A contextual privacy-aware access control model for network monitoring workflows: work in progress

Eugenia I. Papagiannakopoulou, Maria N. Koukovini, Georgios V. Lioudakis, Joaquin Garcia Alfaro, Dimitra I. Kaklamani, Iakovos S. Venieris

A Contextual Privacy-Aware Access Control Model for Network Monitoring Workflows: Work in Progress

Eugenia I. Papagiannakopoulou[1], Maria N. Koukovini[1], Georgios V. Lioudakis[1], Joaquin Garcia-Alfaro[2], Dimitra I. Kaklamani[1], and Iakovos S. Venieris[1]

[1] School of Electrical and Computer Engineering
National Technical University of Athens, Athens, Greece
[2] Institut TELECOM, TELECOM Bretagne,
CS 17607, 35576 Cesson-Sévigné, Rennes, France

Abstract. Network monitoring activities are surrounded by serious privacy implications. The inherent *leakage-proneness* is harshened due to the increasing complexity of the monitoring procedures and infrastructures, that may include multiple traffic observation points, distributed mitigation mechanisms and even inter-operator cooperation. In this paper, we report a work in progress policy model that aims at addressing these concernes, by verifying access requests from network monitoring workflows, with privacy features already contained since their specification phase. We survey related work, outline some of their limitations, and describe an early version of our proposal.

Keywords: Network monitoring, access control, privacy, context, workflows

1 Introduction

Network monitoring is characterised by certain features that stress the need for special mechanisms controlling access to the data that are collected and being processed, as well as the underlying computational resources. To name a few, first of all the protection of privacy is a fundamental issue, since the concerns are not limited to the payload of the packets; sensitive information can be derived by protocol headers, even by not obvious fields [7] and even if there has been prior anonymisation of the data [8][26]. Second, the domain of network monitoring has become a legislated area, with several regulations governing the collection and consequent processing of the associated data (e.g., [11][12][13]); the regulations should be taken into account when designing access control systems and specifying the policies [19][29]. Third, something that cannot be neglected is the fact that network monitoring deals with very high data rates, exceeding the order of Gbps; in this context, access control has to face the stringent requirement of ultra fast responsiveness. Fourth, there is an emerging trend of *collaborative monitoring*, reflecting the cooperation between different stakeholders, in order to

effectively cope with current attacks threatening networks, such as botnets and large scale DDoS. Last but not least, access control within network monitoring can be interpreted in a variety of ways, notably access to monitored data, monitoring devices and processing operations, as well as access policies that reflect operational aspects, mostly related to security, such as the behaviour of a firewall or the routing table of a router redirecting malicious traffic to a honeypot.

In this paper, we sketch the definition of a new access control model that aims at dealing with all those aforementioned aspects. It is conceived on the basis of network monitoring, capturing all the underlying concepts, e.g., devices and functions, and providing rich expressiveness. Moreover, in order to deal with performance needs and in line with the "privacy by design" principle, the proposed approach puts in place mechanisms for inherent privacy-awareness of network monitoring *workflows*, by incorporating associated access control provisions already at design-time, thus minimising run-time reasoning overheads. In that respect, a procedure is being developed for the verification of workflows and their enhancement with privacy-preserving features; a challenge here is to capture at design-time the contextual constraints, typically dealt with at run-time.

In the following, we first survey related work and outline the reference framework in Sections 2 and 3. Section 4 describes the access control model, while the paper concludes in Section 5 with some insights on current and future work.

2 Related Work

Privacy protection in network monitoring is typically thought of as the anonymisation of traffic traces, an area where several works have been proposed [14][15] [17][18][24]. Nevertheless, albeit useful as anonymisation libraries, such approaches base on "static" anonymisation patterns, while being vulnerable to attacks able to infer sensitive information [8][26].

