Detecting Credential Abuse in the Grid using Bayesian Networks

Christopher Kunz, Nina Tahmasebi, Thomas Risse, Matthew Smith
L3S Research Institute, Leibniz Universität Hannover
Appelstrasse 9a, 30159 Hannover, Germany
{kunz,tahmasebi,risse,smith}@l3s.de

Abstract

Proxy Credentials serve as a principal for authentication and authorization in the Grid. Despite their limited lifetime, they can be intercepted and abused by an attacker. We counter this threat by enabling Grid users to track their credentials' use in Grid infrastructures, reporting all authentication and delegation operations to an auditing service. Our approach combines modifications to the security infrastructure with a Bayesian classifier in order to provide a reliable method for detecting abusive Grid credential usage and alerting the legitimate user. To validate this approach we created an extensive Grid simulation, simulating different types of legitimate and illegitimate use of credentials. Our experiments show that we can detect 99.5% of all abuse and our solution can thus help to increase security in the Grid.

***Index Terms*—Grid Computing, Proxy Credentials, Auditing, Abuse Detection, Bayesian Classifiers**

I. Introduction

The Grid relies heavily on public-key infrastructures, delegated authorization and single-sign on to realize its goals of an ubiquitous, easy-to-manage computing resource. However, these concepts introduce security risks that so far have hindered adoption of the Grid in sensitive areas of science and industry.

Today there is no technology for completely preventing proxy credential abuse – like unauthorized manipulation of data, resource consumption or tampering with Grid jobs. In this paper, we describe a method to reduce the risks by providing rapid notification as soon as credential abuse is uncovered. Providing the user with timely information about credential abuse and mitigation options increases security of the Grid as a whole and gives the users more control over their credentials. By executing this control, users can actively prevent unauthorized access to their data and jobs, instead of having to trust others to act appropriately.

However, because such mechanisms do not currently exist in the Grid middlewares like the Globus Toolkit [1], we propose a unified system for Grid credential auditing and abuse detection in this paper. This system builds on the concepts presented in [2], [3] and is extended with automatic classification of auditing information, giving the end user decisive feedback if their credentials might have been abused.

The remainder of this paper is structured as follows: After a brief introduction to security measures and implementations in Section II, we introduce our auditing infrastructure and give a broad overview over its components and functionality in Section III. In Section IV we go on to present our approach for automated abuse detection using Bayesian Networks before elaborating on the simulation we used to validate our approach in Section V. The final sections show some related work (Section VI) and sum up our contribution before giving an outlook in Section VIII.

II. Grid security and security risks

In current Grid environments, authentication and authorization of users and resources are key security functionalities. The relevant middlewares handle authentication by means of a public key infrastructure and X.509 certificates [4] to replace mechanisms like conventional username/password authentication. In the Globus Toolkit and its derivatives like gLite [5], a widely used implementation of this solution is provided by the Grid Security Infrastructure (GSI) [6].

In addition to providing a means of authentication, the GSI includes essential features like single sign-on and delegation of rights from users to Grid resources. It relies on proxy certificates to implement these features. In contrast to the certificates issued and signed by a trusted Certificate Authority (CA) – so-called end-entity certificates or EECs –, these proxy certificates are signed by the

user themselves. They can also be derived from another proxy certificate by using that certificate's corresponding private key for signing. By signing each certificate with its predecessor's private key, a connection between derived proxy certificates is established that allows Grid resources to resolve the certificate chain of trust up to the EEC and eventually to the CA. Proxy credentials are regarded as a trustworthy single-factor authentication token and usually inherit the original certificate owner's full set of rights. This *delegation of rights* is a key component of the GSI. In a protocol that is part of the security handshake, the delegation giver offers a delegation to the delegation receiver who then creates a proxy certificate request and has this request signed by the delegation giver. Without transmission of private keys over the network, there is now a new delegation that can be used for authentication and – if not limited by the appropriate flag in the certificate metadata – for delegation.

