ارزیابی تاب‌آوری برنامه کاربردی وب در برابر حملات منع‌خدمت سیلابی در لایه کسب و کار

میترا علی‌دوستی[۱]، دانشجو دکتری؛ علیرضا نوروزی[۲]، استادیار؛ احمد نیک‌آبادی[۳]، استادیار

۱- مجتمع فناوری اطلاعات، ارتباطات و امنیت - دانشگاه صنعتی مالک اشتر - تهران - ایران – Alidoosti@mut.ac.ir
۲- مجتمع فناوری اطلاعات، ارتباطات و امنیت - دانشگاه صنعتی مالک اشتر - تهران - ایران – Nowroozi@mut.ac.ir
۳- دانشکده مهندسی کامپیوتر - دانشگاه صنعتی امیرکبیر - تهران - ایران - Nickabadi@aut.ac.ir

چکیده: طبق گزارش IMPERVA حملات منعقد در لایه کاربرد، حدود ۶۰ درصد از کل حملات منع خدمت را تشکیل می‌دهند. امروزه حملات به لایه کسب و کار منتقل شده‌اند. ابزارهای تحلیل آسیب‌پذیری قادر به شناسایی آسیب‌پذیری‌های لایه کسب وکار (آسیب‌پذیری‌های مربوط به منطق) برنامه‌کاربردی وب نیستند. در این تحقیق راهکار جعبه سیاه برای شناسایی آسیب‌پذیری لایه کسب و کار برنامه‌کاربردی وب در مقابل حملات منع‌خدمت سیلابی را با نام BLDAST پیشنهاد می‌دهیم. هدف BLDAST ارزیابی تاب‌آوری برنامه‌کاربردی وب در برابر حملات منع‌خدمت سیلابی در لایه کسب وکار است. BLDAST ابتدا فرایندهای کسب و کار برنامه را استخراج می‌نماید سپس فرایندهای کسب و کار سنگین را انتخاب می‌کند و در نهایت، سناریوی آزمون منع‌خدمت لایه کسب و کار را اجرا می‌کند. آزمایش‌ها بر روی چهار برنامه‌کاربردی معروف نشان داد، BLDAST قادر است آسیب‌پذیری‌های لایه کسب و کار این برنامه‌ها را شناسایی کند. علاوه بر این نشان دادیم که مهاجم در حملات لایه کسب و کار می‌تواند با مصرف تنها یک در صد از منابع خود در قیاس با حملات لایه‌های دیگر، سیستم هدف را شکست دهد. بنابراین حملات لایه کسب وکار بسیار خطرناک هستند که BLDAST قادر به شناسایی آسیب‌پذیری برنامه‌های‌کاربردی در برابر این حملات است.

واژه‌های کلیدی: آزمون جعبه سیاه، لایه کسب و کار، فرایند کسب و کار، سناریو آزمون منع‌خدمت.

Assessing of Web Application Resiliency against Flooding DoS Attacks in the Business Layer

M. Alidoosti[1], PhD Student; A. Nowroozi[2], Assistant Professor; A. Nickabadi[3], Assistant Professor

1- Faculty of ICT, Malek-Ashtar university of technology, Tehran, Iran, Email:alidoosti@mut.ac.ir
2- Faculty of ICT, Malek-Ashtar university of technology , Tehran, Iran, Email:Nowroozi@mut.ac.ir
3-Faculty of Computer Engineering, Amirkabir University of Tehran, Tehran, Iran, Email: Nickabadi@aut.ac.ir

Abstract: According to IMPERVA report, application layer DoS attacks have involved about 60 percent of total DoS attacks. Today, attacks have been transferred to the business layer. Web application vulnerability scanners cannot detect business logic vulnerabilities (vulnerabilities related to logic). This paper presents BLDAST, A dynamic and black-box vulnerability analysis approach that identify business logic vulnerabilities of a web application against flooding DoS attacks. BLDAST assesses web application resiliency against flooding DoS attacks in the business layer. BLDAST first extracts business logic processes of a web application. Business logic processes with high overload are selected and finally, based on selected processes, BLDAST runs business layer DoS test scenarios. The evaluation conducted on four well-known open source web applications shows that BLDAST is able to detect business logic vulnerabilities. In addition, we show that an attacker in business logic attacks can exhaust target by consuming only one percent of his resources in comparison to other layers attacks. Therefore, business logic attacks are very dangerous and BLDAST is able to identify vulnerable web applications against these attacks.

Keywords: Black-box testing, Business layer, Business logic process, DoS test Scenario.

تاریخ ارسال مقاله: ۱۳۹۶/۰۷/۲۸
تاریخ اصلاح مقاله: ۱۳۹۶/۱۲/۰۲
تاریخ پذیرش مقاله: ۱۳۹۷/۰۲/۱۷
نام نویسنده مسئول: علیرضا نوروزی
نشانی نویسنده مسئول: ایران - تهران – دانشگاه صنعتی مالک اشتر - مجتمع فناوری اطلاعات، ارتباطات و امنیت

۱- مقدمه

برنامه‌های کاربردی وب ساده‌ترین روش ارائه خدمت به کاربران است. نیاز روزافزون به برنامه‌های کاربردی، امنیت برنامه‌های کاربردی را به موضوعات پرطرفدار تبدیل کرده‌است. طبق گزارش ورایزن حدود ۴۰٪ از رخدادهای امنیتی در سال ۲۰۱۶ به دلیل حملات برنامه‌های کاربردی وب بوده‌اند. حملات برنامه‌های کاربردی وب رتبه یک منبع رخنه داده در گزارش ورایزن را دارند. آسیب‌پذیری‌های حوزه وب، اغلب آسیب پذیری‌های گزارش شده در پایگاه داده CVE[۱] را شامل می‌شوند [۱]. در سال ۲۰۱۵ تعداد رخنه‌های امنیتی حدود ۳۵/۵٪ نسبت به سال‌های گذشته افزایش یافته است و اغلب این حملات، به برنامه‌های کاربردی نظامی، تجارت الکترونیک و پزشکی صورت گرفته‌اند [۲]. آسیب‌پذیری های منطقی در دسته آسیب‌پذیری‌های قدرتمند رتبه‌بندی شده‌اند که امنیت برنامه‌های کاربردی را تحت تأثیر قرار می‌دهند [۱].

پویشگرهای خودکار، به دلیل عدم فهم منطق برنامه، قادر به شناسایی آسیب‌پذیری‌های منطق کسب و کار برنامه نیستند. این‌گونه آسیب‌پذیری‌ها تنها از طریق آزمون دستی قابل شناسایی هستند و متکی بر خلاقیت و مهارت فرد آزمون‌گر هستند. شناسایی آسیب‌پذیری‌های منطقی بسیار سخت است و در صورت سوء استفاده، خسارت زیادی را به‌جای می‌گذارد [۱].

تعریف رسمی برای آسیب‌پذیری‌های منطقی وجود ندارد [۳]. فهم منطق برنامه کاربردی، برای ابزارهای خودکار دشوار است بنابراین شناسایی آسیب‌پذیری‌های منطقی بر عهده فرد آزمونگر است و ازآنجایی که آسیب‌پذیری‌های منطقی، مختص برنامه کاربردی است، شناسایی این نوع آسیب‌پذیری‌ها، دشوار است.

برنامه‌های کاربردی مستند رسمی ندارند که حالات داخلی و رفتار مورد انتظار کاربر را توصیف کنند. عدم وجود چنین مستندی، شناسایی آسیب‌پذیری‌های منطقی را دشوار کرده است. به عنوان مثال افزودن مکرر یک کالای خاص به سبد خرید، یک ویژگی معمول برنامه‌های خرید الکترونیک است اما استفاده چندین باره از کد تخفیف، یک آسیب‌پذیری منطقی است. یک انسان به راحتی تفاوت این دو سناریو را متوجه می‌شود در صورتی که یک پویشگر بدون داشتن یک مدل مناسب برای برنامه کاربردی، قادر به تفکیک این دو سناریو نیست [۳].

اخیراً تحقیق‌هایی برای شناسایی خودکار آسیب‌پذیری‌های منطقی انجام شده‌است [۶-۴]. متأسفانه این روش‌ها برای برنامه‌های کوچک، کاربرد دارند. علاوه براین، کد منبع برنامه کاربردی برای تولید مدل مناسب از برنامه لازم است تا بتوان موارد آزمون را تولید نمود [۳]. بنابراین امروزه نیاز به ابزار خودکار برای شناسایی آسیب‌پذیری منطقی ملموس است. در این تحقیق به آسیب‌پذیری‌های منطقی، آسیب‌پذیری های لایه کسب وکار گفته می‌شود.

