3rd. International Conference on September 2017
Information Technology, Computer and Telecommunications
سومین کنفرانس بین المللی شهریورماه ۱۳۹۶
فناوری اطلاعات، مهندسی کامپیوتر و مخابرات

# دسته‌بندی حملات امنیتی و راه‌حل‌های ارائه شده در شبکه های نرم‌افزارمحور

یاسر تیمورزاده[1]

[1] گروه فناوری اطلاعات و ارتباطات، دانشکده آموزشهای الکترونیکی، دانشگاه شیراز، teymurzade@gmail.com

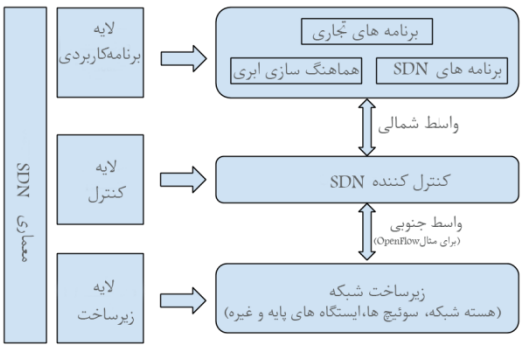
**چکیده**

شبکه‌های نرم‌افزار محور (SDN) برخلاف شبکه‌های سنتی که در آن سطوح کنترل و داده در ابزارهای زیرساخت شبکه همچون سوئیچ‌ها، مسیریاب‌ها و غیره درون همان ابزار قرار دارند، این دو سطح را از هم جدا کرده و تمامی تصمیم‌های کلیدی در خصوص هدایت بسته‌ها در کنترلر شبکه گرفته‌شده و ابزارهای سطح داده بسته‌ها را بر مبنای این تصمیم‌ها پیش می‌رانند. هرچند SDN مزیت‌های فراوانی در خصوص برنامه‌نویسی،کنترل شبکه و اعمال سیاست‌ها به ارمغان آورده، اما با مسائل و مشکلات امنیتی متفاوتی نسبت به شبکه‌های سنتی روبه‌رو است. در این مقاله چالش‌های خاص SDN به هفت دسته کلی دسترسی غیرمجاز، نشت داده‌ها، تغییر داده‌ها، برنامه‌های بدخواه، منع سرویس، مسائل پیکربندی و امنیت سطح سیستم SDN تقسیم شده و هرکدام جداگانه موردبررسی قرار گرفته است. سپس برای هرکدام از این دسته‌بندی‌ها، راه‌حل‌های امنیتی که تاکنون ارائه‌شده به‌اختصار معرفی و در انتها نتیجه‌گیری می‌گردد.

**واژه های کلیدی:** شبکه‌های نرم‌افزار محور، SDN، امنیت شبکه، حملات امنیتی، روش های مقابله

## ۱- مقدمه

بنیاد شبکه‌بندی باز شبکه نرم‌افزار محور (SDN) را جدایی فیزیکی سطح کنترل از سطح پیش‌رانش بسته‌ها تعریف می‌کند که در آن سطح کنترل ابزارهای متعددی را نظارت می‌کند. در شبکه‌بندی متداول، ابزارهای سنتی زیرساخت مانند سوئیچ‌ها و مسیریاب‌ها موجودیت‌هایی مستقل با سطوح هوشمندی متفاوت هستند که هرکدام به‌صورت مجزا با توجه به اطلاعات متعدد و پیچیده در خصوص سرنوشت بسته‌های دریافتی تصمیم‌گیری می‌کنند. در این ابزارها لایه کنترل و لایه داده در کنار یکدیگر و درون همان ابزار قرار دارند. تصمیم‌گیری‌های مرتبط با هدایت بسته‌ها در لایه کنترل انجام‌شده و سپس در لایه داده بسته‌ها به سمت خروجی مناسب هدایت می‌شوند. اما در SDN، کنترلر به‌عنوان مغز متفکر و یک جزء فیزیکی مستقل در شبکه معرفی شده که وظیفه اتخاذ کلیه تصمیم‌ها در خصوص هدایت و پیش‌رانش بسته‌ها را بر عهده دارد. به‌عبارت‌دیگر در شبکه‌های SDN خود ابزارها نقشی در تصمیم‌گیری‌های مرتبط با هدایت بسته ندارند، بلکه باید قبل از هدایت، بسته‌ها از طریق واسط‌های برنامه‌نویسی که در SDN به‌عنوان واسط جنوبی[1] شناخته می‌شوند (مانندOpenFlow ) با کنترلر ارتباط برقرار، کسب تکلیف کرده و سپس بسته‌ها را هدایت کنند . از طرفی برنامه‌های SDN هم می‌توانند به‌صورت مستقیم، صریح و با استفاده از رابط‌های برنامه‌نویسی که واسط برنامه‌نویسی شمالی[2] شناخته می‌شود نیازها و رفتارهای شبکه‌ای دلخواه خود را به کنترلر شبکه منتقل کنند. به شکل ۱ دقت کنید.




**شکل ۱: معماری SDN**

[1] Southbound

[2] Northbound

## ۲- حملات و آسیب‌پذیری‌ها در SDN

برای بررسی مسائل امنیتی SDN، منبع [۱] با توجه به اینکه کدام‌یک از لایه/ واسطها در معماری SDN تحت تأثیر قرار گرفته، آنها را به هفت گروه مختلف دسته‌بندی کرده و برای هرکدام نمونه‌های مشخصی که ممکن است در آن مسئله امنیتی به وجود آید ذکر کرده است. در ادامه مسائل امنیتی که به صورت مستقیم مرتبط با SDN است موردبررسی قرارگرفته و جزئیات سناریوی رخداد حمله نیز ذکرشده است.

