Security in the OCM-G Grid Application Monitoring System*

Bartosz Baliś[1,2], Marian Bubak[1,2], Wojciech Rząsa[1],
Tomasz Szepieniec[2], and Roland Wismüller[3]

[1] Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Kraków, Poland
[2] Academic Computer Center – CYFRONET, Nawojki 11, 30-950 Kraków, Poland
[3] LRR-TUM – Technische Universität München, D-80290 München, Germany
{balis,bubak}@uci.agh.edu.pl, t.szepieniec@cyf-kr.edu.pl,
wrzasa@student.uci.agh.edu.pl, wismuell@in.tum.de
phone: (+48 12) 617 39 64, *fax:* (+48 12) 633 80 54 *phone:* (+49 89) 289-17676

Abstract This paper presents an analysis of a solution proposed to guarantee the security of a grid application monitoring system – the OCM-G. Two aspects of security are distinguished: inter-component communication and forge-component attack. The solution is based on methods of the Grid Security Infrastructure. We show that even authentication, encryption and integrity via digital signatures are not enough to achieve a desired security level. In our system, a special protocol of extending of the logical structures in the OCM-G, so called Virtual Monitoring Systems, was needed in addition to the standard security mechanisms.
Keywords: grid, application monitoring, security, GSI tools

1 Introduction

The Grid is a distributed computing infrastructure designed for advanced science and engineering. The concept of the Grid leans on resource sharing that is not an ordinary file exchange, but *direct access to computers, software, data, and other resources* [6]. Careful control of the sharing requires a well defined infrastructure.

Applications designed for the Grid environment are highly distributed and their development process is significantly complicated. Thus, there is a high demand for Grid-enabled tools for application development. These tools are various in their functionality (e.g., debuggers, performance analyzers, etc.), but they all need a *monitoring infrastructure* to observe and/or control an application. The OCM-G – a Grid-enabled application monitoring system, described in detail in [2], is such an infrastructure.

This paper focuses on security issues related to the OCM-G. At the beginning, it analyses security requirements specific to Grid applications, and then, after presenting the architecture of the OCM-G, it explains the security problems

* This work was partly funded by the European Commission, project IST-2001-32243, CrossGrid [5]

and proposes adequate solutions. We show that just using the standard security mechanisms such as authentication, encryption, and digital signatures is not sufficient; we need additional protocols in order to address all security problems.

2 Security in the Grid

Grid applications use public Internet links to communicate between sites. Therefore no additional network infrastructure is necessary. While this is a significant advantage of the Grid, it also results in additional issues that should be considered. Security – including authorization as well as privacy, authenticity and integrity of transmitted data – is one of them.

2.1 Internet Security

Nowadays the vulnerabilities of the protocols used in the Internet and connected LANs, such as TCP/IP and Ethernet, are widely known and well described [3], [4]. Since a precise description of network vulnerabilities is out of scope of this paper, we only present a brief description of threats.

Sniffing or *eavesdropping* is a significant threat to the confidentiality of a transmission. In a non-switched Ethernet every network interface can read any packet transmitted in the same network segment. In other networks, sniffing is possible by compromising switches, routers, or the links themselves.

Spoofing describes vicious manipulations, which are possible for most of the common Internet protocols. *DNS spoofing* allows to impersonate a host by forging an association between the host's name and the attacker's IP address; *IP spoofing* makes an attacker capable of deceiving authentication methods based on the source IP address of a packet; *ARP Spoofing* allows a malicious host to become an agent between two other hosts, and fully control their connection.

Session take over (or session hijacking) enables an attacker to "steal" an already established TCP/IP session. If authentication is performed only on initialization of a connection, this attack makes it possible to access a host like an authorized user.

A solution which removes the threats mentioned above has been introduced by asymmetric cryptography (or public-key cryptography) techniques [9]. Algorithms based on asymmetric cryptography are capable of establishing secure connections over insecure public Internet links. The requirements of a secure transmission are met.

Various concepts based on asymmetric cryptography enable security in several aspects. (1) **Authentication** is performed by the use of certificates. Protocols based on public-key cryptography are capable of unquestionable verifying the identity of the peer, independent of the unreliable authentication mechanisms used by the network protocols. (2) **Authenticity and Integrity** can be ensured by authenticating each packet transmitted over the network. This is achieved by the use of digital signatures, so that the receiver of the message can verify the identity of the sender. (3) **Confidentiality** is achieved by encrypting all network traffic. Encryption is usually based on symmetric ciphers, which are significanly faster than asymmetric ones, however, asymmetric cryptography is needed to exchange the symmetric *session key* in a secure way.

