيح الرزم، ويستطيع أن يصل إلى الأداء الأفضل عند تشغيله على أي حاسب شخصي PC، بالتالي يمكن استخدامه كأداة فعالة من أجل مراقبة اتصالات الانترنت على موقع معين [9].

صمم Bro من أجل المهتمين بالحصول على المرونة العالية، وقابلية التعديل بما يتناسب مع متطلبات المستخدم المتغيرة دائماً. ويقوم بالوقوف في وجه الهجمات الموجهة ضده، كونه يحتوي على العديد من محللات البروتوكول.

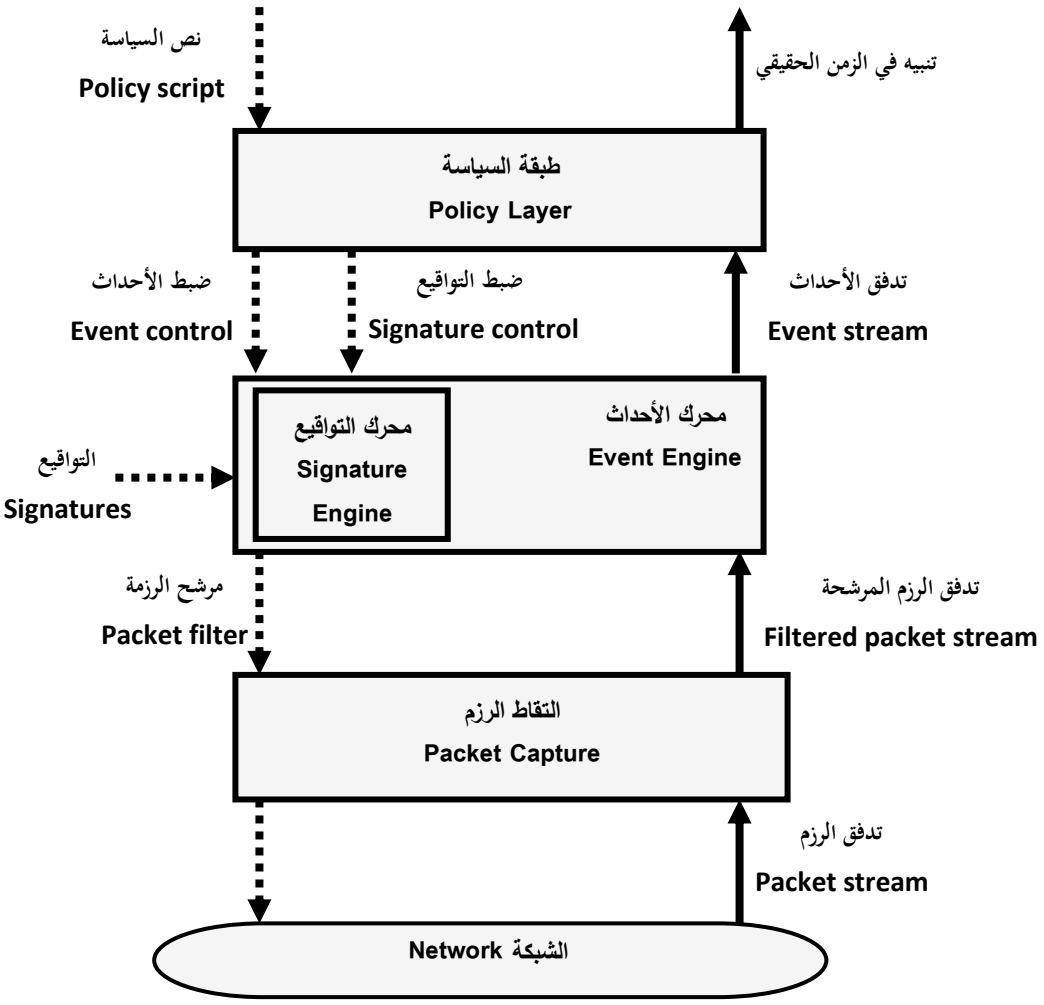
البنية الداخلية

تبدو البنية الداخلية للنظام من النظرة الأولى مشابهة لبنية Snort، ولكن في الحقيقة يختلف النظامان اختلاف كبير في الطرق المستخدمة في التحليل، يعتمد Snort في تحليله على الرزمة كوحدة أساسية، في حين يعالج Bro الاتصال بكل رزمه كوحدة متكاملة، يحتفظ Bro بحالة افتراضية لكل اتصال مفتوح على الشبكة [10].