### ۲-۱ دسترسی غیرمجاز

این دسته از حملات، مرتبط با کنترل دسترسی است. یکی از ویژگی‌های ذاتی SDN کنترل متمرکزشده است. البته کنترل به صورت فیزیکی می‌تواند توزیع‌شده باشد. بنابراین در معماری SDN این امکان وجود دارد که چندین کنترلر به سطح داده شبکه دسترسی داشته باشند. برنامه‌های کاربردی مختلف هم می‌توانند به این کنترلرها متصل شوند. کنترلر سطح تجریدی را برای برنامه‌ها فراهم می‌کند تا بتوانند حالت شبکه را بخوانند یا بنویسند. این کار نیز به‌واقع یک سطح از کنترل شبکه است. اگر حمله‌کننده بتواند خود را به‌عنوان کنترلر یا برنامه کاربردی جا بزند ممکن است بتواند به منابع شبکه دسترسی پیداکرده و عملکرد شبکه را تغییر دهد.

### ۲-۲ نشت داده‌ها

در توصیف سوئیچ OpenFlow برای تعیین وضعیت بسته، اعمال پیش‌رانش بسته، حذف بسته و ارسال بسته به کنترلر تعریف شده است. ممکن است حمله‌کننده عملی را که برای نوع خاصی از بسته‌ها تعریف‌شده به‌وسیله تحلیل زمان‌بندی پردازش بسته‌ها مشخص کرده باشد. برای مثال زمان پردازش بسته زمانی که مستقیماً از پورت ورودی به پورت خروجی هدایت می‌شود کوتاه‌تر از زمانی است که بسته برای پردازش باید به کنترلر هدایت شود. حمله‌کننده می‌تواند پیکربندی کنشی/واکنشی سوئیچ را بفهمد و با استفاده از بسته‌های طراحی‌شده بیشتر، اطلاعات افزون‌تری را در خصوص پیکربندی ابزار به دست آورد و بعدازاینکه تشخیص داد چه بسته (هایی) به سمت کنترلر هدایت می‌شوند می‌تواند مجموعه‌ای از درخواست‌های جریان جعلی را تولید کند که منجر به حملات منع سرویس شود.

یک چالش حل‌نشده دیگر در معماری SDN ذخیره‌سازی امن اعتبارنامه‌ها[3] (برای مثال کلیدها و گواهینامه‌ها) برای چندین شبکه منطقی در سطح داده قابل برنامه‌نویسی است. برای مثال OF-Config می‌تواند چندین سوئیچ منطقی OpenFlow را روی یک سوئیچ باقابلیت OpenFlow نمونه‌سازی کند [۲]. فرض کنید هرکدام از این نمونه‌ها مربوط به مشتری متفاوتی باشد. در این صورت اگر شبکه‌های منطقی و اعتبارنامه‌های مرتبط به‌صورت امن ایزوله یا کانتینر نشده باشند می‌تواند منجر به نشت داده‌هایی شود که مهاجم می‌تواند عملکرد نمونه را در اختیار بگیرد.

[3] Credentials

**جدول ۱: دسته بندی مسائل امنیتی شبکه های SDN**

| مسئله امنیتی | لایه‌های تحت تأثیر در معماری SDN | | | | |
|---|---|---|---|---|---|
| | لایه‌برنامه | واسط‌شمالی | لایه‌کنترل | واسط‌جنوبی | لایه‌داده |
| دسترسی غیرمجاز | | | | | |
| دسترسی غیرمجاز به کنترلر/در اختیار گرفتن کنترلر | | | ✓ | ✓ | ✓ |
| برنامه احراز هویت نشده/بدون مجوز | ✓ | ✓ | ✓ | | |
| نشت داده‌ها | | | | | |
| کشف قانون جریان (حمله کانال جانبی روی بافر ورودی) | | | | | ✓ |
| مدیریت اعتبارنامه‌ها (کلیدها،گواهینامه‌ها در هر شبکه منطقی) | | | | | ✓ |
| کشف سیاست پیش‌رانش (تحلیل زمان‌بندی پردازش بسته‌ها) | | | ✓ | ✓ | ✓ |
| تغییر داده‌ها | | | | | |
| تغییر قانون پیش راندن برای تغییر بسته‌ها (حملات واسطه) | | | ✓ | ✓ | ✓ |
| برنامه‌های بدخواه/در اختیار گرفته‌شده | | | | | |
| واردکردن قوانین ساختگی | ✓ | ✓ | ✓ | | |
| منع سرویس | | | | | |
| به سیل بستن ارتباط سوئیچ با کنترلر | | | ✓ | ✓ | ✓ |
| به سیل بستن جدول جریان سوئیچ | | | | | ✓ |
| مسائل پیکربندی | | | | | |
| فقدان پذیرش TLS (یا سایر تکنیک‌های احراز هویت) | ✓ | ✓ | ✓ | ✓ | ✓ |
| اعمال سیاست‌ها | ✓ | ✓ | ✓ | | |
| فقدان تأمین سازی امن | ✓ | ✓ | ✓ | ✓ | ✓ |
| امنیت سطح سیستم SDN | | | | | |
| فقدان شفافیت روی وضعیت شبکه | | | ✓ | ✓ | ✓ |