Privacy-aware access control has recently evolved to a prominent research area [4]. However, approaches such as [1][5][9][21][22][25] have not been designed for meeting the particular requirements of network monitoring and conceptualising the corresponding functionalities and infrastructures; additionally, they either do not support context-awareness or they only support some straightforward contexts. Furthermore, they are not suitable for highly dynamic and distributed environments and –especially– for automating privacy-awareness. On the other hand, work in the area of access control enforcement in workflow management [6][28] and Model-Driven Security [3][23], though important, suffer from enforcing security policies only at run-time and not during the workflow formation.

Finally, the proposed model draws inspiration from previous works of the authors, notably OrBAC [10][27][2] and PRISM [16][20][19]. OrBAC provides a mature framework for managing contextual policies, and several extensions, e.g., for dynamically deploying security policies; PRISM is an access control approach specifically devised for network monitoring, although limited to single-probe environments. Their enhancement towards fulfilling all the requirements implied here, has been the motivation for the model presented in the following.

3 Reference Framework for Network Monitoring

The network monitoring framework under consideration relies on a modular and service-oriented architecture; it is centred around the concept of the *workflow*, that is, a series of *tasks*, along with their interaction patterns (both data- and control-flow), that are executed in order for a high-level purpose to be fulfilled.

As shown in Fig. 1, a workflow's lifecycle can be seen as consisting of two phases, notably *Planning* and *Execution*. The former refers to the specification of the workflow by its designer, including all steps for its graphical definition, decomposition to elementary tasks, compliance checking and necessary transformations. On the other hand, the Execution Phase relies on the Planning Phase's outcome and refers to the deployment of the workflow to the system and its execution. The execution environment consists of *Agents* providing the service abstractions of the underlying actual components (e.g., detection or mitigation ones); the *Inter-domain Exchange Point* (IXP) constitutes a special Agent, being the functional gateway towards external domains, for the realisation of cooperative monitoring. The execution is coordinated by dedicated *Orchestrators*, while the means for context and capabilities management are also provided.

Of great importance is the procedure for verifying and appropriately adjusting the workflow at design-time, so that it becomes inherently privacy-compliant before its execution. The procedure is conducted by the *Model Checker*, whereas a *Reasoner* provides the necessary intelligence, being the entity that incorporates and infers knowledge from the Policy Model; it consists in three steps:

- *Purpose verification:* Checks regarding purpose compliance are performed; specifically, in order for a workflow to be privacy-aware, its design must be relevant and consistent with a purpose, while the purpose itself should not contradict with the access rights of the person initiating the workflow.
- *Skin task verification*[3]: Each skin task is evaluated both individually and in relation to the rest of the skin tasks. During these checks, the system may introduce modifications, such as task additions, removals or substitutions.
- *Decomposition*: Each composite skin task is decomposed in more elementary subtasks, until it is finally analysed in a combination of atomic tasks that will eventually be included in the final executable form of the workflow.

Fig. 2 illustrates a network monitoring workflow example; Fig. 2(a) depicts the initial workflow, as specified by its designer, and Fig. 2(b) the workflow after some transformations following the verification and transformation procedure.

In practice and especially within workflows, rules remain inactive until a set of conditions are fulfilled. We denote as *contextual* the authorisation policies containing dynamic authorisation provisions. In this regard, authorisation rules may depend on temporal contexts (e.g., authorisations granted only during working

[3] We call *skin tasks* the ones defined by the workflow designer, as opposed to the tasks that their inclusion in the workflow is a result of workflow check and modification; all tasks in Fig. 2(a) are considered to be skin tasks. Their separate examination without considering their decomposition targets the early identification of conflicts.

