SHIELDING THE OASIS RBAC INFRASTRUCTURE FROM CYBER-TERRORISM*

András Belokosztolszki and David Eyers

Abstract OASIS is a distributed RBAC implementation with many extensions. Sound policy design will permit OASIS to protect the distributed resources whose access privileges it controls. However, through operating in a distributed environment, the underlying OASIS infrastructure is open to a number of potential attacks. This paper identifies three main classes of such attack and introduces techniques to extend both OASIS specifically, but also RBAC systems in general, to protect against them.

Keywords: role-based access control, OASIS, security, intrusion detection

1. Introduction

Few would disagree that recent world events have highlighted the risk of globally coordinated terrorist activities. Whilst cyber-terrorism may not lead to the same loss of life, it is a significant threat to critical computer systems - any serious architecture proposing secure access control (and associated administration) must explicitly address it.

Role-Based Access Control (RBAC) [9, 11, 8] has become a popular methodology for controlling the access privileges users are permitted to acquire in a system - section 2 provides a brief introduction to RBAC.

The Opera Research Group at the University of Cambridge Computer Laboratory have developed the OASIS system [2] to extend RBAC to support access control over distributed systems. One proposed application of OASIS is in electronic health record management for the United Kingdom National Health Service (NHS). Given the sensitive nature of this system, we revisit the issue of what attacks a cyber-terrorist might

*This research is supported by King's College Cambridge, the John Stanley Graduate Fund, the Cambridge Australia Trust and the United Kingdom Overseas Research Students Awards Scheme. Thanks also to Dr Ken Moody and the Cambridge Opera Research Group.

apply to OASIS. We examine both *internal* and *external* attacks. By an *internal attack*, we mean an attack made by an authenticated user within the framework of OASIS operations. To address such vulnerabilities requires explicit OASIS policy-design decisions. An *external attack* is one which is targeted at the infrastructure on which the OASIS architecture itself is based. We assume that any committed malicious user will be able to acquire detailed knowledge of the architecture of an OASIS system, since this is published research material. They will also be able to learn about the logical and physical structure of a given deployment, and indeed the likely form of internal policy structure, particularly if they are an *inside agent*. Due to the distributed nature of OASIS, local administrators have minimal ability to ascertain the global state of the system from their local viewpoints. Unlike in single-machine access control, manual intervention is much harder to coordinate.

The aim of this paper is to discuss some current points of failure in the existing OASIS system. Motivated by these faults, we propose both solutions to these issues, and indeed more general extensions to RBAC.

It will not be possible to protect OASIS against all possible attacks - our main focus is protection against the actions of single (or small numbers of) rogue users. To be less conspiratorial, the checks and balances we propose also increase the resilience of OASIS to genuinely unintentional user mistakes and system faults too.

The rest of this paper is organised as follows. Section 2 provides a brief introduction to Role-Based Access Control, before going on to introduce the most important aspects of the OASIS RBAC extensions in section 2.1. Section 3 then discusses some current points of failure in the OASIS system in three particular areas: heartbeat failure, policy design guidelines and policy specification extensions for critical elements, and bounded session lengths. Finally, section 4 draws conclusions relating to the need to strengthen distributed access control system architectures such as OASIS against malicious attack, and reviews the proposals we have made in this regard.

2. Role-Based Access Control

The objective of access control is to protect resources from unauthorised access whilst ensuring access to authorised users. Intensive research into access control began in the early 1960s, as the need to protect that era's databases and operating systems became increasingly critical.

Two major access control methodologies evolved from this research: Mandatory and Discretionary Access Control (MAC and DAC - both models are discussed in [12]). However, there are a number of neces-

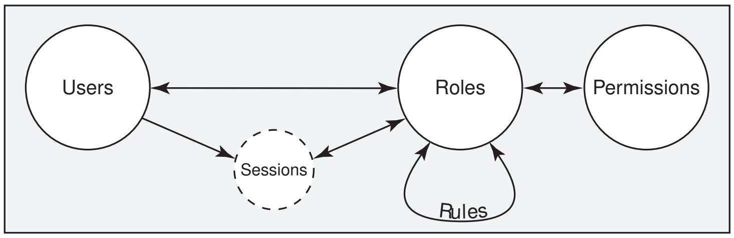

Figure 1. Users, Roles, Sessions, Privileges and Constraints in $RBAC_2$. Note that only one user can hold any given session. All other relationships are many to many.

sary administrative functions which lack convenient support under either scheme. For example, addition and deletion of users and/or protected resources in a given system can sometimes require updating numerous dependent entries. These problems, along with further shortfalls like the rigidness of MAC and the openness of DAC led to the development of the *Role-Based Access Control model (RBAC)*. Significant research was done as early as 1988 in Lochovsky and Woo's proposal of roles [7].