### ۳-۲ تغییر داده‌ها

کنترلر می‌تواند ابزارهای شبکه را برای کنترل جریان ترافیک برنامه‌نویسی کند. اگر حمله‌کننده کنترلر را در اختیار بگیرد درواقع می‌تواند کل شبکه را در اختیار بگیرد و با توجه به سطح امتیازاتی که به دست آورده، قوانین جریان‌ها را در ابزارهای شبکه تغییر داده و سپس بسته‌ها را با توجه به منافع خودش در شبکه هدایت کند. در توصیفات OpenFlow به‌کارگیری احراز هویت دوسویه بین کنترلرها و سوئیچ‌ها با استفاده از TLS مورد تأکید قرارگرفته، اما این ویژگی امنیتی اجباری نیست و در بسیاری موارد استفاده نشده است. اگر حمله‌کننده‌ای بین این دو قرارگرفته باشد می‌تواند خود را به‌عنوان کنترلر جا زده و حملات متعددی را اجرا کند. برای مثال پیام‌های کنترل ارسال‌شده توسط کنترلر را تغییر دهد یا پیام‌های راه‌اندازی مجدد را برای قطع ارتباط تزریق کند. به‌علاوه خود اجزای واسط بین سطح کنترل و سطح داده بایستی امنیت کافی داشته باشند و موجب مسائل امنیتی بیشتر نشوند. یک حمله واسطه[4] زمانی اتفاق می‌افتد که حمله کنند قادر باشد پیام‌های ردوبدل شده بین دو قربانی را جهت بازبینی و تغییر یا تزریق پیام‌ها به کانال ارتباطی بقاپد[5]. چنین فرایندی زمانی که هیچ مکانیزم احراز هویتی بین دونقطه انتهایی ارتباط وجود نداشته باشد امکان‌پذیر است. برای مثال FlowVisor یک مجازی ساز شبکه برای OpenFlow است و چون مکانیزم های مناسب ایزوله سازی را ارائه نکرده این امکان را به حمله‌کننده می‌دهد تا حملات تغییر را روی موجودیت‌های ارتباط اجرا کند. مسئله تغییر داده‌ها در معماری سطوح–مجزای SDN یکی از نگرانی‌های اصلی است [۱].

[4] Man In the Middle

[5] Intercept

### ۲-۴ برنامه‌های بدخواه/در اختیار گرفته‌شده[6]

با توجه به اینکه کنترلر به‌عنوان یک تجرید از سطح داده برای برنامه‌ها عمل می‌کند و SDN این امکان را به برنامه‌های کاربردی طرف ثالث می‌دهد تا با معماری شبکه یکپارچه شوند یک برنامه کاربردی بدخواه می‌تواند همچون یک کنترلر در اختیار گرفته‌شده اثر مخربی روی شبکه داشته باشد. به همین ترتیب یک برنامه کاربردی با طراحی بد یا خطادار می‌تواند ناخواسته سبب ایجاد آسیب‌پذیری در سیستم شود. برای مثال حمله‌کننده می‌تواند یک خطای شناخته‌شده را مورداستفاده قرار داده و برنامه کاربردی را وارد یک حالت ناامن کند.

### ۲-۵ منع سرویس

یکی از نقاط ضعف اصلی SDN به‌واسطه خود معماری SDN است: کنترلر مرکزی به جای کنترل توزیع شده در ابزارها و جدایی سطوح کنترل و داده از یکدیگر. حمله‌کننده می‌تواند به‌واسطه مسیر ارتباطی بین کنترلر و ابزار شبکه با ارسال سیل‌آسای بسته‌هایی که نیاز به تصمیم در خصوص قانون جریان دارند عملاً آن را برای کاربران مجاز شبکه غیرقابل‌دسترس کند. حمله منع سرویس مشابهی را می‌توان در سطح زیرساخت انجام داد. با توجه به این‌که منابع حافظه‌ای در سطح زیرساخت محدود است حمله‌کننده می‌تواند با ایجاد قوانین جریان کاذب، این منابع را کاملاً به خود اختصاص دهد.

### ۲-۶ مسائل پیکربندی

با شناسایی آسیب‌پذیری‌های شبکه، سیاست‌ها و پروتکل‌های امنیتی شبکه هم در حال ایجاد هستند اما اگر این سیاست‌ها و پروتکل‌ها به‌کارگرفته نشوند یا بدون درک پیامدهای امنیتی، غیرفعال شوند نمی‌توانند حفاظت چندانی در پی داشته باشند. بسیار مهم است که مدیران شبکه در شبکه‌های مبتنی بر SDN سیاست‌های امنیتی همچون TLS را اعمال کنند. همچنان که در جدول ۱ نیز اشاره‌شده، پیکربندی ناقص یا نادرست قابلیت‌های امنیتی می‌تواند تمامی لایه‌های معماری را تحت تأثیر قرار دهد.

باز کردن واسط‌ها بین اجزای شبکه می‌تواند بالقوه سبب ورود آسیب‌پذیری‌های قابل‌توجهی به شبکه شود که می‌تواند هم از این نظر که ابزارهایی از فروشندگان مختلف باهم ارتباط برقرار می‌کنند و هم از این نظر که ارتباط داده‌ای و کنترلی روی این واسط‌های جدید انجام می‌شود مدنظر قرار گیرد. یکی از قابلیت‌هایی که SDN ها به ارمغان آورده‌اند برنامه‌نویسی آسان شبکه و ایجاد سیاست‌های جریان پویا است. به‌واقع همین مزیت امنیتی SDN می‌تواند منجر به آسیب‌پذیری‌های امنیتی نیز بشود. برنامه‌های کاربردی متعددی سیاست‌ها را ایجاد کرده‌اند،از طرفی این سیاست‌ها روی ابزارهای متعددی اعمال‌شده‌اند ممکن است تضادهایی بین این سیاست‌ها به وجود آید که برای حل آن‌ها با یکدیگر، باید این ناسازگاری‌ها را پیدا کرد که همین امر می‌تواند چالش‌برانگیز بوده و موجب سوءاستفاده مهاجمین گردد [۱].

