PRIvacy LEakage Methodology (PRILE) for IDS Rules

Nills Ulltveit-Moe, Vladimir Oleshchuk

PRIvacy LEakage Methodology (PRILE) for IDS Rules

Nils Ulltveit-Moe and Vladimir Oleshchuk

University of Agder
Servicebox 509
NO-4898 Grimstad, Norway
{Nils.Ulltveit-Moe,Vladimir.Oleshchuk}@uia.no

Abstract. This paper introduces a methodology for evaluating PRIvacy LEakage in signature-based Network Intrusion Detection System (IDS) rules. IDS rules that expose more data than a given percentage of all data sessions are defined as privacy leaking. Furthermore, it analyses the IDS rule attack specific pattern size required in order to keep the privacy leakage below a given threshold, presuming that occurrence frequencies of the attack pattern in normal text are known. We have applied the methodology on the network intrusion detection system Snort's rule set. The evaluation confirms that Snort in its default configuration aims at not being excessively privacy invasive. However we have identified some types of rules rules with poor or missing ability to distinguish attack traffic from normal traffic.

Keywords: IDS, rules, privacy impact, methodology, privacy violation

1 Introduction

One of the largest threats towards on-line security and privacy today is attacks caused by cyber-criminals. Such attacks can be devastating from a privacy perspective, since they can be used for theft of identity, sensitive information or sensitive transactions. It is important to use counter measures against this threat by using computer security technologies like firewalls, anti-virus and Intrusion Detection Systems (IDS).

However, the operation of IDS systems imply that alarms with potentially sensitive information may be revealed to the security analysts monitoring the alarms. This may be particularly problematic if the IDS monitoring is outsourced to a third party. Contractual means like confidentiality agreements alone cannot hinder potential misuse of this information, for example by a corrupt security analyst performing the monitoring. Such misuse may be subtle and hard to detect. The analyst could for example use sensitive insider information leaked out via IDS alarms for his own gain when buying or selling shares in a monitored company, or he could sell such information to competitors.

It is therefore important to have a methodology that can be used to analyse the privacy impact of Intrusion Detection System (IDS) rules, in order to identify how privacy invasive the operation of signature-based IDS's are in a given scenario and context.

Our methodology is an engineering approach aimed at keeping the average privacy leakage caused by IDS rules below a certain threshold. The approach does not give any privacy guarantees, although the amount of privacy leakage can be chosen arbitrarily low[1]. It is not a replacement for provably secure methods for improving the privacy of IDS operations like for example cryptographic methods for privacy-preserving IDS. We believe the methodology can be useful in order to tune IDS rule sets to be less privacy invasive than what they typically are today. The usefulness comes both as reduced privacy leakage in the form of less exposure to sensitive information and as improved rule efficiency with less false alarms.

We apply the methodology on a case study of how privacy violating the rule set of the Snort IDS is. The rule set is categorised manually based on expert knowledge into five different categories.

This paper is organised as follows: Section 2 goes through some categorisation examples as an introduction to the problems the methodology is attempting to solve. Section 3 performs a theoretical analysis of privacy leakage from IDS rules. Section 4 describes the PRILE evaluation methodology and section 5 discusses the case study where the PRILE methodology was applied to the Snort rule set. Section 6 presents results from the case study, section 7 presents related work and section 8 contains concluding remarks.

2 Categorisation Examples

This section goes through some categorisation examples based on Snort IDS rules, as motivation for the methodology introduced in section 3 and 4. In subsection 2.1, we go through three clear categorisation examples - two attack rules that are not considered privacy violating and one user surveillance rule that is considered privacy violating. In the next subsection, we go through some less clear examples. The last subsection concludes the privacy against security discussion by recommending that a privacy leakage analysis of the IDS operation from a methodological perspective should be performed independently of security considerations as far as practically possible.