The basic idea of RBAC is to simplify the administration of the effective user to privilege mapping by splitting it into two: a mapping from users to roles, and a mapping from roles to privileges. These mappings can be seen in figure 1. RBAC has become widely accepted because of the flexible security policies it facilitates, such as allowing access control roles to correlate with personnel roles in an organisation. The success of RBAC led to a number of incompatible evolutions of terminology - Sandhu et al. assist the discussion of RBAC through the definition of four particular RBAC reference models [9].

One component of the most basic of Sandu's reference models, $RBAC_0$ is the session. Each session is a mapping from a particular user to a set of active roles. This might indicate the roles a user has activated within a particular login session of a system, for example. The user's more powerful roles may thus remain inactive, even though they are permitted to activate them at will. This supports the *principle of least privilege*; that users be provided with the smallest acceptable set of privileges required to complete their immediate tasks. Note that in this case it leaves compliance to this principle at the discretion of the user.

Of the four RBAC models proposed, $RBAC_2$ is the most closely related to the OASIS system. $RBAC_2$ extends the basic $RBAC_0$ model by adding role-role relationships, as shown in figure 1. Each role-role relationship (labelled as 'Rules' in the above figure) can be thought of as a directed edge between roles, and has an associated constraint which must be satisfied if a user is to activate the target role based on their already being active in the source role. $RBAC_2$ facilitates the deployment of powerful policy schemas, two such examples being *cardinality constraints*, and *separation of duties constraints* [13]. In the former, we restrict the number of users who can be active in a certain role. In the latter we divide roles (and thus privileges) into mutually-exclusive sets.

The next section discusses how the OASIS system extends RBAC to allow the management of distributed services.

2.1 OASIS

The *Open Architecture for Secure Inter-working Services (OASIS)* [2, 5] is an RBAC implementation developed at the University of Cambridge Computer Laboratory. Its first-order logic-based model [14] is based on an earlier capability system which extends RBAC in many ways.

OASIS roles are activated in the context of sessions. After initiating a session, the user will be automatically assigned some intial role. However in contrast to most RBAC implementations, OASIS roles and rules are managed in a decentralised manner. Each of the distributed objects for which OASIS provides access control is wrapped by an OASIS service, which itself may be distributed over OASIS servers. These services all operate in an asynchronous manner, and cooperate with each other by use of a publish/subscribe event platform [1]. In basic terms, this means OASIS services and servers subscribe only to the events relevant to them, these events being published by other services in the network.

Current OASIS implementations maintain reliable and secure system operation through a *heartbeat* mechanism. Cooperating components cyclically attempt message exchanges within bounded time periods, assuring these components the OASIS service network is operating reliably.

Recent research into RBAC increasingly discusses the need of *context-awareness* for roles [4]. OASIS supports context-aware behaviour in two main ways. Firstly OASIS roles may carry *parameters*. Secondly *environmental predicates*, which are also parameterised, may be included in OASIS rules. These predicates provide a mechanism through which OASIS rules may depend on local system factors outside the OASIS environment. These two features together allow highly expressive OASIS rules. Note that for simplicity parameters have been left out of the notation presented below.

OASIS also supports delegation through the more abstract concept of *appointment* [2], wherein an appointer will present a given appointee (or group of appointees) a particular appointment certificate. Unlike roles, appointment certificates are long-lived digitally-signed certificates, which might be appropriate to express, for example, academic qualification or membership of an organisation.

