عنوان مقاله:

A Large-scale Empirical Study of Security Patches

هدف از پژوهش:

هدف از این پژوهش مطالعه‌ی وصله[1] های امنیتی، شامل تحلیل چرخه حیات توسعه‌ی آن‌ها و مقایسه‌ی ویژگی‌های آن‌ها با وصله‌های غیرامنیتی، است. این مطالعه می‌تواند مخاطبان را از کاستی‌های فرایندهای فعلی وصله بندی آگاه سازد و بینشی برای بهبود آن‌ها ایجاد کند.

روش حل مسئله:

برای انجام این پژوهش مجموعه‌ای از اطلاعات در مورد آسیب‌پذیری‌ها و وصله‌های آن‌ها باید جمع‌آوری شود تا بررسی‌ها و تحلیل‌های لازم روی آن‌ها صورت بگیرد. NVD[2] یک پایگاه‌داده‌ی رایگان از آسیب‌پذیری‌ها است. در این پژوهش ضمن استفاده از داده‌های موجود در NVD، اطلاعات بیشتری نیز استخراج شده و مورداستفاده قرارگرفته‌است. در NVD برای هر آسیب‌پذیری یک رکورد وجود دارد که شامل اطلاعات خلاصه‌شده‌ای از آن آسیب‌پذیری است. در NVD شدت[3] آسیب‌پذیری‌ها با استفاده از استانداردCVSS[4] مشخص شده‌است. استاندارد CVSS برای تعیین شدت هر آسیب‌پذیری، ریسک و تأثیر بالقوه‌ی آن را در نظر می‌گیرد. به‌علاوه، NVD برای دسته‌بندی و تعیین کلاس هر یک از ضعف‌های امنیتی از استاندارد CWE[5] استفاده می‌کند. در NVD در هر رکورد، به تعدادی مرجع خارجی[6] لینک داده‌شده‌است. این مراجع خارجی معمولاً گزارشات امنیتی یا repository هایی هستند که آن آسیب‌پذیری را برطرف کرده‌اند که در اغلب موارد آن repository، Git است. در این پژوهش با استفاده از این مراجع خارجی به repository های source codeها دسترسی پیداکرده و وصله‌های امنیتی آسیب‌پذیری‌ها جمع‌آوری شده‌است. برای آنکه بتوان ویژگی‌های وصله‌های امنیتی را به دست آورد، ابتدا باید مشخص کرد که کدام‌یک از وصله‌ها امنیتی و کدام‌یک غیرامنیتی هستند. بدین منظور از رگرسیون لجستیک[7] استفاده شده‌است و n-gram ها در commit message های Git مدل شده‌اند. به‌علاوه، برای همه‌ی وصله‌ها ورژن‌های فایل‌های مختلف تغییریافته، قبل و بعد از commit، استخراج شدند. همچنین برای اینکه تنها کدهایی که functionality دارند مورد تحلیل قرار بگیرند کامنت‌ها و فایل‌های غیر کدی فیلتر شدند. در NVD برای هر رکورد، تاریخی که آن آسیب‌پذیری در NVD منتشر شده، ثبت شده‌است که این تاریخ با تاریخ افشای عمومی آسیب‌پذیری لزوماً یکی نیست. در این پژوهش، برای پیدا کردن تاریخ افشای عمومی آسیب‌پذیری‌ها، مراجع خارجی رکوردها مورد تحلیل

[1] Patch
[2] National Vulnerability Database
[3] Severity
[4] Common Vulnerability Scoring System
[5] Common Weakness Enumeration
[6] External Reference
[7] Logistic regression

قرار گرفته‌اند. تاریخ افشای عمومی هر آسیب‌پذیری زودترین تاریخی است که در مراجع خارجی و NVD برای آن آسیب‌پذیری ثبت شده‌است.