حمله منع خدمت همچنان به عنوان یک تهدید وجود دارد [۷]. حملات منع خدمت درطی سال‌های گذشته از لایه‌های پایین مدل OSI[۲] به سمت لایه‌های بالا و لایه کاربرد منتقل شده‌اند. امروزه حملات به لایه کسب و کار منتقل شده‌اند که در واقع به عنوان لایه‌ای در بالای مدل OSI در نظر گرفته شده‌است [۸]. ارتقا حملات منع‌خدمت توسط مهاجم از لایه‌های پایین مدل OSI به لایه کسب و کار، مهاجم را قادر می‌سازد تا با صرف توان کمتری، حملات مخرب‌تری را فراهم آورد. به دلیل ترافیک تولیدی کمتر، شناسایی این نوع حملات نیز دشواتر است.

حملات منع‌خدمت در لایه کسب و کار منجر به اتلاف منابع در سیستم قربانی می‌شود. به طور معمول مهاجم با ارسال یک یا تعداد کمی درخواست، منجر به اعمال بار محاسباتی غیرمعمول در منابع سیستم قربانی مانند پردازنده، حافظه و غیره می‌شود. حملات منع‌خدمت در لایه کسب وکار دارای ترافیک بدخواه و حجیم نیستند. در واقع برنامه کاربردی آسیب‌پذیر به حملات منع‌خدمت در لایه کسب وکار، عملکرد طبیعی خود را دارد ولی متاسفانه توسعه‌دهندگان، متوجه آسیب‌پذیری کد برنامه، درمقابل حملات منع‌خدمت نیستند. دلیل اصلی اینگونه آسیب‌پذیری، نقص در طراحی برنامه است و ناشی از خطای برنامه‌نویسی نیست.

در این تحقیق تاب‌آوری[۳] برنامه کاربردی وب در برابر حملات منع‌خدمت سیلابی در لایه کسب و کار ارزیابی می‌گردد. بدین منظور راهکار جعبه سیاه برای آزمون امنیتی پویای برنامه در لایه کسب وکار را پیشنهاد می‌دهیم. این راهکار به اختصار BLDAST[۴] نامیده می‌شود. BLDAST آسیب‌پذیری‌های لایه کسب و کار برنامه کاربردی وب در مقابل حملات منع‌خدمت سیلابی را شناسایی می‌کند. سناریوهای آزمون پیشنهادی، از زمینه یا به عبارت دیگر، کسب و کار برنامه کاربردی مطلع هستند. روش ارائه شده، مستقل از فنّاوری به کار رفته در برنامه است و به صورت خودکار به شناسایی آسیب‌پذیری می‌پردازد. علاوه بر این نشان داده می‌شود آسیب‌پذیری‌های لایه کسب و کار، می‌توانند تهدیدآمیزتر از آسیب‌پذیری‌های سایر لایه‌ها باشند زیرا توان مهاجم برای اجرای حملات لایه کسب و کار در حدود یک صدم توان برای اجرای حملات دیگر است. منظور از توان میزان مصرف منابع سیستم مهاجم برای اعمال حملات است.

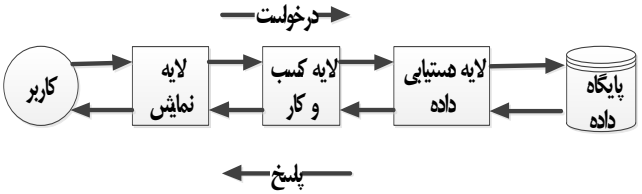
نوآوری‌های موجود در این تحقیق عبارتند از:

۱. تعریف آسیب‌پذیری لایه کسب و کار، حمله لایه کسب و کار و حمله منع خدمت در لایه کسب و کار
۲. ارائه راهکار جعبه سیاه برای آزمون امنیتی پویای برنامه های کاربردی وب به منظور شناسایی آسیب‌پذیری‌های لایه کسب و کار در برابر حملات منع‌خدمت سیلابی
۳. اثبات خطرناک بودن حملات لایه کسب و کار در قیاس با حملات لایه‌های دیگر به دلیل صرف تنها یک درصد از منابع برای شکست دادن سیستم هدف

در این تحقیق در بخش ۲ به مروری بر ادبیات موضوع و پژوهش های مرتبط، در بخش ۳ تعریف آسیب پذیری لایه کسب و کار و همین طور روش پیشنهادی، در بخش ۴ پیاده‌سازی و ارزیابی و در نهایت نتیجه‌گیری بیان می‌شود.

۲- مروری بر ادبیات موضوع و تحقیقات مرتبط

لایه کسب و کار، منطق برنامه کاربردی را مشخص می‌کند. در واقع هم با استخراج، پردازش و مدیریت داده‌ها سر و کار دارد و هم با قوانین و سیاست‌های کسب و کار برنامه کاربردی. علاوه براین داده‌های ورودی را نیز اعتبار سنجی می‌کند [۱]. درشکل ۱ جایگاه لایه کسب و کار در برنامه کاربردی نشان داده شده‌است.



شکل ۱: معماری سه لایه برنامه کاربردی [۱]

لایه دوم، لایه کسب و کار است که قوانین کسب و کار را مورد پردازش قرار می‌دهد و داده‌های دریافتی از لایه نمایش را در اختیار لایه دستیابی داده قرار می‌دهد. بعد از دریافت داده از کاربر، داده در اختیار لایه کسب و کار قرار می‌گیرد. برنامه کاربردی از این داده برای اجرای فرایند کسب و کار استفاده می‌کند. هر فرایند کسب و کار دارای چندین مرحله است که باید به ترتیب مقتضی اجرا شود و ممکن است فرایندها به طور سازماندهی شده‌ای با هم در تعامل باشند.

آسیب‌پذیری لایه کسب و کار یک نقص در لایه کسب و کار برنامه کاربردی است. بردار حملات منطقی در لایه کسب وکار، درخواست‌های متعدد قانونی هستند که دارای مقادیر ورودی قانونی هستند. این گونه بردارها از کسب و کار برنامه کاربردی به منظور ایجاد خرابی، سوء استفاده می‌کنند و به کسب و کار برنامه کاربردی خسارت وارد می‌کنند [۹].

حمله منع خدمت در لایه کسب و کار، با اطلاع از منطق کسب و کار برنامه صورت می‌گیرد و به دلیل ضعف در طراحی برنامه و یا با ایجاد حالت مسابقه، مانع دسترسی کاربران قانونی به برنامه می‌شود. به عنوان مثال در سیستم خرید بلیت هواپیما، مهاجم می‌تواند تمام بلیت‌های موجود را به قصد خرید انتخاب نماید. در این مدت هیچ کاربر دیگری به دلیل پر شدن بلیت‌های موجود قادر به خرید بلیت نیست. به عنوان مثال دیگر، قفل کردن کاربر مورد نظر با سه بار اشتباه وارد کردن رمز عبور وی، در این صورت کاربر مربوطه قادر به ورود نیست [۱۰].

حمله منع‌خدمت در لایه کسب و کار با حمله منع‌خدمت در لایه کاربرد متفاوت است. در حمله منع خدمت در لایه کسب و کار، مهاجم با آگاهی از فرایندهای کسب و کار برنامه، حمله می‌کند در واقع فرایندهای کسب و کاری که دارای سربار محاسباتی بالایی هستند، جهت انتخاب برای اعمال حمله اولویت بالایی دارند. مهاجم از ویژگی و یا عملکرد خاصی از برنامه استفاده می‌کند با این هدف که آن ویژگی و یا عملکرد را غیرفعال کند. در حمله منع سرویس در لایه کسب و کار، ترافیک تولیدی حمله بسیار کمتر از حمله منع سرویس در لایه کاربرد است و مهاجم با توان کمتری نسبت به حمله در لایه کاربرد، قادر به تولید حمله است.

۲-۱- پژوهش‌های مرتبط

۲-۱-۱- حملات منطقی (حملات لایه کسب و کار)

دو روش برای جلوگیری از حملات منطقی (حملات لایه کسب و کار) وجود دارد: ۱- شناسایی زمان اجرای حملات منطقی (روش دفاعی) ۲- شناسایی آسیب‌پذیری‌های منطقی موجود در برنامه‌کاربردی (روش پیشگیرانه). در روش دفاعی، رفتار برنامه زیر نظر گرفته می‌شود در صورت خروج از حالت نرمال، حمله محسوب می‌شود. در روش پیشگیرانه، از بردارهای حمله برای شناسایی آسیب‌پذیری منطقی استفاده می‌شود. روش پیشگیرانه به دو روش جعبه سیاه و جعبه سفید تقسیم می‌شود.

BLOCK [۱۱] و Swaddler [۱۲] از روش دفاعی برای جلوگیری از حملات منطقی استفاده می‌کنند.BLOCK [۱۱] ابتدا مدل رفتاری برنامه را با مشاهده تعامل کاربر با برنامه کاربردی به دست می‌آورد و مجموعه‌ای از ثابت‌ها از توالی درخواست/پاسخ‌ها و متغیرهای نشست را استخراج می‌نماید.BLOCK هر درخواست یا پاسخی که ثابت‌های شناسایی شده را نقض کنند، به عنوان حمله شناسایی می‌کند.

