امنیت در دنیای وب – بخش دوم

پیشگیری

در بخش اول از مجموعه مقالات «امنیت در دنیای وب» درباره تعریف امنیت در دنیای IT و بستر وب صحبت شد. بخش‌های مختلف مدل استقرار **نرم‌افزارهای تحت وب** توضیح داده و در نهایت نقاط ناامن این بستر توضیح داده شد. در انتهای بخش اول دو رویکرد پیشگیری از حمله و کشف حمله معرفی شد. در بخش دوم این مقاله سعی می‌شود رویکرد پیشگیری در بستر وب به صورت کامل توضیح داده شود. در این بخش ابتدا کمی مفصل‌تر درباره این رویکرد صحبت خواهیم کرد و وظایف یک شرکت تولیدکننده نرم‌افزار را در این رویکرد مشخص می‌کنیم. پس از آن در کنار آشنایی با انجمن OWAPS درباره انواع حملات بر روی بستر وب بر اساس تعریف این انجمن صحبت خواهیم کرد و سعی می‌شود با مثال‌هایی ساده، شیوه این حملات و تاثیر این حملات بر بخش‌های مختلف از استقرار نرم‌افزارهای تحت وب را توضیح دهیم. در انتهای این مقاله نیز شما را با استاندارد ASVS آشنا و سطوح مختلف آن و کاربرد های آن را توضیح خواهیم داد.

رویکرد پیشگیری

در بستر وب به صورت ساده، پیشگیری یعنی بستن راه نفوذ و سوءاستفاده از تعاریف و قواعد وب برای دسترسی به اطلاعات موجود در نرم‌افزارهای تحت وب. برای بازکردن

این مفهوم نیاز است آن را به دو بخش تقسیم کنیم: تعاریف و قواعد موجود در بستر وب و سوءاستفاده‌کنندگان.

سوءاستفاده‌کنندگان: هر فرد یا سیستمی است که به دنبال اطلاعاتی است که در سیستم، اجازه دسترسی به آنها را ندارد. هکرها شناخته شده‌ترین سوءاستفاده‌کنندگان هستند. اما آیا فقط هکرهایی که به صورت حرفه‌ای به دنبال اطلاعات هستند تنها مظنونین برای سوءاستفاده هستند؟

در دنیای امنیت پاسخ این سئوال منفی است. بر اساس تعریف امنیت و تمرکز بر سوء استفاده کنندگانی که به دنبال اطلاعات غیرمجاز هستند؛ علاوه بر هکرها، کاربران مجاز نرم‌افزار و سیستم‌های اطلاعاتی مرتبط با نرم‌افزار نیز می‌توانند به صورت بالقوه خطرآفرین باشند.

برای مثال، کارشناسی را در نظر بگیرید که در یک سازمان درخواست دیدن اطلاعات فیش حقوقی مدیرعامل را دارد. این کاربر احراز هویت کرده است و به صورت کاملا مجاز اجازه فعالیت در نرم‌افزار را داراست؛ اما اگر نرم‌افزار هر درخواستی را از این کاربر بپذیرد به این معنی است که افشای اطلاعات روی داده است. درباره سایر سیستم‌های اطلاعاتی مرتبط نیز شرایط، تقریبا مشابه است. یک سازمان را در نظر بگیرید که دارای 2 نرم‌افزار متفاوت اما به هم مرتبط است که بین این 2 نرم‌افزار، تبادل اطلاعات وجود دارد. اگر این تبادل اطلاعات بدون محدودیت باشد و نرم‌افزار خدمات‌دهنده هر درخواستی از نرم‌افزار خدمات‌گیرنده را بپذیرد و اطلاعات درخواست شده را در اختیارش قرار دهد در واقع تمامی سیاست‌های پیش‌بینی شده در نرم‌افزار

برای حفظ اطلاعات از بین می‌رود و عملا اطلاعات، افشای عمومی می‌شود؛ بنابراین رویکرد پیشگیری از این منظر به بدین معنی است که راه های افشا و سوءاستفاده اطلاعات توسط هر کاربر یا سیستم غیرمجاز بسته شود.

قواعد و تعاریف موجود در بستر وب متفاوت است و هر کدام از آن‌ها با توجه به نیازمندی‌های نرم‌افزاری مجموعه‌ای از قابلیت‌ها در اختیار تولیدکنندگان قرار می‌دهند که می‌توانند ناامن باشند. به عنوان مثال مرورگرها کدهای جاوا اسکریپت اجرا می‌کنند.این یعنی اگر سوء استفاده‌کنندگان بتوانند راهی پیدا کنند که کدهای برنامه‌نویسی خود را بر روی مرورگر کاربر هدف اجرا کنند، می‌توانند به اطلاعات کاربر هدف دسترسی پیدا کنند. مثال دیگر این است که ارتباط بین مرورگر کاربر با سرور وب از طریق پروتکل Http است. در این پروتکل درخواست‌های کاربر در قالب خاصی از اطلاعات به صورت رمز نشده برای سرور ارسال می‌شود و همچنین پاسخ‌های سرور نیز با قابلیت خاص خود باز هم به صورت رمز نشده برای کاربر بازگردانده می‌شود. اگر کاربری بتواند در محیط وب در میانه راه، بین کاربر و سرور وب قرار بگیرد، می‌تواند تمامی درخواست‌ها و پاسخ‌ها را شنود کند. مثال آخر این که سرور وب از اطلاعات مختلفی که در درخواست‌های دریافتی وجود دارد، تشخیص می‌دهد که کدامیک از کاربران این درخواست را فرستاده است. اگر سوءاستفاده‌کننده‌ای بتواند این اطلاعات را به دست آورد، می‌تواند با شبیه‌سازی اطلاعات کاربر مجاز، درخواست دریافت اطلاعات را بدهد؛ بنابراین از این منظر رویکرد پیشگیری به معنی بستن راه‌های مختلف نفوذ سوءاستفاده‌کنندگان با استفاده از قواعد موجود در دنیای وب است.

