Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Kierunek Informatyka

Rozprawa doktorska

Implementacja algorytmów synchronizacji urządzeń sieciowych bazujących na paradygmacie programowalnych sieci komputerowych

Autor: mgr inż. Jarosław Homa
Promotor: prof. dr hab. inż. Aleksander Nawrat
Promotor pomocniczy: dr inż. Krzysztof Daniec

Gliwice, 2023

Serdecznie podziękowania,
za okazaną pomoc i wsparcie pragnę złożyć
Promotorowi
Profesorowi dr hab. inż. Aleksandrowi Nawratowi

SPIS TREŚCI

SŁOWNIK SKRÓTÓW I OZNACZEŃ

- sztuczna inteligencja – (ang. *AI - artificial intelligence* – sztuczna inteligencja) – zdolność maszyn do wykazywania ludzkich umiejętności, takich jak rozumowanie, uczenie się, planowanie i kreatywność
- sieci neuronowe – sieci neuropodobne, wzorowane na podstawowych mechanizmach działania ludzkiego mózgu systemy przetwarzania informacji, w których zrealizowano m.in. zdolność do uczenia się, jednoczesnego przetwarzania informacji oraz uogólniania wiedzy.
- uczenie maszynowe – obszar sztucznej inteligencji poświęcony algorytmom, które pozwalają na automatyczną korekcję własnych parametrów na drodze weryfikacji danych pomiarowych, uzyskiwanych podczas realizacji rozwiązania,
- uczenie głębokie – obszar sztucznej inteligencji poświęcony algorytmom uczenia sieci neuronowych, bazujących na rozbudowanych zbiorach danych uczących,
- przetwarzanie brzegowe (ang. *edge computing* – przetwarzanie brzegowe) - przetwarzanie brzegowe jest rozwiązaniem IT opracowanym z myślą o przenoszeniu danych i aplikacji bliżej użytkowników i „rzeczy", które z nich korzystają (*IoT*).
- Internet Rzeczy - (ang. *IoT Internet of Things* – internet rzeczy) - jest siecią połączonych urządzeń, które mogą komunikować się ze sobą i udostępniać dane użytkownikom przez Internet. Urządzenia *IoT* mogą łączyć się z Internetem, często są wyposażone w czujniki, umożliwiające im gromadzenie danych.
- chmura obliczeniowa – (ang. cloud computing) – chmura obliczeniowa - model przetwarzania danych oparty na użytkowaniu usług dostarczonych przez usługodawcę (wewnętrzny dział lub zewnętrzną organizację). Chmura to usługa oferowana przez dane oprogramowanie (oraz konieczną infrastrukturę).
- centrum danych – (ang. *data center* – DC- centrum danych) – jest fizycznym miejscem, w którym zainstalowane są serwery zbierające i przetwarzające dane.
- sieć 5G - 5G to skrót oznaczający piątą generację sieci komórkowej. W porównaniu z poprzednimi generacjami, sieć 5G ma dawać o wiele większą prędkość przekazywania danych i prawie niezauważalne opóźnienia oraz bardziej stabilne połączenia. 5G pozwoli podpiąć do Internetu ogromną liczbę dodatkowych urządzeń.
- programowalna sieć komputerowa – (ang. *SDN Software Defined Network* – programowalna sieć komputerowa) - koncepcja architektury sieci polegająca na wydzieleniu z urządzenia sieciowego inteligentnego, tj. zarządczo-sterującego komponentu i pozostawienie dla tego urządzenia wyłącznie zadań polegających na przesyłaniu danych w pakietach pomiędzy portami.

- Technologia SD-WAN - to akronim oznaczający sieci definiowane programowo (SDN) w sieci rozległej (WAN). SD-WAN upraszcza zarządzanie i obsługę sieci WAN poprzez oddzielenie sprzętu sieciowego od jego mechanizmu sterującego.
- lokalna sieć komputerowa – (ang. *LAN Local Area Network* – lokalna sieć komputerowa) sieć komputerowa łącząca komputery na określonym obszarze (budynek, szkoła, laboratorium, biuro). Sieć LAN może być wydzielona zarówno fizycznie, jak i logicznie w ramach innej sieci. Główne różnice LAN w porównaniu z WAN to wyższy wskaźnik transferu danych i mniejszy obszar geograficzny.
- rozległa sieć komputerowa – (ang. *WAN Wide Area Network* – rozległa sieć komputerowa) – sieć komputerowa znajdująca się na obszarze wykraczającym poza miasto, kraj lub kontynent. Internet jest obecnie największą istniejącą siecią komputerową.
- wirtualizacja funkcji sieciowych – (ang. *NFV network function virtualization* – wirtualizacja funkcji sieciowych) - to koncepcja architektury sieci, która wykorzystuje technologie wirtualizacji IT do wirtualizacji całych klas funkcji węzłów sieciowych w bloki konstrukcyjne, które mogą łączyć się lub łączyć ze sobą w celu tworzenia usług komunikacyjnych
- warstwa sterująca (ang. *control plane* – warstwa sterująca (element koncepcji SDN)
- warstwa transmisji danych (ang. *data plane* – warstwa transmisji danych (element koncepcji SDN)
- tabela przepływów - (ang. *FT flow table* – tabela przepływów wykorzystywana jest w momencie nadejścia do przełącznika nowego pakietu. Przełącznik przegląda wówczas właściwą tabelę w celu wyszukania najlepszej pojedynczej pozycji przepływu z uwzględnieniem cech odebranego pakietu oraz priorytetów poszczególnych pozycji w tabeli. Po wybraniu takiej pozycji przepływu, przełącznik wykonuje na pakiecie instrukcje określone w wybranej pozycji.
- Protokół OpenFlow - protokół komunikacyjny umożliwiający zdalny dostęp do płaszczyzny danych urządzenia sieciowego. Zamierzeniem twórców protokołu było wsparcie budowy programowalnych sieci komputerowych (SDN) dzięki standaryzacji oraz dopasowaniu do możliwości produkowanych urządzeń. Protokół OpenFlow rozdziela płaszczyznę danych (ang. *forwarding plane*) przełączników sieciowych i ruterów oraz sterowania (ang. *control plane*). Płaszczyzna danych realizowana jest przez urządzenia nazywane w tym kontekście przełącznikami OpenFlow, a płaszczyznę sterowania realizują kontrolery OpenFlow.
- Oprogramowanie kontrolera OpenDayLight - kontroler (sterownikiem) architektury SDN dla zróżnicowanych sieci złożonych z elementów od wielu dostawców.

- Interfejs programowania aplikacji - (ang. *API application programming interface*) - interfejs programistyczny aplikacji, interfejs programu aplikacyjnego – zbiór reguł ściśle opisujący, w jaki sposób programy lub podprogramy komunikują się ze sobą.
- REST – (ang. *representational state transfer*) – (zmiana stanu poprzez reprezentacje) – styl architektury oprogramowania wywiedziony z doświadczeń przy pisaniu specyfikacji protokołu HTTP dla systemów rozproszonych. REST wykorzystuje m.in. jednorodny interfejs, bezstanową komunikację, zasoby, reprezentacje, hipermedia, HATEOAS
- Protokół HTTP - (ang. *Hypertext Transfer Protocol*) - protokół http- używany w oknie przeglądarki, poprzedzający adres strony internetowej. Hypertext Transfer Protocol określa w jaki sposób treści są udostępniane, przekształcane i w jaki sposób ma je odczytać serwer.
- HTTPS - (ang. *Hypertext Transfer Protocol Secure*) – szyfrowana wersja protokołu HTTP. W przeciwieństwie do komunikacji niezaszyfrowanego tekstu w HTTP klient-serwer, HTTPS szyfrował dane przy pomocy protokołu SSL, natomiast obecnie używany jest do tego celu protokół TLS. Zapobiega to przechwytywaniu i zmienianiu przesyłanych danych.
- Model OSI – (ang. *ISO OSI RM – ISO Open Systems Interconnection Reference Model* – model odniesienia łączenia systemów otwartych lub OSI – standard zdefiniowany przez ISO oraz ITU-T opisujący strukturę komunikacji w sieci komputerowej. Model ISO OSI RM jest traktowany jako model odniesienia (wzorzec) dla większości rodzin protokołów komunikacyjnych. Podstawowym założeniem modelu jest podział systemów sieciowych na 7 warstw (ang. layers) współpracujących ze sobą w ściśle określony sposób.
- TIA – (ang. *Telecommunications Industry Association*) - Stowarzyszenie Przemysłu Telekomunikacyjnego, to organizacja normalizacyjna zrzeszająca przeszło 1100 firm, głównie amerykańskich, zajmujących się telekomunikacją, obróbką i przeszyłem danych. Powstałe w roku 1998 stowarzyszenie skupia się głównie na standardach dotyczących urządzeń telekomunikacyjnych a zwłaszcza na normach dotyczących przewodów i okablowania.
- Protokół IPv4 - (ang. *Internet Protocol version 4*) – czwarta wersja protokołu komunikacyjnego IP przeznaczonego dla Internetu. Identyfikacja hostów w IPv4 opiera się na adresach IP. Jest to 32 bitowa liczba zapisana binarnie, posiada reprezentację dziesiętną.
- Protokół IPv6 - (ang. *Internet Protocol version 6*) – protokół komunikacyjny, będący następcą protokołu IPv4, do którego opracowania przyczynił się w głównej mierze problem małej, kończącej się liczby adresów IPv4. Jest to 128 bitowa liczba zapisana szesnastkowo, nie posiada reprezentacji dziesiętnej.

- Protokół Ethernet - technika, w której zawarte są standardy wykorzystywane w budowie głównie lokalnych sieci komputerowych. Obejmuje ona specyfikację przewodów oraz przesyłanych nimi sygnałów. Ethernet opisuje również format ramek i protokoły z dwóch najniższych warstw Modelu OSI. Jego pierwotna specyfikacja została podana w standardzie IEEE 802.3.
- Protokół OSPF - (ang. *OSPF Open Shortest Path First*) to dynamiczny protokół routingu zdefiniowany przez Internet Engineering Task Force (IETF) i opisany w dokumencie RFC 1247. OSPF jest protokołem stanu łącza i należy do kategorii IGP. Routing z wykorzystaniem OSPF bazuje na systemach autonomicznych (AS).
- Protokół IGP – (ang. *IGP interior gateway protocol*) – protokół trasowania pakietów danych wewnątrz systemu autonomicznego.
- Protokół BGP - (ang. *BGP Border Gateway Protocol*) zewnętrzny protokół trasowania (routingu). BGP w wersji czwartej jest podstawą działania współczesnego Internetu. Istnieje wiele rozszerzeń protokołu BGP stosowanych przy implementacji protokołów MPLS VPN, IPv6 czy Multicast VPN.
- Protokół MPLS - (ang. *MPLS Multiprotocol Label Switching*) – technika stosowana przez routery, w której trasowanie pakietów zostało zastąpione przez tzw. przełączanie etykiet. MPLS, choć nie jest to konieczne, działa najlepiej na urządzeniach, które wymieniają etykiety sprzętowo.
- wirtualna sieć prywatna - VPN – (ang. *VPN virtual private network*) - (wirtualna sieć prywatna) to bezpieczne, szyfrowane połączenie między dwiema sieciami lub między użytkownikiem i siecią. Dzięki sieci VPN można anonimowo przeglądać Internet.
- lista kontroli dostępu - ACL – (ang. *ACL Access Control List*) – zestaw reguł filtrujących ruch w sieci. Każda z reguł zawiera zestaw warunków jakie muszą być spełnione aby pakiet przechodzący przez router został przepuszczony (bądź odrzucony)
- ściana ogniowa - Firewall - (ang. *Firewall*) – ściana ogniowa - to urządzenie (lub aplikacja) zabezpieczające sieć, które monitoruje przychodzący i wychodzący ruch sieciowy i decyduje o jego przepuszczeniu lub zablokowaniu w oparciu o zestaw określonych zasad. Zapory sieciowe są pierwszą linią obrony w obszarze bezpieczeństwa sieci od ponad 25 lat.
- Protokół TCP – (ang. *Transmission Control Protocol*) – połączeniowy, niezawodny, strumieniowy protokół komunikacyjny stosowany do przesyłania danych między procesami uruchomionymi na różnych maszynach, będący częścią szeroko wykorzystywanego obecnie stosu TCP/IP
- Protokół UDP - (ang. *User Datagram Protocol*) – protokół pakietów użytkownika – jeden z protokołów internetowych warstwy transportowej modelu OSI. Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP). Nie ma też mechanizmów kontroli przepływu i retransmisji.

- Protokół SNMP – (ang. *Simple Network Management Protocol*) - prosty protokół zarządzania siecią, który działa w warstwie aplikacji modelu ISO/OSI, umożliwiając wymianę informacji kontrolnych pomiędzy urządzeniami sieciowymi.
- Protokół ARP – (ang. *Address Resolution Protocol*) - protokół wynajdywania adresów - jest jednym z protokołów sieciowych należących do zestawu TCP/IP (nie związany bezpośrednio z transportem danych), używany do dynamicznego określania fizycznego adresu niskiego poziomu (MAC), który odpowiada adresowi wyższego poziomu (IP) dla danego hosta
- Adres MAC – (ang. *MAC address*) to fizyczny adres karty sieciowej. MAC adres jest 48-bitowy i zapisywany heksadecymalnie (szesnastkowo). Pierwsze 24 bity oznaczają producenta karty sieciowej, pozostałe 24 bity są unikatowym identyfikatorem danego egzemplarza karty.
- Atak DDOS – (ang. *Distributed Denial of Service*) - rozproszona odmowa dostępu. Atak DDoS ma na celu spowodowanie niedostępności serwera, usługi lub infrastruktury.
- Translacja NAT – (ang. *Network Address Translation*) translacja adresów sieciowych - większość systemów korzystających z NAT ma na celu umożliwienie dostępu wielu hostom w sieci prywatnej do Internetu przy wykorzystaniu pojedynczego publicznego adresu IP
- Sonda sieciowa IDS/IPS – (ang. *Intrusion Detection System, Intrusion Prevention System*) – systemy wykrywania i zapobiegania włamaniom) – urządzenia sieciowe zwiększające bezpieczeństwo sieci komputerowych przez wykrywanie (IDS) lub wykrywanie i blokowanie ataków (IPS) w czasie rzeczywistym.
- Zapora UTM - (ang. *unified threat management*) – wielofunkcyjne zapory sieciowe zintegrowane w postaci jednego urządzenia.
- Układ ASIC – (ang. *Application Specific Integrated Circuit*) - dedykowany układ scalony; jest to specyficzny typ elektronicznego układu scalonego, który został zaprojektowany i wyprodukowany do realizacji z góry określonego zadania.
- Konwerter GBIC – (ang. *Gigabit Interface Converter*) – konwerter gigabitowego interfejsu w sieciach komputerowych opartych o standard Ethernet, używany do przejścia z medium światłowodowego na elektryczne.
- Złącze RJ45 - (ang. *Registered Jack – type 45*) – standard dla złącza (interfejs) między okablowaniem klienta a okablowaniem firmy telekomunikacyjnej, najczęściej używany do podłączania niektórych typów modemów. Nie należy go mylić ze złączem 8P8C, które jest używane do tworzenia zakończeń kabli typu skrętka w sieciach standardu Ethernet. Wtyczka RJ-45 podobna jest do 8P8C, ale jest wyposażona w dodatkową wypustkę uniemożliwiającą włożenie jej do typowego gniazda 8P8C.

- PoE – (ang. *Power over Ethernet*) - technologia oparta na kilku standardach przesyłu energii elektrycznej za pomocą skrętki do urządzeń peryferyjnych będących elementami sieci Ethernet, takich jak: urządzenia komunikacji VoIP, adaptery sieci bezprzewodowej, punkty dostępu, kamery internetowe czy urządzania IoT.

1. WSTĘP

W obecnym świecie tematyka bezpieczeństwa sieci i systemów komputerowych, jest w zasadzie osobną dziedziną wiedzy wykraczającą bardzo daleko ze zbioru o nazwie Informatyka. Bezpieczeństwo sieci komputerowych i systemów informatycznych, stało się domeną wielu specjalistów z branży IT, administratorów sieci, administratorów systemowych, administratorów bazodanowych i aplikacyjnych. Podział ten można rozszerzać i zawężać, wyłaniając z tej grupy specjalistów np. w zakresie wirtualizacji systemów i sieci, instrumentalizacji (automatyzacji zarządzania systemami), aplikacji webowych, czy w końcu aplikacji mobilnych. W efekcie tych podziałów powstała elitarna grupa specjalistów IT, posiadająca większość ww. umiejętności, a jej profesja określna jest jako cyberbezpieczeństwo. Specjaliści w zakresie cyberbezpieczeństwa, są osobami dzięki pracy, których bez wątpienia nie moglibyśmy w dzisiejszych czasach korzystać z dobrodziejstwa kolokwialnie nazywanej sieci komputerowej - Internet. Portale i usługi takie jak Googel, Netflix, czy Facebook, są jednymi z lepiej zabezpieczonych systemów teleinformatycznych w dzisiejszym Internecie. Wynika to z faktu, iż nad bezpieczeństwem ww. systemów czuwa cała rzesza specjalistów z dziedziny cyberbezpieczeństwa, paradoksalnie znowu podzielonych na różne zespoły tj. specjaliści techniczni (ang. *DevOps*), pentesterzy (ang. *pentest*), czy administratorzy chmury obliczeniowej (ang. cloud admin). Efektywna praca tych wszystkich ludzi możliwa jest tylko i wyłącznie dzięki, implementacji w sieciach komputerowych (w tym operatorskich sieciach ang. ISP Internet Service Provider) nowoczesnych specjalistycznych systemów monitorowania, wykrywania, analizy, i w końcu mitygacji zagrożeń poprzez walidację adekwatnych polityk bezpieczeństwa. W grupie ponad 70 miliardów zdarzeń związanych z cyberbezpieczeństwem raportowanych przez raport IBM security task [2] każdego dnia, zdecydowana połowa to ataki wykorzystujące luki w systemach operacyjnych, a jedna trzecia to ataki wymierzone przeciwko konkretnym osobom i instytucjom. Obie te grupy zaklasyfikowane są do obszaru zagrożeń pod nazwą: ataki sieciowe. Na świecie średnio co 30 sekund przeprowadzany jest atak sieciowy typu DDoS. Według danych Kaspersky LAB, [1] Polska w II kwartale 2021 roku po raz pierwszy w historii znalazła się w pierwszej trójce państw na świecie pod względem przyjętych ataków sieciowych typy DDoS, wyprzedzają nas tylko USA i Chiny. Domena cyberbezpieczeństwa jest dziedziną produkcyjnie dojrzałą. Zawiera pełen wachlarz komercyjnych produktów i rozwiązań. Niemniej jednak, każdego dnia poziom cyberzagrożeń rośnie, jesteśmy zalewani różnego rodzaju atakami typu spyware, atakami typu SQL injection, czy cross-site scripting, lub szyfrującymi dane (ang. *ransomware*) tak bardzo ostatnio niebezpiecznymi. [3]

Cyberbezpieczeństwo jest zagadnieniem o krytycznej wadze, praktycznie w każdej gałęzi przemysłu i życia codziennego jesteśmy narażeni na zagrożenia. Zaczynając od bardzo wrażliwego sektora bankowego, przez sektor medyczny, energetyczny, militarny czy administracji państwowej do edukacji i rozrywki, wszędzie występują ciągłe próby naruszenia bezpieczeństwa sieciowego. Wymienione obszary bardziej lub mniej krytyczne, wymagają ciągłego monitorowania w szczególności wykrywania i analizy ataków sieciowych. Problem ten jest rozpatrywany i analizowany na poziomie krajowym czy europejskim. Unia europejska wydała dyrektywę o nazwie NIS, natomiast Polska nazwała ustawę o cyberbezpieczeństwie KSC – (ustawa o Krajowym Systemie Cyberbezpieczeństwa). Dokumenty te są aktualnie w fazie uzgodnień i aktualizacji, opisują one

wymagania stawiane ustawowo wrażliwym podmiotom gospodarczym (np. spółki miejskie – wodociągowe, lub firmy z sektora energetycznego) w zakresie zasad cyberbezpieczeństwa. Reasumując wymagania te nakazują podmiotom wrażliwym implementację rozwiązań zabezpieczających w postaci SOC (ang. *Security Operations Center*) – centrów operacji bezpieczeństwa, celem monitorowania i zabezpieczania ww. podmiotów. Centra bezpieczeństwa SOC, mają za zadanie monitorować, wykrywać, i obsługiwać zdarzenia w sieci z zakresu cyberbezpieczeństwa jak również ragować na nie za pomocą odpowiedniej linii wsparcia. Wszystkie działania na rzecz cyberbezpieczeństwa oprócz, odpowiednich ustaw, wykwalifikowanego personelu, specjalistycznych procedur, wymagają sprzętu do realizacji cyber-obrony. Właściwie całych systemów bezpieczeństwa opartych na rozwiązaniach takich jak: agregatorów zdarzeń SIEM, analizatorów SOAR, sond sieciowych typu IDS i IPS, zapór sieciowych FW NG i inne. Umiejętne ich skonfigurowanie czy strojenie lub lepiej dobranie/opracowanie odpowiednich metod, algorytmów często algorytmów sztucznej inteligencji (AI), uczenia maszynowego (ML) czy wręcz głębokiego uczenia maszynowego (DL), stanowią o powodzeniu działania i w konsekwencji wykryciu ataku sieciowego np. typu DDoS.

Intencją autora jest aby praca była innowacyjnym elementem cyberbezpieczeństwa, implementowanym w celu poprawy bezpieczeństwa sieciowego poprzez skuteczne wykrywanie najczęściej występującego rodzaju ataku sieciowego DDoS.

Rozprawa poświęcona jest implementacji algorytmów synchronizacji urządzeń sieciowych dla środowiska programowalnych sieci komputerowych SDN, służących do detekcji ataków sieciowych z rodziny DDoS.

W pracy zaprezentowane zostaną rozwiązania dotyczące zastosowania programowalnych sieci komputerowych SDN do budowy systemu wykrywania ataków sieciowych DDoS. System zostanie zaprojektowany w środowisku sieciowym SDN z centralnie sterującym kontrolerem i wspomagającymi go modułami decyzyjnymi wyposażonymi w algorytmy uczenia maszynowego ML i głębokiego uczenia maszynowego DL. Oprócz zastosowanych algorytmów sztucznej inteligencji AI, system będzie posiadał moduły z zaimplementowanymi algorytmami wykrywania ataków i mitygacji ich skutków. Wytycznymi do realizacji zadania były również kwestie dotyczące niezawodności i bezpieczeństwa samego sytemu. System zostanie skonstruowany z uwzględnieniem redundancji, skalowalności, i eliminacji pojedynczych pkt. awarii. Z przeprowadzonej analizy literatury w połączeniu z wykonanymi badaniami sformułowano następującą tezę i cel rozprawy:

Tezą pracy jest: *Możliwe jest opracowanie systemu mitygacji ataków sieciowych, opartego o rozwiązanie programowalnych sieci komputerowych SDN, wykorzystującego nowatorskie algorytmy sterowania i synchronizacji urządzeń, w tym analizę uzyskaną za pomocą algorytmów sztucznej inteligencji AI.*

Celem rozprawy jest zaprezentowanie aktualnego stanu wiedzy poprzez przegląd literaturowy w zakresie metod wykrywania ataków DDoS z wykorzystaniem sieci komputerowych SDN. Przedstawienie aktualnych badań dot. systemów sieci komputerowych SDN. Analiza mitygacji ataków sieciowych z uwzględnieniem ataków DDoS. Przegląd algorytmów wykrywania ataków sieciowych, algorytmów opartych o standardowe metody, algorytmów sztucznej inteligencji AI, uczenia maszynowego ML i głębokiego uczenia maszynowego DL. Propozycja rozwiązania problemu, poprzez opracowanie systemu informatycznego mitygacji ataków DDoS opartego o paradygmat sieci SDN

z modułami wykonawczymi i zaimplementowanymi w nich algorytmami monitorowania, wykrywania i walidacji bezpieczeństwa. Projektowany system bezpieczeństwa, został w pełni skonfigurowany i uruchomiony. Zostaną zaprezentowane badania w środowisku laboratoryjnym i produkcyjnym, jak również analiza wyników pomiarowych. Rozprawa kończy się podsumowaniem realizacji całości zadania.

Zastosowane technologie:

- Środowisko sieci komputerowych SDN z centralnym wyodrębnionym kontrolerem infrastruktury
- Urządzenia sieciowe wykonawcze: IDS/IPS, przełączniki, routery, firewall implementowane w technologii sieci SDN
- Moduły wykonawcze sytemu mitygacji, analizujące, z zaimplementowanymi algorytmami – w postaci rozszerzeń kontrolera sieci SDN
- Dodatkowe serwery akceleracji sprzętowej na potrzebę implementacji wydajnego środowiska dla algorytmów głębokiego uczenia maszynowego DL
- Środowisko wirtualne testowe dla badań laboratoryjnych, wydajnościowych, obciążeniowych
- Generatory ataków sieciowych

Zakres pracy obejmuje:

- Przegląd aktualnego stanu wiedzy w kierunku sieci SDN i metod mitygacji aktów sieciowych DDoS
- Opracowanie autorskiego systemu mitygacji ataków sieciowych DDoS pracującego w oparciu koncepcje sieci SDN (środowisko oparte i zasymulowane w Mininet z OpenFlow i kontrolerem SDN)
- Opracowanie modułów funkcjonalnych systemu do realizacji założonych celów
- Przedstawienie implementowanych algorytmów w tym z wykorzystaniem ML i DL
- Implementacje sytemu w warunkach laboratoryjnych i produkcyjnych
- Wykonanie testów wydajnościowych, analiza i przedstawienie wyników

Zakres pracy nie obejmuje:

- Opisu klasycznych sieci komputerowych, protokołu TCP/IP
- Szczegółowego przedstawienia mechanizmów AI
- Analizy komercyjnych systemów bezpieczeństwa sieciowego
- Implementacji rozwiązań nieustandaryzowanych, aktualnie promowanych przez organizację ONF (Open Networking Foundation), która wyznacza trendy dla rozwoju sieci SDN[20]
- Sensytywnych szczegółów wdrożenia, które stanowią tajemnicę handlową przedsiębiorstwa

2. PRZEGLĄD LITERATURY

W ostatnich latach liczba urządzeń łączących się z Internetem rosła równolegle ze wzrostem liczby użytkowników Internetu. Wyrafinowane cyberataki, zakłócają sieci komputerowe na całym świecie. Coraz więcej słyszymy o działaniach zorganizowanych grup hackerskich, które na celownik biorą sobie zarówno małe przedsiębiorstwa jak i duże firmy korporacyjne, często ofiarami ataków sieciowych są również organy administracji rządowej i samorządowej. Głównymi zagrożeniami, są rozproszone ataki typu "odmowa usługi" (DDoS) oraz masowo rozprzestrzeniające się robaki internetowe, dla których nośnikami bardzo często jest poczta elektroniczna email. W związku z tym sieci komputerowe wielkoskalowe [4] stoją przed wieloma wyzwaniami w odniesieniu do ich bezpieczeństwa. W szczególności kontroli przesyłanych ogromnych ilości pakietów, bez negatywnego wpływu na szybkość działania systemów, zapobiegania włamaniom do sieci oraz skuteczne wykrywanie i łagodzenie zagrożeń. Uwzględniając ogromny przyrost ilości ataków w sieciach komputerowych, pojawiło się ogromne zapotrzebowanie na wydajne, skalowalne rozwiązania do wykrywania złośliwego oprogramowania jak również dedykowane systemy reagowania i zapobiegania atakom sieciowym.

Ataki typu Distributed Denial of Service (DDoS) są szeroko badane od ponad dwóch dekad, ale nadal obserwuje się ich gwałtowny wzrost. W szczególności ataki oparte na zalewie danych (flooding), protokołów tj. UDP, TCP SYN i ICMP są dominujące. Celem tych ataków jest wyczerpanie zasobów obliczeniowych atakowanego hosta, w tym jego procesora, pamięci i pasma sieciowego poprzez wysyłkę przytłaczającej liczby fałszywych pakietów. Na przykład w protokole TCP SYN flood [5] atakujący przytłacza ofiarę pakietami typu SYN, wyczerpując tabelę połączeń, podczas gdy ataki UDP i ICMP zużywają głównie pasmo sieci, wysyłając sfałszowane pakiety. W ostatnich latach odnotowano również wzrost liczby wielowektorowych ataków DDoS [6]. Znamienitym przykładem jest atak polegający na dużym zalewaniu pakietów UDP w połączeniu z powolnym zalewem żądań HTTP GET [7]. Wprowadza on system ofiary w błąd, system próbuje poradzić sobie z pozornie anomalnym zalewem pakietów UDP, w tym czasie zalew żądań HTTP może powoli wyczerpywać zasoby obliczeniowe serwera HTTP.

W klasycznych sieciach komputerowych, aby okiełznać problem ataków DDoS, podjęto szereg prób, utworzenia systemów zarówno przez środowiska naukowe (badawcze), jak i B+R (R&D) z sektora przemysłowego [8], [9]. Jednak niewiele z opracowanych technik mitygacji ataków DDoS zostało przeniesionych do powszechnego użycia, głównie ze względu na ich złożoność wdrożenia, jak również wygórowane koszty operacyjne. Jednym z głównych powodów był fakt, iż zwykle wymagają one utrzymywania dużych tabel stanów połączeń sieciowych na routerach lub przełącznikach. Powoduje to dodatkowe obciążenia pamięci operacyjnej i procesora. Ponadto niektóre techniki, takie jak znakowanie pakietów [10], wymagają ciągłego monitorowania i zbierania ogromnej ilości pakietów, wiąże się to z dodatkowym obciążeniem w procesie przetwarzania urządzenia sieciowego. Istotne jest to, że większość rozwiązań opiera się na implementacji dodatkowych modułów lub urządzeń, zwiększając złożoność wdrożenia systemu. Pomimo wysiłków na rzecz zaprojektowania systemu autonomicznej odpowiedzi na ataki DDoS [12], [13], ich skalowalność jest wątpliwa w obliczu wdrożenia na dużą skalę, głównie ze względu na intensywną współpracę i komunikację między różnymi

modułami detekcji. Ponadto, chociaż propozycje wykrywania anomalii zaproponowane w [14] mogą zachować niektóre właściwości autonomiczne, ich użyteczność i skuteczność w łagodzeniu DDoS w konwencjonalnych sieciach nie zostały zweryfikowane na dużą skalę.

Nadrzędną kwestią jest, aby łagodzenie skutków ataków DDoS było maksymalnie zautomatyzowane, dobrze skalowalne oraz było możliwie mało obciążające dla systemów sieciowych, a przy tym łatwe w zarządzaniu. Należy podkreślić, że wiele koncepcji projektowych nadal pozostaje nierozwiązanych. Pojawienie się i szybki rozwój koncepcji sieci komputerowych programowalnych (SDN - Software-Defined Networks) oferuje nam możliwości na ponowne przebadanie i ulepszenie projektów anty-DDoS skonstruowanych dla implementacji w klasycznych sieciach komputerowych. Kluczowy jest paradygmat SDN, polegający na oddzieleniu płaszczyzny sterowania siecią i płaszczyzny danych, a także programowalności samego sterownika [15]. Ponieważ kontroler sieci SDN pozwala uzyskać globalny widok stanów sieci jak również osiągnąć scentralizowaną analizę śledczą sieci [16]. Wnioskować można, iż główne mitygacyje ataków DDoS mogą być podobnie implementowane w kontrolerach SDN, jak jest to zaproponowane w [17]. Fakt ten powoduje, że nie będzie wymagana ciągła interwencja człowieka w zakresie zarządzania i utrzymywania schematów mitygacji ataków DDoS. Moduły systemu mitygacji można wyodrębnić i zintegrować w warstwie aplikacji SDN. Instalacja wymaganych wcześniej urządzeń nie jest już niezbędną koniecznością. Narzuty obliczeniowe na routerach i przełącznikach można znacznie zmniejszyć, ponieważ można migrować dużą liczbę stanów połączeń w tabele przepływu (FT) obsługiwane przez kontrolery SDN. Istotną kwestią jest iż, niektóre moduły systemu zabezpieczeń lub interfejsy API mogą być wdrażane w samym kontrolerze SDN, umożliwiając między innymi dostawcą usług internetowych (ISP) zapewnienie łagodzenia skutków ataków DoS w trybie „na żądanie” przez swoich klientów[19].

System wykrywania włamań (IDS) jest jedną z najważniejszych części bezpieczeństwa architektury sieci. W oparciu o różnicę w jaki sposób dane są przetwarzane, IDSy można podzielić na oparte na bazie sygnatur, bądź na wykrywaniu anomalii. Sygnatury są wzorcami wcześniej wykrytych włamań i stanowią bazę wykrywania nieprawidłowości w ruchu sieciowym. Ta metoda opiera się na monitorowaniu i dopasowywaniu zaobserwowanego ruchu sieciowego do bazy danych w celu wykrycia włamania. Znane ataki można wykryć naprawdę dobrze w ten sposób z niskim wskaźnikiem fałszywych alarmów. Jednak ta metoda nie pozwala na wykrycie ataków typu zero-day, które są nowe i nieznane. Metoda wykrywania anomalii tworzy normalny model wyjściowy zachowania na podstawie obserwowanego ruchu sieciowego, a następnie próbuje wykryć każde zachowanie, które odbiega od linii bazowej, co pozwala wykrywać nieznane wcześniej ataki typu zero-day.

Ze względu na szeroką gamę rodzajów wdrożeń sieci SDN, bezpieczeństwo SDN jest poważnym problemem i zostało szeroko zbadane w [28], [29]. Zaproponowano kilka podejść do uczenia maszynowego (ML) w celu zabezpieczenia sieci SDN. Pomimo wysokiej dokładności i wydajności uzyskanej w kilku dziedzinach, algorytmy ML stosowane w wykrywaniu włamań mają zwykle pewne ograniczenia, jak wspomniano w [30]. Należą do nich trudności w określeniu dyskryminatora, dostępność oznakowanych zbiorów danych do klasyfikacji i oceny, wysokie koszty błędów i różnorodność ruchu sieciowego. W ostatnich latach mechanizm ML był używany przez wielu badaczy, do systemów wykrywania włamań (NIDS) w środowisku sieci SDN. Jednak techniki te zwykle prowadzą do wysokiego wskaźnika fałszywych alarmów. Zasadnym wydaje się rozszerzenie dotychczasowych podejść o głębokie uczenie się (DL) w kontekście sieci SDN.

