2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing

Privacy Violation Classification of Snort Ruleset

Nils Ulltveit-Moe, Vladimir Oleshchuk
University of Agder
Servicebox 422
N-4604 Kristiansand
{nils.ulltveit-moe, vladimir.oleshchuk}@uia.no

***Abstract*—It is important to analyse the privacy impact of Intrusion Detection System (IDS) rules, in order to understand and quantify the privacy-invasiveness of network monitoring services. The objective in this paper is to classify Snort rules according to the risk of privacy violations in the form of leaking sensitive or confidential material. The classification is based on a ruleset that formerly has been manually categorised according to our PRIvacy LEakage (PRILE) methodology. Such information can be useful both for privacy impact assessments and automated tests for detecting privacy violations. Information about potentially privacy violating rules can subsequently be used to tune the IDS rule sets, with the objective to minimise the expected amount of data privacy violations during normal operation. The paper suggests some classification tasks that can be useful both to improve the PRILE methodology and for privacy violation evaluation tools. Finally, two selected classification tasks are analysed by using a Naïve Bayes classifier.**

Keywords: IDS, rules, privacy violation, classification

I. Introduction

It is important to analyse the privacy impact of Intrusion Detection System (IDS) rules, in order to understand and quantify how privacy invasive use of network intrusion detection systems can be. The objective in this paper is to classify Snort rules according to the risk of privacy violations in the form of leaking sensitive or confidential information. This can subsequently be used to enhance the data privacy handling of intrusion detection systems and also to tune IDS rule sets to minimise the amount of data privacy violations that can be expected during normal operation. Such classification can be useful for example for corporate Privacy Ombudsmen to quantify the potential privacy impact of a given IDS rule set during privacy impact assessments [1]. It can also be used for automated test tools to estimate the privacy impact of a given IDS rule set.

We use the PRIvacy LEakage (PRILE) methodology introduced in [2] for manual categorisation of the Snort community ruleset. This paper extends the analysis by investigating some indicators in PRILE that hopefully can be used both to improve the methodology and for automatic classification of Snort rules.

This paper is organised as follows: The next section covers an introductory data analysis which gives motivation for the classification problems. Section 3 gives a brief introduction the PRILE methodology and how we performed manual categorisation of Snort rules. Section 4 describes the analysis part, which points out some classification tasks and analyses two of the tasks by using a Naïve Bayes classifier. Section 5 discusses related work. Section 6 contains conclusions and future work.

Term	Occurrences
porn	21
p2p	19
yahoo	12
multimedia	10
talk	9
streaming	4
sex	6
google	5
vp-asp	2
weather	2

Table I
Top ten terms only in rules with significant impact on privacy.

Term	Occurrences
overflow	743
netbios	430
icmp	120
xss	114
rpc	128
injection	92
indexu	80
backdoor	80
exploit	148
web-attacks	51

Table II
Top ten words only in attack detection rules with significant impact on security.

II. Introductory data analysis

We performed an introductory data analysis of the IDS rules by storing them in a relational database and then used data mining to analyse them. This was useful both to understand the classification problems covered in this paper better, and to get more insight into how the data had been classified in the PRILE case study [2]. It was also useful as quality assurance of the manually classified data, in order to detect errors and inconsistencies in the data. We did amongst others do an introductory term frequency analysis of the classified data. Some results from this introductory analysis are discussed in the next section as motivation for using reinforcement based machine learning for classifying the data and also for giving a broader understanding of what a privacy violation consists of.

1066-6192/10 $26.00 © 2010 IEEE
DOI 10.1109/PDP.2010.87

Tables I and II show the top ten terms only occurring in the alert message field of Snort rules for privacy violating and attack detection rules respectively. The tables contains two columns:

- The first column describes the *Term* or word;
- and the second column describes the number of rules that this term *Occurs* in.

The privacy violating rules in Table I trigger on actions that can be considered inappropriate by corporate computer security policies, like surfing for pornography (21 rules) or sex (6 rules). Many rules are devoted to monitoring peer-to-peer (P2P) traffic (19 rules). It also includes commonly used services provided by Yahoo (12 rules), Google (5 rules), weather services (2 rules) or the VP-ASP shopping cart program (2 rules). There are also several rules that can detect downloading of multimedia files (10 rules) and streaming of such files (4 rules). The multimedia rules include detecting common multimedia file extensions like *.smi, .rt, .rp, .rmp, .ram, .wmf* and *.emf*. Even though there is only one rule for each of these multimedia file types, the rules have a very broad scope and report on any activity involving such files on a specific port or a range of ports.