2.2 Grid Security

All issues described above are essential for the security of Grid middleware and applications. Communication between sites should be secure and reliable. However, communication is not the only security issue important in the Grid. To addres the more complex security requirements of the Grid, the Global Grid Forum Security Working Group was formed to work on the Grid Security Infrastructure (GSI) project [7]. The goal of this project is to work out a consistent security infrastructure for the Grid environment. Below we present a brief summary of the requirements described by the GSI working group.

Single sign on – a user should be able to 'log on' the Grid once and then should be able to access all resources he is authorized to.

Delegation (restricted) – a user should be able to make a program to run on that user's behalf. Consequently, such a program should be able to access all resources the user is authorized to. Optionally, the program should be able to make a further delegation to yet another program. In order to minimize threats connected with misused or compromised delegated credentials, a restriction for delegation of authorization rights should be supported.

Integration with local security solutions – since different sites use different security policies, GSI should be able to interoperate with them. It cannot require the replacement of local security solutions.

User-based trust relationship – the user should be able to use resources from different sites without the need of interaction of these sites' administrators.

Authorization – resource owners should be able to control which subjects can access the resource and under what conditions.

At present, there is no single standard addressing all these requirements, therefore, the GSI working group works on a new solution, that will be an extension of existing standards, meeting above requirements. Existing solutions used by GSI rely on asymmetric cryptography.

3 OCM-G Monitoring System

3.1 Architecture

The OCM-G is a collection of two types of components: per-site Service Managers (SMs), and per-host Local Monitors (LMs). Additionally, a part of the OCM-G is linked to the application's executable (Application Module) (see Fig. 1).

The communication topology between these components is as follows: (1) LMs are connected to the SM of their local site, (2) any pair of SMs can be connected, if necessary. Additionally, a tool can connect to any SM and send monitoring requests. Consequently, monitoring requests are distributed in the OCM-G in the

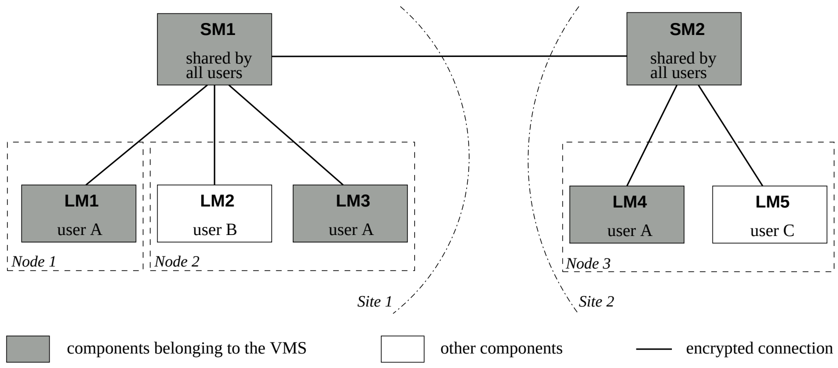

Figure 1. OCM-G architecture

following way: (1) the SM which accepted a request, splits it into sub-requests and forwards them either directly to its LMs, or other SMs (if a sub-request operates on an object on a remote site), (2) sub-requests are executed in LMs, (3) replies are collected back via the same route as the distribution was done, until the tool receives a single reply to the request. Each LM belongs to one Grid user (thus, there may be more than one LM on each host – one for each Grid user), while SMs are shared by many users. However, SMs do not need to run as priviledged processes. An important notion in the OCM-G is the *Virtual Monitoring System* (VMS). Each application has its own VMS which is simply the set of OCM-G components (SMs and LMs) which are involved in monitoring the processes of this application. Note that a VMS is a dynamic entitiy; for example, if a new application process registers in the OCM-G, the VMS may be extended by a new host (LM) or even site (SM). VMSs are important for information distribution – if the state of an application changes (e.g., a new process is created), only components of the associated VMS should be notified, not the whole OCM-G. We will see that VMSs are also important for security.

3.2 Security Requirements

The OCM-G is a common monitoring infrastructure which allows multiple users to monitor many applications, which not only includes data collection, but also manipulation. Security requirements which must be met in such a system are as follows.

(1) *Credibility.* Monitoring results obtained from the system should be correct, specifically, a deliberate alteration should be excluded.

(2) *Process Security.* We must not allow an unauthorized person to take control of another user's processes. In the Grid, it is especially important since in this way the person may obtain the other user's proxy credentials and will be able to pose as this user.

(3) *Confidentiality.* The monitoring results are transmitted across OCM-G components. We should provide a possibility to keep them confidential, at least if a user explicitly asks for it.

(4) *Site security.* An additional system service or an additional connection channel for the SM should not lower the security of the site.