2.1. SIECI KOMPUTEROWE SDN

2.1.1. KONTEKST

Terminu SDN (ang. Software-Defined Networks) użył jako pierwszy w swojej pracy doktorskiej Martin Casado w 2007 roku na Stanford University. Powszechnie jest on uznawany za twórcę koncepcji sieci programowalnych SDN. Sieci definiowane programowo od początku stały się obiecującą technologią i szybko przyciągnęły uwagę środowiska naukowego, sugerując iż jest to sieć przyszłej generacji. Zastosowanie mechanizmów wirtualizacji, uniwersalność SDN, wynikająca z możliwość programowalności obszarów sieciowych, która wcześniej była zarezerwowana dla producentów sprzętu sieciowego, zaowocowały entuzjastycznym ich przyjęciem zarówno w rozwiązaniach akademickich, jak i przemysłowych.

Zanim przedstawię od podstaw koncepcję działania sieci SDN, chciałbym zdefiniować, mapę drogową tej technologii, odpowiedzieć na pytanie kto ma bezpośredni wpływ na rozwój tej technologii i w jakim aktualnie miejscu się znajdujemy.

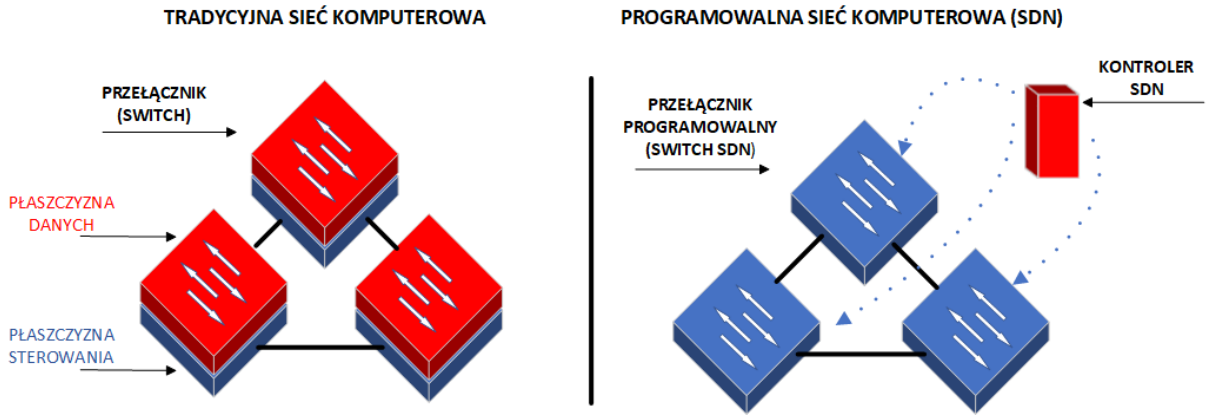
Sieci komputerowe SDN w zasadzie opierają się na koncepcji protokołu OpenFlow[21]. Prace nad rozwojem sieci SDN a w tym głownie nad protokołem OpenFlow, Martin Casado powierzył organizacji ONF (ang. Open Networking Foundation), która wyznacza kierunki dla rozwoju współczesnych sieci SDN. Sam protokół OpenFlow był rozwijany przez ONF od samego początku czyli od 2011 roku. Pierwszy standard interfejsu do separacji płaszczyzny kontrolnej sieci od płaszczyzny danych został opublikowany w 2012 roku pod oficjalną nawą OpenFlow i jest rozwijany do dnia dzisiejszego, jednak nie wprost. W repozytoriach GitHub-a można odnaleźć ostatnie wersje paczki instalacyjnej OpenFlow o nr wersji 1.5.0 datowanej na lipiec 2016 roku. Wynika to z faktu, iż w 2014 roku ONF zaprezentowała kontroler SDN o nazwie ONOS (ang. Open Network Operating System) i przedstawiła go jako wiodący kontroler SDN „Nowej Generacji" dla rozwiązań SDN/NFV (ang. Network Function Virtualization). Oczywiście ONOS zawiera w sobie podstawowy protokół komunikacyjny OpenFlow. W 2017 roku ONF zaprezentowało rozwiązanie o nazwie CORD przedstawiając je jako platforma dla rozwiązania wirtualizacji sieci (SDN/NFV), Cloud – rozwiązań chmurowych, a przede wszystkim dla rozwiązań EDGE – brzegowe rozwiązanie data center ukierunkowane na wsparcie sieci piątej generacji (5G). CORD to konglomerat wszystkich dotychczasowych projektów dla sieci SDN w tym wspomnianego OpenFlow. W 2019 roku ONF połączyło się z ON LAB i „uporządkowało" kwestie rozwijanych projektów z podziałem na obszary mobilne (sieci 5G) i infrastrukturalne. Aktualnie na liście rozwijanych projektów znajdziemy nazwy takie jak między innymi: SD-RAN, SD-CORE, STRATUM, P4, NG-SDN, CORD, XOS, czy MININET. Nie znajdziemy tam jednak nazwy: OpenFlow, co wcale nie znaczy, że nie jest rozwijany – występuje jako moduł w ramach ww. projektów [20].

Badania w doktoracie zostały wykona na dedykowanym obszarze testów laboratoryjnych w środowisku MININET z zainstalowanymi, kontrolerami sieciowymi SDN w wersji POX, Ryu

i kontrolerem OpenDaylight, protokołem OpenFlow w wersjach od 1.0.0 do 1.5.0. We wdrożeniu produkcyjnym został zastosowany kontroler CORD z systemem ONOS i wszystkimi niezbędnymi modułami.

2.1.2. KONCEPCJA SDN

Główną cechą na której opiera się zaproponowana koncepcja sieci SDN jest rozdzielenie płaszczyzn sterowania (ang. *control plane*) i płaszczyzny danych (ang. *data plane*) [20][23].Koncepcje przedstawia rysunek 1. Rolę zarządzania przejmuje w całej sieci Kontroler, który jest odpowiedzialny za działanie wszystkich urządzeń sieciowych SDN w płaszczyźnie danych. Urządzenia sieci SDN w tym przełączniki i routery nie podejmują żadnej decyzji o przekazaniu danych. W sieciach SDN proces modyfikacji reguł i polityk odbywa się w kontrolerze, a następnie dana reguła zostaje wysłana do urządzenia sieciowego w celu określenia sposobu przekazywania. Dzięki temu obniżone są koszty modyfikacji konfiguracji w stosunku do tradycyjnej sieci, w której należy zmodyfikować konfigurację każdego urządzenia sieciowego przekazującego pakiety, aby wprowadzić dowolną zmianę.



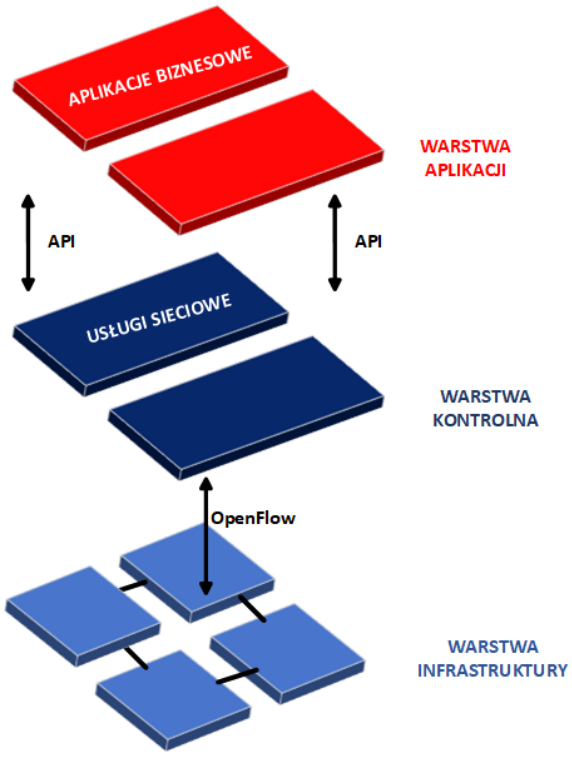
Rysunek 1 Rozdzielenie płaszczyzn sterowania i płaszczyzny danych. Źródło [23]

Architektura SDN oparta jest na trzech warstwach[20], odpowiednio: warstwa przekazywania danych, warstwa kontrolna i warstwę aplikacji. Architekturę trzywarstwową SDN przedstawia rysunek 2. Warstwa przekazywania danych (infrastruktury) zawiera urządzenia wykonawcze do przekazywania danych (przełączniki i routery), przekazują one nadchodzące pakiety zgodnie z tabelą przepływu (FT ang. *flow table*). Warstwa kontrolna (ang. *control plane*) zawiera złożoną logikę sterowania, odpowiadającą za nadzór i zarządzanie całością operacji sieciowych. Warstwa aplikacji zawiera zestaw aplikacji służącym zróżnicowanym celom, np. funkcja balansu obciążenia, monitorowanie i akwizycja ruchu czy wirtualizacja sieci.

SDN posiada wiele korzystnych funkcji przydatnych w aspekcie wykrywania ataków DDoS. Po pierwsze, scentralizowany widok SDN dostarcza kontrolerowi pełnych informacji o sieci. Umożliwia to natychmiastowy wgląd we wszystkie anomalne sytuacje występujące w sieci, są one na bieżąco rejestrowane przez kontroler. Kluczowa funkcjonalność to programowalność, która umożliwia

implementacje modułów dla różnych funkcji, np. aplikacji monitorującej, sterującej przeszyłem pakietów czy walidowania polityk bezpieczeństwa.

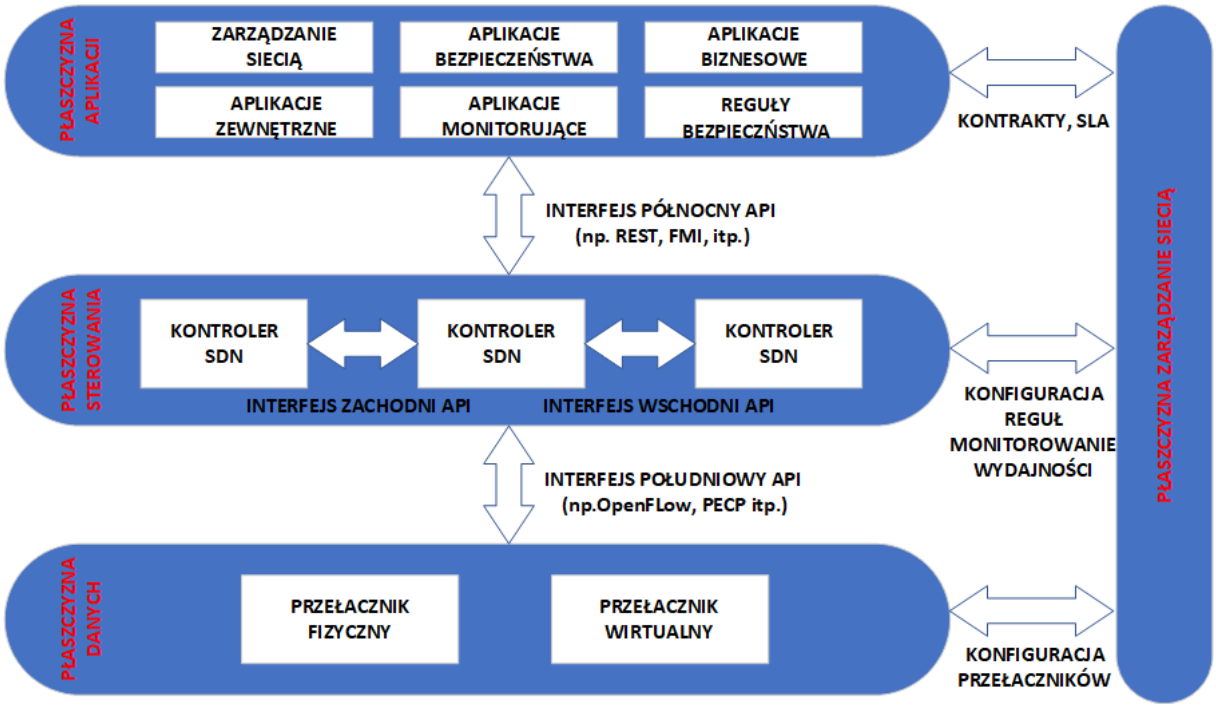
Podstawowymi zaletami sieci SDN wynikającymi z jej architektury są zdecydowanie: bezpośrednio programowalne sterowanie siecią (możliwe jest to dzięki wydzieleniu funkcji przekazywania pakietów). „Zwinność" (ang. *agile*) czyli dynamiczne dostosowywanie przepływu w całej sieci do zmieniających się uwarunkowań. Centralne zarządzanie w programowym kontrolerze SDN utrzymujący globalny widok sieci (stanowi on dla aplikacji i reguł w sieci pojedynczy logiczny przełącznik). Programowalna konfiguracja umożliwia bez zwłoczne zarządzanie, poprzez wdrażanie konfiguracji, z polityką bezpieczeństwa a w konsekwencji płynne optymalizowanie zasobów sieciowych. Mamy możliwość użycia automatyzacji programów, zdefiniowanych na otwartych standardach, nade wszystko dysponujemy pełną niezależnością od producentów sprzętu sieciowego. Podstawową ideą architektury SDN jest oddzielenie płaszczyzny sterowania i przekazywania danych sieci, co umożliwia zastosowanie funkcji sterowania siecią bezpośrednio z pozycji kontrolera SDN. W ten sposób umożliwia wyodrębnienie podstawowej infrastruktury dla aplikacji i różnych usług sieciowych. Przedstawione cechy sprawiają, iż rozwiązanie sieci SDN jest odpowiednią technologią dla środowisk aplikacji wysokowydajnych[21].



Rysunek 2. Trzywarstwowa architektura SDN. Źródło[20]

2.1.3. ARCHITEKTURA REFERENCYJNA SDN.

Architekturę referencyjną SDN [22] i obszary używane do komunikacji między płaszczyzną sterowania a innymi płaszczyznami SDN pokazano na rysunku 3.



Rysunek 3. Referencyjna architektura SDN. Źródło: [22]

Zaprezentowana architektura jest podzielona na cztery płaszczyzny funkcjonalne:

• **Płaszczyzna danych**. (ang. *data plane*) Ta płaszczyzna jest również znana jako płaszczyzna infrastruktury (ang. *infrastructure layer*). Składa się ona z niezbędnych zasobów do przetwarzania ruchu przychodzącego, posiada niezbędne funkcjonalności, w zakresie wirtualizacji, bezpieczeństwa, dostępność, QoS (ang. *Quality of Service*) itp. Płaszczyzna danych składa się głównie z elementów przekazywania (ang. *forwarding*), w tym przełączników fizycznych, przełączników wirtualnych, routerów i punktów dostępu. Za pośrednictwem otwartego interfejsu i wykonuje przełączanie i przekazywanie pakietów. Płaszczyzna danych wykonuje przekazywanie zgodnie z określonymi regułami w kontrolerze. Płaszczyzna danych nie może podejmować samodzielnie decyzji w zakresie przekazywania ruchu. Istnieje możliwość konfiguracji do podejmowania autonomicznych decyzji dotyczących przekazywania w przypadku awarii sieci, utraty komunikacji z kontrolerem . Znamienitym przykładem urządzenia do przekazywania w sieci SDN jest przełącznik OpenFlow. Jego działanie jest uproszczone, ponieważ zasady przekazywania są definiowane przez kontroler, a nie przez sprzęt lub oprogramowanie przełącznika. [21][22]

• **Płaszczyzna sterowania**. (ang. *controle plane*) Płaszczyzna sterowania jest oddzielną logicznie, scentralizowaną płaszczyzną. Sterownik sieci SDN jest nadrzędny, implementuje on różne

funkcjonalności płaszczyzny sterowania. Kontroler jest odpowiedzialny za zarządzanie całą siecią, pracuje pod kontrolą sieciowego sytemu operacyjnego NOS (ang. Network Operating System). Kontroluje również przetwarzanie przepływu na płaszczyźnie danych, instalując reguły przekazywania przepływu w elementach ścieżki danych. Mamy trzy interfejsy, które umożliwiające kontrolerowi komunikowanie się z innymi płaszczyznami SDN: Southbound API (południowy), Northbound API (północny) i East/Westbound API (wschodni/zachodni). Południowy Interfejs API służy kontrolerowi do komunikacji z elementami płaszczyzny danych. Standardowym protokołem, który umożliwia komunikację między logicznie scentralizowanym kontrolerem a elementem przekazywania pakietów jest OpenFlow. Za każdym razem, gdy przepływ dociera do przełącznika OpenFlow, sprawdzany jest wpis przepływu przychodzącego w jego tabeli przepływów. Jeśli zostanie znaleziony pasujący wpis, pakiet zostanie przekazany zgodnie z instrukcjami zawartymi w regule. Jeśli jednak nie zostanie znaleziony pasujący wpis, nagłówek przepływu zostanie wyodrębniony i przekazany do kontrolera. W związku z tym kontroler instaluje nową regułę przepływu w tabeli przepływu przełącznika, instruując w nim przekierowanie przepływu do określonego portu, albo odrzucenie (drop) pakietów z tego portu źródłowego. Północny interfejs SDN jest interfejsem pomiędzy aplikacjami SDN a kontrolerem, który zapewnia głównie abstrakcyjne widoki sieci i umożliwia programowanie i zarządzanie siecią. Interfejs wschód/zachód umożliwia komunikację pomiędzy grupami sterowników.[22]

• **Płaszczyzna aplikacji**. (ang. *application plane*) Zapewnia aplikacje użytkownika końcowego do interakcji z urządzeniami sieciowymi za pośrednictwem kontrolera i umożliwia sterowanie urządzeń sieciowych. Aplikacje SDN to programy, za pomocą których możemy w sposób programowy wpływać na zachowanie sieci z pozycji kontrolera SDN. Istotnie, mogą również wykorzystywać abstrakcyjny widok sieci, do wewnętrznych celów decyzyjnych.[22]

• **Płaszczyzna zarządzania siecią**. (ang. *network management*) Płaszczyzna zarządzania siecią wykonuje zadania statyczne, które mogą być obsługiwane poza płaszczyznami aplikacji, sterowania i danych. Na przykład przypisywanie zasobów do klientów, konfiguracja sprzętu fizycznego, koordynowanie osiągalności poświadczeń między logicznymi i fizycznymi urządzeniami. [22]

2.1.4. PROTOKÓŁ OPENFLOW

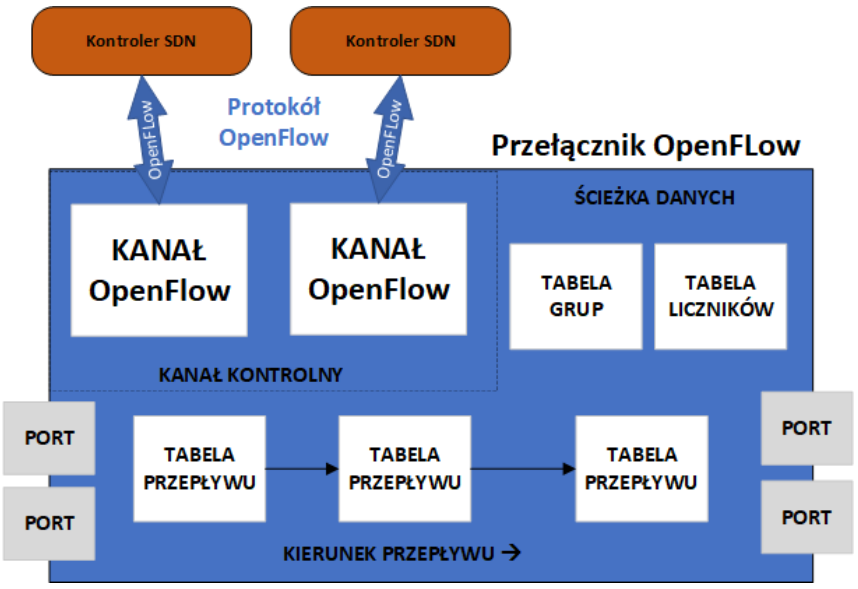
OpenFlow to koncepcja na której opierają się sieci SDN. Jest protokołem, który określa, w jaki sposób logicznie scentralizowany kontroler SDN powinien komunikować się z urządzeniami (płaszczyzny danych) przekazującymi w środowisku sieci SDN [24]. Protokół OpenFlow dostarcza do sterownika sieci danych, dzięki którym mamy globalny widok sieci komputerowej, co ułatwia identyfikację słabych punków sieci, zarządzanie całą siecią i wdrażanie polityk wysokiego poziomu. Programistom sieci SDN, umożliwia sprostanie nowym wyzwaniom poprzez zdefiniowanie dodatkowych elementów protokołu. Protokół OpenFlow zmniejsza wysiłek administratorów sieci związany z wdrażaniem innowacyjnych protokołów routingu i przełączania w sieci. Początkowa koncepcja OpenFlow, głosiła iż, protokół może całkowicie zastąpić funkcjonalność protokołów warstwy L2 i L3 w komercyjnych przełącznikach i routerach. Kluczowe komponenty modelu OpenFlow to

przełącznik OpenFlow (vSwitch), kontroler OpenFlow (SDN Kontroler) i protokół OpenFlow, stały się częścią powszechnej definicji sieci SDN[21]. Jest stosowany w aplikacjach, takich jak sieci o wysokim poziomie bezpieczeństwa, w sieciach o dużej mobilność maszyn wirtualnych w sieci telefonii komórkowej nowej generacji (5G) opartej na protokole IP.

OpenFlow jest tak naprawdę zestawem protokołów i API. Jego główną funkcją jest komunikacja pomiędzy płaszczyzną danych w której znajdują się urządzenia przekazujące tj. przełączniki OpenFlow, a płaszczyzną sterowania czyli miejsce gdzie znajduje się zgodnie z koncepcją sieci SDN, kontroler – sterownik sieci. Protokół operuje na portach, przepływach i akcjach. Porty to często fizyczne porty przełączników ale czasem też abstrakcyjne pseudo-porty, lub porty przełącznika wirtualnego. Przepływy operują na tabelach przepływów, tabelach grup, wytyczne przekazywane są jak pozycje przepływu. Akcje to wynik przetwarzania pakietów przez przełącznik, typową akcją jest przekazanie pakietu na określony port lub do kontrolera, ewentualnie odrzucenie pakietu. OpenFlow potrafi również odczytywać wartości liczników, tworzyć mierniki i kolejki, definiować grupy służce do określonych faz przetwarzania pakietów. Prowadzi komunikacje pomiędzy przełącznikiem a kontrolerem i raportuje o różnych zdarzeniach, wysyła również pakiety kontrolne[15]. Posiad funkcje replikacji, jako protokół przewodowy wprowadza możliwość operowania na wpisach przepływów, co daje możliwości dostęp u do całego nagłówka pakietu dla warstwy L2 i L3, który jest dostępny dla działań dopasowywania i modyfikowania. W zakresie konfiguracji i rozszerzeń powstał protokół of-config, zbudowany według schematów XML modeli danych Yang i protokołu NETCONF[15][21].

2.1.5. PRZEŁĄCZNIK OPENFLOW.

Przełącznik OpenFlow jest integralną częścią sieci SDN opartej na OpenFlow. Architektura przełącznika OpenFlow jest przedstawiona na rysunku 4[25].



Rysunek 4. Architektura przełącznika OpenFlow[25].

Najważniejszymi komponentami przełącznika OpenFlow są: jedna lub więcej tabel przepływu (ang. *flow table*), tabela grup (ang. *group table*), tabela liczników (ang. *meter table*), porty (fizyczne lub logiczne) oraz jeden lub więcej kanałów OpenFlow (ang. *OpenFlow channel*), łączących się ze sterownikiem zewnętrznym (kontrolerem)[20]. Tabela przepływów w przełączniku zawiera zestaw wpisów przepływu pakietów danych. Po nadejściu pakietu, rozpoczyna się dopasowywanie, od pierwszej tabeli przepływu i jest kontynuowane w sekwencji do dodatkowych tabel przepływu. Jeśli nastąpi dopasowanie, pakiet jest przetwarzany zgodnie z określoną akcją we wpisie przepływu. Jeśli jednak nie zostanie znaleziony pasujący wpis, akcja zostanie podjęta zgodnie z instrukcjami w tabeli przepływu chybień (ang. *miss flow entry*). Możliwy zestaw akcji w przypadku braku wpisu w tablicy to: przekaż pakiet do kontrolera poprzez kanał OpenFlow, kontynuuj wyszukiwanie w następnej tabeli przepływu, odrzuć (drop) pakiet, itp. Tabela grup składa się z zestawu wpisów grupy, a każdy wpis grupy składa się z zestawu zasobników akcji o określonej semantyce w zależności od typu grupy. Akcje w co najmniej jednym zasobniku akcji są stosowane do pakietów wysyłanych do grupy. Pakiety przychodzące mogą być przekazywane do portu zgodnie z akcją wymienioną w pasującym wpisie przepływu. Porty są zazwyczaj fizyczne, jednak mogą być logiczne lub zastrzeżone.

2.1.6. KONTROLER (STEROWNIK) SIECI SDN

Paradygmat sieci SDN opiera się na trzech najważniejszych koncepcjach tj. rozdzielenie płaszczyzny sterowania i danych, programowalność, oraz zarządzanie stanem całej sieci w scentralizowanym modelu sterowania[21]. Koncepcje te zawarte są w wyidealizowanej strukturze sieci SDN, ucieleśnieniem tej struktury jest właśnie Kontroler SDN (zwany czasem sterownikiem sieci). Kontroler SDN jest analogicznie jak procesor dla komputera, mózgiem dla całej sieci SDN. W oparciu o swój globalny widok sieci optymalnie steruje kontrolą przepływu na płaszczyźnie danych. Jak również dostarcza inne funkcjonalność, jak monitorowanie czy zarządzanie z zcentralizowanego sterownika SDN. Jest odpowiedzią na rozwiązanie niektórych problemów współczesnego Internetu, w tym bezpieczeństwa, złożoności zarządzania, multiemisji, równoważenia obciążenia czy efektywności energetycznej[81].

Do tej pory opracowano i udostępniono wiele implementacji kontrolerów SDN. Są one przeznaczona zarówno do zastosowań badawczych (akademickich) jak i produkcyjnych, są stosowane w różnych wdrożeniach w obszarze telekomunikacyjnym i przemysłowym. Istnieją rozwiązania komercyjne jak i wersje otwartej (ang. *open source*). Implementacje kontrolerów SDN są napisane w różnych językach programowania jak i na różne systemy operacyjne. Wybierając kontroler SDN dla własnych zastosowań, dobrze jest kierować się określonymi kryteriami doboru. Do najważniejszych kryteriów należą:

- ocena wydajności kontrolera (czas odpowiedzi na zapytanie PACKET_IN, ilość komunikatów /s)
- obsługa szyfrowanego kanału komunikacji TLS, obsługa protokołu OpenFlow (ver. 1.0 do 1.5)
- dostępne moduły dla interfejsów (północnego, południowego)
- obsługa protokołów (REST API)
- obsługa modułowość (jaki jest udostępniany język programowania modułów)

- funkcje dodatkowe, GUI
- język programowani w którym został napisany
- wsparcie producenta/organizacji rozwijającej
- rodzaj licencji

W niniejszej pracy skupiono się na implementacjach kontrolerów SDN w wersjach z otwartym kodem (ang. *open source*), wspierających protokół OpenFlow. W zasadzie większość kontrolerów SDN wspiera protokół OpenFlow, a tak naprawdę opiera swoje działanie na nim, lecz warto wspomnieć iż istnieją implementacje komercyjne nie posiadające obsługi OpenFlow.

Jednym z pierwszy kontrolerów z omawianej grup jest niewątpliwie kontroler NOX opracowany w 2008 roku napisany w języku C++. Jego nowszą implementacją w języku Python jest POX. Wartym uwagi kontrolerem jest Ryu (po japońsku FLOW) napisany również w języku Python. W zestawieniu powinny się znaleźć również napisane w języku Java kontrolery Floodlight, OpenDayLight i ONOS – obecnie jako część CORD-a. Porównanie funkcjonalne kontrolerów przedstawiono w tabeli nr 1 [21], [81], [82].

Funk./Kontroler	NOX	POX	Ryu	FloodLight	OpenDayLight	ONOS
Obsługa TLS	Nie	Tak	Tak	Tak	Tak	Tak
Południowy interfejs	OF 1.0	OF 1.0	OF 1.0 - 1.5 NETCONF OFCONFIG OVSDB	OF 1.0 – 1.5	OF 1.0 -1.5 NETCONF YANG OVSDB PCEP BGP/LS LISP SNMP OFCONFIG	OF 1.0 -1.5 NETCONF
REST API	NIE	NIE	TAK	TAK	TAK	TAK
Obsługa OS	LINUX	LINUX, MAC OS, WINDOWS	LINUX	LINUX, MAC OS, WINDOWS	LINUX, MAC OS, WINDOWS	LINUX, MAC OS, WINDOWS
Obsługa języka prog.	C++	Python	Python	Java	Java	Java
GUI	Tak	Tak	Tak	Web UI	Web	Web
Wirtualizacja	Tak	Mininet OvS	Mininet OvS	Mininet OvS	Mininet OvS	Mininet OvS
Przeznaczenie	Kampus	Kampus	Kampus	Kampus	Data center SD-WAN	Data center SD-WAN

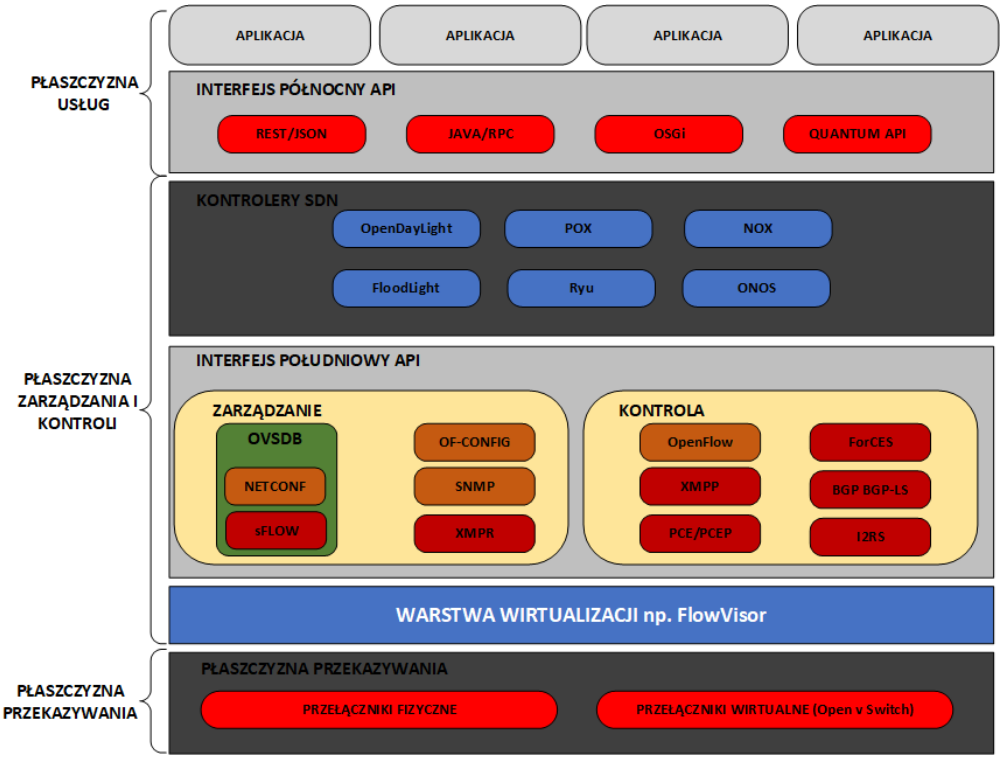
Tabela nr 1 Zestawienie funkcjonalności wybranych kontrolerów SDN

Na rysunku 6 przedstawiono architekturę sieci SDN podzieloną na płaszczyzny z uwzględnieniem płaszczyzny kontrolerów i wyszczególnieniem obsługiwanych interfejsów (północ/południe).

Fizyczne implementacje kontrolerów to aplikacje osadzone w systemach operacyjnych, działające w trybie serwera. Twórcy kontrolerów SDN udostępniają je w postaci pakietów instalacyjnych dla odpowiednich systemów operacyjnych. Istnieją również gotowe instalacje w postaci maszyny wirtualnej dla środowiska hyperwaisora np. VirtualBox. W zastosowaniach badawczych sterowniki SDN są często zaimplementowane w oprogramowaniu laboratoryjnym tj. Mininet czy Packet Tracer czy GNS3.

Dzięki częściowej standaryzacji sieci SDN poprzez implementacje protokołów tj. OpenFlow, czy NETCONF, istnieje wymiany informacji pomiędzy kontrolerami. Komunikacja odbywa się za pomocą protokołu TLS. Kontrolery mogą działać w mieszanym środowisku tzn. w jednej logicznej sieci SDN mogą pracować kontrolery różnych producentów.

Ze względów wydajnościowych i bezpieczeństwa (kontroler SDN może stać się pojedynczym punktem awarii) zaleca się aby w środowiskach produkcyjnych stosować rozwiązania redundantne. Przykłady wdrożeń kontrolerów SDN opierają się o różne rozwiązania wysokiej dostępności (HA). Dobrymi metodami jest instalacja kontrolerów w środowisku wirtualnym np. VMware i korzystanie z funkcjonalności samego hypervisor do zabezpieczania takich funkcji jak: HA, redundancja, rozproszenie. W dużych sieciach ISP kontrolery SDN są wyniesione do różnych lokacji, a połączenie między nimi są tunelowane i szyfrowane. Taka architektura eliminuje pojedynczy pkt. awarii, dostępność ze względu na obciążenie, balansowanie obciążenia dla sieci z wieloma urządzeniami SDN.



Rysunek 6. Architektura sieci z płaszczyzną kontrolerów SDN[81].

Po analizie funkcjonalnej przedstawionych kontrolerów w pracy zastosowano implementacje sterowników SDN: POX, Ryu, OpenDaylight dla wirtualnego środowiska testowego Mininet. Natomiast w środowisku produkcyjnym został zastosowany kontroler CORD z systemem ONOS. Wykorzystano

również możliwość implementacji kontrolerów SDN różnych producentów w tej samej logicznej sieci SDN.