### ۲-۷ امنیت سطح سیستم SDN:

در SDN، نگرانی‌های امنیتی متعددی هم در سطح سیستم وجود دارد. یک نگرانی عمده امنیتی، برآورده کردن پروسه ممیزی[7] در شبکه است. این‌که بتوان ازنظر سازگاری و عملکردی مجموعه‌ای از ابزارهای شبکه‌ای قابل‌کنترل ایجاد کرد از اهمیتی حیاتی برخوردار است. برای مثال باید بدانیم که چه ابزارهایی در حال حاضر مشغول به کار هستند، ارتباطشان با شبکه چطور است و غیره. برای مثال سوئیچ OpenFlow می‌تواند دریکی از دو حالت fail-secure یا fail-standalone عمل کند[۲]. زمانی که ارتباط سوئیچ با کنترلر قطع‌شده، منطق داخلی سوئیچ می‌تواند یکی از این دو حالت را انتخاب کند. از دیدگاه عملکردی برای اپراتور مهم است که بداند در زمان قطع ارتباط سوئیچ چه حالت عملیاتی را انتخاب کرده، رفتار پیش‌رانش بسته‌ها به چه صورت بوده است. قابلیت بیرون کشیدن اطلاعات این‌چنینی (سطر حسابرسی) و ممیزی بسیار مهم است و باید به نحوی در [illegible] مدیریت شود.

**جدول۲: راه حل های امنیتی ارائه شده برای مشکلات امنیتی SDN** [illegible] ارای ارتباط مجدد چه بوده

مختصصان شبکه برای اینکه بتوانند دسترسی درستی را به منابع فراهم کنند باید تکنیک‌های متعددی را استفاده کنند که پیچیدگی شبکه را افزایش می‌دهد و مدیریت آن را سخت می‌کند. ویژگی‌های اصلی یک ارتباط امن شبکه‌ای عبارت‌اند از: محرمانگی، تمامیت داده‌ها و دسترس‌پذیری آن‌ها. این ویژگی‌ها با تکنیک‌های احراز هویت، مجوز دهی و رمزنگاری پشتیبانی می‌شوند. جمع این ویژگی‌ها شبکه‌ای را به دست می‌دهد که در آن داده‌ها، دارایی‌های شبکه (برای مثال ابزارها) و تراکنش‌های ارتباطی، امن بوده و از حمله‌های بدخواهانه یا صدمه ناخواسته در امان هستند. در این قسمت بخشی از آسیب‌پذیری‌های احتمالی در شبکه‌ای با قابلیت SDN موردبحث قرار گرفت. برای اینکه زیرساخت شبکه، عملکرد امنی داشته باشد باید حفاظت‌هایی لازم را در معماری شبکه تعبیه کرد. در قسمت بعدی راه‌حل‌های ارائه‌شده برای مسائل امنیتی در SDN مرور می‌شود.

---

[6] Compromised

[7] audit

| مسئله امنیتی | لایه‌های تحت تأثیر در معماری SDN | | | | |
|---|---|---|---|---|---|
| | لایه‌برنامه | واسط‌شمالی | لایه‌کنترل | واسط‌جنوبی | لایه‌داده |
| دسترسی غیرمجاز | | | | | |
| Securing Distributed Control, Byzantine-Resilient SDN | | | ✓ | ✓ | |
| Authentication for Resilience | | | ✓ | | |
| PermOF | | | | | |
| OperationCheckPoint | ✓ | ✓ | | | |
| SE-Floodlight | ✓ | ✓ | ✓ | ✓ | |
| AuthFlow | ✓ | | ✓ | ✓ | ✓ |
| نشت داده‌ها | - | | | | |
| تغییر داده‌ها | - | | | | |
| برنامه‌های بدخواه | | | | | |
| FortNOX | ✓ | ✓ | ✓ | ✓ | |
| ROSEMARY | ✓ | | ✓ | | |
| LegoSDN | ✓ | ✓ | ✓ | | |
| منع سرویس | | | | | |
| AVANT-GUARD, CPRecovery | | | ✓ | ✓ | ✓ |
| VAVE | ✓ | | ✓ | ✓ | ✓ |
| Delegating Network Security | ✓ | ✓ | ✓ | ✓ | ✓ |
| مسائل پیکربندی | | | | | |
| NICE | ✓ | ✓ | | ✓ | |
| Flow-Checker, Flover, Anteater, VeriFlow, NetPlumber | ✓ | ✓ | ✓ | ✓ | |
| Security-Enhanced Firewall, FlowGuard, LPM | ✓ | | ✓ | ✓ | ✓ |
| Frenetic, Flow-Based Policy, Consistent Updates | ✓ | ✓ | ✓ | ✓ | |
| Shared Data Store, Splendid Isolation | ✓ | | ✓ | ✓ | ✓ |
| Verificare, Machine-Verified SDN, VeriCon | | ✓ | ✓ | ✓ | |
| سطح امنیت سیستم SDN | | | | | |
| Debugger for SDN | ✓ | | | ✓ | |
| OFHIP, Secure-SDMN | | | | ✓ | |
| FRESCO | ✓ | ✓ | ✓ | ✓ | |