تعالج الطبقة السفلية الكمية الأكبر من المعطيات، وكلما ارتفعنا في الطبقات، يقل حجم المعطيات التي يتم معالجتها، مما يتيح إمكانية زيادة المعالجة على عناصر المعطيات.

على الرغم من أن النظامين Snort و Bro، مزودان بطرق من أجل المطابقة باستخدام السياق، إلا أن Bro مزود بطرق من أجل استخدام السياق في المستوى الأدنى low-level context، وهذا يتم باستخدام التعابير المنتظمة من أجل مطابقة النماذج، واستخدام آخر للسياق في مستوى أعلى high-level context عن طريق الاستفادة من محاسن محلل البروتوكول في Bro و لغة النصوص scripting language، والشيء المميز في Bro أنه في حال تم الكشف عن حالة تطابق مع نموذج من النماذج فمن الممكن أن نكتب توقيع لتحديد الاستجابة التي نريدها على هذا الحدث. الشكل (4) يوضح البنية التفصيلية للنظام. وهي:

- **التقاط الرزم Packet Capture**: كما في Snort، يقوم Bro بالتقاط الرزم باستخدام مكتبة libpcap [10].
- **محرك الأحداث Event Engine**: تطبق هذه الطبقة العديد من أنواع الفحص للتأكد من صحة الترويسة في كل الرزم وأنها لا تخالف التعريفات الأساسية للبروتوكولات، وفي هذه المرحلة يُعاد تجميع رزم IP التي جُزئت من قبل، مما يمكن محللات البروتوكول فيما بعد من تحليل الرزمة الأساسية بالكامل، وترسل هذه الطبقة الأحداث التي تكشفها إلى طبقة السياسة.
- **محرك التواقيع Signature Engine**: يقوم بفحص محتوى الرزم، ويولد حدثاً في كل مرة يتم فيها تطابق رزمة من الرزم مع توقيع معرف لديه، هذه الأحداث تتم معالجتها من قبل طبقة السياسة فيما بعد.
- **طبقة السياسة Policy Layer**: يقوم مترجم نصوص السياسة بتنفيذ النصوص التي تمت كتابتها بلغة خاصة بنظام Bro، وهذه النصوص تحدد معالج الأحداث event handler الذي سيُربط مع الحدث المولد من قبل طبقة محرك الأحداث، يقوم معالج الأحداث بتنفيذ أي أوامر نصية من أجل توليد أحداث جديدة، مثل تسجيل رسائل تذكير بالزمن الحقيقي للحدث، أو تخزين البيانات على القرص الصلب [10].



الشكل (4) بنية نظام Bro

التواقيع signatures

أهم العقبات التي يواجهها Bro مع Snort، هو كون Snort يتم تحديث تواقيعه بشكل دائم و مستمر، ولكن من الممكن أن يقوم المستخدم بكتابة التواقيع الخاصة به باستخدام لغة Bro الخاصة، وهناك تقنيات تمكن من تحويل تواقيع Snort ليتمكن Bro من إعادة استخدامها، وذلك باستخدام نص تمت كتابته بلغة python، لكن فيما بعد قام Snort بتطوير بنية القواعد التي يطبقها لكي تقوم هذه القواعد بالاستفادة من المعالجات الأولية الموجودة في النظام، مما صعب عملية التحويل بين النظامين.