2.1.7. BEZPIECZEŃSTWO SIECI SDN PRZEZ OCHRONĘ KONTROLERA SDN.

Struktura sieci SDN ma pewne problemy projektowe, które sprawiają, że jest podatna na różne zagrożenia bezpieczeństwa. Warstwa kontrolna jest uważana za najbardziej czuły cel ataku DDoS [26]. Poniżej przedstawiono zagrożone funkcje architektury SDN atakowane przez DDoS:

- Przeciążenie pamięci tabeli przepływów. Atak sieciowy DDoS generuje dużą ilość przepływu skierowanego do przełącznika krawędziowego. Ponieważ te przepływy nie są wcześniej widoczne dla przełącznika, przełącznik nie znajdzie reguł dopasowania do przekazywania tych przepływów. Następnie przełącznik wprowadza wpisy do nowych przepływów i wysyła do kontrolera, aby zażądać zasad przekazywania. Jednak przełącznik ma ograniczone zasoby, a ciągle przychodzące pakiety przeciążają pamięć, dlatego nie może przetworzyć legalnego ruchu.[26]

- Przeciążenie zasobów kontrolera. Kontroler jest mózgiem sieci SDN, która odpowiada za kontrolę i zarządzanie wszystkimi operacjami sieciowymi. Gdy procesor i pamięć kontrolera jest przeciążona przez zalew żądań wygenerowany przez atak sieciowy DDoS, nie może on przetworzyć nowych przepływów przychodzących. Powoduje to pogorszenia wydajności w całej sieci i wpływa na dostępność kontrolera. [26]

- Przeciążenie przepustowości na linii przełącznik-kontroler. Zalew żądań ataku typu DDoS powoduje przeciążenie kanału komunikacji OpenFlow. W ten sposób następuje odmowa usługi legalnym użytkownikom.[26]

Sieci definiowane programowo(SDN) pojawiły się, jako obiecująca technologia, która ma unikalne funkcje, takie jak programowalność kontrolera, globalny scentralizowany widok stanów sieci i oddzielenie płaszczyzn sterowania i danych. Wykorzystując te właściwości, możliwe jest zbudowanie solidnych i elastycznych systemów wykrywania anomalii, a także skuteczne reagowanie na te anomalie. W dużej części podstawowa architektura SDN składa się z jednego kontrolera. W związku z tym scentralizowana funkcja sterowania ma jeszcze jedną wadę. Pojedynczy kontroler ma ograniczoną pojemność i pojedynczy punkt awarii, wszystko to sugeruje budowanie środowiska redundantnego [19]. Architektura SDN ma również inne projektowe luki strukturalne, którymi są oprócz scentralizowanego kontrolera, interfejs sterowania danych i interfejs kontroli aplikacji. Podatności na pewno zostaną wykorzystane przez hakerów do przeprowadzenia co najmniej kilku rodzajów ataków.

Kreutz i in.[27] wprowadzili siedem wektorów zagrożeń w sieciach SDN. Trzy z nich są specyficzne dla SDN i związane ze kontrolerem: interfejs sterowanie-dane i interfejs sterowanie-aplikacji. Sam kontroler jest podatny na ataki, ponieważ tworzy pojedynczy punkt ataku i awarii. Szyfrowanie kanału komunikacyjnego przy użyciu protokołu TLS (ang. Transport Layer Security) jest

również czasami nieskuteczne i opcjonalne. Architektura SDN jest podatna na kilka rodzajów ataków. Na przykład ataki Distributed Denial of Service (DDoS) przytłaczają kontroler i kanał komunikacyjny sztucznymi wywołania mi usług. Atak Man-in-the-Middle może przerwać połączenie między kontrolerem a przełącznikami i przejąć kontrolę nad siecią w sposób przeźroczysty dla administratora.

Belyaev i in. [82] zaproponowali dwupoziomowe rozwiązanie równoważenia obciążenia w sieciach SDN w celu wydłużenia czasu przetrwania systemu podczas ataku DDoS. Ich główną pracą jest równoważenie obciążenia, a tym samym nie łagodzą ataku DDoS, ale pozwala to na zwiększenie czasu przetrwania, dzieląc obciążenie.

Dao i in. [71] zaproponowali twardy mechanizm limitu czasu w celu stopniowego wycofywania fałszywych wpisów tabeli przepływu utworzonych przez atakującego w celu zatkania kanału komunikacyjnego przełącznik-kontroler i przepełnienia pamięci TCAM przełącznika. Limit czasu jest jawnie stosowany do dowolnych rzadkich przepływów dla pakietów błędów lub DDoS, których jedynym celem jest przytłoczenie pojemności przełącznika OF.

StateSec [72] wykorzystuje technikę stanową (stateful) SDN w kontekście ochrony DDoS i deleguje lokalne przetwarzanie do przełączników. Przełączniki bezpośrednio obsługują monitorowanie ruchu istotnych funkcji (np. źródła i miejsca docelowego IP, źródła portu i miejsca docelowego) za pomocą programowania stanów, zmniejszając w ten sposób obliczenia i w konsekwencji obciążenie kontrolera. Dokładne wyniki są następnie przekazywane do opartego na entropii algorytmu wykrywania ataków na kontrolerze

2.2. TECHNIKI WYKRYWANIA ATAKÓW DDOS

Wykorzystanie właściwości sieci SDN do zbudowania systemu bezpieczeństwa sieciowego to obszar badawczy, który ostatnio zyskał szczególną uwagę. Przeprowadzone liczne badania próbujące rozwiązać wiele problemów przy użyciu różnych metod i technik, wykrywanie włamań nadal nie zostały odpowiednio uwzględnione w kontekście sieci opartych na strukturze SDN. W tej sekcji przedstawiono techniki wykrywania włamań, godne uwagi rozwiązania, które wykorzystują koncepcję sieci SDN do wykrywania ataków sieciowych DDoS.

2.2.1. TEORIA INFORMACJI – METODY Z TRADYCYJNEJ SIECI

W tradycyjnej sieci techniki oparte na teorii informatyki są szeroko stosowane w skuteczny sposób w celu radzenia sobie z wykrywaniem anomalii. Techniki teorii informacji wykorzystują podejście probabilistyczne i teorię statystyczną do modelowania entropii[31]. Entropia służy do pomiaru średniej informacji o dystrybucji cech ruchu sieciowego w danym przedziale czasu. Algorytm oparty na entropii odniósł sukces w analizowaniu drobnoziarnistych wzorców przy zmniejszonych kosztach obliczeniowych, co czyni go odpowiednim do wykrywania odchyleń zachowania ruchu.

W wielu metodach skutecznie zastosowano algorytm oparty na entropii, w sieciach opartych na koncepcji SDN. Korzystając z oszacowania maksymalnej entropii, Mehdi i in. [32] definiują rozkład normalnego ruchu w celu wykrycia anomalii w sieciach domowych i biurowych opartych na SDN. Używają maksymalnego oszacowania entropii do opracowania podstawowego łagodnego rozkładu dla każdej klasy zawierającej zestaw pakietów. Obserwują klasę rozkładu pakietów w danym okresie czasu, a następnie porównują ją z dystrybucją bazową. Jednak w eksperymencie wykorzystali tylko ruch sieciowy o niskim natężeniu ruchu i skupili się na małym środowisku (SOHO). [52]

W pracy [33] badacze wykorzystali podejście oparte na entropii do identyfikacji anomalii. Zaproponowali architekturę składającą się z kolektora sFlow, modułu wykrywania anomalii i łagodzenia anomalii. Zintegrowana technika próbkowania może skutecznie zminimalizować obciążenie kontrolera w porównaniu z natywnym podejściem OpenFlow. Eksperyment ten pokazuje skuteczność rozwiązania w pokonywaniu różnych typów ataków, w tym takich jak DDoS. Ataki typu rozprzestrzenianie się robaków i ataki skanowania portów ze 100% wskaźnikiem wykrywalności. Jednak wykrywanie oparte na entropii zapewnia wysoki wskaźnik fałszywych alarmów sięgający 40%.[52]

Podobnie, aby przezwyciężyć ograniczenie skali pojedynczego kontrolera, Wang i in.[34] zaproponowali rozproszone IDS w sieci wielkoskalowej. Zaproponowali wykrywanie ataków DDoS za pomocą metody opartej na entropii działającej na przełączniku brzegowym, aby zminimalizować obsługę pakietów kontrolera, zmniejszając w ten sposób przeciążenia kontrolera. [52]

Sahoo i in. [35] zaproponowali uogólnioną entropię (GE) do wykrywania ataków DDoS o niskiej częstotliwości na kontrolerze na wczesnym etapie. Ich eksperymenty wykazały, że GE zapewnia wysoki wskaźnik wykrywalności w porównaniu z innymi wskaźnikami odległości informacji. [52]

Ograniczeniem wszystkich wcześniejszych propozycji jest to, że zależą one od jednego kontrolera, co czyni je podatnymi na awarie, zwłaszcza w sieci wielkoskalowej[36]. Co więcej, ich metoda wykrywania opiera się tylko na entropii. W algorytmie opartym na entropii, chociaż obliczenie pojedynczej wartości kilku cech jest skuteczne pod względem analizy, powiązane informacje o tych cechach zostaną utracone, co prowadzi do ukrycia efektów anomalii[40]. [52]

2.2.2. TECHNIKA OPARTA NA REGUŁACH

Metoda wykrywania włamań oparta na regułach jest jednym z najczęściej stosowanych podejść opartych na wiedzy[38]. Metoda oparta na regułach jest zaprojektowana tak, aby pasowała do bieżącego stanu systemu zgodnie z zestawem reguł przechowywanych w silniku reguł. Gdy wystąpi dopasowanie, oznacza to obecność potencjalnego ataku. Silnik reguł generuje alert i zwalnia jedną lub więcej reguł. Aplikacja Snort to najpopularniejszy IDS oparty na regułach. Jest to rozwiązanie na licencji open-source, w które użytkownicy mogą ingerować, gdy rozpoznają nowe anomalie.

Xing i in. [39] wykorzystali elastyczną rekonfigurację sieci SDN i połączenie IDS-a Snort, aby zapewnić ochronę sieci opartych na chmurze. Agenci Snorta zbierają dane i wysyłają je do serwera Snort, aby dopasować je do istniejących reguł. Po wykryciu ataku, informacja jest przekazywana do kontrolera, aby zastosować odpowiednią politykę przez interfejs OpenFlow. Jednak ich eksperyment wykazał tylko możliwość zintegrowania struktury SDN z IDS-em Snort. Projekt nie zawierał wyników wykrywania anomalii.[52]

Jeong i in.[40] zaproponowali rozwiązanie, w którym IDS Suricata działający w oparciu o środowisko maszyny wirtualnej jest podłączony do kontrolera Floodlight i steruje przełącznikiem z obsługą OpenFlow. Aby zminimalizować przeciążenie kontrolera, wykorzystują techniki próbkowania ruchu. Niemniej jednak dostosowali próbkowanie w zależności od pojemności IDS, a nie od wielkości ruchu sieciowego. [52]

Fawcett i in. [41] zaproponowali wielopoziomowe rozproszone obszary monitorowania i remediacji wspierające wiele kontrolerów o nazwie TENNISON. W tym rozwiązaniu ruch o dużym natężeniu jest monitorowany za pomocą „lekkich" metod na pierwszych dwóch poziomach; w związku z tym, jeśli filtrowane przepływy wymagają głębokiej inspekcji pakietów za pomocą Snort i Bro jest to przeprowadzane na trzecim poziomie. Wprowadzono również niezależną warstwę między płaszczyznami sterowania i aplikacji, odpowiedzialną za zapewnienie szerokiego widoku stanu sieci i koordynacji informacji o przepływie między urządzeniami sieciowymi a warstwą aplikacji. [52]

Yan i in.[42] zastosowali technologię fog computing, edge computing i cloud computing, aby wykrywać ataki DDoS przy użyciu sieci SDN. Zadanie bezpieczeństwa jest rozłożone na wszystkie trzy warstwy, w których silnik detekcji używa IDS Snort i znajduje się w pobliżu ofiary w warstwie chmury,

a odpowiedź jest przetwarzana w pobliżu źródła w warstwie krawędziowej (edge), komunikacja między warstwą chmury i krawędzi (edge) odbywa się przez warstwę mgły (fog computing), która zawiera logikę sterowania. [52]

Techniki oparte na regułach są uważane za skuteczne, elastyczne i mają wysoki wskaźnik wykrywalności, jednak mogą nie być w stanie wykryć nieznanych ataków. Co więcej, są one nieodpowiednie dla dużych sieci, w których szybkość w kontrolowaniu ruchu jest kluczowym czynnikiem, co sprawia, że nielogiczne jest dopasowywanie pakietów do wielu setek zasad.[43]

2.2.3. TECHNIKI OPARTE O UCZENIE MASZYNOWE

Technika uczenia maszynowego jest szeroko stosowana do klasyfikowania ruchu i wykrywania anomalii [44]. Uczenie maszynowe zostało podzielone na uczenie nadzorowane i nienadzorowane. Nadzorowane metody są trenowane przez normalne dane z ruchu, więc zwykle nie mają możliwości wykrywania nieznanych sygnatur. W przeciwieństwie do tego, nienadzorowane uczenie się może wykrywać znane i nieznane ataki. Metody nienadzorowane grupują albo klastrują dane w oparciu o podobieństwo niektórych cech, w których największe klastry są oznaczone jako normalne zachowanie, a mniejsze są traktowane jako nienormalne (podejrzane). [52]

Coraz większa grupa badaczy wykorzystuje koncepcje sieci SDN i wdraża metody uczenia maszynowego do wykrywania ataków sieciowych.

Autorzy [47] zaproponowali framework FADM do wykrywania i mitygacji ataków zalewania DDoS w środowiskach SDN przy użyciu SVM. Zastosowali zmienny poziom próbkowania w zależności od różnych środowisk sieciowych. Jednak metoda klasyfikacji powoduje znaczne narzuty dla kontrolera. Braga i in.[45] zaimplementowali analizę przepływu przy użyciu samoorganizujących się map do wykrywania ataków DDoS. Ich metoda opiera się na sześciu funkcjach ruchu flow zebranych przez kontroler NOX w celu rozróżnienia, między nielegalnymi przepływami a normalnym ruchem. Autorzy wykorzystali Self Organizing Maps (SOM), nienadzorowaną sztuczną sieć neuronową wyszkoloną z cechami przepływu ruchu, aby sklasyfikować przepływy ruchu sieciowego jako prawidłowy lub nieprawidłowy. Moduł analizy osadzili w kontrolerze NOX. Monitują zarejestrowane przełączniki w określonych odstępach czasu do pobierania informacji z przepływów. Te przykładowe informacje są następnie przekazywane do modułu SOM, który klasyfikuje ruch jako normalny lub atak. Chociaż ich praca osiągnęła wysoki wskaźnik skuteczności wykrywania ataków, ich eksperymenty opierają się na małej topologii i pojedynczym kontrolerze. W pracy [46] badacze zaproponowali framework o nazwie ATLANTIC do wykrywania, klasyfikowania i łagodzenia ruchu anomalii w oparciu o SDN. Wykrywanie anomalii przebiega w dwóch trybach; w normalnej sytuacji identyfikator działa w trybie „lekkim" przy użyciu entropii, a gdy obserwuje się odchylenie entropii, identyfikator zmienia się na tryb „ciężki" wykorzystujący klasyfikację SVM i technikę klastrowania „k-means". Jeśli wystąpi nieznany atak lub anomalne zachowanie, system będzie potrzebował interakcji człowieka, aby go zidentyfikować i zaktualizować profile anomalii. Rozwiązania proponowane w pracach zarówno[46], jak i [47] mają problem ze skalowalnością, ponieważ te rozwiązania obsługują tylko jeden kontroler SDN.

Przeanalizowano również kilka wybranych badań, które integrują zastosowanie wielu kontrolerów SDN i uczenia maszynowego.

W [43] badacze przedstawili freamework Athena, jako rozproszoną strukturę wykrywania anomalii. Athena wykorzystuje rozproszoną bazę danych i klastry obliczeniowe do implementacji algorytmu wykrywania anomalii w sieciach wielkoskalowych[4] przy użyciu instancji z wieloma kontrolerami. Bhunia i Gurusamy [48] zaproponowali rozwiązanie SoftThinges, framework oparty na SDN, do wykrywania i reagowania na ataki na wczesnym etapie przy użyciu hierarchicznego multikontrolera i SVM. Po wykryciu nowego ataku w jednym klastrze kontroler główny otrzymuje informacje o ataku i wdraża je w innych klastrach. Jednak ich eksperyment były przeprowadzana dla małej sieci. Zastosowana architektura sieci wymaga większej ilość badań pod względem metody łączenia kontrolera, metod komunikacji z kontrolerami. Jak pokazano powyżej, żadne z wymienionych rozwiązań nie wyczerpuje założeń dla rozwiązania w pełni kompleksowego. W szczególności w pracach[33], [46], [47] oparto się na rozwiązaniu z pojedynczym kontrolerem, który ma ograniczoną pojemność, a zatem nie spełnia wymagań skalowalności rozwiązania w większych sieciach. Ponadto niektóre z nich [33], [41], [42], nie przyjmują metody uczenia maszynowego lub używają jej bezpośrednio[47], [48] co sprawia, że nie nadaje się do zastosowania w sieci wielkoskalowej[4]. Wreszcie, Athena [43] obsługuje wiele kontrolerów i zapewnia wiele algorytmów uczenia maszynowego w celu wsparcia i ułatwienia badań w SDN, ale ogranicza wybór, gdy chcemy wykorzystać inną metodę wraz z uczeniem maszynowym. W pracy [49] zaproponowano rekonfigurowanie sieci na schemacie wykorzystującym SDN przeciwko atakowi za pomocą sieci botnetów. W przypadku możliwego ataku DDoS serwer przekierowuje chronioną usługę na nowy zestaw utrzymywanych publicznych adresów IP, wykorzystując centralne i dynamiczne zarządzanie siecią oferowane przez paradygmat SDN. Podobne podejście do przekierowywania ruchu zastosowano w pracy [53]. Jednak zamiast przekierowywać usługi na nowe adresy IP, po identyfikacji ataku przekierowują ruch z dala od ofiary na alternatywne trasy lub do zapadlisk (sinkholes). W [51] autorzy wprowadzili przypadek użycia systemu mitygacji ataków DDoS opartego na SDN, aby zapewnić autonomiczną i szybką rekonfigurację podejrzanego ruchu sieciowego. Praca ta w obecnej formie jest bardzo podstawowa, nie posiada koncepcji i wniosków w postaci wyników.[52]

Bohatei [53] proponuje elastyczny system obrony DDoS, który wykorzystuje możliwości NFV do płynnego zmieniania wymaganej skali (volume atack) i typu (np. SYN proxy vs. obrona reflektora DNS) obrony DDoS. System kieruje podejrzanym ruchem przez wykresy strategii obrony, które są oparte na liczbie pakietów i predefiniowanych podejrzanych zrachowaniach.

W pracy [54] autorzy wykorzystali kombinację technik SDN i uczenia maszynowego do wykrywania i blokowania ataków odbicia amplifikacji (DDoS). Przełącznik OpenFlow kopiuje ruch do agenta wykrywania, który stosuje metodę maszyny wektora wsparcia (SVM) w celu sklasyfikowania pakietów jako złośliwych lub niezłośliwych. Po wykryciu złośliwego ruchu moduł agenta powiadamia kontroler o zablokowaniu złośliwych pakietów.

W [51] autorzy wykorzystali przepływy sFlow z kontrolera SDN, aby skutecznie wykryć i złagodzić atak na system DNS. Jeśli ruch przychodzący zawiera pasujący identyfikator zapytania żądania w rekordach przepływu, jest klasyfikowany jako normalna odpowiedź DNS, w przeciwnym razie jest oznaczony jako podejrzany, ponieważ w pierwszej kolejności nie zainicjowano żadnego

zapytania. Oznaczone przepływy są następnie przekazywane do kontrolera SDN w celu mitygacji skutków.[52]

System ochrony oparty na reflektorach (FL-GUARD) zaproponował w [55] trzyetapowe podejście do ataków DDoS. Najpierw stosują dynamiczne wiązanie adresów IP w celu rozwiązania problemu fałszowania adresów IP, następnie używają algorytmu SVM do klasyfikowania ruchu ataku, a na koniec wykorzystują scentralizowany kontroler do blokowania ataków na porcie źródłowym.[52]

Uczenie maszynowe (ML) zostało wykorzystany przez kilku badaczy w [56] do wykrywania włamań w tradycyjnej sieci. Wykorzystali samoorganizującą się mapę (SOM) w swoich badaniach i osiągnęli dokładność 75,49%. W [57], Mukherjee i in. zbadali kombinację wyboru funkcji opartego na korelacji, wzmocnienia informacji i współczynnika wzmocnienia w celu redukcji funkcji. Następnie zredukowany zestaw danych został sklasyfikowany przez naiwny algorytm Bayesa, który daje dokładność 97,78%.

Głębokie uczenie maszynowe (DL) zostało również zastosowany do wykrywania włamań, ale tylko w tradycyjnych sieciach. Autorzy Javaid i in.[58] wykorzystali algorytm głębokiego uczenia maszynowego na zbiorze danych NSL-KDD do wykrywania włamań. Jako klasyfikator zastosowali miękką regresję -max i osiągnęli dokładność 92,98%. Yin i in.[59] zaproponowali podejście głębokiego uczenia się podczas korzystania z powtarzających się sieci neuronowych do wykrywania włamań. Uzyskali dokładność 83,28% dzięki eksperymentowi na zbiorze danych NSL-KDD. Shone i in.[60] zaproponowali niesymetryczny głęboki autoencoder do nienadzorowanego uczenia się cech. Osiągają dość obiecujące wyniki zarówno w zbiorach danych KDD Cup 99, jak i NSL-KDD. Podejścia te nie mogą być jednak stosowane bezpośrednio w kontekście SDN.

W kolejnych analizowanych artykułach zaproponowano kilka algorytmów i podejść do wykrywania włamań w celu zabezpieczenia sieci SDN opartych na OpenFlow. W przypadku podejść do wykrywania opartych na anomaliach, SOM i Support Vector Machine (SVM) są często używane ze względu na ich wysoką dokładność wykrywania. Lekka metoda wykrywania ataków DDoS oparta na funkcjach przepływu ruchu została przedstawiona w [45] z wyodrębnianiem sześciu funkcji: średnia pakietów na przepływ, średnia bajtów na przepływ, średnia czasu trwania na przepływ, procent przepływów, wzrost pojedynczych przepływów i wzrost ilości różnych portów. SOM – technika oparta jest na sieci neuronowej - została używana jako metoda klasyfikacji. SVM został użyty w pracach [61][62] do wykrywania ataków DDoS w sieciach SDN z wysokim wskaźnikiem skuteczności. Jednoklasowa maszyna SVM została przetrenowana z zestawem danych złośliwych w celu uzyskania niskiego wskaźnika fałszywych alarmów w [63]. AlEroud i in. [66] połączył metodę k-Najbliższy sąsiad i teorię grafów, aby sklasyfikować ataki DDoS z przepływu w sieci SDN. W pracy [65] autorzy połączyli sieci neuronowe i teorię zagrożeń dla w kontekście odporności sieci na ataki DDoS. S. M. Mousavi zaproponował w [66] metodę wczesnego wykrywania ataków DDoS na kontroler SDN. Metoda ta opiera się na odmianie entropii docelowych adresów IP z przepływów danych. W pracy [67] autorzy połączyli ustawione na twardo progi wykrywania i rozmyty system wnioskowania w celu wykrycia ryzyka ataków DDoS w oparciu o rzeczywistą charakterystykę ruchu (rozkład średniego czasu między przybyciem pakietu danych, rozkład ilości pakietów na przepływ i ilość przepływu na serwer) w stanach normalnych i podczas ataków.

Aby poprawić skalowalność natywnego protokołu OpenFlow, w pracy [68] zaproponowano połączenie OpenFlow i sFlow w celu stworzenia skutecznego i skalowalnego mechanizmu wykrywania i łagodzenia anomalii w środowisku SDN. Wykorzystując programowalność architektury SDN, autorzy [69] pokazują, że programowalny router sieci domowej może zapewnić idealną platformę i lokalizację w sieci do wykrywania problemów bezpieczeństwa dla sieci SOHO (Small Office/Home Office). Autorzy zaimplementowali cztery znaczące algorytmy wykrywania anomalii ruchu (próg random walk z ograniczeniem stawek opartych na kredytach, ograniczenie szybkości, detektor maksymalnej entropii i NETAD). Eksperymenty wskazują, że algorytmy te są znacznie dokładniejsze w identyfikowaniu szkodliwych działań w sieci SOHO niż te oferowane przez ISP (ang. Internet Service Provider), oraz iż detektor anomalii może pracować z szybkością łącza bez wprowadzania nowych narzutów wydajności dla ruchu sieci domowej. System SPHINX został wprowadzony w pracy [69] w celu wykrywania zarówno znanych, jak i potencjalnie nieznanych ataków w SDN poprzez wykorzystanie nowatorskiej metody abstrakcji wykresów przepływowych, które ściśle przybliżają rzeczywiste operacje sieciowe. SPHINX dynamicznie uczy się nowego zachowania sieci i generuje alerty, gdy wykryje podejrzane zmiany w płaszczyźnie kontroli sieci. Autorzy analizują dokładność, opóźnienia i przepustowość sieci z analizą SPHINX, aby pokazać, że system SPHINX jest w stanie wykrywać ataki w czasie rzeczywistym przy niskim nakładzie pracy.

2.2.4. INNE ROZWIĄZANIA MITYGACYJNE.

Techniki oparte na tokenach, takie jak te zgłoszone w [73], [82], [83] zwykle zakładają, że nadawca i odbiorca muszą ustanowić uprzywilejowany kanał komunikacji: nadawca prosi o token możliwości, od odbiorcy, a jeśli odbiorca zgodzi się na nawiązanie połączenia, wysyła token możliwości do nadawcy, który następnie osadza ten token w kolejnych pakietach. Oczywiście systemy oparte na tokenach, możliwościach mogą zapobiegać atakom DDoS, ale konieczna jest obsługa infrastruktury w całej sieci. System może nie działać w przypadku awarii urządzenia sieciowego. Kolejnym prewencyjnym podejściem anty-DDoS jest inteligentna kontrola zatorów, która ma na celu ograniczenie natężenia ruchu w oparciu o podane progi. Jednym z typowych przykładów jest Pushback [76], który ostrzega routery nadrzędne ofiary o porzuceniu ruchu ataku na podstawie sygnatur przeciążenia, które są wcześniej oceniane, podczas gdy tylko legalny ruch może przepływać przez sieć. Jednak takie schematy filtrowania wymagają również wdrożenia dodatkowych reguł na każdym routerze w sieci. Cały system może się zatrzymać, jeśli jedno urządzenie na routerze podrzędnym ulegnie awarii, ponieważ sygnatura przeciążenia nie będzie propagowana do nadrzędnego routera.

Prace zgłoszone w [77], [82] pokazują, że niektóre stanowe zasady łagodzenia skutków ataków DDoS można stosować w celu przekierowania ruchu DDoS do skrzynek pośrednich (middleboxów) na żądanie, gdy klient doświadczy ataku. W szczególności model dFence zaproponowany w[57], wymaga wdrożenia skrzynek pośrednich w sieci bazowej oraz przechwycenia obu kierunki ruchu IP. Ponadto zaproponowano również mechanizm śledzenia adresów IP [11] w celu złagodzenia ataków DDoS. Chodzi o to, aby oznaczyć pakiety IP pewnymi informacjami, takimi jak interfejs przychodzący routera lub jego identyfikator, który następnie zostanie wykorzystany do rekonstrukcji ścieżki od ofiary do

źródła ataku. Jednak ze względu na fakt, że różne ścieżki mogą kończyć się tym samym identyfikatorem ścieżki [79], wskaźnik fałszywych alarmów jest dość wysoki. Ponadto generowanie i konserwacja znaków wprowadza przetwarzanie overhead na routerach, a ofiara jest zobowiązana do zebrania dużej liczby pakietów w celu identyfikacji ścieżki ataku i zainicjowania procesu śledzenia wstecznego (traceback).

FireCol [79] przedstawia system współpracy na poziomie ISP w celu wykrywania ataków DDoS jak najbliżej źródła (źródeł) ataku opartych na powodziach (flooding). Wiele IPS tworzy sieci pierścieni ochronnych wokół subskrybowanych klientów i współpracują poprzez obliczanie i wymianę „przekonań" na temat potencjalnych ataków. Atak jest mierzony na podstawie ogólnej przepustowości ruchu kierowanego do klienta w porównaniu z maksymalną przepustowością, którą obsługuje.

CoFence [80] proponuje mechanizm DDoS defense między sieciami domen równorzędnych opartych na NFV. CoFence umożliwia sieciom domenowym udostępnianie zasobów innym elementom równorzędnym w oparciu o wzajemną funkcję użyteczności. Umożliwia to sieciom domenowym atakowanym przez przeciążenie DDoS skuteczne przekierowywanie nadmiernego ruchu do innych współpracujących domen w celu filtrowania.

3. PROPONOWANE ROZWIĄZANIE PROBLEMU.

W wyniku bardzo wnikliwego przeglądu bieżącego stanu wiedzy; wykonanych szeregu testów i badań z zakresu rozwiązań systemów mitygacji ataków sieciowych DDoS, jak i drobiazgowego studiowania zagadnień sieci komputerowych SDN [84]; dokonanej analizy porównawczej metod i algorytmów wykrywania ataków również w oparciu o algorytmy uczenia maszynowego, powstała koncepcja nowatorskiego sieciowego systemu informatycznego mającego za zdanie skuteczne wykrycie i mitygację, współczesnych ataków sieciowych typu DDoS. Zaproponowany system posiada budowę modułową uwzgledniającą, określone warunki brzegowe działania, skalowalność jak również rozproszone działanie w sieci Internet. System nosi nazwę STRAINER. Został on zaprojektowany do ochrony rozproszonych lokalizacji sieciowych przedsiębiorstwa. Istnieje możliwość jego wdrożenia w środowisku przemysłowym jak i w środowisku operatora telekomunikacyjnego ISP.

Założenia bazowe co do sposobu i kierunku działania projektowanego sytemu:

1) System STRAINER zaprojektowany jest do wykrywania dwóch podstawowych grupy ataków sieciowych z rodziny DDoS. Pierwsza grupa to ataki wolumetryczne, a druga to ataki aplikacyjne.
2) System zaprojektowany jest w oparciu o paradygmat sieci komputerowych SDN. Jego centralnym punktem detekcji anomalii jest kontroler sieci SDN z niezbędnymi modułami i systemami dodatkowymi. Sieć SDN, i jej urządzenia stanowi kręgosłup dla wdrożonych rozwiązań. Rozwiązania te są z obszarów detekcji, zarządzania i walidacji określonych reguł decyzyjnych. Koncepcja system zostanie przedstawiona w dalszej części pracy.
3) System posiada budowę modułową. Moduły i zawarte w nich autorskie rozwiązania, algorytmy wykrywania, zostały dobrane i zaprojektowane w sposób umożliwiający ich wydajne i efektywne działanie.
4) STRAINER został zaprojektowany w technologii HA, uwzględniając pełną redundancję, skalowalność i elastyczność implementacji.
5) System posiada wariantowe scenariusze wdrożenia, między innymi do ochrony pojedynczej lokacji firmowej, rozproszonej lokalizacji użytkowników firmowych, wielu oddziałów firmy z wyniesioną domeną SDN czy wdrożenie w strukturze ISP z możliwością rozproszenia ataków.
6) Zaprojektowany system posiada wewnętrzne mechanizmy ochrony rdzenia sieci (kontrolera SDN).

3.1.1. GENEZA DZIAŁANIA SYSTEMU STRAINER

Projektując system detekcji ataków sieciowy w kierunku DDoS, czyli system anty-DDoS, należy zdecydowanie rozróżnić starego typu ataki DoS od DDoS. Zgodnie z założeniem tej pracy, autor nie koncentruje się nad znanymi od lat rodzajami ataków DoS i DDoS, a przez to nie buduje rozwiązania, tożsamego z istniejącymi. Detekcją znanych ataków, opisanych sygnaturami w systemie STRAINER zajmuje się między innymi moduł MOD1. Oczywiście autor nie bagatelizuję tej części ataków i nie

wyklucza, że system STRAINER nie będzie zmuszony do ich wykrycia czy odparcia. System ukierunkowany jest na „nowego typu" ataki DDoS z podziałem na ataki wolumetryczne i aplikacyjne.

Głównym efektem ataku DDoS jest tak zwana odmowa dostępu, w wyniku rozproszonego ataku (ang. distributed denial of service). Mająca na celu unieruchomienie systemów ofiary poprzez, wysycenie zasobów jego systemu (hosta), przepustowości sieci czy dosłowne „zalewanie" danymi urządzeń pośredniczących w transmisji tj. routery, przełączniki czy dostępowe UTM-y. Wszystkie te działania maja na celu doprowadzenie do niedostępności serwisów internetowych, usług firmowych czy maksymalnego wysycenia łączy internetowych.

Ataki wolumetryczne, cechują się wysyłaniem przez agresora z rozproszonych hostów typu „zombi", stanowiących grupę atakujących, niechcianych pakietów na określone adresy IP ofiary. Atak ten może odbywać się przez łącze lub łącza internetowe ofiary, a nadchodzących pakietów danych jest tak wiele, iż systemy nie są w stanie ich przyjąć i obsłużyć. Takie działania agresorów są skoordynowane, hakerzy, grupy przestępcze cyberprzestrzeni posiadaj własne zasoby służce do ataku, lub przygotowując się do ataku mają do dyspozycji zainfekowana wcześnie „przejęte systemy typu zombi". Cała operacja ataku jest pod nadzorem i odbywa się w sposób synchronizowany. Siła ataku zależy od ilości maszyn (hostów) użytych w ataku. Długości trwania ataków są zróżnicowane, od kilku minut do miesiąca, ponad 40% to ataki trwające jedną godzinę[83]. Wskaźnikiem rozmiaru ataku jest przepływność. Typowe łącza dostępowe w Polsce mają przepływność pomiędzy 50 a 100 Mb/s. Klasyfikacja przez systemy anty-DDoS ataków wolumetrycznych jest na poziomie od 6 Mb/s przez 250Mb/s do 600 Mb/s. Największy wykryty atak DDoS na świecie był o przepustowości 2,4 Tb/s. i miał miejsce w chmurze Microsoft (Azure), trwał 10 minut[86].

Drugą grupą są ukierunkowane ataki aplikacyjne – celowane w konkretną aplikację internetową, często rozproszoną. Polega on na całkowitym wyczerpaniu zasobów systemowych hosta aplikacji tj. zasoby pamięci czy procesora. Ataki te nazywane są również atakami typu slow. Charakteryzuję się one tym, iż atakowana aplikacja internetowa otrzymuje olbrzymią ilość, a wręcz jest zalewana zapytaniami do aplikacji przy jednoczesnym maksymalizowaniu wydłużania czasu na odpowiedź. Fakt ten powoduje, iż aplikacja otwiera bardzo dużo jednoczesnych sesji typu klient-serwer, dla każdej rezerwuje osobne zasoby. Przy wydłużonym maksymalnie czasie na sesje powoduje to bardzo szybkie wysycenie zasobów serwera. Objawami takiego ataku jest drastyczne zwolnienie pracy aplikacji. Ataki na poszczególne aplikacje można na bieżąco monitorować w serwisie internetowym Downdetector [87].

