مروری بر معماری وب

تهیه کننده : امیر حسین شریفی

SSL/TLS

در اینجا فرض را بر آن گذاشتیم که خواننده عزیز با پروتکل HTTP و HTML آشنایی دارد و می خواهیم مختصر بسیار کوتاهی درباره SSL بیان کنیم.
یکی از استثناهای واضح و آشکاری که بیشتر برنامه های کاربردی تحت وب امروزه استفاده می کنند پروتکل لایه سوکتهای امن (Secure Sockets Layer) می باشد که در بالای HTTP قرار گرفته است. SSL در اصل برای رمزگذاری لایه انتقال ساخته شده است بنابراین یک میانجی بین مشتری و سرویس دهنده نمی تواند متن اصلی ردو بدل شده را به راحتی بخواند. می توان گفت که SSL به صورت یک لفافی برای HTTP ساخته شده است. SSL به صورت ذاتی پایه و اساس درخواست-پاسخ (Request-Response) پروتکل HTTP را تغییر نداده است. SSL برای امنیت برنامه های کاربردی هیچ کاری انجام نداده است بلکه فقط استراق سمع بین مشتری و سرویس دهنده را کمی مشکل تر کرده است. **گواهینامه سمت مشتری** یکی از خصوصیات اختیاری پروتکل SSL می باشد که پیاده سازی شده است. یعنی یک احراز هویت دو طرفه که باید انجام شود.(گواهینامه مشتری باید به عنوان یک هویت محرز شده توسط سرویس دهنده امضا شود) . اگر چه تعداد کمی از سایتهای روی اینترنت امروزه این کار را انجام می دهند.
نسخه قدیمی SSL ، لایه امنیت انتقال (Transport Layer Security) بود. SSL/TSL از طریق پروت 443 عمل می کنند.

مدیریت وضعیتها؛ Cookies

همان طور که می دانید در حقیقت HTTP یک پروتکل Stateless می باشد. یعنی هیچ وضعیتی از نشستها به وسیله خود پروتکل حمایت نمی شود. به عنوان مثال اگر شما برای درخواست منبعی که کرده اید یک پاسخ نامعتبری دریافت کنید دوباره اقدام به درخواست فوق می کنید اما سرویس دهنده این درخواست را به صورت کاملا جداگانه و واحد ملاحظه می کند. برای رفع این ضعف مکانیزمهایی وجود دارد که باعث می شود این پروتکل به صورت یک پروتکل Stateful عمل کند. یکی از مکانیسم هایی که امروزه به طور گسترده ای استفاده می شود Cookie ها می باشند که به عنوان بخشی از درخواست – پاسخ های HTTP بین سرور و کلاینت ردو بدل می شوند و باعث می شود که برنامه کاربردی و مشتری اینگونه فکر کنند که

آنها از طریق یک حوزه مجازی به هم متصل شده اند. (این مکانیسم به صورت کامل تری در RFC 2965 بیان شده است). کوکیها بهترین روشی ارتباطی برای این منظور می باشند برای این که هر کاربری به وسیله یک نشانه با یک سایت وب ارتباط برقرار کند و تا هر زمانی که این نشانه به همراه درخواست کاربر فرستاده می شود ، آن کاربر مجاز به استفاده از سایت مورد نظر می باشد.آنها می توانند هم در حافظه ذخیره شوند و هم می توانند به صورت پایدارتری در دیسک سخت ماندگار باشند. کوکیها هرگز بدون عیب و نقص نمی باشند (به خصوص هنگامی که به صورت ضعیفی ساخته می شوند) و استفاده از آنها پیامدهای زیادی را برای امنیت برنامه های کاربردی دارد . ولی در حال حاضر هیچ مکانیزم بهتر دیگری برای این مشکل وجود ندارد.

مشتری های (Clients) وب

برنامه کاربردی استاندارد برای مشتری ، مرورگر وب می باشد که از میان پروتکل HTTP ارتباط برقرار می کند و متون HTML ای که دریافت می کند را ترجمه کرده و نمایش می دهد. به صورت کلی HTML و HTTP مامور شده اند که داده هایی که به وسیله سرور پردازش شده اند برای مشتری وب نمایش داده شود .

