An attack-resilient Grid auditing infrastructure

Christopher Kunz, Jan Wiebelitz and Matthew Smith
Distributed Computing Security Group
Gottfried Wilhelm Leibniz Universitaet / L3S
Hannover, Germany

Abstract

As recent experiments have shown, current Grid infrastructures are highly vulnerable against root exploits. In these attacks legitimate Grid user credentials were used to compromise vulnerable Grid head and worker nodes. Any such attack against a distributed working environment is critical.
However, in the Grid it is particularly devastating, attacks against the head node affect unencrypted Grid proxy certificates. Using these, an attacker can act with the permissions of the original owner, undermining the Grid security concept. Even after the original attack has been detected and the affected systems have been sanitized, the attacker is still in possession of the stolen proxy.
In previous work we introduced an auditing infrastructure that gives Grid users a way to reconstruct usage of their delegated credentials and detect their possibly abuse . We achieve this by including an X.509 certificate extension in a proxy credential signed by the Grid user – making the users request to track credential usage tamper-proof.
In this paper, we extend the auditing infrastructure by a novel encryption aware watchdog, which can detect proxy certificate misuse even in the face of complete root compromise of all accessible Grid resources. It correlates network communication in the Grid with the auditing infrastructure and can thus detect proxy certificate misuse and tampering with the auditing framework. ***Keywords***—

X.509, proxy certificate, abuse detection, auditing, security, PKI, OCSP, revocation, network security, network sniffing, SSL, TLS, certificate

I. Introduction

Securing Grid middlewares has been a hot topic for many years now and significant improvements have been made. However, the Grid is still far from bullet proof. Grid proxy credentials in particular are still an open issue, since their abuse enables an attacker to compromise entire Grid landscapes. They are also particularly critical since compromise of a proxy credential allows the attacker to pose as user and steal or modify his data and software. While preventing this sort of attack is a very important goal, it is equally important to be able to detect the misuse cases which could not be prevented. In previous work, we presented a solution with which credential use could be audited to enable users to detect unauthorized use. This solution was integrated into the Grid headnode and is resilient against attacks with user privileges. However, since recent experiments have shown gaining root privileges within the Grid (see section II) is trivially easy , this is no longer sufficient.

C. Kunz, J. Wiebelitz and M.Smith are with the Distributed Computing Security Group, Gottfried Wilhelm Leibniz Universitaet in Hannover, Germany. They can be reached under {kunz,wiebelitz,smith}@dcsec.uni-hannover.de.

978-1-4244-5849-3/10/$26.00 ©2010 IEEE

A. About this paper

In this paper, we present a novel extension of our certificate auditing solution to make the system resilient in the face of attacker with root privileges. The paper shows several types of attacks against Grids and the auditing infrastructure and shows how the presented solution can still reliably audit Grid proxy credential misuse.

In the papers first section, we introduce the reader to current practice in Grid security, review related work and outline the concept of proxy credential auditing. In section III, we present possible threats against the auditing infrastructure and – where applicable – provide adequate solutions. Section IV points out some security issues inherent in the current architecture; a solution to which is presented in section IV.
The idea of using a network sniffing watchdog application is evaluated on basis of experiments in section 5. We finalize with a conclusion and an outreach on future work.

B. Current practice in Grid Security

Modern Grid infrastructures rely on an X.509 [3] PKI for authentication and authorization. Typically, users and infrastructure components are issued End-Entity Certificates (EEC) that are used to reliably identify them in the Grid. Users can also derive so-called "proxy certificates" [4] from their EEC; these are certificates with a shorter lifetime (typically between 12 hours and 7 days), signed by the user themselves instead of a trusted Certificate Authority (CA).
A widely used implementation of a such PKI is the GSI [5], implemented and maintained as a part of the Globus [6] project. In the GSI, Proxy credential and EEC are stored in the file system and are protected by the according ACLs. This protection does not prevent attacks by users with elevated (i.e. root) privileges.