Proponując rozwiązanie ochronne, należy zadać sobie pytanie na temat sposobu ochrony przed atakami typu DDoS.

Jak ochronić sieci i systemy przed atakami DDoS ?

Odpowiedź nie jest taka oczywista, wynika to z faktu, iż skuteczna ochrona przed atakami sieciowymi DDoS wymaga wdrożenia solidnej polityki bezpieczeństwa, zwierającej wielu strategii ochrony, które to mogą podlegać różnym kryteriom klasyfikacji. Celem przedstawienia szczegółowo koncepcji rozwiązania STRAINER, autor postara się umieścić system w procesie bezpieczeństwa i wskazać jego obszar działania.

3.1.2. MIEJSCE SYSTEMU STRAINER NA „MAPIE DROGOWEJ" OCHRONY ANT-DDOS

Projektując systemy bezpieczeństwa sieciowego tj. anty-DDoS należy dokonać podziału i klasyfikacji metod obrony. Co w wyniku uwypukli obszar, w którym znajdzie się nasze rozwiązanie. Trzeba jasno powiedzieć, iż nie istnieją systemy, które w 100% ochronią sieci i infrastrukturę. Natomiast są lepsze lub gorsze strategie, polityki bezpieczeństwa a mówiąc wprost procedury i wytyczne, których przestrzeganie zwiększa w dużym stopniu nasze bezpieczeństwo i ryzyko ataku sieciowego.

Klasyfikacja metod obrony anty-DDoS, podzielna jest na mechanizmy prewencyjne i reaktywne. Projektowany system STRAINER zdecydowanie znajduje się w grupie mechanizmów reaktywnych.

Mechanizmy prewencyjne:

- Zapobieganie atakom – mechanizm ma na celu wprowadzenie adekwatnych zmian w konfiguracji systemów operacyjnych i urządzeń sieciowych, które maksymalnie eliminują możliwości ataku. W szczególności w obszarze bezpieczeństwa systemów operacyjnych należy dołożyć wszelkich starań i podjąć działania w zakresie:
    - Wdrożenia i monitorowania systemów automatycznej aktualizacji systemu i oprogramowania
    - Uruchomienie i skonfigurowanie wbudowanej zapory firewall
    - Bezwzględne objęcie systemu ochroną antywirusową
    - Monitorowanie i logowanie dostępu do hosta i jego kluczowych zasobów
    - Uruchomienie systemów IDS

  Wdrożenie powyższych zasad nie wątpliwie wpłynie na zwiększenie bezpieczeństwa hosta, a co za tym idzie zminimalizuje ryzyko wykorzystania go jako ofiary ataku DDoS (często wykorzystywane w atakach DDoS są podatności systemu wynikające z braku aktualnego oprogramowania co prowadzi do infekcji za pomocą wirusa). Powszechną praktyką jest również „przechwytywanie" przez hakera zainfekowanego komputera ofiary i wykorzystanie go do ataku DDoS w postaci członka sieci Botnet.

  W obszarze bezpieczeństwa protokołów powinniśmy dążyć do właściwej (bezpiecznej) konfiguracji, przestrzegania zasada dotyczących:
    - Strategii otwartych portów – tylko szyfrowanych
    - Wrażania protokołów typu proxy i VPN
    - Zabezpieczania ruchu protokołu „łatwo dostępnego" np. http za pomocą szyfrowania np. SSL (https)
    - Wyłącznie nie używanych lub nie bezpiecznych usług
- Zapobieganie DDoS przez przetrwanie – ta strategii opiera sią na próbie przeczekania ataku DDoS poprzez limitowanie zasobów i dublowanie zasobów. W pierwszym przypadku oszczędzamy zasoby limitując je szczegółowo dla

użytkowników, natomiast w drugiej opieram się na sporym zapasie i nadmiarowości zasobów.

Mechanizmy reaktywne:

Mechanizmy te w dużej mierz koncentrują się na ochronie naszych systemów poprzez:

- Wykrywanie ataków – działania polegają na jak najszybszym a zarazem skutecznym wykryciu anomalii a przy tym właściwej klasyfikacji na pozytyw i negatyw. Strategia wykrywania jest sklasyfikowana ze względu na sposób detekcji:
  - Na podstawie sygnatur
  - Na podstawie anomalii
  - W sposób hybrydowy
  - Na podstawie zewnętrznych sygnałów (Traceback)

  Wykrywanie na podstawie sygnatur opiera się na analizie i przyrównaniu ruchu versus dane posiadane w bazie wzorców ataku. Za pomocą tej metody jesteśmy w stanie wykryć tylko znane ataki.

  Wykrywanie ataków na podstawie anomalii – mechanizm ten opiera się na posiadanym modelu wzorcu ruchu „normalnego" w naszej sieci. Okresowo nasz wzorzec porównywany jest z ruchem bieżącym, określna jest średnia przepustowość czy obciążenie. Metoda ta umożliwia wykrywanie nowych nieznanych dotąd ataków, jednak metoda ta wymaga precyzyjnych nastaw i okresowej aktualizacji modelu wzorcowego. Wykrywanie hybrydowe – ta metoda łączy strategię opartą na sygnaturach i na wykrywaniu anomalii – jest to bardzo skuteczna metoda, ale jest w niej jeszcze dużo do dopracowania w zakresie jej pełnej automatyzacji. (jest przedmiotem badań autora pracy)

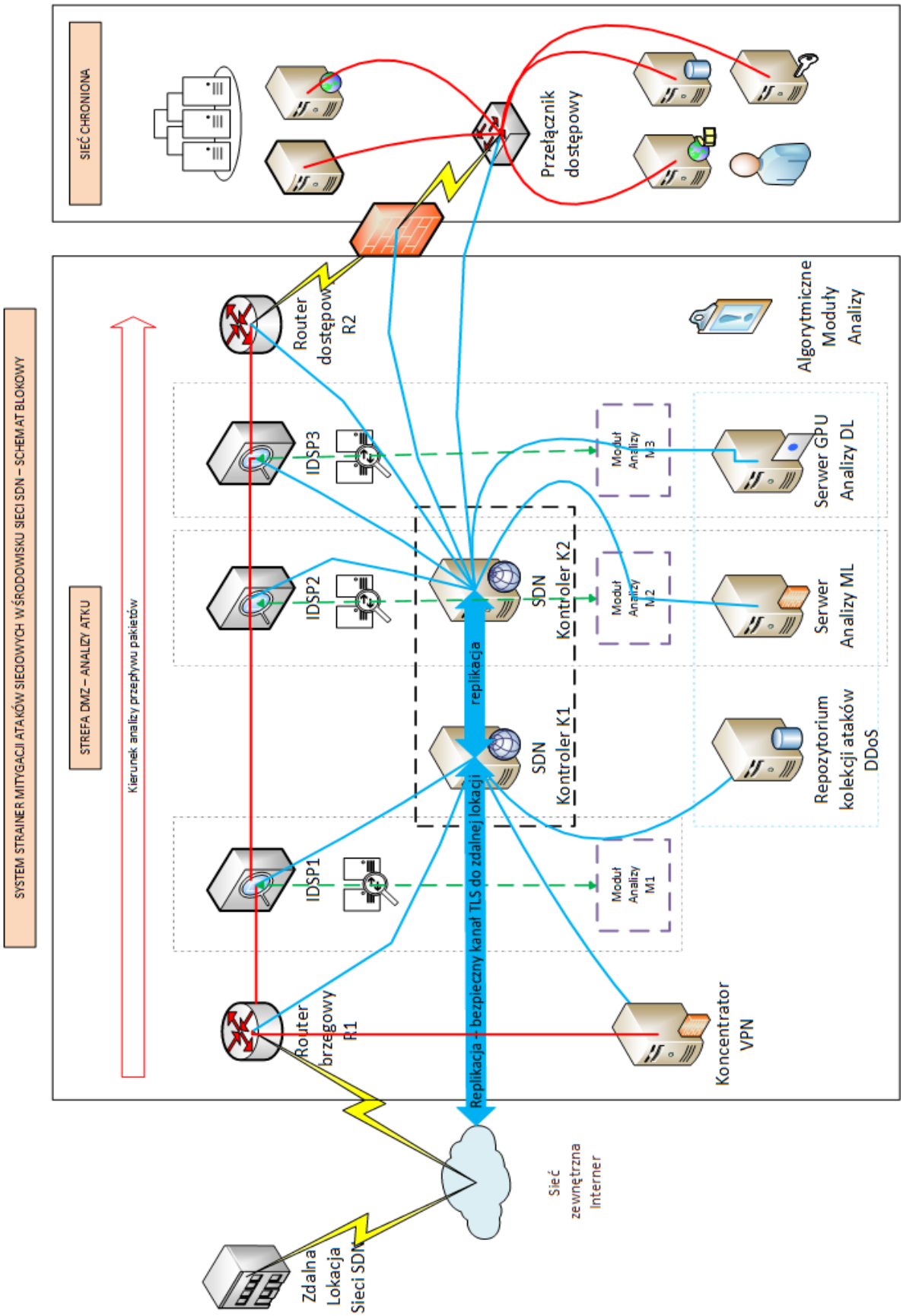
- Reakcja na atak – działania mające na celu zablokowanie lub rozproszenie wykrytego ataku
- Mechanizmy autonomiczne – to klasyczne IDS-y i firewall-e
- Mechanizmy kooperatywne – przykładem może być Pushback [88]
- Mechanizmy współzależne – wymagają dużego spektrum obserwacji

Podsumowując ten przegląd, wskazaniem obszaru działania systemu STRAINER jest to obszar mechanizmów reaktywnych, wykrywania ataków ze względu na wszystkie zaklasyfikowane sposoby detekcji. System posiada również funkcjonalności w obszarze reakcji na atak, i obszarze mechanizmów autonomicznych. Wynika to wprost z jego konstrukcji opartej o strukturę sieci SDN.

3.2. KONCEPCJA SYSTEMU MITYGACJI ATAKÓW DDOS

Projektując system STRAINER autor pracy dokonał wnikliwej analizy systemów ataków DDoS[84]. W pracy autor nie przedstawia całej historycznej już metodologii wykrywania ataków sieciowych DDoS w klasycznych sieciach komputerowych. Nie przedstawia również drobiazgowego opisu i klasyfikacji metod czy rodzajów i podziału ataków znanych już od ponad 20 lat. Nie zagłębia się również w szczegóły opisu protokołu TCP/IP, na który to ataki są wykonywane z narastająca skutecznością od ponad dwóch dekad.

System STRAINER posiad trzy reaktywne moduły detekcji anomalii sieciowych w kierunku ataków DDoS. Pierwszy moduł (MOD1) o bardzo szybkim działaniu detekcyjnym oparty o „lekkie" algorytmy wykrywania na podstawi bazy znanych sygnatur. Dwa pozostałe moduły detekcji to moduły z analizą opartą o mechanizm uczenia maszynowego ML i głębokiego uczenia maszynowego DL. Odpowiednio MOD2 i MOD3. System posiada również autorski sposób implementacji ww. modułów, adekwatnie co do wymagań wydajnościowych. Składową systemu detekcji jest baza kolekcji zebranych ataków sieciowych DDoS. W roli sond sieciowych występują systemy IDS, w różnych wariantach implementacji. Założenie projektowe i teza pracy opiera się na podstawowym warunku stworzenia systemu mitygacyjnego w środowisku sieci SDN. Centralnym punktem sterowania systemu jest logiczny Kontroler sieci SDN. Steruje ona całym systemem mitygacji za pomocą wbudowanych funkcjonalność własnych, protokołów tj. OpenFlow, NETCONF, TLS. Zarządza on bezpośrednio podłączonymi do niego modułami detekcji. Zbierając dane z całej domeny sieci SDN, posiada przez to, globalny widok całej chronionej sieci. Wszystkie urządzenia sieciowe w systemie są w standardzie sieci SDN. Zostają w ciągłym połączeniu z kontrolerami za pomocą ruchu sterującego. Posiadają funkcje monitorujące, detektujące, przekazywania ruchu, routingu, jak również walidacji określonych polityk wynikających z wstrzykiwanych reguł bezpośredni z tablicy przepływu (FT) kontrolera. System posiada również infrastrukturę dodatkową, w postaci serwerów z wyniesionym modułami (MOD3) detekcji opartymi o analizę z wykorzystaniem głębokiego uczenia maszynowego. Schemat blokowy systemu STRAINER znajduje się na rysunku 7. Na rysunku 7 widzimy elementy sytemu z podziałem na grupy modułów detekcji (MOD1 do MOD3). Sondy sieciowe IDS1 do IDS3. Logiczną instancję kontrolera SDN. Serwery pomocnicze, jak również routery brzegowe i dostępowe, odpowiednio dla łącza do sieci zewnętrznej (nie chronionej) Internet jak i do sieci chronionej intranet (firmowej). Kierunek przepływu analizy ruchu wchodzącego na rysunku jest przedstawiony od strony lewej do prawej. Kolorem czerwonym zaznaczony jest ruch produkcyjny pakietów, natomiast kolorem niebieskim ruch sterujący kontrolera SDN. Sieć chroniona znajduje się na schemacie blokowym z prawej strony jest zabezpieczona urządzeniem granicznym SDN w funkcji firewall-a. Przełączniki sieciowe w sieci chronionej są również podłączone za pomocą ruch kontrolnego (protokołu OpenFlow) do kontrolera. We wewnątrz sieci chronionej znajdują się typowe zasoby w postaci serwerów i hostów. System umożliwia również dostęp do sieci chronionej dla użytkowników zdalnych poprzez dostęp VPN. Na rysunku 7 koncentrator VPN wpięty jest do rutera brzegowego R1. Lokalizacja koncentratora VPN jest celowo umieszczona w strefie

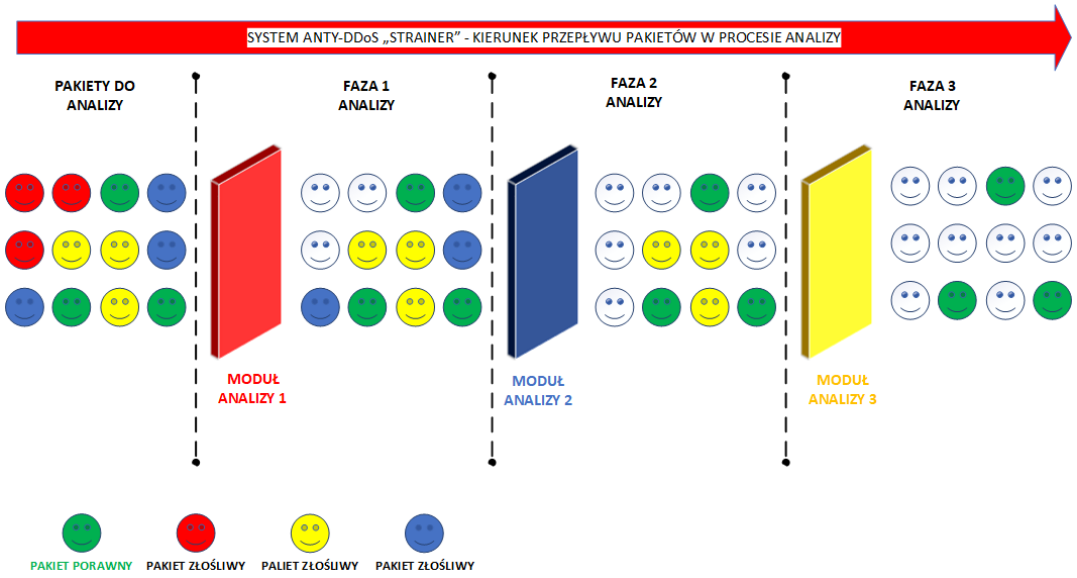


Rysunek 7. Schemat blokowy systemu mitygacji ataków sieciowych DDoS.

DMZ – analizy potencjalnego ataku. Wynika to z założenia ograniczonego zaufania do ruchu w kierunku koncentratora VPN z sieci zewnętrznej. Za pomocą koncentratora VPN możliwe jest również tunelowanie przefiltrowanego ruch do zewnętrznej, firmowej lokalizacji zdalnej. W tym przypadku prowadzona jest również polityka ograniczonego zaufania.

W praktyce przedstawiony system, może działać jak rozproszony – zdalny system ochrony przed atakami DDoS. Posiadając dowolny dostęp do Internetu, obojętnie od jakiego ISP, w dowolnym miejscu geograficznym, można podłączyć się do sytemu STRAINER celem ochrony własnego hosta, dostępu do sieci czy konkretnej aplikacji internetowej. W dalszej części pracy autor przedstawi kilka scenariuszy możliwego wdrożenia systemu. Autor prowadzi od wielu lata badania w kierunku tego typu systemów, jedną z propozycji rozproszonego systemu ochrony przedstawiła w pracy [84]. W swoich badawczych doświadczeniach przedstawił również propozycję sytemu opartego o sieci SDN z brzegowym przetwarzaniem (ang. *edge computing*, koncepcyjnie podobnego do działania systemów sieci piątej generacji 5G [85].

Siła całego systemu opiera się na algorytmicznym systemie reaktywnych detekcji ataków DDoS. System ten jest zbudowany w oparciu o funkcjonalne moduły analizy dla kontrolera SDN. Moduły odpowiednio MOD1 do MOD3, znajdują się w strefie DMZ detekcji ataku. Moduł MOD1 jest pierwszym filtrem sieciowym w systemie STRAINER, który na podstawie (szczegółowego opisanego w dalszej części pracy) algorytmu prowadzi wstępną analizę ruchu w kierunku wykrycia potencjalnego ataku DDoS. W przypadku nie wykrycia ataku, ruch sieciowy poddany jest analizie w MOD2, ta analiza opiera się na zmodyfikowanym algorytmie random forest (szczegółowo opisanym w dalszej części pracy) uczenia maszynowego (ML) wykrywania ataków z wykorzystaniem baz wzorców ataków z Kanadyjskiego Instytutu Badawczego. W przypadku negatywnego wyniku. Ruch przekazywany jest do ostatniego modułu reaktywnego detekcji MOD3. To moduł wykorzystujący algorytmy sztucznej inteligencji, głębokiego uczenia maszynowego, jego głównym atutem jest możliwość wykrywania nie znanych ataków sieciowych DDoS. Dokonuje tego w oparciu o porównanie modelu analizy „normalnego" ruch w sieci chronionej do bieżąco analizowanego ruchu pochodzącego z MOD2. Do dyspozycji posiad również uwspólnioną bazę kolekcji ataków wykrytych. Moduł ten jak i dwa poprzednie szczegółowo wraz z algorytmami będzie przedstawiony w dalszej części pracy. Na rysunku 8 przedstawiono poszczególne fazy analizy ruchu sieciowego. Analiza przepływających pakietów tak jak widać na schemacie odbywa się od strony lewej do prawej. Zadaniem systemu jest wyfiltrowanie złośliwego ruchu (pakietu) i przekazanie informacji na ten temat do kontrolera SDN. Kontroler na podstawie informacji z modułów podejmuje adekwatną reakcję i przekazuję ją do urządzeń sieciowych (router brzegowy, przełącznik czy firewall) celem mitygacji zagrożenia. W zależności od sytuacji, najczęstszą akacją jest dokonanie aktualizacji reguł w tabelach przepływu (FT) za pomocą protokołu OpenFlow w całej sieci SDN, możliwe jest to dzięki globalnemu widokowi sieci z pozycji kontrolera. Informacja o zaklasyfikowanym zagrożeniu wraz z określona polityką bezpieczeństwa może być przekazana za pomocą mechanizmu replikacji kontrolera z wykorzystaniem kanału TLS do zdalnej lokalizacji sieci firmowej. Zgodnie z koncepcja na rysunku 7.



Rysunek 8. Fazy analizy ruchu sieciowego w systemie detekcji aktów DDoS.

Przedstawiony schemat blokowy systemu (rysunek 7), jego budowa modułowa, jak również fazy analizy pakietów (rysunek 8) i kolejności działania (analizy) prezentują ogólny zarys działania mechanizmów i funkcjonalności. W dalszej części zostanie przedstawiona architektura sieciowa systemu, szczegółowy opis jego składowych. W szczególności opis poszczególnych modułów analizy wraz z zastosowanymi rozwiązaniami algorytmicznymi. Zostaną również przedstawione możliwe scenariusze wdrożenia systemu.

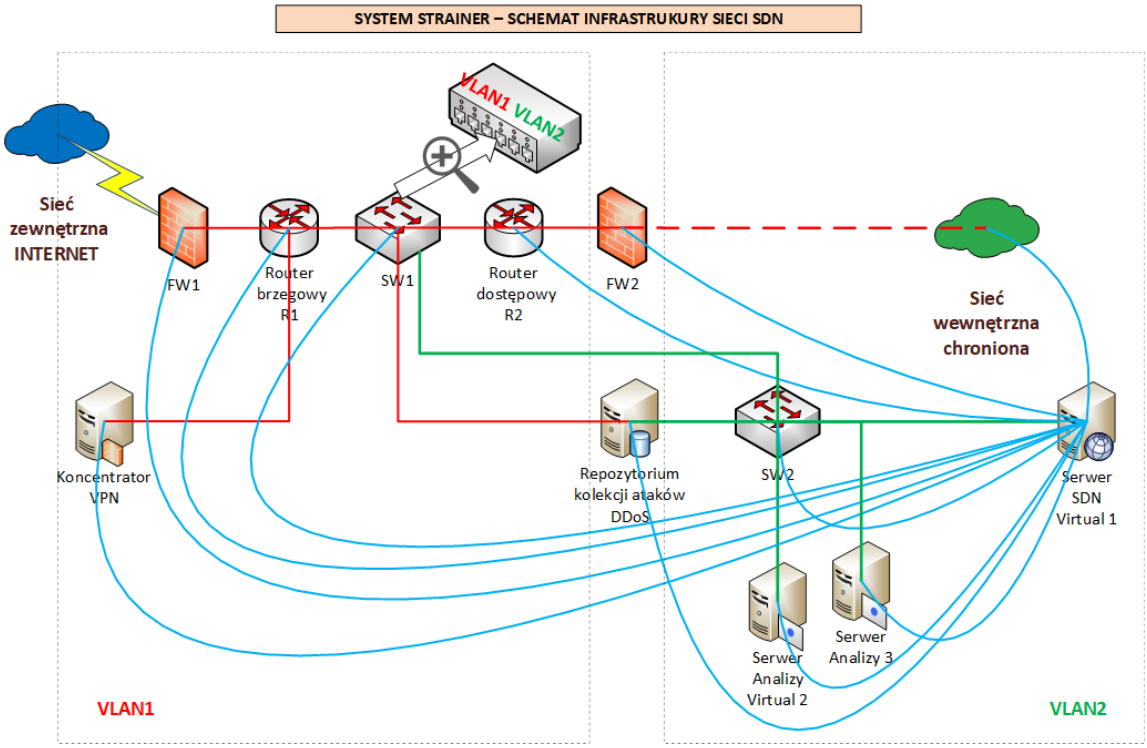
3.3. OPIS ARCHITEKTURY SYSTEMU

Architektura systemu STRAINER opiera się na trójwarstwowej detekcji ataku opartej na algorytmicznym systemie modułów. W tym rozdziale przedstawiona zostanie architektura sieciowa systemu wraz z opisem implementacji sprzętowo-systemowej. Na rysunku 9 przedstawiono schemat sieciowy architektury systemu. Analizując schemat sieciowy wyróżniamy w nim:

- Urządzenia sieciowe SDN
  - Router brzegowy R1
  - Router dostępowy R2
  - Zaporę sieciową FW1
  - Zaporę sieciową FW2
  - Przełącznik sieciowy (L2/L3) SW1
  - Przełącznik sieciowy (L2/L3) SW2
  - Koncentrator VPN (implementacja: wirtualna maszyna)
- Serwery funkcyjne
  - Serwer SDN Virtual 1 (zawiera obrazy wirtualne kontrolerów SDN)

- Serwer Analizy Virtual 2 (zawiera obrazy wirtualne mechanizmów funkcjonalnych MOD1 i MOD2)
- Serwer Analizy 3 (zawiera mechanizmy MOD3)
- Serwer Repozytorium kolekcji ataków DDoS (zawiera przechwycony ruch do analizy z przełącznika SW1, repozytorium zebranych ataków, wzorcowe bazy danych z Kanadyjskiego Instytutu Cyberbezpieczeństwa)

Kolorem czerwonym na schemacie został zaznaczony ruch sieciowy „produkcyjny" niezabezpieczony (VLAN1). Kolorem czerwonym przerywanym został zaznaczony ruch sieciowy „produkcyjny" zabezpieczony. Kolorem zielonym został zaznaczony ruch sieciowy „sterujący" w wydzielonej sieci kontrolnej środowiska STRAINER (VLAN2). Niebieskim kolorem znaczony jest ideowo ruch kontrolny SDN protokołu np. OpenFlow.



Rysunek 9. Schemat sieciowy architektury systemu STRAINER

Analizując zastosowane urządzenia sieciowe SDN w ruchu produkcyjnym zastosowano dwa routery sieciowe R1 i R2. Odpowiednio router R1 jest routerem brzegowym (granicznym), realizującym funkcję bramy sieciowej (ang. *gateway*) do sieci zewnętrznej Internet. Realizuję on funkcję routingu w kierunku sieci WAN. R1 jest również jednym z urządzeń walidacji polityk bezpieczeństwa implementowanych w (FT) tabelach przepływu przez kontroler SDN, wypracowanych za pomocą systemu detekcji STRAINER w modułach algorytmicznych. Router R2 to urządzenie na potrzeby routingu w sieci LAN. Realizuje routing między VLAN-ami. Zaimplementowany jest w sieci w modelu „router on the stick" (ang. *router on the stick* – router na patyku). Dostarcza usługę DHCP dla urządzeń sieci LAN. Obsługuję przefiltrowany ruch produkcyjny jak również wewnętrzny ruch sterujący. R2 jest

elementem sieci SDN, który podobnie jak R1 może walidować polityki bezpieczeństwa wypracowane przez kontroler SDN.

Zapora sieciowa FW1 jest zaimplementowana na wejściu do sieci DMZ jeszcze przed routerem R1 w trybie transparentnym. Stoi na pierwszej „linii ognia", jej głównym zadaniem jest blokowanie złośliwego ruchu wchodzącego do sieci, na podstawie polityki bezpieczeństwa otrzymanej z kontrolera SDN. FW1 poprzez sposób implementacji na interfejsie wejściowym przed ruterem brzegowym, ma możliwość przechwycenia ataku DDoS jako pierwsze urządzenie reaktywne. Fakt wdrożenia go w trybie transparentnym powoduje, iż urządzenie nie jest celem ataku jak np. może nim być router R1. Mitygacja w FW1 ataku w pierwszej kolejności, w znaczący sposób odciąża (a tym samy chroni) router R1 przed skutkami DDoS w jego kierunku. Działanie FW1, może być lekko opóźnione przez fakt, iż w pierwszej fazie wykrycia ataku „złośliwego ruchu" musi on zostać dopuszczony do urządzenia SW1, z którego to za pomocą funkcji mirroringu portu, ruch zostaje przekazany do kolektora ruchu w serwerze repozytorium kolekcji ataków. Co oznacza, iż „złośliwy ruch" musi przejść przez router R1, zanim zostanie wykryty. Może to powodować „chwilowe" obciążenie routera R1 obsługą ataku DDoS. Jednak, od razu po wykryciu ataku, złośliwy ruch jest mitygowany przez FW1 jeszcze przed routerem R1. Takie rozwiązanie skutecznie chroni strategiczne urządzenie brzegowe R1.

Zapora sieciowa FW2 jest urządzaniem reaktywnym wykonawczym zlokalizowanym w sieci produkcyjnej na granicy ruchu przeanalizowanego („czystego") w sieci chronionej i przychodzącego ruch produkcyjnego z przełącznika SW1 (potencjalnie niebezpiecznego). Prowadzi pełną filtracje ruch na wszystkich warstwach modelu OSI w tym w warstwie aplikacji (L2, L3, L4, L7). Jej funkcja i miejsce implementacji dobrana jest w taki sposób, aby w przypadku ataku DDoS, po uzyskaniu odpowiedniej polityki sterującej z kontrolera SDN w skuteczny sposób mitygować atak na wszystkich możliwych warstwach i protokołach TCP/IP celem ochrony sieci wewnętrznej przed atakiem. Jej głównym zdaniem jest walidacji w warstwie aplikacji (L7). Jest w pełni funkcjonalnym UTM-em z zaimplementowaną ochroną sieci w tym ochroną anty-wirusową.

Przełącznik sieciowy SW1 jest zaimplementowany w funkcji przełączania (L2) z pełnym wsparciem dla obsługi SDN, obsługą (FT) tabel przepływu. Ma skonfigurowane VLAN w tym VLAN1 z ruchem produkcyjnym i VLAN2 z ruchem sterującym. Dodatkowo SW1 cały przychodzący ruch sieciowy z interfejsu sieciowego g01/1 ze strony routera R1 kopiuje funkcją „mirroringu portu" na interfejs g24/1 do którego wpięty jest kolektor ruchu przychodzącego, serwera repozytorium ataków. SW1 posiada również interfejsy w VLAN2 w sieci sterującej za pomocą której komunikuje się z kontrolerem SDN. Dodatkowo przełącznik SW1 posiad interfejs „trunk" do którego wpięty jest router R2 celem prowadzenia routingu między VLAN jak również dodatkowej filtracji w warstwie L3. SW1 spięty jest za pomocą Ethernetu (L2) z SW2. Przełącznik SW2 jest przełącznikiem sieci wewnętrznej (sterującej). Skonfigurowany jest do obsługi typu access dla warstwy L2. Wszystkie interfejsy posiada w VLAN2. Posiada pełne wsparcie dla obsługi SDN, i (FT) tabel przepływu OpenFlow.

Koncentrator VPN jest dedykowanym rozwiązaniem serwera VPN. Pracuje w trybach site to site, jak i klient serwer. Jego głównym zdaniem jest tunelowanie ruch, zestawianie sesji dla zdalnych klientów (mobilnych użytkowników) celem dostępu do sieci chronionej (korporacyjnej). Jak również zestawianiem stałych kałów transmisji VPN site to site, celem dołączenia zdalnych lokalizacji firmowych. Jego sposób podłączenia jest również celowy, a mianowicie dopięty jest w części sieci DMZ, bezpośrednio do interfejsu routera R1. Ruch z koncentratora VPN traktowany jest jako potencjalnie

niebezpieczny. Jego implementacja, to dedykowany serwer sprzętowy z rozwiązaniem w architekturze HA z osadzonymi w systemach operacyjnych usługami VPN. Serwer ukierunkowany jest na wydajność dla tunelowania i szyfrowania sesji VPN. Kontrolowany jest z poziomu kontrolera SDN.

Serwer SDN Virtual 1 to serwer hypervisor, zawierający w sobie 3 zestawy maszyn wirtualnych. Są to odpowiednio instancje w trybie HA maszyn wirtualnych z osadzonymi kontrolerami SDN w różnych implementacjach. Razem jest to 6 maszyn wirtualnych z oprogramowaniem kontrolerów SDN i modułami towarzyszącymi. Pierwszy zestaw zawiera rozwiązanie kontrolera SDN w implementacji Ryu. Drugi to implementacja OpenDaylight. Trzeci zestaw implementacja kontrolerów POX.

Serwer Analizy Virtual 2 to dedykowane wirtualne środowisko dla analizy ruch pod katem wykrycia ataku DDoS. Serwer zawiera maszyny wirtualne z systemami operacyjnymi i osadzonymi w nich wyniesionymi modułami kontrolera SDN. Odpowiednio pierwsza maszyna wirtualna zawiera w sobie środowisko modułu decyzyjnego MODSAI (SAI - System Informacji o Atakach) sterującego pracą całego podsystemu decyzyjnego. Druga maszyna wirtualna zawiera w sobie środowisko z MOD1 (wyniesiony moduł kontrolera SDN) dla rozwiązania analizy opartej na entropii ruchu sieciowego. Trzecia maszyna wirtualna zawiera środowisko dla MOD2 – moduł uczenia maszynowego (ML) opartego o zmodyfikowany algorytm random forest.

Serwer Analizy 3 jest znacznie bardziej wymagający pod kątem zasobów systemowych. Jest serwerem dwu-procesorowy, wyposażony w 1 TB pamięci operacyjnej RAM i trzy karty GPU do akceleracji trybu głębokiego uczenia maszynowego. Tworzy platformę dla podsystemu MOD3 zawierającego rozwiązanie głębokiego uczenia maszynowego DL.

Serwer Repozytorium kolekcji ataków DDoS to serwer pełniący kilka kluczowych funkcji systemu. To środowisko dla modułu kolekcjonera ruchu sieciowego MODKOL. Przechowuję kopię przechwyconego online ruch produkcyjnego do analizy, pochodzącego bezpośrednio z przełącznika SW1. Ruch ten uzyskiwany jest za pomocą funkcji mirroringu portu i składowany na dyskach w serwerze. Ruch ten poprzez systemy IDS poddawany jest analizie i przekazywany do modułów MOD2 i MOD3 celem wykrycia potencjalnych ataków. Dodatkowo na serwerze znajduje się repozytorium zebranych i wykrytych ataków typu DDoS. Jest również miejscem do składowania, wzorcowego „czystego", typowego ruch z sieci chronionej służącego do analizy i porównania celem wykrycia nieznanego typu ataku DDoS w MOD3. Serwer zawiera również aktualne bazy wzorcowe danych o atakach DDoS, uzyskane jako zbiory wzorcowe dla MOD2 pochodzące z Kanadyjskiego Instytutu Cyberbezpieczeństwa. Serwer posiada interfejsy sieciowe w sieci produkcyjnej (VLAN1) jak i w sieci sterującej (VLAN2). Posiada dodatkowe zabezpieczenie przed forwardowaniem pakietów z VLAN1 do VLAN2 (potencjalnie niebezpieczne miejsce wymiany ruchu pomiędzy siecią produkcyjną a sterującą - wewnętrzną). Sieć sterująca oparta jest na adresacji wewnętrznej nieroutowanej wszystkie mechanizmy typu NAT są wyłączone.