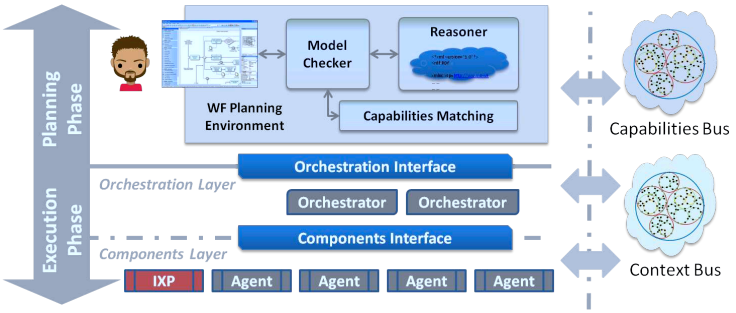
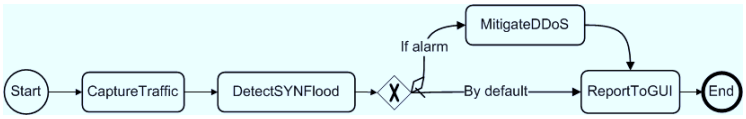
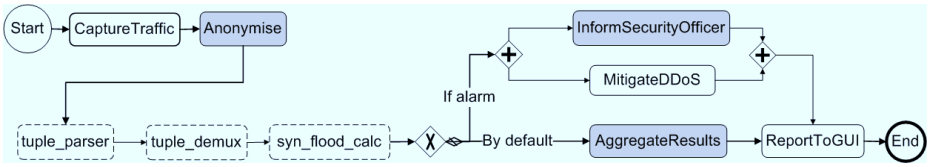


Fig. 1: Overall Architecture



(a) Initial Workflow (as specified by its designer)



(b) (Partially) Processed Workflow

Fig. 2: Workflow Example

hours), geographical contexts (e.g., permission inside the physical boundaries of a company), *a priori* contexts (in which a permission to execute a set of actions can only be carried out as a result of the completion of previous actions). Therefore, it is essential that not only the contextual conditions are captured by the model, but also that they are taken into consideration during the verification and transformation procedure, providing for the specification of context-based differentiated workflow behaviours, already during workflow formation.

4 Policy Model

The *Policy Model* regulates the system's operation and drives the workflow verification and transformation process. It consists of a semantically rich information model, providing abstractions of the underlying concepts, and access control rules. Similar to the *subject–verb–object* linguistic pattern, everything that takes place in the context of the system's operation can be seen as an *operation* of an *actor* on a *resource*. This metaphor is the basis on which *actions* and *tasks* are defined, being the core elements of access control rules and workflows and the "seed" for knowledge extraction. The following outline the basic concepts.

4.1 Information Model

In a typical case and at a concrete level, a set of *Users* (U), participating (e.g., working) in *Organisations* (Org), are –directly or indirectly– using *Operation Containers* (OpC), deployed over *Machines* (M) and offering *Operation Instances* (OpI)[4], in order to act upon *Objects* (Obj), with the latter referring to everything that is affected by or required for the execution of an action, such as *Data* (D) being collected and/or processed.

At an abstract level, the users are assigned with *Roles* (R), their actions instantiate some *Operations* (Op) and are performed for fulfilling some *Purposes* (Pu). Moreover, data, organisations, machines and operation containers are characterised by *types*, reflecting the semantic class they fall under; thus, sets of *Data Types* (DT), *Organisation Types* ($OrgT$), *Machine Types* (MT) and *Operation Container Types* ($OpCT$) are defined, respectively.

Additional elements of the model include *Context* (Con), that enables the definition of contextual parameters, *Attributes* (Att), that are leveraged for describing properties and characteristics of other elements, as well as *Alerts* (Al), i.e., notices regarding events, along with the corresponding *AlertTypes* (AlT).

While most of these notions are either typically present in state-of-the-art models or intuitively self-explained, a few remarks are deemed necessary for some of the concepts. Specifically, the OpC and $OpCT$ are introduced in order to model components or other functional structures that typically offer a set of operations together. For instance, an `IntrusionDetectionSystem` clusters several operations related with intrusion detection. Apart from the convenience it introduces regarding several modelling aspects (such as the inheritance of attributes), these structures are also helpful for describing a variety of concepts related with "horizontal" dependencies and transfer of characteristics.