مانند HTTP ، مرورگر وب نیز خیلی ساده به نظر می رسد. توسعه پذیری HTML این امکان را برای ما فراهم می کند که بتوانیم متون استاتیک و دینامیک را در آن بیامیزیم. از محتواهای فعال به کار رفته می توان ActiveX ها و Java را نام برد. گنجاندن یک ActivX در HTML را می توانید در ذیل مشاهده کنید:

```
<object id=”scr”
        classid=”clsid:09243BD5-48AA-11D2-092267C3FBC”>
</object>
```

در کلمات وب ، همه حروف در کد ASCII می باشد. وقتی یک مترجم مرورگر وب با یک برچسب object برخورد کرد متوجه می شود که باید آن را از یک سایت دور دانلود کند و یا از به صورت مستقیم از کامپیوتر محلی بارگذاری کند و سپس آن را اجرا کند. البته این ActivX در صورتی اجرا می شود که یا قبلا در کامپیوتر شما نصب شده باشد و یا احراز هویت شده باشد که این کار به وسیله Microsoft Authenticode انجام می شود. یعنی در حین اجرای آن یک صفحه باز شده و امضای دیجیتال شخص سازنده کد را نمایش می دهد و از شما سوال می کند که آیا فایل مورد نظر را اجرا کند یا خیر . اگر کاربر جواب مثبت دهد کد اجرا می شود. لازم به ذکر است که اجرای بسیاری از ActiveX ها می توان امنیت یک سیستم را به خطر بیاندازد. پس هیچگاه به اینگونه برنامه هایی که ناآشنا و نا مشخص می باشند اجازه اجرا شدن ندهید.

HTML یک زبان توانا می باشد اما محدودیت های زیادی دارد. بعد از سالها ، تکنولوژیهای جدیدی مانند HTML دینامیک و Style Sheet ها پدیدار شدند تا چاشنی برای نمایش محتویات صفحات وب باشند. اما تغییران بنیادی تری نیز در حال وقوع است. XML (eXtensible Markup Language) به آهستگی جایگزین HTML می شود.

بالاخره اینکه مرورگر می تواند با دیگر پروتکلها نیز ارتباط برقرا کند. به عنوان مثال می تواند به وسیله پروتکل SSL با یک سرور وب ارتباط داشته باشد. و همچنین می تواند با دیگر پروتکلها مانند FTP ارتباط برقرا کند. به درستی که مروگر وب یکی از بزرگترین سلاح های قابل دسترس برای نفوذگران وب می باشد.

سرور وب

سرور وب در تعریف به صورت یک سرویس دهنده HTTP می باشد که درخواست ها را برای منابع دریافت می کند و بعضی از تجزیه ها را روی آن انجام می دهد تا مطمئن شود که این منبع قابل دسترس می باشد یا خیر و سپس آن را برای پردازش ، تحویل برنامه کاربردی می دهد و وقتی که برنامه پاسخ را برگرداند ، سرویس HTTP آن را به مشتری تحویل می دهد.

امروزه سرور های محبوب قابل دسترس زیادی وجود دارد مانند IIS Apache Software Foundation ، سرور Apache HTTP (که به اختصار Apache گفته می شود) ، AOL/Netscaps Enterprise و Sun iPlant .

اگر چه سرورهای وب خیلی ساده به نظر می آید ، اما ما دوباره باید درباره سوراخهای امنیتی زیادی که در آنها وجود دارد و طی سالیان دراز کشف شده اند بحث کنیم. به طوری که سوراخهای امنیتی در سرور های وب یکی از برترین مسائل امنیتی می باشد که از سال ۱۹۹۰ به این طرف بیان شده است.

برنامه کاربردی وب

هسته اصلی سایتهای وب قسمت منطقی سمت سرور می باشد. (اگر چه منطق سمت مشتری هنوز با مرور گر وب آمیخته می باشد). این مدل به n-tire مشهور می باشد که به طور معمول شبیه یک سرور HTTP می باشد که به صورت دینامیک طراحی شده است و تقریبا به صورت یک برنامه کاربردی یکپارچه و Stateful که به کاربرها اجازه می دهد که با آن تعامل داشته باشند.