Security issues can arise from another peculiarity of proxy certificates: While the private key for an EEC is usually protected by a passphrase (creating a two-factor authentication device) this is not the case for proxy credentials[1]. If their private key was encrypted, single sign-on would not be possible since the correct passphrase would have to be entered by the user for each and every usage in the Grid. In addition, proxy credentials are stored in unencrypted form on Grid resources, creating a significant abuse potential. Any person with appropriate operating system privileges can read and potentially use these credentials.

If a Grid resource is successfully compromised by an attacker who has gained elevated operating system privileges, all proxy credentials on that resource at the time of the attack are available for abuse. Since the Grid is comprised of many different resource providers, one administrator's failure to secure their resources can lead to a Grid-wide security breach. The attacker can abuse the proxy credentials to consume resources in a legitimate user's name, manipulate data and jobs as well as launch further attacks on Grid resources. They can also outmaneuver a proxy credential's limited lifetime by constantly monitoring for new credentials that enter the compromised system, giving them effectively unlimited access not only on the Grid resource that they originally attacked, but all others that accept the intercepted proxy credentials. Since Grid initiatives all over the world cooperate closely, this could mean worldwide access.

This potential is deemed an unacceptable risk by many – prospective users and resource providers alike – and hinders the adoption of Grid Computing in many fields of science and industry. In addition, many regulatory bodies require detailed logging and tracking of authentication procedures. We have therefore designed a system that allows Grid users to track every usage of their delegated proxy credentials in the Grid and thus makes security measures more transparent.

[1]The term "proxy credential" implies a proxy certificate, its corresponding private key and the whole chain of issuing certificates including the EEC.

III. A Grid Auditing System

Proxy credentials are repeatedly created for various use cases during the lifetime of a Grid job. Since private keys should never be transmitted over the network, a new proxy is derived every time a communication peer (i.e., a Grid resource) needs a delegation to perform a given task. The delegation process is completely hidden from the user and they have no way to inquire about usage of their certificate. We address this issue with a proxy credential auditing system that track all credential usage. The Grid auditing system has been outlined in previous work [2], [3], so we are going to resort to only a brief introduction to the key concepts.

Our system gives the user a possibility to track all usage of his/her proxy credentials within a Grid infrastructure, see which resources currently hold a delegation and assess what exactly they are being used for. The system then aggregates this information and applies statistical methods in order to detect potential abuse. The proxy

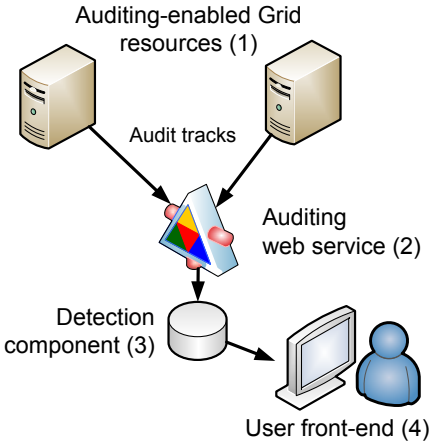

Fig. 1. Components of proxy auditing infrastructure

auditing infrastructure as shown in Figure 1 consists of the following key components: (1) Modifications to Grid middleware components to enable proxy usage logging. (2) An auditing web service that receives messages from modified components. (3) An aggregation and detection component that infers meaning into the raw data. Finally

(4), a comprehensive front-end for the user, clearly indicating abusive usage.

We have modified the libraries that form the GSI layer for the Globus Java WS-Core to check incoming proxy certificate chains for presence of a specific X.509 extension that indicates the user's wish to audit this certificate's usage in the Grid. Inclusion of such information as an X.509 extension has a number of advantages, with the most important being non-repudiation and accessibility – since the proxy credential is available at any Grid resource, no further in-band or out-of-band communication is necessary.

If said extension is present, the modified GSI libraries track the certificate usage and compose an audit record. This record is then sent over a GSI-secured channel to the central element of our system – a WSRF [7] web service which is deployed into a standard Globus container application server. The auditing web service aggregates all data and saves it to a back-end database.