## ۳- راهکارهای امنیتی ارائه‌شده برای مسائل SDN

در جدول ۱ هفت دسته کلی از مسائل/حملات امنیتی SDN خلاصه شدند. جدول ۲ خلاصه‌ای از راهکارهای ارائه‌شده بر اساس این دسته‌بندی ارائه می‌کند. همان‌طور که مشخص است تاکنون برای نشت داده و تغییر داده‌ها راهکاری ارائه نشده است و بیشترین توجه مربوط به دسترسی‌های غیرمجاز و مسائل پیکربندی است. مسئله دیگری که مشخص است این است که راه‌حل‌ها تمامی لایه/واسط‌های SDN را تحت تأثیر قرار می‌دهند. بااین‌وجود لایه داده کمتر از همه تحت تأثیر قرارگرفته است. دلیل این امر هم این است که اکثر این راه‌حل‌ها سخت‌افزاری نبوده بلکه بر روی نرم‌افزار متمرکزشده‌اند. علاوه بر این راه‌حل‌ها بر اساس اینکه کدام بخش از معماری SDN را تحت تأثیر قرار می‌دهند طبقه‌بندی‌شده‌اند. در ادامه به‌صورت خلاصه تعدادی از این راه‌حل‌ها را مرور می‌کنیم. برای جزئیات بیشتر به [۱] مراجعه کنید.

### ۳-۱ دسترسی غیرمجاز

دو مسیر حمله مرتبط با دسترسی غیرمجاز در جدول ۱ مطرح شد: کنترلرهای غیرمجاز و برنامه‌های غیرمجاز؛ مسائلی که می‌تواند با توجه به امکان پیکربندی مجدد شبکه توسط عنصر غیرمجاز SDN، اثر ویرانگری روی شبکه داشته باشد.در [۳] با ارائه یک مدل کنترلی ترکیبی تلاش شده تا گلوگاه کنترلی شبکه، حذف و کارایی شبکه افزایش یابد. کنترل در شرایط معمول، مرکزی است اما وقتی که بار کاری شبکه زیاد باشد یک

ابزار شبکه وظیفه نصب قوانین جریان را روی سایر ابزارهای شبکه از طرف کنترلر به عهده می‌گیرد؛ در واقع، کنترل بین موجودیت‌های شبکه توزیع می‌شود. در این روش، سیستم به یک مدیر اعتماد مرکزی نیاز دارد و سربار قابل‌توجهی را در انتقال پیام‌ها و بررسی امضاها ایجاد می‌کند.

مشکل دادن امتیازات کامل OpenFlow به برنامه ها بدون هیچ حفاظتی در [۴] موردتوجه قرارگرفته است. PermOF مجموعه‌ای از اجازه‌های دسترسی و مکانیزم های ایزوله سازی را پیشنهاد می‌کند تا در ورودی API بتوان آن‌ها را پیاده‌سازی کرد. این راه‌حل به‌صورت مؤثری حداقل امتیازات را به برنامه‌ها داده و از شبکه در مقابل حملات لایه کنترل حفاظت می‌کند.در [۵] مفهوم اجازه‌های دسترسی سیستم بسط داده‌شده است. روش OperationCheckPoint برای کنترلر Floodlight طراحی و پیاده‌سازی شده که مجموعه‌ای از اجازه‌های دسترسی را که برنامه باید در مقداردهی اولیه با کنترلر مورد توافق قرار دهند مشخص و یک OperationCheckPoint را تعریف می‌کنند. قبل از این‌که به دستورهای برنامه مجوز دهی شود یک بررسی دسترسی‌ها را پیاده‌سازی می‌کند. یک سابقه (log) عملیات مجوز داده نشده برای ممیزی فعالیت بدخواهانه استفاده‌شده تا یک پروفایل برای حملات لایه کاربردی SDN ایجاد شود.

راه‌حل دیگر SE Floodlight است که درواقع یک افزونه برای کنترلر OpenFlow Floodlight است. افزونه SE-Floodlight یک هسته اعمال امنیت که درواقع بهبودیافته FortNox است و یک API مجوز داده‌شده دیجیتالی شمالی را معرفی می‌کند. لازم است تا یک مدیر، کلاس جاوای برنامه OpenFlow را از قبل امضا کرده باشد که قابلیت تائید دیجیتال آن توسط SEK و در زمان اجرا وجود دارد. بعدازاینکه برنامه‌ای امضا و تائید شد می‌تواند شبکه یا ترافیک شبکه را تغییر داده و یا مورد پرس‌وجو قرار دهد. AuthFlow یک مکانیزم کنترل دسترسی مبتنی بر اعتبارنامه‌های میزبان برای ممانعت از دسترسی غیرمجاز است که با یک کنترلر OpenFlow، یک احراز هویت کننده و سرور RADIUS پیاده‌سازی شده است [۷]. احراز هویت کننده پیام‌های پروتکل EAP را که بین میزبان و سرور احراز هویت RADIUS مبادله می‌شود دریافت کرده و یک پیام احراز هویت را برای کنترلر OpenFlow آماده می‌کند. کنترلر بر این اساس که پاسخ احراز هویت چه باشد ترافیک را منع کرده یا اجازه عبور می‌دهد. کنترل دسترسی با جفت کردن مجموعه‌ای از اعتبارنامه‌های میزبان با مجموعه‌ای از جریان‌ها امکان‌پذیر می‌گردد.

