دوازدهمین کنگره ملی سراسری
فناوریهای نوین در حوزه توسعه پایدار ایران
12th National Congress of
the New Technologies in Sustainable Development of Iran
senaconf.ir

امنیت شبکه‌های مبتنی بر نرم‌افزار (SDN)

محمد حزبه زاده (نویسنده مسئول)[1]، علی غیبی دهناشی(نویسنده دوم) [2]

[1] دانشجوی دانشکده فنی و حرفه‌ای پسران شهید چمران اهواز، اهواز moohmadhazbhzadh@gmail.com

[2]مدرس دانشکده فنی و حرفه‌ای شهید چمران اهواز، اهواز agheibi@tvu.ac.ir

چکیده

شبکه‌های مبتنی بر نرم‌افزار (SDN) یک روش جدید برای بهبود مدیریت و امنیت شبکه‌ها در برابر حملات نفوذگران می باشد. SDN امکان کاهش انواع حملات از جمله (حمله انکار سرویس) DoS را فراهم می‌کند. بسیاری از تکنولوژی‌ها و تکنیک‌های جدید برای حل آسیب‌پذیری‌های امنیتی SDN پیشنهاد شده‌اند و برخی اقدامات نیز می‌تواند برای حل آن‌ها به کار بسته شوند. تحقیقات فعلی در حوزه SDNبیشتر در رابطه با گسترش تکنولوژی‌های SDN می‌باشد . ازآنجاکه OpenFlow معروف‌ترین پروتکل SDN است تحقیقات زیادی برای استفاده و بهبود این پروتکل انجام شده است . تحقیقات آینده احتمالاً این روندها را با بهبود پروتکل OpenFlow و یافتن جایگزین‌های بهتر دنبال می‌کنند که می‌تواند شامل توسعه بیشتر ابزارها برای آزمایش طرح‌های شبکه و تحقیق بهینه‌سازی OpenFlow در هنگام استفاده در محیط‌های مختلف باشد. این مقاله یک بررسی جامع از تحقیقات مربوط به امنیت در شبکه‌های مبتنی بر نرم‌افزار را که تا به امروز انجام شده است، ارائه می‌دهد. همچنین پیشرفت‌های امنیتی که با استفاده از چارچوب SDN حاصل می‌شود و هم چالش‌های امنیتی معرفی شده توسط این چارچوب موردبحث قرار می‌گیرند. با دسته‌بندی کارهای موجود، مجموعه‌ای از نتیجه‌گیری‌ها و پیشنهادها برای جهت‌گیری‌های تحقیقات آتی ارائه می‌شود

واژه‌های کلیدی

نرم‌افزار، شبکه، مدیریت، تکنولوژی

مقدمه

معماری شبکه‌های سنتی برای برآوردن نیاز شرکت‌ها و کاربران امروزی مناسب نیستند. به لطف تلاش گسترده در صنعت کامپیوتر که توسط بنیاد شبکه‌های باز (ONF) رهبری می‌شود، شبکه‌های مبتنی بر نرم‌افزار (SDN) در حال تغییر معماری شبکه هستند. شبکه‌های مبتنی بر نرم‌افزار (SDN) با مجموعه‌ای از دستگاه‌های در حال توسعه ، به سرعت از رویا به واقعیت در حال حرکت است. این شبکه‌ها کاربرد تجاری خود را در فناوری‌های رایانش ابری و مجازی‌سازی پیدا کرده است. مزایای SDN در بخش‌های مختلف (مثلاً سازمانی، مرکز داده و غیره) و در سراسر شبکه‌های اصلی مختلف ثابت شده است [۱]. با این حال، چالش‌هایی برای اجرای شبکه در مقیاس کامل SDN وجود دارد. یکی از چالش‌های کلیدی که دارای اهمیت زیادی می‌باشد، امنیت در SDN است [۲].

معماری SDN را می‌توان برای افزایش امنیت شبکه با ارائه یک سیستم نظارت، تحلیل و پاسخ امنیتی مورداستفاده قرار داد. کنترل‌کننده مرکزی کلید این سیستم است. تجزیه‌وتحلیل ترافیک یا روش‌های تشخیص ناهنجاری که در شبکه وجود دارد، داده‌های مرتبط با امنیت سیستم را تولید می‌کنند که به طور منظم به کنترل‌کننده مرکزی منتقل می‌شوند. برنامه‌ها در کنترلر اجرا می‌شوند تا بازخوردها از شبکه را تجزیه‌وتحلیل کند. بر اساس این تجزیه‌وتحلیل، سیاست امنیتی جدید و به روز شده در سراسر شبکه منتشر می‌شود. این رویکرد تلفیقی به طور مؤثر کنترل و مهار تهدیدات امنیتی شبکه را سرعت می بخشد [۳].

با معرفی راه حل های شبکه‌های مبتنی بر نرم‌افزار (SDN) عملکرد شبکه به طور قابل توجهی تغییر می‌کند: سخت‌افزار عمومی، نرم‌افزار مجازی‌سازی و خدمات قابل برنامه ریزی با SDN ، هزینه عملیات و نگهداری شبکه را کاهش می‌دهد، استفاده از منابع بهبود می‌یابد، انعطاف‌پذیری شبکه افزایش می‌یابد و زمان ارائه به بازار خدمات به طور قابل‌توجهی کاهش می‌یابد [۴]؛ بنابراین، SDN به‌عنوان فناوری نو برای تکامل شبکه‌های مخابراتی در نظر گرفته می‌شوند.

