A System For Time Crictical Applications Using Temporary Access Control Policies

V.Evanjalin sabeena[1] , A.M.Rubina[2]

[1]M.E.Student, [2]Assistant professor
Department of Computer Science and Engineering, Dhanalakshmi College of Engineering,Chennai

***Abstract* - During natural disasters or emergency situations, an essential requirement for an effective emergency management is the information sharing. In this paper, we present an access control model to enforce controlled information sharing in emergency situations. An in-depth analysis of the model is discussed throughout the paper, and administration policies are introduced to enhance the model flexibility during emergencies. Moreover, a prototype implementation and experiments results are provided showing the efficiency and scalability of the system.**

***Keywords* - Access controls, privacy, security, data sharing**

I.INTRODUCTION

In the last years, natural catastrophic events, e.g., floods earthquakes, hurricanes, and man-made disasters, e.g., airplane crashes, terrorist attacks, nuclear accidents, highlight the need for a more efficient emergency management. In particular, attacks of September 11, 2001, have shown that the lack of effective information sharing resulted in the failure to intercept the terrorist attacks . This example points out the need of a more efficient, timely and flexible information sharing during emergency management. Indeed, during an emergency there is often the need to access resources that are not allowed during the normal system operations. However, such downgrading of object security classification should be controlled and temporary.To cope with these requirements, we propose an access control model to enforce controlled information sharing in emergency situations. Our model is able to enforce flexible information sharing within a single organization through the specification and enforcement of emergency policies. Emergency policies allow the instantiation of temporary access control policies (tacps) that override regular policies during emergency situations. More precisely, each emergency is associated with one or more tacp templates, describing the new access rights to be enforced during specific emergency situations. In general, in emergency management scenarios the response plans are defined by experts on the field based on regulations and laws and based on reports resulting by the emergency preparedness phase, during which emergency managers conduct a risk assessment study . We believe that all these documents represent a solid base from which emergencies, emergency policies, and emergency obligations can be specified. In this paper, we propose an extended version of the model in. One of the main extensions concerns administration policies. Indeed, this is a crucial task in every application scenario, but in case of emergency management is even more strategic due to the need of a real-time adjustment of the authorization state upon the modification of security requirements. Moreover, the specification of emergency policy requires both a security expert as well as an expert of the domain of the considered emergencies.

This is captured in our model through the definition of proper scopes that limit the right to state emergency policies only to specific emergencies. The paper also presents an in-depth analysis of the checks performed by our system to ensure policy correctness, which where only roughly sketched. Finally, the prototype implementation presented has been extended to implement the correctness validity checks and administration policy enforcement. We also report new performance tests on the prototype, more extensive, and detailed than those presented in . The remainder of the paper is organized as follows: Section 2 presents an overview of the model . Policy correctness is analyzed in Section 3. Section 4 presents administration policies. The prototype implementation and performance evaluation is provided in Section 5. Section 6 surveys related work, whereas Section 7 concludes the paper. The paper contains five Appendixes, which can be found on the Computer Society Digital Library at http:// doi.ieeecomputersociety.org/10.1109/TDSC.2013.11.InAppendix A, available in the online supplementary material, we give more details about the validity checks introduced in Section 3, whereas an additional emergency administrative policy example is presented in Appendix B,available in the online supplementary material. In Appendix C, available in the online supplementary material, we provide formal results showing the correctness of the proposed validity checks as well as of the enforcement of administration policies. Further experiments on the prototype are provided in Appendix D, available in the online supplementary material.

To enforce flexible information sharing during emergencies,it is usually necessary to grant users access to resources not normally authorized. Moreover, it is often the case that specific actions should be performed to manage the emergency. To fulfill both these requirements, the model presented in supports tacps to be activated during emergencies and obligations that have to be fulfilled when an emergency is detected. The connection of an emergency with the corresponding tacps and obligations is modeled by emergency policies. The key characteristic of the model in is that emergencies are specified through events on top of Complex Event Processing (CEP) systems . A language, called Core Event Specification Language (CESL), is used to define events describing the beginning/ending of an emergency. The formal syntax of CESL operators is reported in . Note that the identifier plays a key role in that it ensures the connection between init and end events (see for further details) as shown in the

following example, which also illustrates the reference scenario used throughout the paper. This has been chosen to show how our model works in a challenging domain, where the number of emergencies and related emergency policies is large and the level of policy granularity is high. Even if we are aware that this is not a typical domain for emergency management (e.g., disaster management), we decide to select it because it gives us the opportunity to provide more complex
examples of emergency descriptions and policies.