Due to the design of the Grid, there is currently no way to track a proxy credential's usage during its lifetime. In the optimal case – when all participants in a Grid are trustworthy –, this is not an issue. However, the Grid is no longer a completely benevolent environment. Even in purely scientific environments, competition can be fierce.

C. Related work

Apart from our project described in section I-D, there is currently no work that suggests auditing Grid credentials. With regards to anomaly detection by other means, traditional and Grid-specific intrusion detection systems (such as outlined in [10]) use different approaches to abuse detection. Our idea to detect abuse in encrypted Grid network traffic by selectively sniffing and decrypting data streams has so far not been attempted in the Grid context.

D. A proxy credential auditing system

In previous work, we introduced our concept for proxy credential auditing. We will briefly recap the solution here, for further information the reader is referred to [8] and [9]. Our approach includes three main components:

1) An X.509 certificate extension that – if present in a proxy certificate – signifies that this certificate's usage should be audited,
2) Modifications to the Grid Security Infrastructure to allow sending of auditing information (so-called "audit tracks") to a remote location,
3) and a Web Service that receives and stores audit information and aggregates it into "audit trails".

It is an important property of our solution that the certificate owner can choose if they want their delegated credentials to be audited or not. With our solution, an extension is embedded into the first proxy certificate before it is signed by the user – any modification would void the signature.
Our auditing infrastructure is currently available for deployment in Globus 4 WS-Core based resources (e.g. OGSA-DAI or GRAM4) and under development for the C GSI implementation (e.g. GRAM5, MyProxy, GridFTP). Development takes place in a Globus Incubator Project[1].

II. Attacks on Grid sites

Most Grid sites offer interactive access via GSI-secured or standard SSH sessions to one of their nodes to facilitate development and deployment. Since GSI-SSH allows the use of proxy credentials as an authentication token, any Grid user with a valid proxy can access these resources.
In 2009, there were several critical security issues in the Linux kernel which were trivial to exploit.We carried out a random inspection of interactive nodes in the German D-Grid infrastructure in November 2009 and found about half a dozen vulnerable nodes which, given the size and administrative constraints, is a comparatively good result, however, since all the attacker usually needs is a single point of entry to gain access to proxy credentials this still offers easy way to totally compromise the Grid.
All of the vulnerable nodes carried valid proxy certificates and had host certificates signed by an EUGridPMA [13]-conformant CA. Attackers can use these certificates for their purposes – i.e. submit jobs, steal or change data, sniff network traffic, etc.

Thus, due to the complex nature of current Grid infrastructures and the multitude of components involved, we must assume that attacks on Grid sites – either from the inside or by third parties – will be successful in many cases and gain elevated privileges on a Grid resource. We must take this into account when evaluating possible threats to our auditing infrastructure.

III. Attacks on the auditing infrastructure

In section II, we have shown that it is quite easy for an attacker to gain elevated privileges on a Grid resource. If this resource receives or creates delegations, the attacker can obtain these delegations and use them for further abuse. Attackers who try to bypass the existing Grid credential auditing system can try to circumvent it in a number of ways. We have designed the system so that it will withstand most attacks and will now elaborate on attack scenarios and countermeasures and discuss the potential weaknesses.

a) Removal of the certificate extension: As pointed out in section I-D, a proxy credential is flagged as "to

[1] http://dev.globus.org/wiki/Incubator/Proxy-Audit

audit" by embedding an X.509 extension in it before signing.Since the GSI checks the chain of trust for a proxy credential by checking the signature on each proxy in the certificate chain, there is no way for an attacker to remove the certificate extension from the first proxy without invalidating the whole proxy certificate chain.

b) Disruption of communication to the Auditing Service: Instead of taking the relatively "loud" approach of performing a DoS attack on the Auditing Service, an attacker who has gained elevated privileges on a Grid node can prevent that node from communicating with the Auditing service (for example by setting local firewall rules). Since it is not possible to reliably detect these attacks on a Grid node itself, this attack vector is effective against a normal auditing infrastructure. The solution presented in this paper will be able to detect this form of tampering.

c) Removal of the auditing-enabled GSI libraries Another effective form of attack is the removal of the auditing components from the headnode. As pointed out earlier, our solution requires a modified GSI implementation that includes auditing hooks.