در بخش اول مجموعه این مقالات درباره 3 عنصر محیط استقرار، کاربر استفاده‌کننده

و تولیدکننده نرم‌افزار و وظایف آن‌ها در بحث امنیت به طوری مختصر صحبت شد. اما در بحث پیشگیری جایی است که باید به صورت درستی تقسیم وظایف بین این بخش‌ها صورت گیرد. وظیفه کاربران است که با آموزش‌های لازم از اطلاعات کاربری و سیستمی محافظت کنند. وظیفه مدیران شبکه سازمان استفاده‌کننده از نرم‌افزار نیز این است که محیط استقرار امن را برای نرم‌افزار فراهم کند. به عنوان مثال بتواند به جای http از ساختار امن رمزنگاری شده https استفاده کند.

رویکرد ما در این مقالات متوجه تولیدکننده نرم‌افزار است و وظایف اوست. تولید کننده نرم‌افزار باید نرم‌افزاری تولید کند که:

1-با پیکربندی مناسب اجازه سوءاستفاده از اطلاعات را به هیچ کاربر مجاز یا غیرمجاز و یا سیستم‌های مرتبط ندهد.

شرکت تولیدکننده نرم‌افزار با ایجاد معماری مناسب و ساختار تعیین دسترسی مناسب نرم‌افزار باید بتواند همه درخواست‌های دریافت اطلاعات را از نظر امنیتی مدیریت کند و به صورت درست و تاثیرگذار، مجاز و یا غیرمجاز بودن درخواست‌ها را تشخیص دهد. این کار باید در زمان تولید نرم‌افزار بر اساس تعریف نیازمندی‌های امنیتی و در فازهای نیازسنجی، تحلیل، طراحی و معماری نرم‌افزار، تامین شود.

2- بتواند در محیط‌های استقرار با هر درجه از امن سازی کار کند.

نرم‌افزار باید بتواند پروتکل‌های امنیتی مختلف را پشتیبانی کند به طوریکه با استقرار آن در محیط‌های عملیاتی مختلف، خللی در کارکرد نرم‌افزار ایجاد نشود. پروتکل‌ها و

عناصر امنیتی روزبه‌روز در حال توسعه و به روزرسانی هستند به این دلیل که همواره هکرها در حال یافتن راه‌های نفوذ جدید در بخش‌های مختلف استقرار نرم‌افزارها هستند. معمولا در زمانی که حفره امنیتی کشف می‌شود پروتکل‌های امنیتی به روز می‌شوند تا این حفره‌ها را از بین ببرند. در بسیاری از مواقع، به‌روزرسانی‌ها تاثیری در نرم‌افزار تولید شده ندارند؛ اما در بعضی از موارد که نرم‌افزار از این پروتکل‌ها به صورت مستقیم استفاده می‌کند نیاز است که با ایجاد تغییر در نرم‌افزار بتوان آن را برای استقرار در محیط استقرار امن‌تر، آماده کرد. برای مثال الگوریتم‌های استفاده شده در پروتکل HTTPS را در نظر بگیرید. به مرور زمان الگوریتم‌های PCT 1.0 و نسخه‌های2 و 3 الگوریتم SSL که پایه این پروتکل بودند توسط هکرها شکسته شده و دیگر در محیط‌های عملیاتی امن مورد استفاده قرار نمی‌گیرند.

تا دو سال پیش نیز نسخه‌های1 و 1.1 و 1.2 از الگوریتم TLS معتبر بودند؛اما نسخه 1 این الگوریتم نیز شکسته شد. پس این رویداد شرکت مایکروسافت بسته‌های به‌روزرسانی‌ای برای نسخه‌های مختلف Sql server خود منتشر کرد که قابلیت پشتیبانی از نسخه 1.2 را در اتصالات به آن، داشته باشد و سپس یک نسخه جدید برای .Net Framework خود منتشر کرد که با استفاده از الگوریتم TLS 1.2 بتوان ارتباط امن با دیتابیس ایجاد کرد یا بتوان درخواست وبی را برای سرورهایی که از این نسخه الگوریتم پشتیبانی می‌کنند، فراهم کرد. (این نسخه‌ها سال گذشته منتشر شدند).

پس از این حوادث، شرکت‌های نرم‌افزاری که نرم‌افزار های کلاینت، سروری یا نرم‌افزار های تحت وب ایجاد می‌کردند، بروزرسانی نرم‌افزار خود برای استفاده از نسخه جدید .Net Framework را شروع کردند تا این قابلیت به نرم‌افزار آن‌ها نیز اضافه شود.

3-قابلیت پیکربندی و سیاستگذاری امن سیستم را در اختیار مدیران استفاده‌کنندگان از سیستم قرار دهد.

وظیفه تولیدکنندگان نرم‌افزار ایجاد قابلیت‌هایی برای مدیران امنیت سازمان‌های استفاده‌کننده نرم‌افزار است. در بستر وب همانگونه که در بخش اول این مقاله دیدیم معماری استقرار اینگونه نرم‌افزار ها مجموعه‌ای از تعاریف را همراه خود دارد. هر نرم‌افزار تحت وب با مجموعهای از مفاهیم مانند احراز هویت یا مفهوم نشست ایجاد شده بین سرور و مرورگر کاربر و ... ادغام شده است. نرم‌افزارهای امن باید بتوانند قابلیتی را برای مدیریت امنیت این بخش‌ها در نظر بگیرد. این قابلیت‌ها در واقع باید در فاز نیازسنجی نرم‌افزار به عنوان قابلیت‌های غیر عملیاتی (nonfunctional) در نظر گرفته شوند و در چرخه توسعه نرم‌افزار تولید شوند. به عنوان مثال نرم‌افزارها در زمینه احراز هویت باید بتوانند سیاست‌های پیچیدگی رمز عبور، شیوه نمایش تصویر امنیتی، محدودیت در تعداد دفعاتی که یک نام کاربری می‌تواند رمز خود را اشتباه وارد کند، احراز هویت از طریق عامل‌های دوم مانند امضای دیجیتال، OTP ها و ... را دارا باشند. این قابلیت‌ها به مدیران امنیت سازمان‌ها کمک می‌کنند تا سطوح مختلف امنیت را در سازمان خود پیاده‌سازی کنند.

4-راه‌های نفوذی که بر اساس مفاهیم و قواعد بستر وب وجود دارد را ببندد.