3.4. MODUŁY SYSTEMU STRAINER

System STRAINER podzielony jest w części analizy ataków DDoS, na moduły funkcjonalne dla kontrolera SDN. Ze względu na swoją funkcję i sposób wykrywania ataków DDoS wszystkie moduły posiadają odmienną implementację. Wszystkie te moduły posiadają autorskie rozwiązania wdrożenia i algorytmiczne. Moduły odpowiednio realizują funkcję wykrywania poprzez:

- MOD1 – IDS_EBEDM działa w oparciu o zasadę entropii ruchu sieciowego
- MOD2 – moduł uczenia maszynowego, RFoRF działa w oparciu o zmodyfikowany algorytm Random Forest.
- MOD3 – moduł głębokiego uczenia maszynowego, IDS_GRUoGPU działa w oparciu o neuronowe sieci bramkowe GRU.
- MODSAI – moduł decyzyjny systemu informacji o atakach, informuje kontroler SDN o wykrytym ataku.
- MODKOL - moduł kolekcjonera ruchu sieciowego – zbiera dane o ruchu sieciowym wybranymi metodami.

3.4.1. MODUŁ MOD1 (ENTROPIA RUCHU SIECIOWEGO)

Moduł MOD1 o nazwie IDS_EBEDM swoją zasadę działania opiera na pojęciu Entropii w rozumieniu *teorii informacji*.

Autor zamodelował i zaimplementował w rozwiązaniu problemu statystyczny algorytm wyliczania prawdopodobieństwa wystąpienia ataku DDoS, w oparciu o pojęcie Entropii na analizowanym ruchu sieciowym przychodzącym do chronionej sieci komputerowej.

Entropia jest to pewna średnia ilość informacji przypadająca na pojedynczą wiadomość ze źródła informacji. Nazywana jest również średnią ważoną. [89]

Entropia to miara nieokreśloności. W teorii systemów jest miarą uporządkowania - im mniejsza tym system jest bardziej uporządkowany, a wraz ze wzrostem spada jego uporządkowanie (najwyższy poziom świadczy o chaosie). Zerowa entropia oznacza system doskonale uporządkowany. W takim systemie można przewidzieć zdarzenia z prawdopodobieństwem równym jedności. W momencie w którym oczekiwane zdarzenie ma miejsce, zmniejsza się poziom niewiedzy i niesie to pewną informację. Informacja wypiera tutaj pewien poziom entropii. Dzięki temu można uznać entropię za miarę informacji.[88]

Wartość entropii wyrazić można wzorem:

$$H(p) = -\sum_{i=1}^{n} p_i log p_i \quad (1)$$

gdzie, H(p) - to nieokreśloność zajścia zdarzenia

p – prawdopodobieństwo zajścia niezależnego zdarzenia [88]

Model wykrywania ataku DDoS.

Aby dokonać entropi ruchu sieciowego przychodzącego, wykorzystano funkcjonalność sieci SDN polegającej na globalnym widoku sieci. Dokładniej wykorzystano, fakt iż kontroler SDN posiada szczegółową informacje o przepływie danych w urządzeniach sieci. Informację tę, kontroler SDN uzyskuje za pomocą protokołu OpenFlow. Konkretnie w tym zastosowaniu, entropii ruchu sieciowego celem wykrycia ataku DDoS, agregujemy informacje z kontrolera SDN dotyczące, ruchu przychodzącego na określone *IP_DESTINATION* (adres IP docelowy) hosta (atakowanego), znajdującego się w sieci chronionej. Informacja *IP_DESTINATION* znajduje się w ramce ruchu obsługiwanej przez tabelę przepływu (FT) w przełączniku sieciowym SW1. Przedstawiony poniżej algorytm pokazuje metodę detekcji ataku DDoS, na podstawie uzyskanej wartości entropii w stosunku do poziomu odniesienia - alarmu.

Parametrem atrybutu agregacji przepływu mogą być wartości z pól ramek przekazywanych z wykorzystaniem (FT) tablicy przepływu w przełączniku SDN. W badanym przypadku autor założył iż, atak DDoS odbywa się na więcej niż jednego hosta w chronionej sieci. W związku z tym agregacja danych z kontrolera odbywa się na atrybucie *IP_DESTINATION.* Alternatywnie w przypadku ataku DDoS na określoną aplikacje a nie hosta, użyto by atrybutu *DST_PORT*. Wartości agregacji można płynnie zmieniać w zależności od zdefiniowanego sposobu wykrywania.

Implementując funkcje entropii do modelu wykrywania w oparciu o *IP_DESTINATION,* dla aktywnych adresów lokalnych IP (działających w sieci hostów) można określić liczbę pakietów do nich wysyłanych w interwale czasu T w następujący sposób:

$$X_i = \sum_{m=1}^{l} N_{IF_m}[IP_{DEST} = IP_i] \tag{2}$$

gdzie,

Xi – Liczba pakietów wysyłanych do aktywnego hosta o adresie IPi podczas interwału T

N - Liczba aktywnych lokalnych adresów IP podczas interwału T

IF – przepływ wejściowy do przełącznika (np. o *IP_DESTINATION, DST_PORT*)

Otrzymujemy $x = \{X_1, X_2, X_3, \dots, X_n\}$ jako liczba przychodzących pakietów wysłanych do każdego aktywnego adresu IP.

Przy obliczeniu wartości entropii, używamy częstotliwości każdego adresu IP aby z pewnym przybliżeniem oszacować jego prawdopodobieństwo (P).

$$P_i = \frac{X_i}{\sum_{i=1}^{N} X_i} \tag{3}$$

gdzie,

Pi – prawdopodobieństwo lokalnego adresu IP, IPi w czasie T

Możemy otrzymać rozkład prawdopodobieństwa lokalnego adresu IP, $P = \{P_{1,}P_{2,}P_{3,\dots,}P_n\}$. A więc dla przełącznika SW1 obliczmy wartość entropi z wzoru:

$$H(SW1_j) = -\sum_{i=1}^{N} P_i log P_i \tag{4}$$

gdzie,

$H(SW1_j)$ $-$ $wartość\ entropii\ przełacznika\ SW1$

Podsumowując zgodnie z założeniami przy N>1, w oparciu o charakterystykę funkcji entropii przyjmujemy wartość pomiędzy [0,1], przez znormalizowanie jak poniżej:

$$H'(SW1_j) = \frac{H(SW1_j)}{logN} \tag{5}$$

Ustalenie ruchu anomalnego

Autor przyjął oznaczenie $H_N(SW1_j)$ jako oznaczenie stanu entropii podczas normalnego ruchu sieciowego, a $H_A(SW1_j)$ jako oznaczenie stanu entropii podczas ataku. Ruch sieciowy jest względnie stabilny w sytuacji bez ataku. Przyjmując oznaczenie $Ś(SW1_j)$ jako średnią wartość entropii przy normalnym ruchu oraz wiedząc, iż w trakcie ataku liczba pakietów mających to samo IP_{DEST} gwałtownie wzrośnie, powodując iż wartość $H_A(SW1_j)$ zacznie gwałtownie spadać. Dla ustalenia wartości ataku przyjmujemy A oznaczoną jako:

$$Ś(SW1_j) - H_A(SW1_j) > A \tag{6}$$

Biorąc jednak pod uwagę, iż wzmożony normalny ruch sieciowy może spowodować taki sam rezultat dla uniknięcia fałszywego rozpoznania autor przyjął założenie iż jeśli co najmniej M razy w ostatnim przedziale W razy ΔT, $H(SW_j)$ spełni powyższe równanie, będzie można przyjąć iż jest to narastający atak DDoS (biorąc pod uwagę jego ciągłość).

Ponieważ systemy atakujące tworzą ruch ataku za pomocą skryptów, podobieństwo między nimi jest silniejsze niż między „normalnym" ruchem sieciowym. Entropia ma potencjał do oszacowania podobieństwa ruchu. Im bardziej podobny jest ruch, tym mniejsza jest entropia. Ze względu na podobieństwo ruchu ataku, entropia informacyjna niektórych cech będzie odbiegać od normalnej. Na podstawie powyższej dyskusji zostanie obliczona entropia cech ruchu każdego adresu IP przełącznika. Gdy wartość ta znacznie różni się od wartości normalnego ruchu, ustala się, że ruch ataku kierowany jest do sieci na ten właśnie adres IP.

Również ze względu na fakt, iż wartość przepływu w sieci jest zmienna w czasie potrzebna jest adaptacyjność wykrywania anomalii (nie można przyjąć stałej wartości). Zarówno wskaźnik średniej entropii $\acute{S}(SW1_j)$, jak i wartość ataku A nie będą używać stałych wartości. Jeśli poprzednio obliczona wartość entropii nie spełnia formuły (6), zostaje ona przyjęta jako kolejna wartość ruchu normalnego.

W tym celu skorzystano ze średniej ważonej dla obliczenia wartości średniej ruchu normalnego $H_N(SW1_j)$dla następnych U upływów czasu ΔT:

$$\acute{S}(SW1_j) = \sum_{i=1}^{k} w_i \cdot H_N(s_j)[i] \tag{7}$$

$$\sum_{i=1}^{k} w_i = 1 \tag{8}$$

Dla określenia najbliższej wartości entropii należy dobrać współczynniki ważenia $w_i < w_j$,

dla i < j. Są one stałe, ale mogą zostać zmienione przez kontroler SDN.

Odchylenie standardowe dla tych wartości $H_N(SW1_j)$ obliczamy ze wzoru:

$$\sigma = \sqrt{\frac{1}{k}\sum_{r-1}^{k}(H_N(SW1_j)[i] - \acute{S}(SW1_j))^2} \tag{9}$$

Następnie ustawiamy próg ataku AT w następujący sposób:

$$AT = \lambda\sigma \tag{10}$$

gdzie λ jest współczynnikiem mnożenia. Wartości początkowe Ś(SW1j) oraz A są obliczane na starcie działania modułu.

Implementacja testowa

Moduł EBEDM_01 zaimplementowany został w wirtualnym przełączniku Open vSwitch v2.16.0 na wirtualnej maszynie z Linux Ubuntu 20.04.1 w postaci skryptu w języku Python w wersji 3.9.7 wraz z bibliotekami ovsdbapp oraz ovs, służącymi m.in. do obsługi protokołu OVSDB komunikacji ze vSwitchem (wirtualnym przełącznikiem).

Algorytm wykrywania ataku DDoS dla docelowych adresów IP

Algorytm ten monitoruje dane statystyczne dotyczące przepływów w tabeli przepływów kontrolera SDN, następnie wykonuje analizę danych i po stwierdzeniu podejrzenia ataku wysyła informację do kontrolera SDN.

Krok 1: Ustal początkowe wartości parametrów U,ΔT, A, λ, O, M. Oblicz początkowe wartości Ś($SW1_j$) dla poszczególnych adresów IP

Krok2: Oblicz prawdopodobieństwo wystąpienia docelowych adresów IP (*IP_DESTINATION)*

Krok 3: Oblicz entropię H($SW1_j$)

Krok 4: Jeśli jest spełnione równanie (6) oraz jeśli występuje M razy w oknie czasowym O wyślij informację o podejrzeniu ataku

Krok 5:w przeciwnym razie oblicz nowe wartości korzystając ze wzorów (8),(9),(10)

Krok 6: wróć do kroku 2

Entropia informacyjna zaproponowana przez Shannon to koncepcja teorii informacji[90]. Mechanizm detekcji oparty na entropii ma stosunkowo niskie obciążenie obliczeniowe. Entropia służy do pomiaru zmiany losowości przepływów przychodzących w danym okresie czasu. Jako mechanizm wykrywania anomalii oparty na przepływie ruchu sieciowego, autor zaimplementował go z wykorzystaniem funkcjonalności sieci SDN (sterowania przepływem), aby obliczać wartość entropii ruchu sieciowego. Ze względu na niskie obciążenie obliczeniowe jest to moduł fazy „lekkiej" wykrywania i jest w pierwszej linii obrony proponowanego systemu.

Zastosowana przez autora entropia jako metody wykrywania ataków DDoS w module MOD1, umożliwia implementację szybkiej metody wykrywania ataków DDoS o znanym charakterze jego przebiegu. Dzięki wykorzystaniu właściwości kontrolera SDN możliwe jest agregowanie monitorowanych parametrów i ich zmienianie w procesie działania modułu. Daje to możliwości wykrywania szerszego spektrum ataków DDoS poprzez dynamiczną zmianę agregowanych parametrów.

Wyniki skuteczności działania MOD1 zostaną przedstawione w dalszej części pracy.

3.4.2. MODUŁ MOD2 (RANDOM FOREST O RANDOM FOREST)

Moduł MOD2 o nazwie RFoRF swoją zasadę działania opiera na nadzorowanej metodzie uczenia maszynowego. Autor zaimplementował algorytm lasu losowego (Random Forest), modyfikując go autorsko do postaci lasu losowego lasów losowych.

Metoda Random Forest została zaproponowana przez Leo Breimana i Adele Cutler [91] jako algorytm uczenia maszynowego łączący wyniki wielu drzew decyzyjnych, by osiągnąć jeden wynik.

Las losowy składa się z wielu drzew decyzyjnych, w tej części pokrótce zostanie opisany algorytm drzewa decyzyjnego.

Drzewa decyzyjne wykorzystywane są dla klasyfikacji, gdy przewidywanym wynikiem jest dyskretna klasa do której należą dane. W przypadku regresji przewidywany wynik jest liczbą rzeczywistą. Opis drzew decyzyjnych dla klasyfikacji i regresji został opisany przez Leo Breimana i innych w 1984 roku [92].

Dla drzew decyzyjnych zestaw danych jest dzielony na podzbiór treningowy oraz testowy.

Drzewa decyzyjne starają się znaleźć najlepszy podział dla podzbioru danych i zazwyczaj trenowane są za pomocą algorytmów CART[92], ID3, C4.5, C5.0.[95] Do oceny jakości podziału stosowane są metryki, taki jak przyrost informacji, średni błąd kwadratowy, zanieczyszczenie (inpurity) Gini. Użycie binarnych lub wielokrotnych podziałów i kryteriów reguł podziału różnicują te algorytmy. Przyjęcie kryteriów reguł podziału jest jedną z ważniejszych części przy stosowaniu drzew decyzyjnych.

Algorytm CART do klasyfikacji:

Krok 1: Zacznij od węzła głównego ze wszystkimi wystąpieniami treningowymi

Krok 2: Wybierz atrybut na podstawie kryteriów podziału

Krok 3: Rekurencyjne partycjonowanie wystąpień zgodnie z wybranym atrybutem

Partycjonowanie zatrzymuje się gdy:

- nie ma już przykładów,
- wszystkie przykłady dla danego węzła należą do tej samej klasy
- nie ma pozostałych atrybutów do dalszego partycjonowania

Model drzewa klasyfikacji dzieli rekurencyjnie przestrzeń obiektów w taki sposób, że ostatecznie każdy region jest zdominowany przez jedną klasę.

Metryką używaną w CART do pomiaru zanieczyszczeń jest zanieczyszczenie Gini, osiąga swoje minimum gdy wszystkie przypadki w węźle należą do jednej kategorii docelowej.

Dla zestawu danych w klasie **A**, przy założeniu iż $j \in \{1,2,\dots,A\}$ oraz p_j jest ułamkiem pozycji oznaczonych klasą **j** w zestawie, obliczenia zanieczyszczenia Gini dokonamy następująco:

$$I_{GINI}(p) = \sum_{j=1}^{A}\left(p_j \sum_{k\neq j} p_k\right) = \sum_{j=1}^{A} p_j(1-p_j) = \sum_{j=1}^{A}(p_j - p_j^2) = \sum_{j=1}^{A} p_j - \sum_{j=1}^{A} p_j^2$$
$$= 1 - \sum_{j=1}^{A} p_j^2$$

(11)

Algorytmy ID3 oraz C4.5 wykorzystują entropię Shannona.

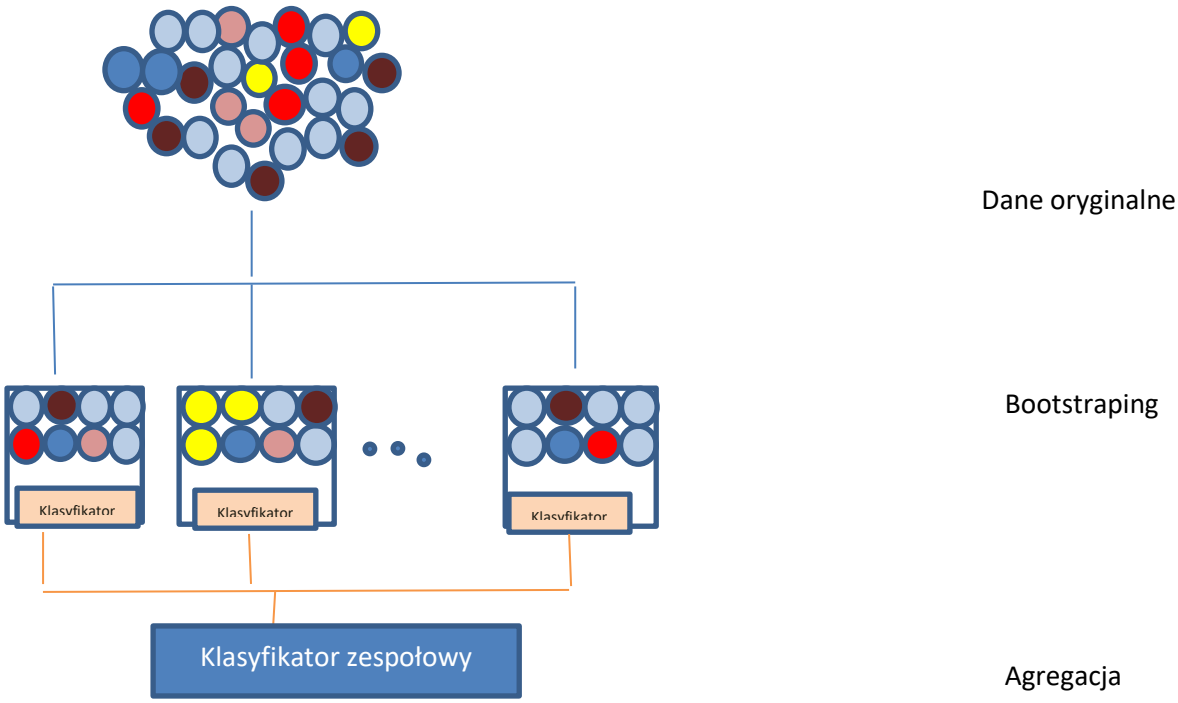
$$-\sum_{k=1}^{k} p_k \, log(p_k)$$

(12)

Kryteria te preferują bardziej „czyste” węzły, więc jest bardziej prawdopodobne iż wyróżnią zestaw czystego węzła końcowego.

Las losowy (Random Forest) to nadzorowany algorytm uczenia maszynowego. Jest on szeroko stosowany w problemach regresji i klasyfikacji. Jedną z ważniejszych cech tego algorytmu jest fakt, iż może on obsługiwać zestaw danych zawierający zmienne ciągłe i zmienne kategoryczne. Osiąga lepsze wyniki w przypadku problemów klasyfikacji. Jest to rodzaj nadzorowanego algorytmu klasyfikacji i reprezentuje zbiór losowo wybranych drzew decyzyjnych (Decision Tree).

Lasy losowe charakteryzuje technika zespołowa używana przez nie do doboru danych. Jest to tzw. pakowanie w worki (Bagging) – wybiera losową próbkę ze zbioru danych. Model jest generowany z próbek (próbki Bootstrap) dostarczonych przez oryginalne dane z zastąpieniem znanym jako próbkowanie wierszy – ten etap próbkowania wierszy z wymianą nazywamy Bootstrap. Każdy z takich modeli jest trenowany niezależnie, generując wyniki. Ostateczny wynik opiera się na głosowaniu większościowym w przypadku problemów klasyfikacji lub na uśrednianiu w przypadku problemów regresji. Łączenie wszystkich wyników i generowanie wyników finalnych nazywamy agregacją.



Rysunek 10. Schemat ideowy pakowania w worki (Bagging = Bootstrap Aggregation)

Algorytm lasu losowego:

Krok 1: Liczba n losowych rekordów jest pobierana ze zbioru danych o k liczbie rekordów

Krok 2: Dla każdej próbki konstruowane są indywidualne drzewa decyzyjne

Krok 3: Każde drzewo decyzyjne generuje dane wyjściowe

Krok 4: Ostateczny wynik jest osiągany na podstawie głosowania większościowego dla problemów klasyfikacji lub uśredniania dla problemów regresji.

Do ważnych cech lasu losowego należą m.in.:

1. Różnorodność – nie wszystkie cechy (zmienne, atrybuty) są brane pod uwagę
2. Odporność na przeuczenie
3. Dokładniejsze odtworzenie złożonych zależności niż w przypadku drzew decyzyjnych

Lasy losowe są skutecznym narzędziem przewidywania, a użycie odpowiedniego rodzaju losowości czyni je dokładnymi klasyfikatorami i regresorami. W klasyfikacji przypadkowe dane wejściowe i losowe cechy dają lepsze wyniki w klasyfikacji, niż w przypadkach regresji. Lasy losowe cechuje domyślna zdolność do korygowania nawyku drzew decyzyjnych w zakresie nadmiernego dostosowywania się do danych zestawu treningowego. Użycie metody pakowania w worki oraz losowego wyboru funkcji prawie całkowicie rozwiązuje problem nadmiernego doposażenia (który prowadzi do niedokładnych wyników). Lasy losowe są

również bardziej wydajne niż pojedyncze drzewa decyzyjne podczas wykonywania analizy dużych zbiorów danych. Również brak pewnych danych nie wpływa na dokładność przewidywania. Metoda może obsługiwać tysiące zmiennych wejściowych bez ich usuwania. Właśnie te cechy skłoniły Autora do zaimplementowania lasów losowych jako metody w jednym z modułów systemu.

Model wykrywania ataku DDoS:

Analiza danych odbywa się w oparciu o klasyfikację bieżącego ruchu sieciowego produkcyjnego. Cały badany ruch sieciowy pobrany jest z MODKOL (modułu kolekcjonera) za pomocą narzędzie TCPDUMP. MOD2 klasyfikuje ruch sieciowy w oparciu metodę RFoRF (las losowy lasów losowych) na podstawie opisanych poniżej zbiorów danych treningowych CIC. Poprzez zestawienie klasyfikatorów pochodzących z danych ze zbiorów treningowych w stosunku do ruchu badanego otrzymujemy klasyfikację bieżącego ruchu sieciowego. Informacja ta przekazywana jest do MODSAI celem podjęcia dalszych działań decyzyjnych przez kontroler sieci SDN.

Dobór zbiorów danych

Przy doborze zbiorów danych uczących autor po wnikliwej analizie dostępnych i możliwych do wykorzystania w pracy zbiorów postanowił oprzeć się na zestawach danych udostępnianych przez Kanadyjski Instytut Cyberbezpieczeństwa (CIC)[93]. The Canadian Institute for Cybersecurity (CIC) jest multidyscyplinarną jednostką badawczo-rozwojową, szkoleniową oraz przedsiębiorczą. Ma siedzibę w University of New Brunswick i jest wiodącą kanadyjską jednostką badawczą w dziedzinie cyberbezpieczeństwa. Wywodzi się z Wydziału Informatyki UNB i od 2007 roku (ISCX Information Security Centre of Excellence – Centrum Doskonałości ds. Bezpieczeństwa Informacji) odgrywa ważną rolę w dziedzinie cyberbezpieczeństwa. ISCX był prekursorem CIC. Tworzy m.in. raport Cyber Daily – aktualną stronę informacyjną z dziedziny cyberbezpieczeństwa i technologii[94].

Zbiory danych CIC są wykorzystywane na całym świecie zarówno przez uczelnie, niezależnych naukowców, jak i przemysł. CIC prowadzi akademicki projekt badawczy HONEYNET – sieć honeypotów umożliwiającą obserwowanie hakerów w akcji i zbieranie informacji o ich zrachowaniach i używanych metodach czy narzędziach. Lista firm, uniwersytetów oraz ośrodków badawczych, które korzystały z zestawów danych CIC obejmuje 197 pozycji (dane na rok 2021).

Biorąc pod uwagę różnorodność i złożoność obecnie występujących ataków sieciowych modyfikujemy metodę w następujący sposób: Naszym lasem losowym jest zestaw pięciu lasów losowych wyćwiczonych na następujących zestawach danych: CIC-Bell-DNS2021, CIRA-CIC-DOHBrw-2020, IDS 2018, NSL-KDD, DDoS 2019

Opis postępowania/metodologia ze zbiorami danych

Krok 1 – Zestawy danych

Zestaw danych CIC-Bell-DNS-2021 zawiera 400000 niezłośliwych i 13011 złośliwych próbek przetworzonych z 1000000 niezłośliwych i 51453 znanych złośliwych domen. Złośliwe próbki

obejmują trzy kategorie: spamu, phisingu i złośliwego oprogramowania. Zestaw danych replikuje rzeczywiste scenariusze z ruchem niezłośliwym i złośliwym. Zawiera 32 zdefiniowane funkcje dyskryminacyjne (leksykalne, statystyczne DNS i zewnętrzne).

Kategoria	Domeny oryginalne	Domeny przetworzone	Pakiety przetworzone
Złośliwe oprogramowanie	26 895	9 432	182 266
Spam	8 254	1 976	61 046
Wyłudzenia (phishing)	16 307	12 586	95 492
Niezłośliwe	988 667	500 000	6 907 719

Tabela 1. Podział zbioru danych CIC-Bell-DNS-2021 pod względem liczby zebranych domen.

Zestaw danych CIRA-CIC-DoHBrw-2020 zawiera zestaw danych do analizy ruchu DoH (DNS over HTTPS) w ukrytych kanałach i tunelach. Zawiera 28 wyodrębnionych statystycznych cech ruchu. Zestaw zawiera niezłośliwy oraz złośliwy ruch DoH oraz ruch nie-DoH.

Ruch nie-DoH jest to ruch generowany przez dostęp do strony internetowej korzystającej z protokołu HTTPS.

Ruch niezłośliwy DoH jest to niezłośliwy ruch generowany przez przeglądarki internetowe w analogiczny sposób do ruchu nie-DoH.

Ruch złośliwy DoH jest to ruch generowany przez narzędzia tunelowania DNS (Iodine, DNSCat2, dns2tcp) które mogą wysyłać ruch TCP hermetyzowany w kwerendach DNS (czyli tworzą tunele zaszyfrowanych danych).

Zestaw danych CSE-CIC-IDS2018 na AWS zawiera siedem różnych scenariuszy ataków (Brute-force, Heartbleed, Botnet, DoS, DDos, ataki internetowe i infiltrację sieci od wewnątrz). Infrastruktura ataku obejmuje 50 maszyn, organizacja posiada 5 działów i 420 maszyn i 30 serwerów.

Typ ataku	Narzędzia	Czas trwania	Napastnik	Ofiara
Atak Bruteforce	FTP – Patator, SSH - Patator	24h	Kali Linux	Ubuntu 16.4 serwer WWW
Atak DoS	Hulk, GoldenEye, Slowloris, Slowhttptest	24h	Kali Linux	Ubuntu 16.4 Apache
Atak DoS	Heartleech	24h	Kali Linux	Ubuntu 12.04 Open SSL
Atak Web	Damn Vulnerable Web App (DVWA), In-house selenium framewor (XSS i Brute-force)	48h	Kali Linux	Ubuntu 16.4 serwer WWW
Atak infiltracyjny	Pierwszy poziom: pobieranie Dropbox na komputerze z systemem Windows	48h	Kali Linux	Windows Vista i Macintosh

	Drugi poziom: Nmap i portscan			
Atak bootnet	Ares (w Python): zdalna powłoka, przesyłanie/pobieranie plików, przechwytywanie, zrzuty ekranu i rejestrowanie wpisów klawiatury	24h	Kali Linux	Windows Vista, 7, 8.1, 10 (32- i 64-bitowy)
DDoS+PortScan	LOIC dla żądań UDP, TCP lub HTTP	48h	Kali Linux	Windows Vista, 7, 8.1, 10 (32- i 64-bitowy)

Tabela 2. Lista przeprowadzonych ataków dla zbioru danych CSE-CIC-IDS-2018 na AWS

Narzędzie CICFlowMeter pozwala z tego zestawu wyodrębnić 80 funkcji ruchu sieciowego.

Zestaw danych NSL-KDD jest ulepszonym zbiorem danych KDD`99 i posiada znacznie zredukowaną liczbę nadmiarowych rekordów w stosunku do oryginału. Zestaw ten jest szeroko używany w pracach badawczych głównie z powodu braku publicznych zestawów danych dla sieciowych systemów IDS. Nadal może być stosowany jako skuteczny zestaw danych porównawczych dla różnych metod wykrywania włamań.

Zestaw danych CIC-DDoS2019 zawiera następujące rodzaje ataków DDoS: SNMP, DNS, NTP, SYN, UDP-Log, UDP, MSSQL, LDAP, NetBIOS, PortMap. Zestaw zawiera również wyniki analizy ruchu sieciowego przy pomocy narzędzia CICFlowMeter z oznaczonymi przepływami na podstawie znacznika czasu, źródłowych i docelowych IP, portów źródłowych i docelowych, protokołów i ataków. Pozwala na wyodrębnienie ponad 80-ciu funkcji ruchu sieciowego.

Krok 2 – Wybieranie funkcji ze zbioru danych

Z poszczególnych zestawów danych zostały wyodrębnione poszczególne cechy w zależności od rodzajów ataków zebranych w zbiorach danych.

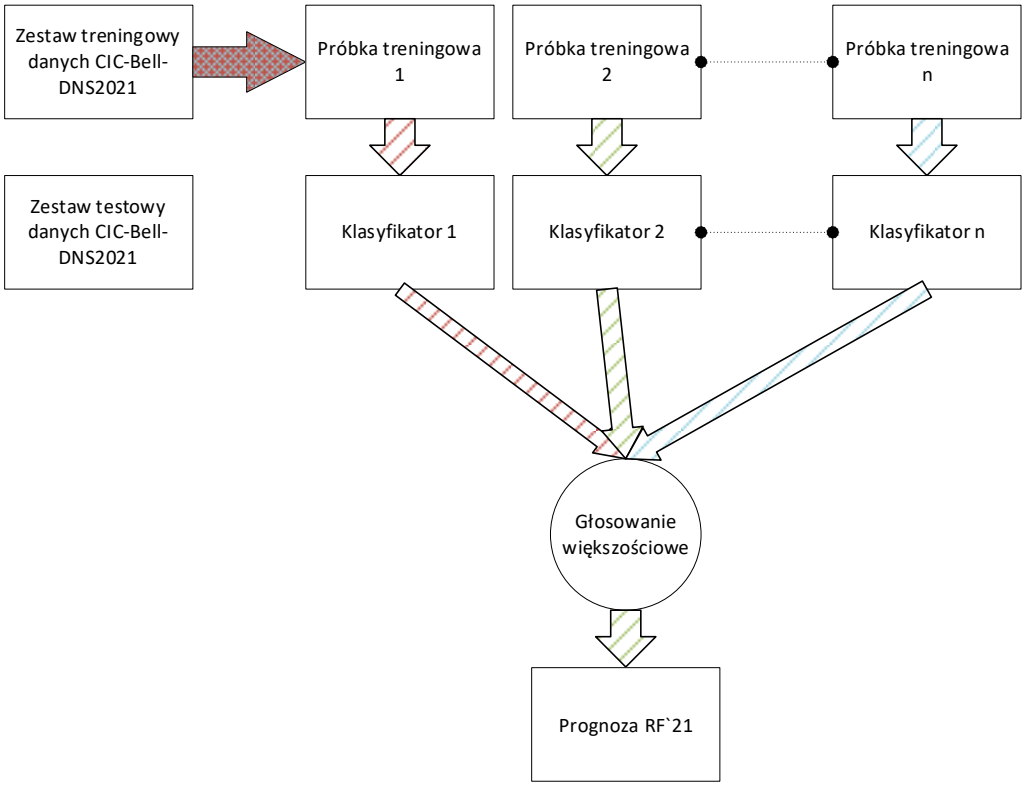
Dla ruchu z zestawu danych CIRCA-CIC-DoHBrw-2020 zostały wyodrębnione dwie klasy cech za pomocą narzędzi DoHLyzer oraz CICFlowMeter: cechy statystyczne, takie jak średnia, średnia mediana oraz funkcje sieciowe, takie jak źródłowy adres IP, numer portu, flagi.

Krok 3 – Korelacja wybranych funkcji ze zbioru danych z danymi dostępnymi w protokole OpenFlow

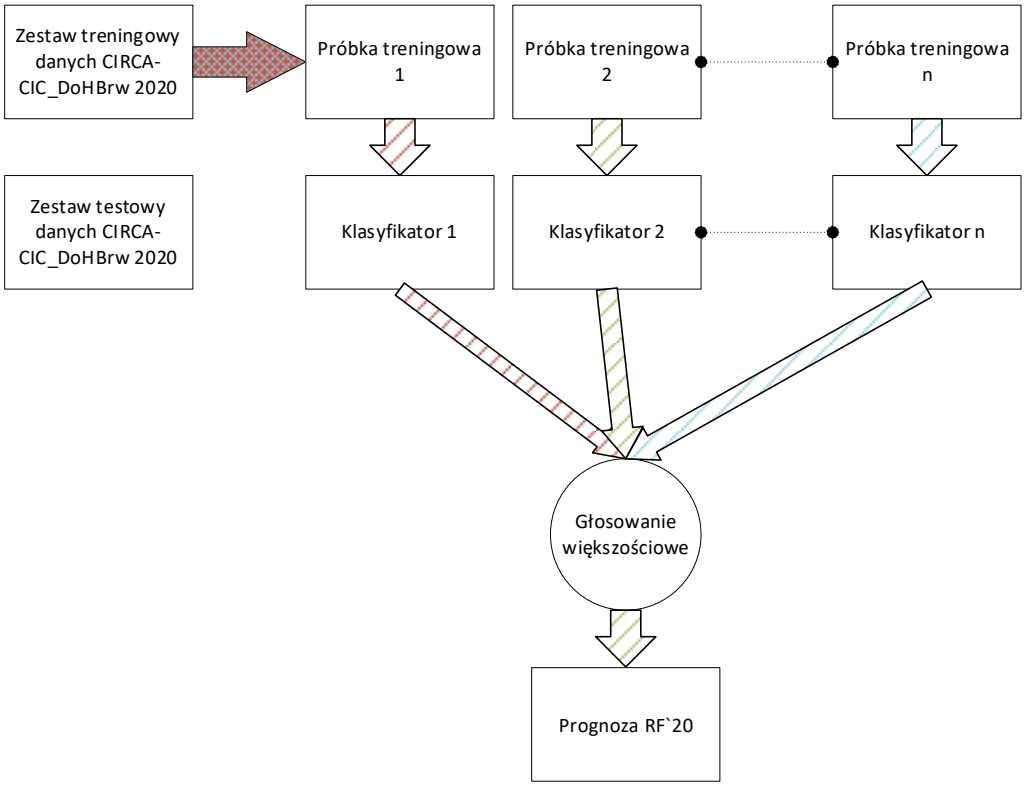
Korelacja ma na celu stworzenie agregatów funkcji na kontrolerze SDN, które będą przesyłane do modułu RFoRF w celu ich analizy pod kątem możliwości wystąpienia ataku.