با این حال، SDN نیز چالش‌های امنیتی جدیدی را برای شبکه‌های مخابراتی به همراه دارند. این چالش‌های امنیتی جدید عبارت‌اند از:

•مرز امنیت فیزیکی مبهم، و اینکه شبکه توسط دستگاه‌های امنیتی ایستا و پاسخ‌های امنیتی غیرفعال محافظت می‌شود که این امر منجر به عملکرد و نگهداری امنیتی با کارایی پایین و پاسخ‌های تأخیری به حملات امنیتی می‌شود [۵].

شبکه‌های مبتنی بر نرم‌افزار (SDN) یک فلسفه طراحی شبکه است که مبتنی بر جداسازی صفحه داده شبکه از صفحه کنترل شبکه است. صفحه داده ، نشان‌دهنده تمام داده‌هایی است که از طریق شبکه ارسال می‌شوند، مانند بسته‌ها و سخت‌افزارهایی که برای ارسال آن استفاده می‌شوند، مانند سوئیچ‌ها. صفحه کنترل نشان‌دهنده تمام منطق و دستگاه‌هایی است که مسئول تصمیم‌گیری در مورد چگونگی و محل ارسال داده‌ها در صفحه داده هستند [۶]. شبکه‌های سنتی این دو صفحه را بر روی دستگاه‌های مشابه ترکیب می‌کنند و هر دستگاه وظیفه دارد تا تصمیمات ارسال خود را بر اساس پروتکل‌های مسیریابی توزیع‌شده بگیرد. از سوی دیگر،SDN اجازه می‌دهد تا صفحه کنترل دارای دید کلی از شبکه باشد، و اجازه می‌دهد تا اقداماتی اعمال شوند که همه حالت شبکه را در نظر بگیرند.

این فلسفه طراحی، از ایده شبکه‌های فعال به وجود آمده است [۷] که از توانایی در برنامه‌نویسی در داخل بسته‌ها حمایت می‌کند . شبکه‌های فعال امکان انعطاف‌پذیری زیادی را در پردازش بسته شبکه و نحوه عبور آن‌ها از شبکه فراهم می‌کند .در حال حاضر تحقیقات اندکی مستقیما بر روی شبکه‌های فعال انجام شده است، اما مفهوم شبکه فعال منجر به ایده SDN می‌شود. SDN به پشتیبانی سخت‌افزار تخصصی (مانند شبکه‌های فعال) نیاز دارد [۸] .

معروف‌ترین استاندارد SDN، OpenFlow [۹] ، SDN را با ارائه یک پروتکل ارتباطی اجرا می‌کند که به یک کنترل‌کننده متمرکز اجازه می‌دهد تا با سخت‌افزار تخصصی سطح داده ارتباط برقرار کرده و برنامه را اجرا کند .در اصل، یک کنترل‌کننده OpenFlow قواعد ارسال را برای سوئیچ‌ها می‌نویسد و اقداماتی را که باید روی بسته‌ها انجام شود، مشخص می‌کند و هر بسته‌ای که با قواعد موجود مطابقت ندارد

را به کنترلر ارسال می‌کند. کنترل‌کننده تمام تصمیمات لازم را مشخص می‌کند و سوئیچ‌ها فقط اقداماتی را اعمال می‌کنند که توسط کنترل‌کننده مشخص شده است [۱۰].

بیشتر تحقیقات فعلی در مورد امنیت SDN بر پروتکل OpenFlow متمرکز است. آسیب‌پذیری هایی که در OpenFlow وجود دارد قابل‌تعمیم به هر سیستم SDN است که از یک کنترل‌کننده متمرکز استفاده می‌کند. رایج‌ترین آسیب‌پذیری که توسط تحقیقات اخیر [۱۰،۱۱، ۱۲] ذکر شده است، گلوگاه ارتباطی است که بین صفحه داده و کنترل‌کننده در یک شبکه OpenFlow قرار دارد. با توجه به محل متمرکز یک کنترلر در یک شبکه OpenFlow، بارگذاری بیش از حد مسیر ارتباطی که توسط سوئیچ‌ها برای برقراری ارتباط با کنترلر انجام می‌پذیرد. همان‌طور که توسط Shine et. al. [۱۲]، بیان شده است ایجاد بسته‌هایی که به‌سرعت کانال ارتباطی کنترل را اشباع کند، منجر به‌نوعی حمله انکار سرویس (DoS) در شبکه، احتمالاً با جلوگیری از دسترسی به کنترلر می‌شود .این موضوع همچنین می‌تواند به‌عنوان نقص کنترلر ظاهر شود [۱۳، ۱۴] . به کمک یک کنترل‌کننده توزیع‌شده می‌تواند شبکه را نسبت به یک حمله DoS مقاوم‌تر می‌کند.

کارهای اخیر دیگر در تحقیقات امنیتی SDN به دنبال استفاده از قابلیت برنامه‌ریزی که توسط SDN به‌منظور افزایش امنیت در برابر حملات سنتی فراهم شده است [۱۷، ۱۶، ۱۵] . این راه‌حل‌ها تنها محدود به OpenFlow نیستند، بلکه از اصول SDN بهره می‌برند. این تکنولوژی‌ها امکان پیاده‌سازی استراتژی‌های کاهش مشترک، از جمله استفاده از جعبه‌های میانی، الگوریتم‌های تشخیص و الگوریتم‌های طبقه‌بندی که برای امنیت شبکه‌های سنتی استفاده می‌شوند را فراهم می‌کنند .این راه‌حل‌ها برای حصول اطمینان از اینکه SDNدر برابر شبکه حساس شناخته‌شده، به‌ویژه حملات هدف‌مند در میزبان‌ها آسیب‌پذیر نیستند، مفید هستند .

