CORBASEC Frequently Asked Questions and Answers

Maintained and edited by
Konstantin Beznosov*

June 18, 1999

Abstract

This is a list of Frequently Asked Questions about CORBA Security Service (CORBASEC). It contains both general and technical information about CORBASEC: status, what it is, what it specifies, and more. Please read this FAQ carefully before you post questions about CORBASEC to various CORBA-related mailing lists to see if your question is already answered here first.

The document is available in electronic form at http://cadse.cs.fiu.edu/corba/corbasec/faq/.

*http://www.cs.fiu.edu/ beznosov

Contents

1 Disclaimer and Copyright, Copying, and Other Information

1.1 Disclaimer

EXCEPT WHEN OTHERWISE STATED IN WRITING, THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DOCUMENT AND INFORMATION CONTAINED IN IT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE INFORMATION IN THE DOCUMENT IS WITH YOU. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE DOCUMENT AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION CONTAINED IN THE DOCUMENT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.2 Copyright Policy

This document is copyrighted by its all contributors.[1] and its maintainer[2] . This document may be distributed under the terms of General Public License.[3]

1.3 $\surd$ Who sponsors the FAQ?

The document is sponsored by the Center for Advanced Distributed System Engineering[4] at Florida International University[5] and by Baptist Health Systems of South Florida[6].

2 About this FAQ

2.1 Why is this FAQ created?

CORBA Security service specification is comparatively long (more than 300 pages) and complex. It is difficult to understand every point of the specification, and it is even more difficult to grasp the "big" picture of how everything is supposed to "work." On the other hand, many different people (ORB developers, application developers, system and enterprise administrators, security guys, etc.) with different technical backgrounds want to understand

[1] See question 2.14, "Who contributed to this FAQ?", on page 12.
[2] See question 2.8, "Who is maintaining this FAQ?", on page 11.
[3] http://www.fsf.org
[4] http://cadse.cs.fiu.edu
[5] http://www.fiu.edu
[6] http://www.baptisthealth.net

different aspects of the specification and its implementations. Even more, sometimes the intent of the authors, expressed in the particular wording, is interpreted differently by different readers, which causes problems not only to those readers but also to others. Also, it was observed that when various working groups and teams begin discussions on CORBA security, they spend a substantial amount of effort and time just getting to the common ground and agreeing on a common understanding of security aspects of CORBA technology. This FAQ was created to help all those who need to understand CORBA Security service specification and to provide readers with additional information that could not be included in the specification but is considered useful to eliminate problems described above.

2.2 Notations

Answers that are new or are updated since the last month revision, are marked with the sign "$\surd$". Questions that do not have answers yet are marked with the sign "$\triangle$".

Changes in the text made since the last month revision are marked with change bars like the one on the left margin of this paragraph.

2.3 Who should read it?

Those who do or will use CORBA Security service specification or its implementation, or who is curious about security aspects of CORBA technology.

2.4 How should I read this FAQ?

Find a question which is similar to yours. Read the answer to it.

If you want to grasp the context of the thread from which the answer was originated, and if the answer has an e-mail message ID of the original message, then use the ID to track down the thread in various archives and news groups. Read the thread to get better understanding and all details of the answer.

2.5 Where can I get the most recent copy of this FAQ?

The most recent copy is always available on its site – http://cadse.cs.fiu.edu/corba/corbasec/faq/.

2.6 $\surd$ I could not find an answer to my question here, where else can I ask my question?

Send your questions to the mailing list on CORBA security (question 2.7, "Are there any newsgroups or mailing lists about CORBASEC that I can join?", on page 11).

2.7 Are there any newsgroups or mailing lists about CORBASEC that I can join?

Konstantin Beznosov (November, 1998)[7] : This is to announce availability of a new mail list <corba-security@cs.fiu.edu>.

The intent of the list is to provide forum for discussing issues related to security in CORBA-based systems and applications.

Information on how to subscribe to the mail list is available at
http://www.cs.fiu.edu/~beznosov/corba/security/mail-list/

It's a majordomo-based mail list. Usual majordomo commands can be used with the list.

2.8 Who is maintaining this FAQ?

Konstantin Beznosov[8].

2.9 $\surd$ What is SecSIG?

Konstantin Beznosov (May, 1999):

SecSIG is OMG's special interest group on security. You can visit SecSIG's home page[9] or send an e-mail message to the group mailing list at ¡secsig@omg.org¿. Since only employees of the OMG member companies can be on the mail list of the SecSIG, its mail list is used more for coordination of the group work.

2.10 What is secsig@omg.org mail list for?

Konstantin Beznosov (May, 1999):

<secsig@omg.org> is not the list about CORBA security. It's the list for facilitating communications among members of the OMG security special interest group (SecSIG). There is no any special mailing list for discussion of CORBASEC. If you have questions about security in CORBA-based systems, you might want to send them however to <corba-security@cs.fiu.edu> mailing list (see question 2.7). If your company is a member of the OMG, you can also get subscribed to the secsig@omg.org list in case you are interested in participating in SecSIG.

2.11 What is SecRTF?

SecRTF is OMG's revision task force created exclusively for revising CORBASEC.

[7] Message-Id: Pine.GSO.4.05.9811171310320.10792-100000@grads.cs.fiu.edu
[8] http://www.cs.fiu.edu/~beznosov
[9] http://www.omg.org/homepages/secsig/

2.12 I found a mistake in this FAQ, where can I submit it?

The best way is to send an e-mail message to its maintainer (question 2.8, ”Who is maintaining this FAQ?”, on page 11).

2.13 How can I contribute to this FAQ?

The best way is to send an e-mail message with contribution to its maintainer (question 2.8, ”Who is maintaining this FAQ?”, on page 11).

Your contribution to this list is subject to its copyright policy described in section1.2, ”Copyright Notice”, on page 9.

2.14 $\surd$ Who contributed to this FAQ?

The following is the list of people who contributed to this Q&A list (in alphabetical order):

- Habtamu Abie <Habtamu.Abie@kjeller.fou.telenor.no> : 5.1.1
- Shahzad Aslam-Mir <sam@expersoft.com>: 5.1.1
- Nick Battle <nick.battle@x400.icl.co.uk> : 4.1.23, 4.1.31, 4.1.26, 6.1, 6.1, 3.11
- Jim Beale <eghx@gdeb.com> : 5.1.1
- Konstantin Beznosov <beznosov@baptisthealth.net>: 3.10, 2.9, 2.10, 2.11, 7.1.5, 4.1.3, 4.1.4, 3.6, 3.11, 2.7, 7.5.3, 4.1.2
- Jonathan Biggar <jon@floorboard.com>: 4.1.22, 4.1.23, 4.1.26, 4.1.27, 4.1.28, 4.1.29, 4.3.3
- Bob Blakley <blakley@dascom.com> : 4.3.1, 4.3.5, 4.3.6, 4.3.7, 4.3.8, 4.3.9, 4.4.7, 4.4.8, 4.4.9
- Gerald Brose <brose@inf.fu-berlin.de>: 4.3.2
- Bob Burt <bburt@2ab.com>: 4.3.2
- Ludwig Brinckmann <LBrinckman@datastreamicv.com>
- Jeff Calog <j649281@marlin.bdy.wi.ameritech.com> : 7.5.1
- David Chizmadia <david_chizmadia@omg.org> : 4.2.8, 4.1.32
- Tom Damiano <tdamiano@crusher.jcals.csc.com>: 7.5.4
- William Edwards <wedwards@corp.inprise.com>: 7.5.4
- Luis Espinal <lespin03@cs.fiu.edu>: 3.10
- Belinda Fairthorne <belinda.fairthorne@x400.icl.co.uk>: 4.4.4

- Ted Gamester <gamester@umich.edu>: 7.5.7, 7.5.8
- Rohit Garg <rohit@objectScape.com>: 5.1.1
- Jose Ignacio Gijon <nacho@eunet.es>: 7.5.4
- Linda Gricius <lgricius@iclinc.com>: 7.1.5, 4.1.18, 4.1.19, and all subsections in 5.2.1.
- Rajan Gupta <listing@rocketmail.com>: 4.1.14, 7.5.4
- Tom Herron <therron@erols.com> : 5.1.1
- Polar Humenn <polar@adiron.com> : 5.1.1, 7.5.13, 4.3.6, 4.3.9
- Dan Hushon <dan.hushon@East.Sun.COM> : 5.2.2.4
- Bill Janssen <janssen@parc.xerox.com> : 5.1.1
- Marc Laukien <ml@ooc.com> : 7.5.4
- B.G. Mahesh <mahesh@paragon-software.com> : 3.11
- Mike <mcoram@interzone.com> : 7.5.15
- Jishnu Mukerji <jis@fpk.hp.com>: 4.1.2
- Dale Nagata <dnagata@creo.com>
- Olivier Onimus <Onimus@danet.de> : 4.1.23
- Steve Parker <sparker@visa.com> : 3.10
- Michelle L. Patterson <mlp@epoch.ncsc.mil>
- Dave Sames <dsames@tis.com> : 7.5.13
- Rudolf Schreiner <ras@muc.de>: 3.11, 5.2.2.5
- John Sebes <ejs@tis.com>: 4.2.2
- Chris Shutters <chris_shutters@omg.org> : 3.8
- Andre Srinivasan <andre@inprise.com>: 7.5.6, 4.2.5, 5.1.1
- Gregg Tally <tally@home.com> : 5.2.2.8
- Serban Tatu <statu@starvision.com> : 5.1.4
- Bruno Traverson <Bruno.Traverson@der.edfgdf.fr>: 4.1.18, 5.2.1.3, 5.2.2.3
- Roland Turner <raz@arrakis.com.au> : 4.2.4
- Andreas Vogel <avogel@visigenic.com>: 4.1.14, 7.5.2, 4.2.3, 7.5.10, 7.5.11, 7.5.9
- Charles White <chas@blackwhite.com> : 3.8, 5.1.4
- Jim Williams <jgw@mitre.org>: 3.11
- George Wolke <george_wolke@customer-insight.com> : 5.2.2.6

2.15 Wish list

The following is the list of wish items maintained by the maintainer. These are various wishes about this FAQ from different people. The list is used by the maintainer to improve the FAQ usefullness. Anyone who thinks they have a great idea about how to make the FAQ more helpfull to the Internet community, are welcome to send suggestions to the FAQ maintainer.

- Make it possible to find full thread from which the information for the answer was used.
- Have a scope section that would describe what goes into the FAQ and what does not.
- Clean and possibly merge sections 4.1.18 and 4.1.19.

3 General Questions

3.1 What's "CORBA Security Service"?

"CORBA Security Service" is one of CORBA core services specified in CORBA 2.x and higher. Specification of it defines security functionality interfaces available in a CORBA ORB.

3.2 What's "CORBASEC"?

Konstantin Beznosov (December, 1997):

"CORBASEC" is a shorter and more informal name for "CORBA Security Service."

3.3 What's the difference between "CORBA Security Service" and "CORBASEC"?

Konstantin Beznosov (December, 1997):

There is no difference between "CORBA Security Service" and "CORBASEC."

3.4 What's CORBA?

Konstantin Beznosov (December, 1997):

CORBA stands for **C**ommon **O**bject **R**equest **B**roker **A**rchitecture. If you are reading these lines, you probably know something about CORBA. If not, read question 3.5, "Where can I find more about CORBA?", on page 15.

3.5 Where can I find more about CORBA?

Konstantin Beznosov (December, 1997):

The most comprehensive information about CORBA technology is available at the OMG[10] web site.

3.6 Where can I find more about CORBASEC?

Konstantin Beznosov (December, 1997): The definitive guide to CORBASEC is of course the CORBASEC specification itself. Also, see read several following questions in this FAQ.

3.7 What books describe or review CORBASEC and in what detail?

3.7.1 "Instant CORBA"

Konstantin Beznosov (December, 1997):

Chapter 11 (28 pages) presents informal description of CORBASEC. The authors walk through key CORBASEC features (according to the book: authentication, privilege delegation, authorization, audit trail, non-repudiation, non-tampering and encryption, security domains, security policies management) and summarizes CORBASEC interfaces. It also walks through several scenarios of exercising security by security-aware applications. The level of the presentation does not assume any background in security and it requires nominal knowledge of the CORBA technology.

Book Information

Title	"Instant CORBA"
Authors	Robert Orfali, Dan Harkey, Jeri Edwards
Publisher	John Wiley & Sons, Inc.
Year:	1997
ISBN	0-471-18333-4

3.8 △ What magazine publications describe, review, compare, critique CORBASEC?

Charles White (October, 1998)[11] if you go to our web page at: http://www.blackwhite.com/press/articles.html

There is a brief description of a paper which appeared in Distributed Computing Magazine recently called : "Licensing and Metering" by Julia Miller and Alex Kotopoulis.

[10] http://www.omg.org

[11] Message-Id: 199810161732.KAA28995@sparky.qds.com

It covers several aspects of Security and may talk a little bit about our product Object/LM. If not there is info about that on our web pages. The full text of the article is available at: http://www.distributedcomputing.com.

Chris Shutters (May, 1999) [12] : The first of a series of articles by David Chizmadia on CORBA Security is online at http://www.sdmagazine.com/supplement/ss/features/s996f1.shtml

It is titled: "CORBAsec: Securing Distributed Systems: Find out what CORBAsec has to offer for security-aware applications."