The OASIS model features a number of different types of rules. The assignment of a user to a role, which will be active within a session, is managed by a given *role activation rule*. Similarly, the assignment of a particular privilege to a role is managed by an *authorisation rule*. The structure of a role activation rule is as follows:

$$r_1, r_2, ..., r_{n_r}, ac_1, ..., ac_{n_{ac}}, e_1, ..., e_{n_e} \vdash r$$

The r_i, ac_j and e_k terms represent the n_r prerequisite roles, n_{ac} appointment certificates and n_e environmental constraint predicates in this rule respectively – note that it is acceptable for any of n_r, n_{ac} or n_e to be zero, provided at least one is non-zero. Predicate expressions on the left hand side of the rule are called *preconditions*, and must be valid for a given user to activate r, the *target role*. Roles and appointment certificates are valid if they have not been revoked. Environmental predicates are valid if they evaluate to be true.

Authorisation rules are of the following form:

$$r, e_1, ..., e_{n_e} \vdash p$$

There is one and only one prerequisite role r. The environmental constraints e_k behave as for role activation rules, and finally p is the target privilege of this rule. A set of the above role activation and authorisation rules defines the policy for a given OASIS service.

One of the main strengths of OASIS is its *fast revocation mechanism*. By default, each precondition will be checked for validity only at the time of evaluation of a given rule. However, it is possible to tag any such precondition as a *membership condition*, which means it will be specifically monitored by the OASIS system and must remain valid for the target role to remain active. This is indicated in a rule by tagging the precondition with a superscript '*'. For example the first environmental constraint in the above authorisation rule would become e_1^*.

OASIS services achieve fast revocation by means of so called *credential records*: small structures stored at each OASIS service to indicate their knowledge about the validity of a certain prerequisite. When they believe a prerequisite is invalid, revocation takes place. Due to transitive dependencies, revocation can trigger a cascade of revocations throughout the OASIS network - a vulnerability we discuss in section 3.

3. Proposed solutions to current points of failure

This section identifies a number of potential points of failure in the current OASIS architecture were it to come under extreme forms of attack. We propose techniques to alleviate the risks associated with each type of vulnerability. We also introduce some extensions to the RBAC methodology itself.

3.1 Heartbeat Failure

The first of the major attacks we examine is an external attack on the underlying heartbeat event system on which OASIS is based.

An OASIS service sends at least one message within every heartbeat period. Each of these events contain a sequence number, thus allowing all services to be able to locally detect heartbeat event delays or losses, and for services to handshake with each other. If there is no state change to be transmitted, an empty packet is sent. For performance reasons the heartbeat receiver only sends an acknowledgement once for each cycle of a pre-configured number of heartbeats. In addition the period of the heartbeat can be configured independently for each service, allowing them to set their own trade-off between failure tolerance and security.

As mentioned in section 2.1 , there are two types of preconditions: those which are required only for the initial evaluation of a rule and those which are required to remain active for the duration that the target role or privilege of this rule is active. Most critical appointments and prerequisite roles (the management of which are discussed in section 3.2) are likely to be membership conditions.

In case of interconnected services each local service stores its belief about the state of all the relevant remote services' credential records in local external-credential records. If a heartbeat fails, the relevant external-credential records are annotated with the special tag *unknown*, until the heartbeat resumes. Unknown states may trigger cascading revocation.

A malicious user may be able to make an attack on the network layer of current OASIS implementations to disturb the heartbeat events. By isolating services for a sufficient duration in excess of the normal heartbeat period, critical conditions will sense this loss and may begin cascading revocation, which may cause highly undesirable denial of service. Alternatively, if the system waited until heartbeat was re-established before revoking roles, an attacker can instead specifically disturb the heartbeat, and then abuse the increasingly incorrect local role-state.

Clearly it is important to set the actual heartbeat period carefully. However, we are faced with two irreconcilable goals; the heartbeat period should be short enough to permit true fast-revocation - likely to be at machine-level speeds. On the other hand, for more stable behaviour in the face of a loss of heartbeat, we need a period in the order of manual-intervention speed, which will seriously curtail fast-revocation.

Our compromise is to extend membership conditions to optionally specify their behaviour in the face of heartbeat loss. We suggest the following types of superscripts for rule preconditions:

Time-delayed revocation is represented by the superscript $\tau(t)$ (or `Time(t)` in textual form). The tagged precondition can hold up to t milliseconds after the deadline for the missing heartbeat.