Moreover, the machines (and their types) play a fundamental role in network monitoring and, therefore, our model cannot be limited to a level of abstraction exclusively centred around functionalities; in any case, functionalities are provided by machines which, on the one hand, are characterised by attributes (e.g., topological ones) that may be inherited to the hosted functionalities and, on the other hand, create inherent dependencies between the hosted functionalities.

All these concepts comprise graphs of elements that are characterised by relations; the latter are implemented by predicates defining AND- and OR-hierarchies and enabling the inheritance of attributes and rules, as well as the specification of dependencies. For instance and with respect to the DT graph, three partial order relations are defined: $isA(dt_i, dt_j)$, $lessDetailedThan(dt_i, dt_j)$ and $contains(dt_i, \langle dt \rangle^k)$, where $dt_i, dt_j \in DT$, and $\langle dt \rangle^k \subseteq \mathcal{P}(DT)$, reflecting, respectively the particularisation of a concept, the detail level and the inclusion of some data types to another. Moreover, the model specifies the necessary predicates in order to link concepts from different graphs; for example, the predicate $mayActForPurposes(r, \langle pu \rangle^k)$, where $r \in R$, $\langle pu \rangle^k \subseteq \mathcal{P}(Pu)$, appoints the legitimate purposes $\langle pu \rangle^k$ for which the users assigned with the role r may act.

[4] In Web Services' terms, Operation Containers correspond to a service `portType`, whereas Operation Instances represent the associated `operations`.

4.2 Actions, Tasks and Workflows

We use the term *Action* in order to refer to a structure similar to the *subject–verb–object* metaphor, and describe situations where an *operation* op_i is performed by an *actor* a_i on a *resource* res_i, i.e., $act_i = \langle a_i, op_i, res_i \rangle$. Following the hierarchical relations of operations *Op*, an action can be either *atomic* or *composite*, depending on whether the associated operation can be decomposed to more elementary operations or not. In addition, the definition of an action can be complemented by a fourth parameter, notably the *organisation* within which it is defined; in such a case, it is expressed as $act_i = \langle a_i, op_i, res_i, org \rangle$.

Several of the aforedesribed types of entities may constitute actors and resources; they can be either concrete, e.g., *Users* and *Data*, or abstract, e.g., *Roles* and *Data Types*. Depending on whether actors and resources are defined at abstract, concrete or mixed level, several variations of Actions are identified, such as *Abstract Actions*, *Concrete Actions* and *Semi-Abstract Actions*, the formal description of which is beyond the scope of this work-in-progress overview.

Actions are used for describing *Tasks* and *Workflows*, the definition of which is interrelated. A task t_i is an action act_i when being part of a workflow w, written as a tuple $t_i = \langle a_i, op_i, res_i \rangle_w$, or $t_i = \langle a_i, op_i, res_i, org \rangle_w$. On the other hand, a workflow consists in a finite number of tasks, i.e., $w = \langle t_1, t_2, ..., t_n \rangle$, along with the control- and data-flow relationships among them.

4.3 Access Control Rules

Access control rules are used for defining *permissions*, *prohibitions* and *obligations* over *Actions*, that is, they specify authorisations between actors, operations and resources within organisations. The following predicates are used:

- *Permission*(*pu*, *act*, *preAct*, *cont*, *postAct*)
- *Prohibition*(*pu*, *act*, *preAct*, *cont*, *postAct*)
- *Obligation*(*pu*, *act*, *preAct*, *cont*, *postAct*)

In these expressions, apart from the action *act* that the rule applies to, additional provisions are defined. These include contextual conditions $cont \in \mathcal{P}(Con)$, the purpose $pu \in Pu$ under which the rule is applicable, as well as structures of actions, *preAct* and *postAct*, that should respectively precede and follow the rule's enforcement. It should be noted here that *preAct* and *postAct* may comprise complex logical structures of actions, including negation. This enables the specification of expressive *Dynamic Separation of Duty* constraints, whereby conflicts between tasks can be defined based on any of the elements.