3.9 △ Are there any papers or articles that compare CORBA security with security of other distributed object computing frameworks such as Java RMI or DCOM?

3.10 √ Where can I get a tutorial about CORBASEC?

Konstantin Beznosov (July, 1998):

There was tutorial on CORBA Security service during the first day of the Second Workshop on Distributed Object Computing Security[13] given by Bret Hartman (Concept Five Technologies, Inc.).

Steve Parker (January, 1999) [14]:

There is a downloadable presentation in PDF format on "Security Concepts for Distributed Component Systems" By Walt Smith of Tekna, which contains an overview of the CORBASEC architecture.

This was a presentation to the 21st National Information Systems Security Conference

It is available at: http://csrc.nist.gov/nissc/1998/proceedings/tutorB2.pdf.

Luis Espinal (May, 1999) [15]: "A Quick Tour Of the CORBA Security Service" is available at http://www.omg.org/news/corbasec.htm, by David Chizmadia, reprinted from "Information Security Bulletin", September 1998.

3.11 What papers about CORBASEC and related issues are out there?

Jim Williams (June, 1998): The document "CORBA Threat-Mitigation Model" was presented to the Security SIG at the June '96 meeting in Washington DC. It is available as document 98-06-01 at http://www.omg.org/docs/security/

Rudolf Schreiner (September, 1998)[16] : A good paper on CORBASEC is Ulrich Lang's M.Sc. thesis: http://www.cl.cam.ac.uk/ ul201/mscdissertation.pdf

[12] Message-Id: 7gur6q6ud1@nnrp1.deja.com
[13] http://www.omg.org/docsec/1998
[14] Message-Id: 95288CC7E7F7D111883B0001FAF85C03017AFBBB@sw720x014.visa.com
[15] Message-Id: Pine.GSO.4.05.9905191055520.24395-100000@bach.cs.fiu.edu
[16] Message-Id: Pine.BSF.3.91.980916161723.12740I-100000@phobos.muc.de

B.G. Mahesh (October, 1998)[17] :

- Charles Cavanaugh. *CORBA Security White Paper*. Advance Software, Switzerland (www.corba.ch)
- Kyeongbeom Kim, Youngkyun Kim, Youngkee Song, Soran Ine. *A Software Platform for Secure Applications based on CORBA*. IEEE, 1997.
- Susan L. Chapin, William R. Herndon, LouAnna Notargiacomo. *Security For The Common Object Request Broker Architecture (CORBA)*. IEEE, 1994.
- Robert H. Deng, Shailendra K. Bhonsle, Weiguo Wang, Aurel A. Lazar. *Integrating Security in CORBA Based Object Architectures*. IEEE, 1995.

Konstantin Beznosov (November, 1998): There is a technical report from IBM that describes a mathematical model of CORBA access control mechanisms and also shows how Mandatory Access Control (MAC) policy could be realized using CORBA Security. The paper reference is the following:

G'unter Karjoth. *Analysis of Authorization in CORBA Security*. Technical report, IBM Research Division, Zurich Research Laboratory, December 1996.

Also, OMG Security Working Group released "OMG White Paper on Security" in 1994, where they described issues with security design in distributed object systems and outlinened CORBASEC model. The paper available at the OMG site as document 94-04-16.

Nick Battle (September, 1998)[18] : There is a white paper at http://www.peerlogic.com/sbs/dais/security.pdf that gives an overview of CORBA Security.

3.12 △ Is there any group or lab research on CORBA security service?

4 CORBASEC specification

4.1 General

4.1.1 Where can I get the official specification of CORBASEC?

Konstantin Beznosov (May, 1999): The OMG Board of Directors adopted CORBA Security Service specification v1.2 on November 10, 1998. It is available as the OMG formal document `formal/98-12-17` in pdf and ps formats. Also modifications to CORBA Core interfaces related to CORBASEC v1.2 can be found as ptc/98-01-04 in pdf and ps formats. Read question 4.1.2 for information on the IDL code for the interfaces.

[17] Message-Id: 199810191603.MAA03007@grub.paragon-software.com
[18] Message-Id: 35FF72E3.2BB81E28@x400.icl.co.uk

4.1.2 Where can I get IDL code of CORBASEC interfaces?

Jishnu Mukerji (August, 1998) : For all interested in a compilable set of Security 1.2 IDL files, they have now been placed in ZIP form on the OMG server as document number ptc/98-08-02.

Konstantin Beznosov (May, 1999): The OMG has now a page with the references to all IDL files from official specifications of CORBA Services, including CORBASEC. Below is the information taken from the site:

DCE_CIOPSecurity.idl formal/98-10-36

NRService.idl formal/98-10-37

SECIOP.idl formal/98-10-38

Security.idl formal/98-10-39

SecurityAdmin.idl formal/98-10-40

SecurityLevel1.idl formal/98-10-41

SecurityLevel2.idl formal/98-10-42

SecurityReplaceable.idl formal/98-10-43

SSLIOP.idl formal/98-10-44

4.1.3 What is the current version of CORBASEC official specification?

Konstantin Beznosov (November, 1998): 1.2

4.1.4 Are there any upcoming updates of the specification?

Konstantin Beznosov (May, 1999): Yes. The text of the new revision 1.5 is available at http://www.omg.org/docs/ptc/98-12-03.pdf.

4.1.5 Who is responsible for producing specification updates?

Konstantin Beznosov (March, 1998): SecRTF (see question 2.11, ”What is SecRTF?”, on page 11).

4.1.6 Are there any upcoming new releases of the specifications?

Konstantin Beznosov (March, 1998): Not yet.

4.1.7 Who is responsible for producing new specification releases?

Konstantin Beznosov (March, 1998): The OMG Technical Committee.

4.1.8 Where can I find a list of outstanding issues in CORBASEC specification?

Konstantin Beznosov (March, 1998): Look at http://www.omg.org/issues/sec-rev.html.

4.1.9 I found a typo in the specification, where can I submit it?

Konstantin Beznosov (March, 1998): First, you want to make sure that the typo or the problem you found is not discovered by somebody else before. For it, you want to check the list of all outstanding issues on CORBASEC specification. (read question 4.1.8, "Where can I find a list of outstanding issues in CORBASEC specification?", on page 19. The best way to submit a report on error or typo in CORBASEC specification is to send an e-mail message to SecRTF reporting it (question 2.11, "What is SecRTF?", on page 11).

4.1.10 △ Is there a set of UML diagrams for the CORBASEC Specification?

4.1.11 I found an error in the specification, where can I submit it?

Konstantin Beznosov (March, 1998): See question 4.1.9, "I found a typo in the specification, where can I submit it?", on page 19.

4.1.12 I have an idea how to "improve" the specification, where can I propose it?

Konstantin Beznosov (March, 1998): First, report the idea to SecRTF. They will consider the proposal and decide if it is valid and if it is in the scope of the revision TF or it goes beyond it. If it is in the scope of the RTF, then there will be a separate issue number assigned so that it can be tracked and eventually resolved. If the proposed improvement is beyond the RTF scope, your idea most probably should go trough the standard technology adoption process (i.e. with issuing an request for proposals (RFP)) in the OMG. Please refer to OMGweb site to find more information about how CORBA technologies are adopted.

4.1.13 △ What are the shortcomings of CORBA Security service?

4.1.14 Is it completely true that the CORBA Security service is a direct lift of DCE Security?

Rajan Gupta (February, 1998): It is correct. In summary, CORBA security is Object Based Security Model while DCE security is more procedural.

Andreas Vogel (February, 1998): Essentially, the CORBA security service defines a framework and interfaces in which you can plug in security mechanisms of your choice. DCE has defined a set of mechanisms (one for each security concern) upfront.

4.1.15 What is ”Principal”, and what is meant by ”Principal authentication”?

Linda Gricius (March, 1998):

Principal authentication is the process of proving your identity to the security enforcing components of the system so that they can grant access to information and services based on who you are. This applies to both human users of the system as well as to applications.

A user or application that can authenticate itself is known as a principal. A principal has a name that uniquely identifies it.

For human users, the process of authenticating to the system is informally known as ”logging on”. In a typical system, an application is provided to collect information proving the user’s identity. This application is often referred to as the ”user sponsor”. In order to successfully authenticate to the system, it is important that a principal can provide some proof that it is who it claims to be. Proof of authentication is usually achieved by demonstrating knowledge or possession of a ”secret” known only to the ”real principal”, such as a password or cryptographic key.

It is important that a successfully authenticated principal can be given some unforgeable evidence that it has recently authenticated, in order to prevent the principal from having to continually re-authenticate itself to different parts of the system. The unforgeable evidence that is returned to authenticated principals is known as the principal’s credentials.

4.1.16 What are credentials?

Linda Gricius (March, 1998):

Credentials contain the security attributes of the principal, a ”lifetime”, and a few other fields. Credentials are used:

- as a means of making the principal accountable for its actions
- as a means of obtaining access to protected objects
- as a means of identifying the originator of a message.

Security attributes include both authenticated attributes and unauthenticated attributes.

Authenticated attributes include identity attributes, which identify the principal, and privilege attributes, which grant rights to the principal.

4.1.17 How are attributes used?

Linda Gricius (March, 1998):

Attributes are type/value pairs that are associated with authenticated users, and held in their credentials. There are two types of attribute: identity attributes (known as identities) and privilege attributes (known as privileges).

An identity attribute has a value which identifies the principal. Examples of identity attributes that a principal might carry are *AuditID* (which is the label recorded in audit records

relating to this principal, and which may be different from *AccessID*) and *AccountingID*, which is the number to be used when charging the principal for resources used.

A privilege is a right granted to a principal that enables them to perform some action that would otherwise be denied. Examples of privileges are *AccessID* (which is the name that they authenticated as) and *Clearance* level.

4.1.18 What does it mean to be conformant to CORBA Security specification?

Linda Gricius (March, 1998): Main security functionality. There are two possible levels:

Level 1 - provides a first level of security for applications which are unaware of security and for those having limited requirements to enforce their own security in terms of access controls and auditing.

Level 2 - provides more security facilities and allows applications to control the security provided at object invocation. It also includes administration of security policy, allowing applications administering policy to be portable.

Security functionality options

These are functions expected to be required in several ORBs, so are worth including in this specification, but are not generally required enough to form part of one of the main security functionality levels specified above. At present, there is only one such option in the specification, non-repudiation.

Security Replaceability

This specifies if and how the ORB fits with different Security services. There are two possibilities:

ORB Services replaceability. The ORB uses interceptor interfaces to call on object services, including the security ones. It must use the specified interceptor interfaces and call the interceptors in the specified order. An ORB conforming to this does not include any significant security-specific code, as that is in the interceptors.

Security Service replaceability. The ORB may or may not use interceptors, but all calls on Security services are made via the replaceability interfaces specified in Section 15.7, Implementor's Security Interfaces of the CORBASEC specification. There interfaces are positioned so that the Security services do not need to understand how the ORB works (for example, how the required policy objects are located), so they can be replaced independently of that knowledge.

If the ORB does not conform to one of these replaceability options, the standard security policies defined in the CORBASEC specification cannot be replaced by others, nor can the implementation of the Security services. For example, it would not be possible to replace the standard access policy by a label-based policy if one of the replaceability options is not supported. Note that some replaceability of the security mechanism used for security associations may still be provided if the implementation uses some standard generic interface for Security services, such as the Generic Security Service API (GSS-API).

Secure interoperability

Secure interoperability – Standard. An ORB conforming to standard secure interoperability can generate and use security information in the Interoperable Object Reference (IOR) and can send and receive secure requests to/from other ORBS using the General Inter-ORB Protocol/ Inter-ORB Interoperability Protocol (GIOP/IIOP) protocol, with the Secure Inter-ORB Protocol (SECIOP) enhancements defined in Section 15.8, Security and Interoperability, of the CORBASEC specification, if they both use the same underlying security technology.

Standard plus DCE-CIOP – Option. An ORB conforming to standard plus DCE-CIOP secure interoperability supports all functionality required by standard secure interoperability, and also provides secure interoperability (using the DCE Security services) between ORBs using the DCE-CIOP protocol.

If the ORB does not conform to one of these, it does not use the GIOP security enhancements, so will interoperate securely only in an environment-specific way.

Common Secure Interoperability (CSI) - confined to secure interoperability of object requests and replies via the GIOP/IIOP protocol.

Bruno Traverson (September, 1998)[19]: Clause C.2 [ed: of the Appendix C in the CORBASEC sepcification v1.2] makes clear that conformance can be claimed at two levels:

1. CORBA Security Functionality that contains three folders:
   - main functionality (level 1 or 2),
   - functionality options (non repudiation),
   - security replaceability (ORB services, Security services, Security Ready).
2. CORBA Secure Interoperability :
   - IIOP-SECIOP (SPKM, Kerberos, CSI-ECMA),
   - Interop options (IIOP-SSL, DCE-CIOP),
   - CSI Level 1,
   - GSS Kerberos Protocol using MD5 Cryptographic profile.

4.1.19 What about conformance to the Common Secure Interoperability specification?

Linda Gricius (March, 1998):

The CSI specification is part of the overall CORBASEC specification.