نصوص السياسة policy scripts

يُعد نص السياسة الذي يُكتب في Bro المحلل الأساسي الذي يستخدم من أجل اتخاذ القرار فيما إذا كان الحدث الذي تم كشفه يستحق الإنذار عنه، وتحدد هذه النصوص أيضاً الأحداث التي يجب تطبيقها للرد على المهاجم، وطريقة الإنذار عن الأحداث [9].

مزايا Bro

من الممكن تلخيص مزايا Bro بما يلي [7]:

- يقوم Bro بإعادة تجميع الرزم قبل الوصول إلى محرك الأحداث، عملية إعادة التجميع في هذا المستوى يمكن Bro من كشف الاختراقات المخفية بسبب TCP segmentation.
- يستطيع القيام بعمليات فحص عميقة ضمن الرزم على مستوى التطبيق application، وبالتالي يستطيع تحليل محتويات الملف الذي يتم تبادله من قبل بروتوكول طبقة التطبيق.
- يستطيع القيام بعمليات كشف وتحليل الأنفاق Tunnel، (بما فيها Ayiya, Teredo, GTPv1)، حيث يقوم بفك تغليف النفق ومن ثم تحليل محتوياته وكأن النفق غير موجود.

سيئات Bro [7]

- يحتاج Bro إلى بيئة UNIX ، ويعمل على أنظمة Linux, FreeBSD, Mac OS فقط
- يرسل Bro التقارير إلى ملفات فقط، ولا يملك واجهة رسومية.

مقارنة نظامي Snort و Bro

مما سبق يمكننا المقارنة بين Snort و Bro من عدة نقاط، يلخص الجدول (3) المقارنة بين النظامين. [11] [7]

الجدول (3) مقارنة Snort مع Bro

Snort	Bro	المعامل Parameter
لا	نعم	التواقيع المعتمدة على السياق Contextual signatures
بشكل متوسط	بشكل كبير	المرونة في الضبط Flexible site customization
بشكل متوسط	بشكل كبير	العمل على الشبكات ذات السرعات العالية High speed network capability
نعم	لا	مجتمع مستخدمين كبير Large user community
نعم	لا	واجهة رسومية للإعداد Configuration GUI
كثير	قليل	واجهة تحليل رسومية Analysis GUI

التنصيب والنشر Installation /deployment	صعب	سهل
التوافق مع أنظمة التشغيل Operating System compatibility	UNIX	أي نظام
الرخصة License	BSD License	GNU GPL v.2
العمل كنظام منع اختراقات IPS feature	نعم	لا

منهجية البحث

تعتمد الفلسفة المسيطرة في مجال اختبار أنظمة الكشف على إنشاء بيئات اختبار منفصلة عن بعضها البعض، مما يعني البحث عن هجمات موجودة ومعروفة ومن ثم يتم تضمين هذه الهجمات في الرزم الاصطناعية التي يتم توليدها وبثها على البيئات أو الشبكات المنفصلة التي يتم اختبارها، ويجب التحكم بحجم وكمية الرزم والهجمات التي تبث على الشبكة مما يمكننا من معرفة متى سيبدأ نظام الكشف بمواجهة مشاكل بالقدرة على مواكبة سرعة الشبكة، وبالتالي معرفة كمية الرزم التي سيقوم النظام بإهمالها في حالة سرعات الشبكات العالية.

توصيف الاختبار

يعتمد الاختبار على بناء رزم شبكية بالاعتماد على تواقيع اختراقات معروفة، حيث يتم بناء رزمة شبكية تحوي على توقيع اختراق واحد، يتم بعدها إرسال هذه الرزم على الشبكة سندرس خرج النظامين، وكمية الإنذارات التي ستطلقها عند تشغيل الأداة، وفي النهاية سنعرض الاستنتاجات التي حصلنا عليها من الاختبار، ومقارنة أداء النظامين.