Count-delayed revocation is represented by the superscript $\kappa(c)$ (or `Count(c)` in textual form). The tagged precondition may hold for up to c heartbeat periods after detecting heartbeat loss.

Lazy-revocation will not revoke the role based on loss of heartbeat. It can be represented by either $\tau(\infty)$ or $\kappa(\infty)$. In textual form this is either `Time(inf)` or `Count(inf)`.

Quick-revocation revokes the role if heartbeat loss is detected (based on the expected heartbeat period for each particular service). This is the existing OASIS strategy, and can continue to be indicated with a superscript '*' tag, or either of $\tau(0)$ or $\kappa(0)$.

Clearly in any case the system will need to have a mechanism through which to alert local human administrators to any loss of heartbeat, since this may then require them to activate certain emergency roles in their local system (discussed in section 3.2).

Modifying existing rules to include these new membership condition tags will be significantly aided by analysing the OASIS history logs. Informed decisions can be made based on the statistical distribution of target role validity times (e.g. the frequency of role activation, or the average and the standard deviation of the target role activation time).

If an OASIS system contains a variety of these extended membership condition tags, its overall behaviour in the face of heartbeat loss will become significantly less deterministic under an attack. We suggest that this will make heartbeat attacks less appealing to potential terrorists, since it is highly unlikely they will be able to ascertain details relating to all the time-delay and count-delay parameters on which the success of their attack will depend.

3.2 Threshold-based rule evaluation

Identifying critical OASIS appointments and roles in a given policy store is predominantly a protection against internal attacks. Although it is necessary for the theoretical OASIS model to present all appointment certificates and roles as being similar, there will be significant differences in their respective usage. The design of policy rules should take these differences into account.

As mentioned previously, the policy description language in the OASIS system is very flexible and expressive. Naturally, the OASIS system itself has no understanding of the underlying semantics of the roles and rules within a system. On one hand, this flexibility can ease the creation of new policy, since administrators can express rules in a large number of different ways. On the other hand it can increase the difficulty of policy maintenance - it can be hard to see the consequences of even minor changes, particularly if these policies involve dependencies on remote systems. Checking policies against a set of goals and constraints may be eased by the use of *meta-policies* [3].

As another tool we describe a trivial static-analysis approach to calculating *dependency estimates* to assist policy administrators in identifying which roles and appointments may have dangerous cascading revocation potential, and then propose an extension to OASIS and RBAC rule evaluation which may assist in avoiding this becoming a point of failure. Note that we can only estimate dependency factors via static analysis since not all OASIS preconditions are membership conditions, and because the behaviour of environmental predicates is dynamic and will usually be unknown to the OASIS system.

Let us assume we wish to calculate a dependency estimate for the OASIS appointment certificate a. First we find the set $\mathcal{R} = \{r_1, r_2, ..., r_n\}$ of all rules which include a in their left hand sides. A crude measure of dependency might simply be the cardinality of $\mathcal{R}$. Obviously this will not take transitivity into account.

We propose that instead, the set $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ is formed from all of the corresponding targets of each of the roles in the set $\mathcal{R}$. The dependency estimate of a is simply $de(a) = \sum_{i=1}^{n} de(t_i)$, that is to say merely the sum of the dependency estimates of the target roles based on it. Defined recursively, $de(t_i) = \sum_{j=1}^{m} de(t_j)$, for each role t_j dependent on t_i. The base case of a role t_k on which no other role depends is $t_k = w$ for some weight factor w. We suggest that terms in $\mathcal{R}$ and $\mathcal{T}$ which might cause cycles be filtered from the summation in the preceding equation.

Further improvements of the dependency estimate could be made via use of statistical information about role activation history. For example, non-membership conditions should ideally have a scaled-down significance in the calculation of dependency estimates.

The above estimates will provide an indication of the critical preconditions within a given OASIS domain, and advise administrators to use techniques such as *threshold-based rule evaluation* (introduced below) to protect them, or at least be aware of their existence.