Based on these formalisms, the model provides the system with the necessary knowledge regarding access rights and their applicability, hierarchical relations and inheritance of attributes and access primitives across the information graphs, as well as associations between the model's different components. As an additional remark here, in some cases rules can be a priori evaluated; this puts in place a separation of the real-time and non-real-time aspects of access control procedures, resulting in performance advances. A few examples of knowledge extraction are provided in the next section.

4.4 Knowledge Extraction

Let's assume the rather typical case where a user holding a role $r_{init} \in R$ initiates a workflow $w = \langle t_1, t_2, ..., t_n \rangle$, where $t_i = \langle a_i, op_i, res_i, org \rangle_w$, declaring a purpose $pu_w \in Pu$. Sample knowledge that will be requested in the context of workflow verification and transformation and will consequently be inferred after reasoning over the model includes:

—Whether r_{init} justifies triggering the execution of w, in order for pu to be served. For instance, a `NetworkAdministrator` should be able to execute a workflow for the purpose of `NetworkSecurity`, while an `Accountant` should not.

—Whether the operations op_i contained in w's tasks are in line with pu_w. For example, all functions in Fig. 2(a) are relevant to the purpose of `NetworkSecurity`, while a task `InterceptCommunications` is not and would have been rejected.

—Whether a task $\langle a_i, op_i, res_i \rangle_w$ is in principle valid, i.e., the actor a_i has the right to perform operation op_i on resource res_i, regardless other constraints.

—The tasks that should complement the execution of a task, i.e., precede, follow or be executed in parallel. For instance, there may be the case that whenever a `DDoSAttack` is identified and reported by an alarm, the prompt notification of the `SecurityOfficer` should take place, along with the associated mitigation actions (represented by the high-level task `MitigateDDoS`); as Fig. 2(b) depicts, the `InformSecurityOfficer` task has been added to be executed in parallel, while the `AggregateResults` task is added for being executed before `ReportToGUI`.

—The possible $\langle a_i, res_i \rangle$ combinations allowed for the execution of an operation op_i, given w, r_{init} and pu_w. For instance, depending on the actor a_i in charge of executing the `ReportToGUI` task, the resource res_i to be delivered to the task may be plain or aggregated detection data. Such provisions may result in the incorporation of conditional branches within the workflow, such as different execution paths for different actors.

—The possible decompositions of a task t_i, for given r_{init} and pu_w. For instance, Fig. 2(b) illustrates a simplified decomposition of the `DetectSYNFlood` task to the subtasks `tuple_parser`, `tuple_demux` and `syn_flood_calc`; nevertheless, there are alternative decompositions that could be leveraged, so it is assumed here that the specific decomposition has been selected based on the parameters applying (e.g., r_{init} may permit only this decomposition).

—Whether a task within the workflow requires another task (or a series of tasks), along with its exact or relative position in the workflow; such requirements might also depend on context. For instance, the `Anonymise` task has been added in Fig. 2(b), assuming to be a prerequisite for `tuple_parser`'s execution under the control of the actor in charge and for given r_{init}, pu_w, etc.

—Identification of possible incompatibilities and conflicts that may exist among some tasks within the workflow and, possibly, their resolution. For instance, the addition of the `Anonymise` task in Fig. 2(b) could be the resolution of an incompatibility between the `CaptureTraffic` and `ParseTuple` tasks, taking into account the actors and resources of the tasks, as well as the r_{init}, pu_w, etc.

—Identification of the possible workflow differentiations, based on contextual parameters and alarms/events. For example, Fig. 2 illustrates a conditional branch,

that depends on whether an alarm is raised or not; while here it is supposed that the workflow designer has defined the conditional branch, there can be cases where such differentiations are inferred by the Policy Model.