تم الاعتماد في الاختبار على أداة IDSWakeup [14] وهذه الأداة من الأدوات الهامة في عملية اختبار أنظمة الكشف، لأنها تقوم بتوليد رزم تحوي هجمات مزيفة، تشبه إلى حد كبير الهجمات الفعلية التي يواجهها نظام كشف الاختراقات الشبكية.

نتائج عملية اختبار أنظمة الكشف تعتمد بشكل كبير على نسخة برنامج snort التي تم استخدامها في عملية الاختبار وبالتالي على مجموعة القواعد المستخدمة من قبل النظام، وكذلك تعتمد على الإعدادات وطريقة ضبطها، وإعدادات المعالجات الأولية، والوصلات المستخدمة plug-in ونظام التشغيل المستخدم وسرعة الشبكة المستخدمة، ومواصفات بيئة العمل مثل سرعة المعالج.

IDSwakeup

هي مجموعة من الأدوات التي تم بناؤها من أجل اختبار أنظمة كشف الاختراق، مكتوبة بلغة برمجة مفسر الأوامر shell script، تقوم بتوليد هجمات مزيفة، تشبه إلى حد كبير الهجمات المعروفة من أجل كشف ما إذا كان النظام سيطلق إنذار عليها، [14]. تستخدم أدوات أخرى هي hping, iwu . تشكل IDSwakeup نقطة البداية حيث تمكن المستخدم من اختيار الهجمات التي يرغب في استخدامها، استخدمت hping من أجل عملية توليد الرزم، و iwu ترسل رزم من الممكن التحكم فيها مثل تغيير عنوان المرسل والمستقبل و حقل TTL.

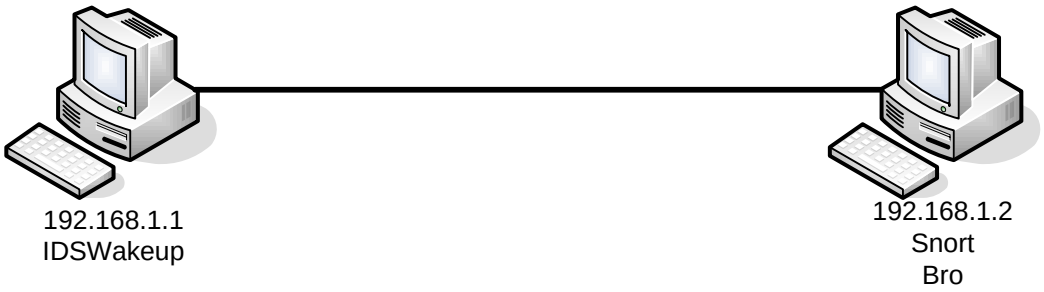
تُشغل الأداة بتحديد عنوان IP للمرسل <src add>، وعنوان IP للمستقبل <dst add>، وتحديد عدد النسخ المراد إرسالها من كل اختراق [nb]، وتحديد قيمة حقل TTL في رزم IP التي سيتم إرسالها [ttl]:

#./IDSWakeup <src add> <dst add> [nb] [ttl]

إعدادات الشبكة المستخدمة في عملية الاختبار

عناوين الشبكة المستخدمة في عملية الاختبار من الصنف C class C وعنوان الشبكة 192.168.1.0، والعناوين المستخدمة هي 192.168.1.1 للجهاز الذي يشغل IDSWakeup، و192.168.1.2 للجهاز الذي يشغل Snort و Bro.

في حالة الشبكات التي تعتمد على مبدلة Switch فيجب أن يتم وصل جهازي Snort و Bro على منفذين في المبدلة switch من نوع span port، لكي يتم تمرير جميع الرزم التي تصل إلى المبدلة إلى هذين المنفذين، حتى لو لم تكن موجهة إليها، لكن لا تدعم جميع المبدلات هذا النوع من المنافذ. يبين الشكل (5) مخطط الشبكة التي تم استخدامها في عملية الاختبار.