If an attacker has succeeded in becoming the root user on a Grid resource, they can simply replace these libraries with the default Globus libraries and effectively disable auditing.

The presented solution will be able to detect this form of misuse while at the same time retaining backwards compatibility (by not marking the certificate extension as "critical").

d) Significance of attacks: Some resource administrators have argued that once a Grid host has been compromised, there are so many points of abuse that credential auditing becomes obsolete. From a single resource administrator's perspective, this argumentation could seem correct – if a Grid host was compromised, malicious job submission from that host is possible even without copying proxy credentials.

However, globally this argument does not hold. Grid users are primarily concerned with the integrity and privacy of their data and algorithms. The auditing infrastructure needs to be reliable even if the Grid headnode has been completely compromised.

To overcome shortcomings in the initial auditing concept, we have created a concept for an "auditing watchdog" which monitors site-internal Grid traffic and detects irregularities in auditing. This adapts the concept of dual control to the world of auditing Grid proxy credentials.

IV. A Grid auditing watchdog

The new component – dubbed "audit watchdog" – is essentially a network tap which is situated within a Grid site, in the same network segment as the machines that comprise that site. It monitors communication between components in that Grid site (and that site *only*) and the local end of communication with the outside as applicable. The watchdog only looks for occurrences of the SSL handshake that precedes GSI-secured communication. The client's proxy chain is always provided as a part of this handshake. If auditing has been requested by the user, one of the proxies in this chain includes the auditing extension.

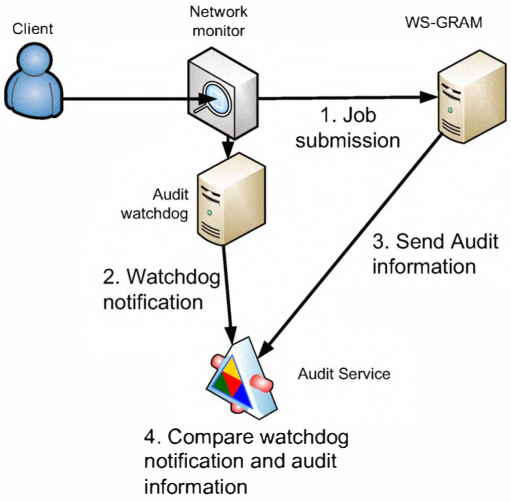

Fig. 1: Auditing infrastructure w/watchdog

Figure 1 denotes how the different components work together:

1) The user submits a job using an auditing-enabled proxy credential.
2) The audit watchdog recognizes the appropriate X.509 extension in the proxy and notifies the Audit Service.
3) During job submission, the WS-GRAM submits an audit track to the Audit Service.
4) The Audit Service compares the notification received by the watchdog and the audit track sent by the WS-GRAM.

If step 3) is ommitted, the Audit Service cannot match the watchdog notification with an audit track. This chain of events indicates that something is wrong: Audit tracks were not sent by the WS-GRAM, although it was indicated that the proxy should be audited. Due to backwards compatibility reasons, two possible explanations for this behavior exist:

(a) The resource in question is not auditing enabled.
(b) The resource in question was auditing enabled, but the auditing libraries have been replaced/tampered with.

Clearly, option b) should raise suspicion. It indicates that either a Grid site administrator has mistakenly replaced the auditing enabled libraries or an attacker has gained

administrative privileges on the machine in question.
In this case, the auditing service will raise an alarm when it finds watchdog notifications without a matching audit track.