The Common Secure Interoperability specification defines the standards for common secure interoperability when using GIOP/IIOP, by defining:

- standard security mechanisms and associated cryptographic algorithms

[19] Message-Id: 3606293F.909@der.edfgdf.fr

- details of the SECIOP protocol messages and IOR security tags when using these mechanisms and algorithms
- the security functionality supported when interoperating using these security mechanisms.

It also defines what is required to conform to the mandatory and optional parts of the specification.

An ORB conforming to CSI level 2 can support all the security functionality described in the CORBA Security specification. Facilities are more restricted at levels 0 and 1. The three levels are:

CSI level 0 Identity based policies without delegation – at this level, only the identity (no other attributes) of the initiating principal is transmitted from the client to the target, and this cannot be delegated to further objects).

CSI level 1 Identity based policies with unrestricted delegation – at this level, only the identity (no other attributes) of the initiating principal is transmitted from the client to the target. The identity can be delegated to other objects on further object invocations, and there are no restrictions on its delegation, so intermediate objects can impersonate the user.

CSI level 2 Identity and privilege based policies with controlled delegation – at this level, attributes of initiating principals passed from client to target can include separate access and audit identities and range of privileges, such as roles and groups. Delegation of these attributes can be controlled so that they can only be used at certain locations.

4.1.20 What are the protocols used by CSI?

Linda Gricius (March, 1998):

CSI Common Security Protocols define the details of the tokens in the SECIOP messages. Three protocols are defined:

SPKM Protocol - this protocol supports identity based policies without delegation (CSI level 0) using public key technology for keys assigned to both principals and trusted authorities. The SPKM protocol is based on the definition in The Simple Public-Key GSS-API Mechanism, Internet Draft draft-ietf-cat-spkmgss-06.txt January 1996.

GSS Kerberos Protocol - this protocol supports identity based policies with unrestricted delegation (CSI level 1) using secret key technology for keys assigned to both principals and trusted authorities. It is possible to use it without delegation (so providing CSI level 0).

The GSS protocol is based on the IETF GSS Kerberos V5 definition, which specifies details of the use of Kerberos V5 with GSS-API. It includes updates to RFC 1510; e.g., how to carry delegation information. It is specified in RFC 1964. This itself is a profile of the Kerberos V5 mechanism as defined in IETF RFC 1510, September 1993.

CSI-ECMA Protocol - this protocol supports identify and privilege based policies with controlled delegation (CSI level 2). It can be used with identity, but no other privileges, and without delegation restrictions if the administrator permits this (CSI level 1), and can be used without delegation (CSI level 0).

4.1.21 △ What about CSI with SSL?

4.1.22 What is a "Session"?

Jonathan Biggar (July, 1998) : A session probably maps pretty much to the same thing as the duration of a security context. SecIOP, the corbasec standard security protocol can handle more than one context in parallel and sequentially on a single TCP connection.

Nick Battle (July, 1998) : A good question. I think what you may have in mind is more properly called an "association". An association is a state that exists between peers who have authenticated (possibly mutually) and established random cryptographic keys for the protection of messages between them. An association has a lifetime, defined by policy.

The SECIOP protocol establishes an association between the peers and then protects messages between them. The SECIOP protocol is security mechanism independent, but carries more primitive messages for mechanisms such as Kerberos that "do the hard work".

4.1.23 How does security context get established between client and server?

Olivier Onimus (July, 1998) : This is done by sending client's credentials. The server can authenticate the client and get the session key with which the communication will be encrypted. Then the communication will be encrypted, using this context. The credentials are not sent anymore, only a reference on an existing context.

Jonathan Biggar (July, 1998) : [in addition to the above] The encryption is optional, depending on the Quality of Protection (QoP) you have chosen.

Nick Battle (July, 1998) : [in addition to the above] The Credentials object (capital C) isn't naively transported to the target in an object-relocation sense, though part of the originator's Credentials are re-instantiated at the target (where they're known as received credentials) so that they can be queried for such things as the user's credential attributes.

It's important to realise that these received credentials are not identical to the originator's since (for example) they may not necessarily be used for making further on going associations (called delegation), and they certainly won't allow the target to set_attributes (eg. change the current active role of the client). Perhaps "reconstructed" is a better way to view it.

The session key doesn't actually come from the initiator's Credentials, but rather from a complex (and mechanism specific) protocol message that is separate from the Credentials data - look at the SECIOP protocol EstablishContext message and the specific message contexts for mechanisms such as Kerberos and CSI-ECMA.

[ed: The communication is encrypted] if that's what policy requires. The message may just be integrity protected, or it may require no protection at all. There is also replay protection which is context based.

Subsequent SECIOP messages are all *MessageInContext*, quoting the context reference that was created when the target was first contacted.

4.1.24 Is there somewhere a description of the context management?

Nick Battle (July, 1998) : I don't think this is very well specified. The reason is probably that a lot of the specifics of context (association) management depend on the features that a particular security mechanism supports.

4.1.25 △ What is the validity of a context?

4.1.26 Does a new context for a target have be established if a client is accessing a new target on the same server?

Jonathan Biggar (July, 1998) : Yes, the client will establish a second security context for the new target.

Nick Battle (July, 1998) : First we should clarify a few terms. A host is a machine which runs processes, each process possibly under a different user-id. CORBA objects run in processes; several different objects can run in the same process.

Credentials are not "for" a host, a process or an object. They only describe the user (the principal in fact); they can be expressed at any host/process/object.

The granularity of how contexts are created between clients and targets depends on the security characteristics of the underlying operating system. In UNIX/NT type systems the context is established between the client's process and the target's process, rather than between hosts or objects. This is because the context contains secret keys and all objects (ie. application code) in a process have equal access to these keys, so there is no extra security gained by forming multiple contexts for objects in the same process - though it would do no harm. Processes are partitioned from each other by the OS, so there is some protection to be gained by forming different contexts for objects running in different processes. Similarly for different hosts.

If you wanted to share contexts more widely than between processes, you would have to jump through hoops to make the OS do that in a secure manner. Processes are the natural level of protection granularity - though some operating systems make take this to the thread level.

[ed: About necessity of establishing a new context:] If the 2nd object is co-located with the first in the same process, then the context can be re-used. If the objects are running in different processes, a new context must be established because the 2nd target has no idea what the session key(s) are that were established between the client and target A – it also did not directly authenticate the client itself, so it would have to trust the other process before it shared the context. That might not be a trust we can assume.

4.1.27 Will the current context be valid for all requests of the client (and all replies of the server) till the client decides that the context is not valid anymore?

Jonathan Biggar (July, 1998) : Not necessarily. I believe that a server is allowed to signal to the client that a context is no longer valid, which means the client must reestablish a new context. Also, it is possible for a single server to manage objects in different security domains that may require more than one context be established to the server from a single client.

Nick Battle (July, 1998) : The context is certainly long lived (more than one invocation) - this is a matter of efficiency, since session key establishment is typically very expensive. The context can be destroyed by either peer (eg. by deleting the object or its object reference) or it can timeout. The lifetime is a matter of policy (and cannot be increased by the peers). Contexts may also be destroyed as part of handling a SECIOP protocol error.

4.1.28 Which instance manages the contexts?

Jonathan Biggar (July, 1998) : Both sides. The client initially requests the setup of a context, but I believe that either side can invalidate the context.

Nick Battle (July, 1998) : The ”interceptors” are responsible for managing the contexts once they have been created. The SecurityContext object has operations to protect messages in the context, and could in practice do some of the management of the context when messages are sent/received (ie. it will notice that the context has expired when it tried to encrypt a message, and can arrange for the interceptors to try to reform a new context first).

4.1.29 Which instance decides that now, the ”Session” is over, and the context can be deleted?

Jonathan Biggar (July, 1998) : See question 4.1.28, ”Which instance manages the contexts?”, on page 26.

4.1.30 △ Are the any interfaces specified in CORBASEC for controlling security context by security-aware applications?

Extended Question: Examples of controlling security context could be the following:

- Switch context, a possibility of ”switching context service” between already established contexts
- Hold-on context in case of mobility
- Close/take-down context
- Re-establish context after hold-on context service
- Refresh context in case of revocation or as the same operation as *Refresh()* in *SecurityContext*

4.1.31 How is access controlled?

Nick Battle (July, 1998) : Access is controlled using the initiator's authenticated attributes which are retrieved from the received credentials; audit is similarly able to record the user's access quoting their authenticated attributes.

4.1.32 How are privacy and non-repudiation addressed by CORBASEC?

David Chizmadia (September, 1998)[20] : CORBASEC currently includes an optional Non-Repudiation Security service. The SecSIG currently expects to issue an RFP (Request for Proposal) for a Data Protection service that would allow for applications to request that a block of data be "wrapped" to protect both its integrity and confidentiality. In both cases, the services are easily implemented using (IDUP[21])-GSS as their foundation. This is in addition to the message-level privacy and integrity services already provided as part of the current CORBASEC.

4.2 Application developer

4.2.1 How does CORBA security affect application writers?

Linda Gricius (March, 1998):

In a secure CORBA system, the same client calls the same target object that it would call in an unsecured system. The invocation request is intercepted by the ORB Security service at both client and target, and the level of protection required by the current policy settings is applied. Security may be enforced at the client side, the target side, or both. This includes support for any or all of the following:

- Establishing secure associations between the client and target;
- Integrity and/or confidentiality protection for individual requests and replies sent between the client and target;
- Access control checks to determine if the principal is allowed to perform this operation on the target object;
- Auditing of security-relevant events.

Since the invocation is intercepted in the ORB, it is transparent to the application. However, applications that want to enforce their own security controls can call on the Security services directly.

It's important to note that object implementations do not need to be changed to fit into and be protected by a secure ORB. A distributed application may be made up of many small objects, and it is unusual for all the application developers to be sufficiently security knowledgeable to make the right calls on the security facilities.

[20] Message-Id: 01bdeb1888404de01b033390@moccasin.tycho.ncsc.mil
[21] [ed: Independent Data Unit Protection]

4.2.2 Do we need to pass the UserId as a parameter or there is some other way of getting it?

John Sebes (April, 1998):

Briefly, the answers are: transport of client ID data is performed by CORBA security mechanisms. One does not need to pass user IDs as parameters, and even if one did, there would still be the authentication issue. Authentication functionality is part of the CORBA security mechanism for transporting client ID. There is no CORBA Security Service Context in IIOP per se, because the context data is carried as part of a security protocol for IIOP– either SSL or SecIOP.

As an example of how such things are done today, consider a typical IIOP/SSL implementation. There is no identity or authentication data in IIOP at all. The SSL session setup protocol includes authentication and exchange of digital certificates that include identity information. Security-aware applications can obtain client identity information either: (a) via CORBASEC interfaces for accessing the ”Current” object, or (b) proprietary or ORB-specific interfaces for accessing data in X.509 certificates passed via SSL.

This should answer your questions with respect to mechanisms. However, there are several open issues for actually using these mechanisms, for example: availability of FSP from ORB vendors, implementation of CORBASEC, SSL, and/or SecIOP; integration/bundling of public-key infrastructure elements (e.g. certificate management); facilities for trust management in certificate evaluation (how can you control which certificates you actually believe?)

In summary, the mechanisms exist, but may not be available in FSP, and in any case require some careful thought for safe and effective use.

4.2.3 How would one incorporate security into an ORB system in the next 6 months, so that the solution would not be obsoleted in the following 6?

Andreas Vogel (October, 1997): Use SSL.

4.2.4 Does CORBA security guarantee that the request and reply are not tampered and not intercepted on their way between the client and the target?

Roland Turner (July, 1998): Both of these levels of protection are catered for by CORBA's Security service. What a vendor provides is its own choice, often subject to relevant legislation. (Note that the entire CORBA Security service is a service and thus not a mandatory part of a CORBA [ed: Core] implementation.)

4.2.5 Is it necessary to secure naming service?

Andre Srinivasan (July, 1998)[22]: It depends what you're doing.

[22]Message-Id: qy4svyqbgp.fsf@ahi.visigenic.com

If you're concerned about authentication, your DNS lookup was not secure and it would give you confidence you were talking to the right name server. On the other-hand, the client could have a list of trust points (certificates) to compare the identity presented by servers against, and a rogue name service will not be able to direct you to a rogue target.

If you're concerned about privacy when communicating with the name service, someone could snoop the wire waiting for you to contact the name service and then watch where you go next. I would therefore argue that your privacy would be compromised anyway (with respect to the TCP association).

Nick Battle (July, 1998)[23]: We went round the houses discussing the issue about security of name services (but based on a CSI-ECMA protocol). In the end, we decided that a CORBA name service (actually our Trader service) couldn't sensibly be (CORBA) secured. It came down to bootstrap problems, such as not being able to secure the connection to the Trader to get trustworthy object references for the Authentication and Privilege (logon) services until you'd already logged on etc.

I realise an SSL based scheme might well be different in this respect, since an SSL client is more autonomous (not requiring the services of external objects in order to obtain Credentials, true?). But I'm not really sure about this ... hence the question 7.5.12.

I think that [ed: securing naming server] is not just an option, but a REQUIREMENT on clients, unless they can otherwise obtain a secure source of object references. Clients that don't do this can assume nothing about the true identity of the target - eg. whether data returned is trustworthy, or whether effects they believe they have produced have actually occurred. And this is true EVEN if they turn on *EstablishTrustInTarget*, which is particularly frightening.