الشكل (5) مخطط شبكة الاختبار

مواصفات الأجهزة المستخدمة في الاختبار:

- معالج Intel® Celeron® CPU 570 @ 2.26 GHz
- الذاكرة 1 GB of RAM
- نظام التشغيل openSUSE 11 with i586 architecture

يلخص الجدول (4) الأدوات المستخدمة، نسخة كل أداة ونظام التشغيل المستخدم.

الجدول (4) الأدوات المستخدمة في الاختبار

الأداة	نظام التشغيل	النسخة	تاريخ النشر
IDSWakeup	openSUSE	1.0	2001
Snort	openSUSE	2.8.4	2009
Bro	openSUSE	1.4	2008

إعدادات أدوات الاختبار

نُصب Snort ليتعامل مع mysql لتخزين الإنذارات في قاعدة معطيات تم إنشاؤها بشكل يتناسب مع بنية Snort، ومُنح النظام السماحيات المناسبة ليقوم بالتعديل على قاعدة المعطيات.

ضُبطت إعدادات Snort بما يلائم متطلبات الشبكة، عن طريق التعديل في ملف الإعدادات snort.conf. وشُغل Snort من خلال مفسر الأوامر Shell ليقوم بالتقاط الرزم وإصدار الإنذارات وتخزينها في قاعدة المعطيات المخصصة للإنذارات. وعُدلت قواعد Snort بما يتناسب مع الأداة IDSWakeup المستخدمة، من أجل الكشف عن الاختراقات المولدة من قبل IDSWakeup. نُصب Bro بالإعدادات الافتراضية من الحزمة البرمجية، وتم تشغيله من خلال أوامر Shell. ضُبطت إعدادات IDSwakeup لتقوم بتوليد رزم تحوي تواقيع جميع الاختراقات المضمنة فيها، وشغلت في بيئة سطر الأوامر shell بما يناسب إعدادات شبكة الاختبار.

النتائج والمناقشة

عند تشغيل IDS wake up استلم snort 297 رزمة، ولم يتم رمي أي رزمة من الرزم وتم توليد 59 إنذار.

توزعت الإنذارات على النحو التالي:

- 18 إنذار تم تصنيفها بأنه غير مصنف unclassified
- 5 غير معروفة bad – unknown، أو انها badtraffic
- 36 إنذار المتبقية كانت مقسمة على الشكل التالي:

5 web application – attack

1 shell code – detect

11 attempted – recon

8 attempted – dos

5 web – application – activity

1 misc – attack

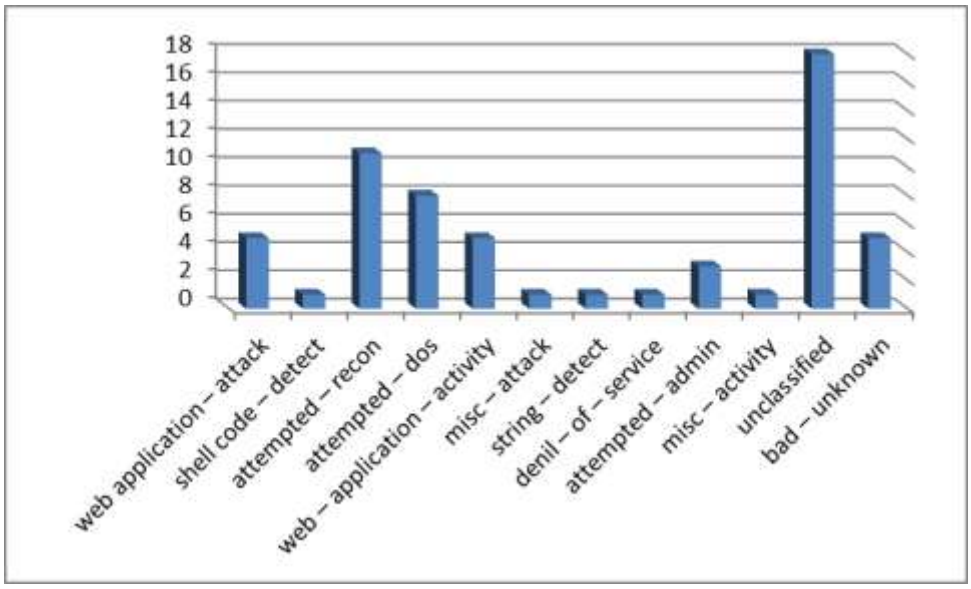
1 string – detect

1 denil – of – service

3 attempted – admin

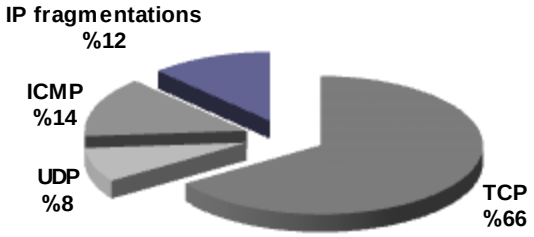
1 misc – activity

يبين المخطط (1) توزع الإنذارات التي أطلقها Snort عند تشغيل IDSWakeup.



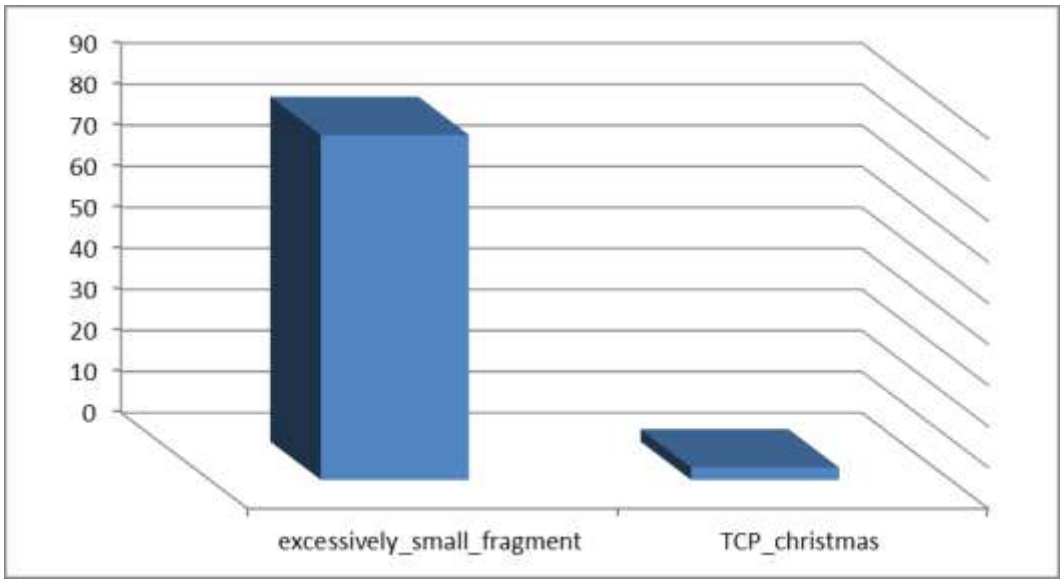
المخطط (1) توزع إنذارات Snort على IDSWakeup

ويبين المخطط (2) توزع الرزم التي ولدتها IDSwakeup حسب البروتوكول.



المخطط (2) توزع الرزم المستلمة حسب البروتوكول

أما بالنسبة لنظام Bro فقد تم توليد 87 إنذار صُنفت في نوعين فقط توزعت كما في المخطط 3.



المخطط (3) – توزع إنذارات Bro على IDSWakeup

الاستنتاجات والتوصيات

1. الميزة الأساسية في Bro هو احتفاظه بمعلومات عن حالة كل اتصال، ويستخدم هذه المعلومات لاحقاً من أجل الكشف عن النشاطات المريبة.

2. يستخدم Bro لغة خاصة به من أجل كتابة نصوص السياسة، وتتمتع هذه اللغة بالسهولة، مما يمكن المستخدم من كتابة نصوص السياسة الخاصة به بسهولة.