این مقاله در بخش ۲ با ارائه یک نمای کلی از تحقیقات مربوط به SDN ادامه می‌یابد. در بخش ۳، چندین کار اخیر در زمینه امنیت SDN و اینکه چگونه این کارهای خاص با جنبه‌های مختلف SDNها ارتباط دارند را مرور می‌کنیم. در بخش ۴، بحث در مورد فناوری‌های امنیتی خاص OpenFlow ارائه می‌شود، و در بخش ۵، فناوری‌های که به‌طورکلی برای SDN قابل‌اجرا هستند را شرح می‌دهیم. بخش ۶، تحلیلی از روند تحقیقات امنیتی SDN و پیش بینی در مورد چگونگی پیشرفت تحقیقات SDN در آینده ارائه می‌دهم، و مقاله را در بخش ۷ نتیجه‌گیری می‌کنم .

۱. ارزیابی SDN

شبکه‌های مبتنی بر نرم‌افزار (SDN) از چندین مسیر تحقیقاتی مختلف تکامل یافته‌اند، که با تحقیق در مورد کارهای شبکه فعال شروع شده است. اگرچه برخی از این مسیرها ناموفق بودند، اما انگیزه همه آنها چالش‌های پیش روی مدیریت اینترنت در حال رشد و تمایل به داشتن شبکه های انعطاف پذیرتر و قابل برنامه ریزی بود. [۶].

۱،۱. شبکه‌های فعال

تحقیق در مورد SDN ابتدا با حوزه شبکه‌های فعال آغاز گردید. شبکه فعال توانایی تعبیه محاسبات را در بسته‌ها و دستگاه‌های شبکه فراهم می‌کند که به محاسبات امکان می‌دهد در داخل شبکه به‌عنوان بسته‌ای که در شبکه حرکت می‌کند، انجام شود [۷].

۱،۲. SDN اولیه

شبکه‌های فعال، علی‌رغم محدودیت‌های خود، یک طرح را ارائه می‌دهند که تلاشی است برای فراهم‌کردن انعطاف‌پذیری که استراتژی‌های کنونی SDN را پی‌ریزی می‌کند.شبکه‌های فعال بر روی مسائل محدودتر و واضح‌تر تمرکز کردند که منجر به جدایی بین صفحه کنترل و صفحه داده شد .این تمرکز با افزایش حجم ترافیک با افزایش اندازه اینترنت صورت گرفت که منجر به این شد که مدیران شبکه به دنبال یک رابط کنترل جدید برای شبکه‌های خود باشند .تکنولوژی‌های اولیه، روش‌های متفاوتی را برای ایجاد جدایی بین سطوح کنترل و داده، به کار بردند [۱۰] . بااین‌حال، بسیاری از این تکنولوژی‌ها استفاده ازAPI های استاندارد را برای کنترل صفحه داده پیشنهاد کردند. تکنولوژی‌های جدید طرح‌های متمایزکننده را برای کنترل متمرکز شبکه‌ها ایجاد کردند [۱۸] .

Open Flow .۳.۱

شبکه‌های OpenFlow (OF) با بازنگری چشمگیر در رابطه بین داده‌ها و سطوح کنترل دستگاه شبکه، خود را از زیرساخت‌های شبکه قدیمی متمایز می‌کنند. OpenFlow الگوی زیرساخت‌های سوئیچ بسیار قابل‌برنامه‌ریزی را در بر می‌گیرد و نرم‌افزار را قادر می‌سازد تا یک تصمیم مسیریابی جریان بهینه را بر اساس تقاضا محاسبه کند. برای شبکه‌های مدرن که باید به طور فزاینده‌ای با مجازی‌سازی میزبان و انتقال پویا برنامه‌های کاربردی سروکار دارند، OpenFlow چابکی لازم برای مدیریت هماهنگ‌سازی ارائه دهد. برای یک سوئیچ OpenFlow، صفحه داده، جایی است که جریان‌ها به‌صورت پویا در یک جدول جریان مشخص می‌شوند. جدول جریان شامل مجموعه‌ای از قوانین جریان است که مشخص می‌کند چگونه صفحه داده باید تمام جریان‌های فعال شبکه را پردازش کند. به طور خلاصه، قوانین جریان OpenFlow دستورالعمل‌هایی را ارائه می‌دهند که نحوه ارسال، تغییر یا رهاکردن هر بسته‌ای را که از سوئیچ فعال شده OF عبور می‌کند، کنترل می‌کند. صفحه کنترل سوئیچ برای پشتیبانی از پروتکل OpenFlow ساده شده است که به سوئیچ اجازه می‌دهد تا آمار و درخواست‌های جریان جدید را به یک کنترل‌کننده شبکه OpenFlow خارجی ارسال کند. در عوض، قوانین جریان را دریافت می‌کند که مجموعه قوانین جدول جریان آن را گسترش می‌دهد [۱۹].

پروتکل OpenFlow در ابتدا برای استقرار در شبکه‌های دانشگاه طراحی شد، اما با نشان‌دادن کاربرد پروتکل، علاقه فروشنده سخت‌افزار افزایش یافت. علاوه بر این، پروتکل شروع به استقرار در دیتاسنترها کرد و همراه با ارتقای پروتکل توسط توسعه‌دهندگان آن، فروشندگان سخت‌افزار شروع به ایجاد دستگاه‌های شبکه‌ای کردند که پشتیبانی سخت‌افزاری برای پروتکل دارند. کار بر روی سوئیچ‌های نرم‌افزاری، مانند Open vSwitch به محققان اجازه می‌دهد تا به‌سرعت برنامه‌های کاربردی جدید ایجاد کنند و سپس از سخت‌افزار فروشنده برای استقرار آنها استفاده کنند [۶].