Swaddler [۱۲] برای شناسایی حملات روشی بر مبنای شناسایی ناهنجاری ارائه می‌کند. به عبارت دیگر حالت داخلی برنامه در مرحله یادگیری نظارت می‌شود و متغیرهای نرمال حالت برنامه، استخراج می‌شود که مشخص کننده نمایه هستند. سپس در مرحله شناسایی، حالت‌های ناهنجاری شناسایی می‌شوند.

SENTINEL [۱۳] و پلگرینو [۳] با استفاده از روش پیشگیرانه و به صورت جعبه سیاه به شناسایی آسیب‌پذیری منطقی می‌پردازند. SENTINEL [۱۳] به صورت پیشگیرانه ضعف‌های منطقی دسترسی به پایگاه داده، را شناسایی می‌کند. پرسمان‌هایی را شناسایی می‌کند که ویژگی‌های شناخته شده را نقض می‌کنند. هر پرسمان مخربی که ثابت های شناسایی شده را نقض می‌کنند، به عنوان حمله شناسایی می‌شود.

پلگرینو و همکارانش [۳] به صورت خودکار تعدادی الگوی رفتاری از ترافیک کاربران برنامه‌کاربردی استخراج می‌نمایند سپس سناریوهای آزمونی را برای بررسی الگوهای رفتاری شناسایی شده، طراحی می‌کنند. در [۱۴] مدلی بهبود یافته و سه لایه برای طراحی سطح منطقی پایگاه داده تحلیلی، ارائه شده‌است. در [۱۵] روشی برای پیش‌بینی درخواست آتی کاربران با استفاده از مدل مارکوف ذکر گردیده‌است.

۲-۱-۲- دفاع در برابر حملات منع‌خدمت لایه کاربرد

حملات منع خدمت در لایه کاربرد با مصرف سریع منابع کارگزار، خدمت رسانی به کاربران قانونی را دچار اختلال می‌کنند [۱۶]. آسیب‌پذیری‌های منع‌خدمت در لایه کاربرد معمولاً منجر به عدم پاسخ و یا از کار افتادن نرم‌افزار می‌شود. بدین صورت که مهاجم با ارسال

ورودی‌های مخرب، سیستم را خاموش می‌کند [۱۷]، سیستم را در یک حلقه بی پایان قرار می‌دهد و یا منجر به فراخوانی‌های بازگشتی با پیچیدگی فوق خطی [۱۸، ۱۹] می‌شود.

تحلیل‌های پویا برای شناسایی آسیب‌پذیری‌های منع‌خدمت در لایه کاربرد، ورودی‌هایی تولید می‌کنند که منجر به بدترین حالت زمان اجرا، می‌شوند و یا برنامه را وارد حلقه بی‌پایان می‌کنند [۲۱-۱۹]. تحلیل پویای برنامه‌های بزرگ، کار دشواری است و گاهی نمی‌توان ورودی‌هایی تولید کرد که بدترین زمان اجرا را داشته باشد [۲۲].

تحلیل ایستا به منظور شناسایی آسیب‌پذیری‌های منع‌خدمت در لایه کاربرد، قطعه کدهایی از برنامه با پیچیدگی بالا را شناسایی می‌کنند که وابسته به ورودی کاربر هستند و مهاجم با اعمال ورودی موردنظر به هدف خود [۱۸، ۲۳]. SAFER [۱۸، ۲۳] ابزار تحلیل ایستا برای شناسایی آسیب‌پذیری منع‌خدمت قبل از استقرار برنامه است که منجر به حملات اتلاف منابع می‌شوند. معمولاً مهاجم با ارسال یک یا تعداد کمی درخواست موجب اعمال بار محاسباتی بالا بر روی منابع داخلی سیستم مانند پردازنده و حافظه پشته[۵] می‌شود.

۳- آزمون امنیتی پویای لایه کسب و کار

۳-۱- تعریف آسیب پذیری لایه کسب و کار

در این بخش ابتدا تعاریف موجود برای آسیب‌پذیری لایه کسب و کار بیان می‌گردد سپس روش پیشنهادی به صورت سطح بالا تعریف می شود. در نهایت آسیب‌پذیری لایه کسب و کار، حمله لایه کسب و کار و آسیب‌پذیری منع‌خدمت در لایه کسب و کار تعریف می‌گردد.

چند تعریف از آسیب‌پذیری لایه کسب وکار تاکنون ارائه شده است است [۳، ۶، ۲۷-۲۵]. تعاریف قبلی به بخشی از آسیب‌پذیری‌های لایه کسب و کار پرداختند و بخشی از موضوع را تعریف نمودند. تعاریف قبلی در سطح برنامه کاربردی هستند در صورتی که تعریف ارائه شده در این تحقیق، در سطح لایه کسب و کار است. در ادامه به چند مورد از تعاریف آسیب‌پذیری لایه کسب و کار و نقاط ضعف آن‌ها پرداخته می‌شود.

پلگرینو [۳] آسیب‌پذیری منطقی را در نتیجه اعتبار سنجی نامناسب فرایند کاری برنامه‌کاربردی می‌داند که هم در سطح کنترل (به عنوان مثال حرکت بین صفحات مختلف) و هم در سطح داده (به عنوان مثال جریان داده‌ای که پارامترهای صفحات مختلف را به هم مرتبط می‌کند) ممکن است اتفاق بیفتد.

لی [۲۵]، کاودون [۶] و استرگیوپلوس [۲۶] آسیب‌پذیری‌های منطقی را نتیجهٔ اشتباه در منطق برنامه می‌دانند و به دلیل اینکه آسیب‌پذیرهای منطقی مختص عملکرد برنامه هستند، بیان تعریف جامعی از آن‌ها را دشوار می‌دانند.

OWASP[۲۷] تعریف بهتری برای آسیب‌پذیری منطقی نسبت به تعاریف موجود ارائه کرده‌است. طبق تعریف OWASP آسیب‌پذیری منطقی به مهاجمین امکان سوء استفاده از برنامه با استفاده از قوانین کسب و کار را می‌دهد. آسیب‌پذیری‌های منطقی از جریان کاری برنامه به منظور ایجاد تأثیر منفی به برنامه استفاده می‌نماید [۲۷].

در ادامه آسیب‌پذیری لایه کسب و کار، حمله لایه کسب و کار و حمله منع خدمت در لایه کسب و کار تعریف می‌گردد. به منظور تعریف آسیب پذیری لایه کسب و کار، مفاهیم لایه کسب و کار، آسیب‌پذیری و آسیب‌پذیری لایه کسب و کار را به ترتیب از مراجع [۲۸، ۲۹، ۳۰] بیان می‌کنیم.

تعریف ۱ (لایه کسب و کار) [۲۸].

لایه کسب و کار: در این لایه به سرویس‌های کسب و کار سیستم، پرداخته می‌شود.

تعریف ۲ (آسیب‌پذیری) [۲۹]. آسیب‌پذیری، نقص نرم‌افزاری است که می‌تواند مخرب باشد و مورد سوء استفاده قرار بگیرد.

بر اساس تعریف ۱ و ۲، تعریف ۳ ارائه می‌شود.

تعریف ۳ (آسیب‌پذیری لایه کسب و کار). آسیب‌پذیری لایه کسب و کار برنامه کاربردی، یک نقص نرم‌افزاری تولید یا عملیاتی در لایه کسب و کار برنامه کاربردی است که در فازهای مختلف تولید نرم‌افزار (طراحی، پیاده‌سازی، استقرار، پیکربندی و غیره) به وجود آمده است.

برای تعریف حمله لایه کسب و کار، از مفاهیم آسیب‌پذیری لایه کسب و کار (تعریف شده در بالا) و حمله امنیتی استفاده می‌شود.

تعریف ۴ (حمله امنیتی) [۳۰]. هر اقدامی که با سوء استفاده از آسیب‌پذیری، امنیت اطلاعات سازمان را به مخاطره می‌اندازد، حمله امنیتی است.

بر اساس تعریف ۳ و ۴، تعریف ۵ ارائه می‌شود.

تعریف ۵ (حمله لایه کسب و کار). حمله به لایه کسب و کار برنامه کاربردی، اقدامی است که با هدف قراردادن یک آسیب پذیری لایه کسب و کار برنامه کاربردی، امنیت آن را به مخاطره انداخته است.

تعریف ۶ (حمله منع خدمت سیلابی) [۳۱]. در حمله منع خدمت سیلابی، مهاجمین ارتباط کاربران قانونی را با اشغال پهنای باند شبکه مختل می‌کنند (مانند حمله سیلابی UDP، ICMP، DNS و VOIP).