—What are the possible workflow instantiations, taking into account the access rights, as well as the available capabilities. As an example, let's assume that Ingrid, the engineer on duty at the time of a `DDoSAttack` alert, holds the `JuniorNetworkAdministrator` role and, therefore, in order to implement the `InformSecurityOfficer` task, she is authorised to only use the `MakeVoIPCall` operation offered by `VoIPSoftwareClient` operation containers. In addition, Ingrid is authorised to use only a small number of `PersonalComputer` machines with `VoIPSoftwareClient` software deployed. Thus, for the instantiation of the workflow during Ingrid's duty hours, the model should enable the identification of a `VoIPSoftwareClient`-enabled `PersonalComputer` machine that Ingrid is authorised to use, in order to include the concrete task in the workflow.

5 Conclusions and Current Work

Motivated by the necessity of enhancing network monitoring architectures in terms of privacy-awareness, we are working towards the specification of a new policy model for controlling access to associated resources, such as data, operations and infrastructures. Our model aims at allowing the definition of network monitoring workflows, with privacy features already contained since their specification phase. In this paper, we have surveyed existing solutions, outlined some of their drawbacks, and presented an early version of our proposal. Our model takes full advantage of the integration of contextual properties; this allows us to cover the definition of both simple and complex business processes, as well as describing rich contextual categorisation of network resources. As a result, our model allows to potentially reduce the definition of the concrete policies which will need to be deployed in the end, over legacy network monitoring systems. As future perspectives, we aim at implementing and empirically verifying the powerfulness of our model, by the application of a proof-of-concept version of our approach carried out through a representative real-world case study.

6 Acknowledgements

This research was partially supported by the European Commission, in the framework of the FP7 DEMONS project (Grant agreement no. FP7-257315).

The research of M. N. Koukovini is co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: *Heracleitus II. Investing in knowledge society through the European Social Fund.*

References

1. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases. In: VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases. pp. 143–154. VLDB Endowment (2002)
2. Ajam, N., Cuppens-Boulahia, N., Cuppens, F.: Contextual privacy management in extended role based access control model. In: Garcia-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N., Roudier, Y. (eds.) Data Privacy Management and Autonomous Spontaneous Security, Lecture Notes in Computer Science, vol. 5939, pp. 121–135. Springer (2010)
3. Alam, M., Hafner, M., Breu, R.: Constraint based role based access control in the sectet-framework a model-driven approach. Journal of Computer Security 16(2), 223–260 (2008)
4. Antonakopoulou, A., Lioudakis, G.V., Gogoulos, F., Kaklamani, D.I., Venieris, I.S.: Leveraging access control for privacy protection: A survey. In: Yee, G. (ed.) Privacy Protection Measures and Technologies in Business Organizations: Aspects and Standards. IGI Global (2011)
5. Ardagna, C.A., Camenisch, J., Kohlweiss, M., Leenes, R., Neven, G., Priem, B., Samarati, P., Sommer, D., Verdicchio, M.: Exploiting cryptography for privacy-enhanced access control: A result of the prime project. Journal of Computer Security 18(1), 123–160 (2010)
6. Ayed, S., Cuppens-Boulahia, N., Cuppens, F.: Deploying security policy in intra and inter workflow management systems. Reliability and Security, International Conference on Availability pp. 58–65 (2009)
7. Bellovin, S.M.: A technique for counting NATted hosts. In: IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. pp. 267–272. ACM, New York, NY, USA (2002)
8. Burkhart, M., Schatzmann, D., Trammell, B., Boschi, E., Plattner, B.: The role of network trace anonymization under attack. SIGCOMM Computer Communications Review 40(1), 5–11 (2010)
9. Byun, J.W., Li, N.: Purpose based access control for privacy protection in relational database systems. The VLDB Journal 17(4), 603–619 (2008)
10. Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual Security Policies. International Journal of Information Security 7(4), 285–305 (2008)
11. European Parliament and Council: Directive 95/46/EC of the European Parliament and of the Council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Communities L 281, 31–50 (November 1995)
12. European Parliament and Council: Directive 2002/58/EC of the European Parliament and of the Council concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). Official Journal of the European Communities L 201, 37–47 (July 2002)
13. European Parliament and Council: Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. Official Journal of the European Communities L 105, 54–63 (April 2006)