2.1 Clear Categorisation Examples

In some cases, it is relatively easy to determine that IDS rules are violating the user's privacy. IDS rules can often be presumed to contain bad or exceptional traffic if they describe malicious activities, like backdoors, viruses, worms, denial

[1] There are probably both technological and economical limits for how low the privacy leakage threshold can be set presuming today's IDS technology.

of service attacks, spoofing, shellcode or other attacks. It is further expected that attack rules without a significant privacy impact are reasonably precise, meaning that they most probably will detect the malicious activities without generating too many false alarms which may reveal privacy sensitive information about ordinary users.

The privacy impact of Snort rules vary greatly. Some rules are very specific and target a given attack. Especially when the rule targets a binary attack vector, for example an incoming worm or virus, then the utility from a security perspective can be expected to be high and the privacy impact from monitoring this event low because the revealed payload consists of binary code, which is more or less unintelligible and the rule is precise at matching an attack. One example of such a rule, is the rule with Snort ID (*sid:*) 2003 "MS-SQL Worm Propagation attempt". The modeled vulnerability (CVE-2002-0449) has a Common Vulnerability Scoring System (CVSS) score in the Common Vulnerabilities and Exposures (CVE) Database[2] of 7.5 out of 10, so this is considered a quite serious attack from a security perspective. This rule looks like the following:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (\
msg:"MS-SQL Worm propagation attempt";\
content:"|04|"; depth:1;\
content:"|81 F1 03 01 04 9B 81 F1 01|";\
content:"sock";\
content:"send";\
reference:bugtraq,5310;\
reference:bugtraq,5311;\
reference:cve,2002-0649;\
reference:nessus,11214;\
reference:url,vil.nai.com/vil/content/v_99992.htm;\
classtype:misc-attack;\
sid:2003;\
rev:8;)
```

The rule matches MS-SQL worm propagation attempts. These are UDP requests from any port on the external network towards the well known port number of Microsoft's SQL server on the home network. Snort usually presumes that alerts can only be caused by traffic to or from your own network, which is why it defines the variable $HOME_NET. The *msg:* field shows the IDS alert message that will show up in the IDS console when this rule is triggered. In this case, "MS-SQL Worm propagation attempt". The *content:* field matches specific strings or patterns in the payload. Four different content patterns are required to be present in an UDP packet to trigger this rule. Some of the matched content is binary data whereas other is ASCII text. The first *content:* field matches at depth (offset) 1 into the payload. The rule contains 5 authoritative references to other sources that describe the vulnerability, including Bugtraq and the Common Vulnerabilities and Exposures (CVE) databases of publicly known security

[2] Common Vulnerabilities and Exposures http://cve.mitre.org

vulnerabilities. *Classtype:* is Snort's rule classification. This rule is classified as a *misc-attack* rule, which indicates that the rule matches a known attack. *Sid:* is the unique Snort rule identity and *rev:* is the rule revision.

This is a specific rule, backed up by authoritative references. It is a quite serious exposure for vulnerable systems, as indicated by the CVSS score of 7.5. It is targeted against a system service that is not normally exposed to end-users on the Internet and should be reasonably precise at only matching attacks. The utility from a security perspective can in other words be expected to be high and the privacy impact low for this rule. We therefore categorised this as an attack rule, that is not regarded as sensitive from a privacy perspective.

A rule that can be expected to violate users' privacy, is *sid:1437* "MULTIMEDIA Windows Media download". This is a broad policy rule that matches download of any windows media files via the web. It does however not indicate which file that was downloaded[3].