مفهوم n-tire یا n لایه ای برای فهمیدن و درک برنامه کاربردی مهم می باشد لایه برنامه کاربردی می تواند خودش شامل چندین لایه دیگر باشد. اما نمایش عمومی آن به صورت معماری ۳ لایه ای نمایش داده می شود که به نامهای **لایه نمایش**[1] ، **لایه منطقی**[2] **و لایه داده**[3]

1 - Presentation Layer

معروف می باشد که در شکل ۱ نمایش داده شده است. اجازه بدهید که آنها را به طور خلاصه بیان کنیم.

لایه نمایش تسهیلاتی را برای گرفتن ورودی ها و نمایش نتیجه فراهم می کند. لایه منطقی ، داده ها را از لایه نمایش دریافت کرده و کارهایی را روی آن انجام می دهد. هر زمان که نیاز به کمک لایه داده باشد، داده ها به لایه نمایش به عنوان نتیجه کار بر گردانده می شود. سرانجام لایه داده یک منبع ماندگاری از داده ها را فراهم می کند که می توان روی آنها جستجو انجام داد و هر دفعه به وسیله لایه منطقی به روز رسانی می شود. لایه داده این امکان را فراهم می کند که لایه منطقی بدون اینکه نیازی به کدهای مشکل برنامه نویسی داشته باشد بتواند آن را به آسانی به روز رسانی کند و از آن استفاده نماید.

برای اینکه بفهمید اینها چگونه با هم کار می کنند اجازه بدهید مثالی را بیان کنیم. یک برنامه کاربردی تحت وب ساده را فرض کنید که تمام فایلهای محلی درون هارد سرور را برای وجود متنی که کاربر درخواست کرده است ، جستجو می کند و نتیجه را نمایش می دهد. لایه نمایش شامل یک فرم با فیلدهایی برای دریافت متن ورودی توسط کاربر می باشد که این متن باید مورد جستجو قرار گیرد. لایه منطقی یک برنامه اجرایی می باشد که رشته ورودی را دریافت کرده و پس از اینکه مطمئن شد که این رشته حاوی کاراکترهای مخرب نمی باشد ، ارتباط دهنده مناسب به پایگاه داده را برای ایجاد یک ارتباط با لایه داده ، فراخوانی می کند. سرانجام یک Query به وسیله ورودی ساخته می شود. لایه داده ممکن است شامل پایگاه داده ای باشد که شاخصی از کلیه فایلهای درون ماشین محلی را در خود ذخیره کرده است و به صورت Real-Time به روز رسانی می شود. Query پایگاه داده مجموعه ای از رکوردها را انتخاب کرده و آنها را به لایه منطقی بر می گرداند. لایه منطقی رکوردهای برگشتی را ترجمه و تحلیل می کند و رکوردهایی که لازم نبوده اند را حذف کرده و رکوردهای درخواست شده را به لایه نمایش بر می گرداند. این رکوردها نیز در HTML آمیخته شده تا به صورت مناسبی برای کاربر نمایش داده شود. و از سرور وب به سمت مرورگر برگردانده می شود.

خیلی از تکنولوژیهای در حال حاضر ، به صورت واقعی و عملی از یک یا بیشتر این لایه ها استفاده می کنند بنابراین اغلب تشخیص دادن لایه ها از یکدیگر کمی مشکل می باشد و حتی بعضی لایه ها درون بعضی دیگر استفاده می شود. به عنوان مثال برنامه Active Server Page (ASP) به شما اجازه می دهد درون صفحات وب در لایه نمایش از کدهای لایه منطقی استفاده کنید.بنابراین نیازی نیست که یک کد اجرایی مجزا برای درخواستهایتان از پایگاه داده ، داشته باشید. (اگر چه خیلی از سایتها از COM object برای دسترسی به پایگاه داده استفاده می کنند و ممکن است که این کار در بعضی موارد ایمن تر باشد.)

2 - Logic Layer
3 - Data Layer

تکنیکهای متنوع زیادی برای ایجاد سایتهای وب چند لایه وجود دارد. بعضی از این تکنیکها در جدول شماره ۱ آمده است.

