پدافندغیرعامل

در حوزه مسیریاب

فهرست

مقدمه

وب[1] یکی ازسرویسهای نسبتاً نوین اینترنت است که درسال ۱۹۹۲ پا به عرصه وجودگذاشته است.درآن زمان از وب صرفاً جهت اطلاع رسانی استفاده میشد ودر واقع ابزار سادهای بود که بین سرویس‌دهندهها ومشتریان آن سرویس یک ارتباط گرافیکی برقرار می‌کرد . باتوجه به محبوبیت وب نزدکاربران، این سرویس شروع به فراگیرشدن کرد تا آنجاکه امروز هم مردم اینترنت را باسرویس وب آن میشناسند .

پس از گسترش روزافزون سرو یس وب ، نیاز به ارتباط دو طرفه ب ین کاربر و سرویس‌دهنده امری حیاتی به نظر میرسید و این مسئله برنامه‌نویسان و توسعهدهندگان را بر آن داشت تا ابزار ی ابداع کنند که سرویس‌دهنده وب را از حالت اطلاع رسانی محض خارج کند. شروع این کار با توسعه برنامههای کاربردی تحت وب انجام شد. این برنامهها توانستند دادههایی را از کاربران در یافت نموده، سپس پردازش‌ه‌ای لازم را روی آن انجام ده‌ند و در نهایت نتایج را از طریق وب در اختیار کاربران قرار ده‌ند. برنامههای کاربرد ی اولیه از پروتکل CGI[2]استفاده م یکردند. ا ین پروتکل پردازش اطلاعات دریافتی را در زبان c انجام داده و سپس نتایج را به صورتHTML در اختیار کاربر قرار می‌دهد. مشکلات موجود در CGI باعث شد که این پروتکل دوام زیادی در دنیای نرمافزارهای تحت وب نداشته باشد. آسیب‌پذیریهای موجود در سرویس - های متنی و همچنین فقدان قابلیت گستر ش‌پذیریاین پروتکل ، دو مشکل عمده‌ای است که CGI را از میدان خارج کرده است. مشکل دوم باعث می‌شود که به ازای هر درخواست، یک پروسه CGI فراخوانی شود و ا ین مسئله برا ی پیاده‌سازی برنامههای کاربردی با پردازش بالا مشکلساز است. همچنین ا ین نوع برنامهها نسبت به تع یین

---

[1] web
[2] Common Gateway Interface

ورودی صحیح نیز آسیب‌پذیرهستند و مشکلات سرریز بافر در آن ها بسیار دیده می شود.با توجه به مشکلات یاد شده، حملات DoS[1]نیزبر روی این نوع برنامهها به‌راحتی میتواند انجام پذیرد.

مشکلات یاد شده، توسعه‌دهندگان وب را برآن داشت تا بتوانند محیطی پویاتر برای برنامهنویسان برنامههای تحت وب فراهم کنند. چرا که محبوبیت فراوان سرویس وب نزد کاربران باعث گسترش استفاده از پروتکل HTTPو حتی فراوانی مرورگرهای وب شد. در این زمان بود که وب سرورهای تجاری شروع به پشتیبانی از رابط‌های برنامه نویسی برای نوشتن صفحات پویا نمودند که از آنجمله می‌توان به ISAPI[2]مایکروسافت اشاره کرد. با این عمل سرورهای وب بیش از پیش در معرض حملات ناشی از آسیب‌پذیری این نوع سرویسها قرار گرفتند. ماژولها ی اضافه شده امکان استفاده از زبانها ی سمت سرور مانند Perl، PHP و ASP را فراهم آوردند و همین امر راه نفوذ به سرور را از طریق این ماژولها هموار ساخت، برای نمونه در ماه فوریه سال ۲۰۰۲ میلادی مشکلی که در mod_php برای آپلود فایلیافت شد، به نفوذگر این امکان را میداد که کد دلخواه خود را به صورت ریموت اجرا کند.