```
alert tcp $EXTERNAL_NET 80 -> $HOME_NET any (\
msg:"MULTIMEDIA Windows Media download";\
flow:from_server,established;\
content:"Content-Type|3A|"; nocase;\
pcre:"/^Content-Type\x3a\s*(?=[av])(video\/x\-ms\-(w[vm]x|asf)|
      a(udio\/x\-ms\-w(m[av]|ax)|pplication\/x\-ms\-wm[zd]))/smi";\
classtype:policy-violation;\
sid:1437;\
rev:6;)
```

This rule matches TCP traffic originating from the external network and with destination to any port on the home network. It is in other word an HTTP reply message. The alert message given in the IDS console is "MULTIMEDIA Windows Media download" and it matches established TCP sessions originating from the server. The rule first performs a case insensitive string match on the HTTP header element "Content-Type:" in the payload. If the *content:* rule matches, then the regular expression as indicated in the *pcre:* field will be executed to match the Windows Media multimedia MIME types for *wvx,wmx,wma,wmv,wax,wmz* and *wmd* files. This rule is *not* backed up by any external references like CVE or Bugtraq, so the rule does not present any evidence of having any significant security impact. It is purely a rule for detecting violation of an IT usage policy, where downloading media files is not allowed. This is also indicated in Snort's *classtype:* field which classifies it as a "policy-violation", which broadly means a violation of corporate IT policy[4]. A user being monitored will probably regard such monitoring as a privacy violation, since the monitoring effectively limits what a user can see and do, and it affects both legal and illegal activities. This

[3] It is technically possible to record all network traffic over a limited time span using network forensic interfaces [1]. The monitoring organisation can therefore still detect downloaded media files, if they desire to do so.

[4] It should however be noted that the policy-violation rules not are enabled by default in Snort.

rule is therefore categorised as a privacy violating rule. Further information about interpreting Snort rules can be found in the Snort user's manual included in the source code distribution[5].

2.2 What About Conflicting Rules?

Some IDS rules are designed to monitor entire applications. For example SID 2372 is a rule that monitors all access to the file *showphoto.php*. The IDS rule detects two critical SQL injection vulnerabilities (CVSS score 10) that are remotely exploitable. SQL injection vulnerabilities can often be attributed to poor software engineering practices, so if one such vulnerability is detected, then it is reasonable that other similar vulnerabilities may exist as well in the vulnerable application. Another reason for adding general application monitoring, is that SQL injection attacks often have a large set of potential SQL-based attack vectors that can target the vulnerability, which means that it may seem easier and simpler for the author of the IDS rule to safeguard and write one rule that catches all application activities, than to cover all potential SQL injection attack vectors. Another complication, is the variety of encoding schemes used on the web, which can be used to evade attack detection. Examples of such encoding schema are URL encoding, HEX or Unicode[6].

This means that there is a significant chance that real attacks, hidden by a particular encoding scheme, may go undetected by a security analyst viewing the alert. It would be better in this case to use a more complex rule, that is able to trigger on the core of the problem, instead of general application activity monitoring.

However, there also exist some vulnerabilities where it can be harder to avoid general application monitoring. For example vulnerabilities that give direct or partial access to an unrestricted execution environment like the underlying operating system. These vulnerabilities are often due to lack of input data validation before external programs are called.

An example of such an input validation error is SID 1717, WEB-CGI simplestguest.cgi access. This is covered by the vulnerability CVE-2001-0022, which has a CVSS score of 10. The vulnerability allows remote attackers to execute arbitrary commands via shell meta characters in the guestbook parameter of the CGI script due to lack of parameter checking. It is not possible to know in advance which set of commands that may be attempted executed, so the safest thing to do, is to monitor all access to the vulnerable parameter of the CGI-script. However, it would be even better if the rule could simulate the input validation and let the most common normal use cases of the vulnerable parameter pass through without any alerts, to reduce the amount of false alarms and privacy leakage from using the rule.

[5] Snort is available from http://www.snort.org.

[6] See Ofer Maor and Amichai Schulman SQL Injection Signatures Evasion http://www.imperva.com/docs/SQLInjectionSignaturesEvasion.pdf.