Algorytm działania Modułu RFoRF.

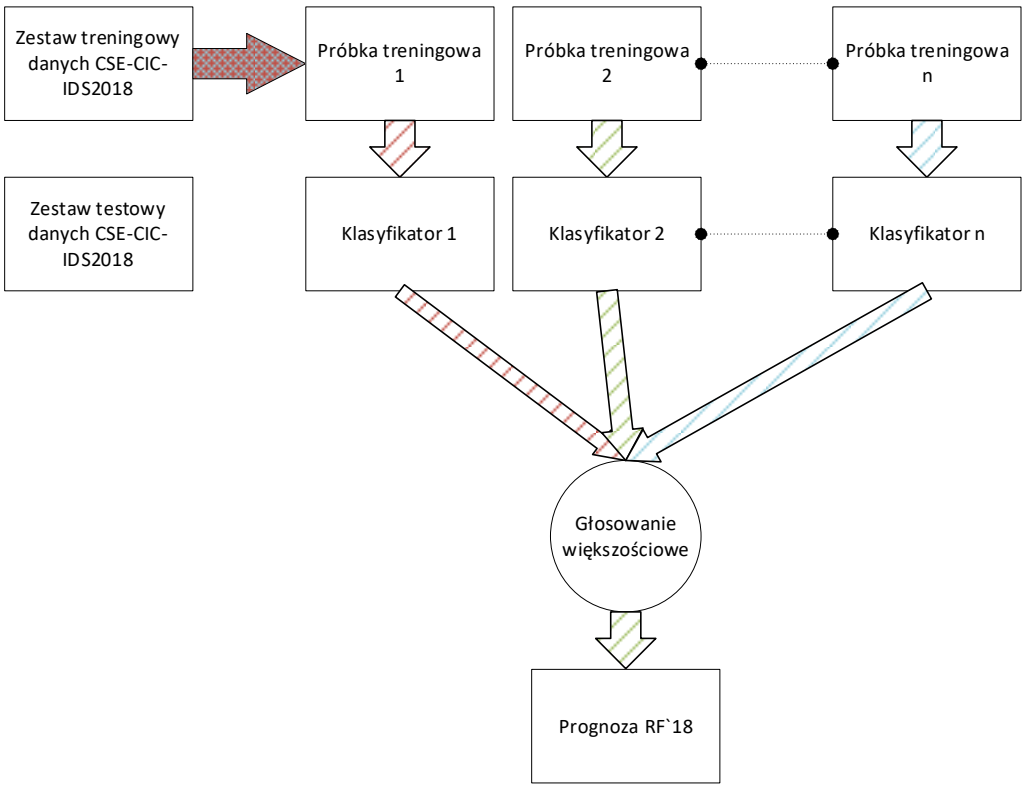
Algorytm został wyuczony za pomocą niezbalansowanych zbiorów danych. Zaletą lasów losowych jest ich wydajność dla dużych zbiorów danych.



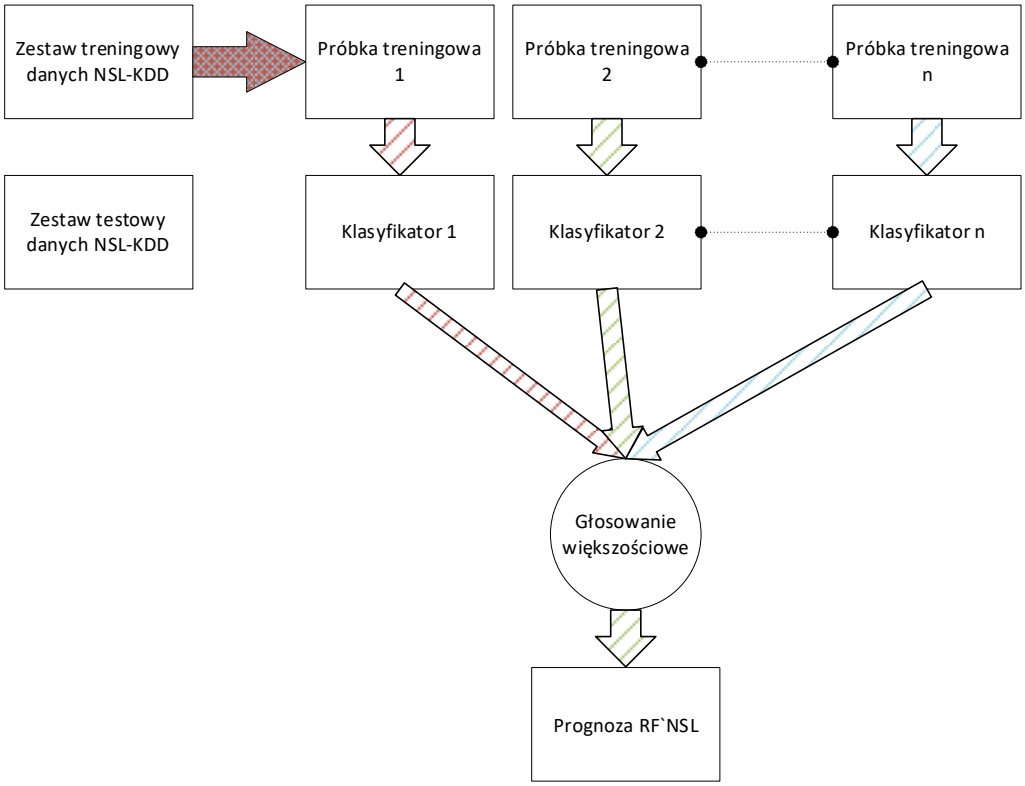
Rysunek 11 Schemat ideowy lasu dla zestawu danych CIC-Bell-DNS2021



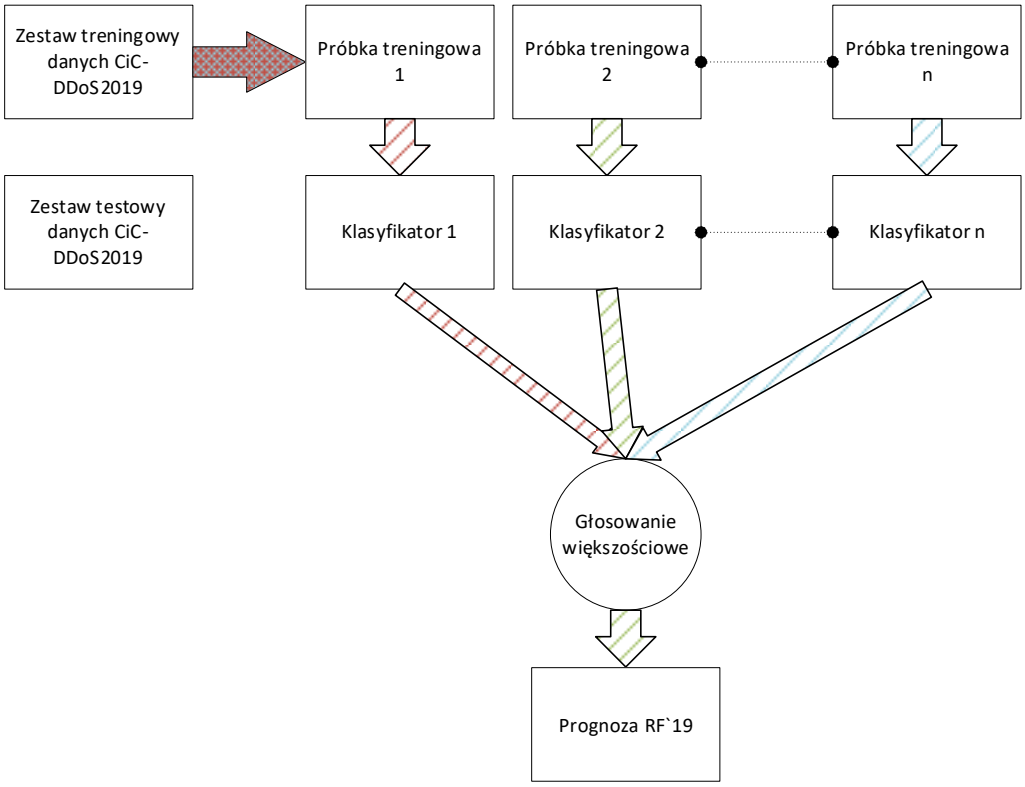
Rysunek 12. Schemat ideowy dla zestawu danych CIRCA-CIC-DoHBrw-2020



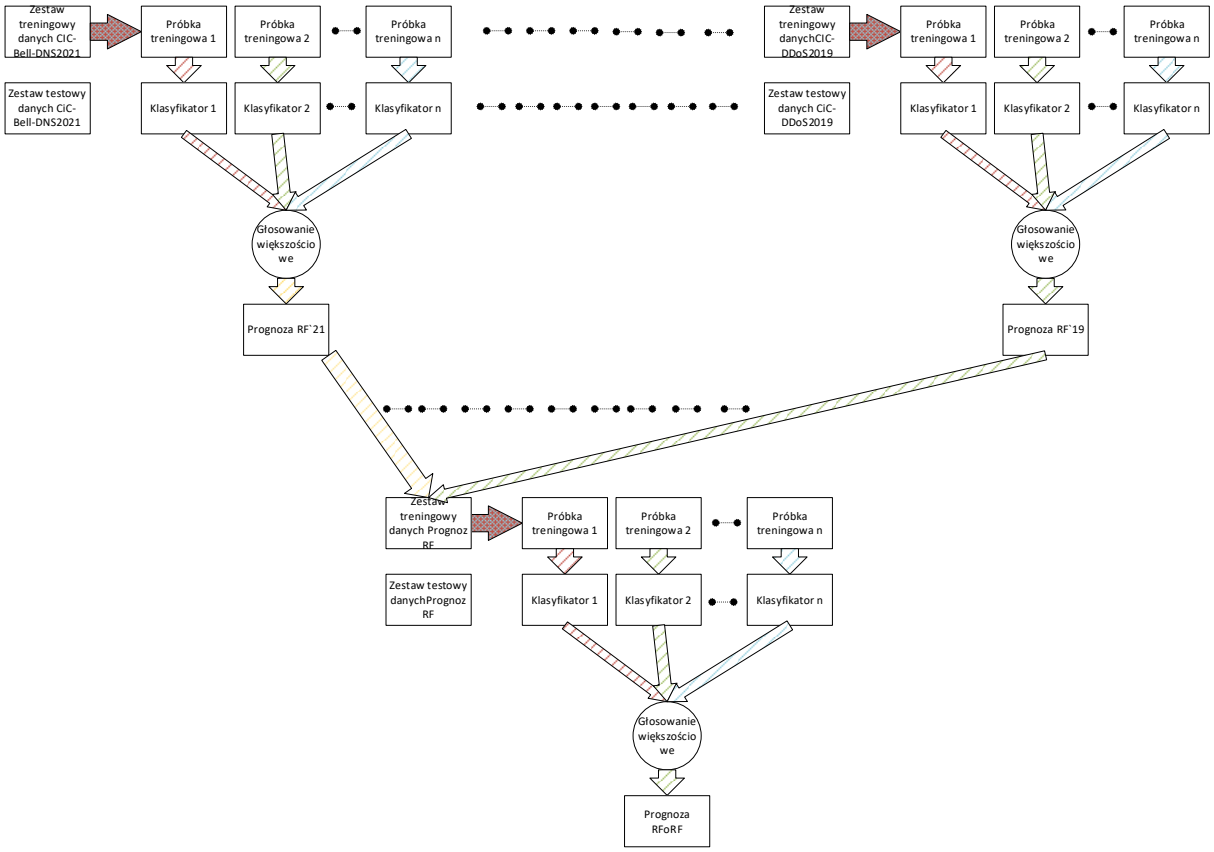
Rysunek 13. Schemat ideowy dla zestawu danych CSE-CIC-IDS-2018



Rysunek 14. Schemat ideowy dla zestawu danych NSL-KDL



Rysunek 15. Schemat ideowy dla zestawu danych CIC-DDoS-2019



Rysunek 16. Schemat ideowy RFoRF

Implementacja w środowisku testowym

Moduł RFoRF został zaimplementowany w postaci maszyny wirtualnej opartej na środowisku Linux Ubuntu 20.04.1 w postaci skryptów w Pyton 3.9.7 wraz z biblioteką scikit-learn 1.0.1. Maszyna zawiera również zbiory danych wykorzystanych do uczenia z Kanadyjskiego Instytutu Cyberbezpieczeństwa (ang. *Canadian Institute for Cybersecurity*). Rozwiązanie testowano również dla kontrolerów SDN w wersji OpenDaylight (Silicon-SR2) oraz Ryu – opracowano moduły agregacji funkcji dla przesyłania wybranych cech ruchu, z ruchu bieżącego w celu zmniejszenia obciążenia kontrolera.

Zamierzony rezultat:

Wykorzystanie uczenia maszynowego nadzorowanego w celu wykrywania potencjalnych ataków sieciowych, w szczególności ataków typu DDoS. Wykorzystanie w tym module zestawów danych treningowych zarówno najnowszych (lata 2021, 2019, 2017), jak i zestawu już „klasycznego" (NSL-KDD), stworzenie metody wnioskowania większościowego na podstawie wytrenowanych pod zbiorczych lasów losowych. Stanowią one nowy zbiór danych dla głównego lasu losowego, co pozwoliło na uzyskanie dużej dokładności wykrywanych ataków przy jednoczesnym uwrażliwieniu wykrywania na większe spektrum ataków sieciowych niż w rozwiązaniach opierających się na jednym zestawie danych (często nieaktualnym).

Podsumowanie

Zastosowana przez autora w MOD2 zmodyfikowana metody RFoRF pozwala elastycznie modyfikować ją o kolejne nowsze (aktualne, bądź specyficzne w danym obszarze zastosowań) zbiory danych treningowych. To rozwiązanie daje dużą skuteczność wykrywania różnorodnych (w tym złożonych) ataków sieciowych DDoS. Analiza wykrywania m.in. odbywa się o zestawione w tabeli 2 rodzaje ataków, jednak zastosowanie innego zestawu treningowego zwiększ spektrum wykrywanych ataków.

Wyniki skuteczności działania MOD2 zostaną przedstawione w dalszej części pracy.

3.4.3. MODUŁ MOD3 (GRUOGPU)

Moduł MOD3 o nazwie IDS_DL-GRUoGPU swoją zasadę działania opiera o głębokie uczenie maszynowe oparte na nienadzorowanych bramkowanych sieciach neuronowych, wspomagane akceleracją kart GPU.

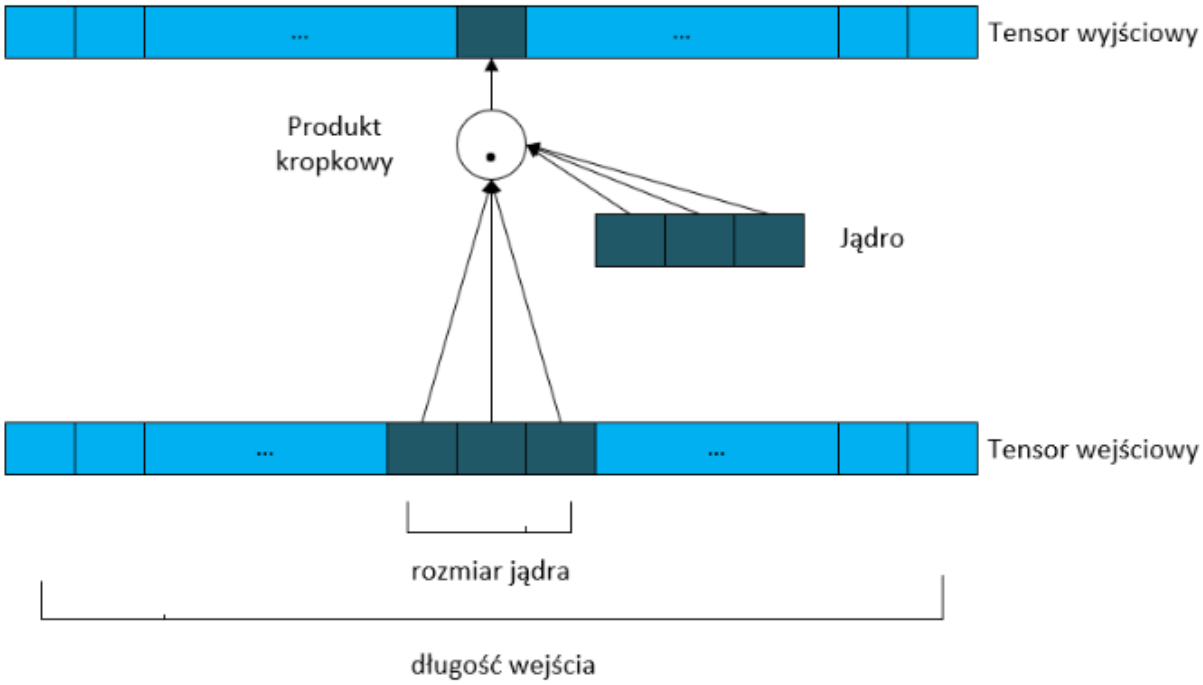
Model wykrywania ataku DDoS.

Model wykrywania ataku DDoS opiera się na korelacji dostępnych w protokole OpenFlow parametrów sieciowych z charakterystykami ataków sieciowych. Dane o bieżącym ruchu sieciowym dostarczane przez moduł MODKOL (Kolekcjonera Ruchu Sieciowego) z portu mirror przełącznika brzegowego przechwytywane za pomocą narzędzia tcpdump są korelowane parametrami sieciowymi dostępnymi w protokole OpenFlow i stanowią konglomerat danych wejściowych do Modułu MOD3.

Opis algorytmu

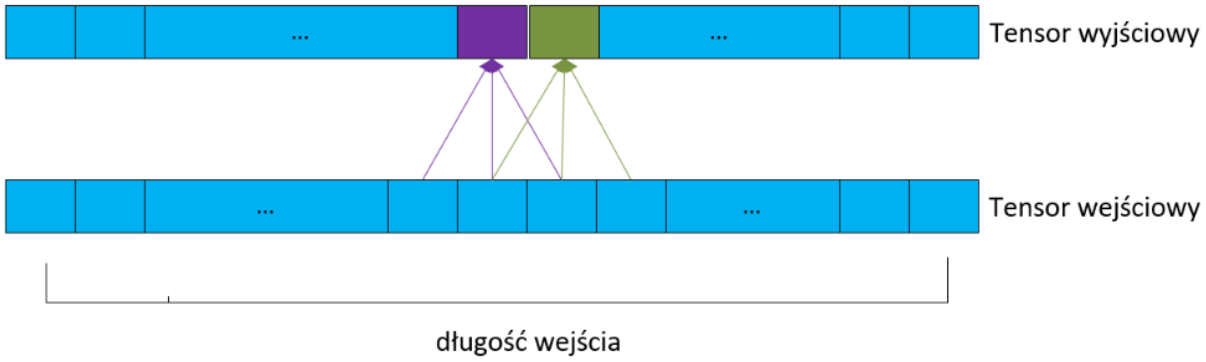
Autor początkowo zaimplementował wielowarstwową sieć GRU, jednak w trakcie testów okazało się iż jednostki GRU mają tendencję do zapominania wyuczonych wzorców dla bardziej złożonych ataków sieciowych rozciągniętych w czasie. Powodowało to opóźnienia w detekcji niektórych ataków w trakcie stałego działania systemu.

Autor postanowił jako dodatkową warstwę wejściową przed jednostkami GRU zastosować znaną z konwolucyjnych sieci neuronowych warstwę splotową 1D. Warstwa ta jest używana w czasowych CNN (TNN).



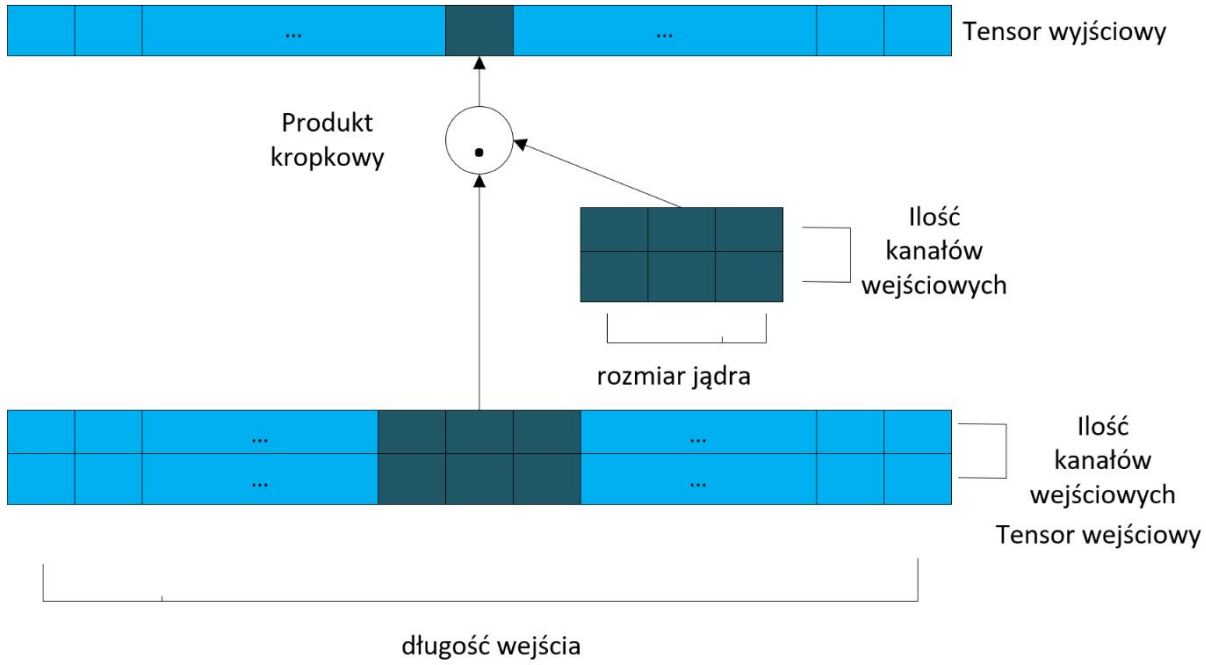
Rysunek 17. Sposób obliczania jednego elementu tensora wyjściowego sieci splotowej 1D

Jedna pojedyńcza warstwa splotu 1D otrzymuje wejściowy tensor i wyprowadza tensor wyjściowy. Aby obliczyć jeden element wyniku, bierzemy pod uwagę szereg kolejnych elementów o długości rozmiaru jądra (na ilustracji rozmiar przyjęto jako równy 3). Aby otrzymać wynik bierzemy iloczyn skalarny podciągu wejścia i wektor jądra z wyuczonymi wagami o tej samej długości. Aby uzyskać następny element wyniku, stosuje się tę samą procedurę, ale okno o rozmiarze rozmiar jądra sekwencji wejściowej jest przesuwane w prawo o jeden element (przy przyjętym tutaj kroku równym 1).



Rysunek 18 Dwa kolejne elementy wyjściowe i odpowiadające im pod sekwencje wejściowe.

Dla przypadku z wieloma kanałami wejściowymi proces jest powtarzany dla każdego kanału wejściowego, ale za każdym razem z innym jądrem, co prowadzi do pośrednich wektorów wyjściowych, które następnie są sumowane, by otrzymać ostateczny wektor wyjściowy.



Rysunek 19. Przypadek dla wielu kanałów wejściowych.

Autor używa warstwy splotowej 1D do przetwarzania sekwencji i zapewnienia uczenia się długoterminowych wzorców w przypadkach złożonych ataków sieciowych. Przy rozmiarze jądra większym niż krok uzyskuje się użycie wszystkich danych wejściowych do obliczenia danych

wyjściowych warstwy, co sprawia, iż model uczy się zachowywać przydatne informacje, a zastosowanie warstwy splotowej pozwala warstwom komórek GRU wykryć dłuższe wzory.

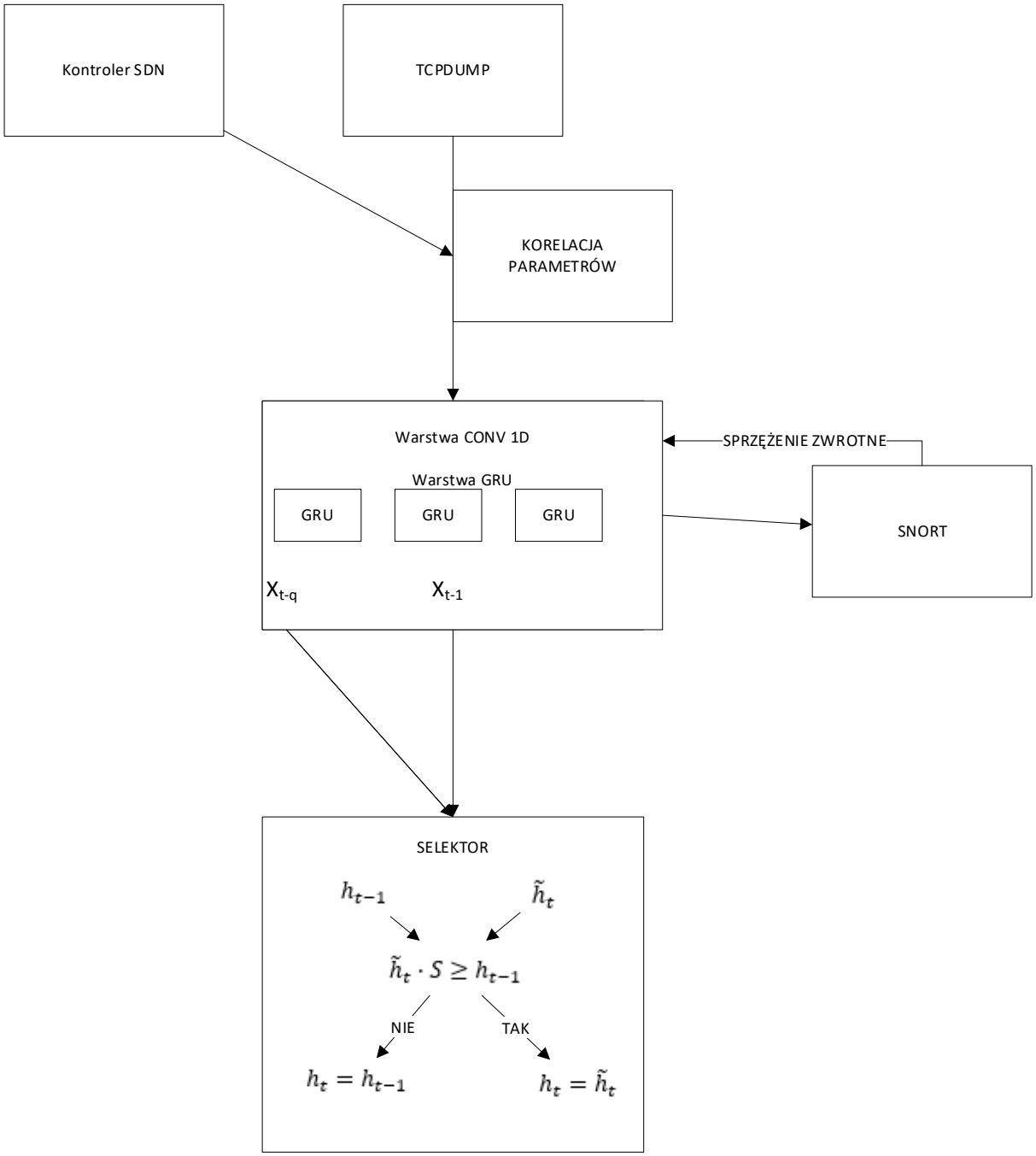
Ostatecznie Autor zaimplementował sieć neuronową złożoną z warstw splotowych 1D i powtarzających się warstw komórek GRU.

Dane wejściowe pochodzą ze zbioru przechwytywanego na bieżąco za pomocą programu tcpdump ruchu sieciowego i są skorelowane z parametrami sieciowymi dostępnymi w kontrolerze sieci SDN.

Funkcja korelacyjna
Źródło/Docelowy adres IP / Źródło/Docelowy numer portu/ Znacznik czasu
Szybkość / Liczba wysłanych lub odebranych bajtów przepływu
Wariancja / Pochylenie od mediany / Pochylenie od trybu / Współczynnik zmienności / Odchylenie standardowe / Średnia / Mediana długości pakietu
Odchylenie standardowe / Współczynnik zmienności / Pochylenie od mediany / Pochylenie od trybu / Wariancja / Średnia / Mediana / Tryb czasu pakietu
Mediana / Odchylenie standardowe / Współczynnik zmienności / Pochylenie od mediany / Pochylenie od trybu / Odchylenie / Średnia / Różnica czasu żądania/odpowiedzi
Czas trwania przepływu

Tabela 4. Tabela wybranych statystycznych cech ruchu korelowanych z protokołu OpenFlow

Statystyczne cechy ruchu i jego parametry zostały dobrane ze względu na charakterystyki wybranych ataków sieciowych.



Rysunek 20. Schemat działania modułu

W modelu autor zastosował również dodatkową modyfikację w postaci mechanizmu selekcji aktualizacji stanów ukrytych komórek GRU (w przypadku ataków DDoS następuje nagły przyrost danych i w takim przypadku stan ukryty powinien zostać zaktualizowany) oraz dodatkowe sprzężenie zwrotne z oprogramowania SNORT celem dążenia modelu do optymalizacji parametrów (czyli wykrywania złożonych ataków).

Oprogramowanie SNORT zostało zastosowane w celu ustalenia charakterystyki normalnego ruchu sieciowego dla środowiska produkcyjnego. Ze względu na specyfikę przedsiębiorstwa, w sieci znajduje się np. dużo urządzeń typu IoT oraz serwery i komputery przeznaczone do wdrażania i programowania urządzeń. Przez okres 12-tu miesięcy zbierano dane o ruchu sieciowym wewnątrz firmy i poddano go

analizie, w wyniku której ustalono charakterystykę ruchu normalnego. Oprogramowanie SNORT zawiera funkcjonalność ustawiania alertów progowych i między innymi tą funkcjonalność wykorzystano jako sprzężenie zwrotne, jeżeli moduł MOD3 wykryje potencjalny atak i jego charakterystyka wychodzi również poza normę ustaloną w oprogramowaniu SNORT, to informację tę wykorzystujemy do wzmocnienia uczenia.

Implementacja w środowisku testowym

Moduł MOD3 zaimplementowano na fizycznym serwerze z czterema kartami GPU NVIDIA RTX 3080 w środowisku Linux Debian 20.04 wraz z Python v.3.8. i bibliotekami KERAS 2.1.5 i TensorFlow 2.5.0 oraz PyTorch 1.10, CUDA Toolkit 11.2 i 11.3, cuDNN SDK 8.1.

Zamierzony rezultat

Moduł MOD3 opierający się na rekurencyjnych bramkowanych sieciach GRU wspomaganych obliczeniami na kartach GPU, pozwala na detekcję złożonych bądź nieznanych ataków sieciowych. Przeprowadzone testy wykazały wysoką skuteczność działania modułu przy jednoczesnej małej wartości fałszywie dodatnich detekcji i niskim czasie klasyfikacji.

Podsumowanie

Zastosowany przez autora w MOD3 mechanizm GRU został zmodyfikowany (w wyniku przeprowadzonych badań) o dodatkową warstwę wejściową splotową 1D, modyfikację stanu ukrytego komórek GRU oraz wzmocnienie uczenia na podstawie danych z analizy programu SNORT. Siła rozwiązania oprócz zastosowanego algorytmu tkwi w implementacji fizycznej. W rozwiązaniu wykorzystano fizyczny serwer z 4 kartami akceleracji sprzętowej GPU o dużej mocy obliczeniowej. Serwer posiada 1TB pamięci RAM i dwa procesory Intel Xeon Gold 6320. Taka konfiguracja sprzętowa umożliwia złożone obliczenia arytmetyczne wynikające z zastosowanego algorytmu i uzyskiwanie wyników w, krótkim (akceptowalnym) czasie co umożliwia szybką reakcję systemu w przypadku detekcji ataku DDoS. Jest to rozwiązanie, które jako jedyne umożliwia detekcję nowych nieznanych ataków w tym typu Zero-Day.

Wyniki skuteczności działania MOD3 zostaną przedstawione w dalszej części pracy.

4. OPIS ŚRODOWISKA TESTOWEGO, TESTY WERYFIKACYJNE SYTEMU

4.1. OPIS ŚRODOWISKA TESTOWEGO

W celu możliwości oceny rezultatów działania systemu STRAINER, zbadania wydajności, dokładności i szybkości działania zaproponowanego systemu mitygacji, autor zbudował testowe środowisko laboratoryjne.