وظیفه تولید کنندگان نرم‌افزار این است که حفره‌های بستر یعنی وب را از بین ببرند. حملاتی که در بستر وب صورت می‌گیرند معمولا شیوه‌های شناخته شده‌ای هستند. برای

تمامی این شیوه‌ها نیز راهکارهایی برای هر زبان برنامه‌نویسی وجود دارد. وظیفه تولیدکنندگان نرم‌افزار این است که در فاز پیاده‌سازی در کدی که برای تولید نرم‌افزار نوشته می‌شود از این راهکار ها استفاده کنند تا اجازه نفوذ به نرم‌افزار داده نشود. این مورد به نظر ساده می‌آید؛اما در واقعیت جزو سنگین‌ترین و پیچیده‌ترین وظایف تولیدکنندگان نرم‌افزار است. تجربه شخصی نگارنده موید این است که محدودیت‌های امنیتی تولید نرم‌افزار در بستر وب گاهی حتی به تغییر معماری نرم‌افزار منجر می‌شود. شرکت‌های تولیدکننده نرم‌افزار هزینه‌های زیادی برای تولید کد امن متقبل می‌شوند و گاهی تولید آن در مقایسه با تولید کد غیرامن هزینه‌ها را تا 3 برابر افزایش می‌دهد؛ اما این هزینه‌ها باید پرداخت شود زیرا اعتمادسازی مهمترین دغدغه شرکت‌های تولیدکننده نرم‌افزار در سراسر دنیا است.

برای تولید کد امن در بستر وب باید شیوه‌های نفوذ به بخش‌های مختلف معماری استقرار نرم‌افزارهای وبی را شناخت و بر اساس تمامی قسمت‌ها راهکار هایی برای جلوگیری از آن‌ها را ارائه داد. خوشبختانه امروزه این کار بسیار ساده است. انجمن‌هایی مانند OWASP (Open Web Application Security Project) تمامی نیازمندی‌های شرکت‌های تولیدکننده نرم‌افزار را در بستر وب فراهم کرده‌اند؛ اما با توجه به اینکه شناخته شده‌ترین این انجمن‌ها و یا استانداردها در ایران OWASP است در این مقاله با مرجع قرار دادن آن مفاهیم مربوط به شیوه پیشگیری در بستر وب را توضیح خواهیم داد.

OWASP

OWASP یک انجمن غیر انتفاعی است که برای ایجاد امنیت در بستر وب تلاش می‌کند. این انجمن مرجع کاملی برای امنیت نرم‌افزار های وبی است که تا به حال نیز در این حوزه موفق بوده است به طوریکه حتی بعضی از کشورها پایه گواهینامه امنیتی نرم‌افزارهای تحت وب خود را بر اساس آن بنا کرده‌اند.

این انجمن سه رویکرد را دنبال می کند:

- 1-مرجعی برای اطلاع‌رسانی درباره همه شیوه‌های نفوذ در بستر وب باشد.
- 2-به ازای هر شیوه، راهکاری برای از بین بردن راه نفوذ ارائه دهد.
- 3-استانداردی برای آزمودن امنیت نرم‌افزارهای تحت وب ارائه دهد.

برای اطلاع‌رسانی درباره شیوه‌های نفوذ، این انجمن فهرستی از 10 شیوه مهم نفوذ با عنوان معروف«OWASP Top Ten» ارائه کرده و در آن، حملات به نرم‌افزارهای وبی را به 10 دسته کلی تقسیم‌بندی کرده است. به ازای هر شیوه نفوذ، مجموعه‌ای از راهکارها را برای تمامی زبان‌های معروف برنامه‌نویسی به صورت مقاله یا ابزارهای جانبی ارائه و در نهایت استانداردی به نام ASVS (OWASP Application Security Verification Standard Project) ارائه داده است که طی آن می‌توان سطح امنیت نرم‌افزارهای وبی را اندازه گرفت.

OWASP Top Ten

انجمن OWASP 10 شیوه مهم و معروف از حملات به نرم‌افزار های وبی را ارائه داده است. این انجمن این کار را در بازه های زمانی مختلف انتشار می دهد و در حال حاضر که در حال تهیه این مقاله هستم نسخه های 2010 و 2013 این 10 شیوه منتشر شده است و نسخه 2017 آن کاندید انتشار (RC) است. بنابراین با توجه به اینکه نسخه 2017 هنوز انتشار نهایی نشده است و احتمال تغییرات در آن وجود دارد در این مقاله 10 نوع حمله نسخه 2013 بیان می شود؛ البته باید این را عنوان‌کرد که بسیاری از شیوه‌های حمله با توجه به اینکه هنوز بسیار مورد استفاده هکرهاست بین دو نسخه مشابه است.
شیوه‌های معروف حملات به شرح زیر دسته‌بندی شده است:

- تزریق
- شکسته شدن احراز هویت و مدیریت نشست
- تزریق اسکریپت از طریق سایت (XSS)
- ارجاع مستقیم به اشیا به صورت ناامن
- پیکربندی اشتباه امنیت
- افشای داده‌های حساس
- از دست رفتن کنترل دسترسی‌ها در سطح توابع
- درخواست‌های تقلبی از طریق سایت (CSRF)
- استفاده از مولفه‌های جانبی با آسیب پذیری‌های شناخته شده
- تغییر مسیرها و ارجاعات نامعتبر

در ادامه هر یک از این نوع حملات به صورت خلاصه توضیح داده شده است.

تزریق

تزریق یا Injection شیوه‌ای از حمله است که حمله‌کننده سعی می‌کند بر اساس مولفه‌های مختلف یک نرم‌افزار وبی، دستورات برنامه‌نویسی را در بخش‌های مختلف سرور وب یا سرور دیتابیس اجرا کند. به صورت خلاصه و ساده حمله‌کننده سعی می‌کند از طریق درگاه‌های مختلف ارسال اطلاعات دستورات اجرایی برای پلتفرم‌های مختلف را ارسال و آن‌ها را روی سرور اجرا کند. نکته مهم این است که هدفگذاری این نوع حملات اجرای کد بر روی سرورهاست و تاثیری بر روی کاربران ندارد. این نوع حمله را هم ممکن است کاربران شناسایی نشده انجام دهند و هم کاربران شناسایی شده که احراز هویت شده‌اند و نرم‌افزار نباید برای این نوع حملات به هیچ کاربری با هر سطحی از دسترسی اعتماد کنند.