1) Decrypting SSL streams: All TLS/SSL communication is preceded by a handshake period in which ciphers and cryptographic protocols are negiotiated between the peers. No matter what protocol is used for the exchange of actual information, all communication can only be decrypted by a third party if they possess the appropriate keying material.
In Globus job submission, the job submission client receives the GRAM's certificate during the handshake period and uses the public key in this certificate to encrypt messages to the GRAM. Therefore, the GRAM's private key is necessary to decrypt these messages.

We propose that the responsibility for the auditing watchdog applications is split and delegated on the site level and each distinct Grid site will run their own watchdog application.
In this case, only the private keys for those hosts which are located in the Grid site are required by the watchdog. This means that the private keys do not leave the administrative domain and remain under the full control of the person or persons who are in charge of them anyway. Since the watchdog server is a passive component and offers no interactive login, storing an additional copy of the Grid private keys is not critical.

2) Security implications: As a number of resource host keys are stored on the auditing watchdog, the administrator has to ensure that the auditing watchdog server is secure against attacks. It should be made sure that its only two means of communication should be passively listening to the network communication via a dedicated network tapping interface and sending of notifications to the Auditing Service , thus the only way to log on the maschine is to physically connect to it.
With the combination of these restriction and operating system hardening, the auditing watchdog system can be considered to be highly secure and a valuable security resource for the Grid.
Since it is not easily possible to harden the diverse, flexible and powerful headnode it is possible to create a small, non-interactive, external monitoring system which can alert users to potential misuse of their proxy-credentials.

V. Evaluation

In Table I, we have summarized which attacks against a Grid auditing infrastructure are detected with the concepts introduced in this paper.

ISSUE	NO WATCHDOG	WATCHDOG
Removal of certificate extension	✓	✓
Disabling of audit service	✓	✓
Disruption of communication with audit service (with root privileges)		✓
Removal of auditing functions in GSI (with root privileges)		✓
Arbitrary code injection attack against the watchdog		(✓)[7]

TABLE I: Attacks detected by audit watchdog

With regards to the network watchdog's performance, it is not necessary for the auditing watchdog to decrypt all traffic on the Grid site. Using transport-level security, the only relevant part of a web service communication is the protocol handshake since this first part of an encrypted session contains the important proxy credential information. Therefore, even in large installations there will be relatively little data to decrypt; the full SSL handshake around than 10 packets with a total size of about 3KByte on the wire.
We have set up a simple virtual testbed to test our proof-of-concept system consisting of two VirtualBox VMs with 1 GByte RAM each, running on a dual core host computer. The tests were conducted with Ubuntu 9.04 VMs, one of which was running a WS-GRAM and a GridFTP server. We submitted simple batch Globus jobs in a loop (about 1 job per second) and used Wireshark to monitor the resulting traffic. The stock GnuTLS-based SSL dissector that comes with WireShark can decrypt SSL streams when provided with a list of hosts and corresponding keys. It decrypts all traffic and is therefore not indicative of final performance.
The tests showed that our proof-of-concept system was able to follow job submission streams for multiple WS-GRAM installations, even if decrypting the complete SSL session (including WS calls). To put further load on the system, we introduced a GridFTP transfer at line speed (10 MByte/sec on virtualized Fast Ethernet interfaces, reading from `/dev/random` and writing to `/dev/null`) and selectively captured only the GridFTP control channel. Communication on the control channel contains about 40 packets until the data channel is opened – this is still a relatively low figure. Accordingly, load on the network sniffer was very low, similar to the load imposed by the WS-GRAM job submission test. We conclude that under the conditions mentioned, even a large amount of communication can be monitored and decrypted by commodity hardware.

[7]Due to the fact that the passive watchdog server's OS can be locked down significantly, most exploit code can be reliably prevented from affecting system integrity. None the less, while we deem the risk to be very small, it is still existant.

