CORBA وبررسی امنیت در شبکه

شاداب دهقان دانشگاه شهریار – شهر قدس

سرکار خانم مهندس حسینی

چکیده : این مقاله در ارتباط با CORBA و بررسی کلیهٔ مواردی که در ارتباط با امنیت برقرار کردن آن است بحث شده و در انتها کلیهٔ مواردی که برتری CORBA را نشان می دهد آمده است .

کلمات کلیدی : سیستم های توزیع شدهٔ ناهمگون ، CORBA، OMG، ORB، Cross Plat Form ، گذرگاه شی، SSL، IIDP، اعطای مجوز یکپارچگی داده ها ،کرم رد ، تجسم شئی، خادم، رابطه شئی، قابل حمل، PDA،TCP،IDL،GLOP

CORBA و بررسی امنیت در CORBA (5)

CORBA[1]

در سال ۱۹۸۹ ، گروه OMG[2] به منظور بررسی مشکلات ایجاد سیستمهای توزیع شده ناهمگون قابل حمل ، شکل گرفت .

در جهان امروزی CORBA سر دسته سیستمهای توزیعی است و راه حلی است برای مشکلات سیستمهای ناهمگون از جنبه های سخت افزاری و نرم افزاری بتوانند در جوار یکدیگر عمل نمایند .

CORBA یک گذرگاه نرم افزاری بنام Object Request Broker (ORB) را معرفی می کند که این قدرت را می دهد تا ارتباط cross platform (ویژگی یک کاربرد نرم افزاری یا یک وسیله نرم افزاری که قابلیت اجرا شدن روی بیش از یک زیر بنای سیستم را داشته باشد) در میان اشیاء توزیع شده و برنامه های کاربر وجود داشته باشد .

(2)

ORB یک میان افزار است که ارتباطات clint-server بین اشیاء را برقرار می کند . در واقع ORB یک محیط میانی برای همکاری اجزاء در یک محیط توزیع شده بوجود می آورد .

مشخصات CORBA

۱) بسیاری از زبانهای موجود را پشتیبانی می کند .

۲) هم توزیع و هم شیء گرایی را پشتیبانی می نماید .

۳) CORBA یک استاندارد صنعتی است که باعث رقابت بین شرکت های تجاری می شود و همچنین اطمینان می دهد که یک پیاده سازی همراه با کیفیت وجود دارد .

٤) CORBA دارای درجه بالایی از interoperability (اشاره به مولفه های از سیستم های کامپیوتری دارد که می تواند در محیط های گوناگون عمل کند) است . که ضمانت می کند اشیاء توزیع شده ای که روی محصولات CORBA ساخته شده اند می توانند با هم ارتباط برقرار کنند .

CORBA دارای رقبای زیادی می باشد ، رقبایی چون :

DEC,DCOM,RPC,Shared memory based interaction,Named Pipe communication و رقیب مهم DCOM. CORBA است که به وسیله شرکت مایکروسافت ایجاد و خلق شده ، وضعیت محصولات مایکروسافت در بازارهای تجاری DCOM را یک رقیب جدی و واقعی برایCORBA کرده است .

(1)

امنیت (Security)

(2) امنیت برای سیستم های کامپیوتری مدرن امری ضروریست ، بخصوص برای سیستم های توزیع ، که نسبت به سیستم های قدیمی نفوذ پذیرتر می باشند . زیرا مکانهای بیشتری برای حمله کردن به آنها وجود دارد. بنابراین سیستم های CORBA نیز که طبیعت توزیعی دارند به امنیت نیاز بیشتری دارند.

امنیت چیست؟

بطور کلی امنیت، یک سیستم اطلاعاتی را از دسترسی غیر مجاز و نفوذ و تداخل در عملکرد هایش محافظت می نماید. بطور دقیق ترامنیت شامل موارد زیر می باشد:

(4)

(1)

علی رغم امتیازات ذکر شده فوق، SSL / CORBA هنوز احتیاجات امنیتی authorization (اعطای مجوز) را برای کاربر برآورده نمی کند.