If this is true, I think the consequences should be made much more visible to application writers. For example, one consequence is that security UNAWARE applications - those that can't call get_security_names and check them - MUST operate on a trusted source of object references, since they can't check security names for themselves.

4.2.6 △ How to come up with application security design using CORBA Security service?

4.2.7 △ How does a security-aware application specify the use of a specific algorithms for supporting communication confidentiality and integrity?

Konstantin Beznosov (November, 1998): Currently, there is no a standard way for a security-aware application to specify specific QoP algorithms. However, the OMG technical committee issued "Quality of Protection Management and Control" Request For Proposal[24] on November 13, 1998, to address exactly this problem.

[23] Message-Id: 35C579BF.3E3AE0D9@x400.icl.co.uk

[24] http://www.omg.org/techprocess/meetings/schedule/QoP_Management_&_Control_RFP.htm

4.2.8 What is available in CORBASEC for strong (writer-to-reader) authentication?

David Chizmadia (September, 1998)[25] : The CORBA Security Specification (CORBAsec) currently give apps the ability to indicate that they want the ORB to establish a mutually authenticated connection between a client object and a target object. The strength of the mutual authentication is determined by a policy set by the ORB security administrator and optionally overriden by the application.

4.3 Administrator

4.3.1 What are the semantic connotations for rights in CORBA rights family?

Extended Question Gerald Brose: "The Security Service Specification (Rev 1.2) specifies an default access control policy. This policy uses "rights" for authorizations. Also, a default rights family "corba" for use with the domain access policy is defined (p.15-124) that has rights (s,g,m,u) for set, get, manage and use. The option of defining new rights families, is severely limited by the definition of rights families as structs, and is explicitly discouraged in the spec. to keep things simple.

Actually, I think this is absolutely inappropriate, but I might be missing the essential points that justify this design. My question therefore is:

Given that the corba rights family is intended to serve most cases, what **exactly** are the semantic connotations for these four rights? Are they simply chosen in an ad hoc way, or is there some deeper reasoning behind this choice, such as why it would make administration easier in some cases? If so, how and in which cases?"

Bob Blakley (April, 1998):

This is a good question, and one which we discussed extensively during the initial definition of the specification. The basic motivation for defining a small, standard set of rights and strongly encouraging everyone to live with that set is that there are a potentially unlimited number of methods in any given CORBA system (each new class can introduce large numbers of them), and the set of methods is semantically very complicated from the viewpoint of the administrator – methods with the same name may do different things, methods with different names may do the same thing, methods may have names which do not at all suggest their function or sensitivity, and methods belonging to the same class may have very different consequences if invoked on different instances with different internal states. This makes it almost impossible for administrators to manage policy using methods. Rights are thus introduced as a way to "group" methods. We could have stopped after introducing the notion of rights, and allowed implementors or even administrators to define arbitrary collections of rights, but we felt that this would lead to a chaotic situation in which the population of rights would be widely variable across different vendors' implementations and different customers' or even departments' deployments, making training and interoperability a nightmare.

[25] Message-Id: 01bdeb1888404de01b033390@moccasin.tycho.ncsc.mil

We chose instead to conceive of rights as a kind of language, to be used definers of new object classes to communicate the sensitivity of their classes' methods to the security administrator. We defined a small language of rights which corresponded generally to the KINDS of operations which an object-oriented system's methods perform, namely:

method reads and returns one of the object's data members method writes one of the object's data members method executes one of the object's member functions

We defined a right corresponding to each of these basic KINDS of operations, and added one more right to deal with the real-world fact that some operations of the same KIND are more sensitive than others of the same KIND.

Hence the intended semantic connotations of the rights in the "corba" family are:

s ("set"): required to access methods which modify an object's internal state (e.g. setter methods for data members)

g ("get"): required to access methods which return, but do not change, an object's internal state (e.g. accessing readonly attributes or other data members; getter methods)

u ("use"): required to access methods which perform computations or call other objects (e.g. member functions)

m ("manage"): required, usually in addition to one of the other three rights, to access methods which perform management activities, are unusually sensitive, or are otherwise intended for use only by specially privileged callers.

Note that these semantics are NOT "exact" in the sense that they have neither formal nor normative definitions. Nevertheless, I think it's quite clear to both class definers and system administrators what they are supposed to mean, and how they can be used.

Clearly they aren't an exact match for all possible security policies in a CORBA environment, but I don't think a system which supports an exact match for all possible policies would be one which could be administered by normal humans.

4.3.2 How to use the access control mechanism?

Extended Question: Christoph Haenle[26]: "I got stuck reading chapter 15 of the Corba Security Spec (Nov '96). I wonder if Corba's ACL scheme can really provide access control on a per method and per user grain. At least I can't see from the spec how this could be accomplished.

If I got it right, individuals have a bunch of privilege attributes such as access_id, group, security clearance, role, etc. The administrator can grant rights to those attributes, for example rights to call read-methods (get), write-methods (set), or management-methods (manage). The DomainAccessPolicy object stores all this security data in a table, such as (simplified)

[26]Message-Id: m0zQ7uF00401fC@jacht.cs.vu.nl

Privilege attribute	Granted Rights
access_id:alice	get, set
access_id:bob	get, set

On the target side, a ”RequiredRights” table exists, specifying which rights are required for each method. For example

Required Rights	Operation
get	m1
get	m2

Now, because Alice holds get (and the set) right, she can call both methods. How could we specify get-permission on m1 for Alice, and on m2 for Bob, but no permission on m1 for Bob and on m2 for Alice?”

Bob Burt (October, 1998)[27] : Try this:

Privilege attribute	Granted Rights
access_id:alice	get, set, do_m1
access_id:bob	get, set, do_m2

Required Rights	Operation
do_m1	m1
do_m2	m2

Gerald Brose (October, 1998)[28] : That will do only for that particular situation, but the basic problem still remains: the default CORBA DomainAccess Policy scales neither up nor down! It does not scale down (meaning: provides fine-grained access control on a per-user/per-object/per-method basis) because there is no way to grant rights to access individual objects within a domain. As a consequence, you will have to manage a large number of very small domains each containing only one object of a particular type.

On the other hand, the scheme does not scale up to large numbers of users (pricipals) and different object types, because the number of different combinations of granted/required rights is finite, so in order to avoid ”rights clashes” like the one outlined in the first posting, you will again have to restrict your way of writing policies to relatively small domains - which is inconvenient if your policies happen to apply to large number of objects and principals. All in all, I’d say that there are quite a few limits in that paricular access model.

[27] Message-Id: 3618BF52.2045DD17@2ab.com
[28] Message-Id: 3619BEF3.8909468A@inf.fu-berlin.de

4.3.3 Do I have to ”protect” every object, even those which are not thought to be used from outside?

Extended Question: If I have several CORBA-Servers in one HOST. Some objects of the servers are thought to be accessed from remote Hosts, but other objects are thought to be accessed by other servers which are located in the same Host (the inter-process communication in the Host is made with CORBA). Is there a possibility to (in an authenticated manner) know, if a request to an object comes from ”outside” or from the local Host ? If there is no possibility, have I to ”protect” every object, even those which are not thought to be used from outside?

Jonathan Biggar (July, 1998) : This sounds like a good place to have a firewall. The new CORBA firewall specification which might be approved shortly will allow a firewall to mediate CORBA invocation access and prevent access to ”internal only” objects. This will allow you to have a simpler (or no access control scheme) for those objects.

Nick Battle (July, 1998) : A target has access to a client’s authenticated credential attributes, but these do not specify the location of the client. I don’t think there is a CORBA standard way to achieve what you are asking, though at a lower level some security mechanisms may be able to give you trustworthy information about the location of the peers in an association.

If protection is to be achieved on the basis of location, and CORBA does nothing to help verify location, yes. CORBA Security protection works on the basis of a principal’s credential attributes, not its location. This is sometimes a surprise to people who are used to thinking about security in terms of firewalls and other such location/topological constraints.

4.3.4 △ How is related work at OMG on Security Administration and Common Management Facilities ?

4.3.5 √ What is the granularity of access control on object invocations?

[ed. For more detailed and alternative answers see SecSIG mail list discussion thread titled “Granularity of Invocation Access Controls”]

Bob Blakley (June, 1999)[29]:

CORBAsecurity provides access control whose granularity is 5. GROUP of operations on a GROUP of instances of (any number of) interfaces How? Like this:

1. Pick some number of interfaces whose instances you want to control. For each such interface, assign its operations ”required rights”.
2. Pick any number of instances of any number of interfaces. Put them into a domain
3. Assign an ACL to the domain. The ACL assigns ”granted rights” to ”privilege attributes”.

[29] Message-Id: 031e01beb1dc28578b6024a13994@shaggy.austin.dascom.com

Now, for every instance in the domain, a requesting user may invoke any operation whose required rights are ”dominated by” the ”granted rights” which the requesting user has because of his privilege attributes.

4.3.6 √ Where are access control lists stored?

[ed. For more detailed and alternative answers see SecSIG mail list discussion thread titled “Granularity of Invocation Access Controls”]

Bob Blakley (June, 1999)[30]:

In Policy objects, which are associated with the *DomainManager* instance corresponding to the domain whose policy they define.

Polar Humenn (June, 1999) [31]:

If one subscribes to the D[omain]A[ccess]P[olicy]/R[equired]R[ights] access decision logic, they are stored in basically two places. A *DomainAccessPolicy*, which maps security attributes to rights (although that mapping is not well defined), and the *RequiredRights* object (which is locality constrained). I guess their persistence is up for grabs behind the implementation as far as the specifciation goes.

4.3.7 √ How do servers “know” what domain to put new objects into and when to create new security policy domains?

[ed. For more detailed and alternative answers see SecSIG mail list discussion thread titled “Granularity of Invocation Access Controls”]

Bob Blakley (June, 1999)[32]:

The intent is that there should be a policy governing which domains newly-created objects are put into, and that this policy should be administered just like any other security policy. Given this policy, servers can simply programmatically assign objects to the correct domains as they’re created.

ACLs certainly DON’T need to go away when the objects they control are destroyed. There’s no reason ”empty” domains shouldn’t stay around – people might want to put new objects into them later.

4.3.8 √ What about transient objects created by factories?

[ed. For more detailed and alternative answers see SecSIG mail list discussion thread titled “Granularity of Invocation Access Controls”]

[30] Message-Id: 05c301beb1e97e24f48024a13994@shaggy.austin.dascom.com
[31] Message-Id: Pine.LNX.4.10.9906081549200.30704-100000@marcy.adiron.com
[32] Message-Id: 05c301beb1e97e24f48024a13994@shaggy.austin.dascom.com

Bob Blakley (June, 1999)[33]:

The idea here is that factory objects should conform to the administered creation-time policy which determines what domain(s) an object should be assigned to. This is really independent of whether or not the objects are transient or persistent, and of whether they are named or anonymous.

4.3.9 √ How would access control mechanisms be applied to secure, let's say, naming service?

Extended Question: (Paul H Kyzivat) : Consider a server implementing CosNaming. Now this server is available widely. Everyone can get a reference to an initial NamingContext. The NamingContext interface permits reading, writing, traversing, contexts as well as creation and destruction of new contexts. Now I probably don't want everyone to have free reign to do all of these things. Rather, I most likely want to administer it in similar way to how a file system is administered:

- Some set of users should have full access to the initial context.
- Most other users should be able to have access to read and traverse this initial context.
- Additional contexts will exist and be bound into the initial context. Full access to these should be grantable to other users.
- Repeat the above recursively.
- Other filesystem-like access control that I failed to mention.

Can you show me how CORBA Security can provide this functionality?

Polar Humenn: (June 1999)[34]:

If you prescribe to the Domain Access Policy and Required Rights model, this is how it should be done.

The first thing you have to do is define a *RequiredRights* object that each context will have access to. Logically, all contexts will see the same required rights object, but that cannot be enforced, since the naming context can bind contexts of different services coming from anywhere.

But lets say for the moment, the naming service and each context in it prescribes to the same required rights object however that is done.

Give the "resolve" operation on the naming context interface the "0 0 get" right, (0=family definer, 0=family) give the "bind" and "bind_context" operations the "0 0 set" right, with a combinator of *SecAllRights.*

Now, set up a domain manager for each context! Each domain manager must have a SEPARATE Access Policy.

For each Access Policy you narrow it to a *DomainAccessPolicy* and you must give the "0 0 get" right to the one attribute that represents the user. (This could be *SecAttributeType* (0,1,2) Access Id (0=family definer, 1=family, 2=type). Give the "0 0 set" right to the attribute that represents users you want to be able to do binds.

[33]Message-Id: 05c301beb1e97e24f48024a13994@shaggy.austin.dascom.com
[34]Message-Id: Pine.LNX.4.10.9906091432030.32438-100000@marcy.adiron.com

Here are the limitations: The initial context will have a problem, however, that you cannot specify the ”get” right for ”ALL” users. One would think that you could specify that ”no” rights are required for access to the initial context’s ”resolve” operation and that would do it. However, if everyone prescribes to the same *RequiredRights* object, your hosed, because you wouldn’t be able to protect the ”resolve” operation on other contexts.