بر اساس تعریف ۳ و ۵ و ۶، تعریف ۷ ارائه می‌شود. در واقع برای تعریف حمله منع خدمت در لایه کسب و کار، از مفاهیم آسیب پذیری/حمله لایه کسب و کار (تعریف ارائه شده در بالا) و حمله منع خدمت سیلابی استفاده می‌شود.

تعریف ۷ (حمله منع خدمت سیلابی در لایه کسب و کار). حمله منع خدمت سیلابی در لایه کسب و کار برنامه کاربردی وب، با سوء استفاده از یک یا چند آسیب پذیری لایه کسب و کار، به‌صورت هوشمندانه، دنباله‌ای از درخواست‌های آگاه از فرایندهای کسب و کار برنامه را به سمت برنامه کاربردی ارسال کرده تا منابع مختلف سیستم قربانی مانند پهنای‌باند، پردازنده و حافظه را سریع اشغال نموده و قربانی را از عملکرد صحیح خود بازداشته و دسترس‌پذیری سیستم وی را به مخاطره بیاندازد.

همان‌طور که مشاهده شد در این تحقیق تعریف جامع‌تری برای آسیب‌پذیری لایه کسب و کار بیان شد که نشان می‌دهد وجود هر گونه ضعف در لایه کسب و کار برنامه، امکان وجود آسیب‌پذیری را فراهم می‌آورد و آسیب‌پذیری لایه کسب و کار را در سطح لایه کسب و کار تعریف نمودیم. تعریف ارائه شده در این تحقیق، تعاریف موجود برای آسیب‌پذیری‌های منطقی را پوشش می‌دهد و جامعیت را حفظ نموده است.

۳-۲- روش پیشنهادی

برای ارزیابی تاب‌آوری برنامه کاربردی وب در برابر حملات منع‌خدمت لایه کسب و کار، BLDAST[۶] را پیشنهاد می‌دهیم. BLDAST از روش آزمون امنیتی پویای لایه کسب و کار برنامه کاربردی وب، برای شناسایی آسیب‌پذیری برنامه، در برابر حملات منع‌خدمت سیلابی استفاده می نماید.

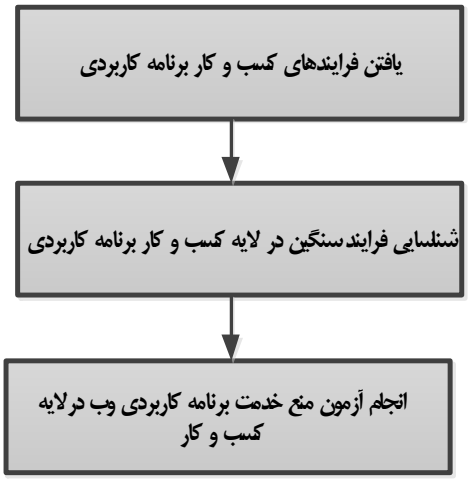
BLDAST با پویش برنامه، فرایندهای کسب و کار برنامه را استخراج می‌کند، سپس با در نظر گرفتن معیارهایی، فرایندهای کسب وکار را شناسایی می‌کند که در مقابل حملات منع خدمت آسیب‌پذیر هستند و مهاجم با صرف توان کمتری می‌تواند از طریق آن‌ها برنامه کاربردی را از دسترس خارج کند. BLDAST با شناسایی فرایندهای بحرانی که مستعد بروز حملات منع‌خدمت هستند و با آگاه نمودن توسعه‌دهندگان، مبنی بر وجود چنین گلوگاه‌هایی و برطرف نمودن آن‌ها، موجب افزایش تاب آوری برنامه، در مقابل حملات منع‌خدمت می‌شود. به عبارت دیگر BLDAST فرایندهای کسب و کار بحرانی را که امکان تبدیل شدن به گلوگاه را دارند، می‌یابد و به کاربر جهت رفع اشکال آن‌ها هشدار می دهد.

ترافیک تولیدی توسط سناریو آزمون محدود است و همین ترافیک تولیدی قادر به از دسترس خارج کردن برنامه کاربردی است. به دلیل تولید ترافیک کم توسط سناریو آزمون، امکان کشف سناریو آزمون توسط ابزارهای شناسایی کاهش می‌یابد. در نتیجه مقابله با چنین آزمون‌هایی دشوار خواهد بود.

BLDAST دارای سه مرحله اصلی است. این سه مرحله عبارتند از:

۱. یافتن فرایندهای کسب و کار برنامه
۲. شناسایی فرایند سنگین در لایه کسب و کار برنامه کاربردی
۳. انجام آزمون منع خدمت برنامه کاربردی وب در لایه کسب و کار

شکل ۲ مراحل پیشنهادی برای آزمون امنیتی پویای لایه کسب و کار برنامه کاربردی وب، را نشان می‌دهد. در ادامه هر کدام از این مراحل را توضیح می‌دهیم.



شکل ۲: روش آزمون امنیتی پویای لایه کسب و کار برنامه کاربردی وب به منظور شناسایی آسیب‌پذیری برنامه در برابر حملات منع خدمت سیلابی

۳-۳- یافتن فرایندهای کسب و کار برنامه کاربردی

در این مرحله تمامی صفحات موجود در برنامه کاربردی را خزش می‌کنیم. البته خزش به صورت هوشمندانه صورت می‌گیرد بدین صورت که صفحات وابسته به هم و فرایندهای موجود در برنامه شناسایی می‌شوند. برای یافتن فرایندهای کسب و کار برنامه کاربردی، ابتدا برنامه را به صورت ماشین حالت [۱۶] مدل می‌کنیم. پس از تولید ماشین حالت برنامه کاربردی، فرایندهای کسب و کار برنامه را با استفاده از ماشین حالت استخراج می‌کنیم.

برای استخراج فرایندهای کسب وکار برنامه از ماشین حالت تولید شده لازم است حالت‌های نهایی را در ماشین حالت مشخص نماییم. منظور از حالت نهایی، حالتی است که اگر برنامه کاربردی در آن حالت قرار گیرد، به معنی اتمام یک فرایند کسب و کار است. با بررسی پاسخ‌های پیام می‌توان حالت‌های نهایی را مشخص کرد. به عنوان مثال در فرایند خرید یک کالا، پس از اتمام خرید عبارتی مانند "از خرید شما متشکریم" نمایش داده می‌شود. با مشخص کردن مجموعه‌ای از این عبارت‌ها و جست‌وجوی این عبارت‌ها در پاسخ پیام‌های دریافتی، می‌توان حالت‌های نهایی را مشخص نمود. تمامی مسیرهای موجود در ماشین حالت از حالت اولیه (صفحه آغازین برنامه کاربردی) تا حالت‌های نهایی تعیین شده، فرایندهای کسب و کار برنامه هستند.

۳-۴- انتخاب فرایند دارای سربار بیشتر در لایه کسب و کار برنامه کاربردی

BLDAST بعد از شناسایی صفحات و فرایندهایی که منطق کسب و کار برنامه را اجرا می‌کنند، در این مرحله صفحات و فرایندهایی را شناسایی می‌کند که در برابر حملات منع خدمت سیلابی آسیب‌پذیر هستند. به عبارت دیگر صفحات و فرایندهایی شناسایی می‌شوند که بار بیشتری بر روی کارگزار وب اعمال می‌کنند. مهاجم با استفاده از این

صفحات و فرایندها با مصرف توان کمتری می‌تواند برنامه کاربردی را از دسترس خارج نماید. BLDAST این صفحات و فرایندهای کسب و کار را شناسایی می‌کند و با هشدار به توسعه‌دهنده برنامه، مبنی بر برطرف نمودن چنین ضعفی در برنامه کاربردی، بر تاب‌آوری برنامه در برابر حملات منع‌خدمت سیلابی می‌افزاید.

BLDAST از معیارهایی برای شناسایی صفحات و فرایندهای سنگین استفاده می‌کند. **Error! Reference source not found.** معیارهای انتخاب فرایند سنگین را نشان می‌دهد. با توجه به معیارهای بیان شده در جدول ۱ BLDAST صفحات و فرایندهای سنگین را انتخاب می‌کند.

۳-۵- طراحی سناریو آزمون منع خدمت برنامه کاربردی وب در لایه کسب و کار

هدف BLDAST شناسایی آسیب‌پذیری منع خدمت در لایه کسب و کار است. BLDAST به منظور ارزیابی هر چه بهتر تاب‌آوری برنامه در مقابل حملات منع خدمت لایه کسب وکار، سعی می‌نماید تا جایی که امکان دارد رفتار مهاجم در هنگام اعمال حمله منع خدمت را شبیه‌سازی نماید بنابراین در این مرحله، سناریو آزمون منع خدمت سیلابی طراحی می‌گردد و صفحات و فرایندهای منتخب در لایه کسب و کار را مورد هدف قرار می‌دهد. سناریو آزمون تولیدی در این مرحله را به اختصار BLDoS می‌نامیم.