آنچه که لازم است درباره این تکنیکها بدانید این است که آنها شبیه یک فایل اجرایی ترجیحا ایستا کار می کنند. برای مثال یک درخواست برای یک اسکریپت PHP ممکن است به صورت زیر باشد:

http://www.somestie.net/article.php?id=425&format=html

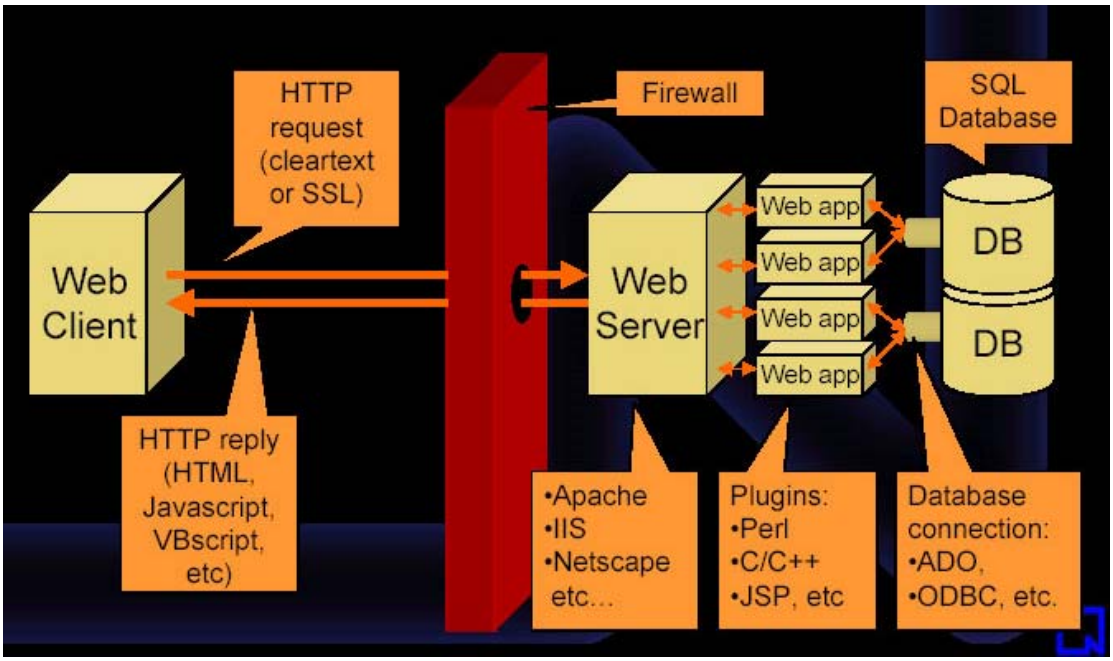
همانطور که مشاهده می کنید ، فایل article.php شبیه یک فایل اجرایی عمل می کند و پارامترهایی که همراه با آن ارجاع داده شده اند مانند ورودی ها و یا آرگومانهای این فایل اجرایی می باشند.