برای تمامی این نوع حملات قابلیت‌هایی بر روی زبان‌های برنامه‌نویسی و پروتکل‌های تعریف‌شده بین بخش‌های مختلف وجود دارد تا بتوان از حملات جلوگیری کرد.
تولیدکنندگان نرم‌افزار باید از این قابلیت ها به صورت کامل استفاده کنند و کارشناسان مسئول و ارشد این شرکت‌ها در زمان بازنگری کد در چک لیست خود باید حتما کدها را از نظر استفاده این قابلیت‌ها و عدم وجود فضای نفوذ بررسی کنند. در فاز تست نرم‌افزار نیز باید تمامی ورودی‌های بیان شده در این نوع حمله به منظور مقاومت در برابر درخواست‌های تزریق کد، تست شود تا میزان مقاومت نرم‌افزار در این نوع حملات بررسی شود.

شکسته شدن احراز هویت و مدیریت نشست

احراز هویت و مدیریت نشست 2 عنصر اصلی در بستر وب است. هر کاربری برای استفاده از نرم‌افزار ابتدا باید احراز هویت کند. همچنین بر اساس ماهیت ناهمگام میان کاربر و سرور وب نیاز است که برای کاربر استفاده‌کننده از نرم‌افزار، نشستی ایجاد شود تا به ازای هر درخواست کاربر سرور بتواند متوجه شود کدام کاربر درخواست را ارسال کرده است. در احراز هویت، حمله‌کنندگان سعی می‌کنند با شیوه‌های مختلف، توابع نوشته شده برای احراز هویت را بشکنند. آنها سعی می‌کنند به شیوه‌های مختلف، اطلاعات اشخاص معتبر استفاده‌کننده نرم‌افزار را به دست بیاورند.

در مدیریت نشست، حمله کننده به دنبال مشخصات نشست کاربر معتبری است که با نرم‌افزار در حال کار است. این حمله‌کننده می‌تواند از افراد ناشناس و یا از کاربران مجاز سیستم باشد که اهدافی مانند ارسال درخواست‌های مختلف به سرور و یا انجام کاری به صورت مخفیانه را دنبال می‌کنند.

نرم‌افزارها باید توابع مدیریت نشست خود را با عناصر مختلفی تقویت کنند. به عنوان مثال الگوریتم تولید شناسه باید الگوریتمی کاملا تصادفی باشد، شناسه نشست قبل و بعد از احراز هویت یک کاربر باید تغییر کند، اجازه مدیریت تعداد نشست‌های فعال به کاربر یا مدیر امنیتی نرم‌افزار در سیستم وجود داشته باشد و کاربر بتواند در هر زمانی نشست خود را خاتمه دهد. وظیفه تولیدکنندگان نرم‌افزارها این است که این قابلیت‌ها را در نرم‌افزار ایجاد کنند و مدیریت آن را در اختیار مدیران امنیت یا کاربران استفاده‌کننده نرم‌افزارها قرار دهند.

تزریق اسکریپت از طریق سایت (XSS)

این شیوه حمله بسیار شبیه به حمله تزریق است. در این شیوه نیز سعی می‌شود کدهای اسکریپتی درون نرم‌افزار تزریق شود. ولی تفاوت اصولی‌ای میان این شیوه و شیوه تزریق وجود دارد. این تفاوت در هدف حمله است. در شیوه تزریق هدف اجرای کد بر روی سرور های وب یا پایگاه داده بود اما در اینجا هدف اجرای کد بر روی مرورگر کاربران است. حمله کنندگان سعی می کنند کدهایی را در نرم‌افزار ذخیره کنند که هیچ تاثیری در سرور ها ندارد و اصولا بر روی سرور قابلیت اجرا ندارد اما وقتی که این دستورات برای درخواست یک کاربر به مرورگر کاربر منتقل می‌شود مرورگر این اطلاعات را به عنوان اطلاعات قابل اجرا تشخیص می‌دهد و آن را بر روی مرورگر کاربر اجرا می‌کند. هدف، به دست آوردن اطلاعات منتقل شده بر روی مرورگر کاربر یا به دست آوردن شناسه نشست و یا مشخصات حساب کاربری‌ست.

شیوه‌های مختلفی برای جلوگیری از این نوع حمله وجود دارد؛ اما همه این شیوه‌ها بر اساس یک اصل ساده استوار است و آن اینکه هیچیک از اطلاعات دریافتی سیستم‌ها یا کاربران مورد اطمینان نیست. وظیفه جلوگیری از این نوع حمله بر عهده تولیدکننده نرم‌افزار است و تولیدکننده باید این اصل را در تمامی بخش‌های نرم‌افزار خود رعایت کند.

ارجاع مستقیم به اشیاء به صورت ناامن

این نوع حمله ساده ولی کاربردی‌ترین نوع حملات است.برای توضیح آن یک مثال می‌زنم. فرض کنید کاربر معتبری وجود دارد که به نامه‌ای با شناسه 1 دسترسی دارد و به نامه با شناسه 2 دسترسی ندارد. نرم‌افزار نام امنی برای مشاهده یک نامه به کاربر شناسه نامه را در آدرس URL دریافت می‌کند و آن را به کاربر نمایش می‌دهد. کاربر با توجه به دسترسی خود که دارد ابتدا درخواست مشاهده نامه با شناسه 1 را می‌دهد. بعد از نمایش نامه متوجه می‌شود که در آدرس URL متغیری وجود دارد که مقدار آن 1 است. کاربر این مقدار را 2 می‌کند و درخواست جدیدی به سرور ارسال می‌نماید. به این ترتیب کاربر به صورت ناامن ارجاع مستقیم به نامه با شناسه 2 داشته است.اگر نرم‌افزار دسترسی او بر روی این نامه را بررسی نکند کاربر به هدف خود دست یافته و اطلاعاتی که به آن دسترسی نداشته را به دست آورده است. جلوگیری از این نوع حمله کاملا بر عهده شرکت تولیدکننده نرم‌افزار است.

پیکربندی اشتباه امنیت

در این نوع حملات مهاجمان سعی می‌کنند به اکانت پیش فرض، صفحات استفاده نشده، نقص‌های اصلاح نشده، فایل‌ها و دایرکتوری‌های محافظت نشده و ... دسترسی پیدا کنند. برای جلوگیری از این نوع حملات برای تولیدکننده و استفاده‌کننده نرم‌افزار وظایفی وجود دارد و این دو باید با هم نقشه‌ای از پیکربندی امن تهیه کنند و آن را در محیط استقرار نرم‌افزار اجرا کنند.