14. Fan, J., Xu, J., Ammar, M.H., Moon, S.B.: Prefix-preserving IP address anonymization. Computer Networks 46(2), 253–272 (2004)
15. Foukarakis, M., Antoniades, D., Antonatos, S., Markatos, E.: Flexible and high-performance anonymization of NetFlow records using anontool. In: SECURECOMM Conference (2007)
16. Gogoulos, F., Antonakopoulou, A., Lioudakis, G.V., Mousas, A.S., Kaklamani, D.I., Venieris, I.S.: Privacy-aware access control and authorization in passive network monitoring infrastructures. In: CIT 2010: Proceedings of the 10th IEEE International Conference on Computer and Information Technology (2010)
17. Koukis, D., Antonatos, S., Antoniades, D., Markatos, E., Trimintzios, P.: A generic anonymization framework for network traffic. In: Communications, 2006. ICC '06. IEEE International Conference on. vol. 5, pp. 2302 –2309 (June 2006)
18. Li, Y., Slagell, A., Luo, K., Yurcik, W.: Canine: A combined conversion and anonymization tool for processing netflows for security. In: International Conference on Telecommunication Systems Modeling and Analysis (2005)
19. Lioudakis, G.V., Gaudino, F., Boschi, E., Bianchi, G., Kaklamani, D.I., Venieris, I.S.: Legislation-aware privacy protection in passive network monitoring. In: Portela, I.M., Cruz-Cunha, M.M. (eds.) Information Communication Technology Law, Protection and Access Rights: Global Approaches and Issues, chap. 22, pp. 363–383. IGI Global (2010)
20. Lioudakis, G.V., Gogoulos, F., Antonakopoulou, A., Mousas, A.S., Venieris, I.S., Kaklamani, D.I.: An access control approach for privacy-preserving passive network monitoring. In: ICITST 2009: Proceedings of the 4th International Conference for Internet Technology and Secured Transactions (November 2009)
21. Masoumzadeh, A., Joshi, J.: Purbac: Purpose-aware role-based access control. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems: OTM 2008, Lecture Notes in Computer Science, vol. 5332, pp. 1104–1121. Springer (2008)
22. Massacci, F., Mylopoulos, J., Zannone, N.: Hierarchical hippocratic databases with minimal disclosure for virtual organizations. The VLDB Journal 15, 370–387 (November 2006)
23. Menzel, M., Meinel, C.: SecureSOA. IEEE International Conference on Services Computing pp. 146–153 (2010)
24. Minshall, G.: Tcpdpriv. http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html
25. Ni, Q., Bertino, E., Lobo, J., Brodie, C., Karat, C.M., Karat, J., Trombetta, A.: Privacy-aware role-based access control. ACM Transactions on Information and System Security 13(3), 1–31 (2010)
26. Pang, R., Allman, M., Paxson, V., Lee, J.: The devil and packet trace anonymization. Computer Communication Review (CCR) 36(1), 29–38 (2006)
27. Preda, S., Cuppens, F., Cuppens-Boulahia, N., Garcia-Alfaro, J., Toutain, L.: Dynamic deployment of context-aware access control policies for constrained security devices. J. Syst. Softw. 84, 1144–1159 (July 2011)
28. Russello, G., Dong, C., Dulay, N.: A workflow-based access control framework for e-health applications. In: WAINA 2008: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops. pp. 111–120. IEEE Computer Society, Washington, DC, USA (2008), http://portal.acm.org/citation.cfm?id=1395080.1395523
29. Sicker, D.C., Ohm, P., Grunwald, D.: Legal issues surrounding monitoring during network research. In: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. pp. 141–148. ACM, New York, NY, USA (2007)