Table II shows terms for security related rules. These rules are designed to detect attacks on critical infrastructure. Typical attack terms include buffer overflow (743 rules), Cross-Site Scripting (XSS) (114 rules) SQL injection (92 rules) backdoor (80 rules), exploit (148 rules) and web-attacks (51 rules). In addition, there are many terms related to attacks on commonly used services or critical infrastructure like Netbios (430 rules), RPC (128 rules) and IndexU (80). ICMP attack related and informational messages are also included here (120 rules). It is not expected that ICMP messages will have a significant privacy impact, and they are useful for network managers to detect abnormal network events reported, like for example server or network unavailability (ICMP destination unreachable).

This shows that there clearly are some terms in the alert message field indicating that some IDS rules are policy rules focused at identifying illegal use according to some IT policy, whereas other rules are related to detecting attacks on hosts or other network units. The next section will go more in detail on PRILE indicators that are required to perform automatic classification of IDS rules.

III. Brief Introduction to PRILE

The Snort ruleset we base our classification experiments on, was categorised manually based on our PRILE methodology for IDS rules. The methodology and summary results of the manual categorisation of the Snort rules is described in [2]. Snort was selected as our case, because it is a popular IDS system used by large public and private organisations, and it contains a comprehensive set of IDS rules available for free. We presume that the PRILE methodology can be used as a gold standard for categorising IDS rules and that our manual classification follows the methodology sufficiently well. It should however be noted that a larger manual study involving PRILE evaluations from several experts is needed in order to provide more objective results. We used the Snort community ruleset containing 3669 rules [3]. This paper aims to investigate some classification tasks in PRILE, which are useful both to improve the methodology and as a first step towards implementing automatic tools for testing privacy leakage from IDS rules.

The aim of our evaluation methodology is to provide a gold standard for evaluating the privacy impact of IDS rules. Privacy leakage is defined as the fraction $p = \frac{r}{s}$ of exposed communication sessions r that are not attack related to all communication sessions s. An IDS rule signature R is considered to consists of two main parts:

- a protocol specific part used to address a specific message in a given protocol;
- and an attack distinguishing part, which aims at matching one or more attack patterns as described by a software vulnerability.

In our experiments, we have chosen a maximum exposure percentage of $p = 1\%$. This exposure percentage can be measured directly for a given IDS case or it can be estimated for a rule set based on knowledge about the size of the attack distinguishing pattern x in bytes, the maximum amount of payload inspected per session b_{max} and a measured or estimated occurrence frequency f of the attack distinguishing pattern in ordinary data sessions. The formula for calculating the maximum exposure percentage is then $p = \frac{b_{max} f}{x}$. As a rule of thumb, attack distinguishing patterns should at least cover 3 bytes, and not be a commonly used term, as indicated in [2]. Furthermore, we have defined a 5 level scale for privacy leakage (PRILE) that focuses on how wide scope the privacy violation has. The privacy leakage scale is defined below:

0 *None* - no privacy leakage expected from the IDS rule.

1 *Vulnerability* - the IDS rule models attacks based on a known *vulnerability* in a specific way. This means that the IDS rule can be expected to expose less than a given percentage *p* of all user sessions being investigated by it.

2 *Program file* - more than *p* percent of all sessions targeted at a given *program file* as part of an application are being monitored.

3 *Application* - more than *p* percent of all sessions targeted at a given *application* or *service* are being monitored. An *application* is presumed to consist of several program files.

4 *Platform* - more than *p* percent of all sessions targeted at a given *platform* are being monitored. For example monitoring of specific files or file types across all services for a given operating system.

5 *Policy* - The IDS rule is applied on *network-wide level* and is not necessarily relevant from a security perspective. It is defined to monitor or control usage of services being monitored. For example, monitoring use of end-user services like chat, instant messaging, VoIP, email or web.

Furthermore, the PRILE methodology gives different priority to privacy and security for different service classes:

- *End-user* and *system services*: Must not be privacy leaking.
- *System administration* and *unexpected* services: Security has priority over privacy, i.e. may be privacy leaking.

The enumerated scale from 0 to 5 can then be used for quantitative measurements of privacy invasiveness. Furthermore, we define a *privacy violating rule* as a *rule that leaks more information than level 1*. That means that level 2, 3, 4 and 5 IDS rules are privacy violating by definition.