### ۳-۲ برنامه‌های بدخواه/در اختیار گرفته‌شده

برای جلوگیری از استقرار برنامه‌های بدخواه/در اختیار گرفته‌شده، حداقل کار این است که کنترلرها و برنامه باید یک ارتباط قابل‌اطمینان با یکدیگر برقرار کرده و قبل از ردوبدل کردن پیام‌های کنترلی از هویت هم اطمینان حاصل کنند.در [۸] یک هسته اعمال امنیت به‌عنوان راه‌کاری برای برنامه‌های بدخواه کنترلر معرفی شده است. FortNOX برای مجوز دهی به هر برنامه کاربردی OpenFlow از یک احراز هویت مبتنی بر نقش، استفاده می‌کند. FortNOX اگر قانون جریان جدید با یکی از قوانین جریان فعلی تداخل داشته باشد شناسایی و اگر یک نویسنده با اولویت بالاتر آن‌را ایجاد کرده باشد جایگزین قانون اولیه می کند، در غیراین‌صورت آن را نادیده می‌گیرد.

یک سیستم‌عامل شبکه قدرتمند و با کارایی بالا به‌نام ROSEMARY در [۹] معرفی شده که ایده اصلی آن، بهبود انعطاف‌پذیری سطح کنترل برای برنامه‌های دارای خطا و برنامه‌های بدخواه است. برای رسیدن به این هدف نویسندگان، یک سیستم‌عامل با معماری میکرو معرفی می‌کنند. هر برنامه OF درون یک نمونه از ROSEMARY اجرا می‌شود که به‌صورت مؤثری برنامه را حصارکشی(sandbox) کرده و لایه کنترل را از هرگونه آسیب‌پذیری یا عملکرد بدخواهانه برنامه حفاظت می‌کند. این راه‌حل برنامه‌های شبکه را از پایه محاسباتی قابل‌اطمینان سیستم عامل شبکه مجزا می‌کند، منابع استفاده‌شده توسط هرکدام از برنامه‌ها را کنترل و پایش کند، عملیات‌های برنامه‌ها مانند فراخوانی‌های سیستم ویژه را هم همین‌طور و یک فرایند امن راه‌اندازی مجدد NOS برای هرکدام از برنامه‌ها پیاده‌سازی می‌کند. این روش پیشرفته‌ترین رویکردی است که تاکنون برای حفاظت در برابر برنامه‌های بدخواه/در اختیار گرفته‌شده مورداستفاده قرارگرفته است.

### ۳-۳ منع سرویس

همچنان که در بخش اول توضیح داده شد جدا کردن لایه کنترل از لایه داده موجبات یک ضعف عمده را فراهم آورده که منجر به حملات منع سرویس روی کنترلر یا جدول جریان سوئیچ می‌گردد. برای حل مشکل گلوگاه ارتباطی بین لایه کنترل و لایه داده ، [۱۰] راه‌حلی به نام AVANT-GUARD را با محدود کردن تقاضاهای جریان ارسال‌شده به سطح کنترل با استفاده از ابزار مهاجرت ارتباط[۸] معرفی می‌کند. این راه‌حل با تمرکز روی حمله ارسال سیل‌آسای بسته‌های SYN در TCP از یک ابزار مهاجرت ارتباط برای حذف نشست‌های ناتمام TCP استفاده می‌کند. این کار احتمال اشباع یا انجام حمله DoS را صرفاً با ارسال درخواست‌های جریان به لایه کنترل جهت کامل کردن فرایند دست دهی سه‌گانه مرتفع می‌کند.

[8] Connection migration tool

شِمای اعتبارسنجی آدرس مبدأ معرفی‌شده در [۱۱] یک حفاظت پیشگیرانه در برابر جعل آدرس IP است که منجر به حملـه DoS می‌شـود. این شِما، مرز مجازی اعتبارسنجی آدرس منبع[9] نامیده شده است که برای حفاظت در برابـر جعـل آدرس هـم از قابلیـت تحلیـل ترافیـک SDN استفاده می‌کنند و هم از قابلیت به‌روزرسانی پویای قوانین. یک بسته ورودی به سوئیچ OpenFlow که با هیچ‌کدام از قوانین مطابقت پیدا نمی‌کند برای اعتبارسنجی آدرس مبدأ به کنترلر ارسال می‌شود؛ درصورت تشخیص جعل آدرس IP، قانونی در سـوئیچ نصـب می‌شـود تـا ترافیـک را از آن آدرس مبدأ متوقف کند. یک آداپتور قانون برای محدود کردن قوانین اختصاصی استفاده شده است، برای مثال یک قانون می‌تواند حـوزه جریـانی شامل چندین آدرس مبدأ را پوشش دهد. این کار توانایی برای حمله به سیل بستن جدول جریان سوئیچ را هم محدود می‌کند.

### ۳-۴ مسائل پیکربندی

چندین راه‌حل برای مسئله تضادهای سیاستی که سرمنشأ آن از برنامه‌های کاربردی متعدد است ارائه‌شـده اسـت. هرکـدام از ایـن راه‌حل‌هـا رویکردهای متفاوتی را در پیش‌گرفته‌اند که در ادامه تعدادی از آنها بررسی می‌شود.