با صفحه داده ارائه‌شده توسط هر دو کلید نرم‌افزاری و سخت‌افزاری، در برای توسعه معماری صفحه کنترل باز شد. معماری‌های زیادی برای کنترل پیشنهاد شده‌اند، مانند NOX، POX، ریو و ... ، و همه اینها قادر به برقراری ارتباط برای سوییچ‌ها با استفاده از پروتکل OpenFlow هستند. علاوه بر این، نیازی نیست که این ساختارها بر روی سخت‌افزار خاصی اجرا شوند، که اجازه می‌دهد سخت‌افزار کالا برای پیاده‌سازی کنترل‌کننده‌ها مورد استفاده قرار گیرد [۱۹].

علی‌رغم پشتیبانی ترکیبی برای OpenFlow توسط تحقیقات و جوامع صنعتی، باید توجه داشت که OpenFlow مترادف با SDN نیست. در حال حاضر OpenFlow محبوب‌ترین پیاده‌سازی SDN است، اما پیاده‌سازی دیگری می‌تواند وجود داشته باشد، مانند طراحی‌های خصوصی از نهاده‌ای صنعتی مانند سیسکو، مایکروسافت یا گوگل. همچنین باید توجه داشت که OpenFlow یک راه‌حل کامل نیست. تحقیقات چندین موضوع امنیتی را در پروتکل شناسایی کرده‌اند که در بخش ۴ به تفصیل به آن‌ها پرداخته شده‌است. مشکلاتی در به‌کارگیری به روز رسانی های جدید در پروتکل مشاهده شده‌است. تنها تعریف پروتکل اول، نسخه ۱. ، معمولا توسط فروشندگان پشتیبانی می‌شود، علی‌رغم میزان زیادی از حمایت جامعه، اما تعریف اخیر چندین تجدید نظر را پشت سر گذاشته‌است [۶].

شبکه‌های تعریف‌شده نرم‌افزار (SDN) اخیراً به‌عنوان یک پارادایم جدید قدرتمند برای توانمندسازی نوآوری در تحقیق و توسعه شبکه ظاهر شده است. ایده اصلی جداکردن صفحه کنترل شبکه از صفحه داده است. درحالی‌که کارهای قبلی قابل‌توجهی در این حوزه وجود داشته است اخیراً Openflow به حامل استاندارد برای شبکه SDN تبدیل شده است.

Openflow پروتکلی است که اجازه می‌دهد سوئیچ‌ها و روترهای حاوی جداول جریان داخلی توسط یک کنترل‌کننده خارجی مدیریت شوند. هر جدول جریان در داخل یک سوئیچ شامل مجموعه‌ای از ورودی‌های جریان است. هر ورودی جریان شامل یک هدر (در برابر بسته‌های ورودی مطابقت می‌شود) و همچنین مجموعه‌ای از اقدامات صفر یا بیشتر برای اعمال به بسته‌های منطبق است. تمام بسته‌های پردازش شده توسط سوئیچ با جداول جریان آن مقایسه می‌شوند. اگر یک ورودی جریان منطبق پیدا شود، هر اقدامی از آن ورودی بر روی بسته انجام می‌شود. اگر ورودی منطبقی یافت نشد، بسته به کنترل‌کننده ارسال می‌شود. ممکن است کنترل‌کننده تصمیم بگیرد که در این نقطه جریان‌ها را بر اساس هدر بسته در سوئیچ نصب کند. همچنین می‌تواند بسته را از طریق سوئیچ بدون تنظیم جریان ارسال کند. [۲۰].

به‌جای گسترش مکرر مشخصات OpenFlow، ما استدلال می‌کنیم که سوئیچ‌های آینده باید از مکانیزم‌های انعطاف‌پذیر برای تجزیه بسته‌ها و فیلدهای هدر منطبق پشتیبانی کنند که به برنامه‌های کنترل‌کننده اجازه می‌دهد تا از این قابلیت‌ها از طریق یک رابط مشترک باز استفاده کنند (یعنی یک API جدید "OpenFlow 2.0"). چنین رویکرد کلی نسبت به استاندارد OpenFlow امروزی ساده‌تر، ظریف‌تر و آینده‌نگرتر خواهد بود. [۲۱].