We did an experiment a while ago, trying to map Unix id’s into CORBA and use the naming service as a map of a Unix file system. Each context was a directory, and each file was a *CosSerializable* object (or some such animal).

For each context, the context had to create ”on the fly” a domain manager and an *AccessPolicy* object (not a *DomainAccessPolicy* object). This was easy, since the Naming Service in question was a single server and could do these sorts of things.

The *AccessPolicy* object mapped a certain security attribute representing the user to its corresponding Unix id, and threw back the rights, ”read”, ”write”, ”search”, according to the permissions of the Unix id based on its authorization information on the local system, i.e. owner, group, other.

However, we couldn’t do that kind of policy with a plain *DomainAccessPolicy*, because there isn’t a way, unless we had some sort of ”wild-card” security attribute, that you could say ”other”. But even then the specification of the *DomainAccessPolicy* rules aren’t good enough to do an ordered check, say, first check ”owner”, then check ”group”, then check ”other” relationships.

Bob Blakley (June 1999)[35]:

There are actually a couple of ways (in detail) to do it. I’ll give one example.

Naming context is an interface, so it needs to have an entry in a RequiredRights table. Add this interface to the system’s unique RequiredRights object and set up required rights for the its methods. For purposes of argument, map ”read” and ”traverse” operations to the ”g” right and other operations to ”s”. This may be too simplified; if a more complex mapping is required it’s straightforward to produce it. If you need one, a custom rights family can be defined.

- *Some set of users should have full access to the initial context*

  Right. Form a group ”InitialContextAccessors”. Add the privilege attribute corresponding to this group to the credential of each user who is to be granted full access to the initial context.

- *Most other users should be able to have access to read and traverse this initial context*

  Right; form another group ”InitialContextLimitedAccessors”. Add the privilege attribute corresponding to this group to the credential of each of the users who does not have full access, but does have ”read and traverse” access to the initial context.

  Now put the context object into a domain. Give it an ACL with two entries; one which grants ”gs” to ”InitialContextAccessors” and another which grants ”g” to InitialContextLimitedAccessors”.

[35] Message-Id: 025b01beb2b94dc1e84024a13994@shaggy.austin.dascom.com

- *Additional contexts will exist and be bound into the initial context. Full access to these should be grantable to other users.*

  Not clear exactly what you mean here, but I'm guessing what you mean here is that the same users who have access to the parent context also have the same level of access to all embedded contexts; additionally some other set of users who do not have access to the initial context should have at least some rights (full access in your example) to the embedded context.

  There are a couple of ways to do this; depending on your implementation you may have several options. If your implementation allows nested domains (i.e. domains within domains) then you've got an easy task here:

  Simply create a new domain and put it in the first domain. Add the embedded context to the new domain. This will have the effect of causing the embedded context to inherit the policy governing the initial context. Now:

  1. Create a new group "FirstEmbeddedContextAccessors".
  2. Add the privilege attribute corresponding to this group to the credential of all users who should have access to the users who have access to the embedded context (including perhaps some who don't have any access to the initial context).
  3. Create an ACL (i.e policy object) for the new domain. Give it one entry, which grants "gs" to "FirstEmbeddedContextAccessors".

- *repeat the above recursively.*

  repeat the above recursively.

This results in a structure with one domain per context, one ACL per domain, and (on average, excluding one extra entry in the initial domain) one entry per ACL. Each user is a member of as many groups as the domains he needs to access. The policy data here scales as follows:

1. Domains: linearly proportional (n=1) to the number of naming contexts to be protected
2. ACL data: ACLs AND entries linearly proportional (n=1) to the number of contexts to be protected, but policy is inherited reducing what would otherwise be an n*log(n) scale.
3. Groups: for each user, linearly proportional to the number of naming contexts which the user needs to access – but this is worst case, assuming that user populations needing to access disjoint naming contexts are always disjoint. If naming contexts overlap substantially in authorized user populations, then you can achieve sub-linear scale here.

4.4 Implementor

4.4.1 △ Where can I find some source code which implementation Security Service?

4.4.2 △ Is there any document on how to implement the CORBA security service?

4.4.3 △ If I want implement the CORBA security service, what should I do?

4.4.4 What is the intent of the credentials object design?

Belinda Fairthorne (August, 1998)[36]: In case it helps, the following is some background into the definition in the original OMG spec about the application (or interceptor) interface for credentials and security attributes.

The ”credentials” object represents the ”principal”. It was defined with a number of different possible underlying security mechanisms in mind - which might use (PK) attribute certificates, Kerberos/DCE service tickets etc. Strictly, credentials can contain both unauthenticated and authenticated attributes (see 15.3.2) - the unauthenticated case being used mainly for ”public” where the principal has no authenticated id, so only has access to public information.

CORBA Security defines the interfaces, not the architecture, in line with other OMG standards. As a security person, I clearly want to know if the credentials can have been tampered with, but that is a feature of the implementation, not the interfaces. So we said CORBA Security implementors must define ”assurance” information about their products, including threat information, and users have to check that. Would the OMG rules allow statements about the level of authentication of the id and other attributes? If so, it would be nice to clarify this, though I’m not sure how easy it is to define. (The current spec allows different levels of security here and elsewhere).

Security attributes were intended to include both identities and privilege attributes - both seem needed. Therefore a ”principal” definition which does not allow privilege attributes would seem a retrograde step.

We did not define the syntax of possible name types, though, as Nick says, this seems a good thing to add into the standard. The problem was that different underlying mechanisms use different name types e.g. X.509 names, DNS ones, operating system ones, so applications which need to be portable between different mechanisms should be able to work with different types.

We wanted to allow a range of access control mechanisms, including roles, groups, clearances etc, and also ones specific to particular organisations (for which we had seen requirements). Hence having privilege attributes as well as identities in the credentials. Specifying standards for more families, types, associated value syntaxes etc in the OMG standard would be good. (However, a quick look at the ietf draft of GAA_API seems more restricted than CORBA Security, and also includes an access decision interface which CORBA Security already has.)

The application sees a credentials list (rather than just a single credential) in cases where the delegation model used can provide both the credentials of the ”initiating”

[36] Message-Id: 28091.221183884@x400.icl.co.uk

principal and also credentials for one or more intermediates. The initiating principal is the one whose attributes will be used for access controls, auditing etc. However, the target application may not want to use these unless they have come via trusted intermediates (and may want to control access based on the intermediate's attributes in some cases). We discussed whether we could specify a recommended delegation model for CORBA Security, but there were several delegation models in existing implementations, all of which had pros (as well as cons). There did not seem any generally market preferred model, and we felt these delegation models were still immature, so we specified an application interface which allowed for the delegation models in a number of products, both secret and public key based ones - the ones we looked at included Kerberos, DCE, SESAME and Tivoli, all of which were different, hence the delegation variants defined in 15.3.6.3 etc. (We also looked at "reference restriction" forms of delegation such as Sun were implementing, but could not get an agreed definition, so did not include in the formal definiton). The semantics of the credentials list was not defined, as it is dependent on the mechanism (e.g. simple delegation would only provide a single credentials object, where composite and traced delegation would give a list), but the initiating principal's credentials are always first (the only ones wanted in many cases). OK, this was a bit of a cop-out, but is there yet a delegation model which is generally agreed enough yet (and being implemented) to include in CORBA Security as the standard?

... For the intial CORBA Security spec, delegation was discussed at length, and we found different models suited different requirements, but there was no one model which satisfied most requirements and was easier enough to administer. Simple delegation was the nearest (as it meets the ease of admin requirements, if not some of the control ones).

4.4.5 △ Does the existing Authorization Service of CORBASec scale in a "well" distributed-object environment?

4.4.6 Can a client implementation circumvent administrative security policies?

Extended Question (Christoph Haenle[37]): From the CORBA security spec, I see that administrators can enforce policies such as "whether this client can use this operation on this target object, whether the invocation should be audited, [...]" (section 15.6.1 of CORBA Spec).

This part of security is enforced in the ORB, for example, through the "client access control interceptor" (Figure 15-53).

My question is: Can't the client circumvent the restrictions? Assume we use C++ for the application and the stubs objects. Now, does the ORB run in a different address space than the application

(incl. stubs)? Or is the client part of the ORB linked to the application at compile time (and thus running in the same address space)? If the ORB were in the same address space, then the client could just modify the ORB code to circumvent the policies that the administrator has imposed. Even if the ORB were running in a different

[37] Message-Id: 729cas9771@star.cs.vu.nl

address space, the client could still write a piece of software which behaves like the ORB, but leaves out the routines where administrative restrictions are enforced. The client could then his ”custom ORB implementation” rather than the ”real” ORB for communication, hereby circumventing administrative security policies.

Rudolf Schreiner (November, 1998)[38] : Security enforcement on the client side is always a bad idea. CORBASEC enforces authentication and authorization on the server side. The client passes a kind of ticket to the server (Kerberos!!). Without the right ticket the server refuses to talk to the fake client. On the client side you can do whatever you want, without the right ticket (credentials) you can't do nothing on the server side.

In my Kerberos 5 based implementation I use GSS-API to get a ticket from outside the CORBA system. Then I use GSS-API to establish a security context between client and server. This is done at a very low level during the bind, before any IIOP messages are sent. If the server can't authenticate the client the connection is dropped. All IIOP messages are encrypted/decrypted using GSS function.

In theory such a system is quite bullet proof. In the real life an attacher might be able to exploit a buffer overrun on the server ORB and execute arbitrary code. There are some other possible attacks, too.

4.4.7 √ What is the ”public” security attribute of a principal?

Bob Blakley (June 1999)[39]:

The ”public” attribute is what you get in your credential before you authenticate (i.e. after you've initialized the ORB, which comes up with an ”own” credential, but before anything has called *PrincipalAuthenticator*.

4.4.8 √ Under what circumstances do *Credentials* contain the “public” attribute?

Bob Blakley (June 1999)[40]:

All credentials of unauthenticated users certainly contain this attribute; the intention was that all credentials always contain it but I'm going to have to look to make sure that this was faithfully captured in the spec.

4.4.9 √ What is the value and the defining authority of the “public” attribute?

Bob Blakley (June 1999)[41]:

Its value is ignored (i.e. it has no useful value). Its defining authority should be OMG.

[38] Message-Id: Pine.BSF.3.91.981111194841.11152A-100000@phobos.muc.de
[39] Message-Id: 019001beb3660$d7ec$34024a13994@shaggy.austin.dascom.com
[40] Message-Id: 019001beb3660$d7ec$34024a13994@shaggy.austin.dascom.com
[41] Message-Id: 019001beb3660$d7ec$34024a13994@shaggy.austin.dascom.com

5 CORBASEC implementations

5.1 General

5.1.1 √ Where can I find an implementation of Security Services ?

Alphabetical list of CORBA Security products or announcements about such products:

CBConnector from IBM http://www.software.ibm.com/ad/cb/ (Jim Beale (December, 1997))

DAIS Security from PeerLogic http://www.peerlogic.com/products/dais/f_dais.htm. See section 5.2.1 for more information on DAIS Security.

Dreadnought from Phalanx Technologies http://www.phalanxtech.com/Dreadnought.htm:

Tom Herron (May, 1999) [42]:
The current list of features includes but is not limited to:

- Pure java code base
- XML security policy
- ORB-level secure association based on SSL protocol
- Password authentication
- Group and hierarchical roles authorization based on digitally signed/time stamped delegatable identities
- Multidomain effective rights - objects are placed in effective rights domains with separate privilege authority (identities recognized relative to parent or sub-domain identity based in LDAP style naming)
- Confidentiality per operation can be set on or off
- Integrity per operation can be set on or off
- Auditing - basic, recorded for authentication and authorization and all failures
- Credential delegating firewall proxy
- Remote administration which includes limited creation/authoring of effective rights subdomain identities/privileges to subdomain administrators (i.e. allows top level (heavy weight)administration and numerous light weight subdomain identity authoring)

Expersoft Shahzad Aslam-Mir (January, 1998): Expersoft[43] will be offering security service in 1998.

Inprise Andre Srinivasan (November, 1998)[44]: Inprise will be shipping, as an addon to the AppServer and VisiBroker, a security product that builds on our successful SSL based product to provide features like instance based access control and auditing as described by CORBAsec.

[42] Message-Id: 374ABD89.E3DE7000@phalanxtech.com
[43] http://www.expersoft.com
[44] Message-Id: qyr9vigh4k.fsf@ahi.visigenic.com

IntraVerse from DASCOM[45]. The press release issued by DASCOM on June 14 had the following information on the product:

> "... IntraVerse for CORBA (IVCorba(TM)) solves the primary problem limiting full CORBA deployment at many organizations, the need for security and authorization services. IVCorba 3.0 is the only enterprise CORBA single sign-on and authorization solution that supports secure interoperability between Object Request Brokers (ORBs) from multiple vendors, providing users with the flexibility in their development and deployment of CORBA applications.
>
> Furthermore, IVCorba 3.0 provides the only CORBA Security Service to implement controlled delegation. This function provides the ability to pass user and server information with a transaction request, enabling extremely fine-grained access control.
>
> ...
>
> IVCorba 3.0 provides full support for Iona (NASDAQ: IONA - news) OrbixWeb(TM) as well as Inprise (NASDAQ: INPR - news) VisiBroker(TM), the most widely deployed ORBs. By providing a cross-ORB external authorization solution, DASCOM has freed developers from trying to individually implement security services for each application as it is developed. Furthermore, IVCorba provides a single sign-on solution, enabling personalized services and simplified administration. Users can log in once and access all the resources that are appropriate for them.
>
> ...
>
> Availability
>
> IVCorba 3.0 began shipping June 14 on Solaris and Windows NT. For more information, contact DASCOM at info@dascom.com. "

Nephilim (Java Implementation of CORBA Security Services) of UIUC http://choices.cs.uiuc.edu/Security/nephilim/

ORBAsec SL2 from Adiron[46]

> **Polar Humen** (May, 1999)[47]:
>
> > ORBAsec SL2 2.0 from Adiron is a Java implemenation of CORBA Security giving programmers the ability to write objects and clients using encryption and authentication integrating with Kerberos authentication services or SSL. ORBAsec SL2 uses standard CORBA Security Level 2 interfaces from the CORBA Security Specification Revision 1.5. ORBAsec SL2 2.0 has been available since 2Q 1999.