VI. Conclusion

In this paper, we have shown that root privilege attacks are easy to execute in production Grid environments and thus that proxy credentials are easy to steal and abuse. We discussed a number of attacks against proxy credential auditing services and show how our solution is capable of detecting misuse in the face of these attacks. By implementing an "audit watchdog" that analyzes GSI traffic between nodes in a Grid environment, we have found a possibility to guarantee auditing of proxy credentials even if several hosts in a Grid are compromised and their GSI libraries have been tampered with. The solution can be integrated transparently and unintrusively. Full control over all host certificates and private keys stays with the current administrative staff. The capability of guaranteed auditing of all proxy-credential use is a significant step towards building trust in the Grid for commercial users, since abuse of proxy credentials is still one of the most significant risks posed to the privacy of user data and software.

The evaluation (section V) showed that for the small test site built to evaluate the qualitative aspects of the system performance was not an issue. Specialized protocol dissectors for GSI-secured HTTP and GridFTP traffic can further reduce the performance requirements by selective only monitoring the handshake instead of the entire traffic stream as currently is the case with the Wireshark solution. This implementation will be released as an add-on for the proxy credential auditing Globus incubator project.

A. Future work

Our evaluation (section V) showed that selectively monitoring Grid data with a network tap is feasible, but no statement with regards to stability and scalability can be made at this point. It is therefore our next step to implement a libpcap-based network sniffer with dissectors specialized to GSI-secured HTTP and GridFTP traffic.
We will deploy the resulting network monitor in our testing environment and focus on generating real-life data as well as performance metrics. Cooperation with other national and international Grid projects is planned and will be taken up in 2010.

References

[1] M. Humphrey, M. Thompson, and K. Jackson, "Security for grids," *Proceedings of the IEEE*, vol. 93, no. 3, pp. 644–652, March 2005.

[2] M. Humphrey and M. R. Thompson, "Security implications of typical grid computing usage scenarios," *Cluster Computing*, vol. 5, pp. 257–264, July 2002.

[3] *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

[4] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson, "RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile," [Online]. http://www.ietf.org/rfc/rfc3820.txt, IETF, June 2004.

[5] 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.

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

[7] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1998, pp. 83–92.

[8] 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*, April 2009, pp. 505–510.

[9] C. Kunz, C. Szongott, J. Wiebelitz, and C. Grimm, "Design and implementation of a grid proxy auditing infrastructure," in *eScience 2009, 5th IEEE International Conference on (to appear)*, December 2009.

[10] A. Schulter, J. A. Reis, F. Koch, and C. B. Westphall, "A grid-based intrusion detection system," in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL)*, 2006.

[11] T. Ormandy and J. Tinnes, "Linux Kernel 'sock_sendpage()' NULL Pointer Dereference Vulnerability," 2009. [Online]. Available: http://www.securityfocus.com/bid/36038/info

[12] Common Vulnerabilities and Exposures Project. (2009) CVE-2009-3547. [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3547

[13] EUGridPMA, "European Policy Management Authority for Grid Authentication," [Online]. http://eugridpma.org/, 2006. [Online]. Available: http://eugridpma.org/

[14] J. Wiebelitz, S. Piger, C. Kunz, and C. Grimm, "Transparent Identity-based Firewall Transition for eScience," in *Proceedings of the 2009 Fifth IEEE International Conference on e-Science (to appear)*. IEEE Computer Society, 2009.

[15] S. L. M. Horowitz, *RFC 2228 - FTP Security Extensions*, The Internet Society Network Working Group Std., October 1997. [Online]. Available: http://www.faqs.org/rfcs/rfc2228.html

Christopher Kunz is a research associate with the Distributed Computing Security Group at L3S research center in Hannover, Germany.

Jan Wiebelitz is a research associate with the Distributed Computing Security Group at L3S research center in Hannover, Germany.

Matthew Smith is an assistant professor and leads the Distributed Computing Security Group at Leibniz Universitaet Hannover, Germany.