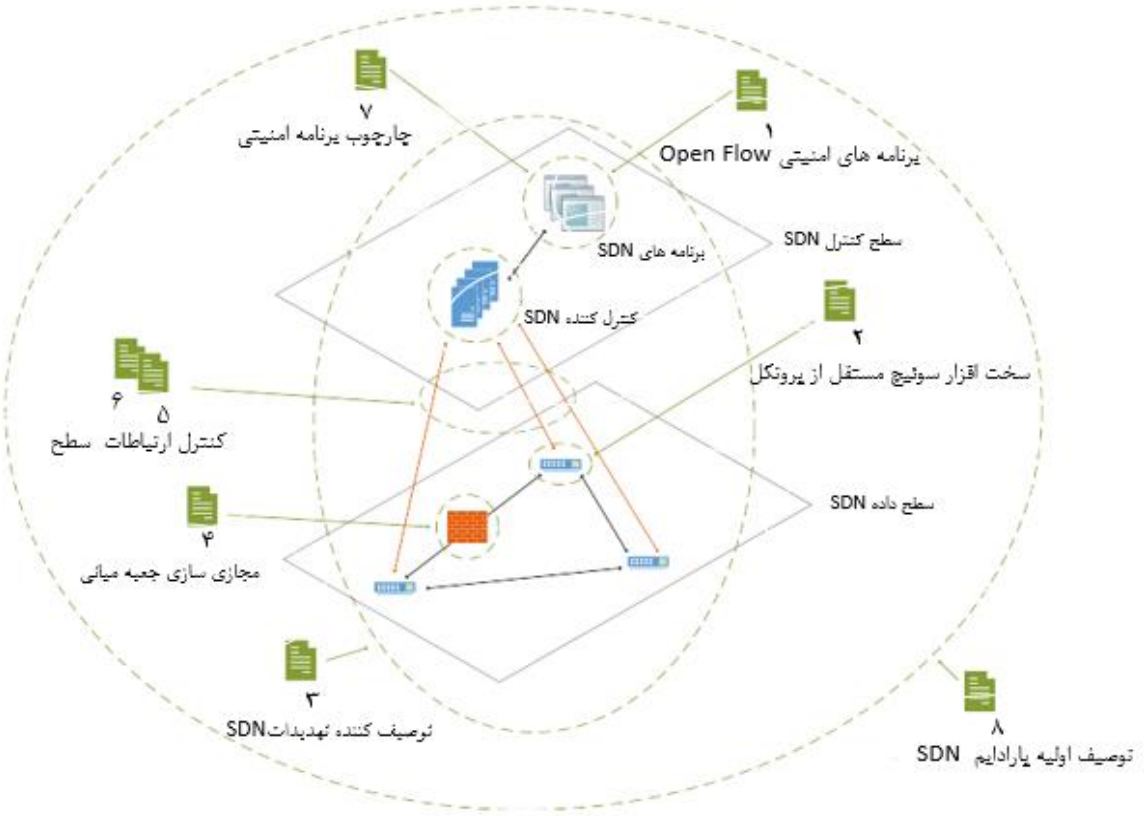
۲. بررسی امنیت SDN

برای این بررسی از تحقیقات فعلی SDN، ۸ مقاله در این زمینه مورد بررسی قرار گرفت که ارتباط مستقیمی با امنیت SDN دارند.برخی از این آثار قدیمی هستند، اما برای نشان‌دادن روند تحقیقات امنیتی در طول توسعه SDN کمک‌کننده هستند. تصویری از چگونگی ارتباط مقالات موردبحث در این بررسی با عناصر مختلف SDN در شکل ۱ نشان‌داده‌شده است.

راه‌اندازی و نگهداری یک شبکه کامپیوتری یک کار دشوار است. برای بیان خط‌مشی‌های شبکه سطح بالا، اپراتورهای شبکه باید هر دستگاه شبکه مجزا را – شامل سوئیچ‌ها، مسیریاب‌ها، جعبه‌های میانی و غیره - با استفاده از دستورات خاص پیکربندی کنند. علاوه بر پیچیدگی پیکربندی، شبکه‌ها پویا هستند و اپراتورها مکانیسم‌های کمی برای پاسخگویی خودکار به رویدادهای شبکه دارند یا اصلاً ندارند؛ بنابراین اجرای سیاست‌های موردنیاز در چنین محیطی که دائماً در حال تغییر است دشوار است.

با جداشدن صفحه کنترل از صفحه داده‌ای که زمینه را برای پارادایم کاری شبکه تعریف شده نرم‌افزاری ایجاد می‌کند، سوئیچ‌های شبکه ۱ برای دستگاه‌های بازدارنده ساده می‌شوند و منطق کنترل در یک کنترل‌کننده منطقی متمرکز اجرا می‌شود، هرچند در اصل به‌صورت فیزیکی. توزیع شده. درSDN ، کنترل‌کننده نهادی است که رفتار شبکه را دیکته می‌کند. تمرکز منطقی منطق کنترل در یک ماژول نرم‌افزاری که در یک سرور استاندارد - سیستم‌عامل شبکه اجرا می‌شود - چندین مزیت را ارائه می‌دهد. اولاً، اصلاح خط‌مشی‌های شبکه از طریق نرم‌افزار نسبت به پیکربندی دستگاه‌های سطح پایین ساده‌تر و کم خطاتر است. دوم، یک برنامه کنترلی می‌تواند به طور خودکار به تغییرات جعلی وضعیت شبکه واکنش نشان دهد و بنابراین سیاست‌های سطح بالا را در جای خود حفظ کند. سوم، متمرکزکردن منطق کنترل در یک کنترل‌کننده با دانش جهانی از وضعیت شبکه، توسعه عملکردهای شبکه پیچیده‌تر را ساده می‌کند؛ بنابراین، این توانایی برای برنامه‌ریزی شبکه به‌منظور کنترل صفحه داده‌های زیربنایی، ارزش پیشنهادی حیاتی SDN است.

SDN راه‌های جدیدی را برای حل مشکلات قدیمی در کار شبکه ارائه می‌کند، درحالی‌که به طور هم‌زمان امکان معرفی سیاست‌های شبکه پیچیده، مانند امنیت و قابلیت اطمینان را فراهم می‌کند. یک معماری SDN که به مدیران اجازه می‌دهد تا سیاست‌های کنترل دسترسی دقیق را اعمال کنند. بااین‌حال، امنیت و قابلیت اطمینان خود SDN تا کنون موضوعی نادیده گرفته شده است [۱۰].