Our proposed *threshold-based rule evaluation* extends the expression of OASIS rules to support multiple-party or weighted voting-based agreements and hand-over arrangements. Similar schemes are proposed in [6, 10]. Under threshold-based evaluation, rules can be expressed as follows:

$$r_1 \cdot x_1, r_2 \cdot x_2, ..., r_{n_r} \cdot x_{n_r}, ac_1 \cdot y_1, ..., ac_{n_{ac}} \cdot y_{n_{ac}}, e_1 \cdot z_1, ..., e_{n_e} \cdot z_{n_e} \vdash_w r$$

where $\sum_{i=1}^{n_r} x_i + \sum_{i=1}^{n_{ac}} y_i + \sum_{i=1}^{n_e} z_i \geq w$ must hold for this rule to evaluate true. If these weight factors are omitted from the rule, implicitly x_i, y_i and z_i values are taken be equal to one, and $w = n_r + n_{ac} + n_e$.

The rule $a \cdot 3, b \cdot 1, c \cdot 1, d \cdot 1 \vdash_5 r$, can represent an example of multiple-party appointment. In this case, let us assume that a is an initial role, and b, c and d all represent appointments regarding the same qualification for this particular user. By setting the threshold weight at 5, we have effectively specified a policy whereby a two-thirds majority of the appointing parties is acceptable. Note that there is an underlying separation of duties constraint here - any of the appointing parties must not be implicitly permitted to act as any of the other appointing parties.

This extension of rule evaluations can also assist with a number of legitimate problems possibly causing appointment revocation, for example appointment hand-over or emergency role activation. Consider the rule $a \cdot 3, b \cdot 1, b_{override} \cdot 1, a_{emergency} \cdot 4 \vdash_5 r$. In the case of appointment hand-over, the party which originally provided appointment b may be phased out of the OASIS system. To make sure we do not get unnecessary revocation, a super-user may enable the $b_{override}$ appointment certificate temporarily until the hand-over target party can provide a new b appointment. As an overall measure, the $a_{emergency}$ role requires only one other precondition to activate role r.

Via multiple-party rules, we can ensure that a single rogue user, particularly an 'inside agent' with knowledge about the structure of role prerequisites in the system, has a much more difficult task mounting an attack on critical OASIS rule preconditions.

3.3 Bounded session durations

This section examines protection against internal attacks from malicious users attempting to bring about denial of service (DoS) attacks on the OASIS service. The two main risks identified here are:

OASIS network overload through bogus state changes. A malicious user can attempt to overload the OASIS network by cyclically activating and deactivating roles in the system in cases where they know some event channels will have subscribed to information about the state changes under their control. Using appointments and revocation certificates might also allow them to increase system load, particularly if the malicious user in question is able to find out about rule-checking critical elements of their domain (as mentioned in section 3.2).

OASIS server memory overload through garbage. Here, a malicious user instead tries to exhaust the storage space of an OASIS service through creation of an excessive number of appointment certificates or active role certificates. A form of this problem may actually arise from the normal operation of an OASIS service. To monitor membership conditions, servers need to store data which should be garbage collected after the relevant session terminates. However, various network failures, irresponsible user behaviour, or complex cross-references between records (as they are distributed over servers) may lead to failures of the garbage collection algorithms normally used to lower the number of stale records.

There is a particular risk of this sort of attack when policy-design does not explicitly limit local users' parameterisation of appointment certificates. Equivalently, a malicious user may initiate numerous new sessions without terminating older ones - if automated, this process would clearly quickly overload the OASIS service.

In both cases we propose modifications to the local OASIS session management. In particular, sessions are all tagged with an explicit lifetime duration attribute, which is propagated to the membership records dependent on this session. The lifetime duration will depend on the OASIS service and the initial roles associated with the session in question. The lifetime granted by a server might change as statistical data is accumulated about the system's roles and their usage.

We suggest that the OASIS services should class users into different types of session depending on the expected level of impact they will have on the OASIS server, and remote OASIS services.

To cater for the case of network overload, the first type of DoS situation presented above, the local OASIS service should apply an activity-based flow-control protocol to each user's local sessions. The parameters of this flow control should be determined when using local identification information to evaluate an OASIS authentication rule (possibly based on their position within an organisation).