جدول ۱: معیارهای انتخاب فرایند دارای سربار بیشتر

معیار	توضیح
زمان پاسخ	هرچه زمان پاسخ صفحه بیشتر باشد، صفحه موردنظر، سربار بیشتری بر روی کارگزار وب اعمال می‌کند.
نرخ بازدید از صفحه	هر چه صفحه محبوبیت بیشتری داشته باشد، پربازدید تر خواهد بود. صفحه پربازدید، نسبت به صفحه‌ای که نرخ بازدید آن کمتر است، بار بیشتری بر روی کارگزار اعمال می‌کند.
نرخ به‌روزرسانی صفحه	تعداد دفعات به‌روزرسانی، تعداد درخواست‌های فرستاده از صفحه خاص را افزایش می‌دهد. هر چه تعداد دفعات به‌روزرسانی یک صفحه در واحد زمانی خاص، بیشتر باشد، صفحه موردنظر بار بیشتری بر روی کارگزار وب اعمال می‌کند.
وزن صفحه	هر صفحه با توجه به درخواست‌های متعددی که برای بارگذاری محتویات، عکس‌ها و دیگر فایل‌های آن صفحه وجود دارد، وزن مخصوص به خود را دارد. هر چه وزن صفحه بیشتر باشد، بار بیشتری بر روی کارگزار اعمال می‌کند.

BLDAST پیشنهادی برای طراحی سناریو آزمون BLDoS، مجموعه درخواست‌هایی تولید می‌نماید که فرایند کسب و کار منتخب را تقاضا کند. هدف سناریو آزمون BLDoS از دسترس خارج نمودن برنامه کاربردی است به‌طوری که قادر به پاسخگویی به کاربران قانونی نیست.

چالش‌هایی که برای طراحی سناریو آزمون BLDoS وجود دارد عبارتند از:

۱. ذخیره و ارسال کلوچک[۷] مانند مرورگرها به‌منظور نگه‌داری نشست
۲. تعداد تقاضا برای اجرای فرایند مورد نظر
۳. شبیه‌سازی مقادیر پویای نشان CSRF[۸] در درخواست‌های HTTP

برای حل چالش کلوچک‌ها می‌بایست پیمانه‌ای برای مدیریت کلوچک طراحی شود. در صورتی که پاسخ HTTP شامل کلوچک باشد، می‌بایست کلوچک موردنظر ذخیره گردد و در تمامی درخواست‌های HTTP بعدی مورد استفاده قرار گیرد. استفاده از کلوچک در اجرای فرایندهای کسب و کار ضروری است زیرا برای انجام منطق کسب و کار موردنظر، نیاز به نگه‌داری نشست وجود دارد. وظیفه پیمانه مدیریت کلوچک، اجرای مقتضی مجموعه درخواست‌ها به منظور پیاده‌سازی منطق کسب و کاری مورد نظر است.

تعداد تقاضا برای اجرای فرایند موردنظر باید به‌گونه‌ای باشد که اولاً ترافیک بالا تولید نکند زیرا در این صورت احتمال شناسایی سناریو آزمون افزایش می‌یابد و ثانیاً به اندازه‌ای هم کم نباشد که بار قابل توجهی به کارگزار اعمال نکند. BLDAST برای پیدا کردن تعداد تقاضا برای اجرای منطق کسب و کار موردنظر، کمترین تعداد تقاضا برای اجرای منطق کسب و کار را انتخاب می‌نماید. به طوری که برنامه کاربردی قادر به پاسخگویی به کاربران قانونی نخواهد بود.

BLDAST پیشنهادی، برای حل چالش نشان CSRF، از عبارات با قاعده، استفاده می‌نماید بدین صورت که پیمانه‌ای برای تولید عبارات باقاعده طراحی می‌نماید. مقادیر تولیدی منحصر به فرد هستند. در درخواست‌های HTTP اغلب از نوع POST، پارامترهای درخواستی که دارای مقادیر پویا هستند، از پیمانه طراحی شده استفاده می‌نمایند.

بنابراین با توجه به تمهیدات بیان شده، ترافیک تولیدی سناریو آزمون BLDoS، قادر به اجرای فرایند کسب و کار منتخب است. احتمال کشف چنین سناریو آزمونی پایین است زیرا ترافیک تولیدی شبیه ترافیک کاربر قانونی است و با تعداد درخواست کمتری، سناریو آزمون اعمال می‌شود.

۳-۶- ارتقا سناریو آزمون منع خدمت طراحی شده به منظور مورد هدف قرار دادن فرایند کسب و کار منتخب

با توجه به اینکه حمله منع خدمت آهسته، یک دسته از حملات معروف و تأثیر گذار است، می‌توان به سناریو آزمون منع خدمت پیشنهادی BLDoS، ویژگی آهستگی نیز اضافه کرد تا تأثیر سناریو آزمون دوچندان گردد و برنامه کاربردی (در این تحقیق هدف مورد آزمون) به صورت آهسته و با حجم ترافیکی محدودتری مورد آزمون قرار گیرد. در این

صورت با توجه به محدود شدن ترافیک آزمون، احتمال شناسایی چنین آزمونی توسط ابزارهای شناسایی کاهش می‌یابد. علاوه بر این توان کمتری برای تولید چنین سناریو آزمونی مصرف می‌گردد.

در سناریو منع خدمت آهسته که slow BLDoS می‌نامیم، BLDAST درخواست‌های HTTP را به صورت آهسته و با تأخیر به سمت برنامه کاربردی ارسال می‌کند. تا نشست ایجاد شده نگه داشته شود و طولانی گردد و منابع تخصیص داده شده به نشست، مادامی که نشست برقرار است، نگه داشته می‌شوند.

در Slow BLDoS پیشنهادی، به دلیل اینکه فرایند کسب و کار انتخاب شده مورد حمله قرار می‌گیرد و از آنجایی که فرایند کسب وکار مجموعه‌ای از درخواست‌های HTTP POST و یا GET است، بر حسب نوع درخواست HTTP موردنظر،Slow POST و یا Slow GET انجام می‌گیرد بدین ترتیب که در Slow GET، سرآیندهای درخواست HTTP با تأخیر فرستاده می‌شوند و در Slow POST بدنه پیام HTTP به صورت آهسته و با تأخیر فرستاده می‌شوند.

به عنوان مثال فرض کنید که فرایند خرید در فروشگاه خرید الکترونیک، به عنوان فرایند کسب و کار با سربار بیشتر در نظر گرفته شده است، اضافه کردن ویژگی آهستگی در این فرایند، بدین معنی است که BLDAST، هر مرحله از فرایند خرید را به قدری طولانی می‌کند تا نشست ایجاد شده برای خرید کالا، همچنان منابع کارگزار را اشغال نگه دارد.

در حین اعمال سناریو آزمون به سمت برنامه‌کاربردی، درخواست هایی از سمت کاربر قانونی فرستاده می‌شود درصورتی که برنامه کاربردی قادر به پاسخگویی نباشد نسبت به حمله منع خدمت در لایه کسب و کار آسیب‌پذیر است.

پس از اعمال سناریو آزمون منع خدمت به برنامه کاربردی، صفحات و فرایندهای کسب و کار بحرانی شناسایی می‌شوند در واقع این صفحات و فرایندهای کسب و کار بحرانی، می‌توانند به عنوان گلوگاه برای برنامه کاربردی محسوب شوند و برنامه کاربردی را در مقابل حملات منع خدمت در لایه کسب و کار آسیب‌پذیر نمایند. با شناسایی این صفحات و فرایندهای کسب و کاری بحرانی، توسعه‌دهندگان را از وجود چنین فرایندهای بحرانی مطلع نموده تا بر تاب‌آوری برنامه کاربردی افزوده شود.

۴- پیاده‌سازی و ارزیابی

برای پیاده‌سازی روش پیشنهادی، بستر آزمایش، شبکه‌ای دارای یک کارگزار وب (هدف مورد آزمون) و دو مشتری[۹] (سیستم BLDAST و کاربر قانونی) است. کارگزار وب و مشتری‌ها بر روی ماشین مجازی بارگذاری شده‌اند. ماشین کارگزار وب (هدف مورد آزمون) دارای سیستم عامل windows 8.1، CPU: Pentium dual core-2.20 GHZ و دارای ماشین مجازی با سیستم عامل windows 7 و RAM: 1G است. دو مشتری (سیستم BLDAST و کاربر قانونی) دارای سیستم عامل windows 8.1، CPU: Pentium dual core i3-3210 GHZ و دارای ماشین مجازی با سیستم عامل windows 7 و RAM: 1G است.

بر روی کارگزار وب (هدف مورد آزمون)، برنامه‌کاربردی‌های بیان شده در جدول ۲ نصب شده‌اند و قصد آزمون امنیتی برنامه‌های فوق را داریم.