We have also defined some qualitative test criteria for the PRILE methodology, which goes beyond what we are able to present in this short paper. For a more thorough introduction to PRILE, see [2].

The PRILE methodology presumes that it somehow is possible to differentiate between the *protocol specific part* that typically is repeated for every normal data session and the *attack specific* part(s) of an IDS rule's detection patterns, which presumably only triggers during attacks.

IV. Analysis

A. Privacy Leakage Level Classification

No privacy leakage (level 0) can for example occur for rules detecting protocol violations or denial of service attacks that are not expected to happen from normal user behaviour. Rules categorised as level 0 in the PRILE test include traffic on port 0, loopback traffic on the internet, same source and destination IP, IP reserved bit set, SYN to multicast address, other protocols than TCP and UDP in IP datagrams, undefined ICMP code, empty UDP packets and UDP flooding. These rules *detect anomalistic behaviour*. It should be possible to detect such behaviour, for example based on knowledge about limitations and boundaries of the TCP/IP protocol suite, or by using some kind of learning system.

Vulnerability specific rules (level 1) can be expected to expose less than a given percentage p of all sessions being monitored by it. This can be measured directly for a given IDS system, provided that a representative input data set exists. If that does not exist, then it is possible to estimate p based on knowledge about the *size* of the attack specific pattern and knowledge or presumptions of the occurrence frequency of the attack specific pattern in the input data. It would therefore be very useful to be able to automatically classify parts of the IDS rule pattern as either *attack specific* or *protocol specific*.

Privacy violating rules for *program file, application* or *platform* respectively (levels 2,3 and 4) means that the rule is privacy violating. It exposes more than p percent of all sessions. The classification task is then to determine whether the rule belongs to either of these three categories.

Policy specific rules will for example not be security relevant in the sense that they aim at detecting a specific vulnerability. The problem is therefore how to differentiate between *policy specific IDS rules* and *attack rules*.

This discussion has identified the following classification tasks in PRILE that need further investigation:

1) How to differentiate between *policy specific rules* and *attack rules*.
2) How to classify the service type of an IDS rule into the groups *System administration, System service, End-user* and *Unexpected*.
3) How to detect rules with *no privacy leakage*, which means how to identify protocol and service anomalities.
4) How to identify the *attack specific* part(s) of an IDS rule's detection patterns.
5) How to classify the scope of an attack as either *program file, application* or *platform*.

We only cover the first two classification tasks in this short paper. Future work involves a more comprehensive study.

B. Classification as policy specific rules or attack rules

Potential indicators of a policy specific rule (PRILE level 5) are:

- Lacks authoritative external references to vulnerability databases like Common Vulnerabilities and Exposures (CVE)[1] or Bugtraq[2] in the *reference:* attribute of policy specific rules.
- The alert message (*msg:*) attribute may indicate policy specific rules, as discussed in section II.
- The *classtype:* attribute in Snort rules may indicate policy specific rules.

This initial study will use a Naïve Bayesian classifier, in order to investigate some of the promising indicators. We chose a Bayesian classifier for these introductory experiments, because it is a simple classifier that often works well [4]. A feature extractor extracts a feature set from the potential indicators listed above.

The rule pattern is not considered in this simple analysis, but may be included in a more comprehensive study later. The manually categorised rule set was used as a gold standard. We divided the set of rules by random, uniform selection into two parts - a *training* data set and an *experiment set*, each containing 1834 rules. We did not use a tuning data set, to get as many policy rules as possible included in each sample. This is because the level 5 policy rules are sparsely distributed in the rule set. Also, we did not perform any ad-hoc changes to the feature extractor that would warrant using a separate tuning data set.

The training and experiment sets were randomly reshuffled between the tuning sessions to avoid testing for idiosyncracies [5]. We used the Natural Language Toolkit (NLTK) to classify the data sets [6].

The experiments show that contrary to our expectations, the *references:* rule attribute is a poor classifier for policy violating IDS rules. Although the overall accuracy seemed high (97.0%), all the level 5 policy specific rules were misclassified. The high accuracy can be explained by the low occurrence frequency of policy specific rules (only 114 rules), which means that if all of them were misclassified, this would still give an accuracy of around 97%. Also, even though most of the external references were empty for the policy rules (98 out of 114 policy rules) and the remaining 16 rules were URL

[1] http://cve.mitre.org

[2] http://www.securityfocus.com/archive/1

	Attack rule	Policy rule
Attack rule	1770.0 ± 6.0	6.3 ± 1.7
Policy rule	8.0 ± 2.1	49.7 ± 5.3

Table III
CONFUSION MATRIX FOR CLASSIFICATION OF POLICY SPECIFIC RULES USING THE *classtype:* ATTRIBUTE.