شکل ۱: ارتباط بین مقالات مختلف که SDN را مورد بررسی قرار داده‌اند

مقالات نشان‌داده‌شده در شکل ۱

۱. امنیت SDN [۱۶]

۲. برنامه‌نویسی پردازشگر بسته مستقل از پروتکل [۲۱]

۳. به‌سوی شبکه ایمن و قابل‌اعتماد مبتنی بر نرم‌افزار [۱۰]

۴. فعال‌کردن پردازش سریع و پویا شبکه [۱۵]

۵. مقیاس مدیریت جریان برای شبکه با کارایی بالا [۱۱]

۶. جریان سوئیچ مقیاس‌پذیر و هوشیار مدیریت در SDN [۱۷]

۷. خدمات امنیتی ماژولار ترکیبی برای SDN [۱۱]

۸. بررسی کلی رویکردهای کنترل و مدیریت کار شبکه [۸]

۱،۲. تحقیقات اولیه SDN

تحقیقات شبکه‌های مبتنی بر نرم‌افزار ریشه در شبکه‌سازی فعال و تحقیقات در مورد معماری‌های ۴ بعدی، دارد. این دو اثر، به‌ویژه، آثاری از تحقیقات اولیه در زمینه‌های مرتبط با SDN هستند. اگرچه امنیت‌محور این دو اثر نبود، اما چندین ملاحظات امنیتی ارائه شده است. ما شاهد احیای جستجوی معماری اینترنت هستیم که از شبکه‌های مبتنی بر نرم‌افزار (SDN) برای غلبه بر محدودیت‌های شبکه فعلی استفاده می‌کند که بزرگ‌ترین آنها پیکرسازی شبکه است. برای ایجاد یک شبکه تکامل‌پذیر، SDN متمرکزکردن صفحه کنترل و صفحه داده را پیشنهاد می‌کند. ایجاد یک صفحه کنترل نرم‌افزاری که به‌راحتی قابل‌تغییر است، SDN امکان سوئیچینگ و مسیریابی بسته‌های انعطاف‌پذیر را فراهم می‌کند و پذیرش پروتکل‌های جدید اینترنتی (به‌عنوان‌مثال، IPv6) یا تغییرات در پروتکل‌های موجود را تسریع کند [۱۵] .

۲،۲. تحقیقات اخیر SDN

تحقیقات اخیر در مورد امنیت SDN دو مسیر خاص را دنبال می‌کند، تحقیقات OpenFlow و جستجوی عمومی SDN. ازآنجایی‌که OpenFlow برجسته‌ترین استقرار SDN است، باعث کاهش مشکلات امنیتی در پروتکل وجود دارد، زیرا در بسیاری از تنظیمات تولید استفاده می‌شود. بااین‌حال، سایر تحقیقات امنیتی در استراتژی‌های کلی SDN امکان پیشرفت به سمت پروتکل‌های جدید فراتر از OpenFlow را فراهم می‌کند.

۱،۲،۲. امنیت SDN

OpenFlow با وجود محبوبیتش، تنها روش پیاده‌سازی SDN نیست. این کار روشی را برای تطبیق قوانین با فیلدهای هدر که توسط برنامه‌نویس تعریف شده است پیشنهاد می‌کند که به سیستم اجازه می‌دهد از پروتکل‌های جدیدی پشتیبانی کند که دارای هدرهایی با اندازه‌های متفاوت با پروتکل‌های موجود هستند [۶] .

۲،۲،۲. OpenFlow Security

همان‌طور که قبلاً گفته شد، پروتکل OpenFlow بیشترین میزان پشتیبانی را از جامعه تحقیقاتی و صنعت دریافت کرده است. با توجه به محبوبیت این پروتکل، مسائل امنیتی که در مشخصات اولیه به آنها توجه نشده بود، اکنون در اولویت قرار دارند، زیرا این پروتکل در سیستم‌های تولید استفاده می‌شود [۶] .

۳. امنیت OpenFlow

سه دسته از تحقیقات مربوط به امنیت OpenFlow وجود دارد. اولین مورد تحقیقاتی است که سعی در حل مسائل مربوط به مقیاس‌پذیری و تحمل خطا دارد که در طراحی کنترلر OpenFlow وجود دارد. این مسائل مستقیماً به دلیل نگرانی‌های امنیتی ایجاد نمی‌شوند، اما مستقیماً قابل‌اجرا هستند؛ زیرا دوام شبکه را تحت بارگذاری بهبود می‌بخشند، همان‌طور که در طول حمله DoS مشاهده می‌شود. دسته دوم تحقیقاتی است که مستقیماً به توانایی‌های آسیب‌پذیر امنیتی موجود در مشخصات OpenFlow می‌پردازد. مهم‌ترین این مسائل، گلوگاه ارتباطی بین این دو صفحات داده و کنترل است که می‌توانند به‌راحتی در بسیاری از موقعیت‌ها با ترافیک کنترلی غرق شوند. دسته سوم تحقیقاتی است که از OpenFlow برای حل آسیب‌پذیری‌های امنیتی موجود استفاده می‌کند. با توجه به قابل‌مشاهده بودن شبکه که برای کنترل‌کننده ثابت شده است، برنامه‌ها می‌توانند از پروتکل برای ایجاد خط‌مشی‌های گسترده شبکه استفاده کنند که مؤثرتر از آنچه در شبکه‌های سنتی موجود است [۹] .