افشای داده‌های حساس

در این نوع حمله که توسط افرادی که به داده‌های حساس یا نسخه پشتیبان آنها می‌توانند دسترسی داشته باشند اتفاق می‌افتد، مهاجمانی که به صورت عادی نمی‌توانند به داده‌های رمزگذاری شده دسترسی داشته باشند برای دسترسی به داده‌ها کار دیگری مانند سرقت کلیدها انجام می‌دهند و یا داده‌هایی که در حال انتقال بین بخش‌های مختلف معماری استقرار نرم‌افزار هستند را سرقت می‌کنند. برای جلوگیری از این شیوه حمله، داده‌های حساس باید چه برای ارسال به کاربر و چه برای نگهداری در پایگاه داده رمزگذاری شوند.

از دست رفتن کنترل دسترسی ها در سطح توابع

این حمله ممکن است توسط هر کاربری که توانایی ارسال درخواست به نرم‌افزار را دارد روی دهد، مهاجم با تغییر Url یا پارامترهای آن، اقدام به اجرای روشی می‌کند که به آن دسترسی ندارد. نرم‌افزارها همیشه از سطوح دسترسی توابع نرم‌افزار حفاظت نمی‌کنند. بعضی زمان‌ها حفاظت به وسیله تنظیمات مدیریت می‌شود و اگر این تنظیمات سیستمی درست پیکربندی نشده باشند و یا در برخی از زمان‌ها توسعه‌دهندگان فراموش کنند این کار را در کد انجام دهند راهی برای نفوذ مهاجمان ایجاد می‌شود تا بدون دسترسی توابع نرم‌افزار را فراخوانی کنند.

در این حمله، وظیفه جلوگیری از نفوذ بر عهده تولیدکننده نرم‌افزار است. تولیدکننده باید تعریف شفاف و ساده‌ای از توابع و دسترسی‌های سوار بر توابع داشته باشد و بر

اساس این تعریف به پیاده‌سازی کدها اقدام کند. این پیاده‌سازی باید در سمت سرور وب موثر باشد؛ چراکه تاثیر بررسی دسترسی‌ها در مرورگر کاربر به تنهایی کفایت نمی‌کند.

درخواست‌های تقلبی از طریق سایت (CSRF)

این شیوه یکی از هوشمندانه‌ترین و همچنین شایع‌ترین شیوه‌های حمله به نرم‌افزار های وبی است که به آن «حمله از تب کناری» هم گفته می‌شود. در این حمله سعی می‌شود با باز شدن وب سایت مهاجم در تب کناری نرم‌افزار قربانی، درخواست‌هایی از طریق وب سایت مهاجم به نرم‌افزار قربانی ارسال می‌شود. تمامی این شیوه حمله بر اساس ساختار استاندارد مروگرها بر روی سیستم کاربران هستند؛ اما همچنان وظیفه جلوگیری از این نوع حمله بر عهده تولیدکننده نرم‌افزار است.

استفاده از مولفه‌های جانبی با آسیب‌پذیری‌های شناخته شده

تمامی تولیدکنندگان نرم‌افزار از مولفه‌ها و ابزارهای جانبی در کنار کدنویسی خود استفاده می‌کنند و برخی از وظایف نرم‌افزار را بر عهده آنها قرار می‌دهند. از این ابزارها به عنوان مولفه‌های طرف سوم در تولید نرم‌افزار نام برده می‌شود که به سرعت تولید و بالا بردن قابلیت‌های نرم‌افزار کمک می‌کنند.

این ابزارهای جانبی همواره در حال توسعه هستند و نسخه‌های مختلفی از آنها به مرور

زمان، تولید می‌شوند؛ اما نکته مهم این است که این ابزارها خود می‌توانند دارای باگ‌هایی باشند که باعث ایجاد حفره‌های امنیتی می‌شوند. این حفره‌های امنیتی بعد از کشف، تبدیل به آسیب‌پذیری‌های شناخته‌شده می‌شوند و اغلب در نسخه‌های بعدی این حفره‌ها بسته و ابزارها امن می‌شوند. وظیفه جلوگیری از این حمله کاملا بر عهده تولیدکننده نرم‌افزار است. تنها راه رفع این حفره‌ها این است که ابزار جانبی استفاده شده در نرم‌افزار با نسخه‌ای از این ابزار که آسیب‌پذیری را رفع کرده است، به روز شود.

تغییر مسیرها و ارجاعات نامعتبر

آخرین دسته‌بندی انواع حملات تغییر مسیرها و ارجاعات نامعتبر است که نرم‌افزارها گاهی بر اساس نیازمندی‌های سیستمی آن را انجام می‌دهند. اغلب نرم‌افزارها کاربران را برای دسترسی به سایر صفحات و به مسیرهای دیگر هدایت می‌کنند. یا اینکه از ارجاعات داخلی استفاده می‌کنند تا کاربر بتواند فرایند حرکتی انجام یک عمل را کامل کند. وظیفه جلوگیری از این نوع حمله نیز بر عهده تولیدکنندگان نرم‌افزار است. آنها یا باید به صورت ساده این نوع امکانات را نداشته باشند و یا اگر می‌خواهند از این امکان در نرم‌افزار خود استفاده کنند حتما باید مقدار متغیری را بررسی کنند که ارجاع آن به صفحه‌ای در داخل نرم‌افزار منتهی شود و هر ارجاعی با خارج از نرم‌افزار را رد کنند.

انجمن OWASP علاوه بر معرفی این حملات به ازای تک‌تک آنها حالت‌های مختلف حمله را بیان کرده است. به ازای زبان‌های برنامه‌نویسی و تکنولوژی‌های پایه بر روی هر زبان در وب راهکار هایی برای کدنویسی امن و جلوگیری از راه‌های نفوذ ارائه داده

است. این انجمن همچنین شروع به تولید ابزارهایی کرده است که مجموعه این راهکار های امن‌سازی در آن قرار دارند و می‌توانند به عنوان ابزارهای طرف سوم به کمک تولیدکنندگان نرم‌افزار آمده و به سرعت و دقت امن‌سازی نرم‌افزارها کمک‌کنند. آنها حتی راهنماهایی برای توسعه نرم‌افزار و شیوه تفکر امن در فازهای مختلف تولید نرم‌افزار ارائه داده‌اند.