اهداف حملات به پایگاههای وب در این دوران معمولاً تغییر در صفحه نخست پایگاه و یا بدست آوردن اطلاعات محرمانه از آن پایگاه بود. در همان دوران رشد فزآور ی‌های زبانها ی تحت وب بیشتر بر پایه توسعه کارایی بود و به مسئله امنیت کمتر توجه می‌شد . همزمان، پیشرفت در معماریهای امنیتیو ساخت فایروال‌های سخت افزاری منجر به کاهش حملات از طریق سوء استفاده از آسیبپذیریهای سایر سرویسهایغیر وب شد.چرا که مدیران شبکه به راحتی دسترسی به این سرویسها را تنها برای کاربران خاصی از شبکه مجاز میشمردند. ولی محدودسازی ترافیک وب امری بیهوده به‌نظر میآید چرا که بایستی این سرویس برای دسترسی عموم به پایگاه

[1]Denialof Service
[2]Internet Server Application Programming Interface

وب مجاز باشد. از اینرو توجه بیشتر نفوذگران متوجه سرویس پورت ۸۰و ۴۴۳ شد. با توجه به ا ین موضوع ، وبسرورها، سرو یسهای آن ها و به‌طور کلی نرم افزارهای کاربردی تحت وب نوک پیکان حملات قرار گرفتند.

در فصل ۱، ابتدا حملاتی که در گذشته علیه وب انجام شدهاند، معرفی خواهند شد و سپس تغییراتی که در ماهیت حملات در طی این سال‌ها انجام شده بیان میشود و برخی از جدیدترین حملات مهمی که علیه برنامههای کاربردی تحت وب انجام پذیرفته است به اختصار معرفی میشوند.

فصل دوم به بیان ویژگیهای آزمون نفوذ جعبه سیاه می‌پردازد.انتهای کتابچه نیز به نتیجه‌گیری اختصاص یافته است.

فصل ۱

تاریخچه حملات تحت وب

در اوایل دهه اخیرو در ابتدای شروع گسترش پایگاههای وب، شرکت‌های امنیتی دنیا مانند eeye، گزارشها ی متعدد ی از نحوه سوء استفاده از آس یبپذیری سرورIIS[1] به صورت از راه دور در اختیار قراردادند. آس یب‌پذیری‌های نخستین IIS بسیار خطرناک بودند چراکه IIS پردازه ای بود که کاربرِ SYSTEM سیستمعامل، آن را اجرا می‌کرد و از این رو نفوذگر را قادر میساخت که کنترل سیستم را بطور کامل در دست بگیرد. در این دوره معمولاً حملات انجام شده به وب سرورها بیشتر براساس آس یبپذیریاجزای تشکیل‌دهنده وب سرور بود که میتوان به سوء استفاده از آس یب پذیریهایموجود در کد نرمافزار وب سرور و یا توابع کتابخانهای آن اشاره کرد که البته این آس یبپذیریها عموماً در نرمافزارهای دیگر نیز وجود داشتند.

1Internet Information Server

حملات قدیمی وب

بطور کلی میتوان آسیبپذیریهایقدیمی سرویسدهندههای وب را به صورت زیر خلاصه نمود:

- حملات ناشی از سرریز بافر

در این حمله، ورودی کاربر وارسی نمی شود و دادهها بیش از اندازه بافر و بدون در نظر گرفتن حجم آن درون بافر قرار م یگیرند. در ا ین حالت نفوذگر می‌تواند کد مخرب خود را درون قسمتها یی از stack و یا heap قرارداده و آنها را با مجوز کاربر سرور وب (مثلاً SYSTEM)اجرا کند.

- عدم بررسی ورودی

بسیاری از اوقات ورودی دریافت شده از سمت کاربر به درستی بازبینی نمیشود و این مسئله میتواند به راحتی با ارسال یک فرمان ساده امنیت سرور وب را به مخاطره بیاندازد.