3. يستخدم Bro التعابير المنتظمة regular expression من أجل القيام بعملية المطابقة، في حين يستخدم snort محرك تعابير منتظمة متوافقة مع لغة Perl يسمى pcre engine (Perl Compatible Regular Expression).

4. عملية تطوير وتعديل تواقيع snort أسهل بكثير من تطوير تواقيع Bro.

5. يحتاج التعامل مع نظام Bro خبرة في التعامل مع نظام UNIX

6. يعتبر Bro نظام كشف مناسب للشبكات ذات السرعات العالية، ولكن لايدعم واجهة تعامل رسومية ويعتبر صعبا في التنصيب والإعداد.

7. نظام Snort أسهل في التنصيب والإعداد، ومن الممكن تنصيبه على أغلب أنواع أنظمة التشغيل، ولكنه غير مناسب للعمل على الشبكات ذات السرعات العالية.

8. من الممكن تطوير البحث ليتم مقارنة النظامين مع أدوات اختبار أخرى أخرى، و من الممكن الانطلاق من المشروع والاختبار الذي تم عرضه من أجل اختبار أنظمة كشف اختراق أخرى غير التي تم عرضها.

9. يمكن تطوير بيئة الاختبار بإضافة نظام snort ولكن بتنصيبه على نظام windows من أجل دراسة أداء هذا النظام عند تغيير نظام التشغيل، وتأثير ذلك على أدائه.

المراجع:

[1] SARMAH, A. *Intrusion detection systems definition, need and challenges*, October 3 2001,1 May. 2015.

<https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343>

[2] MUKHERJEE, B.; HEBERLEIN, L. T.; LEVITT, K. N. *Network intrusion detection*. IEEE Network, 1994. 1 May. 2015.

<http://wenke.gtisc.gatech.edu/ids-readings/network_id.pdf>

[3] AXELSSON, S. *Intrusion Detection Systems: A Taxomomy and Survey*. Dept. of Computer Engineering, Chalmers University of Technology, Sweden, 14 March 2000.

[4] ROESCH, M. *Snort – Lightweight Intrusion Detection for Networks*, 1999, 1 May. 2015. < https://www.usenix.org/event/lisa99/full_papers/roesch/roesch.pdf>

[5] REHMAN, R. U. *Intrusion Detection Systems with Snort:Advanced IDS Techniques with Snort, Apache,MySQL, PHP, and ACID*. 1st. Ed. Prentice Hall PTR, New Jersey, 2003, 275.

[6] SNORT® Users Manual 2.9.7, The Snort Project, October 13 2014, 1 May. 2015.

< http://manual.snort.org/>

[7] AMBATI, S. B.; VIDYARTHI, D. "*A Brief Study And Comparison Of Open Source Intrusion Detection System Tools*". International Journal of Advanced Computational Engineering and Networking , 2013

[8] FU, T. " *An Analysis of Packet Fragmentation Attacks vs. Snort Intrusion Detection System*". International Journal of Computer Engineering Science (IJCES), May 2012.

[9] PAXSON, V.; ROTHFUSS J.; TIERNEY, B. *Bro:Quick Start Guide* version 0.9, 2004, DRAFT. 1 May. 2015.

<http://www.gnu-darwin.org/www001/src/ports/security/bro/ work/bro-1.2.1/doc/user-manual/Bro-user-manual.pdf>

[10] SEEBERG,V. E. *BRO - an IDS - Intrusion Detection and Prevention*. Dec. 2005.

[11] MEHRA, P. *A brief study and comparison of Snort and Bro Open Source Network Intrusion Detection Systems*. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2012.

[12] Snort website, 1 May. 2015. <http://www.snort.org>

[13] Bro website, 1 May. 2015. <http://www.bro-ids.org>

[14] SCHAUER, H. *IDSwakeup*, 2002, 1 May. 2015. <http://www.hsc.fr/ressources/outils/idswakeup/index.html.en>