	End-user	Sysadm.	Syst. srv.	Unexp.
End-user	1129.9 ± 13.3	0.0	12.7 ± 4.0	1.4 ± 0.8
Sysadm.	70.6 ± 5.6	35.7 ± 3.8	9.7 ± 2.8	1.1 ± 0.7
Syst. srv.	0.1 ± 0.2	0.1 ± 0.7	490.3 ± 11.5	0.7 ± 0.8
Unexp.	0.3 ± 0.4	1.2 ± 3.5	59.5 ± 7.4	20.7 ± 4.6

Table IV
CONFUSION MATRIX FOR SERVICE GROUP CLASSIFICATION USING PORT NUMBER AS CLASSIFIER.

references to non-authoritative sources, the problem is that a significant amount of attack rules also lack external references (717 rules) which confuses the classifier. Similar tests ruled out using words in the alert message (*msg:*) field as a useful classifier, since it was not able to identify any policy specific rules correctly.

However, the Snort *classtype:* attribute works better. It manages to achieve an accuracy of 99%. The confusion matrix in Table III shows average number and standard deviation of correctly and incorrectly classified rules from an ensemble of 100 experiments. The row in the table refers to the gold standard and the column refers to test values. The Naïve Bayes classificator manages to on average correctly identify 49.8 of 57.7 sampled policy rules (86% precision) and misclassifies 8 of the policy rules as attack rules (14%). On average 6.3 of the sampled attack rules are misclassified as policy rules (89%).

The most informative feature of the classtype attribute was, perhaps not surprisingly, the class *policy-violation*. The sampled set of policy violating rules was too small to conclude on the ranking of other classtypes containing policy violating rules. However several other classes contain some rules manually categorised as policy rules. It would be an advantage for the precision of this classifier to have a larger set of IDS rules to train on.

C. Service group classification

This test aims at classifying the service type of an IDS rule into the groups *System administration, System service, End-user* and *Unexpected*. Potential indicators of type of service include:

- Sender and receiver port number in the rule;
- Alert message (*msg:*) attribute of the rule.

Our experiments show that the port number is a significantly better service group classifier than the alert message. The port number classifier gives an accuracy of 91%, whereas words in the alert message field only gives 86%. The confusion matrix for service group classification using a Naïve Bayes classifier is shown in Table IV. The row in the table refers to the gold standard and the column refers to test values. The table is based on average and standard deviation from an ensemble of 100 experiments. The table shows that End-user services are classified with a precision of 99%, but are sometimes also wrongly classified as a System service (1%).

The classifier does not work so well for System administration services, which only has a precision of 31%. System administration services are most often confused with End-user services (60%) and occasionally also System service (8%) and Unexpected (1%). The main reason for this, is that a service definition only based on port-numbers is not able to correctly classify system administration services running over ports that are normally used by End-user or System services. An example of a service that would be wrongly classified, is web-based system administration or application management interfaces.

The next classifier recognises System services precisely (99.8%). However, rules for Unexpected services are very unprecise with only 25% correctly classified. They are in 73% of the cases wrongly classified as System services and in the rest of the cases either as System administration or End-user services. Rules for Unexpected services target backdoors, bad traffic, DDOS attacks, exploits, shellcode and BOT traffic. One possible explanation is that this is a deficiency in the feature extractor. If a port that for example is not assigned by IANA is being used, then perhaps it would be better to only consider this port instead of both ports, since the other port may be a system service. It could also be related to the methodology or manual classification, so this needs further analysis.

V. RELATED WORK

This paper is an extension of an earlier work where we developed the PRILE methodology and did a case study where the Snort community ruleset was manually evaluated [2]. New contribution in this paper is an analysis of different classification problems from the PRILE methodology and experimental analysis of two of these classification problems by using the existing case study as a gold standard. There is as far as we know no other attempts at classifying the privacy leakage from IDS rules. There are however similar scoring systems in the security domain, for example The Common Vulnerability Scoring System (CVSS), which is an industry standard metric for the characteristics and impacts of IT vulnerabilities [7]. This score is useful to indicate the security relevance of a given IDS rule. It has also got a confidentiality indicator which measures the level of potential confidentiality loss from a vulnerability. However, CVSS does not cover the potential confidentiality loss that can occur from IDS monitoring activities.