- آشکارسازیرشته ورودی

در ا ین حمله برنامه کاربرد ی تحت وب بدون بازبینی مقاد یر ارسال شده توسط کاربر، آن ها را نما یش م یدهد. البته ا ین حمله در زبان های برنامه نویسی غ یرC، به ندرت دیده می‌شود.

- Encode کردن درخواست

در صورتی که درخواست کاربر به صورت خاصی encode شده باشد، ممکن است در طرف سرور وب درخواست مذکور به گونه‌ای دیگر تفسیر شود. این امر به نفوذگر این امکان را می‌دهد تا بتواند از برخی فیلترهای عبارات بگذرد.

- ارتقای امتیاز کاربر

نفوذگر با استفاده از این دست آسیب‌پذیری‌ها می‌تواند به راحتی مجوزها/دسترسی‌های خودش را ارتقا بخشد، این امر می‌تواند به صورت راه دور و یا محلی انجام پذیرد. نمونه‌های متعددی از این آسیب‌پذیری در نرم‌افزارهای مختلف دیده شده که سرور وب نیز از این امر مستثنی نیست. این آسیب‌پذیری‌ها در سیستم‌عاملهای مختلف نیز دیده شده است.

- تغییر در اطلاعات فرمها

برخی از حملات می‌تواند تنها با تغییر محتویات فرمها قبل از ارسال کردن آن ها به سمت سرور صورت پذیرد، درصورتی که مقادیر ارسال شده در سمت سرور بازبینی نگردند، یک نفوذگر می‌تواند به راحتی از این مسئله سوء استفاده کند.

حملات جدید وب

اقدامات گسترده‌ای که در سال‌های ۲۰۰۳ تا ۲۰۰۴ در خصوص کاهش عوارض ناشی از آسیب‌پذیری‌های مختلف صورت گرفت (از جمله این موارد میتوان به بررسی اجزای تشکیل‌دهنده وب سرورها، بازبینی کدها توسط شرکتها ی تولید کننده این نوع نرم‌افزارها، تداوم در امر تحقیق پیرامون آسیب‌پذیری‌ها، توسعه نرم افزارهای بازرسی کد و ارتقای امنیت در سیستمهای عامل اشاره کرد)، تاحدودی منجر به پایداری امنیت در آن سال‌ها گردید. کاهش آمار حملات موفق به سرویس‌دهنده -

های وب در این سال‌ها نمایانگر صحت این ادعا است.هر چند که راههای نفوذ از طریق موارد فوق به صورت کنترل شده‌تری در آمد ه است، اماهنوز نسبت به توسعه سریع نرم‌افزارهای تحت وب و مشکلات امنیتی آن راهحلی اندیشیده نشده است. این امر با توسعه وبلاگ ها، فرومها و سرو یسهای تحت وب شدت بیشتریافته است. در گذشته اکثر نفوذگران با اهداف مالی اقدام به حمله علیه سایتها مینمودند، درحالی که اهداف د یگری همچون سوء استفاده از اشخاص و نرمافزارها ی تولیدی توسط شرکت‌های مختلف نیز بدان افزوده شده است.

استفاده از پایگاههای داده در نرمافزارهای تحت وب، بیش از پیش این نرمافزارها را در معرض خطر از دست دادن اطلاعات قرار داد ه است و روش‌های نو ینی برای سوءاستفاده از آسیبپذیریهای وب بوجود آمده است که نمونهای از آن ها به قرار زیر است:

- حملات XSS

کلمه XSS اختصار یافته عبارت Cross Site Scripting است. در ا ین حمله میزبان آسیبپذیر، کدهای ارسالی نفوذگر را به مرورگر کاربران باز م یگرداند . این حمله به دو صورت کلی وجود دارد:

- انعکاسی
- ذخیره شده

در حالت انعکاسی نفوذگر میکوشد کاربر را از طریق اجرای یک لینک فریب دهد، در این صورت کد مخرب نفوذگر در مرورگر کاربر از طریق آن پایگاه آسیب - پذیر اجرا خواهد شد.در حالت دوم که به آن اصطلاحاً HTML Injection نیز گفته میشود، نفوذگر م یکوشد کد خود را درون سرور ذخ یره کرده (پا یگاه داده، فایل‌های درون سیستم،...) تا پس از آن سرور، ا ین اطلاعات را به کاربران نشان دهد. نمونه بارز ا ینحالت، فرومها یی هستند که در آن‌ها کاربران م یتوانند اطلاعات ر ا به

صورت HTML پست کنند. حمله اخ یر از اهم یت و یژه‌ای برخوردار است چراکه نفوذگر تنها یک بار کد را تزر یق کرده در حال‌ی که م ی‌تواند کاربران متعدد ی را آلوده سازد.

- تزریق کد SQL

در ا ین روش نفوذگر م یکوشد از طر یق راهحلها ی ممکن اقدام به استفاده از اطلاعات پایگاههای داده کند و یا آن‌ها را به نفع خو یش تغییر دهد. روزانه نمونههای زیادی از ا ین نوع حمله در پا یگاههای مختلف صورت م ی‌پذیرد. بیشتر نرمافزارهای مدیریت محتوا قربانی این نوع حملات هستند. بسیاری از حملات به پا یگاههای وب در چند سال اخیر از طریق این روش انجام گرفته است.

- دسترسی مستقیم به اشیاء

این مسئله هنگام‌ی رخ م یدهد که نو یسنده برنامه تحت وب، ارجاع به یک شئ (مانند فایل، شاخه و یا رکورد یک پایگاه اطلاعاتی) را بهوسیلهیک URL انجام دهد. یک نفوذگر به راحتی میتواند با استفاده از تغ ییر در URL و بدون داشتن مجوز به سایر اشیاء دسترسی داشته باشد. در مثال زیر نیز هر چند که ه یچ نوع تزر یقSQL ای م یسر ن یست، ول‌ی کاربر به راحتی م یتواند متغیرID را تغییر دهد:

```
intcartID = Integer.parseInt( request.getParameter( "ID" ) );
String query = "SELECT * FROM table WHERE ID=" + ID;
```

- اجرای کد مخرب

در این روش نفوذگر سع‌ی م یکند با سوء استفاده از ضعف برنامه‌نویس در ارجاع دادن urlهای محل‌ی و یا ر یموت به متغیرهای فایل‌ی برنامه (بدون بازبین‌ی آن‌ها)، اقدام

به اجرای فایل‌های مخرب از این طریق نماید. نفوذگر با استفاده از این آسیب‌پذیری قادر است:

- کدهای راه دور را اجرا کند.
- شل‌های راه دور[1] را نصب کرده و اطلاعات زیادی در رابطه با پیکربندی سیستم به دست آورد.
- برخی از اجزای سیستم‌عامل ویندوز را با استفاده از امکانات PHP در اختیار گیرد.

جعل امتیاز

در این حمله نفوذگر می‌کوشد با استفاده از کاربرانی که مجاز شمرده می‌شوند، درخواست خود را به یک نرم‌افزار تحت وب آسیب‌پذیر ارسال کند. اساس این آسیب‌پذیری آن است که برای درخواست‌های مهم هیچ نوع احراز اصالتی انجام نمی‌پذیرد. اکثر پایگاه‌های وبی که نسبت به حمله XSS آسیب‌پذیر هستند، به این نوع حمله نیز آسیب‌پذیرند. نمونه‌ای از کد آسیب‌پذیر به فرم زیر است:

```
<img
src="http://www.somebank.com/transfer.do?frmAcct=document.form.frm
Acct &toAcct=1345754&toSWIFTid=134343&amt=13424.43">
```

در این جا کاربر تنها با بارگذاری این tag باعث می‌شود کاربری که اکنون توسط پایگاه somebank.com مجاز شمرده می‌شود، از حساب خود به حساب دیگری (حساب نفوذگر) پول واریز نماید.