جدول ۲: برنامه‌های کاربردی منتخب جهت ارزیابی

برنامه کاربردی	توضیح
TomatoCart-1.1.8.6.1	برنامه کاربردی فروشگاه الکترونیک [۳]
Drupal-7.38	برنامه کاربردی سیستم مدیریت محتوا[۱۰]
oscommerce-2.3.4	برنامه کاربردی فروشگاه الکترونیک [۳ و ۱۱]
Wordpress-4.7.2	برنامه کاربردی سیستم مدیریت محتوا

ارزیاب برای انجام آزمون امنیتی و تولید سناریوهای آزمون منع خدمت، از ابزار JMeter[11] نسخه ۲,۱۳ استفاده می‌کند. JMeter ابزاری منبع باز برای آزمون قوانین کسب و کار و کارایی نرم‌افزار است و به منظور شبیه‌سازی بار سنگین بر روی کارگزار وب مورد استفاده قرار می‌گیرد.

برای انتخاب فرایندهای دارای سربار بیشتر، معیارهای نرخ بازدید از صفحه و نرخ به‌روز رسانی صفحه محاسبه نشده است زیرا این دو معیار برای برنامه‌هایی قابلیت اجرا دارد که امکان دسترسی عموم را دارند. دو معیار وزن صفحه و زمان پاسخ برای انتخاب صفحات و فرایندهای دارای سربار بیشتر استفاده شده است.

۴-۱- ارزیابی

در این بخش، سناریو آزمون منع خدمت BLDoS و Slow BLDoS با دیگر سناریوهای آزمون منع خدمت در لایه کاربرد [۱۷] از لحاظ حجم ترافیک تبادل شده در واحد زمان، تعداد درخواست، میزان مصرف پردازنده در سیستم ارزیاب، گذردهی[12] کارگزار در حین اعمال سناریو و تعداد ارتباطات TCP مقایسه شده‌است. ابزار OWASP HTTP POST Tool[13] قابلیت تولید سناریو آزمون منع خدمت در لایه کاربرد را دارد. نتایج سناریو آزمون پیشنهادی نیز با این ابزار مقایسه می‌شوند. تمامی آزمایش‌ها سه بار تکرار شده‌اند و نتیجه نهایی، میانگین سه بار تکرار است. در جدول ۳ سناریوهای آزمون منع خدمت سیلابی در لایه‌های کاربرد و لایه کسب و کار مورد استفاده در ارزیابی مشخص شده‌اند.

به منظور مقایسه سناریوهای آزمون معرفی شده در جدول ۳، سناریوهای آزمون بر روی چند برنامه کاربردی پرکاربرد، اعمال شده‌اند. با توجه به اینکه برنامه‌های کاربردی منتخب برنامه‌های منبع‌بازی هستند که داده کمی در داخل پایگاه داده‌شان وجود دارد، برای شبیه سازی محیط واقعی لازم است داده‌های کافی در پایگاه داده برنامه‌های پیشنهادی وجود داشته باشد. به منظور تولید داده در برنامه‌ها، با توجه به متفاوت بودن فرایند تولید داده در هر برنامه، به صورت مستقل برای هر برنامه سناریویی در **JMeter** ایجاد شد و داده‌های کافی در داخل پایگاه داده هر برنامه کاربردی گنجانده شد.

جدول ۳: فهرست سناریوهای آزمون منع خدمت سیلابی در لایه‌های کاربرد و کسب و کار مورد استفاده در ارزیابی

نوع سناریو آزمون	توضیح
Repeated Request DoS [۱۷]	سناریو آزمون سیلابی به صفحه آغازین و یا صفحه محبوب برنامه کاربردی
Recursive Request DoS [۱۷]	سناریو آزمون سیلابی به صفحات مختلف برنامه کاربردی
Repeated Workload DoS [۱۷]	سناریو آزمون سیلابی برای جست‌وجو در پایگاه داده و یا درخواست فایل‌های تصویری حجیم
ابزار OWASP HTTP POST	سناریو آزمون منع خدمت سیلابی به صورت آهسته
سناریو آزمون پیشنهادی BLDoS	سناریو آزمون منع خدمت سیلابی در لایه کسب و کار برنامه کاربردی
سناریو آزمون پیشنهادی slow BLDoS	سناریو آزمون منع خدمت سیلابی در لایه کسب و کار برنامه کاربردی به صورت آهسته

معیارهای ارزیابی و سنجش سناریوهای آزمون BLDoS و Slow BLDoS پیشنهادی عبارتند از:

۱. تعداد درخواست‌ها

معیار تعداد درخواست‌ها، مشخص کننده این است که در سناریو آزمون منع خدمت سیلابی چند درخواست به صورت موازی به سمت برنامه کاربردی (هدف مورد آزمون) باید ارسال گردد تا هدف از دسترس خارج گردد.

۲. درصد میزان مصرف پردازنده در سیستم مجری BLDAST

درصد مصرف پردازنده در حین اعمال سناریوها در سیستم BLDAST یکی دیگر از معیارهای محاسبه شده برای ارزیابی است.

۳. تعداد ارتباطات TCP ایجاد شده در حین اعمال سناریو آزمون

معیار ارزیابی دیگر، تعداد ارتباط TCP ایجاد شده در حین اعمال سناریوهاست.

۴. گذردهی کارگزار وب (هدف مورد آزمون) در واحد زمان در حین اعمال سناریو آزمون

گذردهی کارگزار، تعداد درخوا ست‌هایی ا ست که در واحد زمان (ثانیه) در حین اعمال ســناریو به ســمت کارگزار وب فرستاده می‌شود.

۵. ترافیک تبادل شده برحسب KB/sec در حین اعمال سناریو

معیار ترافیک تبادل شــده، بیان کننده مقدار ترافیک مبادله شده در حین اعمال سناریوها برحسب KB/sec است.

شکل ۳ تعداد درخواست سناریوها بر روی برنامه‌های کاربردی منتخب را نشان می‌دهد همان‌طور که مشاهده می‌شود تعداد درخواست برای سناریو آزمون پیشنهادی BLDoS و Slow BLDoS برای تمامی برنامه‌های کاربردی منتخب از باقی سناریوها کمتر است. این بدین معنی است که مهاجم با توان کمتری قادر به اجرای حملات لایه کسب و کار است و این موضوع خطرناک بودن حملات لایه کسب و کار و لزوم وجود راهکار آزمون، جهت شناسایی آسیب‌پذیری‌های لایه کسب و کار را برجسته‌تر می‌کند.

شکل ۴ درصد میزان مصرف پردازنده را در سیستم ارزیاب به هنگام اعمال سناریوها نشان می‌دهد. مشاهده می‌شود مقدار مصرف پردازنده به هنگام اعمال سناریو آزمون BLDoS و Slow BLDoS از باقی سناریوها پایین‌تر است. در نتیجه، سناریوهای آزمون پیشنهادی با مصرف منابع کمتری قابل اجرا هستند. بنابراین مهاجم راحت‌تر می‌تواند حملات لایه کسب و کار را اجرا نماید.

شکل **۵** تعداد ارتباطات TCP ایجاد شده در حین اعمال سناریوها را نشان می‌دهد. مشاهده می‌شود در هنگام اعمال سناریو BLDoS و Slow BLDoS تعداد ارتباطات TCP ایجاد شده، نسبت به باقی سناریوها کمتر است. در نتیجه ترافیک تولیدی کمتر است و شناسایی این نوع سناریوها دشوارتر است.

شکل ۶ مقدار گذردهی کارگزار وب (هدف مورد آزمون) را در حین اعمال سناریوها نشان می‌دهد. مشاهده می‌شود در هنگام اعمال سناریو BLDoS و Slow BLDoS مقدار گذردهی هدف مورد آزمون پایین‌تر از سناریوهای دیگر است که حاکی از این واقعیت است که در حین اعمال سناریوهای BLDoS و Slow BLDoS تعداد درخواست‌های کمتری در واحد زمان به سمت کارگزار ارسال می‌شود.

جدول ۴ ترافیک تبادل شده بر حسب KB/sec در حین اعمال سناریوها را نشان می‌دهد. مشاهده می‌شود در سناریو آزمون پیشنهادی BLDoS و Slow BLDoS، ترافیک تبادل شده نسبت به باقی سناریوها کمتر است.