The other OASIS services may include a measure of the maximum network usage expected of a particular service, if its local sessions have resource allocation bounds. This means that should an OASIS service be hijacked completely, the other services will be able to sense this out-of-band behaviour, and do their best to shield themselves from that particular service. Naturally they should also signal this abnormal state to a human administrator.

For the case of garbage collecting in the system, we can again use the network-use flow-control proposed above. In particular, we need to have some sort of local estimate as to the global resources being consumed by that particular session. Again bounds could be placed on the permissible estimated global resource usage based on users' session types.

4. Conclusions

Any system intended to provide a distributed security framework must come under close scrutiny not only in terms of the fine-grained privileges in controls within the framework it defines, but also in terms of the vulnerability of the framework itself.

Proposed large-scale sensitive distributed systems such as electronic health record networks in the UK National Health Service may well be targets for cyber-terrorism. With this in mind, we revisit some of the architectural features of the OASIS system, and RBAC in general.

We discuss handling OASIS heartbeat failure by means of extended membership conditions. We also recommend threshold-based rule evaluation as an extension to RBAC. This allows the expression of policies such as multiple-party constraints for the sake of protecting critical elements of a given access control environment. Finally, we discuss a number of reasons why we believe it necessary for OASIS sessions to have bounded maximum durations. These proposals all increase the resilience of OASIS architecture to a premeditated attack.

Future work leading from our research will include integration of our proposed changes into the OASIS formal model, development of specific dependency-estimation algorithms and analysing the computational consequences of our rule-evaluation modifications on the OASIS unification process.

It will probably never be possible to create a completely secure distributed system of any decent functionality and size. However, it would be naive to assume that tomorrow's distributed systems will not at times come under intense pressure from rogue users.

References

[1] Jean Bacon, Ken Moody, John Bates, Richard Hayton, Chaoying Ma, Andrew McNeil, Oliver Seidel, and Mark Spiteri. Generic support for distributed applications. *IEEE Computer*, pages 68–77, March 2000.

[2] Jean Bacon, Ken Moody, and Walt Yao. Access control and trust in the use of widely distributed services. In *Middleware 2001*, volume 2218, pages 300–315, November 2001.

[3] András Belokosztolszki and Ken Moody. Meta-policies for distributed role-based access control systems. In *Policy 2002: IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, pages 106–115, June 2002.

[4] Michael J. Covington, Wende Long, Srividhya Srinivasan, Anind K. Dev, Mustaque Ahamad, and Gregory D. Abowd. Securing context-aware applications using environment roles. In *Sixth ACM Symposium on Access Control Models and Technologies*, pages 10–20, 2001.

[5] John H. Hine, Walt Yao, Jean Bacon, and Ken Moody. An architecture for distributed OASIS services. In *Middleware*, pages 104–120, 2000.

[6] Savith Kandala and Ravi S. Sandhu. Extending the BFA workflow authorization model to express weighted voting. In *IFIP Workshop on Database Security*, pages 145–159, 1999.

[7] Frederick H. Lochowsky and Carson C. Woo. Role-based security in data base management systems. In Landwehr Ed., editor, *Database Security: Status and Prospects*, pages 209–222, Amsterdam, The Netherlands, 1988. North-Holland Publishing Co.

[8] Matunda Nyanchama and Sylvia Osborn. The role graph model and conflict of interest. *ACM Transactions on Information and System Security (TISSEC)*, 2(1):3–33, 1999.

[9] Ravi Sandhu, Edward Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[10] Ravi Sandhu. Transaction control expressions for separation of duties. In *In Proceedings of the Fourth Aerospace Computer Security Applications Conference*, pages 282–286, December 1988.

[11] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: towards a unified standard. In *Proceedings of the fifth ACM workshop on Role-based access control*, pages 47–63, 2000.

[12] Ravi Sandhu and Pierrangela Samarati. Access control: Principles and practice. *IEEE Communications Magazine*, 32(9):40–48, 1994.

[13] Richard T. Simon and Mary Ellen Zurko. Separation of duty in role-based environments. In *PCSFW: Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1997.

[14] Walt Yao, Ken Moody, and Jean Bacon. A model of OASIS role-based access control and its support for active security. In *Sixth ACM Symposium on Access Control Models and Technologies*, pages 171–181, 2001.