۱،۳. عملکرد و تحمل خطا

با فعال‌کردن یک شبکه مقاوم‌تر در برابر خطا، سیستم در برابر حملاتی مانند DoS مقاوم می‌شود. با افزایش عملکرد، کنترلر بهتر می‌تواند به رویدادهایی که در صفحه داده رخ می‌دهد، پاسخ سریع دهد. کار ارائه شده توسط محققان کنترل‌کننده‌های توزیع‌شده‌ای را ایجاد

می‌کند که امکان افزایش کارایی را از طریق متعادل‌سازی بار روی یک مجموعه کنترل‌کننده و تحمل خطای کنترل‌کننده را فراهم می‌کند، زیرا مجموعه کنترل‌کننده‌ها قادر به بازیابی هستند. محققان خاطرنشان می‌کنند که مزیت اصلی SDN ها، یعنی جداسازی صفحه کنترل و صفحه داده، بسیاری از آسیب‌پذیری‌هایی را ایجاد می‌کند که آنها شناسایی می‌کنند. محققان همچنین خاطرنشان می‌کنند که کار کمی برای رفع این آسیب‌پذیری‌ها انجام شده است و به همین دلیل، چندین راه‌حل ممکن برای این آسیب‌پذیری‌ها پیشنهاد می‌کنند [۱۸] .

۲،۳. برنامه‌های امنیتی در OpenFlow

تحقیقات قبلی که از نظر امنیت OpenFlow موردبحث قرار گرفت، بر ارائه راه‌حل‌هایی برای آسیب‌پذیری‌های شناخته شده در پروتکل، یا استفاده از مکانیسم‌های خاص پروتکل برای محافظت از زیرساخت شبکه متمرکز بود. بااین‌حال، کار کمی در مورد اجرای استراتژی‌های شناخته شده کاهش تهدید شبکه، مانند تشخیص نفوذ و فایروال‌ها که از سیستم‌های میزبان نهایی در برابر حمله محافظت می‌کنند، ارائه شده است. تحقیقات ارائه شده قبلی هدف را به‌عنوان کار شبکه و پروتکل OpenFlow به طور خاص تعریف کرده است؛ بنابراین کارهای بیشتری در این زمینه می‌تواند انجام پذیرد [۶].

۴. امنیت عمومی SDN

همان‌طور که قبلاً توضیح داده شد، سیستمی است که امکان ایجاد توابع شبکه ماژولار ، مانند توابعی که در جعبه‌های میانی (فایروال‌ها، پراکسی‌ها، سیستم‌های تشخیص نفوذ و غیره) ایجاد می‌شوند و کپسوله‌کردن آنها در یک ماشین مجازی بسیار سبک‌وزن این ماشین‌های مجازی را می‌توان به‌سرعت ثابت کرد، و بنابراین به طور بالقوه می‌توانند بر اساس تقاضا ایجاد و از بین بروند. همچنین یک معماری برای اجرای کلی SDN نسبت به OpenFlow ارائه می‌کند که اجازه می‌دهد قوانین در فیلدهای هدر تعریف‌شده توسط برنامه‌نویس تطبیق داده شوند، و اجازه می‌دهد تا یک فرایند کامپایل‌سازی برای پشتیبانی از پیاده‌سازی‌های سخت‌افزاری متنوع باشد. هر دوی این فناوری‌ها تأثیرات مستقیمی بر امنیت SDN دارند [۶] .

۱،۴. ClickOS

پیامدهای ClickOS حتی برای شبکه‌های دارای OpenFlow قابل‌اجرا هستند، اگرچه تمرکز سیستم بر جعبه‌های میانی است که به‌طورکلی به مسائل امنیتی مبتنی بر میزبان می‌پردازد. بااین‌حال، با توجه به سرعتی که می‌توان ماشین مجازی‌های جعبه‌های میانی را نمونه‌سازی کرد، امکان توسعه اپلیکیشن‌های امنیتی جعبه‌های میانی جدید بر اساس تقاضا وجود دارد. به‌این‌ترتیب، تعادل بار بالقوه ترافیک شبکه را می‌توان انجام داد که امکان انعطاف‌پذیری بیشتری را برای اپراتورهای شبکه در نحوه اجرای سیاست‌های امنیتی در سراسر شبکه فراهم می‌کند [۱۵] .

۲،۴. پردازنده‌های بسته‌بندی مستقل از پروتکل قابل‌برنامه‌ریزی

پیامدهای این کار بیشتر به نحوه تلاش برنامه‌نویسان حرفه‌ای برای شناسایی و طبقه‌بندی حملات مربوط می‌شود. در سیستمی که برنامه‌نویسان می‌توانند فیلدهای هدر را که می‌خواهند با ورودی‌های جریان مطابقت دهند، انعطاف‌پذیری بیشتری را برای برنامه‌نویسان فراهم می‌کند تا هم نحوه عبور ترافیک حمله از شبکه را شناسایی و هم کنترل کنند.

۵. تحقیقات امنیتی آینده

پس از بررسی آثار مختلف اهمیت OpenFlow برای جامعه تحقیقاتی باید مورد توجه قرار گیرد، زیرا این پروتکل توسط اکثر تحقیقات امنیتی اخیر متمرکز شده است. این به دلیل پذیرش صنعت از پروتکل منطقی است که جامعه تحقیقاتی را به تلاش برای بهبود مشخصات و استفاده از آن برای پیاده‌سازی راه‌حل‌های امنیتی ترغیب می‌کند. بااین‌حال، تحقیقات دیگر فناوری‌هایی را پیشنهاد می‌کنند که انعطاف‌پذیری بیشتری نسبت به آنچه توسط OpenFlow ثابت شده است، ارائه می‌دهد. چنین فناوری‌هایی ابزارهای قدرتمندی را در

اختیار اپراتورهای شبکه و توسعه‌دهندگان برنامه‌ها قرار می‌دهند، اما متأسفانه، پیاده‌سازی برخی از آن‌ها دشوار است؛ زیرا مستلزم تغییراتی در سخت‌افزار است.