This paper focuses on the component that is denoted as (3) in Figure 1. This detection component correlates usage patterns obtained via auditing records to reach a decision if a specific usage record is indicative of proxy credential theft and abuse. The victim, i.e the legitimate user whose credentials were abused, is then notified via the user front-end or other alarm mechanisms.

IV. Abuse Detection

In the previous sections, we have identified a threat to PKI-based Grids and deducted a need for auditing proxy credentials. We have also introduced an auditing infrastructure that accumulates credential usage data. In its raw, unprocessed form, the collected auditing data has little meaning for the end user – they are not familiar with many of the internal workings in a Grid and can hardly differentiate legitimate from abusive usage patterns. It makes no sense to put the burden of abuse detection on the user – this is in fact one of the key features for a comprehensive auditing infrastructure. Ideally, the system should clearly flag suspicious usage tracks and notify the user of possible credential abuse.

When designing an abuse detection component, we had to first choose between different concepts of detection rulesets. A traditional rule-based approach, as it is implemented in many security software products ranging from intrusion detection to malware removal, requires expert knowledge of the Grid infrastructure and must constantly be tuned to counter new threats. In a highly dynamic infrastructure like the Grid, this would require significant manpower. Therefore, we decided to tackle the problem of credential abuse detection using methods of machine learning.

Machine learning encompasses algorithms and techniques that allow computers to make intelligent decisions based on a small amount of manually labelled training data, significantly reducing the necessary manpower. There is no need for knowledge of underlying probabilities and thus auditing data is an ideal use case for these techniques. Additional training data can be collected whenever the user is alerted and agrees or disagrees with the system's decision. Among the various methods of inferring decisions using only limited domain-specific knowledge, Bayesian classifiers have been employed successfully for decision-making in various fields, from power-network failure detection to e-mail filtering (see section VI for some examples). They have been shown to be robust, scalable and having a high accuracy [8]. The classification is based on underlying Bayesian networks (also called "belief networks" in literature) which are used for modeling uncertain knowledge.

A. Bayesian networks and classifiers

A Bayesian network (see [9]) is represented as $B = \{G, P\}$, with G being a directed acyclic graph (DAG) and P being a joint probability distribution. Directed edges between two nodes in the DAG indicate a dependency (i.e. a probabilistic influence) between the variable denoted by the source and that of the target of the edge. For each node X in the network, the probability distribution for X is only conditionally dependent on its parents. These probabilities can be known beforehand or estimated from data, for example by means of the Maximum-likelihood estimation [10].

A Bayesian classifier combines the Bayesian network with the (estimated or known) probabilities to create a classifier. The classifier is build using training data and can then be applied to make decisions on unknown behavior. In our case, the data to decide upon is generated from proxy credential auditing – and the classifier is tasked with labelling specific audit trails as legitimate or abusive, based on its prior training.

B. Obtaining training data

Obtaining training data in a live Grid environment proved infeasible for mainly two reasons. First, since our auditing infrastructure is currently not part of any major Grid toolkit and only used in testbed environments in Germany and the UK, generation of training data with significant variation was not possible. A small test bed cannot be representative of a nationwide Grid infrastructure like D-Grid or even trans-national Grids like EGI. In addition, there are currently no known Grid abuse cases – perhaps owed to a lack of detection capabilities – to train

the classifier on. While we were able to create abuse cases in our controlled environment, doing so in an actual production Grid would be a criminal offense.

After careful consideration of the necessary parameters and conditions, we decided to undertake a simulation of a Grid infrastructure to generate the data necessary for training and validation of our classifier.

C. Modeling credential usage in the Grid

Our decision to validate the approach of Bayesian abuse detection on a simulated Grid gives us a bigger flexibility than the limited testbed data would have. We can now validate our approach even under very adverse conditions, such as very atypical Grid usage or a very high number of auditing tracks per user. Many of these borderline scenarios would be almost untestable using a lab environment – and obtaining a proof of concept in a live environment is not good practice.