In order to respond to the above requirements, connections between OCM-G components should address the following aspects: (1) *Authentication* – on connection establishment, each party should be able to unquestionably identify the other peer. It cannot be achieved with traditional network protocol methods, since they can be easily deceived. (2) *Authenticity and integrity* – each OCM-G component should be able to verify who is the author of a received piece of information. This should also ensure that information was not altered during the transmission. (3) *Confidentiality* – no one should be able to eavesdrop transmissions between OCM-G components.

4 Security in the OCM-G – Problems and Solution

4.1 Communication Security – GSI

Inter-component communication in the OCM-G meeting the requirements outlined in the previous section is achieved by the use of GSI solutions. In order to perform reliable authentication, certificates are used. Authenticity and integrity of transmitted data is achieved by the use of digital signatures. The data transmitted between peers of the connection are encrypted by GSI. The symmetric *session key* is known only to the communicating peers, which ensures confidentiality.

As we mentioned above, GSI uses certificates and asymmetric cryptography based algorithms to establish a secure channel over insecure network protocols. Consequently, each OCM-G component as well as any tool connecting to the OCM-G should be provided with certificate.

The **tools** and **Local Monitors** always belong to a specific user and work on behalf of him. Since each Grid user has his own certificate which confirms his identity in the Grid, it seems to be natural for these components to use credentials of their owner. Therefore, no additional certificates are required for tools and LMs.

Service Managers are shared between users, therefore they cannot use credentials of any of them. Users who run multi-site applications are forced to trust the remote sites' SMs, thus they should be able to recognize each SM on the basis of its identity. Moreover, to ensure the required security level, we should be able to distinguish different SMs, consequently each SM should use a different identity confirmed by a corresponding certificate. Additionally, it would be useful to be able to unquestionably verify the location of a SM by the use of its identity.

Therefore, we have decided to provide each SM with a certificate with a specific *Distinguished Name* (DN) in the *Subject* field. Since a DN has a hierarchical structure, it will be possible to obtain information about the location of a SM from its identity. It should also be possible to obtain a certificate for SM with a subject corresponding to its site, and with a fixed *Cannonical Name*. E.g., the DN of a Service Manager for site `cyfronet.krakow.pl` could look like:

```
/C=PL/O=GRID/O=Cyfronet/OU=cyfronet.krakow.pl
/CN=OCM-G Service Manager /Email=root@cyfronet.krakow.pl
```

On establishing a connection with a SM, we are able to verify that the peer address matches the address in the DN, and that the certificate was issued for a Service Manager.

4.2 Threat of Forging a Service Manager

We assume that all security solutions related to communication and authentication/authorization described in the previous section are applied. We will show that this is not enough – there is a threat that is not removed despite of the use of GSI to ensure security of communication and authentication.

Source of the threat – forged Service Manager. The problem we consider results from the fact that everyone is able to run a Service Manager. One may obtain the SM's source code and modify it, or even write his own program to seemingly behave as a Service Manager. Such a forged SM could be used to control all monitored applications belonging to different users on different sites. This is possible since once a component is part of the VMS, it is allowed to send any monitoring request to LMs. Note that for the LM it is necessary to trust the SMs, since the SMs need to split requests and combine replies, i.e. they need to modify messages on behalf of the user.

Site administrators – trusted persons? Since we use the GSI to authenticate the peer of each connection, we can identify a Service Manager on the basis of its certificate. Thus, the one who wants to perform the forged-SM attack would need a valid SM certificate. Normally, only a site administrator can obtain an SM certificate. A principal assumption of many security solutions is that we *can* trust the site administrator. More exactly: we should assume that a site administrator will not make harm to his own users. However, that is exactly the point in this case. An administrator who could simulate a valid SM would be able to control the activity of processes belonging to **other users** on **other sites**. This is obviously not acceptable, since we cannot demand site administrators to trust administrators of other sites.

Solution – secure VMS extension protocol. Conceptually, the solution is simple: since only a member of a Virtual Monitoring System is allowed to send requests related to the application associated with this VMS, we should secure the protocol of extending a VMS. Obviously, we should grant the membership in a VMS only to those SMs which are involved in monitoring of the application associated with the VMS.

This is possible, since the current procedure of joining a VMS is bottom-up: (1) the request to join comes from a LM to the SM; (2) the SM contacts other SMs, already in the VMS; (3) when all the SMs of the VMS share the information about a new SM, the VMS is extended. Thus, we can use a digital signature to **verify that the SM requesting membership in the VMS previously received the appropriate request from a LM**.

Thus, the extended registration message should be prepared by a Local Monitor in the following way: The registration message received from the application process is concatenated with the SM's Distinguished Name and signed by the LM on behalf of the user. Note, that the DN of the Service Manager is known by the LM, since a GSI secure channel is established between them before any request is sent. Such an **extended register message** is an irrefutable proof that the **user wishes** this SM to be attached to the Virtual Monitoring System.