II.RELATED WORK

Our model enforces fine-grained access control with attribute-level granularity. Many models have been proposed in the literature, in support of fine-grained access control, for instance models derived from ABAC or the XACML standard . A remarkable model supporting fine-grained access control in a healthcare domain is C-TMAC presented This approach allows teambased access control by also integrating contextual information.

In , we intentionally gave a high-level definition ofuser with a SMS conversation using SMS modem. The user can then monitor the intrusion from anywhere, on any Internet enable device by accessing the cloud's web interface. If the intrusion is genuine, the user is provided with options to stealthily alert neighbours, play alarm sounds or even report to the police. Using these techniques, burglary can be evaded effectively. the access control model, whereas in this paper, we adopt RBAC-A. We believe the above-mentioned models can be adopted in our system instead of RBAC-A. However, none of them support emergency detection through CEP technology, which is a total novelty in access control systems. Since one of the target scenarios of our system is healthcare, we believe it is interesting to see how our proposal differs from the many existing access control models proposed for this domain (e.g., . The main difference is that these models do not support emergency descriptions and the use of CEP for emergency detection. Moreover, the literature offers also many works in disasters management domain, for instance in the field of dynamic coalitions. Dynamic coalitions are formed during international crisis, i.e., emergency situations; therefore, there are similarities between our model and dynamic coalitions models. However, none of those proposals address the dynamic reconfiguration of access control rights upon the triggering of an emergency. Rather, they address the possibility of enforcing information sharing among heterogeneous entities and related security and interoperability issues. In traditional access control models, permissions are usually known in advance, but during emergency situations a more flexible and adaptable approach is necessary. Breakthe- glass (BtG), introduced in , is an approach for such a flexible support of access control. Ferreira et al. presented a first approach based on special accounts, i.e., temporary accounts that comprise more powerful access rights with a more detailed logging. Another BtG model , presented by Brucker and Petritsch, is based on the notion of emergency levels which allow a finegrained control of policies that can be overridden using BtG. Another remarkable BtG model is based on the concept of Policy Spaces. Policy spaces are used to store regular and exception policies; when an access is not explicitly denied or permitted by a regular policy, the system checks exception policies and if there is an already planned exception the access is granted, whereas if the exception is unplanned, the system denies or permits the request whether the global status is normal or critic. Policy spaces allow also incremental policy spaces population. While BtG models are more flexible w.r.t. to our approach, they could bring the system to an unsafe state due to abuse of BtG policies. In contrast, our model is more rigid, but this is not a limitation in disaster management domain because rescue plans and information sharing requirements are defined a priori.

However, as discussed in the conclusion, we would like to extend our prototype so as to integrate BtG policies. Moreover, this paper introduces an administration model for emergency policies. Administration policies have been widely studied in the literature. Notable examples are the ARBAC97 model proposed by Sandhu et al., up to the recent XACML v3.0 administration and delegationprofile working draft , through several other work. A traditional policy administration model controls the types of policies that individuals can create/modify. In our context, our administration model controls the types of tacp that a subject can create/modify and, in addition, it controls which kinds of emergencies a subject can specify in a policy.

III.PROPOSED METHODOLOGY

The prototype is implemented in Java on top of a StreamBase CEP platform. We describe how the prototype works during the three most important phases: 1) specification of emergency descriptions, tacp templates and emergency policies; 2) emergency activation/deactivation; and 3) user access.