OrbixSecurity from IONA Technologies http://www.iona.com/products/orbixenter/security/index.html. The web page reads the following, as of June 1999:

> "OrbixSecurity 3 ... available in **Q3, 1999**
>
> OrbixSecurity 3 extends the security capability of the OrbixOTM container (OrbixSSL) by providing a manageable, scalable security infrastructure for Enterprise

[45] http://www.dascom.com/

[46] http://www.adiron.com/

[47] Message-Id: Pine.LNX.4.10.9905241618420.8225-100000@marcy.adiron.com

systems. A full implementation of the CORBA security service Level 1 and more, OrbixSecurity 3 delivers a comprehensive security service systems based on the IETF's Secure Socket Layer (SSL) security.

Secure your Enterprise system with OrbixSecurity 3

Based on, and incorporating all features of the IETF's SSL V3.0, OrbixSecurity 3 extends the functionality of SSL, adding CORBA Security Service Level 1 functionality and extra features such as an Administration GUI to make the creation and updating of security policies an easy, manageable and scalable task. OrbixSecurity 3 provides fine grained control of security in the system - security is added at the application level allowing access control checks to be performed on a per object, or even per method basis. CORBA Security Level 1+ Services enforce basic audit and access control functions in security-unaware applications, as well as providing limited Applications Programming Interfaces (APIs) for enabling security-aware applications to manage their own security. Ideal for Internet and Enterprise systems, OrbixSecurity 3 provides the following features and functionality:

All features of CORBA Level 1 Security including:

Identification and Authentication: based on SSL Authentication. User Ids with password login or SecurID tokens can be used for identification. These different options mean administrators can choose to implement the authentication method most appropriate to their system - depending on the sensitivity of the data in the system, and the degree of potential risk to that data.

Authorization and Access Control: Allows access to resources to be controlled based on user identity. Support for multiple user types is included - the user name, group or organization can all be used to make authorization checks. At the Server side, access control decisions can be made on a per server basis, per interface or per method. This allows for controlled access to sensitive company data, ensuring confidentiality of any non-public information.

Security of communication: Data privacy and confidentiality are provided through use of the Secure Socket Layer - providing encryption and digital signature algorithms. This ensures that data cannot be read or modified whilst in transit. System users can rest assured that sensitive information such as financial information, or company proprietary data, cannot be read by eavesdroppers or hackers.

Delegation of Privileges: Allows a client to delegate security privileges to an intermediate application acting on its behalf. CORBA Level 1 supports simple (unrestricted) delegation of privileges. Administrators can decide whether to allow or prohibit delegation.

Security Auditing: Allows security-authorized administrators/personnel to monitor users' actions in the system, and what system resources they are attempting to access. All security-relevant events are audited and logged if necessary, to allow detection and assessment of damage of successful attacks. This allows the security system administrators to detect and gauge the damage caused by successful attacks.

Unitary login is also supported, providing a secure repository of mainframe authentication data and passwords to allow developers to build applications that gain access

to mainframe systems, thus requiring users to log in to an enterprise system only once.

Managing large scale secure systems with OrbixSecurity 3

OrbixSecurity 3 is designed to make securing a large scale Enterprise system an easier task. CORBA Level 1 Security services provide security to applications that are typically unaware of the presence of the security service in the system, and can be added to applications without having to alter any existing code. This makes adding security to an existing system a much easier task. OrbixSecurity 3 also provides a comprehensive GUI interface to make implementing security policies quick and trouble free. A Java - based graphical user interface enables security administrators to easily create and update security policies.

OrbixSecurity 3 will provide full cross language functionality for both Orbix and OrbixWeb C++ and Java applications, and will be available on OrbixOTM 3 supported platforms."

See section 5.2.2 for more information on OrbixSecurity.

PC-DCE/NetCrusader from Gradient[48]

SecureBroker from Promia[49] http://www.promia.com/products.html

5.1.2 △ Where can I find exactly what product implements what Security level and options?

5.1.3 √ What ORBs claim to have "security" functionality?

Here is a list of those products that do not claim compliance with CORBA Security specification. Nonetheless they state that they have some security functionality. Read answer to question 5.1.1 to see what implementations of CORBASEC are available:

Bill Janssen (November, 1998)[50] : We have just released ILU 2.0alpha13, which contains a **security system based on the IETF's GSS**. It allows you to wrap client-server communication in arbitrary security contexts, depending on which security mechanisms you choose in your application. It also provides GSS-based principal identification. It's NOT CORBAsec. More ILU info at ftp://ftp.parc.xerox.com/pub/ilu/ilu.html.

Rohit Garg (December, 1998)[51] : ObjectScape is pleased to announce the availability of JBroker 2.0 beta. JBroker 2.0 provides high performance and scalable Java CORBA ORB, services, development, and management tools that have been specially designed for building high end Java servers including Enterprise JavaBeans servers.

The beta features include:

- IDL to Java, and Java RMI to IIOP compilers

[48] http://www.gradient.com
[49] http://www.promia.com/
[50] Message-Id: y1azpa5mx4u.fsf@watson.parc.xerox.com
[51] Message-Id: 366EB933.EE76CBC6@objectScape.com

- Portable Object Adapter (POA)
- Java Objects by Value
- Server Activation
- IIOP Connection Concentrator
- **Pluggable Authentication and Identity propagation**
- Support for plugging in a Java Transaction Service
- IIOP Bootstrap Protocol
- Secure, Load Balancing, and Fault Tolerant COS Name Service with JNDI support
- Management/Monitoring support
- SPIs for writing adapters, threading/queueing models, building multi-machine clusters, etc.

IIOP/SSL and light-weight multicast object invocation support will also be released shortly.

For details, please see:

1. http://www.objectScape.com/jbroker.html for JBroker 2.0 data sheet

2. http://www.objectScape.com/performance.html for performance comparison with other commercial Java ORBs

3. http://www.objectScape.com/appserver.html for white paper on ”Building an EJB Server using JBroker”

If you are interested in evaluating JBroker 2.0 for your projects, please send an email stating you interest to evaluations@objectScape.com. If your interests meet our beta objectives we will contact you with details on how to get our software.

5.1.4 Does anyone know of a product that is IIOP compliant and provides CORBA security service level 1?

Charles White (June, 1998): Our product [ed: http://www.blackwhite.com] Object/LM can handle these tasks. It is (the only) an implementation of the CORBA Licensing Service and has been implemented for both Orbix and Visibroker.

It handles postage stamp (# of uses) and gas style (length of time) metering, plus authentication, authorization and encryption.

Serban Tatu (November, 1998)\emailmessageid{3641DBB7.6792FE30@starvision.com} : [ed: white paper at] http://www.iona.com/support/whitepapers/orbixsecurity/ shows you how Orbix implements the (Level 1) security service.

5.1.5 △ Is there any free/trial/evaluation version of an ORB with Security Service for Java?

5.1.6 △ What would be the most suitable ORB product(s) when buliding a (very) small lab for evaluating, testing and implementing security functions in a CORBA system?

5.1.7 △ Are CORBAsec implementations from the US generally subjected to export control?

5.2 Particular Implementations

5.2.1 DAIS Security

5.2.1.1 What is DAIS Security?

Linda Gricius (March, 1998): DAIS is a CORBA 2.0 ORB, plus CORBA services, developed by ICL. In 1997, ICL released the first Beta of the DAIS Security service. Based on feedback from program participants, a second beta was released in late 1997.

5.2.1.2 What is the current version of DAIS Security

Linda Gricius (March, 1998): 1.0.2 beta.

5.2.1.3 What is the current status of DAIS Security?

Bruno Traverson (September, 1998)[52]: DAIS is now commercialized by PeerLogic. See http://www.peerlogic.com

5.2.1.4 Does DAIS conform to CORBASEC specifications?

Linda Gricius (March, 1998):

Required functionality – DAIS Security provides near level 2 conformance with the CORBASEC specification. The only Level 2 features not supported are: Multiple credentials for a user Delegation Application level audit and access control Administration by standard policy objects.

Security Functionality Options – DAIS Security does not support non-repudiation - the only CORBASEC option at this time.

Security Replaceability - ORB replaceability isn't supported. The ORB interceptors in DAIS Security version 1 are not standard. Services replaceability is supported (ORB security objects are implemented to CORBA standard interfaces and could therefore be replaced).

Secure Interoperability - In DAIS, secure associations are established using SECIOP messages in DAIS Security, but the SECIOP messages are carried inside IIOP requests rather than following the interoperability standard.

[52] Message-Id: 36121736.2FBB@der.edfgdf.fr

DAIS Security does not support DCE-CIOP.

Conformance to CSI Spec - DAIS (i.e. SESAME) generates full identity and privileges (i.e. it has ECMA PACs), which points DAIS at CSI level 2. But level 2 also has controlled delegation. The first release of DAIS doesn't support delegation, which points DAIS back at level 0. So DAIS does level 0, and bits of level 2 (the privileges in addition to identity), but don't do controlled delegation.

5.2.1.5 Why did ICL choose the CSI-ECMA security mechanism in its DAIS Security implementation?

Linda Gricius (April, 1998):

Perhaps the best known security mechanism is Kerberos, which was developed by MIT. Kerberos does not provide all of the functionality required by the full CORBA Security model. Therefore, DAIS Security uses a different mechanism called SESAME, because ICL believes that the functionality of the full CORBA model is required to implement enterprise strength security systems. Basically SESAME implements the CSI-ECMA protocol of the CORBA Security interoperability specification. SESAME V4 is essentially Kerberos V5 extended in various ways (and rewritten), in accordance with the ECMA Security standard, known as ECMA-219.

Regardless of the security mechanism used, the DAIS Security service accesses the mechanism via a generic API, called the Generic Security Services API or GSS-API. This is a standard API that presents the same interface to the caller, regardless of the mechanism underneath being used to implement the functions.

5.2.1.6 What features does DAIS Security offer?

Linda Gricius (April, 1998):

DAIS Security provides near level 2 conformance with the CORBASEC specification.

DAIS uses SESAME (CSI-ECMA) as it's security mechanism. This brings a rich set of features to the DAIS Security service implementation, including the use of roles and support for both public/private and secret key technologies.

In addition, DAIS Security provides a full GUI administration tool, which allows the administrator to manage principals, roles, trust relationships, domains, required rights, invocation policies, etc. The GUI is written in Java and is distributed - using DAIS Security itself to secure the communications. This provides much more functionality than what is defined in the CORBASEC specification.

DAIS Security also provides an integrated, offline public key certificate management system, which also has a secure distributed Java GUI.

DAIS Security also offers a SecurityLevel2 programming interface to security aware applications, currently in C++ only.

5.2.1.7 What is the advantage of using roles in DAIS Security?

Linda Gricius (March, 1998):

Principals gain access to services by presenting credentials, which contain attributes, and give them access rights to services. These attributes can be assigned to each individual, or

to a role. By then allowing a principal to authenticate with a given role, all of the attributes associated with that role are put into the principal's credentials.

So, the use of roles is more efficient when administering principals - rather than having to administer attributes for each principal, you can assign attributes to roles, and then allocate roles to principals.

5.2.1.8 What are the advantages (and disadvantages) of using public key technology in DAIS Security?

Linda Gricius (March, 1998):

Public key users have an inherently stronger binary key with which to authenticate. The key is not memorable, and was not selected by the user in any case. Because of their strength, public keys need be changed less frequently - and the certificate that holds the public key contains a built-in lifetime to enforce its replacement after a chosen period. The involvement of trusted "officers" in the creation and maintenance of public keys is also inherently more secure, requiring the collaboration of more if they wanted to abuse the system.

Users that form associations with their public key do not require the services of a Key Distribution Service, even during inter-domain association, which is more efficient at runtime.

Public key users also have an advantage in being able to use the same key to authenticate to many principal domains (if they are configured as members of those domains). Their keys are trusted on the basis of a Certification Authority (CA), the scope of which may span many principal domains. So having authenticated once to the CA system in order to get a public/private key pair, the key may be used to authenticate the user wherever that CA is trusted (until the key expires).

Since public key certificates may be freely distributed and certified at the point of use, there is also a fundamental scalability advantage to systems that use public keys.