نتایج:

به مدت‌زمانی که یک آسیب‌پذیری در code base باقی می‌ماند تا اصلاح شود، طول عمر آسیب‌پذیری گفته می‌شود. طول عمر آسیب‌پذیری‌ها معمولاً زیاد است. بیش از نیمی از آسیب‌پذیری‌ها طول عمری بیش از ۱۴,۴ ماه دارند. طول عمر یک‌سوم آسیب‌پذیری‌ها بیش از سه سال است. تنها ۶,۵ درصد آسیب‌پذیری‌ها طول عمرشان کمتر از ده روز است. رابطه‌ای بین شدت یک آسیب‌پذیری و طول عمر آن وجود ندارد. طول عمر آسیب‌پذیری‌ها بسته به اینکه در کدام کلاس از ضعف‌های امنیتی قرار دارند بسیار متفاوت است. به‌عنوان‌مثال آسیب‌پذیری‌های مربوط به وب طول عمر بسیار کوتاه‌تری نسبت به خطاهای مدیریت حافظه دارند.

اکثر آسیب‌پذیری‌ها (۷۸,۸ درصد) قبل از آن‌که افشای عمومی شوند، اصلاح می‌شوند. بیش از یک‌سوم آسیب‌پذیری‌ها پس از یک ماه از اصلاح آن‌ها، هنوز افشای عمومی نشده‌اند. عدم اطلاع اشخاص متأثر از آن آسیب‌پذیری شرایط را برای سوءاستفاده‌ی مهاجمان فراهم می‌کند. درصد کمی از آسیب‌پذیری‌ها (۲۱,۲ درصد) قبل از افشای عمومی اصلاح‌نشده‌اند که این شرایط نیز به مهاجمان زمان کافی را برای حمله می‌دهد. شدت یک آسیب‌پذیری روی این‌که قبل از افشای عمومی اصلاح شود یا پس‌ازآن، به‌طور قابل‌توجهی تأثیرگذار است. ۸۸,۱ درصد از آسیب‌پذیری‌هایی که شدت بالایی دارند قبل از افشای عمومی اصلاح می‌شوند. به‌علاوه توسعه‌دهندگان، آسیب‌پذیری‌هایی که شدت بالاتری دارند را سریع‌تر اصلاح می‌کنند (آسیب‌پذیری‌هایی که قبل از افشای عمومی اصلاح‌نشده‌اند).

وصله‌ها ممکن است منجر به ایجاد اختلال در عملکرد کد یا ایجاد خطاهای جدید در آن شوند. حدود ۷ درصد از وصله‌های امنیتی آسیب‌پذیری را به‌طور کامل برطرف نکرده‌اند و برای اصلاح کامل به وصله‌ی دیگری نیاز است که این فرایند حدود نصف یک سال زمان می‌برد. حدود ۵ درصد از وصله‌ها باعث به وجود آمدن خطاهای جدید می‌شوند که اصلاح آن حدود یک ماه طول می‌کشد.

وصله‌ها الزاماً همیشه در کد برنامه تغییر ایجاد نمی‌کنند بلکه ممکن است تغییرات را در بخش‌های دیگر مانند پیکربندی‌ها، تنظیمات، کتابخانه‌ها و مستندات اعمال کنند که این مسئله برای وصله‌های غیرامنیتی محتمل‌تر است. ۶,۱ درصد وصله‌های غیرامنیتی و ۱,۳ درصد وصله‌های امنیتی هیچ تغییری در کد برنامه ایجاد نمی‌کنند. وصله‌های غیرامنیتی به‌طور قابل‌توجهی از وصله‌های امنیتی بزرگ‌تر و پیچیده‌تر (بر اساس معیار LOC[8]) هستند، تغییرات منطقی بیشتری ایجاد می‌کنند و تمایل بیشتری به تغییر اندازه‌ی code base دارند. وصله‌های امنیتی در اعمال تغییرات، محلی‌تر از وصله‌های غیرامنیتی عمل می‌کنند و فایل‌های source code و توابع کمتری را تغییر می‌دهند. این ویژگی‌ها ممکن است برای تولید خودکار و سریع وصله‌های امنیتی و تحلیل امنیت آن‌ها بتوانند مورداستفاده قرار بگیرند.

[8] Lines of Code