A sensible model of Grid usage was necessary to create a valid simulation. From observations in our live Grid infrastructure, we took the assumption that in most regular Grid use cases, workflows are rather similar. First, a job is composed on a machine that serves as the user interface, then submitted to the job scheduling machine – in this case a Globus GRAM. Before computation commences, input data is copied to the cluster nodes. These execute the actual job and copy back any output data to the storage subsystem. These very basic assumptions stem from the fact that computation without data is rarely useful; at least not in the scientific disciplines that make up the bulk of Grid users.

We modelled Grid usage in our simulation around this basic workflow while allowing for large variations in the workflow. While some users follow the basic workflow, others make heavy use of data storage facilities using them multiple times during a job. Others show extremely compute-intensive usage patterns. We are convinced that by allowing for this variation, we have achieved a model that – although neither fit nor intended as a complete Grid model – illustrates the different ways Grids and Grid credentials are used.

For each step in the workflow mentioned above, proxy credentials play a crucial role for authentication and delegation. Each credential usage, either for authentication or as delegation source, will be reported to the auditing service providing the current location (i.e., host name), the type of usage (authentication or delegation) and the proxy's unique serial number. Using the location information as a topological indicator, we can trace the credential's "path" through the Grid. We store this topology information in a *Grid path graph*, a directed graph where each node in the graph corresponds to a resource in the Grid. There exists

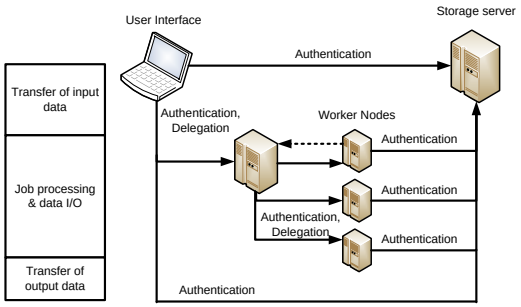

Fig. 2. A graph of credential usage during the phases of a Grid job.

an edge between node A and B if a credential was used in communication from resource A to resource B in the Grid. Using frequency information, we can estimate the probabilities that a credential will be sent between two resources in the Grid.

We can also identify the level on which the credential is located and its position among its siblings for each credential. We model the credentials with a *credential graph*. This graph has a tree structure in which the root node is the first-level proxy credential that includes the auditing extension. For each subsequent delegation, a new node is created and an edge is added that connects the node to its parent. Nodes that have the same parent are called siblings and are ordered based on time of creation. Using frequency information we can the probability that a credential has a certain number of ancestors or that a credential will be used to create a certain number of children. Together, the Grid path graph and the credential graph are indicative of abuse.

Figure 3 shows an example of a credential graph. In this example credential 1.1. is used three times for delegation and thus 1.1.1, 1.1.2 and 1.1.3 are created. These credentials are considered siblings and the numbering indicates that 1.1.1 is created before 1.1.2. The rightmost credential usages are for authentication purposes and thus no further delegations are created.

If an attacker obtains a valid set of Grid credentials on a worker node, they might use these credentials for malicious purposes, such as submitting a Grid job back to the GRAM from the worker node (as denoted by the dashed line in Figure 2). The credential would then travel along an edge in the grid path graph which is very infrequently used, i.e., differs from the usual workflow and thus raise suspicion.

One alternative attack scenario is if the attacker obtains a set of credentials and uses the same credentials for repeated delegations.Then in the credential graph the credential would have more children, i.e., there exist credentials with unusually high sibling number, than the

typical credential and hence raise suspicion. If, on the other hand, the attacker has compromised several nodes and sends credentials repeatedly back and forth between the nodes, a given credential might have many more parents than is typically known for a non-compromised credential. For this reason, we will also employ the credential paths as a method of detecting abuse.