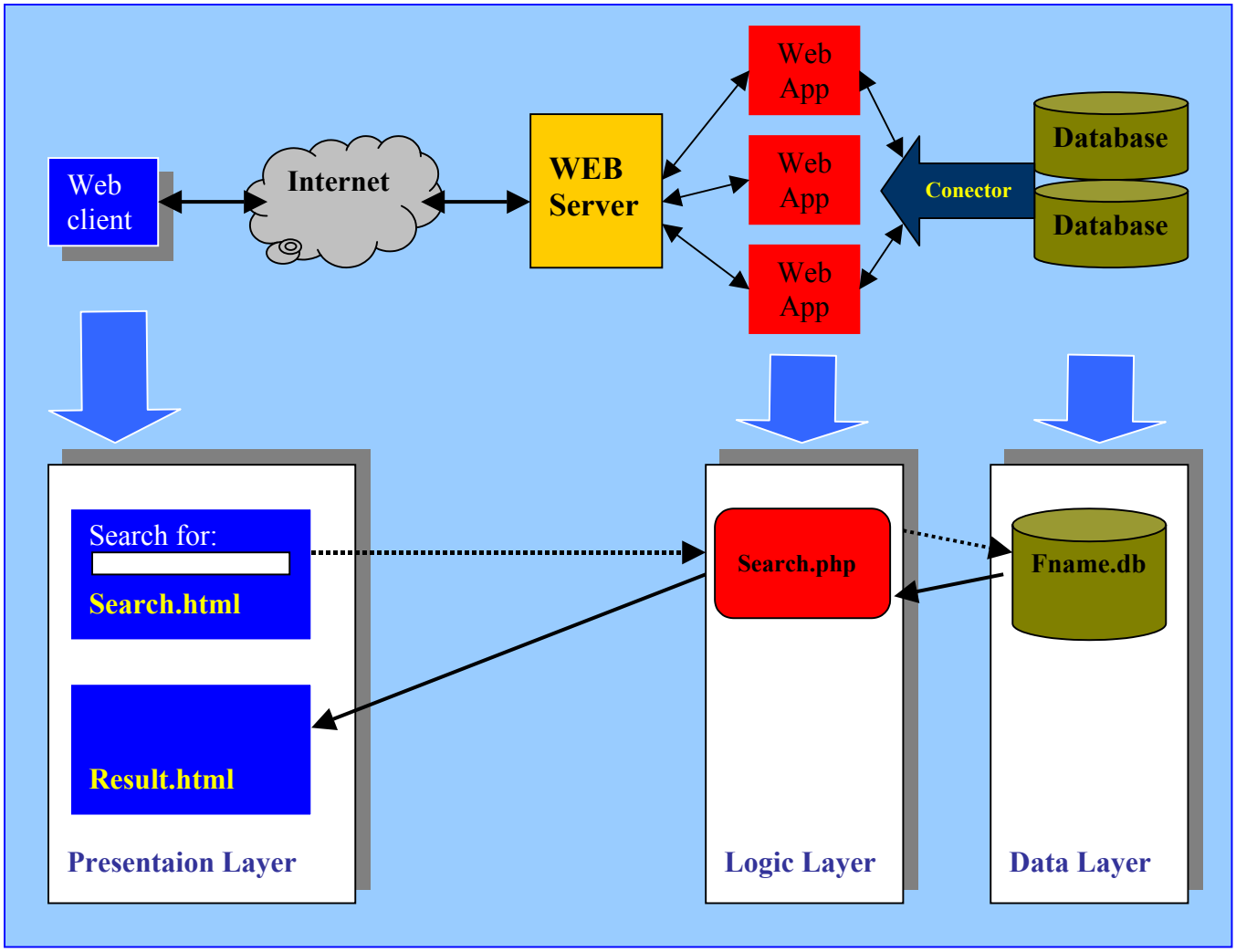
پایگاه داده

لایه داده معمولا به عنوان آخرین لایه یک برنامه کاربردی وب در یک معماری چندلایه ای می باشد. شاید خیلی بیشتر از چیزهای دیگر، پایگاههای داده مسوول تحول یک فرم استاتیک بر پایه HTML به فرمهای دینامیک ، بازیابی اطلاعات سیال و روان و تجارت الکترونیک شده باشند.

بیشترین زمینه کاری پایگاههای داده در وب روی دو عنوان می چرخد: SQL و Oracle. مولفه های لایه منطقی از ارتباط دهنده های معروف برای برقراری ارتباط با پایگاههای ، ساختن query ها ، به روز رسانی رکوردها و ... استفاده می کند. یکی از عمومی ترین رابطهایی که امروزه استفاده می شود Open Database Connectivity و یا ODBS می باشد.

در زیر شمای کلی یک درخواست و پاسخ و معماری برنامه های کاربردی را مشاهده می کنید.





شرکت ها	تکنولوژیها
Microsoft	Actice Server Page (ASP) ASP .NET ASAPI Common Object Model (COM) JavaScript
Sun Microsystem **IBM Websphere** **BEA Weblogic**	Java 2 Enterprise Edition (J2EE), including Java Servlets Java Server Pages (JSP) CORBA
Apache Software Foundation	PHP (Hypertext Perprocessor) Jakarta (server – side Java)
(none)	HTML CGI (including Perl)