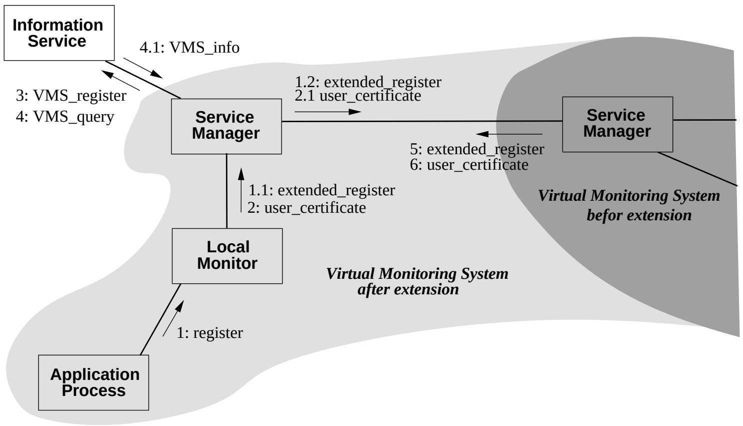

Figure 2. VMS extending protocol.

The secure protocol of the Virtual Monitoring System extension is presented in Fig. 2. It uses network links secured by GSI and consists of the following steps:

1. An application process sends a `register` message to the Local Monitor (message 1).
2. The Local Monitor connects the Service Manager (SM1). When the connection is established, it knows the Distinguished Name of SM1.
3. The LM prepares the `extended_register` message and sends it to SM1 (message 1.1) with the user's proxy certificate (message 2). The certificate is necessary to verify the user's signature.
4. SM1 registers in an information service (message 3) and obtains information about other Service Managers involved in the application (VMS) (messages 4 and 4.1).
5. SM1 requests connections with other SMs from the VMS. After the connections are established, the SMs exchange their `extended_register` messages (messages 1.2 and 5), check the signatures and the SMs' Distinguished Names, and update their information about the VMS.

The main principle of this solution is as follows. (1) A Service Manager, to join a VMS, should present a *written permission* of the application owner to the other SMs belonging to VMS. (2) A SM joining the VMS should extend his VMS data structures with information about the other SMs only if these SMs present the user's *written permission*, issued for these SMs. The *written permission* is implemented via a digitally signed register request containing the SM's Distinguished Name.

5 Summary

In this paper, we have shown a concept of the security solution for the OCM-G Grid application monitoring system. Communication between OCM-G components is secured with methods introduced by the *Grid Security Infrastructure*. Thus, secure channels can be established, to avoid network protocols vulnerabilities. A second aspect of the OCM-G security is connected with components that are shared between different users and applications. We have shown, that by using forged components, site administrators could get control of a user's processes on remote sites. To solve this problem, we introduced a secure protocol for adding new OCM-G components to an existing *Virtual Monitoring System*. The protocol uses digital signatures to ensure that the VMS is extended only on a written permission of the user.

Acknowledgement We are very grateful to Dr Włodzimierz Funika for many discussions and helpful remarks.

References

1. Adams, C., Farrell, S. *Internet X.509 Public Key Infrastructure Certificate Management Protocols*, RFC 2510
2. Baliś, B., Bubak, M., Szepieniec, T., Wismüller, R., and Radecki, M.: OCM-G – Grid Application Monitoring System: Towards the First Prototype. Proc. Cracow Grid Workshop 2002, Krakow, December 2002.
3. Bellovin, S., *Security Problems in the TCP/IP Protocol Suite*, Computer Communication Review, 19(2):32-48, April 1989. `http://www.research.att.com/~smb/papers/ipext.ps`
4. Bellovin S. *Defending Against Sequence Number Attacks*, RFC 1948.
5. The CrossGrid Project. http://www.eu-crossgrid.org
6. Foster, I., Kesselman, C., Tuecke, S. *The Anatomy of the Grid*, Intl. J. Supercomputer Applications, 15(3), 2001.
7. GSI working group homepage: `http://www.ggf.org/security/gsi/index.htm`
8. Ludwig, T., Wismüller, R., Sunderam, V., and Bode, A.: *OMIS – On-line Monitoring Interface Specification (Version 2.0)*. Vol. 9 of LRR-TUM Research Report Series. Shaker Verlag, Aachen, 1997. `http://wwwbode.in.tum.de/~omis/OMIS/Version-2.0/version-2.0.ps.gz`
9. Menezes, A., van Oorschot, P., Vanstone, S. *Handbook of Applied Cryptography* CRC Press, 1996. `http://www.cacr.math.uwaterloo.ca/hac/`