5.2.1.9 What are the advantages (and disadvantages) of using secret key technology (passwords) in DAIS Security?

Linda Gricius (March, 1998):

Password users have a simple life - they carry their authentication information (their password) with them in their heads, and most users already understand the idea of passwords. This has the advantage of not requiring that they login at a workstation that has any secrets pre-installed for them. It is also conceptually and procedurally easier for them to be set up to use a password and change the password at regular intervals.

The disadvantage of using passwords is that, by depending on a comparatively short memorable string, a password is inherently weaker than a large binary key value. Users often choose very guessable passwords, or write them down, or never change them, or only change between a small number of alternatives.

In addition a KDS[53] is required to form associations for clients that authenticate using passwords, and in the case of inter-domain invocations, the KDSs of both domains are involved. This is less efficient at runtime.

[53] ed: Key Distribution Service

Password users that have accounts in several principal domains should have different passwords for each domain. Each password is only trusted for use between the user and one domain's Authentication Service (AS).

5.2.1.10 Why have domains in DAIS Security?

Linda Gricius (March, 1998):

DAIS Security supports three types of domain - principal, policy, and trusted identify domains.

The major security advantage in dividing a system into domains is to achieve separation of unrelated parts of an organization - either people/departments or sets of applications/data. By separating the system into domains, the appropriate security controls can be put in place for each domain, and access to information between parts can be controlled. If a domain member deliberately tries to damage the system, the damage they can do will be limited by their domain's security limitations.

The best separation of the security system is most likely to mimic organizational structures that you already have. For example, the reason the Sales department is separate from the Development department is that the two communities deal with different business processes, and consequently have different applications/data and security requirements. Furthermore, the two departments are unlikely to require access to each other's systems. By reflecting this in the domain structure, appropriate controls can be put in place to protect applications/data within domains and to control access between domains.

Principal domains are the easiest way to divide communities of users and application objects. By putting users and related applications in a principal domain, access to objects from users within the domain is possible, but access from users outside the domain has to be explicitly granted.

After separation of responsibilities, the second reason to divide a system into domains is to allow the overall size of the system to grow without unmanageable growth in the size of its parts. A small pilot system is unlikely to have a large number of users, roles, and applications. Therefore, little is to be gained by dividing the system into many principal and policy domains. The overhead of the extra effort to install and manage the system via multiple domains would outweigh the benefits. As an installation becomes larger, the benefits of splitting it into separate domains become more apparent - particularly in terms separation of responsibilities and of the workload on any one particular administrative or ”run-time” security component.

5.2.1.11 Why policy domains?

Linda Gricius (April, 1998):

Within a principal domain, policy domains can be used to group object types (i.e., instances of an interface) and workstations into high, medium, and low security groups. You can have different policy settings for the same interface in different domains. Some groups of very sensitive objects would have very stringent access control requirements and message protection. Other groups may require little or no security because of their insensitive nature.

You can also achieve separation of domains by not defining certain target interface types in certain policy domains; e.g., if the ”lowsec” domain doesn't have the interface ”mis-

sile_launch" defined, then there's no way that clients from that domain can ever invoke objects of that type.

5.2.1.12 Where can find more information on DAIS Security?

Linda Gricius (March, 1998):

More information about DAIS and DAIS security service can be found on its web site www.daisorb.com.

5.2.2 OrbixSecurity

5.2.2.1 △ What is the conformance level of OrbixSecurity?

5.2.2.2 △ Where do I start from in order to use OrbixSecurity?

5.2.2.3 What DCE components are required to use OrbixSecurity?

Bruno Traverson (September, 1998)[54]: The DCE components required for OrbixSecurity - which can be bought from Transarc (Solaris), Gradient (Windows) or HP for HP-DCE, are essentially the Security libraries. In more detail, the components required are :

OrbixSecurity Application Development requires :

- DCE ADK (including audit library)

DCE client hosts require:

1. DCE runtime library
2. DCE Daemon
3. CDS clerk
4. Security clerk

DCE server hosts (in addition to client host components) require:

1. CDS server
2. DTS server
3. Security server
4. Audit server

You will need at least one server host per cell. You do not need DCE DFS.

[54] Message-Id: 360F5F3D.5B9A@der.edfgdf.fr

5.2.2.4 Can a user on a remote machine still run the server and call its methods if he or she changes their username on the remote machine deliberately to match the registered users list?

Dan Hushon (November, 1997): The Orbix documentation warns you that you should not run the daemon from the root uid, as the root on another machine will be able to control and invoke methods on the particular instance.

Dale Nagata (November, 1997): If the user Foo is allowed to invoke or launch the server Bar on host X, then any user on any remote machine Y can invoke or launch the server if Orbix thinks the user is Foo. What you do at the remote client to make Orbix think you are Foo is up to you, whether you actually login with that id, use a filter, or whatever.

5.2.2.5 What authentication process is used in OrbixSecurity?

Extended Question: Sanjeev K. Asher: The OrbixSecurity whitepaper states that (page 20, last line)- "The client must first obtain a security token/key by performing a DCE login."

But DCE uses a RPC call for client-server communication. The CORBA standard is based on connection-oriented calls or TCP calls. Does this mean that the OrbixSecurity is not CORBA complaint? Can somebody please clarify the authentication process used in OrbixSecurity.

Rudolf Schreiner (June, 1998):
I read in the white paper that the Orbix security service uses GSSAPI to access the DCE security service. This doesn't mean that the ORB uses RPC instead of (SEC)IIOP and is CORBA compliant. In this case the internals of DCE and RPC are unimportant. DCE security is just a security mechanism with GSSAPI, like Kerberos V5 or SESAME.

Ludwig Brinckmann (June, 1998):
GSSAPI as such does not actually define how to log on to a system, but how to exchange secure messages between authenticated principals. For this a principal has to present its credentials to the GSSAPI routines. To **obtain** these credentials, a DCE login is used. This actually requires a DCE installation on the client machine and the protocol used for this is DCE RPC. Once the DCE login context has been established, it is passed to the `gss-dce_login_context_to_cred` routine to obtain the GSSAPI credential. (The DCE implementation of GSSAPI provides a few routines prefixed with `gssdce` that bridge between DCE and GSSAPI.) The credential is then used in GSSAPI routines to for a handshake between client and server and then to routines like `gss_sign` to encrypt the messages between peers. Technically the Orbix/DCE security implementation is hybrid: it uses DCE to establish the authentication of client and server and the Tickets obtained are then used to encrypt IIOP traffic. This is good, because it provides the best implementation of authentication available for the mass market and bad, because you will need a separate DCE installation (with all the maintenance and licensing costs).

5.2.2.6 How does OrbixSecurity work and how and what component of DCE needs to be installed?

George Wolke (July, 1998): We were told that we needed to purchase and install DCE software (only Gradient Technology's solution is currently supported for NT and I don't know what is required for HP) BEFORE we could install the Orbix code. The DCE code is about 3K for the server side and $100 for each client.

Also, OrbixSecurity uses OrbixFilters. This is a potential problem if you use filters in your system since you cannot define the location of the security related filters within your filter chain.

5.2.2.7 △ Can we use Orbix security to provide Access control at Object instance level?

5.2.2.8 Authentication Security Exception

Extended Question: When I want the Server to authenticate it self to the Client, or when I want to put a secure invocation policy on the Server, then I get an error message telling me ”Authentication Security Exception”, ”Target is not secure”. The opposite is no problem, the Client can authenticate itself to the server.

Is there anybody who have came across this problem, or who have suggestions on how I should solve it?

Gregg Tally (July, 1998): The error message appears to indicate that the principal for orbixd, <host>/IT_daemon, does not support authentication when acting as the target. Have you used getnvit to see if <host>/IT_daemon supports the authentication policy and QoP required by your server and client principals? If not, you can use setnvtit on <host>/IT_daemon so that orbixd will support authentication when acting as a target.

5.3 VisiBroker

5.3.1 △ Does anyone has experience on implementing system access control and security service using VisiBroker for Java?

5.4 omniORB[55]

5.4.1 △ If there is a security service supported by omniORB and if not are there any plans to create one?

5.5 Intraverse

5.5.1 △ Has anybody integrated DASCOM's Intraverse and Entrust (PKI), and Iona's OrbixWeb?

Extended Question "Neil Crago" <Neil.Crago@btinternet.com> (March, 1999) : Has anybody integrated DASCOM Intraverse and Entrust (PKI), and Iona's OrbixWeb, to provide secure user authentication, over-the-wire security etc., and if so can you tell me whether you were using these products to handle CORBA Level 1 or 2 security and roughly how intrusive / practical these products really are? What level of support you got from Iona, DASCOM and Entrust?

6 Applying CPRBASEC

6.1 How do I secure a Naming Service?

Nick Battle (August, 1998)[56]: Naively you have to use a CORBA product with the Security service implemented.

Looking at the problem a bit more closely, there can be some nasty subtleties depending on what you are actually trying to secure. If you haven't already done so, you should read the recent thread in this [ed: comp.object.corba] group titled "Naming Service and SSL".

It is fair enough to want only "authorized clients" to write to a naming service, but (potentially) any object that wishes to export its object reference to a name server has to have some write permission, and also the permission to remove its own reference. It may be possible to give everyone the ability to export objrefs, but only privileged clients power to change the structure of the namespace - I'm affraid I don't know the name server interface well enough to say, but CORBA Security is capable of distinguishing different operations on an interface.

There can be bootstrap problems with secure name services in some CORBA Security systems. But these occur when you want to treat the name service as a source of TRUSTWORTHY object references. Basically, to bootstrap CORBA Security you may need to obtain some trustworthy object references (for example of Authentication

[55] http://www.orl.co.uk/omniORB/

[56] Message-Id: 35C8199D.B2073AD@x400.icl.co.uk

and Privilege services) BEFORE you can talk securely to anyone, including the name service. I believe this situation is different for simple public key based systems (such as CORBA SSL) that can bootstrap without external object support.

The subtlety in the other thread is a consequence of the fact that CORBA Security does not guarantee that the security name embedded in an object reference is the CORRECT security name. Just because you secure access to the name server does not necessarily mean that clients are not exporting object references with false information in them. Changing the security name in an objref would allow an attacker to masquerade as the genuine object, even though the security service is active (including the EstablishTrustInTarget feature!).

To overcome this with a secure name service, you would need very application/system specific access controls (that look at message content) or you would have to trust your (exporting) clients. CORBA Security (appendix D.6.2 of the latest specification) says that end users (clients) of object references are responsible for verifying their security names unless they have a trusted source of objrefs. I'm just pointing out that putting access control and message protection on a name server doesn't necessarily provide a trusted source of objref CONTENTS.

It is sometimes perfectly possible for clients to know and check the security names of objects. For example in DAIS Security the bootstrap problem mentioned above is avoided because the authentication client can predict the correct security name of the appropriate Authentication and Privilege services for its particular domain. However, in general, it is very difficult for clients to know the correct name – and in the case of security UNAWARE applications they are not supposed to know anything about security anyway! This is also discussed (and solutions proposed) in the other thread.

6.2 △ How can security-aware applications apply confidentiality and integrity to data (e.g. electronic documents)?

6.3 △ Is it possible to specify the data to be protected as a parameter to the interface, or as data protection service?

7 Related Security Technologies

7.1 SESAME

7.1.1 What is SESAME?

Linda Gricius (March, 1998):

SESAME (a Secure European System for Applications in a Multi-vendor Environment) is a research and development project, partly funded by the European Commission under its RACE program. It is also the name of the technology that came out of that project.

SESAME is a construction kit – it is a set of security infrastructure components for product developers. In a nutshell, SESAME:

- supports single sign-on to the network;

- provides role based distributed access control using digitally signed Privilege Attribute Certificates, with optional controlled delegation of access rights;
- supports full cryptographic protection of exchanges between users and remote applications;
- supports multiple domain operation with different security policies;
- can be scaled to operate over very large networks through its use of public key technology;
- builds on work done in international standards - it is an Open Systems solution;
- uses the widely accepted Generic Security Service API (GSS-API) - the SESAME user gets mechanism transparency.

7.1.2 How does SESAME work?

Linda Gricius (March, 1998):

This is what happens:

To access the distributed system, a user first authenticates to an Authentication Server to get a cryptographically protected token used to prove his or her identity. The user then presents the token to a Privilege Attribute Server to obtain a guaranteed set of access rights contained in a Privilege Attribute Certificate (or PAC). The PAC is a specific form of Access Control Certificate that conforms to ECMA and ISO/ITU-T standards. The promulgation, protection and use of PACs are central features of the SESAME design.

The PAC is presented by the user to a target application whenever access to a protected resource is needed. The target application makes an access control decision according to the user's security attributes from the PAC, and other access control information (for example an Access Control List) attached to the controlled resource. A PAC can be used more than once at more than one target application. It is digitally signed to prevent it being undetectably tampered with.

The PAC is cryptographically linked with the authenticated user to which it was issued, to prevent anyone other than the original owner (or one of their delegates) from using it. To provide this protection SESAME needs to establish temporary secret cryptographic keys shared pairwise between the participants. Kerberos key distribution protocols can be used for dialog key establishment, but they can also be either supplemented, or where appropriate completely replaced by public key technology. SESAME also supports Certification Authorities, X.509 Directory user certificates, following ISO/ITU-T standards.

User data passed in a dialogue between a client and a server can optionally be either integrity protected or confidentiality protected or both, using specially created Dialog Keys.