2.3 Privacy against Security

IDS rules used in Managed Security Services (MSS) can in other words leak private and sensitive information. Customers will have particular concern about this for outsourced MSS. On the other hand, outsourcing MSS is usually cost effective and more efficient than running the service in-house from a security standpoint. Few companies can for example afford running their own 24x7 monitoring service. There is in other words a trade off between potential privacy leakage caused by a monitoring organisation running an MSS, and the privacy leakage caused by adversaries.

It should in this respect be noted that the effects of privacy leakage to criminals can be devastating and is without any regulatory control, whereas the privacy leakage from MSS are presumed to be measurable and under regulatory control. The MSS providers will however be liable if they breach the confidentiality agreement with the customer. It should therefore be a goal for the monitoring organisation to minimise the harm on privacy and confidentiality for the subjects being monitored, both as work ethics and because this reduces potential liabilities for the MSS provider.

The privacy invasiveness of IDS rules vary a lot. As the discussion above has shown, there are both specific and unspecific IDS rules with both high and low CVSS score. Analysing the *privacy impact* of IDS rules should in general be done *independently* of the security relevance of the IDS rule[7]. In conflicting situations where there is a privacy against security dilemma for an IDS rule, then the privacy leakage as a result of false alarms (false positives) can almost always be reduced significantly by investing some more effort into the design of the IDS rule. For example by making a more specific IDS rule that performs a more accurate test for the attack pattern and that also only matches vulnerable versions of applications instead of monitoring all versions of a given application or service. This can in many cases be done without significantly affecting the amount of missed real attacks (false negatives), as we have indicated in the examples in section 2.2. As an additional benefit, the MSS provider will probably reduce the costs of processing false IDS alarms.

3 Quantifying Privacy Leakage

We will in this section attempt to quantify the privacy leakage from IDS rules. An IDS rule signature R can be considered to consist of two parts as shown in Figure 1:

- *A protocol specific part P consisting of one or more patterns used to address a specific part of a session. The protocol specific part(s) trigger for every session for the chosen scope (platform, service, program or file level).*

[7] It can for example not be claimed that a high CVSS score in general warrants more privacy invasive monitoring, since this disregards privacy rights. More privacy invasive monitoring can only be warranted if this is the only practical solution for detecting the attack. If it is viable to detect the attack in a more privacy-friendly way, then this should be attempted.

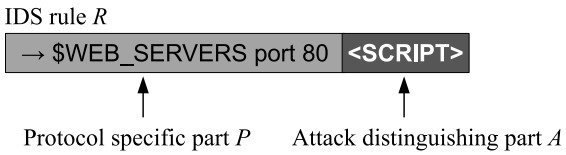

Fig. 1. Illustration of the protocol specific part and attack distinguishing part for an IDS rule (SID 1497, WEB-MISC cross site scripting attempt).

– *An attack distinguishing part A consisting of one or more patterns, which aims at matching an attack vector, for example given by a software vulnerability.*

The privacy leakage of an IDS rule describes how much potentially sensitive information that leaks out from applying the rule, that is not attack relevant. The privacy leakage is defined as:

Definition 1. *Let S be a sufficiently large set of communication sessions*[8]*, $S = \{s_i | i = 1, ..., n\}$, identified by the protocol specific part P of an IDS rule R. Let $E \subseteq S$ be a set of sessions that have been exposed by the IDS via alert messages that are false alarms. The privacy leakage p can then be calculated as the fraction $p = \frac{|E|}{|S|}$ of exposed communication sessions that are not attack related to all communication sessions.*

From this, it is apparent that the privacy leakage is proportional to the false positive rate for attack rules, however it is not the same measure. The false positive rate is the fraction of false alarms to the total number of alarms, whereas the privacy leakage instead is related to the total number of data sessions for a given scope.

It should be noted that this definition of privacy leakage implicitly presumes that real alarms (true positives) do not leak private or sensitive information. This may not necessarily be the case for rules identifying attack vectors where the user is lured into performing the attack by the adversary. Examples of such attacks are trojans and web bugs. It may be important from a security standpoint to investigate such attacks, regardless of whether it was the attacker or the user who initiated the attack. This means that real alarms (true positives) must be investigated, however additional privacy enhancing techniques like pseudonymisation or anonymisation of the alert data should be considered in these cases to protect the user's privacy.