با توجه با مسئله ذکر شده بر (3) لهایی که در آنها به موازات تأیید اعتبار، یکپارچگی داده ها و Privacy، اعطای مجوز کاربر (Authorization) نیز لازم است، سرویس امنیتی (CORBA Security Service) به بازار عرضه شد.

اهداف سرویس امنیتی CORBA:

همان اصولی که در مورد سیستم های امن ذکر شده است، اساس سرویس امنیتی CORBA را نیز تشکیل می دهند.

اگر چه سیستم های CORBA بطور کلاسیک سیستم اطلاعاتی نیستند اما طبیعت ذاتی آنها منجر به ایجاد تهدیدات بالقوه می شود که ممکن است در معماری آنها لحاظ نشده باشد بنابراین بعضی از اهداف امنیتی، مختص امنیت CORBA می باشد. این اهداف شامل موارد زیر است:

- فراهم نمودن امنیت از طریق سیستم ناهمگون (Heterogenous)
- شی گرایی
- فراخوانی های امن: اطمینان بخشی از حفظ امنیت فراخوانی ها
- کنترل و نظارت (Control & Auditing): اطمینان بهشی از اینکه نظارت و کنترل بروی دسترسی و فراخوانی های Object ها انجام می گیرد.

سرویس امنیتی CORBA

سرویس امنیتی CORBA یک مشخصه مؤثر و با حجم بالا است که قبل از تصویب آن در سال ۱۹۹۶ بیش از دو سال تلاش برای به ثمر رساند ام شده است. بعد از تصویب آن تقریباً دو سال دیگر (2) برای اولین پیاده سازی آن وقت صرف شد تا محصول مورد نظر به بازار آید. علت زمان طولانی برای ورود آن به بازار دو مسئله بود اول آن که امنیت توزیع شده یم مسئله پیچیده و مشکل است و دوم اینکه فضایی که بوسیله ویژگی CORBA فراهم می شود معمولاً تنها در سیستمهای بسیار بزرگ مورد نیاز است که تا مین زمانهای اخیر تعداد خیلی کمی از آنها بر پایه CORBA عمل می کردند. با فراگیر شدن CORBA بعنوان یک تکنولوژی غالب در سیستم های بزرگ تجاری، تقاضا برای یک راه حل امنیتی قوی نیز افزایش یافت.

ویژگی های سرویس امنیتی CORBA:

انجام رساندن این هدف، پروتکلی خاصی بنام (SECIOP) ure Inter- ORB Protocol (4) معرف نموده است.

قابلیت جایگزینی Replaceability: هر زمان که ممکن باشد، سرویس امنیتی CORBA اجازه جایگزینی سرویسهای امنیتی مانند سرویسهای تأیید اعتبار، auditing و غیره را می دهد.

رابطهای اشیاء قابل حمل

اشیاء CORBA توسط توابع برنامه نویسی اطلاعات، پیاده سازی شده و ارائه می گردند. موجودیتهای زبان برنامه نویسی که پیاده سازی را ارائه می دهند خادم می نامند. از آنجائیکه برای هر شیء CORBA باید یک خادم وجود داشته باشد، آنها را تجسم شیء CORBA نیز می نامند.

در محیط CORBA، رابطهای شیء، دنیای اشیاء CORBA را به دنیای خادمهای اشیاء متصل می

همانطور که شکل بالا نشان می دهد، رابطه های شیء ارتباط بین اشیاء CORBA و دنیای خادمهای اشیاء هستند. رابطهای اشیاء سرویسهایی را ارائه می دهند که توسط آنها بتوان اشیاء را ایجاد نموده و درخواستها را به آنان ارسال نمود. رابط شیء استاندارد امروزه POA ها هستند که به خادمهای زبانهای برنامه نویسی متفاوت امکان می دهند که توسط ORB های مختلف پشتیبانی شوند. همانطوریکه در شکل نشان داده شده است، تا زمانیکه ORB فرمان گوش دادن درخواست توسط سرور را اعلام نکند، سرور آنها را قبول نخواهد کرد. بعلاوه از آنجائیکه یک کاربرد سرور ممکن است توسط چندین POA مورد درخواست قرار گیرد، جریان درخواستها توسط POAهای مختلف توسط مدیر POA کنترل خواهد شد. علاوه بر کنترل جر (2) راستها در POA، مدیر POA می تواند صدرخواستها را تشکیل داده یا آنها را مرجوع نماید.