۶. نتیجه‌گیری

در این مقاله، مروری بر تحقیقات فعلی امنیت SDN همراه با پیش‌بینی مسیرهای تحقیقات آینده انجام شد. پژوهش انجام شده به دودسته محبوب‌ترین استقرار SDN، OpenFlow، و تحقیقات عمومی SDN تقسیم می‌شوند. این تقسیم به دلیل حمایت گسترده جامعه و صنعت از OpenFlow است که انگیزه تحقیقات برای کمک به بهبود پروتکل در محیط‌های تولیدی است.

این تحقیق تلاش برای بررسی امنیت SDN را نشان می‌دهد که تا همین اواخر حتی در زمینه OpenFlow بسیار کمبود. مزایای ارائه شده توسط SDN نیز چالش‌های امنیتی جدیدی را معرفی می‌کند. تحقیقاتی که در اینجا موردبحث قرار می‌گیرند، راه‌حل‌های معتبری را برای برخی از این مسائل امنیتی نشان می‌دهند، اما باید کارهای بیشتری انجام شود تا بتوان این آسیب‌پذیری‌ها را به طور رضایت‌بخش کاهش دهد.

منابع

[1] Gens, F. (2012). IDC Predictions 2013: Competing on the 3rd Platform. Int. Data Corporation.

[2] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

[3]Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. IEEE Communications Surveys & Tutorials, 17(1), 27-51.

[4]Carapinha, J., Feil, P., Weissmann, P., Thorsteinsson, S. E., Etemoğlu, Ç., Ingþórsson, Ó., ... & Melo, M. (2010). Network virtualization-opportunities and challenges for operators. In Future Internet-FIS 2010: Third Future Internet Symposium, Berlin, Germany, September 20-22, 2010. Proceedings 3 (pp. 138-147). Springer Berlin Heidelberg.

[5]Joseph, D. A., Tavakoli, A., & Stoica, I. (2008, August). A policy-aware switching layer for data centers. In Proceedings of the ACM SIGCOMM 2008 conference on Data communication (pp. 51-62).

[6]Coughlin, M. (2014). A survey of SDN security research. University of Colorado Boulder.

[7] Tennenhouse, D. L., Smith, J. M., Sincoskie, W. D., Wetherall, D. J., & Minden, G. J. (1997). A survey of active network research. IEEE communications Magazine, 35(1), 80-86.

[8] Greenberg, A., Hjalmtysson, G., Maltz, D. A., Myers, A., Rexford, J., Xie, G., ... & Zhang, H. (2005). A clean slate 4D approach to network control and management. ACM SIGCOMM Computer Communication Review, 35(5), 41-54.

[9] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. ACM SIGCOMM computer communication review, 38(2), 69-74.

[10] Kreutz, D., Ramos, F. M., & Verissimo, P. (2013, August). Towards secure and dependable software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 55-60) .

[11] Curtis, A. R., Mogul, J. C., Tourrilhes, J., Yalagandula, P., Sharma, P., & Banerjee, S. (2011, August). DevoFlow: Scaling flow management for high-performance networks. In Proceedings of the ACM SIGCOMM 2011 Conference (pp. 254-265).

[12] Shin, S., Yegneswaran, V., Porras, P., & Gu, G. (2013, November). Avant-guard: Scalable and vigilant switch flow management in software-defined networks. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (pp. 413-424).

[13] Hassas Yeganeh, S., & Ganjali, Y. (2012, August). Kandoo: a framework for efficient and scalable offloading of control applications. In Proceedings of the first workshop on Hot topics in software defined networks (pp. 19-24)

[14] Dixit, A., Hao, F., Mukherjee, S., Lakshman, T. V., & Kompella, R. (2013). Towards an elastic distributed SDN controller. ACM SIGCOMM computer communication review, 43(4), 7-12.

[15] Martins, J., Ahmed, M., Raiciu, C., & Huici, F. (2013, August). Enabling fast, dynamic network processing with clickos. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 67-72).

[16] Mehdi, S. A., Khalid, J., & Khayam, S. A. (2011). Revisiting traffic anomaly detection using software defined networking. In Recent Advances in Intrusion Detection: 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. Proceedings 14 (pp. 161-180). Springer Berlin Heidelberg.

[17] Shin, S. W., Porras, P., Yegneswara, V., Fong, M., Gu, G., & Tyson, M. (2013, February). Fresco: Modular composable security services for software-defined networks. In 20th annual network & distributed system security symposium. Ndss.

[18] Feamster, N., Rexford, J., & Zegura, E. (2013). The road to SDN: An intellectual history of programmable networks. Queue, 11(12), 20-40.

[19] Shin, S. W., Porras, P., Yegneswara, V., Fong, M., Gu, G., & Tyson, M. (2013, February). Fresco: Modular composable security services for software-defined networks. In 20th annual network & distributed system security symposium. Ndss.

[20] Mehdi, S. A., Khalid, J., & Khayam, S. A. (2011). Revisiting traffic anomaly detection using software defined networking. In Recent Advances in Intrusion Detection: 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. Proceedings 14 (pp. 161-180). Springer Berlin Heidelberg.

[21] Hamadi, S., Blaiech, K., & Cherkaoui, O. (2015, July). Semantic-based forwarding model for network devices. In 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS) (pp. 1-7). IEEE

[22] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., ... & Walker, D. (2014). P4: Programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review, 44(3), 87-95.