We can now proceed with analysing the privacy leakage p for a group of commonly used IDS rules: match of an attack pattern in b different byte positions

[8] Data on session level is preferred to data on packet level since the data forensics involved in determining whether an alert is a false positive or a real attack often requires that the entire data session is loaded from a data forensics tool like a Time Machine http://www.net.t-labs.tu-berlin.de/research/tm/.

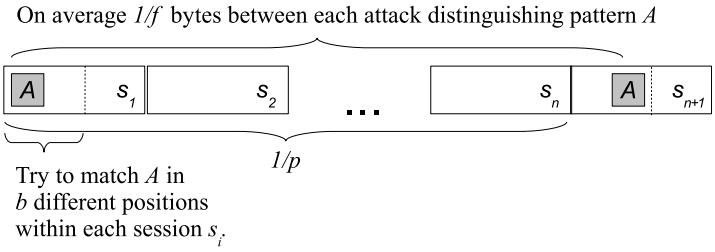

Fig. 2. Match of attack pattern in b different positions within each session s_i.

within each session s_i. It is in particular interesting to analyse the borderline case with a one byte wide attack distinguishing pattern A, since that occurs relatively frequently, and it is not obvious that a one byte wide pattern is sufficient to keep the privacy leakage below a chosen maximum privacy leakage threshold.

It is presumed that the occurrence frequency f of the attack distinguishing pattern A for a byte stream of normal traffic is known or can be measured. The privacy leakage caused by matching the attack pattern A in b different byte positions can then be calculated using the relation $f \leq \frac{p}{b}$, as indicated in Figure 2, which gives:

$$bf \leq p\,. \tag{1}$$

This means that the number of byte positions matched times the occurrence frequency must be less than or equal to the privacy leakage.

Example 1. The IDS rule (SID 2666) for a format string vulnerability in the Courier-IMAP server uses the following regular expression:
/^PASS\s+[^\n]?%/smi*. In this regular expression, it is only one character, the '%' sign, that differentiates this IDS rule from a normal IMAP password authentication session. Furthermore, the regular expression matches the percent sign in any position of the password. It is in other words only the *percent sign* that is the *attack distinguishing part* of the IDS rule. The maximum privacy leakage threshold is chosen to be $p = 1\%$.

Presuming that the probability of hitting a percent sign in a random password has been measured[9] to $f = 0.0017$, and that the average password size is $b = 8$ bytes, this means that $bf = 1.35\%$ by using Equation 1. Since this is larger than 1%, this means that the IDS rule is considered privacy violating.

This example shows that IDS rules with a one byte wide attack distinguishing pattern can be privacy violating. In general, the occurrence frequency of the attack pattern f is not available. It is therefore only possible to get good estimates of the privacy leakage for some special cases like this. It is however

[9] Matt Weir Reusable Security - Character Frequency Analysis Info http://reusablesec.blogspot.com/2009/05/character-frequency-analysis-info.html.

in many cases still possible to estimate the privacy leakage for extreme cases where one can argue that the attack distinguishing pattern either occurs sufficiently frequently to cause a a privacy leakage or sufficiently infrequently to not cause a privacy leakage based on qualitative arguments. For example can rules that detect attacks based on protocol violations in many cases be expected to have little or no privacy leakage, presuming that they seldom or never happen in ordinary traffic. Also, many overflow detecting rules match so wide patterns for typed user input that they probably not will be privacy leaking for normal traffic.

4 Evaluation Methodology

The aim of our evaluation methodology is to provide a gold standard for evaluating the privacy impact of Network-based IDS rules. We have defined a 5 level scale for privacy invasiveness that focuses on how wide scope the privacy violation has. This scope is also important for the privacy leakage calculation, since it defines the split between the protocol specific part P and the attack distinguishing part A for a given IDS rule. The privacy leakage scale is defined below:

0-None No privacy leakage expected from the IDS rule. This can for example happen for rules detecting protocol violations or denial of service attacks that can not happen from normal user behaviour.

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 sessions being investigated by it. Another way of interpreting this level is as a *tolerable* average privacy leakage.

2-Program file More than p percent of all sessions targeted at a given *program file or module* 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, which potentially can cause monitoring of any application on that given platform. The scope of all sessions S must here be limited to the number of relevant sessions. If specific files or file types are being monitored on platform level, then S must only consist of sessions that contain the monitored files or file types.