پروتکلهای درون ORB

تا قبل از CORBA 2.0، نیاز زیادی برای تعریف یک پروتکل استانداردی که بتواند امکان ارتباط برنامه های کاربردی را با یکدیگر ایجاد کند ضروری به نظر می رسید. تا آن زمان هر فروشنده ORB پروتکل شبکه خود را بکار می برد و این امر باعث می شد تا جزایر برنامه های کاربردی ORB ایجاد شود، بطوریکه ORB ها نمی توانستند با هم ارتباط بر قرار کنند. CORBA 2.0 معماری عمومی را برای قابلیت میان عملیاتی ORB ارائه داد که آنرا پروتکل عمومی درون ORB یا اختصار GIOP [1] نامیدند. GIOP پروتکلی است که قواعد انتقال و فرمت پیامها را به نحوی که مستقل از ORB باشند، تعریف می کند. IIOP [2] یا پروتکل درون ORB اینترنت، مشخص می کند که چگونه GIOP روی پروتکل TCP/IP پیاده سازی می شود. با استفاده از IIOP یک CORBA ORB از یک فروشنده می تواند با ORB از فروشنده دیگر ارتباط برقرار کند. محصولاتی با CORBA سازگار هستند که با IIOP پیاده سازی شده باشند. در چنین حالتی IIOP قابلیت میان عملیاتی بین محصولات CORBA را تعیین می کند. پروتکلهای دیگری برای ارتباط بین ORBها وجود دارد، اما IIOP به خاطر استاندارد بودن و نیز پروتکل انتقال TCP/IP سریعتر از بقیه همه گیر گشت.

قابلیت میان عملیاتی ORB نیاز دارد که فرمت مرجع اشیاء استاندارد شود. مراجع شیء شامل اطلاعاتی است که یک ORB برا ایجاد ارتباط بین مشتری و شیء هدف به آنها نیاز دارد. فرمت استاندارد مرجع شیء را مرجع شیء قابل میان عملیات یا IOP می نامند. یک IOP شامل لیست پروتکلهای پشتیبانی شده و اطلاعاتی راجع به آنهاست. در ادامه راجع به مرجع شیء قابل میان عملیات به بحث خواهیم پرداخت.

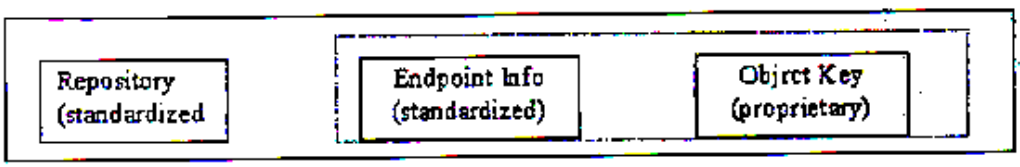
مرجع شیء با قابلیت عملیاتی

مراجع اشیاء تنها راه دستیابی به شیء هدف هستند. یک مشتری بدون داشتن مرجع شیء نمی تواند با آن ارتباط برقرار کند. یک سرور می تواند به طرق زیر مراجع اشیاء را فراهم کند:

(4)

(3)

مرجع را تحت یکی از سرویسهای شناخته شده مانند سرویس نام گذاری یا سرویس معاملات از گرداند.
مرجع شیء را با استفاده از مکانیسمهایی مثل پست الکترونیکی یا صفات Web در اختیار قرار دهد.
رایجترین روش برای مرجع مشتری از طریق فراخوانی یک عملگر می باشد. **محتویات مرجع شیء**
سه بخش اصلی اطلاعات یک IOP همچنان که در شکل زیر نشان داده شده، عبارتند از:



شکل ۲- محتوای مرجع شکل

Repository ID

CORBA یک مخزن واسطه فراهم می کند که امکان می دهد تا در زمان اجرا بتوان به تعاریف IDL

۱- جریان درخواست مطابق مراحل زیر از مشتری آغاز شده، ORB را پشت سر گذاشته و به سوی سرور ادامه می یابد: واسطه فراخوانی پویا[1] بفرستد. در استفاده از DII می توان به عضوی دست یافت که اطلاعات از آن در زمان کامپایل در دست نبوده است. در چنین فراخوانی، مشتری IOR هدف، نام صفات/ عملگرها را مشخص نموده و پارامترها را مر فرستد. سرویسهای IR برای دستیابی به اطلاعات در مورد عملگرها و نوع پارامترهای ارسالی استفاده می شوند.