D. From Networks to Classification

As the Grid path graph and the credential graph both contribute to our knowledge of abuse versus legitimate usage, we will use knowledge from both to create our Bayesian classifiers. We overlay the Grid path graph on top of the credential graph to correlate the information in both graphs. We consider an augmented credential graph, where each node, in addition to the unique credential id (cID), the number of ancestors ($\#ans$) and the number of siblings ($\#sib$), also keeps information about the path that the credential has taken in the Grid path graph ($gridPath$) and to which user ($userID$) the credential belongs. Each node in the augmented graph hence contains the following tuple: $(userID, cID, \#ans, \#sib, gridPath)$. Each feature of this tuple is considered to be one variable in the resulting Bayesian network. It can be argued that the $gridPath$ variable is independent from the other variables; any credential is equally likely to take a certain path in the Grid. However, while the variables $\#ans$ and $\#sib$ are conditionally dependent on cID, we will assume independence between the variables.

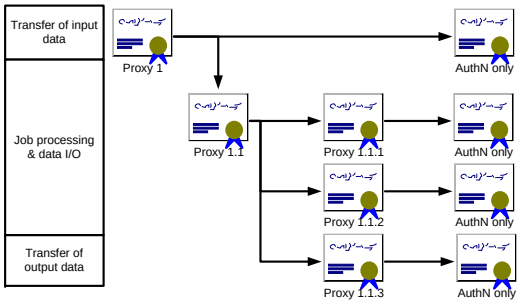

Fig. 3. The credential graph corresponding to Fig. 2.

We use the Weka software [11] to learn the structure of the Bayesian network (G from Section IV-A) from a training set. Once we have found the DAG structure, we use a Maximum-Likelihood estimation [10] to estimate the probabilities needed for the probability tables of each node (P from Section IV-A). Finally we are ready to classify previously unknown behavior as either being malicious or non-malicious.

V. Evaluation

For evaluating the performance we used our simulations to generate labelled training data. Each simulation contained a set of different user groups, each with a different user profile and workflow. We varied these profiles, the number of users, jobs per user and so on, as well as the attack to get an overview of how the system would work in a general setting. In the coming section we describe the method for generating training data. Following that, we will report on our results.

A. Simulating a Grid with WEKA

When simulating the Grid, we opt to simulate a large variety of different user behavior to best capture variance between different types of Grid usage as well as users. We ran 5 different experiments, each generated randomly based on a specific experimental profile. Each experiment was run 5 times with the same parameters and generated different, but similar behaviour, based on the profile. The experiments contain 6 different user groups that all have the same number of users. These user groups correspond to Virtual Organizations (VOs) or scientific communities in the Grid nomenclature. The behavior between user groups differ – as it would between different VOs – and there are overlapping users between the different user groups which can be the case if a user is a member of more than one VO (for example, because they are members of several Grid projects). Based on the experiment profile, users in a user group have similar behavior to each other – which is likely to happen in the real world, where users within one VO or user community often use similar tools and employ similar workflows.

Based on each VO's profile, we randomly generate user profiles and then generate credential usage accordingly. A user profile specifies the probability for a credential to "move"[2] between nodes in the Grid. It also specifies the probability that the same credential is used several times for different purposes, i.e., the probability of a *split* (see Figure 3 for an example). The maximum number of allowed splits for a credential at a given time is specified by the experiment profile and varies between different user groups. In common for all legitimate credential usage, the probability for a credential to be used on the same node is lower by a certain factor specific to the experiment profile. The similarity between the abusive and the legitimate users' profiles is varied based on the experimental profile. In some cases the abusive behaviour differs only marginally from the legitimate users, while in other cases, the difference is substantial.

[2]The credential itself does not change places, but a new delegation is created

For each user, the number of jobs is generated randomly between zero and a fixed upper limit (specified in the experiments profile).

We generated data for abusive behavior using two different additional user groups for attackers. The first case of abuse represents malicious resource consumption by submitting more jobs, reflected by creation of more root-level proxy credentials. In addition, this first abuse case uses a higher probability for a credential to "loop", i.e. be submitted to the host it was created on.

The second case of abuse represents a probing attack where the same credential is reused several times to perform different tasks. In a real Grid, this behavior is occurs when the attacker quickly scans all available resources for interesting information, using a legitimate proxy credential as the authentication token.