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. The legality of Level 5 rules in a given legislative area must be investigated before such rules are enabled. For example, monitoring use of end-user services like chat, instant messaging, VoIP, email or web.

The enumerated scale from 0 to 5 can then be used for quantitative measurements of privacy invasiveness. The scale is bounded and naturally lends itself to further aggregation over a group of rules. 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.

5 Discussion

Manual categorisation of 3669 Snort rules from the community rule set was done according to our PRILE methodology. We presume a network environment where all typical end-user services (HTTP,FTP,POP,IMAP,...) are provided on the Internet in their normal configuration. We furthermore presume English localisation (language) of the environment where the IDS rules are applied. For file storage services, like FTP, we presume that some end-users can have access to both upload and download of data.

The main problem we identified during categorisation, is IDS rules that perform full monitoring of all access to an application or program file as part of an application. This is quite common for web application monitoring rules. The problem with this practice is the privacy leakage that full access monitoring causes.

Our opinion is that IDS systems and rules should be improved to reduce the scope of monitoring in order to only detect attack traffic and preferably only trigger on vulnerable applications. General application monitoring can be so noisy that the utility of it also from a security standpoint can be disputed. However, rules should not be made so specific that they reduce the risks of identifying likely variants of a given attack. So there will in practice often be a trade off between privacy leakage reduction and IDS rule generalisation. However, section 2.2 shows that the current practice often goes too far in the direction of monitoring all access to a given service or application.

Another type of vulnerabilities that often have weak attack distinguishing patterns are format string vulnerabilities. Most of these have only got an attack distinguishing pattern of one byte. For example, SID 2666 targets a format string vulnerability for the password handling of Courier-IMAP. This is the rule that was analysed in Section 3.

6 Results

487 of the 3669 manually categorised Snort rules (13%) appear to have a significant impact on privacy as shown in Table 1. However, in a default Snort installation 15 rule files with 270 rules are disabled. All the level 5 policy specific rules (117 rules) were contained within the set of disabled rule files, which is encouraging. This shows that Snort in its default configuration aims at not being excessively privacy invasive. The policy specific rules detect content like chat, pornography, peer-to-peer or multimedia. The complete rule set has 370 privacy violating rules on level 2 to 4. These are general file, application or

Table 1. PRIvacy LEakage (PRILE) classification of Snort rule sets.

PRILE	Privacy invasiveness	Default rule set	Disabled rules	All rules
0	None	13	4	17
1	Vulnerability	3026	139	3165
Total non-privacy violating:		3039	143	3182
2	Program file	333	5	338
3	Application	24	3	27
4	Platform	3	2	5
5	Policy		117	117
Total privacy violating:		360	127	487
Percentage privacy violating:		11%	47%	13%

platform monitoring rules, which are founded on a known vulnerability. Even if these rules account for only 10% of all IDS rules, they can be expected to cause significant privacy leakage, false alarms and draw processing power from the IDS sensor. This means that there is a large improvement potential from a privacy and efficiency perspective if these rules are tightened up to fall within PRILE 1. In fact, 76% of all rules categorised as privacy violating fall into this category.

7 Related Work

There exists, as far as we know, no similar scoring system that can be used to analyse the privacy leakage of IDS rules. There are however similar scoring systems for system vulnerabilities. The Common Vulnerability Scoring System (CVSS) is an industry standard metric for the characteristics and impacts of IT vulnerabilities [2]. 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, it does not cover the potential confidentiality loss that can occur from IDS monitoring activities.

There is also some relevant work within the area of privacy metrics. Privacy violations of internet sites are described in [3], however this paper is quite general and does not mention any indicators that capture the amount of privacy violations. Other metrics for privacy are entropy-based [4] or based on the combination of k-anonymity [5, 6, 7] and l-diversity [8]. These measures focus more on how anonymous data are than to measure to what extent a network monitoring organisation's operation is privacy-intrusive.