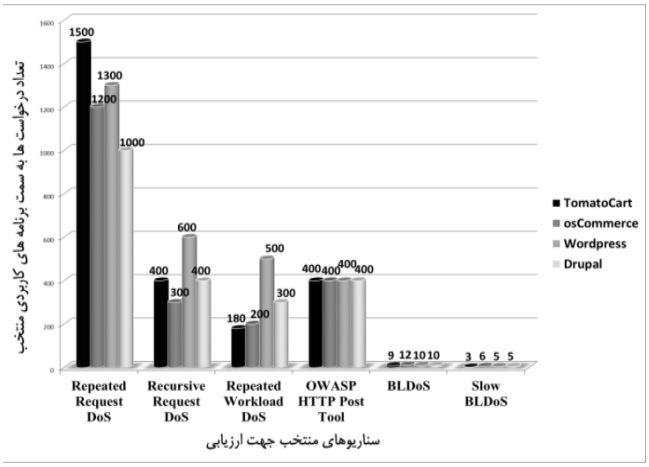
جدول ۵ میانگین معیارهای محاسبه شده برای هر کدام از سناریوها اعمال شده را نشان می‌دهد. با توجه به جدول سناریو آزمون BLDoS و Slow BLDoS میانگین نتایج بهتری نسبت به سناریوهای دیگر دارد.

جدول ۶ توان مهاجم را برای اجرای حملات مربوطه نسبت به سناریو آزمون BLDoS نشان می‌دهد. به عنوان مثال بر طبق معیار "ترافیک تبادل شده" مهاجم برای اجرای سناریو آزمون BLDoS، ۹۵/۵ برابر توان کمتری نسبت به سناریو Repeated Request DoS مصرف می‌کند.

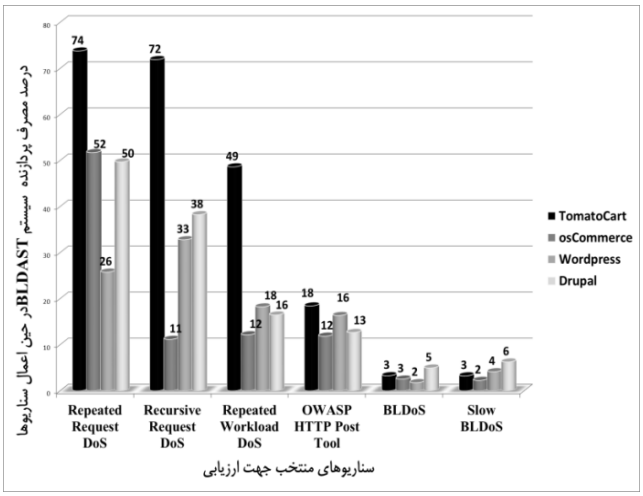
جدول ۷ توان مهاجم را برای اجرای حملات مربوطه نسبت به سناریو آزمون Slow BLDoS **نشان می‌دهد. به عنوان مثال بر طبق معیار "ترافیک تبادل شده" مهاجم برای اجرای سناریو آزمون** Slow BLDoS**، ۱۶۳ برابر توان کمتری نسبت به سناریو** Repeated Request DoS **مصرف می‌کند.**

مقایسه معیارها نشان می‌دهد که مهاجم برای اجرای حملات لایه کسب و کار توان کمتری در حدود یک صدم توان حملات دیگر نیاز دارد. با توجه به اینکه ترافیک تولیدی حملات لایه کسب و کار کمتر از یک صدم ترافیک تولیدی حملات دیگر است

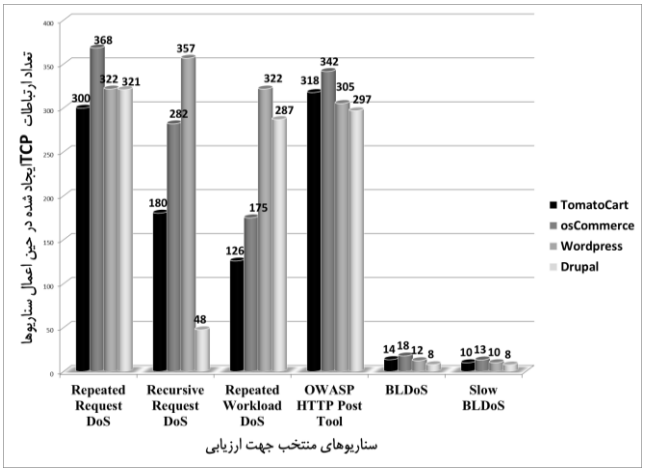
شناسایی این نوع حملات دشوار است. بنابراین حملات لایه کسب و کار، حملات خطرناکی هستند و نیاز به روش‌هایی برای شناسایی آسیب‌پذیری‌های لایه کسب وکار، از اهمیت ویژه‌ای برخوردار است.



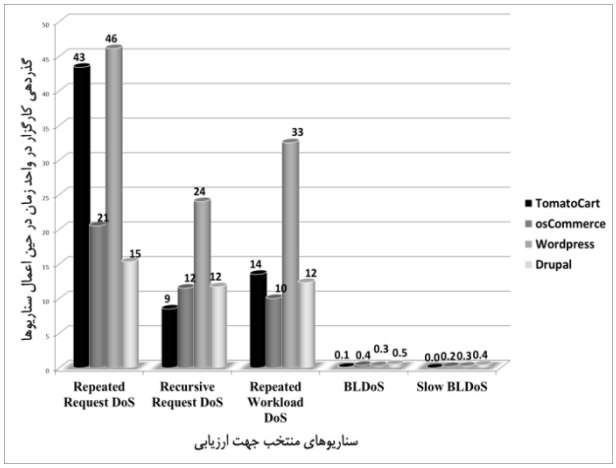
شکل ۳: تعداد درخواست‌ها به سمت برنامه‌های کاربردی



شکل ۴: درصد میزان مصرف پردازنده در سیستم BLDAST



شکل ۵: تعداد ارتباطات TCP ایجاد شده در حین اعمال سناریوها



شکل ۶: مقدار گذردهی کارگزار وب در واحد زمان در حین اعمال سناریوها

جدول ۴: ترافیک تبادل شده بر حسب KB/sec در حین اعمال حملات

برنامه کاربردی / سناریو آزمون	TomatoCart	osCommerce	Wordpress	Drupal
Repeated Request DoS	۴۳۵۳۵	۴۳۰۶۸	۱۳۶۸۸۲	۷۶۰۲
Recursive Request DoS	۴۲۸۲۹	۲۵۰۲۸۵	۱۵۱۸۱۴	۲۰۳۵۸
Repeated Workload DoS	۵۸۵۶۴	۱۳۳۹۶۵۵	۱۳۶۸۸۲	۲۱۱۳
ابزار تولید سناریو آزمون منع خدمتOWASP	۳۴۷۴	۳۴۰۵	۵۰۳۵	۴۲۸۲
BLDoS	۸۶۹	۷۳۴	۷۵۸	۵۶
Slow BLDoS	۷۲۲	۲۹۲	۳۷۲	۳۵

جدول ۵: میانگین معیارها برای هر کدام از سناریوها

برنامه کاربردی / سناریو آزمون	تعداد درخواست حمله	ترافیک تبادل شده	درصد مصرف پردازنده ارزیاب	گذردهی قربانی	تعداد ارتباطات TCP
Repeated Request DoS	۱۲۵۰	۵۷۷۷۲	۵۰/۲	۳۱/۳	۳۲۷/۶
Recursive Request DoS	۴۲۵	۱۱۶۳۲۱	۳۸/۵	۱۳/۹	۲۱۶/۶
Repeated Workload DoS	۲۹۵	۸۲۸۸۱	۲۳/۸	۱۷	۲۲۷/۳
ابزار تولید سناریو آزمون منع خدمت OWASP	۴۰۰	۴۰۴۹	۱۴/۸	NA	۳۱۵/۲
BLDoS	۱۰/۲	۶۰۴	۳/۱	۰/۳	۱۲/۸
Slow BLDoS	۴/۷	۳۵۵	۳/۹	۰/۲	۱۰/۲

جدول ۶: توان مهاجم نسبت به حمله BLDoS

برنامه کاربردی / سناریو آزمون	تعداد درخواست حمله	ترافیک تبادل شده	درصد مصرف پردازنده ارزیاب	گذردهی قربانی	تعداد ارتباطات TCP
Repeated Request DoS	۱۲۱/۹	۹۵/۵	۱۶	۹۹/۹	۲۵/۴
Recursive Request DoS	۴۱/۴	۱۹۲/۳	۱۲/۳	۴۴/۴	۱۶/۸
Repeated Workload DoS	۲۸/۷	۱۳۷	۷/۶	۵۴/۵	۱۷/۶
ابزار تولید سناریو آزمون منع خدمت OWASP	۳۹	۶/۶	۴/۷	NA	۲۴/۴

جدول ۷: توان مهاجم نسبت به حمله Slow BLDoS

برنامه کاربردی / سناریو آزمون	تعداد درخواست حمله	ترافیک تبادل شده	مصرف پردازنده ارزیاب	گذردهی قربانی	تعداد ارتباطات TCP
Repeated Request DoS	۱/۲	۱۶۳	۶/۱	۴/۱	۳۲
Recursive Request DoS	۴/۸	۳۲۶	۷/۹	۶/۶	۲/۲
Repeated Workload DoS	۱/۶	۲۳۲	۰/۶	۲/۷	۲/۲
ابزار تولید سناریو آزمون منع خدمت OWASP	۲/۸	۱۱	۷/۳	NA	۸/۳

۵- نتیجه‌گیری

امروزه حملات وب تهدیدات رایج و جدی هستند. انتقال حملات به لایه‌های بالای OSI و همین‌طور لایه کسب و کار منجر به مخرب‌تر و مؤثرتر شدن حملات شده‌است و شناسایی حملات را دشوارتر کرده‌است. آسیب‌پذیری‌های لایه کسب و کار، آسیب‌پذیری‌های قدرتمندی هستند که امنیت برنامه کاربردی را به مخاطره می‌اندازند. پویشگرهای خودکار نمی‌توانند آسیب‌پذیری‌های لایه کسب و کار را شناسایی نمایند زیرا قادر به درک منطق و کسب و کار برنامه نیستند. برای شناسایی آسیب‌پذیری لایه کسب و کار لازم است منطق و کسب و کار برنامه درک گردد بنابراین این‌گونه آسیب‌پذیری مختص برنامه کاربردی بوده و شناسایی آن دشوار است.