تمامی این اطلاعات به صورت کاملا رایگان در وب سایت OWASP در اختیار تمامی استفاده کنندگان قرار گرفته است.

ASVS

انجمن OWASP جدا از معرفی امنیت در وب و شیوه‌های حمله و راهکارهای جلوگیری از حمله، تحت یک چارچوب با عنوان ASVS (Application Security Verification Standard) استانداری برای اعتبارسنجی میزان امنیت نرم‌افزارها ارائه داده‌اند. آنها از تولید این استاندارد 3 هدف عمده داشته‌اند:

1- **استفاده به عنوان یک شاخص**: این استاندارد می‌تواند شاخص مشترکی بین مجموعه نرم‌افزارهای وبی در بحث امنیت باشد. به این معنی که نرم‌افزارها را از نظر امنیتی قابل مقایسه می‌کند.

2-**استفاده به عنوان راهنما**: این استاندارد می‌تواند برای تولیدکنندگان نرم‌افزار راهنمایی باشد تا نرم‌افزارهای خود را امن کنند.

3-**استفاده به عنوان خریدار**: این استاندارد می‌تواند مبنایی برای شرایط امنیتی نرم‌افزارها در زمان عقد قراردادها باشد.

این اهداف در تولید این استاندارد باعث شده است طیف وسیعی از شرکت‌های خریدار نرم‌افزار و شرکت‌های تولیدکننده بتوانند بر سر یک چارچوب امنیتی ثابت به توافق برسند. در حال حاضر در دنیا بسیاری از شرکت‌ها برای انتخاب و خرید نرم‌افزارهای وبی از نتایج تست این استاندارد استفاده می‌کنند. این عمومیت موجب شد تا در بسیاری از کشورها نقطه شروعی برای تولید استاندارد دولتی نرم‌افزارهای وبی تلقی شود.

اما این استاندارد را می‌توان از دید استفاده‌کنندگان آن به چند دسته تقسیم کرد:

1ـ**تولیدکنندگان نرم‌افزار**: تولیدکنندگان از این استاندارد به منظور داشتن چک لیستی برای بررسی همه جانبه امنیت در نرم‌افزار خود استفاده می‌کنند.

2-**تست کنندگان امنیتی نرم‌افزار**: این استاندارد به کمک تست‌کنندگان امنیتی نرم‌افزارها می‌آید تا با ارائه شیوه‌های مختلف تست قسمت‌های مختلف و شبیه‌سازی حملات بتوانند به استانداردی بر روی نرم‌افزارهای وبی، دست پیدا کنند.

3-**استفاده کنندگان از نرم‌افزار**: این استاندارد به کمک استفاده‌کنندگان نرم‌افزار آمده تا بتوانند شاخصی برای امن بودن یا نبودن نرم‌افزار داشته باشند. آنها می‌توانند قبولی در این استاندارد را پایه‌ای برای خرید آنها قرار دهند؛ همچنین می‌توانند با تولیدکنندگان نرم‌افزار در بحث امنیت به زبان مشترکی برسند.

به شکلی می‌توان دولت‌ها را هم استفاده‌کننده این استاندارد دانست؛ اما با توجه به اینکه کشورها معمولا از مفاهیم این استاندارد برای تولید استاندارد خود استفاده می‌کنند نمی‌توان آنها را به عنوان استفاده‌کننده اصلی در نظرگرفت.

خود استاندارد مجموعه‌ای از نیازمندی‌هایی است که باید هر نرم‌افزار امن وبی داشته باشد. این نیازمندی‌ها در جملاتی کوتاه آورده شده است و هر یک از آنها راه نفوذ یک یا چند شیوه حمله را از بین می‌برد. استاندارد نرم‌افزار ها را به 3 سطح مختلف تقسیم کرده است:

سطح 1: پایین‌ترین سطح تعریف شده در استاندارد است و مشخص‌کننده حداقل نیازمندی‌های امنیتی تعریف شده برای هر نرم‌افزارست؛ به این معنی که تمامی نرم‌افزارها باید حداقل نیازمندی‌های تعریف شده در این سطح را دارا باشند.

سطح 2: این سطح برای نرم‌افزارهایی با سطح حساسیت معمولی طراحی شده است. نرم‌افزارهای اداری یا مدیریتی از این نوع هستند. سطح 2 تمامی نیازهای سطح 1 را پوشش می‌دهد و علاوه بر آن تعدادی نیاز جدید که باید برای این سطح رعایت شود را الزامی می‌کند.

سطح 3: بالاترین سطح امنیتی این استاندارد است و برای نرم‌افزارهایی با حساسیت بالا در نظر گرفته شده است. نرم‌افزارهای بانکی یا نرم‌افزارهای وبی دارای داده‌های سری یا امنیتی در این سطح قرار می‌گیرند. این سطح نیازمندی‌های 2 سطح اول را داراست و علاوه بر آن مجموعه‌ای از نیازمندی‌های جدید را هم ارائه می‌کند.

این سطح‌بندی به تولیدکنندگان نرم‌افزار کمک کرده است بر اساس نرم‌افزاری که می‌خواهند تولید کنند چک‌لیستی از نیازمندی‌های امنیتی برای خود تعریف و آن را پیاده‌سازی کنند. مسلما هزینه تولید این نیازمندی‌ها در شرکت‌های نرم‌افزاری بسیار زیاد است و اگر این سطح‌بندی وجود نداشت هزینه بسیار زیادی برای بحث امنیت به وجود می‌آمد. با این سطح‌بندی شرکت‌های نرم‌افزاری می‌توانند بر اساس کارکرد نرم‌افزار به اندازه لازم و نه کمتر و نه بیشتر نرم‌افزارهای خود را امن کنند و مشتریان آنها نیز از امنیت بدست آمده رضایت کامل داشته باشند.

این استاندارد نیز در طول زمان با توجه به تغییرات در شیوه‌های حمله تغییر می‌کند و نسخه‌هایی جدید و کامل‌تر ارائه می‌شود. در حال حاضر که این مقاله در حال نوشتن است نسخه نهایی این استاندارد 3.0.1 است؛ اما با توجه به اینکه ارتباطی منطقی بین حملات و این استاندارد وجود دارد و ما در این مقاله نسخه 2013 مربوط به 10 حمله برتر را ارائه کردیم در مقاله بعدی، نسخه 3 این استاندارد مورد بررسی قرار می‌گیرد.