One slightly related area, is anomality-based privacy intrusion detection using learning algorithms like dynamic bayesian networks [8]. However this conceptual system aims at detecting internal attacks on databases for stealing large amounts of data, which is quite different from our objectives.

Another related area is privacy enhanced intrusion detection systems. The BRO IDS [9] for example supports a way to anonymise the payload of a packet instead of removing the entire payload [10, 9].

There also exists some earlier work on privacy-enhanced host-based IDS systems that pseudonymises audit data and performs analysis on the pseudonymised audit records [11, 12, 13, 14, 15, 16]. However neither of these go into detail on how the privacy impact of existing IDS rule sets can be reduced.

VI. Conclusion

This paper presents work towards classification of privacy leakage in IDS rules. The PRILE methodology is a general methodology that should be applicable to any IDS sytem. We have identified five general classification tasks that should be analysed in the PRILE methodology and have analysed two of them based on a Naïve Bayes classifier. A limitation is that the analysis in this paper is Snort specific. However some of the results may still be transferrable to other IDS systems, for example the port based classifier for service categories. The paper also identifies the classtype attribute of Snort rules as a good indicator for discriminating between attack rules or privacy violating rules.

The underlying methodology, PRILE, is still in its infancy, so a larger study involving more experts would be required to achieve an objective view of what consists a significant privacy leakage from IDS systems, and to agree upon a set of common quantitative and qualitative indicators of privacy leakage. This also involves doing a more extensive analysis to cover the remaining classification tasks. This opens up the possibility for automated test tools that can estimate the privacy impact of a given IDS rule set.

Acknowledgments: This work is funded by Telenor Research & Innovation under the contract DR-2009-1.

References

[1] International Organization for Standardization, “Iso 22307:2008 financial services – privacy impact assessment,” http://www.iso.org/iso/catalogue_detail?csnumber=40897, 2008.

[2] N. Ulltveit-Moe and V. Oleshchuk, “PRIvacy LEakage Methodology (PRILE) for IDS Rules,” *Proceedings of the PrimeLife/IFIP Summer School 2009, Springer*, To be published.

[3] M. Roesch et. al, “Community rules for Snort current,” http://www.snort.org/pub-bin/downloads.cgi/Download/comm_rules/, Accessed June 2009.

[4] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” in *MACHINE LEARNING*, 1997, pp. 131–163.

[5] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O’Reilly, June 2009.

[6] Steven Bird and Edward Loper and Ewan Klein, “Natural language toolkit,” http://www.nltk.org.

[7] P. Mell, K. Scarfone, and S. Romanosky, “CVSS a complete guide to the common vulnerability scoring system version 2.0,” http://www.first.org/cvss/cvss-guide.pdf, 2007.

[8] X. An, D. Jutla, and N. Cercone, “Privacy intrusion detection using dynamic bayesian networks,” in *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*. Fredericton, New Brunswick, Canada: ACM, 2006, pp. 208–215.

[9] Lawrence Berkeley National Laboratory, “Bro intrusion detection system,” http://bro-ids.org.

[10] R. Pang and V. Paxson, “A high-level programming environment for packet trace anonymization and transformation,” in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. Karlsruhe, Germany: ACM, 2003, pp. 339–351. [Online]. Available: http://portal.acm.org/citation.cfm?id=863994

[11] T. Holz, “An efficient distributed intrusion detection scheme,” in *COMPSAC Workshops*, 2004, pp. 39–40.

[12] M. Sobirey, S. Fischer-Hübner, and K. Rannenberg, “Pseudonymous audit for privacy enhanced intrusion detection,” in *Proceedings of the IFIP TC11 13th International Conference on Information Security (SEC’97)*, May 1997, pp. 151–163.

[13] S. Fischer-Hübner, *IDA - An Intrusion Detection and Avoidance System (in German)*. Aachen, Shaker, 2007.

[14] M. Sobirey, B. Richter, and H. König, “The intrusion detection system aid - architecture and experiences in automated autid trail analysis,” in *Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security*, 1996, pp. 278–290.

[15] R. Büschkes, D. Kesdogan, “Privacy enhanced intrusion detection,” in *Multilateral Security in Communications, Information Security*, G. Müller and K. Rannenberg, Eds. Addison Wesley, 1999, pp. 187–204.

[16] U. Flegel, *Privacy-Respecting Intrusion Detection*, 1st ed. Springer, Oct. 2007.