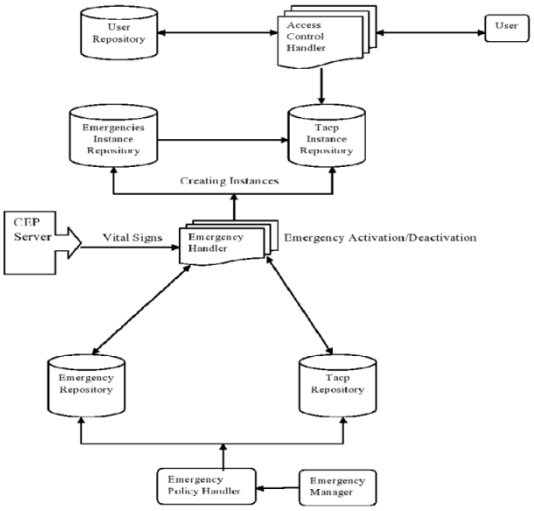

Figure1.1 System Architecture

The Emergency Policy Editor allows the creation of emergencydescriptions, tacp templates, and emergency policies. Emergency descriptions are stored into the Emergency repository. Before emergency registrations, correctness validity checks described in Section 3 are performed by thenCorrectness Checker. Similarly, new tacp templates are stored into the Tacp template repository. When an emergency manager, using the Emergency Policy Editor, creates/updates an emergency policy, then the administration policy enforcement described in Section 4 is performed by the Admin Policies Checker. If this is successfully executed, the new/updated emergency policy is stored in the system.

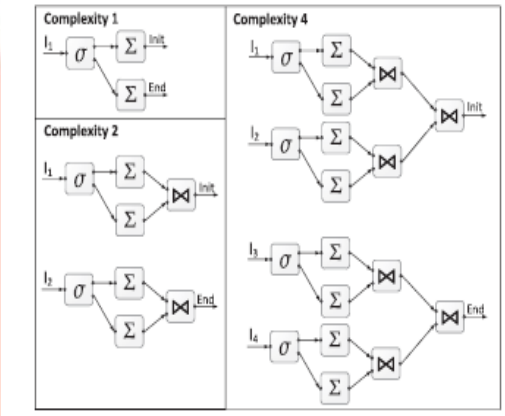

Figure 1.2 Emergency Event Complexity

Emergency activation/deactivation

Once the CEP serverreceives a tuple (1A) triggering an init event (2A), this is immediately sent to the Emergency Handler (3A). Before arriving to the Emergency Handler, the PP module checks through the PP validity check if the tuple might cause an SHP and executes one of the response actions described in Section 3. If this is not the case, the Emergency Handler retrieves from the Emergency Repository the emergency related to the received tuple (4A), if any. Then a new emergency instance is created (5A) unless another emergency instance with the same identifier has been already created (i.e., the emergency policy is already active). Moreover, the Emergency Handler retrieves from the Tacp template repository, templates related to the activated emergency (6A), if any, by also creating the corresponding tacp instance (7A). When the CEP server receives a tuple (1D) that causes the detection of an end event (2D), it sends such a tuple to the Emergency Handler (3D), which checks

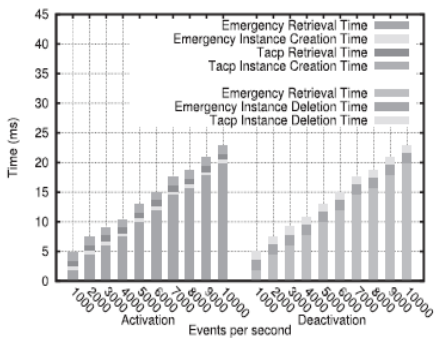

Figure 1.3 Overall Activation/Deactivation

The overall activation time represents the time elapsed between the detection of an emergency and the effectiveactivation of the corresponding emergency policy. This isgiven by:the time needed to retrieve the emergency related t the triggered init event . the time for the creation of the corresponding emergency instance . the time necessary to retrieve the tacp templat related to the emergency. the time to create the corresponding tacp instance (tacp instance creation time).