از منظر دیگری این استاندارد مجموعه نیاز مندی‌ها را به چند دسته‌بندی منطقی تقسیم کرده است. در واقع این استاندارد طبقه‌بندی‌ای برای نیازمندی‌ها ایجاد کرده است تا بتوان ارتباطی منطقی بین نیازمندی‌های مختلف ارائه شده در این استاندارد برقرار کرد. این طبقه‌بندی‌ها به شرح زیر هستند:

1-**معماری، طراحی و مدل‌سازی نخ‌ها**: نیازمندی‌های بیان شده در این طبقه‌بندی باید در زمان طراحی و معماری نرم‌افزار مدنظر قرار گرفته شوند.

2-**نیازمندی‌هایتایید احراز هویت**: نیازمندی‌های این طبقه‌بندی به صورت کامل در بحث

احراز هویت کاربرد دارند و فرایند سیاست گذاری، پیکربندی و اجرای یک احراز هویت امن را شامل می شود.

1-نیازمندی‌های تایید مدیریت نشست: نیازمندی‌های این طبقه‌بندی برای امن‌سازی نشست‌های ایجاد شده بین مرورگر کاربر و سرور وب نرم‌افزار کاربرد دارد. مجموعه این نیازمندی‌ها قابلیت‌هایی را برای مدیریت نشست‌ها در اختیار مدیران و کاربران قرار می‌دهد و همچنین فرایند مدیریت نشست را در نرم‌افزار امن می‌کند.

2-نیازمندی‌های تایید کنترل دسترسی: در این طبقه‌بندی نیازمندی‌هایی برای ایجاد، پیکربندی و اجرای مفهوم دسترسی به اطلاعات در نرم‌افزار آورده شده است.

3-نیازمندی‌های تایید بررسی کردن ورودی مخرب: در این طبقه‌بندی نیازمندی‌هایی مطرح می‌شود که بر اساس آن هر ورودی‌ای از سمت کاربران مورد بررسی امنیتی قرار گیرد و نسبت به ایجاد یا عدم ایجاد مشکل برای نرم‌افزار یا کاربران آن اعتبارسنجی شود.

4-نیازمندی‌های تایید رمزنگاری: در این طبقه‌بندی نیازمندی‌هایی برای پیاده‌سازی صحیح از الگوریتم‌های رمزنگاری، عملیات‌های رمزنگاری متقارن و نامتقارن و عملیات‌های تولید مقادیر تصادفی و استفاده صحیح از آنها در بخش‌های مختلف نرم‌افزار آورده شده است.

5-نیازمندی‌های تایید مدیریت خطا و ثبت وقایع: در این طبقه‌بندی نیازمندی‌هایی تعریف شده است که نرم‌افزار باید در حوزه مدیریت خطا آن ها را رعایت کند و

همچنین نیازمندی‌هایی برای ثبت وقایع روی داده شده در زمان های اتفاق رویدادهای مختلف در نرم‌افزار آورده شده است.

6-نیازمندی‌های تایید محافظت از داده: در این طبقه‌بندی نیازمندی‌هایی تعریف شده است که بر اساس آن از داده در مقابل افراد غیر مجاز محافظت شود، از یکپارچگی داده‌ها محافظت شود و همچنین از در دسترس بودن آنها برای افراد مجاز اطمینان حاصل شود.

7-نیازمندی‌های تایید امنیت ارتباطات: در این طبقه‌بندی با تمرکز بر مسیرهای ارتباطی بین سیستم کاربر و سرورهای نرم‌افزار و همچنین بین سرورهای نرم‌افزار با هم نیازمندی‌هایی برای برقراری امنیت در این مسیرهای ارتباطی معرفی شده است.

8-**نیازمندی‌های تایید پیکربندی امنیت HTTP:** در این طبقه‌بندی نیازمندی‌هایی تعریف شده است که بر روی مفاهیم و قواعد HTTP تمرکز دارد و سعی آن این است که راه‌های نفوذ این پروتکل را ببند و استفاده از آن را برای نرم‌افزارها امن کند.

9ـ**نیازمندی‌های تایید کنترل‌های مخرب:** نیازمندی‌های این طبقه‌بندی برای کنترل و مدیریت‌کردن فعالیت‌های مخرب در نرم‌افزار طراحی شده است. فعالیت‌هایی مانند کدهای مخرب موجود در نرم‌افزار یا معایب منطقی که باعث ایجاد حفره یا تخریب در نرم‌افزار می‌شود.

10-نیازمندی‌های تایید منطق کسب و کار: در این بخش نیازمندی‌هایی آورده می‌شود تا امنیت فرایندها و منطق کسب‌وکار تضمین شود. به عنوان مثال محدودیت‌هایی بر روی

تعداد دفعاتی که یک کاربر می‌تواند یک عملیات را در روز انجام دهد ایجاد می‌کند تا کاربران نتوانند با انجام عملیات‌های بسیار، منطق کسب‌وکار را خدشه‌دار کنند.

11ـ **نیازمندی‌های تایید منابع و فایل‌ها:** در این طبقه‌بندی نیازمندی‌هایی تعریف شده‌اند که تا اطمینان حاصل شود که به فایل‌ها و یا داده‌هایی که از سایر منابع ناامن به نرم‌افزار آمده اعتماد نشده و حتما بررسی‌های امنیتی بر روی این گونه داده‌ها انجام می‌شود.

12– **نیازمندی‌های تایید موبایل:** نرم‌افزارهای موبایلی مانند مرورگرهای کاربران هستند و از همان زیرساخت وب برای انجام کارهای خود با سرور وب ارتباط برقرار می‌کنند. در این طبقه‌بندی مجموعه‌ای از نیازمندی‌ها با تمرکز بر روی نرم‌افزارهای موبایلی و سرورهای وبی که به این نرم‌افزارها خدمات می‌دهند به منظور بالا بردن امنیت این نرم‌افزارها آورده شده است.