[1] Sheller

- ضعف در کنترل دسترسی و مدیریت جلسه

در بسیاری از موارد عدم احراز اصالت کاربران منجر به حملات فوق میشود. به بیان دیگر بسیاری از حملات از جمله حملات ذکر شده در بالا را میتوان با استفاده از محدود نمودن دسترسی برطرف ساخت .محدود ساختن استفاده از یک فایل خاص، مشاهده آن و یا ایجاد محدودیت روی دستورات SQL برای برخی از اعضای خاص میتواند نمونهای از روش های پیشگیری از این مسائل باشد. راهکارهای زیر می توانند در مدیریت جلسه سودمند واقع شوند.

برخی از اوقات عدم استفاده از روش های رمزنگاری داده مانند SSL میتواند منجر به فاش شدن کلمات عبور و یاTokenهای استفاده شده شود. لذا رمزکردن اطلاعات برای جلوگیری از حملاتی مانند جعل امتیاز بسیار مثمر ثمر است. استفاده از رمزنگاری هنگام احراز اصالت کاربرانامر ی ضروری به نظر م یرسد . این کار را می - توان برای جلوگیری از حملات استراق سمع انجام داد.

از دیگر روش‌های مناسب آن است که تنها از امکانات مدیریت جلسه زبان برنامه نویسی استفاده شود و از پذ یرفتن شناسههای جلسه جد ید از طر یقURLها خودداری شود (چراکه این روش خود وسیلهای برای در اختیار گرفتن جلسه کاربران مهم است و اصطلاحاً به آن حمله تصح یح جلسه گفته م یشود). همچنین بایستی از زمان‌بند ی برایکاربران احراز اصالت شده استفاده شود و همچنین بایستی از آشکارسازیTokenها در URL اجتناب شود.

- ضعف در مدیریت خطاها

بسیاری از برنامهها به طور ناخواسته از طر یق خطاها اطلاعاتی را در مورد نحوه پیکربندی و تنظیمات به نمایش میگذارندو معمولاً نفوذگران با مشاهده این خطاها می توانند حملات قویتری را ترتیب دهند. نمونه کاربرد ی این مسئله در حملات

تزریق در SQL است. لذا می‌بایستی مدیریت مناسبی از طریق Exception Handlerها ایجاد کرد و اطلاعاتی که سیستم پس از وقوع خطا آشکار می‌سازد را محدود نمود.

فصل ۲

پویش آسیب پذیری

همانطور که گفته شد، ساز وکارهای سنتی در معماری امنیت بیشتر بر پایه ابزارهایی همچون فایروال، IDS و ارتباطات رمزشده بود، حال آنکه این راهکارها برای جلوگیری از سوء استفاده از نرمافزارهای تحت وب مناسب نیستند. گروه Gartner اعلام کرده است که ۷۵٪ حملات به نرم‌افزارهای تحت وب بوده است [5] و حتی ۹۲٪ از این حملات برای تغییر چهره سایت انجام گرفته است. یکی از شرکتهای تولیدکننده محصولات تجاری برای پویش امنیتی نرمافزارهای تحت وب در گزارشی در سال ۲۰۰۷ اعلام داشته است که ۷۰٪ پایگاههای وب خطر حمله توسط نفوذگران را دارا هستند و ۹۱٪ آن‌ها از آسیب‌پذیریهای جدیای مانند تزریق SQL و XSS برخوردار هستند.

امروزه نرمافزارهای تجاری متعددی برای پویش آسیبپذیریهای نرمافزارهای تحت وب وجود دارد، حتی نسخههای آکادمیکی از این نوع نرمافزارها عرضه شده است. بیشتر نرمافزارهای یاد شده به صورت خودکار به پویش صفحات پرداخته و به دنبال آسیبپذیریهای ممکن میگردند. مدیران شبکه نیز قالباً از نرم افزارهای فوق جهت یافتن مشکلات و آسیبپذیریها استفاده میکنند لذا اطمینان به نتایج ارائه شده توسط این نرمافزارها امری مهم محسوب میشود . البته باید توجه داشت که ممکن است بسیاری از آسیب‌پذیریهای تشخیص داده شده توسط این نرم‌افزارها واقعی نبوده و خطای نرمافزار باشند. بنابراین بایستی نتایج ارائه شده توسط این‌گونه نرم‌افزارها را با دقت بررسی نمود.