در این تحقیق برای ارزیابی تاب‌آوری برنامه کاربردی در مقابل حملات منع‌خدمت سیلابی، روشی جعبه سیاه به نام BLDAST را پیشنهاد دادیم. هدف BLDAST شناسایی آسیب‌پذیری‌های لایه کسب و کار برنامه در مقابل حملات منع‌خدمت سیلابی است. BLDAST شامل سه مرحله است: ۱- استخراج فرایندهای کسب و کار برنامه در لایه کسب و کار ۲- شناسایی فرایند سنگین در لایه کسب و کار ۳- انجام سناریو آزمون منع‌خدمت سیلابی. در آزمایشگاه به کمک BLDAST، تاب‌آوری چهار برنامه را در برابر حملات منع خدمت لایه کسب و کار آزمودیم. نشان دادیم توان مصرف شده برای اعمال حملات لایه کسب و کار در حدود یک صدم توان اعمال حمله در لایه‌های دیگر است. بنابراین آسیب‌پذیری‌ها در لایه کسب و کار به مراتب خطرناک‌تر از آسیب‌پذیری‌ها در لایه کاربرد هستند زیرا مهاجم به راحتی با صرف توانی خیلی کمتر از توانی که برای حملات دیگر مصرف می‌کند، می‌تواند امنیت برنامه را به مخاطره بیندازد.

مراجع

[۱] میترا علیدوستی و علیرضا نوروزی، «روش آزمون امنیتی پویای لایه کسب و کار برنامه کاربردی وب برای شناسایی آسیب پذیری برنامه کاربردی وب در برابر حملات منع خدمت سیلابی،» کنفرانس بین المللی انجمن رمز ایران، چهارده، ۱-۷، دانشگاه شیراز، ۹۶

[2] ITRC, Identity Theft Resource Center Breach Report Hits Record High in 2015, 2015, http://www.idtheftcenter.org/ITRC-Surveys-Studies/2015databreaches.html.

[3] G. Pellegrino and D. Balzarotti. "Toward black-box detection of logic flaws in web applications," Network and Distributed System Security Symposium, pp.23-26, February 2014.

[4] D. Balzarotti, M. Cova, V. V. Felmetsger and G. Vigna, "Multi-module vulnerability analysis of web-based applications," Computer and communications security, 2007.

[5] A. Doupé, B. Boe, C. Kruegel and G. Vigna, "Fear the ear: discovering and mitigating execution after redirect vulnerabilities," Computer and communications security, pp.251–262, 2011.

[6] L. Cavedon, G. Vigna, V. Felmetsger, L. Cavedon and C. Kruegel, "Toward automated detection of logic vulnerabilities in web applications," USENIX Security Symposium, pp.143–160, 2010.

[7] A. Wang, A. Mohaisen, W. Chang and S. Chen, "Capturing DDoS attack dynamics behind the scenes," Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 205–215, 2015.

[8] D. Holmes, *The F5 DDoS Protection Reference*, 2013, https://f5.com/Portals/1/Cache/Pdfs/2421/mitigating-ddos-attacks-with-f5-technology-.pdf.

[9] E. Chai, *Business Logic Attacks – Bots and BATs*, 2009, https://www.owasp.org/images/6/6a/BNL09_OWASP_Benelux_2009,_Business_Logic_Attacks_-_v2.pptx.

[10] Web Application Security Consortium, *WASC threat classification*, 2010, http://projects.webappsec.org/w/page/13246978/Threat Classification.

[11] X. Li and Y. Xue, "BLOCK: a black-box approach for detection of state violation attacks towards web applications," Annual Computer Security Applications, pp.247–256, 2011.

[12] M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna, "Swaddler: an approach for the anomaly-based detection ofstate violations in web applications,"Recent Advances in Intrusion Detection, pp.63–86, 2007.

[13] X. Li, W. Yan, and Y. Xue, "SENTINEL: securing database from logic flaws in web applications," Data and Application Security and Privacy, pp. 25–36, 2012.

[۱۴] محیا ارومیه و نگین دانش‌پور، «مدلی سه لایه در طراحی سطح منطقی پایگاه داده تحلیلی،» مجله مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۲، صفحات ۳۷۱-۳۸۰، ۱۳۹۶.

[۱۵] سیامک عبداله‌زاده، محمدلی بالافر و لیلی محمدخانلی، «استفاده از خوشه‌بندی و مدل مارکوف جهت پیش‌بینی درخواست آتی کاربر در وب،» مجله مهندسی برق دانشگاه تبریز، جلد ۴۵، شماره ۳، صفحات ۸۹-۹۶، ۱۳۹۴.

[16] A. Doupé, L. Cavedon, C. Kruegel and G. Vigna, "Enemy of the state: a state-aware black-box web vulnerability scanner,"USENIX Security Symposium, vol.15, no.2, pp.173-180, 2013.

[17] S. Ranjan, "DDoS-Resilient scheduling to counter application layer attacks under imperfect detection," Communications Society, 2006.

[18] C. Cadar, V. Ganesh, P. M. Pawlowski, D. L. Dill and D. R. Engler, "EXE: automatically generating inputs of death," Information and System Security, vol.12, no.2, pp. 10-24, 2008.

[19] R. Chang, G. Jiang, F. Ivančić, S. Sankaranarayanan, and V. Shmatikov. "Inputs of coma: Static detection of denial-of-service vulnerabilities," Computer Security Foundations Symposium, pp.186–199,2009.

[20] A. Gupta, T. A. Henzinger, R. Majumdar, A. Rybalchenko, and R.-G. Xu, "Proving non-termination,"ACM Sigplan Notices, vol.43, no.1, pp.147-158, 2008.

[21] J. Burnim, N. Jalbert, C. Stergiou, and K. Sen, "Looper: lightweight detection of infinite loops at runtime," Automated Software Engineering, pp.161-169, 2009.

[22] J. Burnim, S. Juvekar, and K. Sen, "WISE: automated test generation for worst-case complexity Software Engineering, pp.463–473, 2009.

[23] O. Olivo, I. Dillig, and C. Lin, "Detecting and exploiting second order denial-of-service vulnerabilities in web applications," Computer and Communications Security, pp. 616–628, 2015.

[24] S. Son and V. Shmatikov, "SAFERPHP: finding semantic vulnerabilities in PHP applications," Programming Languages and Analysis for Security, 2011,

[25] X. Li, and Y. Xue, "A survey on server-side approaches to securing web applications," Computing Surveys, vol 46, no.4, 2014.

[26] G. Stergiopoulos, B. Tsoumas, and D. Gritzalis, "On business logic vulnerabilities hunting: the APP_LogGIC framework," Network and System Security, pp. 236–249, 2013.

[27] OWASP, *Business Logic Security Cheat Sheet*, https://www.owasp.org/index.php/Business_Logic_Security_Cheat_Sheet.

[28] P. K. Ray, *Integrated Management from E-business Perspective: Concepts, Architectures and Methodologies*, Springer Science & Business Media, 2012.

[29] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," dependable and secure computing, vol. 1, no.1, pp.11-33, 2004.

[30] William Stallings, *Computer Data and Computer Communications Eighth Edition*, Prentice Hall, 2011.

[31] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," Communications Surveys and Tutorials, vol.15, no.4, pp.2046-2069, 2013.

زیرنویس‌ها

[1] Common Vulnerabilities and Exposures

[2] Open Systems Interconnection

[3] resiliency

[4] Business layer Dynamic Application Security Testing For DoS

[5] Stack space

[6] Business layer Dynamic Application Security Testing For DoS

[7] cookie

[8] Cross-Site Request Forgery token

[9] client

[10] Content Management Framework

[11] http://jmeter.apache.org/

[12] throughput

[13] https://www.owasp.org/index.php/OWASP_HTTP_POST_Tool