روش VeriFlow صحت پایایی‌ها (invariants) را با قاپیدن قوانین جریان و به صورت بی‌درنگ قبل از رسیدن آن‌ها به شبکه مطالعه می‌کند [۱۲]. در VeriFlow شبکه به‌عنوان یک گراف مدل می‌شود تا حلقه‌ها ، مسیرهای غیرقابل اسـتفاده و غیـره در جـدول‌های مسـیریابی تشـخیص داده‌شود. هدف تشخیص پایایی‌های شبکه به‌صورت بی‌درنگ است و نتیجه حاصل‌شده نمایش‌دهنده کارایی شبکه برحسـب چنـد صـد میلی‌ثانیـه است. Frenetic یک API اختصاصی واسط شمالی است که جهت رفع تضادهای سیاست طراحی‌شده است و از آن برای برنامه‌نویسی مجموعـه‌ای از سوئیچ‌های شبکه که به‌وسیله یک کنترلر مرکزی کنترلر می‌شوند استفاده می‌شود [۱۳]. سیستم زمان اجرا (runtime) قبل از اینکه بـه کنترلـر دستور نصب قوانین جریان را در سوئیچ‌ها بدهد آن‌ها را به مجموعه‌ای از سیاست‌های نا هم پوشان تبدیل می‌کنـد. در [۱۴] اعمـال سیاسـت‌هـای مبتنی بر جریان به‌عنوان یک برنامه کاربردی NOX پیاده‌سازی شده است. این راه‌حل علاوه بر اینکه ایزولـه سـازی شـبکه را امکان‌پـذیر می‌کنـد یکپارچه‌سازی منابع احراز هویت خارجی را هم برای فراهم کردن کنترل دسترسی ممکن می‌کند.Verificare

یک ابزار اعتبارسنجی رسمی است که مدل‌سازی توصیفات و اعتبارسنجی درستی، همگرایی و خصیصه‌های سـیار شـبکه OpenFlow را در هم آمیخته و یک مدل عملیاتی دقیق را پیاده‌سازی و سپس آن را در ابزار اثبات Coq رسمی‌سازی کرده‌است [۱۵]. بر مبنـای همـین مـدل، یـک کامپایلر و سیستم زمان اجرای اعتبارسنجی شده توسعه داده شده و بـه‌عنوان اولـین کنترلـر SDN اعتبارسـنجی شـده به‌وسـیله ماشـین بـا نـام VeriCon پیاده‌سازی شده است. VeriCon می‌تواند امنیت برنامه‌های SDN را با بی‌نهایت حالت بررسی کند. مسائل پیکربندی همچون مسـئله راه‌اندازی اولیه سوئیچ در طول دوره حیات شبکه‌های SDN و اجزای آن وجود دارند و می‌توانند مستقیماً امنیت را تحت تأثیر قرار دهند. همچنان که شبکه در طول زمان با ابزارهای جدید، برنامه‌های جدید و کاربران جدید تغییر می‌کنـد بایـد سیاسـت‌های امنیتـی و روش‌هـای تکنیکـی هـم پیاده‌سازی و نگهداری شود تا از حفاظت شبکه و داده‌ها اطمینان حاصل شود. این کار احتمالاً در شبکه SDN نسبت به شبکه‌های سنتی چالشی‌تر است چون در مدل نوآورانه SDN به‌روزرسانی‌های شبکه بیشتر اتفاق می‌افتند [۱].

### ۳-۵ امنیت سطح سیستم SDN

در [۱۶] یک خطایاب ابتدایی شبکه پیشنهاد شده که به توسعه دهندگان SDN اجازه می دهد دنباله ای از رویدادها که منجـر بـه رخـدادن یک خطا شده بازسازی کرده و علت اصلی آن را کشف کنند. این خطایاب هم می تواند از فرایند خطایابی پشتیبانی کند و هـم بـرای مثـال بـرای مواردی مانند ممیزی شبکه که دنباله ای از رویداها می تواند اطلاعات مفیدی را از وضعیت شبکه به دست دهد مفید باشند.

در [۱۷]، نویسندگان راه حل OFHIP را ارائه کرده اند که در واقع ترکیبی از پروتکل تشخیص میزبـان[10] و OpenFlow اسـت و از کپسـوله سازی امنیتی دنباله (ESP) استفاده می کند. هدف OFHIP ارائه جابه جایی امنیتی با OpenFlow با اجتنـاب از مسـائلی اسـت کـه در معمـاری فعلی ممکن است با تغییر آدرس IP رخ دهد. برای مثال مختل شدن پردازش کردن جریان، قطع شدن یک نشست فعال TLS/SSL، مسائل احراز هویت دو طرفه. OFHIP این امکان را به سوئیچ های OpenFlow می دهد که به صورت امن آدرس IP خود را تغییر دهند که سیار بودن را درون شبکه ها و مابین آنها امکان پذیر می کند.

یکی از کارهای قابل توجه انجام شده FRESCO است که یک چارچوب توسعه برنامه های کاربردی امنیتی در ترکیب با FortNOX که یک هسته اعمال امنیت است ارائه می دهد[۱۸]. ایده اصلی FRESCO ایجاد قابلیت طراحی و توسعه سریع ماژول های خاص امنیتی اسـت کـه مـی توانند به عنوان برنامه های کاربردی OpenFlow مورد استفاده قرار گیرند. کتابخانه ای از ماژول های قابل استفاده مجدد برای تشخیص و کـاهش تهدیدات شبکه ای ایجاد شده است.