۲- ORB مشتری درخواست را به ORB سرور عبور می دهد.

۳- ORB سرور درخواست را به رابط شیء می فرستد شیء هدف است، می فرستد.

٤- رابط شیء درخواست را به رابط شیء می فرستد تا شیء هدف اجاد شود.

مشابه مشتری، سرور هم حق انتخاب استفاده از روشهای ایستا و پویا را در انتخاب خادم دارد، به عبارت دیگر می توند از Skeleton ایستا و پویا استفاده کند.

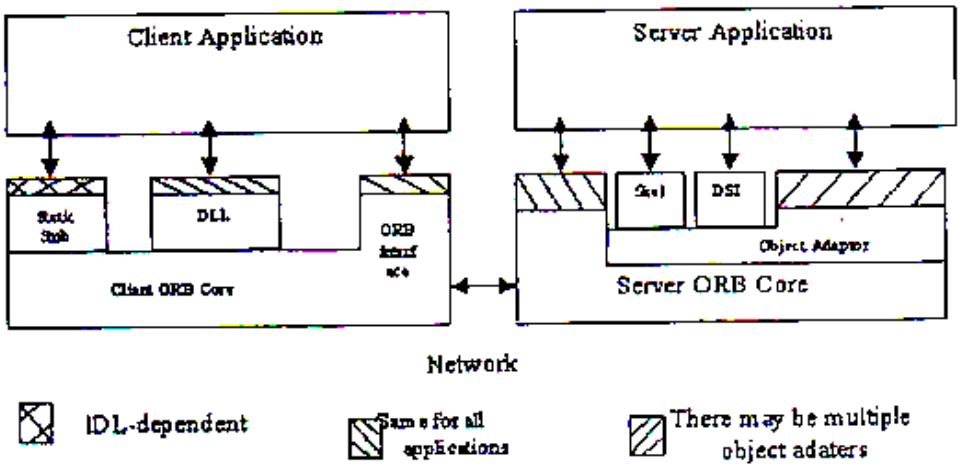
٥- بعد از انتقال درخواست به خادم، پاسخ به مشتری برگردانده خواهد شد.

CORBA چندین قالب از درخواست ها را پشتیبانی می کند که عبارتند از:

مشتری می تواند در خواست همزمان با تأخیر[2] بفرستد، در این مورد مشتری در مدت زمان انتظار پاسخ، به کار خود ادامه خواهد داد.

CORBA همچنین امکان درخواستهای یک طرفه[3] را نیز ایجاد می کند. در این روش هم مانند روش همزمان، مشتری بلوکه می شود تا پاسخ درخواست او باز گردد.

همچنین CORBA می تواند درخواستهای غیر همزمان[4] را نیز پشتیبانی نماید، در این روش مشتری و سرور می توانند هر زمان که لازم باشد با هم ارتباط بر قرار می کنند.



شکل ۳- جریان عمومی درخواست در CORBA

فراخوانی درخواست

مشتری ها، اشیاء را با فراستادن پیامها بکار می گیرند. در زمان فراخوانی OBR توسط مشتری، وی پیامی را برای شیء می فرستد. برای فرستادن پیام به یک شیء، مرجع آن شیء توسط OBR نگه داشته خواهد شد. همانطور که گفته شد، مرجع شیء شناسه یکتایی است که برای شناساندن شیء هدف بکار رفته و اطلاعات رد نیاز OBR را برای فرستادن پیام به مقصد مورد نظر را شامل می شود. زمانی که یک م (5) ملگری را از طریق یک مرجع شیء فراخوانی می کند، OBR کارهای زیر را انجام می دهد:

- مکان شیء هدف را می یابد.
- اگر سرور در حال اجرا نباشد، آنرا فعال می کند.
- کلیه آرگومنهای فراخوانی را به شیء منتقل می کند.
- اگر لازم باشد خادم شیء را فعال می کند.
- منتظر می ماند تا درخواست کامل شود.