7.1.3 How does SESAME relate to Kerberos?

Linda Gricius (April, 1998):

Similar work, aimed specifically at UNIX systems, has been done by the Massachusetts Institute of Technology which has developed a basic distributed single sign-on technology called Kerberos. Kerberos has been proposed as an Internet standard (RFC1510).

In the light of this work, the SESAME project decided that in its early implementation some of the SESAME components would be accessible through the Kerberos V5 protocol (as specified in RFC1510), and would use Kerberos data structures, as well as new SESAME ones. This has shown unequivocally that a product quality approach reusing selected parts of the Kerberos specification is workable and that a world standard is possible incorporating features of both technologies. SESAME extends Kerberos in the following ways:

- It introduces user privilege attributes, contained in a digitally signed Privilege Attribute certificate (PAC) and issued by a Privilege Attribute Service (PAS). This enables users to carry various identities and privileges (groups, roles and any locally defined attribute types) rather than a simple ”name” as Kerberos provides.
- SESAME also uses public keys, optionally, in the formation of associations between clients and targets. Kerberos uses secret key technology only.
- SESAME has controlled delegation of privileges (PACs), so that targets can proxy their client’s privileges to call other services on their behalf (rather than calling them as themselves). Kerberos has only uncontrolled delegation, and SPKM has no delegation.

Regardless of the security mechanism used, the DAIS Security service accesses the mechanism via a generic API, called the Generic Security Services API or GSS-API. This is a standard API that presents the same interface to the caller, regardless of the mechanism underneath being used to implement the functions.

7.1.4 How does SESAME relate to the CORBA Security service?

Linda Gricius (March, 1998):

SESAME’s origins lie in the Open Systems Standards work of ECMA, the European Computer Manufacturers Association. In 1987 ECMA started its work on Security in Open Systems. It was here that the early ideas that are the basis of SESAME were formed. Following that first meeting, experts from all of the major computer manufacturers have at different times been involved in this work.

The Common Secure Interoperability (CSI) specification, which is part of the CORBASEC specification, defines the standards for common secure interoperability when using GIOP/IIOP. A part of this standard is the use of defined protocols. CSI-ECMA is one of these protocols.

7.1.5 How do I find more information about SESAME?

Linda Gricius (March, 1998): The SESAME V4 Overview can be found via the web at http://www.esat.uleuven.ac.be/cosic.sesame.html.

Konstantin Beznosov (October, 1998): http://www.esat.kuleuven.ac.be/cosic/sesame.

7.1.6 △ How do I to get SESAME API?

7.2 GSS-API

7.2.1 What is GSS-API?

Linda Gricius (April, 1998):

Another important development in the field of Open distributed system security has been the Generic Security Services Application Program Interface (GSS-API). This interface hides from its callers the details of the specific underlying security mechanism, leading to better application portability, and moving generally in the direction of a better interworking capability. The GSS-API is independent of communications, and only produces opaque tokens that must be transported separately (by any convenient scheme, such as SECIOP/GIOP).

A GSS-API implementation is viable across virtually any communications method. GSS-API is an Internet and X/Open standard. SESAME is accessed through the GSS-API, extended to support features needed to provide distributed Access Control.

7.2.2 △ How do I to get GSS API ?

7.3 Kerberos

7.4 DCE Security

7.5 SSL

7.5.1 Where can I find more about SSL?

Jeff Calog (December, 1997) :

- http://digitalid.verisign.com/crp_intr.htm,
- http://digitalid.verisign.com/id_intro.htm,
- http://www.rsa.com/rsalabs/newfaq,
- http://www.consensus.com/security/ssl-talk-faq.html,
- http://developer.netscape.com/one/security/index.html,
- http://search.netscape.com/newsref/ref/netscape-security.html.

7.5.2 Have the OMG specified SSL in any standard yet?

Andreas Vogel (October, 1997) : Yes, it has been adopted earlier this summer.

7.5.3 Where can I find the specification of IIOP over SSL?

Konstantin Beznosov: (May, 1999) Section 14 “Integrating SSL with CORBA Security” of CORBASEC specification describes how SSL is supposed to be integrated with CORBA Security service.

7.5.4 Does anybody know a ORB vendor who provides a SSL functionality with their product?

Tom Damiano (May, 1998): Iona Technologies supports SSL in their OTM product.

Rajeev Kumar Gupta (May, 1998): IONA provides SSL support for Orbix.

William Edwards (May, 1998): Visibroker has SSL support for both C++ and Java. There's a brief description of this on both the Visibroker for C++ and Visibroker for Java pages at: http://www.inprise.com/visibroker/products/.

Jose Ignacio Gijon (May, 1998): Mico (http://diamant-atm.vsb.cs.uni-frankfurt.de/ mico/) support secure communication and authentication using SSL. Mico use SSLeay (http://www.psy.uq.oz.au/ ftp/Crypto/) to provide the secure socket layer. And both are free.

Marc Laukien (July, 1998): The ORBacus SSL plug-in provides the necessary tools to develop and deploy secure C++ and Java CORBA applications using the Secure Sockets Layer (SSL) protocol. New applications can be written to take full advantage of SSL, and only a few minor modifications are necessary to enable security in existing ORBacus C++ and Java applications.

The ORBacus Open Communications Interface (OCI) provides the architectural framework that enabled the ORBacus SSL plug-in to be developed without any changes to the ORB core. The ORBacus SSL plug-in replaces the IIOP protocol with the SSLIOP protocol defined as part of the OMG security specification. This protocol is simply IIOP over a secure SSL channel. Applications can be SSL-enabled through the addition of a few simple API calls. The plug-in provides you with full control over the cipher suites used for client/server connection establishment. Furthermore, your applications can use any combination of secure and insecure connections, all under the complete control of the developer.

Other features of ORBacus SSL include:

- FREE for non-commercial use (see the ORBacus SSL Royalty-Free Public License Agreement for details)
- Available with COMPLETE SOURCE CODE
- Complete support for C++ and Java
- Uses OMG Security Service profile tags for compatibility with other SSL implementations
- Implemented as an ORBacus Open Communications Interface (OCI) plug-in
- Supported key exchange algorithms:
  - Diffie-Hellman and RSA
  - Symmetric encryption algorithms: RC4(40/128 bit), RC2(40 bit), IDEA(128 bit), DES (40/56/168 bit)
  - MAC calculation algorithms: MD5 and SHA
- User's manual and HTML API reference documentation
- Example programs

ORBacus SSL has been tested and is known to work on the following platforms:

- ORBacus SSL for C++:
  - SGI C++ 7.1 SGI Irix 6.2 or 6.3
  - SGI C++ 7.2 SGI Irix 6.2 or 6.3
  - SUN C++ 4.1 and 4.2 SUN Solaris 2.5
  - HP C++ A.01.12 HP-UX B.10.20
  - GNU C++ 2.7.2 Intel- or Sparc-based OS
  - EGCS C++ 1.0.x / GNU C++ 2.8.x Any supported OS
- ORBacus SSL for Java:
  - SUN's JDK 1.1.x or compatible

The ORBacus SSL 1.0 Preview Release, including all sources, can be downloaded from http://www.ooc.com/ssl/download.html. Should you have any problems compiling the sources, do not hesitate to ask us for assistance. Just send email to support@ooc.com.

7.5.5 △ Is there a free implementation of CORBA SSL service that will work with VisiBroker 3.* for Java?

7.5.6 If I use naming service and VisiBroker, can I cooperate SSL into the system?

Andre Srinivasan (July, 1998)[57]: The IOR determines whether an SSL connection is attempted.

You can enable SSL with any VisiBroker server by initializing the BOA with SSLTPool (if you pass the command line args into BOA_init, -OAid SSLTPool will do the trick) and initializing the SSL layer with a digital identity (you can do this with initializers to avoid modifying the code). The resulting IOR will contain an SSL component.

7.5.7 How easy it is to use the Visibroker SSL pack with a Java application for the developer as well as the user?

Ted Gamester (August, 1998)[58]: From the object implementers perspective and the client programmer using the Visi SSL pack is simple, just add a few standardized lines of code on the server side and client side and you are in business. Sample code is including with the doc.

7.5.8 Is there an additional client side piece that must be installed in order to use the Visibroker SSL pack with a Java application?

Ted Gamester (August, 1998)[59]: Yes, you must run an installer on the client as the VisiSSL runtime currently uses native code. (100% pure version is coming).

[57] Message-Id: qy4svyqbgp.fsf@ahi.visigenic.com
[58] Message-Id: 35C5EE7E.E0647FDC@umich.edu
[59] Message-Id: 35C5EE7E.E0647FDC@umich.edu

7.5.9 Between what parties does authentication happen when the client and the server communicate over SSL via Visibroker's Gatekeeper?

Andreas Vogel (October, 1998)[60] : Only the gatekeeper will be authenticated via an X.509 certificate. A client could be, if the gatekeeper demands so. When using SSL on the backend the server, the gatekeeper is authenticated. Authentication is always on a OA base, not an object base.

7.5.10 Do any third-party companies have SSL security systems that can be incorporated into either Orbix or Visibroker?

Andreas Vogel (October, 1997): It's a ready there. The SSL implementation [ed: in Visbroker] is third-party anyway.

7.5.11 Does SSL raise any firewall problems when accessing from the outside internet?

Andreas Vogel (March, 1998): Nothing beyond the usual. In fact things are even better, you can run a SSL-enabled gatekeeper on port 443 which should get you through most firewalls.

7.5.12 Do SSL security implementations with CORBA solve or change the problem of securely linking an object reference to the principal that it represents?

Extended Question Nick Battle (July, 1998)[61]: I'm talking about the issue of "Target Object Identities" in CORBASEC appendix E.6.2 [ed: D.6.2 in the current version of the spec] (at least it used to be in the July 97 spec I have to hand!).

The problem is that IORs are not self securing. A security name (which I guess is the certificate name in SSL?) is included as a hint in the IOR (true?), but the system doesn't guarantee that the IOR hasn't been tampered with. So an attacker can tweak an object reference to use the security name of his own principal - something that he has the keys for - and then clients using that object reference think they are operating securely (which in a sense they are!), but are actually talking to the wrong object. All the ORB guarantees is that you are talking securely to the security name in the IOR, not that the name in the IOR is the one you want to talk to.

I understand how this works with the Kerberos and CSI-ECMA mechanisms, but I'm not sure how/if it's different with SSL. I see the problem as fundamental (there's no provable link between an object reference and the security name), so there must be an equivalent weakness in SSL based CORBA systems, true?

As E.6.2 [ed: D.6.2 in the current version of the spec] says, the onus is on the client to check the security name(s) of any target it intends to use, unless it has a trusted source of object references.

[60] Message-Id: 199810141501.IAA21711@sparky.qds.com
[61] Message-Id: 35C579BF.3E3AE0D9@x400.icl.co.uk

Andre Srinivasan (July, 1998)[62]: Currently there are no SSL hints beyond the SSL component in the IOR.

The addition of this check [ed: by client of the security name(s) of any target it intends to use] is pretty straight forward though and the ORB could do it for you. If a communicating party presents something like a wallet that not only contains credentials, but contains trustpoints (i.e. trusted certificates that aren't necessarily roots), the ORB can reject the connection if the target identity does not contain a trustpoint.

I believe that a security unaware application CAN operate in an environment where the source of object references is not necessarily trusted IF a wallet can be administered that contains trustpoints. Basically the wallet notion moves the source of trust to the wallet rather than relying on the source of object references.

7.5.13 What SSL implementations are known to [not] interoperate?

Dave Sames (January, 1999)[63] : We were just trying to get a VB for Java 3.2 SSL server to work with an Orbix 2.3c SSL client. We found that the IOR produced by the VB server contains 1 profile, per the 1.1 IIOP spec. Fortunately, the Orbix client is able to interpret this correctly, and invoke operations on a CORBA object in the VB server. However, when the reverse configuration is attempted, the Orbix server produces a 2-profile IOR - 1.0 IIOP spec compatible - and the VB client cannot interpret it correctly to set up the SSL connection. Iona noted that their 3.3 release will support both 1.0 and 1.1 IIOP specifications.

Polar Humenn (January, 1999)[64] : I think Jishnu [Mukerji] told me at one point that SSL wasn't in IIOP 1.0 anyway. So the SSL "Component" made into a profile it is just an interpretation to get get it to work. What should be support[ed] is the 1.1 IIOP specification. However, this still needs work. Hopefully that will be cleared up with submissions to the new CSIv2 RFP.

7.5.14 △ Does the SSL-certificate certify the server or the object?

7.5.15 What is the normal way of asserting that unauthorized clients cannot connect to an object that an authenticated client is using?

Mike (March, 1999)[65] : SSL authenticates a user on a per session basis, so it does not restrict access on a per-object basis. You'd have to add in code to your Factory object to prevent users from accessing certain objects. Shouldn't be too difficult, and is pretty secure since you've authenticated the user.

[62]Message-Id: qy1zqyqbam.fsf@ahi.visigenic.com
[63]Message-Id: 3.0.3.32.19990118102117.0098c140@pop.hq.tis.com
[64]Message-Id: Pine.LNX.3.96.990120115306.2920D-100000@marcy.adiron.com
[65]Message-Id: 36EF68C7.C51979D8@there.com