Another related area is privacy enhanced intrusion detection systems. The BRO IDS for example supports a way to anonymise the payload of a packet instead of removing the entire payload [9]. There also exists some earlier work on privacy-enhanced host-based IDS systems that pseudonymise audit data and performs analysis on the pseudonymised audit records [10, 11, 12, 13, 14, 15].

8 Conclusion

This paper introduces a new methodology - PRILE for identifying privacy leakage in IDS rules. The methodology itself is intended to be generic and should also be useful for privacy leakage evaluation of other network intrusion detection systems than Snort. A limitation with the methodology is that it does not specify how to define the scope for preprocessors and similar IDS rules that present aggregated data[10]. In these cases, the false alarm rate can be used as an alternative indicator of privacy leaking rules, since it is proportional to our privacy leakage metric for a given IDS rule.

We have performed a proof-of-concept evaluation of the Snort rule set using the PRILE methodology. This evaluation confirms that Snort in its default configuration aims at not being excessively privacy invasive. Level 5 policy rules are for example switched off by default. Problematic areas we have identified are rules with poor or missing ability to distinguish attack traffic from normal traffic. For example general file, application or platform monitoring rules, which are founded on known vulnerabilities. Even if these rules account for only 10% of all IDS rules, they can be expected to cause privacy leakage, false alarms and draw a significant amount of processing power from the IDS sensor. This means that there is a large improvement potential from a privacy, cost and efficiency perspective if these rules are tightened up to fall within PRILE level 1.

In addition, optimisations of the IDS rule set can and should be considered both in the temporal domain based on "smart" IDS rules that disable themselves when a system is patched up and also based on the environment - whether rules are relevant for the platforms and appliances in the network being monitored.

Future research includes adding support for measuring the privacy leakage and occurrence frequency of attack distinguished patterns on an existing IDS system, in order to get better privacy leakage estimates and improve the model. Another possibility is to do a broader study where a representative set of experts perform the same classification to achieve a more objective interpretation of the PRILE methodology. Experiences by applying the methodology can then be used to further improve it. Last, but not least - this methodology may open up a possibility for privacy impact testing tools for IDS systems.

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

[10] False alarms from preprocessors or composed IDS systems may also contain aggregated data with sensitive information. For example false alarms from the Snort portscan preprocessor which may reveal information about user behaviour like web browsing habits.

Bibliography

[1] Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., Shneider, F.: Enriching network security analysis with time travel. SIGCOMM Comput. Commun. Rev. **38**(4) (2008) 183–194

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

[3] Klewitz-Hommelsen, S.: Indicators for privacy violation of internet sites. Electronic Government (2002) 219–223

[4] Sebastian Clauß, S.S.: Structuring anonymity metrics. Proceedings of the second ACM workshop on Digital identity management (2006) 55–62

[5] Ti, P.S.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13** (2001) 1010–1027

[6] Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10** (2002) 557–570

[7] Ciriani, V., di Vimercati, S.C., Foresti, S., Samarati, P.: k-Anonymity. In: Secure Data Management in Decentralized Systems. Springer (2007) 323–353

[8] Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. Cornell University (March 2007) 52

[9] Pang, R., Paxson, V.: 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) 339–351

[10] Sobirey, M., Fischer-Hübner, S., Rannenberg, K.: Pseudonymous audit for privacy enhanced intrusion detection. In: Proceedings of the IFIP TC11 13th International Conference on Information Security (SEC'97). (May 1997) 151–163

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

[12] Sobirey, M., Richter, B., König, H.: 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) 278–290

[13] R. Büschkes, D. Kesdogan: Privacy enhanced intrusion detection. In Müller, G., Rannenberg, K., eds.: Multilateral Security in Communications, Information Security, Addison Wesley (1999) 187–204

[14] Flegel, U.: Privacy-Respecting Intrusion Detection. 1 edn. Springer (October 2007)

[15] Holz, T.: An efficient distributed intrusion detection scheme. In: COMPSAC Workshops. (2004) 39–40