---

9) Virtual source Address Validation Edge(VAVE)

10 Host Identification Protocol (HIP)

## ۴- نتیجه گیری

شبکه‌های نرم‌افزار محور از زمان معرفی تا کنون به رده‌های بالای لیست موضوع‌های مورد علاقه محققین شبکه راه پیدا کرده است. تحقیقات گسترده‌ای در خصوص این شبکه‌ها صورت گرفته که به تعدادی از مهم‌ترین آن‌ها در این مقاله اشاره شد. شبکه نرم‌افزار محور از یک سو می‌تواند موجب تقویت امنیت شبکه شود و از دیگر سو منجر به تهدیدهای امنیتی جدید می‌شود که به هردو سوی این موضوع پرداخته شد. با این وجود به نظر می‌رسد شق اول این دوگانه بیشتر مورد توجه قرار گرفته و تهدیدهای امنیتی پیش‌رو بیشتر مورد بررسی قرار خواهد گرفت. پاسخ به این سوال که آیا شبکه‌های نرم‌افزار محور در حال حاضر امن هستند یا نه احتمالا خیر است. با به‌کارگیری تکنیک‌های امنیت شبکه فعلی و حل مسائل امنیتی شناخته‌شده SDN این شبکه‌ها احتمالا از شبکه‌های سنتی امن‌تر خواهند بود. اما رسیدن به این افق کارهای بسیاری باید انجام شود.

## ۵- منابع

[1] Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer, “A Survey of Security in Software Defined Networks”, **IEEE COMMUNICATION SURVEYS & TUTORIALS**, 2016, pp. 623-654
[2] “OpenFlow Switch Specification Version 1.4,” Open Networking, Foundation. [Online]. Available: https://www.opennetworking.org
[3] O. O. MM and K. OKAMURA, “Securing Distributed Control of Software Defined Networks,” **International Journal of Computer Science & Network Security**, vol. 13, no. 9, 2013
[4] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, “Towards a secure controller platform for openflow applications,” **in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking**. ACM, 2013, pp. 171–172.
[5] S. Scott-Hayward, C. Kane, and S. Sezer, “OperationCheckpoint: SDN Application Control,” in **22nd IEEE International Conference on Network Protocols (ICNP).** IEEE, 2014, pp. 618–623.
[6] OPENFLOWSEC.ORG, “Security-Enhanced Floodlight.” [Online]. Available: www.openflowsec.org
[7] D. M. F. Mattos, L. H. G. Ferraz, and O. C. M. B. Duarte, “AuthFlow: Authentication and Access Control Mechanism for Software Defined Networking.”, **Annals of Telecommunications**, December 2016, Volume 71, Issue 11–12, pp 607–615
[8] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, “A security enforcement kernel for OpenFlow networks,” in **Proceedings of the first workshop on Hot topics in software defined networks**. ACM, 2012, pp. 121–126.
[9] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, “Rosemary: A Robust, Secure, and HighPerformance Network Operating System,” in **Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security**. ACM, 2014, pp. 78–89.
[10] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, “AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks,” in **Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security**. ACM, 2013, pp. 413–424.
[11] G. Yao, J. Bi, and P. Xiao, “Source address validation solution with OpenFlow/NOX architecture,” in **19th IEEE International Conference on Network Protocols (ICNP)**. IEEE, 2011, pp. 7–12.
[12] A. Khurshid, W. Zhou, M. Caesar, and P. Godfrey, “VeriFlow: Verifying network-wide invariants in real time,” **ACM SIGCOMM Computer Communication Review,** vol. 42, no. 4, pp. 467–472, 2012.
[13] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, “Frenetic: A network programming language,” **ACM SIGPLAN Notices**, vol. 46, no. 9, pp. 279–291, 2011.
[14] T. Hinrichs, N. Gude, M. Casado, J. Mitchell, and S. Shenker, “Expressing and enforcing flow-based network security policies,” **University of Chicago**, Tech.Rep, 2008.
[15] A. Guha, M. Reitblatt, and N. Foster, “Machine-verified network controllers,” **in ACM SIGPLAN Notices**, vol. 48. ACM, 2013, pp. 483–494.
[16] N. Handigol, B. Heller, V. Jeyakumar, D. Mazieres, and N. McKeown, “Where is the debugger for my software-defined network?” in **Proceedings of the first workshop on Hot topics in software defined networks**. ACM, 2012, pp. 55–60.
[17] S. Namal, I. Ahmad, A. Gurtov, and M. Ylianttila, “Enabling Secure Mobility with OpenFlow,” in **IEEE Software Defined Networks for Future Networks and Services**. IEEE, 2013.
[18] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyso, “FRESCO: Modular composable security services for software-defined networks,” **in Proceedings of Network and Distributed Security Symposium**, 2013.

3rd. International Conference on September 2017
Information Technology, Computer and Telecommunications
سومین کنفرانس بین المللی شهریورماه ۱۳۹۶
فناوری اطلاعات، مهندسی کامپیوتر و مخابرات

# Classification of Security Issues and Proposed Solutions for Software Defined Networks (SDN)

Yaser Teymurzade

Department of IT, University of Shiraz, E-mail: teymurzade@gmail.com

**Abstract.** Traditional IP networks are very complex and hard to mange. Implementing network according to predefined policies and reconfigure it to respond to faults are cumbersome. Furthermore, control and data layers are bundled together. Software Defined Networks (SDN) are an emerging paradigm that separate data layer from control layer and bring progammability to networks. Although SDNs have many advantages at network programmability, enforcing polices and contoling networks, but they have different security challenges and problems. In this article, security issues that are directly related to SDN are categorized to seven different categories: unathurized access, data leakage, data modification, malicious/compormised application, denial of service, configuration issues and system level SDN security. Each of these issues and proposed solutions for them are reviewed and finally conclusions are made.

**Keywords**: software defined network, SDN, network security, security attacks, counter measurements