B. Results

The result of each experiment is a set of credential usages reported to the auditing system. Because of the random nature of the experiments, the total number of usages differ both within as well as between the different experiments. On average, we have just over 1.4 million credential usages in each experiment, with a minimum of 379 and a maximum of over 6.7 million. Additionally, the ratio between legitimate and abusive usage differs between experiments. In some experiments the ratio is extremely skewed, with very few abusive cases, while others have many more abusive cases. The skewed experiments would represent monitoring Grid usage for a long time and then having an attack occur. The amount of attack data would then be small in comparison to the legitimate data. On average abusive usage constitutes 4.5% of all usage in our experiments.

The two most interesting values in our result set are the values for precision and recall. Precision is defined as in [11]:

$$\text{precision} = \frac{tp}{(tp+fp)}$$

with *tp* being the number of correctly classified credential usage data for a class x and *fp* being the number of usages classified as x among those belonging to another class. It is thus the proportion of data which truly has class x (with x being abuse or non-abuse) among those which were classified as class x.

The recall or true positive rate for a class x is the proportion of examples which were classified as class x among all examples that truly belong to class x. Recall is defined as:

$$\text{recall} = \frac{tp}{(tp+fn)}$$

where *fn* is the number of usages that belong to class x but have been classified as belonging to another class. $tp + fn = |x|$. We employ 10-fold cross-validation [11] and precision and recall are calculated for each class (abuse/nonabuse) separately for an experiment and averaged based on the number of instances in the class. Our experiments show a high average precision and recall. For precision we have 99.5% with a standard deviation of $\pm 1.1\%$. The corresponding values for recall are $99.5\% \pm 1.3\%$.

In our current model, the rate of false positives – i.e., legitimate credential usage which is mistakenly flagged as abusive – is 0.4%, which is a very low rate. The false negative rate (abusive usages which are mistakenly flagged as legitimate) is only marginally higher and amounts to 0.7%. These results underline the very high overall accuracy and versatility of our approach.

VI. Related Work

Grid security is a widely researched field, with many projects related to ours.

The Globus Toolkit itself offers a feature dubbed "GRAM Audit Logging"[3], which, despite its name, is an essential component for usage metering and accounting, not for security purposes.

A number of projects are engaged in research towards limiting a delegation's level of authorization and thus decreasing the abuse potential of Grid credentials. All of these approaches attempt to limit the use that a stolen proxy credential has for a perpetrator.The Globus Toolkit tackles this approach by issuing special "limited" proxy credentials on several occasions.

It should also be noted that the proposed architecture is not an intrusion detection system. In conventional or Grid-based IDS (such as outlined in our previous work in [12] or the D-Grid project GIDS [13]), attack events are being detected by surveying the network or by host-local components, not by analyzing credential usage. Our project also aims at those cases where IDSes regularly fail – when a legitimate user turns malicious, his actions are an intrusion per se and therefore can rarely be detected.

In previous literature, Bayesian classifiers have been used for diverse applications such as modeling of power network failure [14], spam detection in e-mail systems [15], [16] and detection of spam bots in social networks [17]. The work by Wang in [17] shows similarities to our work as it uses graph features such as the number of friends, the number of followers as well as well as the ratio between followers and friends in order to detect spam bots on Twitter. Because of their efficiency and high accuracy that compete with state of the art classifiers, we have chosen to use Bayesian classifiers for our work.

[3]http://www.globus.org/toolkit/docs/4.0/execution/prewsgram/Pre_WS_GRAM_Audit_Logging.html

Our specific area of research has not yet been tackled in the Grid world and the approach to classify Grid credential usage using statistical methods is novel.

VII. Future Work

While our work on simulating Grid usage has served well as a proof of concept, more practical experience is a major priority for our development. In the near future, we will work at moving the auditing infrastructure from controlled testbed environments into the field to cross-validate our approach in production Grids. While other projects (i.e., the UK NGS [18]) have already adopted and built upon our concept, a "'stress test"' in cooperation with a Grid resource provider is our next goal.

To further develop our detection mechanism, we will evaluate how additional auditing attributes (such as job type or type of data retrieved with a delegation) can further improve the detection rates, especially in borderline cases.