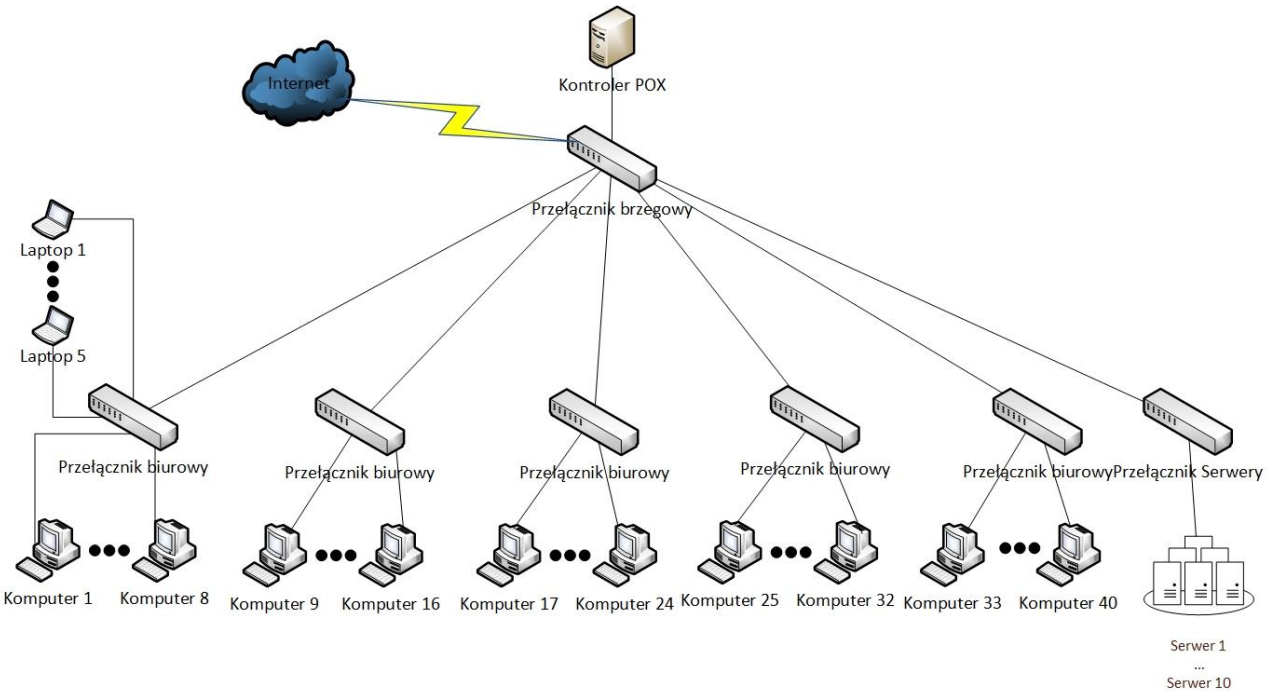
Do utworzenia środowiska testowego został zaprojektowany system sprzętowo-programowy. Środowisko laboratoryjne wirtualne zostało zbudowane w oparciu o oprogramowanie:

1) Mininet 2.3.0 – wirtualne środowisko sieciowe, zawierające wirtualne urządzenia sieciowe (vSwitch), interfejsy do sieci rzeczywistej. [96]
2) Instancje kontrolerów SDN w postaci maszyn wirtualnych, lub zintegrowanych w Mininet
    a. POX [97]
    b. OpenDaylight [98]
    c. Ryu [99]
3) Język skryptowy Python z różnymi bibliotekami [100] – do budowy modułów systemu (skryptów kontrolera SDN).
4) System operacyjny Linux Debian 20.04

Kluczowe dla wykonania testów było użycie oprogramowania Mininet jako emulatora sieci dla ewaluacji poszczególnych modułów systemu. To środowisko pozwala stworzyć realistyczne wirtualne odwzorowanie środowiska sieciowego dla różnych scenariuszy. Na podstawie analogii do sieci produkcyjnej zostało stworzone środowisko sieciowe, złożone z siedmiu przełączników Open vSwitch, 45 hostów oraz 10 serwerów.

Całość środowiska uruchomiono na czteroprocesorowym serwerze (Intel Xeon Gold 6230) wyposażonym w 2 TB pamięci RAM. Pozwoliło to na wykonanie różnego rodzaju ataków, ze zróżnicowaną mocą.

Na rysunku 21 przedstawiono topologia sieci laboratoryjnej (testowej). Wszystkie hosty i serwery są zaimplementowane w postaci maszyn wirtualnych z systemem Linux Debian 20.04. Kontroler sieci SDN przedstawiono ideowo jako POX, w praktyce wersje kontrolerów były różne w zależności od wykonywanego testu.



Rysunek 21 Topologia sieci laboratoryjnej (testowej)

Poniżej przedstawiono zastosowane oprogramowanie symulacji ataków wraz z procedurami, nastawami testowymi.

Do realizacji symulowanych ataków zastosowane oprogramowanie:

1) Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution [101]

Procedury testowe:

Test – Osiem hostów (maszyn wirtualnych z dystrybucją KALI Linux), typ ataku SYN Flood, rozpoczynając od 50000 pakietów/sekundę i stopniowo zwiększając liczbę pakietów. Jako generator ataków DDoS użyty został wirtualne maszyny z otwartym oprogramowaniem KALI Linux.

Atak SYN Flood:

8 maszyn wirtualnych z oprogramowaniem KALI Linux na każdym z nich uruchomiono polecenie hping3 z następującymi parametrami:

$ sudo hping3 -i u20 -S -p 80 -c 50000 192.x.x.x

-S – wysyłanie pakietów SYN

-p 80 – docelowy port 80

-i u20 – czas oczekiwania między pakietami 20 mikrosekund, co daje 50000 pakietów/sekundę

Co dwie minuty na maszynach dokonujących ataku modyfikowano parametr -i u20 dzieląc go na 2 za każdym razem, co skutkuje zwiększeniem o ponad 5000 pakietów na sekundę za każdą iteracją.

2) LOIC – Low Orbit ION Cannon network stress testing application [102]

   Oprogramowanie pozwala wysyłać żądania typu http, TCP i UDP na zdefiniowany adres

3) HOIC -High Orbit ION Cannon

   Oprogramowanie umożliwia atak do 256 serwisów web (http)

4) HULK – http Unbearable Load King [104]

   Oprogramowanie pozwala przeprowadzać ataki na serwer web (http)

5) DDoSIM (DDoS Simulator)

   Oprogramowanie pozwala przeprowadzać ataki TCP na wskazany port

6) PyLoris 3.0 w połączeniu z python i PyLoris [105]

   Skryptowe narzędzie do testowania podatności na ataki DoS. Może używać połączeń SSL i protokołów HTTP, FTP, SMTP, IMAP i TELNET

7) CALDERA [106]

8) Hyenae NG [107]

9) SCAPY [108]

10) TCPReplay – narzędzie do manipulacji przechwyconymi pakietami ruchu sieciowego [109]

11) D-ITG – Distributed Internet Traffic Generator [110]

12) Nmap – skaner sieciowy [111]

13) Wireshark – skaner sieciowy[112]

14) SNORT 3 [113]

4.2. RODZAJE TESTÓW, DLA POSZCZEGÓLNYCH MODUŁÓW SYSTEMU

Ocena testów

Do oceny przeprowadzonych testów wykorzystujemy klasyfikatory:

- Liczbę poprawnych klasyfikacji ataku przyjętą jako wynik prawdziwie dodatni (ang. *True Positive*, TP),
- Liczbę poprawnych klasyfikacji ruchu normalnego przyjętą jako wynik prawdziwie ujemny (ang. *True Negative*, TN),
- Liczbę niepoprawnych klasyfikacji ataku jako ruchu normalnego przyjętą jako wynik fałszywie ujemny (ang. *False Negative*, FN),
- Liczbę niepoprawnych klasyfikacji ruchu normalnego jako ataku przyjętą jako wynik fałszywie dodatni (ang. *False Positive*, FP)
- Skuteczność detekcji (Dokładność wykrycia) przyjętą jako:

$$DW = \left(\frac{TP + TN}{TP + TN + FN + FP}\right) * 100\% \tag{13}$$

Moduł MOD1 IDS_EBEDM

W teście użyto oprogramowania Scapy [108] oraz PyLoris [105] wraz z KALI Linux[101] do generowania ruchu normalnego oraz przeprowadzania ataków. Wzorzec normalnego ruchu został opracowany na podstawie przechwyconego wcześniej ruchu z produkcyjnych komputerów w trakcie normalnego działania firmy. Dane te były zbierane na przestrzeni roboczego tygodnia, a następnie zostały przeanalizowane i na podstawie analizy został przygotowany wzorzec normalnego ruchu wewnątrz sieci. Użycie oprogramowania TCPReplay [109] umożliwiło wykorzystanie przechwyconego wcześniej ruchu dla symulacji rzeczywistego ruchu sieciowego dla lepszego odwzorowania rzeczywistości.

Jako cel ataków wybrano 10 hostów – serwerów wystawiających usługi: stron web zewnętrznej i wewnętrznej, serwer ftp, serwer udziałów sieciowych oraz 5 hostów symulujących laptopy pracowników.

Dla symulacji ataków generowano je z dziesięciu hostów na 10 docelowych serwerów oraz 5 docelowych hostów. 8 maszyn wirtualnych z systemem KALI Linux zostało wykorzystanych do przeprowadzania wzrastającego ataku DDoS SYN Flood na 2 serwery hostujące usługi http. Zaczęto od 50000 pakietów/sekundę i dwóch hostów atakujących, stopniowo zwiększając liczbę pakietów, oraz dołączając kolejne hosty atakujące. Sprawdzono dokładność wykrywania ataku przy różnych konfiguracjach hostów atakujących.

Do tego ataku użyto maszyn wirtualnych z systemem operacyjnym KALI Linux i narzędzia HPING3 w następującej konfiguracji:

Hping3 -c 50000 -d 128 -S -w 64 -p 8000 –flood –rand-source 192.168.10.19
-c (liczba pakietów)
-d (rozmiar pakietu)
-S (pakiet SYN)
-w (rozmiar okna tcp)
-p (port)
- - flood (tryb zalania)
- - rand-source (losowe źródłowe adresy IP)

Po tym SYN FLOOD:

Hping3 -S -p 8000 -V 192.168.10.19

Hping3 -S -p 8000 –flood -V 192.168.10.19

Następnie stopniowo zwiększano natężenie ataku oraz liczbę atakujących maszyn.

Poniżej zestawienie przeprowadzonych testów skuteczności wykrywania ataków

Nasilenie ataku	Dokładność wykrycia	Fałszywie pozytywne	Fałszywie negatywne
2 atakujące hosty, 50000 pakietów/sekundę	96,45%	0	3,55%
2 atakujące hosty, 55000 pakietów/sekundę	98,73%	0,00%	1,27%
4 atakujące hosty, 50000 pakietów/sekundę	99,97%	0,03%	0,03%
5 atakujących hostów, 50000 pakietów/sekundę	100%	0,00%	0,00%
5 atakujących hostów, 55000 pakietów/sekundę	100%	0,00%	0,00%
8 atakujących hostów, 50000 pakietów/sekundę	100%	0,00%	0,00%

Tabela 5 Podsumowanie dokładności wykrywania ataku

Wnioski

Wraz ze wzrostem liczby atakujących pakietów/sekundę dokładność wykrywania ataku znacząco rosła. Skuteczność wykrywania ataku rosła również wraz ze wzrostem ilości atakujących hostów. Jednakże nawet najmniejsza osiągana dokładność wykrywania ataku przy małym jego nasileniu udowodniła skuteczność stosowania algorytmu wykrywania ataku opartego na entropii ruchu przy typowych atakach, których charakterystyki ruchu sieciowego są dobrze znane. Zastosowanie tego modułu wykrywania na przełączniku brzegowym (czyli dodanie małej części „inteligencji" do przełącznika) pozwala w znaczny sposób odciążyć kontroler SDN dla typowych ataków DDoS. Dodanie do tego modułu części mitygacyjnej pozwala na znaczne ograniczenie rozprzestrzeniania się ataku już na brzegu sieci i wydaje się bardzo dobrą wstępną metodą dla ograniczania ataków.

Jednocześnie wydaje się to udowadniać skuteczność przyjętej architektury systemu wykrywania i mitygacji ataków, gdyż w ten sposób można znacząco odciążyć zarówno sam kontroler SDN i wykorzystać jego zasoby do detekcji i mitygacji ataków mniej znanych, bądź nierozpoznanych, jak również ograniczyć nieprawidłowy ruch wewnątrz sieci.

Moduł MOD2 IDS_RFoRF

Podobnie jak wcześniej w teście użyto oprogramowania Scapy [108] oraz PyLoris [105] wraz z KALI Linux [101] do generowania ruchu normalnego oraz przeprowadzania ataków. W celu rozszerzenia rodzajów ataków użyto dodatkowych narzędzi (LOIC, HOIC, DDoSIM, D-ITG).

Jako cel ataków wybrano 10 hostów – serwerów wystawiających usługi: stron web zewnętrznej i wewnętrznej, serwer ftp, serwer udziałów sieciowych oraz 5 hostów symulujących laptopy pracowników.

Parametry eksperymentu	Wartości
Liczba przełączników OpenFlow	7
Liczba hostów	45
Liczba serwerów	10
Liczba atakowanych hostów	15 (10 serwerów wystawiających usługi, 5 hostów symulujących laptopy pracownicze)
Typy ataku	Zalew żądań ICMP, Zalew żądań PING, Zalew żądań TCP SYN, Zalew żądań UDP, Zalew żądań IP Null, Zalew żądań SNMP, zalew żądań NTP, zalew żądań http, Zalew żądań DNS
Generatory ruchu i ataku	Scapy, PyLoris, KALI Linux, TCPReplay
Analiza pakietów	Tcpdump, Wireshark

Tabela 6 – parametry eksperymentu

Dla symulacji ataków wygenerowano z dziesięciu hostów na 10 docelowych serwerów oraz 5 docelowych hostów. 8 maszyn wirtualnych z systemem KALI Linux zostało wykorzystanych do przeprowadzania wzrastającego ataku DDoS SYN Flood na 2 serwery hostujące usługi http. Zaczynaliśmy od 50000 pakietów/sekundę, stopniowo zwiększając liczbę pakietów.

Przeprowadzone testy:

Dokładność wykrywania ataków DDoS

Ataki DDoS można podzielić na dwie ogólne kategorie:

1) Typy ataków DDoS w warstwie aplikacji (warstwa 7 modelu OSI) obejmują powodzie HTTP, powolne ataki(Slowloris, RUDY), ataki zero-day oraz ataki ukierunkowane na luki w systemach operacyjnych, aplikacjach internetowych i protokołach komunikacyjnych.

   Ataki te składają się z pozornie uzasadnionych i niewinnych żądań, których wielkość jest zwykle mierzona w żądaniach na sekundę (RPS), zaś celem ataków jest przytłoczenie aplikacji docelowej żądaniami. Powoduje to wysokie użycie procesora i pamięci, które ostatecznie zawiesza lub powoduje awarię aplikacji.

2) Typy ataków DDoS w warstwie sieciowej (warstwy 3-4 modelu OSI) obejmują powodzie UDP, powodzie SYN, wzmocnienie NTP, wzmocnienie DNS, wzmocnienieSSDP, fragmentację IP i inne. Prawie zawsze są atakami DDoS skonfigurowanymi w celu zatkania protokołów łączących sieć. Prawie zawsze wykonywane są przez botnety, w celu zużycia przepustowości nadrzędnej celu, co powoduje wycenie przepustowości sieci.

   Należy zauważyć, że ataki DDoS mogą również atakować infrastrukturę i usługi pomocnicze – najczęściej serwery DNS celu. Mogą one być nadmiernie obciążone zalewem sfabrykowanych żądań DNS, pochodzących z urządzeń botnetowych.

   W testach tego modułu określono dokładność wykrywania ataku DDoS. W fazie testów przeprowadzono różne typy ataków DDoS. Pod uwagę wzięliśmy następujące typy ataków DDoS:

- Zalew żądań ICMP - ma na celu próbę przytłoczenia zaatakowanego urządzenia pakietami żądania echa ICMP. Obiekt docelowy musi przetwarzać i reagować na każdy pakiet, zużywając jego zasoby obliczeniowe, w wyniku czego legalni użytkownicy nie będą mogli odbierać usługi

- Zalew żądań PING - Rozwinięta wersja zalewu żądań ICMP (atak DDoS jest specyficzny dla aplikacji). Kiedy serwer otrzymuje wiele sfałszowanych pakietów Ping z bardzo dużego zestawu źródłowego adresu IP, jest celem ataku zalewu żądań Ping. Celem takiego ataku jest zalanie celu pakietami ping, dopóki nie przejdzie w tryb offline- został zaprojektowany tak, aby zużywać całą dostępną przepustowość i zasoby w sieci, dopóki zasoby nie zostaną całkowicie wyczerpane i staną się niedostępne. Ten rodzaj ataku DDoS nie jest łatwy do wykrycia, ponieważ może przypominać legalny ruch.

- Zalew żądań TCP SYN - SYN flood (half-open attack) to rodzaj ataku typu "odmowa usługi" (DDoS), którego celem jest uczynienie serwera niedostępnym dla legalnego ruchu poprzez wykorzystanie wszystkich dostępnych zasobów serwera. Wysyłając wielokrotnie pakiety syn (Initial Connection Request), osoba atakująca jest w stanie przytłoczyć wszystkie dostępne porty na docelowym komputerze serwerowym, powodując, że atakowane urządzenie reaguje na legalny ruch powolnie lub wcale.

- Zalew żądań UDP - Powódź UDP to rodzaj ataku typu "odmowa usługi", w którym duża liczba pakietów protokołu UDP (User Datagram Protocol) jest wysyłana do docelowego serwera w celu przyduszenia zdolności tego urządzenia do przetwarzania i reagowania. Zapora chroniąca atakowany serwer może również ulec wyczerpaniu w wyniku zalania UDP, co powoduje odmowę usługi dla legalnego ruchu.

- Zalew żądań IP Null - Pakiety zawierają nagłówki IPv4, które zawierają informacje o tym, który protokół transportowy jest używany. Gdy osoby atakujące ustawią wartość tego pola na zero, pakiety te mogą obejść środki bezpieczeństwa przeznaczone do skanowania protokołów TCP, IP i ICMP. Gdy serwer docelowy spróbuje przetworzyć te pakiety, ostatecznie wyczerpie swoje zasoby i uruchomi się ponownie.
  Zgodnie z regułami RFC nagłówek pakietu IP powinien zawierać informacje o protokole na poziomie transportu w polu Protokół. W przypadku ataku IP Null, złoczyńcy wysyłają pakiety zawierające wartość null w tym polu. Najczęściej routery brzegowe i zapory ogniowe wpuszczają taki pakiet jako niesklasyfikowany. Chociaż obecnie wartość null w polu Protocol jest zarezerwowana dla IPv6 Hop-by-Hop Option (HOPOPT), nie każdy serwer może odebrać i poprawnie przetworzyć taki pakiet. A jeśli takie pakiety przychodzą w dużych ilościach, ich analiza zużyje duży procent zasobów systemowych lub wyczerpie je całkowicie i spowoduje awarię serwera.

- Zalew żądań SNMP - może być używany do ataków wzmacniających. Protokół SNMP jest używany głównie na urządzeniach sieciowych. Atak SNMP amplification jest przeprowadzany poprzez wysyłanie małych pakietów ze sfałszowanym adresem IP celu do urządzeń z dostępem do Internetu z uruchomionym SNMP. Te sfałszowane żądania do takich urządzeń są następnie wykorzystywane do wysyłania powodzi UDP jako odpowiedzi z tych urządzeń do celu. Jednak efekt wzmocnienia w SNMP może być większy w porównaniu z atakami CHARGEN i DNS. Kiedy cel spróbuje rozpoznać ten zalew żądań, w końcu wyczerpie swoje zasoby i przejdzie w tryb offline lub uruchomi się ponownie

- Zalew żądań NTP - Atak wzmacniający NTP to oparty na odbiciu wolumetryczny atak typu "odmowa usługi" (DDoS), w którym osoba atakująca wykorzystuje funkcje serwera NTP (Network Time Protocol) w celu przytłoczenia docelowej sieci lub serwera zwiększoną ilością ruchu UDP, czyniąc cel i otaczającą go infrastrukturę niedostępnymi dla zwykłego ruchu.

- Zalew żądań HTTP - Ataki FLOOD HTTP są rodzajem ataku DDoS "warstwy 7". Warstwa 7 jest warstwą aplikacji modelu OSI i odnosi się do protokołów internetowych, takich jak HTTP. HTTP jest podstawą żądań internetowych opartych na przeglądarce i jest powszechnie używany do ładowania stron internetowych lub wysyłania treści formularzy przez Internet. Łagodzenie ataków w warstwie aplikacji jest szczególnie złożone, ponieważ złośliwy ruch jest trudny do odróżnienia od normalnego ruchu.
  Aby osiągnąć maksymalną skuteczność wykorzystuje się lub tworzy botnety w celu zmaksymalizowania wpływu ataku. Wykorzystując wiele urządzeń zainfekowanych złośliwym oprogramowaniem, atakujący jest w stanie uruchomić większe natężenie ruchu ataku.
  Istnieją dwie odmiany ataków powodziowych HTTP:

Atak HTTP GET - w tej formie ataku wiele komputerów lub innych urządzeń jest koordynowanych w celu wysyłania wielu żądań obrazów, plików lub innych zasobów z docelowego serwera. Gdy obiekt docelowy zostanie zalany przychodzącymi żądaniami i odpowiedziami, nastąpi odmowa usługi w przypadku dodatkowych żądań z legalnych źródeł ruchu.

Atak HTTP POST - zazwyczaj gdy formularz jest przesyłany na stronie internetowej, serwer musi obsłużyć przychodzące żądanie i wypchnąć dane z formularza do warstwy przechowania, najczęściej bazy danych. Proces obsługi danych formularza i uruchamiania niezbędnych poleceń bazy danych jest stosunkowo intensywny w porównaniu z ilością mocy obliczeniowej i przepustowości wymaganej do wysłania żądania POST. Atak ten wykorzystuje różnice we względnym zużyciu zasobów, wysyłając wiele żądań POST bezpośrednio do docelowego serwera, dopóki jego zasoby nie zostaną wysycone i nie nastąpi odmowa usługi.

- Zalew żądań DNS - Serwery Domain Name System (DNS) są "książkami telefonicznymi" Internetu; są one ścieżką, dzięki której urządzenia internetowe są w stanie wyszukać określone serwery internetowe w celu uzyskania dostępu do treści internetowych. Powódź DNS to rodzaj rozproszonego ataku typu "odmowa usługi" (DDoS), w którym osoba atakująca zalewa serwery DNS określonej domeny, próbując zakłócić rozpoznawanie DNS dla tej domeny. Jeśli użytkownik nie może znaleźć serwera DNS, nie może wyszukać adresu w celu wykonania połączenia z określonym zasobem. Zakłócając rozpoznawanie DNS, atak powodziowy DNS zagrozi zdolności witryny, interfejsu API lub aplikacji internetowej do reagowania na legalny ruch. Ataki powodziowe DNS mogą być trudne do odróżnienia od normalnego dużego ruchu, ponieważ duża ilość ruchu często pochodzi z wielu unikalnych lokalizacji, pytając o prawdziwe rekordy w domenie, naśladując legalny ruch.

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
20	Zalew żądań ICMP	6	97,2	1,4	1,4
	Zalew żądań PING	6	97,2	1,4	1,4
	Zalew żądań TCP SYN	13	98,9	0,5	0,6
	Zalew żądań UDP	4	98,25	1,2	0,55
	Żądania IP Null	10	96,36	0	3,64

	Zalew żądań SNMP	9	98,38	0	1,62
	Zalew żądań NTP	9	96,52	2,28	1,2
	Zalew żądań HTTP	12	98,45	0,55	1
	Zalew żądań DNS	12	97,88	0,12	2
	Średnia	9	97,68222222	0,827777778	1,49

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
30	Zalew żądań ICMP	4	98,8	0,6	0,6
	Zalew żądań PING	4	98,8	0,6	0,6
	Zalew żądań TCP SYN	11	99,1	0,5	0,4
	Zalew żądań UDP	3	98,85	1	0,15
	Żądania IP Null	9	98,61	0	1,39
	Zalew żądań SNMP	9	99,38	0	0,62
	Zalew żądań NTP	7	98,57	1	0,43
	Zalew żądań HTTP	11	99,56	0,24	0,2
	Zalew żądań DNS	11	98,99	0,11	0,9
	Średnia	7,66666667	98,96222222	0,45	0,587777778

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
45	Zalew żądań ICMP	4	98,89	0,555	0,555
	Zalew żądań PING	4	98,89	0,555	0,555
	Zalew żądań TCP SYN	11	99,36	0,34	0,3
	Zalew żądań UDP	3	99,41	0	0,59
	Żądania IP Null	9	99,15	0	0,85
	Zalew żądań SNMP	8	99,82	0,1	0,08
	Zalew żądań NTP	7	99,72	0,1	0,18
	Zalew żądań HTTP	10	99,68	0,2	0,02
	Zalew żądań DNS	10	99,01	0,19	0,8
	Średnia	7,33333333	99,32555556	0,226666667	0,436666667

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
50	Zalew żądań ICMP	5	99,49	0,255	0,255
	Zalew żądań PING	5	99,49	0,255	0,255
	Zalew żądań TCP SYN	11	99,47	0,27	0,26
	Zalew żądań UDP	4	99,49	0	0,51
	Żądania IP Null	10	99,19	0	0,81

	Zalew żądań SNMP	9	99,87	0,065	0,065
	Zalew żądań NTP	8	99,77	0,2	0,03
	Zalew żądań HTTP	10	99,69	0,16	0,15
	Zalew żądań DNS	10	99,04	0,16	0,8
	Średnia	8	99,5	0,151666667	0,348333333

Tabela 6 Dokładność wykrywania ataków DDoS

Zamierzony rezultat:

Wykorzystanie uczenia maszynowego nadzorowanego w celu wykrywania potencjalnych ataków sieciowych, w szczególności ataków typu DDoS. Wykorzystanie w tym module zestawów danych zarówno najnowszych (2021, 2019, 2017), jak i zestawu już „klasycznego" (NSL-KDD) i stworzenie metody wnioskowania większościowego na podstawie wytrenowanych wcześniej podzbiorczych lasów losowych, które stanowią nowy zbiór danych dla głównego lasu losowego pozwoliło na uzyskanie dużej dokładności wykrywanych ataków przy jednoczesnym uwrażliwieniu wykrywania na większe spektrum ataków sieciowych niż w publikacjach opierających się na jednym zestawie danych (często nieaktualnym).

Jednocześnie mając świadomość ataków typu zero-day w kolejnym module Systemu opartym na metodach głębokiego uczenia uwzględniamy także dane o atakach zebrane w okresie 2019-2021 z firmowej sieci oraz aktualnie przechwytywany ruch sieciowy. Przekazywanie informacji o wykrywanych na poziomach poszczególnych modułów do Modułu MODSAI pozwala na aktualizację baz ataków oraz dalsze uczenie w trakcie życia systemu. Moduł ten pozwala również na wymianę informacji o atakach z innymi współpracującymi kontrolerami, co daje podstawę do przypuszczeń iż System w ten sposób zaprojektowany będzie na bieżąco aktualizował dane o atakach oraz odpowiednio reagował na nowe zagrożenia.

Moduł MOD3 IDS_GRUoGPU

Podobnie jak w poprzednich testach użyto oprogramowania Scapy oraz PyLoris wraz z KALI Linux do generowania ruchu normalnego oraz przeprowadzania ataków. W celu rozszerzenia rodzajów ataków użyto dodatkowych narzędzi (LOIC, HOIC, DDoSIM, D-ITG). Dodatkowo wykorzystano narzędzia TCPReplay oraz SNORT i Wireshark w celu określenia wzorca ruchu sieciowego normalnego oraz wzmocnienia uczenia modułu (sprzężenie zwrotne).

Wzorzec normalnego ruchu został opracowany na podstawie przechwyconego wcześniej ruchu z firmowych komputerów w trakcie normalnego działania firmy. Dane te były zbierane na przestrzeni roboczego tygodnia, a następnie zostały przeanalizowane i na podstawie analizy został przygotowany wzorzec normalnego ruchu wewnątrz sieci. Użycie oprogramowania TCPReplay umożliwiło wykorzystanie przechwyconego wcześniej ruchu dla symulacji rzeczywistego ruchu sieciowego dla lepszego odwzorowania rzeczywistości.

Jako cel ataków wybrano 10 hostów – serwerów wystawiających usługi: stron web zewnętrznej i wewnętrznej, serwer ftp, serwer udziałów sieciowych oraz 5 hostów symulujących laptopy pracowników.

Dla symulacji ataków generowaliśmy je z dziesięciu hostów na 10 docelowych serwerów oraz 5 docelowych hostów. 8 maszyn wirtualnych z systemem KALI Linux zostało wykorzystanych do przeprowadzania wzrastającego ataku DDoS SYN Flood na 2 serwery hostujące usługi http. Zaczynaliśmy od 50000 pakietów/sekundę i dwóch hostów atakujących, stopniowo zwiększając liczbę pakietów, oraz dołączając kolejne hosty atakujące. Sprawdziliśmy dokładność wykrywania ataku przy różnych konfiguracjach hostów atakujących.

Następnie korzystając z kolejnych narzędzi przeprowadzono ataki innych typów.

Przeprowadzone testy skuteczności wykrywania ataków

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
20	Zalew żądań ICMP	4	98,2	0,9	0,9
	Zalew żądań PING	4	98,2	0,9	0,9
	Zalew żądań TCP SYN	9	98,9	0,5	0,6
	Zalew żądań UDP	4	98,3	1,2	0,5
	Żądania IP Null	9	97,11	0	2,89
	Zalew żądań SNMP	8	98,39	0	1,61
	Zalew żądań NTP	8	97,34	2	0,66
	Zalew żądań HTTP	9	98,65	0,35	1
	Zalew żądań DNS	9	98,22	0,78	1
	Średnia	7,11111111	98,14555556	0,736666667	1,117777778

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
	Zalew żądań ICMP	3	98,81	0,595	0,595
	Zalew żądań PING	3	98,82	0,59	0,59

30	Zalew żądań TCP SYN	8	99,11	0,5	0,39
	Zalew żądań UDP	7	98,89	0,555	0,555
	Żądania IP Null	7	98,65	0	1,35
	Zalew żądań SNMP	7	99,4	0	0,6
	Zalew żądań NTP	7	98,75	1	0,25
	Zalew żądań HTTP	8	99,57	0,23	0,2
	Zalew żądań DNS	8	99	0,5	0,5
	Średnia	6,44444444	99	0,441111111	0,558888889

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
45	Zalew żądań ICMP	3	98,9	0,55	0,55
	Zalew żądań PING	3	98,9	0,55	0,55
	Zalew żądań TCP SYN	7	99,39	0,305	0,305
	Zalew żądań UDP	3	99,45	0	0,55
	Żądania IP Null	6	99,45	0	0,55
	Zalew żądań SNMP	7	99,85	0,1	0,05
	Zalew żądań NTP	7	99,81	0,1	0,09
	Zalew żądań HTTP	8	99,75	0,2	0,05
	Zalew żądań DNS	8	99,05	0,15	0,8
	Średnia	5,77777778	99,39444444	0,217222222	0,388333333

Natężenie ataku (%)	Typ ataku DDoS	Czas klasyfikacji (ms)	Dokładność wykrycia (%)	Fałszywie pozytywne (%)	Fałszywie negatywne (%)
50	Zalew żądań ICMP	3	99,54	0,23	0,23
	Zalew żądań PING	3	99,54	0,23	0,23
	Zalew żądań TCP SYN	7	99,54	0,23	0,23
	Zalew żądań UDP	3	99,55	0	0,45
	Żądania IP Null	6	99,25	0	0,75
	Zalew żądań SNMP	7	99,91	0	0,09
	Zalew żądań NTP	7	99,81	0,1	0,09
	Zalew żądań HTTP	7	99,75	0,2	0,05
	Zalew żądań DNS	7	99,12	0,08	0,8
	Średnia	5,55555556	99,55666667	0,118888889	0,324444444

Tabela 7 Dokładność wykrywania ataków DDoS

Zamierzony rezultat:

Przedmiotem testów w środowisku testowym było zarówno odwzorowanie w sposób najbardziej zbliżony rzeczywistej struktury sieciowej środowiska produkcyjnego oraz przeprowadzenie różnych

ataków sieciowych w celu weryfikacji działania poszczególnych modułów systemu detekcji oraz wdrożenie całościowe systemu wraz z testami.

W tym celu odwzorowano środowisko sieciowe oraz przygotowano odpowiednie maszyny wirtualne. Przeprowadzone testy poszczególnych modułów udowodniły ich skuteczność przy niskim odsetku fałszywych detekcji. Jednocześnie wirtualizacja środowiska pozwoliła w elastyczny sposób scalić elementy systemu w całość i przeprowadzić testy na pełnym systemie.

Przeprowadzone testy udowodniły skuteczność działania całego systemu, a w szczególności doprowadziły do zmian w trzecim module w celu poprawy szybkości oraz jakości jego działania.

4.3. ZESTAWIENIE WYNIKÓW

Testy w systemie produkcyjnym

Testy w systemie produkcyjnym obejmowały zarówno implementacje całego systemu w środowisku sieci produkcyjnej, jak i ponowne przeprowadzenie testów stosowanych w środowisku testowym celem weryfikacji poprawności wdrożenia oraz działania systemu w środowisku rzeczywistym.

Skuteczność systemu w środowisku rzeczywistym nie odbiegała od skuteczności w środowisku testowym, co udowodniło jego przydatność oraz poprawność przyjętej metodologii.

Zaleta systemu jest jego modularna i hybrydowa budowa umożliwiająca rozbudowę zarówno poszczególnych modułów (MOD2 może zostać w łatwy sposób rozszerzony o kolejne bazy ataków, co podniesie jego skuteczność w przypadku nowo pojawiających się zagrożeń), jak i całego systemu o kolejne moduły.

Wirtualizacja komponentów systemu pozwala wdrażać go w różnorodnych środowiskach sieciowych. Wprowadzenie komponentu komunikacyjnego z innymi kontrolerami SDN do Modułu SAI pozwala na bezpieczna komunikacje z innymi kontrolerami SDN celem wymiany informacji o atakach.

Natężenie ataku (%)	Typ ataku DDoS	MOD1		MOD2		MOD3	
		Czas klasyfikacji (ms)	DW (%)	Czas klasyfikacji (ms)	DW (%)	Czas klasyfikacji (ms)	DW (%)
25	Zalew żądań ICMP	2	96,46	4	98,80	3	98,80
	Zalew żądań PING	2	96,45	4	98,80	3	98,81
	Zalew żądań TCP SYN	2	99,60	11	99,00	8	99,10
	Zalew żądań UDP	2	97,46	3	98,84	7	98,87
	Żądania IP Null	-	-	9	98,60	7	98,64
	Zalew żądań SNMP	-	-	8	98,37	7	99,38

	Zalew żądań NTP	-	-	11	99,54	7	98,54
	Zalew żądań HTTP	-	-	11	99,54	8	99,56
	Zalew żądań DNS	-	-	11	98,97	8	99,00
	Średnia	2	100	7,89	98,81	6,44	98,97

Natężenie ataku (%)	Typ ataku DDoS	MOD1		MOD2		MOD3	
		Czas klasyfikacji (ms)	DW (%)	Czas klasyfikacji (ms)	DW (%)	Czas klasyfikacji (ms)	DW (%)
50	Zalew żądań ICMP	2	100	5	99,50	3	99,55
	Zalew żądań PING	2	100	5	99,50	3	99,55
	Zalew żądań TCP SYN	2	100	11	99,49	7	99,55
	Zalew żądań UDP	2	100	4	99,49	3	99,55
	Żądania IP Null	-	-	9	99,19	6	99,25
	Zalew żądań SNMP	-	-	9	99,87	7	99,91
	Zalew żądań NTP	-	-	10	99,78	7	99,81
	Zalew żądań HTTP	-	-	10	99,10	7	99,76
	Zalew żądań DNS	-	-	10	99,04	8	99,00
	Średnia	2	100	8,11	99,44	5,67	99,55

Tabela 8. Dokładność wykrywania ataków DDoS Systemu Strainer

5. PODSUMOWANIE I WNIOSKI

Celem rozprawy doktorskiej było opracowanie systemu informatycznego implementacji algorytmów synchronizacji urządzeń sieciowych dla środowiska programowalnych sieci komputerowych SDN, służących do detekcji ataków sieciowych z rodziny DDoS. Zadnie to realizowane było w ramach „doktoratu wdrożeniowego", co dodatkowo wymagało od autora wdrożenia opracowanego, gotowego systemu bezpieczeństwa w środowisku produkcyjnym przedsiębiorstwa Silsense Technologies S. A. w którym doktorant realizuje zadanie projektowe.