User access

When a user u successfully logs into the system (1UA), the Access Control Handler retrieves its profile from the User Profiles Repository (2-3UA). This contains profile attributes and the set of roles u is authorized to play. To compute the set of objects u is authorized to require, the Access Control Handler verifies each regular access control policy in place by returning objects identified by those policies whose authorized roles (i.e., roles specified in their subject specification) include at least a role assigned to u. To this set, the Access Control Handler also adds objects authorized by some tacp instances (4UA). The object contained in a tacp instance is returned if subject, object,
and context conditions in the tacp are satisfied.

IV.QUALITY OF SERVICE

Data Set

To carry out the experiments on emergency detection, activation, and deactivation, we developed an emergency events generator. By means of this generator, we can create a specific number of init and end events by varying their complexity, which is measured in terms of number of operators (i.e., selection, aggregation and join operators) contained into the event. As shown in Fig. 2, in case of complexity 1, the generated event takes as input a unique stream, over which it evaluates one selection and two aggregations. From this unique input stream, it generates both init and end events. With a complexity of two, the event contains two input streams, two selections, four aggregations, and two join operators. In general, in case of complexity n, the number of input streams is n, the number of selections is n, the number of aggregations is 2n, and the number of join operators is Ppnffiffi i¼1 2i (see, as an example the case of complexity 4 in Fig. 2). The emergency events generator is also able to send a certain number of tuples to input streams at a certain speed (tuples per second) so as to trigger, with a given frequency, the init and end events previously created. The impact of event complexity on the system performance is discussed in It is available in the online supplementary material. Another important aspect of the data set is the number of emergencies and tacps, which is fixed and set to 100. This set of emergencies and tacps are activated and deactivated 100 times during the experiments for a total of 10.000 emergency activations and deactivations. During experiments, the tuples rate varies from 1.000 to 10.000 tuples per second, which means that the number of activated emergencies per hour varies from 3.600.000 to 36 million. Considering, for instance, that the daily volume of 911 calls for New York city is 30.000.

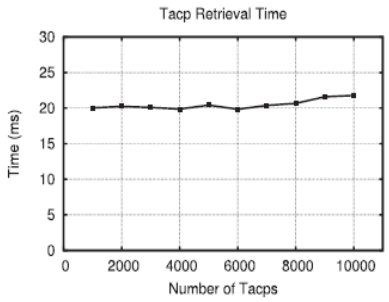

Figure 1.4 Retrieval Time

we believe that our experimental numbers are large enough to guarantee high performance in a real emergency management system.

In this experiment, we measure the time elapsed between the user login and the arrival of the set of objects on which the user can exercise privileges. The number of tacp for each role is fixed, i.e., 100 tacps for each role, which means that each role can exercise privileges over 100 objects, assuming that each tacp identifies a unique authorized object. We carried out the experiment increasing the number of roles. More precisely, in the first experiment we make use of 10 roles, because each role can exercise privileges over 100 objects, the Access Control Handlershould check 1.000 tacps. In the last experiment, we make use of 100 roles, therefore the Access Control Handler should check 10.000 tacps. The results are shown in Fig. 4. In the first case, i.e., 1.000 tacps, the retrieval time of tacp is 20 ms, while in the last case, i.e., 10.000 tacps, the retrieval time is 21.7 ms. The difference between the two times is small and guarantees a high scalability.

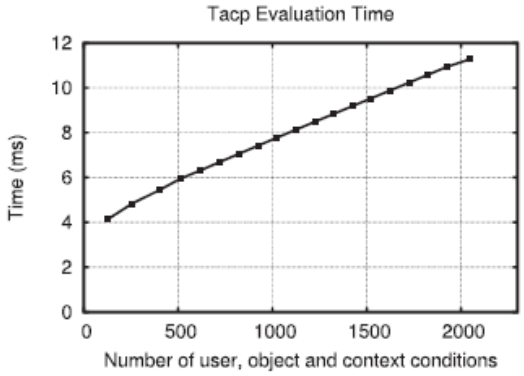

Figure 1.5 Evaluation Time

In this experiment, we measure the time required to evaluate subject, object and context conditions in a single tacp. We measured this time by varying the complexity of the tacp, that is, of its subject, object and context specifications. We vary the number of conditions from 2 to 2.048 in subject, object and context specifications. In the first case, i.e., two conditions in subject, object and context, the tacp evaluation time is 3 ms, while in the last case, i.e., 2.048 conditions, the time is 10.2 ms. The time growth is linear in the number of conditions, as shown in Fig. 5.The experiment results have shown that the prototype is fast in activation/deactivation of emergency policies and more important, the access control is not affected by the emergency policy enforcement. Moreover, increasing the number of emergencies, policies, and users, the time elapsed for each operation grows in a linear way, therefore the system is scalable.