پویش کردن از سه فاز اصلی یعنی تنظیم، نقشه‌برداری و پویش تشکیل شده است. در فاز تنظیم آدرس برنامه تحت وب در اختیار پویشگر قرار میگیرد. در فاز نقشهبرداری، پویشگر سعی در به‌دست آوردن ساختار نرم افزار با استفاده از لینکها

میکند، این فاز مهمترین فاز محسوب میشود چرا که پیدا نکردن برخی از صفحات در این فاز منجر به عدم پویش آن‌ها توسط پویشگر خواهد شد. در نهایت در فاز پویش، برنامه وب تحت تست نفوذ قرار میگیرد. پویشگر این کار را با استفاده از شبیه‌سازی یک کاربر که بر روی صفحات کلیک کرده و یا فرمها را پر میکند انجام میدهد. در این مرحله هزاران آزمایش بر روی صفحات صورت میپذیرد. حتی در این مرحله داده‌هایی نیز برای دریافت خطا ارسال میشود تا بتواند از طریق خطاهای گرفته شده اطلاعاتی را کسب نماید. در این فاز ممکن است لینکها و صفحات جدیدی یافت شوند که به اطلاعات جمع‌آوری شده در فاز قبل افزوده میشوند. پس از انجام این فاز، مجدداً این گامها برای صفحات جدید انجام میشوند و در نهایت گزارشی مبنی بر مشکلات یافت شده به کاربر ارائه میشود. حتی در برخی از این نرمافزارها راهکارهای مقابله نیز بیان می‌گردند تا مدیران شبکه بتوانند از این طریق به رفع مشکلات برنامه بپردازند.

بررسی پایگاه‌های نمونه

امروزه دو روش کلی برای آزمایش نرمافزارهای تحت وب وجود دارد، روش جعبه سفید و روش جعبه سیاه. روش جعبه سفید شامل تجزیه و تحلیل کدهای برنامه است که میتواند به صورت دستی و یا خودکار انجام شود. در روش جعبه سفید از راه حلهای حساس به FIC[1] استفاده میشود. در روش جعبه سیاه که به صورت فازی عمل میکند، پویشگر از ساختار داخلی برنامه و کد آن اطلاعی ندارد و صرفاً دادههایی را به صورت تصادفی به نرمافزار تحت وب میدهد و نتایج آن را مشاهده میکند. این روش نسبت به روش قبل محدودتر بوده و لذا بسیار سریعتر عمل میکند. تمامی نرم‌افزارهای پویش به صورت ریموت، از روش جعبه سیاه استفاده میکنند.

---

[1] Flow, Interprocedular and Context sensitive

برای بررسی میزان آسیب‌پذیری سرورهای وب، تعدادی پایگاه وب برای پویش به صورت جعبه سیاه انتخاب شدند. هدف از انجام پویش بررسی وضعیت امنیتی برنامه‌های تحت وب آنهاست. با توجه به اینکه اکثر مراکز از تجهیزات امنیتی سختافزاری پر قدرتی برای دفاع از شبکه خود بهره میبرند، ولی نتایج ارائه شده حاکی از آن است که اکثر مشکلات، ناشی از عدم توجه به آسیبپذیریهای برنامهها و سرویسهای تحت وب است. پس از انجام پویش، نتایج صحیح از نتایج ناصحیح جدا شده و سپس در بررسی نتایج اعمال گردیدهاند.

با توجه به نتایج مشاهده میشود که درصد بیشتری از آسیبپذیریها مربوط به حملات XSS است. برخی از حملات XSS خود میتواند منشأ بسیاری از حملات دیگر باشد و نتایج نیز نشان می‌دهد که پایگاههای آسیبپذیر به این نوع حمله، درصد قابل توجهی از سایر حملات را نیز دارا هستند.