13ـ **نیازمندی‌های تایید سرویس‌های وب:** سرویس‌های وب نیز مانند نرم‌افزارهای موبایلی از زیرساخت وب به منظور خدمات‌رسانی به سایر نرم‌افزارها استفاده می‌کنند. کار این سرویس‌ها بازگرداندن اطلاعات مورد نیاز سایر نرم‌افزارهاست و در حال حاضر بر اساس مفاهیم رایانش ابری بسیار زیاد مورد استفاده قرار می‌گیرند. در این طبقه‌بندی نیازمندی‌های تامین امنیت این سرویس‌ها بیان شده است.

14-پیکربندی: در این طبقه‌بندی که آخرین طبقه‌بندی بیان شده در این استاندارد است نیازمندی‌هایی تعریف شده است که به پیکربندی اجزای نرم‌افزار و تنظیمات امنیتی نرم‌افزار اشاره دارد. نیازمندی‌هایی درباره اینکه تمامی اجزای نرم‌افزار به روز باشد،

تنظیمات پیشفرض نرم‌افزار امن باشند و اینکه تغییردادن تنظیمات نرم‌افزار به تنظیمات پیشفرض، امنیت نرم‌افزار را در معرض خطر قرار ندهد.

این استاندارد به صورت کامل بر اساس روش پیشگیری ایجاد نشده است؛ بلکه بعضی از نیازمندی‌های تعریف شده در استاندارد مربوط به روش کشف تلاش‌هاست. به عنوان مثال در ضبط وقایع رخ داده در نرم‌افزار نیازمندی‌هایی برای ثبت داده‌های ممیزی دارد و نه صرفا ثبت رویداد؛ یعنی علاوه بر اینکه باید بیان شود چه رویدادی رخ داده باید بیان شود بر روی چه موجودیتی این رویداد اتفاق افتاده و باعث چه تغییری در آن شده است.

به عنوان کسی که بسیاری از نیازمندی‌های این استاندارد را به قابلیت‌هایی از نرم‌افزار تبدیل کرده‌ام می توانم بگویم که این استاندارد در عین حال که ساده، شفاف و کاربردی است، تکلیف و وظایف همه بخش‌های یک تولیدکننده نرم‌افزار را مشخص می‌کند برای بستر وب جامع‌ترین استانداردی است که وجود دارد. این استاندارد می‌تواند تعاریف امنیت را برای همه ذینفعان درگیر در بحث امنیت مشخص کند و هم راهنما و هم اعتبارسنج خوبی برای هر ذینفعی باشد. سادگی تعاریف این استاندارد کمک می‌کند پیچیدگی‌های امنیت نرم‌افزار به خوبی طبقه‌بندی شود ولی در عین حال تاثیرگذاری بسیار زیادی بر روی امنیت در دنیای وب دارد. کاربردی بودن این استاندارد برای ایجاد امنیت برخلاف معمول سایر استانداردها است، معمولا استانداردها به دلیل پیچیدگی‌های زیاد در تعاریفی که بیان می‌کنند خیلی کم کاربردی می‌شوند و حتی در خیلی از مواقع نیاز است موارد تعریف شده در استانداردها تفسیر شوند تا بتوان بر اساس تفاسیر صورت گرفته شیوه‌های کاربردی را تولید کرد؛ اما در رابطه با این استاندارد کاربردی

بودن به حد خیلی خوبی رعایت شده است و عملا می توان از این استاندارد به عنوان نقشه راه امنیتی برای نرم‌افزارهای بستر وب استفاده نمود.

نتیجه‌گیری

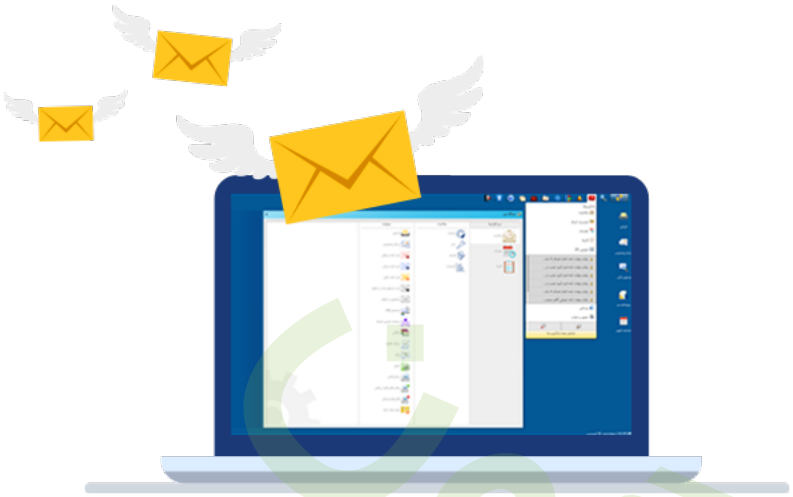
در این بخش از سری مقالات امنیت در وب ابتدا به صورت مفصل درباره مفهوم روش پیشگیری در بحث امنیت صحبت شد و بعد از آن روش‌های مختلف حمله در این بستر بیان شد و به ازای آنها سعی شد تا تاثیرگذارین بر بخش‌های مختلف از معماری استقرار نرم‌افزار مشخص و وظایف تولیدکننده و استفاده‌کننده از نرم‌افزار بیان شود. در نهایت نیز استانداردی برای این شیوه نگاه به امنیت در وب معرفی شد.

باید در نظر داشت که در دنیای امروز وب، شیوه پیشگیری یک اصل انکارناپذیر از امنیت نرم‌افزارست و نیاز است به عنوان فرهنگ قالب در تمامی شرکت‌های تولیدکننده چه شرکت های بزرگ چند صد یا چند هزار نفری و چه شرکت‌های استارت آپ کوچک 5 تا 10 نفری مورد توجه ویژه قرار گیرد. این رویکرد باعث می‌شود اطمینان از دنیای وب در میان استفادهکنندگان آن نه تنها تخریب نشود که ارتقا نیز پیدا کند و امنیت به معنای واقعی کلمه برای هر دو طرف تولیدکننده و استفاده‌کننده ایجاد شود.

در بخش بعدی به رویکرد کشف تلاش‌ها خواهیم پرداخت و «استاندارد معیار مشترک» را به عنوان نمونه‌ای از این نوع رویکرد معرفی خواهیم کرد.

پایان بخش دوم

درخواست دموی نرم‌افزارهای دیدگاه



درخواست دمو
حضوری و آنلاین