Since one of the target scenarios of our system is healthcare, we believe it is interesting to see how our proposal differs from the many existing access control models proposed for this domain (e.g., . The main difference is that these models do not support emergency descriptions and the use of CEP for emergency detection. Moreover, the literature offers also many works in disasters management domain, for instance in the field of dynamic coalitions. Dynamic coalitions are formed during international crisis, i.e., emergency situations; therefore, there are similarities between our model and dynamic coalitions models. However, none of those proposals address the dynamic reconfiguration of access control rights upon the triggering of an emergency.

V. Conclusion and Future Work

In this paper, we proposed an extension of the emergencyaccess control model presented with the possibility of defining administration policies, i.e., which subjects are enabled to define emergency policies and over which scope. Moreover, we have implemented an extended version of the prototype presented , and we have carried out an extensive set of test to check what is the impact of emergency policies into an access control system. A set of correctness checks have also been defined to avoid useless activation /deactivation of emergencies. We plan to extend this work along different directions. First of all, we intend to carry out several experiments to evaluate the time needed for emergency administration policies enforcement. Moreover, we believe that tools to assist security administrator in emergencies and emergency policies definitions can be defined. More precisely, because a large number of risk assessment tools have been developed in the last years, we plan to analyze them so as to automatically extract emergency descriptions, and obligations from emergency scenarios and response plans. In addition, we would like to extend our prototype so as to support the break glass policies. This would allow the system to trace violations of tacps. By analyzing these violations, we plan to extract dynamic description of emergency scenarios, as well as, the corresponding information needs.

References

[1] J.G. Alfaro, "N.: Management of Exceptions on Access Control Policies," Proc. 22nd IFIP TC-11 Int'l Information Security Conf. (IFIPsec '07), pp. 97-108, 2007.

[2] C. Ardagna, S. De Capitani di Vimercati, S. Foresti, T. Grandison,S. Jajodia, and P. Samarati, "Access Control for SmarterHealthcare Using Policy Spaces," Computers andSecurity, vol. 29,pp. 848-858, 2010.

[3] M.Y. Becker, "A Formal Security Policy for an NHS Electronic Health Record Service," Technical Report UCAM-CL-TR-628,Computer Laboratory, Univ. of Cambridge, Mar. 2005.

[4] H.L. Bill Parducci, "eXtensible Access Control Markup Language (XACML) Specification 3.0," Aug. 2010.

[5] W.N. Blog, "Emergency Responders Take 911 Calls Side by Side," 2012.

[6] A. Brucker and D. Hutter, "Information Flow in Disaster Management Systems," Proc. 10 Int'l Conf. Availability, Reliability, and Security (ARES '10), pp. 156-163, Feb. 2010.

[7] A.D. Brucker and H. Petritsch, "Extending Access Control Models with Break-Glass," Proc. 14th ACM Symp. Access Control Models and Technologies (SACMAT '09), 2009.

[8] B. Carminati, E. Ferrari, and M. Guglielmi, "Secure Information Sharing on Support of Emergency Management," Proc. IEEE Third Int'l Conf. Privacy, Security, Risk and Trust (PASSAT), and IEEE Third Int'l Conf. Social Computing (SocialCom), Oct. 2011.

[9] J. Crampton and G. Loizou, "Administrative Scope: A Foundation for Role-Based Administrative Models," ACM Trans. Information System Security, 2003.

[10] T. F. E. M. A. (FEMA), "Emergency Response Plan Implementation @ONLINE," Sept. 2012.