Po wnikliwym przeglądzie literaturowym w obszarze rozwiązań sieci komputerowych SDN i metod obrony przed atakami DDoS, autor pracy zaprojektował, opracował, przetestował i wdrożył system sieciowej obrony pod nazwą „STRAINER". System STRAINER oparty jest o budowę modułową, której główną siłą są autorskie moduły algorytmiczne służące do detekcji zagrożeń sieciowych w postaci ataków DDoS.

Autor pracy zgodnie z założeniami, rozpoczął swoje badania od analizy istniejących metod wykrywania ataków sieciowych typu DDoS z wykorzystaniem środowiska sieci SDN. Po wnikliwej analizie literaturowej zaproponował system informatyczny przedstawiony w rozdziale 3. System STRAINER jest w pełni autorskim rozwiązaniem służącym do detekcji i mitygacji ataków sieciowych DDoS. Autor w rozdziale 3.2 umiejscowił system na „mapie drogowej" tego typu rozwiązań. Po wielu wstępnych testach w środowisku laboratoryjnym, autor zaprojektował system z trzema kluczowymi modułami algorytmicznej detekcji ataków sieciowych DDoS.

Pierwszy moduł oparty jest o statystyczną analizę ruchu sieciowego z wykorzystaniem funkcji entropii na bazie określonych parametrów sieciowych uzyskanych z kontrolera sieci SDN za pomocą protokołu OpenFlow. Moduł drugi opiera swoją analizę na mechanizmie uczenia maszynowego ML z wykorzystaniem zmodyfikowanej funkcji Random Forest (lasów losowych). Analiza przeprowadzana jest na bieżącym ruchu sieciowym pozyskanym ze środowiska produkcyjnego. Klasyfikacja odbywa się w oparciu o wyuczone zbiory lasów losowych na podstawie danych wzorcowych z Kanadyjskiego Instytutu Cyberbezpieczeństwa. Trzeci moduł to moduł najbardziej zawansowany oparty jest o mechanizm głębokiego uczenia maszynowego zaimplementowany w oparciu zmodyfikowaną metodę bramkowanych sieci rekurencyjnych GRU z mechanizmem sprzężenia zwrotnego, wykorzystującego analizę z oprogramowania SIEM SNORT. Siła rozwiązania tkwi w zawansowanym przetwarzaniu danych, akcelerowanym przez jednostki obliczeniowe GPU. Moduł ten potrafi wykrywać nieznane dotychczas, złożone typy ataków DDoS. System posiada oprócz tego moduł kolekcjonera MODKOL, służący do akwizycji danych o bieżącym ruchu sieciowym z badanego środowiska sieciowego. Ponad to posiada moduł MODSAI służący do komunikacji z kontrolerem sieci SDN, przesyłania informacji do innych lub wyniesionych obszarów sieci (innych kontrolerów sieci SDN). W zależności od scenariusza wdrożenia systemu STRAINER, środowisko sieciowe zawiera urządzenia sieciowe wykonawcze odpowiednio dla warstwy L2, L3 jak również systemy UTM, firewall i VPN. Moduły algorytmiczne i sterujące zaimplementowane są jako oprogramowanie funkcjonalne systemy w zwirtualizowanym środowisku sieciowym.

Hybrydowa i modułowa konstrukcja systemu STRAINER, w obszarze analizy i wykrywania ataków sieciowych DDoS działa dosłownie jak sito. Zaimplementowany system pozwala na rozbudowę o nowe moduły funkcjonalne jak i dodatkowe schematy analizy ruchu sieciowego. Istnieją różne warianty jego wdrożenia i implementacji. System detekcji jest w pełni automatyczny, w zakresie podejmowanych decyzji o klasyfikacji ruch sieciowego (ruch zainfekowany/ ruch poprawny). System jednakże wymaga precyzyjnego strojenia jak i okresowej aktualizacji wzorcowych baz danych, dla moduł analizy uczenia maszynowego (MOD2, MOD3). Fakt ten wynika z dynamiki pojawiania się nowych zagrożeń w postaci nowych rodzajów ataków typu DDoS.

Na potrzebę wykonania niezbędnych badań i pomiarów udowadniających tezę pracy, autor zbudowała środowisko sieciowe testowe. Środowisko szczegółowo zostało przedstawione w rozdziale 5 opisującym wszystkie brane pod uwagę aspekty testów. Dobór testów wynikał ze specyficznej budowy modułów analizujących (MOD1-MOD3). Wszystkie testy wykonane w badanym systemie mają jednolity charakter. Uzyskane wyniki ukazują różnicę w działaniu poszczególnych modułów. Można zauważyć, iż w zależności od rodzaju generowanego ataku DDoS, poszczególne moduły wykrywają atak ze zróżnicowaną skutecznością i czasem detekcji. Istotnymi badanymi parametrami systemu były: dokładność wykrywania ataku i czas klasyfikacji.

W wyniku przeprowadzonych badań, należy stwierdzić iż skuteczność działania systemu otrzymujemy tylko w przypadku zastosowania wszystkich trzech modułów systemu łącznie. Wynika to z faktu, iż MOD1 jest modułem o bardzo dużej szybkości działania, jednak o ograniczonym spektrum wykrywania ataków. Nie jest w stanie wykryć nowych, bardziej złożonych rodzajów ataków DDoS. Moduł MOD2 natomiast wykrywa szersze spektrum ataków DDoS (na podstawie dostarczonych danych wzorcowych uczących). Ten moduł jest bardzo skuteczny, jednak ograniczony jest aktualnością dostarczonych baz uczących algorytm wykrywania. Natomiast MOD3 posiada największe spektrum wykrywanych ataków DDoS, z możliwym wykrywaniem ataków typu zero-day. Jednocześnie moduł ten ustępuję co do czasu klasyfikacji w stosunku do MOD1. Wykrycie typowego ataku przez MOD3 jest często potwierdzeniem wcześniejszego wykrycia w MOD1. Oczywiście decyzję o mitygacji wykrytego ataku podejmowane są w MODSAI i przekazywane do kontroler SDN celem dalszego podjęcia działań.

Przeprowadzone badania w środowisku testowym w pełni potwierdzają skuteczność i akceptowalną szybkość działania klasyfikacji ataków typu DDoS. Warto wspomnieć, iż testy w środowisku laboratoryjnym odbywały się przy wykorzystaniu zebranego w okresie 12 miesięcy ruchu rzeczywistego ze środowiska produkcyjnego. Dodatkowo po wdrożeniu systemu autor przeprowadził testy sieciowe wykrywania ataków DDoS w środowisku produkcyjnym. Wyniki testów nie odbiegały znacząco od laboratoryjnych. Sposób testowania w środowisku produkcyjnym był analogiczny do testów w środowisku laboratoryjnym, natomiast testowane środowisko sieciowe było środowiskiem rzeczywistym, działającym w miejscu wdrożenia projektu.

Zaprezentowane rozwiązanie w postaci systemu wykrywania i mitygacji ataków DDoS STRAINER, przedstawione wyniki badań i testów jednoznacznie udowadniaj skuteczność opracowanego rozwiązania. Jednocześnie warto podkreślić, iż zaproponowany system można w dalszym ciągu dostosowywać, rozbudowywać w zależność od specyficznych obszarów zastosowań, np. dla sieci 5G, sieci automatyki przemysłowej, sieci IoT. System może też stanowić element dla systemów wykrywania incydentów sieciowych klasy operatorskiej. W pracy skoncentrowano się głównie na detekcji ataków DDoS, a w mniejszej części na ich mitygacji. Mitygacja ataków DDoS

możliwa jest głownie w systemach operatorskich, z wykorzystaniem zasobów sieci teleinformatycznej. Wskazaniem rozbudowy systemu było by dostosowanie go do klasy operatorskiej. Ze względu na hybrydową budowę warto również rozważyć opracowanie nowych modułów detekcji (nowe algorytmy) w kierunku ataków DDoS jak również innych rodzajów ataków sieciowych. System został zaprojektowany dla środowiska sieciowego autonomicznego IGP. System osiągnął pożądaną skuteczność mitygacyjną ataków typu DDoS w środowisku produkcyjnym firmy Silsense Technologies S.A. Po wdrożeniu systemu STRAINER w sieci komputerowej przedsiębiorstwa, zapory sieciowe typu FWNG w intranecie firmowym przez okres 8 miesięcy testów do momentu napisania tej pracy nie zanotowały żadnej próby ataku typu DDoS, co świadczy o skuteczności rozwiązania. Warte rozważenia jest wdrożenie systemu w środowisku brzegu sieci BGP.

Zaimplementowane rozwiązania w systemie STRAINER w pełni realizują postawioną w pracy tezę tj.:

„Możliwe jest opracowanie systemu mitygacji ataków sieciowych, opartego o rozwiązanie programowalnych sieci komputerowych SDN, wykorzystującego nowatorskie algorytmy sterowania i synchronizacji urządzeń, w tym analizę uzyskaną za pomocą algorytmów sztucznej inteligencji AI."

BIBLIOGRAFIA

[1] Kaspersky LAB 2021, https://statistics.securelist.com/. Data uzyskania informacji 17.05.2021

[2] IBM X-Force Report. https://newsroom.ibm.com/. Data uzyskania informacji 11.05.2021

[3] McGraw G., Morrisett G., 2000. Attacking Malicious Code: A Report to the Infosec Research Council. IEEE Softw. 17(5), pp. 33–41. ISSN 0740-7459.:10.1109/52.877857.

[4] Large Scale Communication Networks. https://www.ipam.ucla.edu/programs/longprograms/large-scale-communication-networks/

[5] M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. Hong, "A flowbased method for abnormal network traffic detection," in IEEE/IFIP Network Operations and Management Symposium (NOMS), vol. 1, Apr. 2004, pp. 599–612 Vol.1.

[6] Akamai, "Prolexic Quarterly Global DDoS Attack Report Q1 2014," Prolexic, Tech. Rep., 2014

[7] R. Hansen, J. Kinsella, and H. Gonzalez, "Slowloris HTTP DoS," 2009

[8] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Networkbased Defense Mechanisms Countering the DoS and DDoS Problems," ACM Comput. Surv., vol. 39, no. 1, Apr. 2007. [Online]. Available: http://doi.acm.org/10.1145/1216370.1216373

[9] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," IEEE Communications Surveys and Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[10] A. Belenky and N. Ansari, "On Deterministic Packet Marking," Comput. Netw., vol. 51, no. 10, pp. 2677–2700, Jul. 2007

[11] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," SIGCOMM Comput. Commun. Rev., vol. 30, no. 4, pp. 295–306, Aug. 2000.

[12] G. N. Koutepas, F. Stamatelopoulos, and V. Maglaris, "Distributed Management Architecture for Cooperative Detection and Reaction to DDoS Attacks," Journal of Network and Systems Management, vol. 12, pp. 73–94, 2004.

[13] D. Sterne, K. Djahandari, B. Wilson, B. Babson, D. Schnackenberg, H. Holliday, and T. Reid, "Autonomic Response to Distributed Denial of Service Attacks," in Recent Advances in Intrusion Detection, ser. LNCS. Springer Berlin Heidelberg, 2001

[14] Z. Zhang, F. Na¨ıt-Abdesselam, P. Ho, and Y. Kadobayashi, "Toward cost-sensitive self-optimizing anomaly detection and response in autonomic networks," Computers & Security, vol. 30, no. 6-7, pp. 525–537, 2011.

[15] „SDN Architecture Overview", ONF, Tech. Rep., 2013.

[16] A. Bates, K. Butler, A. Haeberlen, M. Sherr, and W. Zhou, "Let SDN Be Your Eyes: Secure Forensics in Data Center Networks," in Proceedings of the NDSS Workshop on Security of Emerging Network Technologies (SENT), Feb. 2014.

[17] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, and M. Tyson, "FRESCO: Modular Composable Security Services for Software-Defined Networks," in Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS), 2013.

[18] A. Alwabel, M. Yu, Y. Zhang, and J. Mirkovic, "SENSS: Observe and Control Your Own Traffic in the Internet," in Proceedings of the 2014 ACM Conference on SIGCOMM. New York, NY, USA: ACM, 2014, pp. 349–350

[19] Towards Autonomic DDoS Mitigation using Software Defined Networking Rishikesh Sahay, Gregory Blanc, Zonghua Zhang and Herve Debar SENT' 15, 8.02 2015, San Diego, A,USA Copyright 2015 Internet Society, http://dx.doi.org/10.14722/sent.2015.23004 Data uzyskania informacji 22.03.2021

[20] „Software-Defined Networking (SDN) Definition", Open Networking Foundation. https://www.opennetworking.org/sdn-definition/ Data uzyskania informacji 7.05.2021

[21] Thomas D. Nadeau & Ken Gray „Software Defined Networks" O'Reilly 2013

[22] Kriti Bhushan · B. B. Gupta Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment Journal of Ambient Intelligence and Humanized Computing (2019) 10:1985–1997

[23] https://www.commsbusiness.co.uk/features/software-defined-networking-sdn-explained/

[24] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. 2008,38, 69–74. [CrossRef]

[25] OpenFlow Switch Specification Version 1.5.1 (Protocol version 0x06) ONF

[26] R. Swami, M. Dave, and V. Ranga, "Software-defined Networking-based DDoS Defense Mechanisms," ACM Comput. Surv., vol. 52, no. 2, pp. 1–36, Apr. 2019.

[27] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 55–60.

[28] Tang, T. Software Defined Networking: Network Intrusion Detection System. Ph.D. Thesis, The University of Leeds, Leeds, UK, 2019.

[29] Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, ON, Canada, 8–10 July 2009.

[30] Ibrahim, L.M.; Basheer, D.T.; Mahmod, M.S. A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network. J. Eng. Sci. Technol. 2013, 8, 107–119.

[31] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network anomaly detection and classification via opportunistic sampling," IEEE Netw., vol. 23, no. 1, pp. 6–12, Jan. 2009

[32] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting Traffic Anomaly Detection Using Software Defined Networking," in Recent Advances in Intrusion Detection, vol. 6961, R. Sommer, D. Balzarotti, and G. Maier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 161–180. 2011

[33] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," Comput. Netw., vol. 62, pp. 122–136, Apr. 2014.

[34] R. Wang, Z. Jia, and L. Ju, "An Entropy-Based Distributed DDoS Detection Mechanism in Software-Defined Networking," in 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 2015, pp. 310–317. 2015

[35] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," Future Gener. Comput. Syst., vol. 89, pp. 685–697, Dec. 2018.

[36] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller Based Software-Defined Networking: A Survey," IEEE Access, vol. 6, pp. 15980–15996, 2018.

[37] P. Fiadino, A. D'Alconzo, M. Schiavone, and P. Casas, "Challenging Entropy-based Anomaly Detection and Diagnosis in Cellular Networks," in Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15, London, United Kingdom, pp. 87–88. 2015

[38] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016.

[39] T. Xing, D. Huang, L. Xu, C.-J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment," in 2013 Second GENI Research and Educational Experiment Workshop, Salt Lake, UT, USA, 2013, pp. 89–92.

[40] C. Jeong, T. Ha, J. Narantuya, H. Lim, and J. Kim, “Scalable network intrusion detection on virtual SDN environment,” in 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, Luxembourg, 2014, pp. 264–265.

[41] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, “TENNISON: A Distributed SDN Framework for Scalable Network Security,” p. 14, 2018.

[42] Q. Yan, W. Huang, X. Luo, Q. Gong, and F. R. Yu, “A MultiLevel DDoS Mitigation Framework for the Industrial Internet of Things,” IEEE Commun. Mag., vol. 56, no. 2, pp. 30–36, Feb. 2018.

[43] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran, “Athena: A Framework for Scalable Anomaly Detection in Software-Defined Networks,” in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 2017, pp. 249–260.

[44] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network Anomaly Detection: Methods, Systems and Tools,” IEEE Commun. Surv. Tutor., vol. 16, no. 1, pp. 303–336, 2014.

[45] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” in IEEE Local Computer Network Conference, Denver, CO, USA, 2010, pp. 408–415.

[46] A. Santos da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, “ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN,” in NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 2016, pp. 27–35

[47] D. Hu, P. Hong, and Y. Chen, “FADM: DDoS Flooding Attack Detection and Mitigation System in Software-Defined Networking,” in GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1–7.

[48] S. S. Bhunia and M. Gurusamy, “Dynamic attack detection and mitigation in IoT using SDN,” in 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, 2017, pp. 1–6.

[49] Lim, S.; Ha, J.; Kim, H.; Kim, Y.; Yang, S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In Proceedings of the Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014

[50] Giotis, K.; Androulidakis, G.; Maglaris, V. Leveraging SDN for Efficient Anomaly Detection and Mitigation on Legacy Networks. In Proceedings of the Third European Workshop on Software Defined Networks, Budapest, Hungary, 1–3 September 2014

[51] Jeong, J.; Seo, J.; Cho, G.; Kim, H.; Park, J.S. A Framework for Security Services Based on Software-Defined Networking. In Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Guwangiu, Korea, 24–27 March 2015

[52] International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056

[53] Fayaz, S.K.; Tobioka, Y.; Sekar, V.; Bailey, M. Bohatei: Flexible and Elastic DDoS Defense. In Proceedings of the USENIX Security Symposium, Austin, TX, USA, 10–12 August 2015; pp. 817–832

[54] Chen, C.C.; Chen, Y.R.; Lu, W.C.; Tsai, S.C.; Yang, M.C. Detecting amplification attacks with Software Defined Networking. In Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing, Taipei, Taiwan, 7–10 August 2017; pp. 195–201

[55] Liu, J.; Lai, Y.; Zhang, S. FL-GUARD: A Detection and Defense System for DDoS Attack in SDN. In Proceedings of the International Conference on Cryptography, Security and Privacy, Wuhan, China, 17–19 March 2017; ACM: New York, NY, USA, 2017; pp. 107–111

[56] Ibrahim, L.M.; Basheer, D.T.; Mahmod, M.S. A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network. J. Eng. Sci. Technol. 2013, 8, 107–119.

[57] Mukherjee, S.; Sharma, N. Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technol. 2012, 4, 119–128. [CrossRef

[58] Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Pittsburgh, PA, USA, 13–14 March 2016; pp. 21–26.

[59] Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access 2017, 5, 21954–21961. [CrossRef]

[60] Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell. 2018, 2, 41–50. [CrossRef]

[61] Kokila, R.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 IEEE Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; pp. 205–210.

[62] Phan, T.V.; Van Toan, T.; Van Tuyen, D.; Huong, T.T.; Thanh, N.H. OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks. In Proceedings of the 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), Ha Long, Vietnam, 27–29 July 2016; pp. 13–18.

[63] Winter, P.; Hermann, E.; Zeilinger, M. Inductive intrusion detection in flow-based network data using one-class support vector machines. In Proceedings of the 2011 IEEE 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 7–10 February 2011; pp. 1–5.

[64] AlEroud, A.; Alsmadi, I. Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach. J. Netw. Comput. Appl. 2017, 80, 152–164. [CrossRef]

[65] Mihai-Gabriel, I.; Victor-Valeriu, P. Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory. In Proceedings of the 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 19–21 November 2014; pp. 319–324.

[66] Mousavi, S.M.; St-Hilaire, M. Early detection of DDoS attacks against SDN controllers. In Proceedings of the 2015 IEEE International Conference on Computing, Networking and Communications (ICNC), Garden Grove, CA, USA, 16–19 February 2015; pp. 77–81.

[67] Trung et al. Combined hard thresholds of detection and a fuzzy inference system to detect the risk of DDoS attacks based on the real traffic characteristics (Distribution of Inter-arrival Time, Distribution of packet quantity per flow and Flow quantity to a server) under normal and attack states

[68] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, “Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments,” Comput. Netw., vol. 62, pp. 122–136, Apr. 2014.

[69] Dhawan, M.; Poddar, R.; Mahajan, K.; Mann, V. SPHINX: Detecting Security Attacks in Software-Defined Networks. In Proceedings of the Network and Distributed System Security (NDSS)’15, San Diego, CA, USA, 8–11 February 2015; Internet Society: Reston, VA, USA, 2015.

[70] Belyaev, M.; Gaivoronski, S. Towards load balancing in SDN-networks during DDoS-attacks. In Proceedings of the 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), Moscow, Russia, 27–29 October 2014

[71] Dao, N.N.; Park, J.; Park, M.; Cho, S. A feasible method to combat against DDoS attack in SDN network. In Proceedings of the International Conference on Information Networking (ICOIN), Siem Reap, Cambodia, 12–14 January 2015

[72] Rebecchi, F.; Boite, J.; Nardin, P.A.; Bouet, M.; Conan, V. Traffic monitoring and DDoS detection using stateful SDN. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017; IEEE: Berlin, Germany, 2017; pp. 1–2

[73] X. Liu, X. Yang, and Y. Xia, "NetFence: preventing internet denial of service from inside out," SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, Aug. 2010.

[74] A. Yaar, A. Perrig, and D. Song, "SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks," in Proceedings of the 2004 IEEE Symposium on Security and Privacy, May 2004, pp. 130–143.

[75] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting Network Architecture," SIGCOMM Comput. Commun. Rev., vol. 35, no. 4, pp. 241–252, Aug. 2005.

[76] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: RouterBased Defense Against DDoS Attacks," in Proceedings of Network and Distributed System Security Symposium (NDSS), 2002.

[77] A. Mahimkar, J. Dange, V. Shmatikov, H. Vin, and Y. Zhang, "dFence: Transparent Network-based Denial of Service Mitigation," in Proceedings of the 4th USENIX Conference on Networked S-Systems Design Implementation (NSDI). Berkeley, CA, USA: USENIX Association, 2007, pp. 24–24.

[78] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying Middlebox Policy Enforcement Using SDN," SIGCOMM Comput. Commun. Rev., vol. 43, no. 4, pp. 27–38, Aug. 2013.

[79] François, J.; Aib, I.; Boutaba, R. FireCol: A collaborative protection network for the detection of flooding DDoS attacks. IEEE/ACM Trans. Netw. (TON) 2012, 20, 1828–1841

[80] Rashidi, B.; Fung, C. CoFence: A collaborative DDOS defence using network function virtualization. In Proceedings of the 12th International Conference on Network and Service Management (CNSM), Montreal, QC, Canada, 31 October–4 November 2016; IEEE: Berlin, Germany, 2016; pp. 160–166

[81] Rahamatullah Khondoker, Adel Zaalouk, Ronald Marx, Kpatcha Bayarou, Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers 2014 World Congress on Computer Applications and Information Systems (WCCAIS) 2014 Hammamet, Tunisia

[82] Lusani Mamushiane; Albert Lysko; Sabelo Dlamini; A comparative evaluation of the performance of popular SDN controllers, April 2018 Wireless Days (WD), Dubai, United Arab Emirates

[83] Radware, raport za 2014-2015

[84] J. Homa, A. Nawrat, K. Daniec, K. Jędrasiak, R. Koszewski; The Concept of a Distributed Security System Based on SDN to Improve the Cybersecurity of the Corporate Network, konferencja ACIDS 2020, Puket

[85] J. Homa, A. Nawrat, K. Daniec, A. Starczewska; Analysis of solutions and proposing dynamically reconfigured network architecture (SDN) for deep machine learning systems based on Edge Computing technology. ICCCN 08.2021, Wenecja

[86] www.microsoft.com; Data uzyskania informacji 17.05.2022

[87] https://downdetector.pl/ Data uzyskania informacji 05.2022

[88] Mynarski S. (red.) Elementy teorii systemów i informacji, Akademia Ekonomiczna w Krakowie, Kraków 1989
[89] Górecki H. Teoria informacji, Wyższa Szkoła Informatyki w Łodzi, Łódź 2006
[90] Shannon C E. Prediction and entropy of printed English. In Bell system technical journal, 1951, 30(1): 50-64
[91] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm Data uzyskania informacji 06.2021
[92] Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). Classification And Regression Trees (1st ed.). Routledge. https://doi.org/10.1201/9781315139470
[93] https://www.unb.ca/cic/ Data uzyskania informacji 04.08.2021
[94] http://www.iscx.ca/ Data uzyskania informacji 05.09.2021
[95] Quinlan, J Ross. 1993. C4. 5: Programs for Machine Learning. Elsevier
[96] Mininet http://mininet.org/blog/2021/02/28/announcing-mininet-2-3-0/ Data uzyskania informacji .05.2022
[97] POX https://noxrepo.github.io/pox-doc/html/ Data uzyskania informacji 10.03.2021
[98] OpenDylight https://docs.opendaylight.org/en/latest/downloads.html Data uzyskania informacji 11.03.2022
[99] Ryu https://github.com/faucetsdn/ryu Data uzyskania informacji 09.05.2022
[100] Python https://www.python.org/ Data uzyskania informacji 19.02.2022
[101] Kali Linux https://www.kali.org/ Data uzyskania informacji 17.04.2022
[102] https://sourceforge.net/projects/loic/files/latest/download Data uzyskania informacji 17.05.2022
[103] https://sourceforge.net/projects/highorbitioncannon/ Data uzyskania informacji 1.04.2022
[104] https://packetstormsecurity.com/files/112856/HULK-Http-Unbearable-Load-King.html Data uzyskania informacji 25.05.2022
[105] https://motoma.io/pyloris/ Data uzyskania informacji 05.2022
[106] https://github.com/mitre/caldera Data uzyskania informacji 05.2022
[107] https://sourceforge.net/projects/hyenae-ng/ Data uzyskania informacji 05.2022
[108] https://scapy.readthedocs.io/en/latest/introduction.html Data uzyskania informacji 05.2022
[109] https://tcpreplay.appneta.com/ Data uzyskania informacji 05.2022
[110] http://traffic.comics.unina.it/software/ITG/documentation.php Data uzyskania informacji 05.2022
[111] https://insecure.org Data uzyskania informacji 05.2022
[112] https://wireshark.org Data uzyskania informacji 05.2022
[113] https://www.snort.org Data uzyskania informacji 05.2022

DODATEK A. ZASTOSOWANIA SIECI SDN

Współczesne systemy teleinformatyczne, a w szczególność systemy chmury obliczeniowej czy sieci komórkowych 5G nie istniały by bez zastosowania programowalnych sieci komputerowych SDN. Sieci SDN są mechanizmem instrumentalizacji zadań w czołowych rozwiązaniach gigantów z branży IT takich jak AMAZON, Google, czy Microsoft.

Ze względu na sposób zastosowania można wyodrębnić następujące obszary IT dla sieci SDN:

- Infrastruktura sieciowa
- Chmura obliczeniowa, wirtualizacja
- Komórkowe Sieci 5G
- Analiza danych sieciowych AI, wykrywanie ataków
- Systemy cyberbezpieczeństwa typu SOC, CERT, CSIRT

Infrastruktura sieciowa

Obszar **infrastruktury sieciowej**, czyli kręgosłupa dla sieci LAN/WAN był pierwszym i naturalnym miejscem pojawienia się technologii sieci programowalnych SDN. W miejsce klasycznych urządzeń warstwy L2/L3 modelu OSI tj. przełączników i routerów pojawiły się przełącznik i routery wspierające technologię SDN. Rewolucja w tym obszarze polega na wydzieleniu płaszczyzny sterowania i danych. Taka migracja w warstwach dostępowych i routingu sieci wraz z wydzieleniem części zarządzania siecią w postaci kontrolera SDN umożliwiła centralne zarządzanie i monitorowanie rozbudowanych środowisk korporacyjnych i kampusowych sieci LAN i WAN.

Chmura obliczeniowa, wirtualizacja

(Ang. *Cloud computing*) – czyli **chmura obliczeniowa**, technologia wywodząca się wprost z wirtualizacji systemów serwerowych w dzisiejszym wydaniu dostawców tj. AMAZON, Google czy Microsoft, w dużej mierze opiera się na technologii SDN w obszarze jej instrumentalizacji. Stosowane modele usług chmury obliczeniowej tj. IaaS, PaaS, SaaS (infrastruktura jako usługa, platforma jako usługa, oprogramowanie jako usługa) i inne, prezentują olbrzymią dynamikę w generowaniu docelowych środowisk i funkcjonalności na potrzeby realizacji określonych wymagań biznesowych. Tego typu możliwości za pomocą tradycyjnego formularza w HTML-u „wyklikujemy” na witrynach dostawców usług chmurowych. W przeciągu kilku chwil po zdefiniowaniu potrzeb za pomocą formularza, powstaje po stronie dostawcy „w chmurze” gotowe środowisko do realizacji naszych potrzeb. Środowiska te są wypełni funkcjonalnymi infrastrukturami sieciowo-programowymi wyposażonymi w wirtualne urządzenia sieciowe warstw L2/L3/L4, pule serwerów (maszyn wirtualnych) z gotowymi obrazami systemów operacyjnych z pełni działającymi funkcjami sieciowymi, przełączania, routingu, z translacją DNS, ARP itp. Wszystkie te usługi w tradycyjnych sieciach wymagają

żmudnej i długo trwałej konfiguracji. W opisywanym przykładzie dzięki instrumentalizacji opartej na technologii SDN usługi chmury powstają w ułamkach sekund i są gotowe do działania od zaraz.

Komórkowe Sieci 5G

Współczesne systemy telefonii komórkowej oparte są na rozwiązaniach klasy 5G (sieć piątej generacji). Systemy te ukierunkowane są głównie na ultra szybka transmisję danych. Sama komunikacja głosowa zasadniczo nie różni się od tej w sieciach komórkowych 4-tej czy 3-generacji. Natomiast transmisja danych komórkowych to już zupełnie inna sprawa. W tym obszarze transmisja odbywa się z wykorzystaniem technologii LTE, ale sama transmisja to tylko jeden element sieci 5-tej generacji. W sieciach 5G zastosowano technologię edge computing, czyli przetwarzania „na brzegu sieci". Technologia ta sprowadza się do przetwarzania/uzyskiwania niezbędnych danych dla klienta jak najbliżej brzegu sieci komórkowej tj. w stacji BTS operatora komórkowego. W tym właśnie miejscu pojawia się rozwiązanie SDN, bez którego sieci komórkowe 5-ej generacji by sobie nie poradziły. SDN wspomaga procesy „dostarczania" pożądanego kontentu tj. strona internetowa, transmisja strumieniowa (np. NETFLIX), informacja z portalu, i inne. Głównym kierunkiem rozwoju jest dostarczanie informacji z sieci IoT (Internetu rzeczy), zaimplementowanych w określonym obszarze, komórce sieci 5G np. z sieci przemysłowych, czy urządzeń mobilnych lub sieci IoT w samochodach znajdujących się w zasięgu. Dostęp do takich danych „lokalnych" (w obrębie BTS) luba zewnętrznych dostępnych z poza BTS czyli z sieci Internet możliwy jest dzięki złożonej analizie AI, przetwarzaniu brzegowemu w stacji operatora BTS i instrumentalizacji sieci SDN. Tego typu korelacja danych nie była by możliwa bez sieci SDN analogicznie do rozwiązań chmury obliczeniowej.

Analiza danych sieciowych AI, wykrywanie ataków

Intensywność współczesnych ataków sieciowych np. typu DDoS, dynamika przetwarzania danych BIG DATA, czy zestawianie/generowanie danych na potrzeby sieci 5G wymaga ultra szybkich rozwiązań w zakresie uzyskiwania rzetelnych i potwierdzonych informacji. Z pomocą przychodzą rozwiązania oparte o AI, uczenie maszynowe ML czy DL głębokie uczenie maszynowe, to technologie, bez których nie działają współczesne systemy bezpieczeństwa sieciowego, dynamiczne systemy informacji danych np. dla klientów banków. Rozwiązanie detekcji anomalii sieciowych np. w postaci wykrycia ataku sieciowego typu DDoS to jedno z topowych współczesnych metod eliminacji zagrożeń w sieci Internet. Analiza olbrzymiej ilość danych przy jednoczesnym błyskawicznym reagowaniu na zdarzenie w sieciach komputerowych możliwa jest tylko i wyłącznie w sposób automatyczny. Takie możliwości dają tylko sieci SDN w połączeniu z algorytmami AI. Współczesne systemy bezpieczeństwa sieciowego w Internecie opierają się na złożonej analizie przepływających danych i na skutecznym i trafnym sterowaniu przepływem pakietów opartym o silniki mechanizmów bezpieczeństwa i detekcji anomalii. Systemy tego typu nie były by wstanie działać bez mechanizmów dynamicznej rekonfiguracji sieci komputerowych za pomocą SDN. Rozwiązania analogiczne wykorzystywane są również w handlu za pośrednictwem sieci Internet, z wykorzystaniem np. glokalizacji i analizy BIG DATA.

Systemy cyberbezpieczeństwa SOC, CERT, CSIRT

Cyberbezpieczeństwo to termin bardzo popularny w dzisiejszych czasach, a oznacza zespół czynności mających na celu bezpieczeństwo informacji, transmisji, integralność, poufność. Na potrzeby zarządzania i analizy sytuacji w cyberprzestrzeni powstają centra operacji bezpieczeństwa tj. CERT, CSIRT czy SOC. Centra bezpieczeństwa oparte są na trzech filarach: ludziach, procedurach i

oprogramowaniu. To ostatnie jest najczęściej „szyte na miarę" tzn. jest tworzone pod specjalne zamówienia, mimo iż istnieją czołowi producenci tj. np. IBM z oprogramowaniem do warstwowej analizy zdarzeń, to każde centrum wsparcia typu SOC/CERT stosuje własne/dostosowane oprogramowanie oparte o na koncepcji SDN i wykorzystujące AI. SDN jest bardzo elastycznym często o otwartym dostępie rozwiązaniem, które umożliwia dostęp z poziomu API lub języków skryptowych tj. np. Phyton bezpośrednio do kontrolera sieci, którą to funkcjonalność wykorzystują zespoły cyberbezpieczeństwa. Za pomocą wyrafinowanych, często autorskich metod oprogramowanych w językach skryptowych, działających w postaci modułów funkcjonalnych dla kontrolerów sieci SDN, analizują, weryfikują lub walidują zadaną politykę bezpieczeństwa. Są to rozwiązania bez, których linie wsparcia w SOC czy CERT nie były by wstanie wykonywać skutecznie swojej pracy.