(3)

(6)

مشتری اطلاعی از سیستم عامل ماشین سرور ندارد.

استقلال پروتکل مشتری

مشتری نمی داند که پروتکل ارتباطی برای فرستادن پیامها چیست. اگر چندین پروتکل ارتباطی با سرور وجود داشته باشد، خود OBR یکی را انتخاب می کند.

استقلال انتقال

مشتری نحوه انتقال و لایه ارتباط داده ها را که برای انتقال پیامها بکار می رود نمی بیند. OBR می تواند تکنیکهای مختلف شبکه مثل اتنت ، ATM، Token ring و خطوط سریال را بکار ببرد.

سرویسها و امکانات CORBA:

همانطور که در قبل گفته شد، OBR به عضو اجازه می دهد تا با سایر ارتباط برقرار کند. از آنجائیکه هر کاربردی که اجرا می شود به OBR وابسته است، OBR برای ایجاد کاربردهای توزیع شده، ضروری است. متأسفانه OBR، به تنهایی برای ایجاد کاربردهای توزیع شده کافی نیست. برای این کار سرویسهای مدیریتی لازم است تا سلامت سیستم را در نظر داشته باشد و نیز به دامنه های مشخصی از واسطه ها و قالبهای کاری نیاز است تا توسعه سیستم ها را سرعت ببخشند.

سرویسهای مدیریت اشیاء به دو دسته سرویسهای افقی و عمودی تقسیم می شوند. سرویسهای افقی، سرویسهای عمومی هستند که وجود آنها در کلیه کاربردها به صورت عمومی مورد استفاده قرار می گیرد، و لیکن سرویسهای عمودی سرویسهایی هستند که به دامنه کاربرد وابسته هستند. در زیر در رابطه با این دو نوع سرویس صحبت خواهیم نمود.

سرویسهای اشیاء

سرویسهای شیء واسطه های افقی هستند از دامنه بوده و در بیشتر کاربردهای توزیع شده استفاده می شوند. از این سرویسهای در تهیه واسطه های پیاده سازی CORBA استفاده می شود. بدون استفاده از این سرویسها نوشتن یک کاربرد توزیع شده، کار آ[illegible]ت. سرویسهای شیء مجموعه ای هستند که سرویسهای CORBA نامیده می شوند.

سرویس نام گذاری اشیاء:

مشتری را قادر می سازد تا به IOR عضو دیگری در هر جایی از گذرگاه دستیابی نماید. همچنین به عضو اجازه می دهد تا نامش را به یک متن ناگذاری و این نام باید یکتا باشد.

سرویس اتفاقات:

اتفاقات غیر دوره ای را پشتیبانی می کند. به اعضاء اجازه می دهد تا بصورت پویا نفوذ خود را در یک اتفاق ثبت کنند. طراحی این سرویس قابل گسترش بوده و برای محیطهای توزیع شده سودمند است.

سرویس چرخه زندگی:

عملگرهای ایجاد (create) ، کپی (copy) ، حرکت (move) و حرکت دوباره (remove) اعضاء را در OBR تعریف می کند.

سرویس شیء دائمی[1]:

مجموعه ای از واسطه های عمومی برای ذخیره و مدیریت حالات دائمی اعضاء بوجود می آورد. خصوصیت اصلی POS، باز بودن آن است که اجازه می دهد انواع مختلفی از مشتریها و پیاده سازیهای POS با هم کار کنند.

منابع و ماخذ:

1)Formal Analysis of the CORBA $CSIV_2$ Security (1) polar Humenn,Susan Older

2)BOB Burt , Light Weigt Security Service For CORBA

3)Lona technology

4)Belind fair thorne , Bob Blakley , Introduction to CORBA Security

5)Shiu – Kai Chin , Using Security in CORBA Application