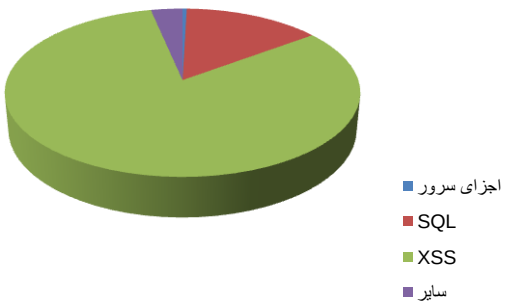
همچنین با توجه به بالا رفتن سطح آگاهی مدیران شبکه و به روزرسانی مداوم نرم‌افزار سرور وب، مشاهده میشود که در تمامی پایگاهها، آسیبپذیریهای سرور، درصد ناچیزی از آسیبپذیریها را به خود اختصاص دادهاند.

در جدول (۱)، درصد سرورهای آسیبپذیر به هریک از حملات ذکر شده مشخص گردیده است. همانطور که مشاهده میشود، درصد قابل توجهی از سرورها نسبت به حمله XSS آسیبپذیر هستند.

جدول (۲): درصد سرورهای آسیب‌پذیر

	اجزای سرور	برنامه کاربردی تحت وب		
		SQL	XSS	سایر
توزیع آسیبپذیری	٪۲۸/۵۷	٪۳۵/۷۱	٪۶۴/۲۹	٪۴۲/۶

در شکل (۱) نیز درصد توزیع آسیبپذیریها در مجموع پایگاه‌ها ترسیم شده است. با توجه به مقادیر به‌دست آمده، حملات XSS و SQL به ترتیب بیشترین درصد را در مجموع حملات ممکن به خود اختصاص داده‌اند.



شکل (۱): درصد توزیع آسیبپذیریها در سرورها

مشاهدات فوق نشان میدهد که سرورهای معتبر نیز دارای آسیبپذیریهایی از این دست هستند. همچنین نتایج جدول (۱) حاکی از آن است که هنوز سرورهایی وجود دارند که دارای آسیب‌پذیریهای قدیمی وب هستند و این امر نشانگر عدم توجه کافی مدیران به این دسته از آسیبپذیریها است.

نتیجه

در این کتابچه پس از معرفی آسیب‌پذیریهای قدیمی و جدید وب، به بررسی این آسیبپذیریها در برخی از پایگاههای مهم کشور پرداخته شد. نتایج به‌دست آمده نشان میدهد که پایگاههای مهم در معرض حملات XSS و تزریق در SQL قرار دارند. از دیگر نتایج مهم این کتابچه، وجود آسیب‌پذیریهای قدیمی در برخی از سرورها است.

با توجه به آن که خطاهای انسانی نقش اول را در بروز حملات تحت وب ایفا میکنند، افزایش دقت عمل برنامه نویسان در کنار راه‌کارهای امنیتی دیگر (مانند استفاده از فایروالها و IDSها) امری ضروری به نظر میرسد. البته آنچه که به‌عنوان دقت عمل برنامه‌نویسان جهت تولید نرمافزارهای تحت وب از آن یاد شد، شاید راهکار مناسبی در جهت تولید نرمافزارهای جدید باشد ولی راه حلی برای نرم‌افزارهای نوشته شده در قدیم نخواهد بود. لذا آشنایی مدیران شبکه با حملات وب و نحوه تشخیص آن‌ها بسیار ضروری است، چرا که برای برخی از آسیب‌پذیریهای برنامههای تحت وب، سناریوهای مختلف حمله وجود دارند که به راحتی میتوان از طریق نفوذ در نرم‌افزارهای تحت وب، سیستم را به صورت کامل در اختیار گرفت. در این کتابچه سعی شده لزوم تشخیص این آسیبپذیریها و جلوگیری از انجام حملات به‌طور واضح بیان گردد.