Middleware support for the non-webservice parts of the Globus Toolkit, especially support for Globus 5, is an additional feature we plan on implementing. This will enable further resources and components for auditing and pave the way for inclusion in middlewares like gLite.

VIII. Conclusion

In this paper, we have presented a comprehensive solution for auditing of Grid proxy credentials and shown that reliable abuse detection based on Bayesian classifiers is feasible. This contribution will help pave the way for Grid applications in fields of science and commerce which require auditing for regulatory or security reasons.

We have created a model for various Grid usage scenarios that allowed us to simulate a vast spectrum of different usage patterns – from fairly standard workflows to very extreme cases. Based on this model, we ran several experiments to generate large sets of training and validation data, incorporating millions of credential audit records.

We used the generated data to build and validate a Bayesian network and classifier. Both precision and recall show consistently high values (99.5%) with little variation indicating that we are able to detect different types of abuse among a variety of different users and user groups. These results show that it is possible to automatically detect abusive Grid usage using solely Grid topology and proxy credential information. We were able to achieve a high average precision and recall, while almost avoiding false positives or false negatives.

References

[1] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," *International Journal of Supercomputer Applications*, vol. 11, pp. 115–128, 1997.

[2] C. Kunz, J. Wiebelitz, S. Piger, and C. Grimm, "A concept for grid credential lifecycle management and heuristic credential abuse detection," in *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, pp. 505 –510.

[3] C. Kunz, J. Wiebelitz, and M. Smith, "An attack-resilient grid auditing infrastructure," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, 2010, pp. 635 –639.

[4] *Recommendation X.509 - The Directory: Public-key and attribute certificate frameworks*, International Telecommunication Union Std., August 2005. [Online]. Available: http://www.itu.int/rec/T-REC-X.509-200508-I/en

[5] The EGEE Project, "glite - lightweight middleware for grid computing," [Online]. http://glite.web.cern.ch/glite/, April 2008.

[6] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*. New York, NY: ACM Press, 1998, pp. 83–91.

[7] K. Czajkowski, D. F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe, "The ws-resource framework," 2004. [Online]. Available: http://www.globus.org/wsrf/specs/ws-wsrf.pdf

[8] P. Langley and S. Sage, "Induction of selective bayesian classifiers," in *CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE*. Morgan Kaufmann, 1994, pp. 399–406.

[9] J. Pearl, "Bayesian networks," *MIT Encyclopedia of the Cognitive Sciences*, 1996. [Online]. Available: http://preprints.stat.ucla.edu/223/223.pdf

[10] K. Murphy, "A brief introduction to graphical models and bayesian networks," 1998. [Online]. Available: http://www.cs.ubc.ca/ murphyk/Bayes/bayes.html

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[12] M. Smith, F. Schwarzer, M. Harbach, T. Noll, and B. Freisleben, "A streaming intrusion detection system for grid computing environments," in *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, 2009, pp. 44 –51.

[13] W. Hommel, N. gentschen Felde, F. von Eye, J. Kohlrausch, and C. Szongott, "Architekturkonzept für ein grid-basiertes ids," Online: http://www.grid-ids.de/documents/GIDS_MS16-1.pdf, 10 2010.

[14] L.-y. He, "Application of bayesian network in power grid fault diagnosis," *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, vol. 1, pp. 61–64, Oct. 2008.

[15] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," 1998. [Online]. Available: http://robotics.stanford.edu/users/sahami/papers-dir/spam.ps

[16] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, K. Elleithy, Ed. Springer Netherlands, 2007, pp. 387–391.

[17] A. Wang, "Detecting spam bots in online social networking sites: A machine learning approach," in *Data and Applications Security and Privacy XXIV*, ser. Lecture Notes in Computer Science, S. Foresti and S. Jajodia, Eds. Springer Berlin / Heidelberg, 2010, vol. 6166, pp. 335–342.

[18] W. Jie. (2010) A proxy credential auditing infrastructure for the uk e-science national grid service - project web page. Online: http://www.nesc.gla.ac.uk/projects/pca/. University of Glasgow.