Implementacja i testy architektury bezpieczeństwa na poziomie 2 Systemu IIP

Jerzy Konorski[1], Krzysztof Cabaj[2], Zbigniew Kotulski[3], Paweł Szałachowski[3], Grzegorz Kołaczek[4], Jerzy Kasperek[5], Piotr Pacyna[5], Dominik Rzepka[5], Wojciech Romaszkan[5], Marcin Rupiński[5], Jehoszafat Zimnowoda[5], Andrzej Kamisiński[5], Paweł Rajda[5]

[1] Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska
[2] Instytut Informatyki, Politechnika Warszawska
[3] Instytut Telekomunikacji, Politechnika Warszawska
[4] Instytut Informatyki, Politechnika Wrocławska
[5] AGH Akademia Górniczo-Hutnicza w Krakowie

Streszczenie Projekt Inżynieria Internetu Przyszłości (IIP) rozwija koncepcję infrastruktury transmisyjnej, stanowiącej jednolity system komunikacyjny (System IIP) do obsługi strumieni danych pochodzących od trzech rodzajów sieci nazywanych Równoległymi Internetami. Architektura tego systemu obejmuje cztery poziomy, przy czym poziom 2 odpowiada za tworzenie i utrzymywanie łączy i węzłów wirtualnych. Niniejsze opracowanie przedstawia rozszerzony opis trzech linii obrony stanowiących architekturę bezpieczeństwa dla tego systemu na poziomie 2, która została opracowana w celu przeciwdziałania atakom opartym na wprowadzaniu obcego ruchu sieciowego do Systemu IIP, a także w celu przeciwdziałania manipulacji bądź fałszowaniu ruchu użytkowego i sygnalizacyjnego Systemu IIP. Przedstawiono wczesne doświadczenia z implementacji mechanizmów obronnych oraz omówiono wyniki testów przeprowadzonych w środowisku PL-LAB. Wyniki przedstawione w tym rozdziale są rozszerzoną i uaktualnioną wersją rezultatów zaprezentowanych w artykułach [1] i [2].

1 Wprowadzenie

Projekt Inżynieria Internetu Przyszłości (IIP) [3] rozwija koncepcję infrastruktury transmisyjnej oraz sieci nazywanych Równoległymi Internetami, tworzących łącznie jednolity system komunikacyjny Internetu Przyszłości. Infrastruktura transmisyjna System IIP jest przeznaczona do przesyłania jednostek danych IIP-PDU, nazywanych ramkami IIP. Każdy Równoległy Internet wykorzystuje łącza i węzły wirtualne wydzielone w oparciu o fizyczne łącza i przełączniki oraz definiuje swój własny stos protokolarny. Dwa spośród Równoległych Internetów: *Data Stream Switching* (DSS) i *Content Aware Network* (CAN) są rozwiązaniami określanymi jako post-IP, zaś trzeci jest oparty na protokole IPv6 ze wsparciem jakości usług (IPv6 QoS). Istnieje także czwarty rodzaj Internetu, *Management*, także oparty na protokole IPv6, będący wydzieloną siecią zarządzania Systemem IIP.

Architektura Systemu IIP zawiera cztery poziomy, przy czym poziom 2 odpowiada za tworzenie i utrzymywanie łączy i węzłów wirtualnych.

W celu przeciwdziałania zagrożeniom na tym poziomie zaproponowano mechanizmy bezpieczeństwa koncentrujące się na:

1. zagrożeniach zewnętrznych (intruzach) dokonujących manipulacji ruchu sieciowego, bądź wprowadzających nielegalny ruch do Systemu IIP, by zakłócić jego funkcjonowanie, jak też
2. wewnętrznych węzłach wirtualnych przejętych przez intruzów w celu fałszowania legalnego ruchu Systemu IIP dla osiągnięcia różnych szkodliwych celów.

Przyjmuje się zatem za możliwe do przeprowadzenia ataki metodą wprowadzania obcego ruchu (*traffic injection*), metodą powtarzania bądź zaburzania kolejności IIP-PDU (*replay/resequencing/reordering*), metodą zaburzania relacji czasowych w strumieniach IIP-PDU, np. poprzez przechwytywanie i przetrzymywanie ramek (*ruffling*) oraz metodą generacji fałszywego ruchu Systemu IIP (*forging*). W pracach [4] i [5] (por. też [6]) zaproponowano architekturę bezpieczeństwa dla poziomu 2 Systemu IIP, składającą się z trzech linii obrony. Pierwsza linia obrony odpowiada za zabezpieczenie ramek za pomocą kryptograficznej sumy kontrolnej HMAC oraz za obsługę numerów sekwencyjnych jednostek IIP-PDU dla zapobiegania atakom metodami *traffic injection* oraz *replay/resequencing/reordering*. Weryfikację HMAC przeprowadza się w węźle wirtualnym po odbiorze każdej ramki. Druga linia obrony wykrywa anomalne zachowania strumienia ruchu bądź węzła spowodowane atakami z użyciem metod zaburzania relacji czasowych lub generacji fałszywego ruchu. W tym celu obserwuje się zdarzenia związane z zagrożeniami bezpieczeństwa (*Security-Relevant Events*, SRE). SRE analizowane są przez umieszczone w węzłach transmisyjnych moduły wykrywania anomalii (*Local Anomaly Detection*, LAD), działające w ramach szerszej jednostki funkcjonalnej pod nazwą lokalny agent bezpieczeństwa (*Local Security Agent*, LSA). Anomaliom noszącym znamiona ataku przypisywane są miary liczbowe, które pozwalają przekształcić je w lokalne metryki reputacji węzła. Metryki te są przekazywane do trzeciej linii obrony, gdzie właściwy dla danego Równoległego Internetu główny agent bezpieczeństwa (*Master Security Agent*, MSA) wypracowuje na ich podstawie miary zaufania związane z każdym węzłem tego Internetu. Ponadto, wykorzystując moduł wykrywania anomalii o zasięgu całego Równoległego Internetu (*Parallel Internet-wide Anomaly Detection*, PIAD) MSA identyfikuje także anomalie o zasięgu szerszym niż zasięg pojedynczego węzła.

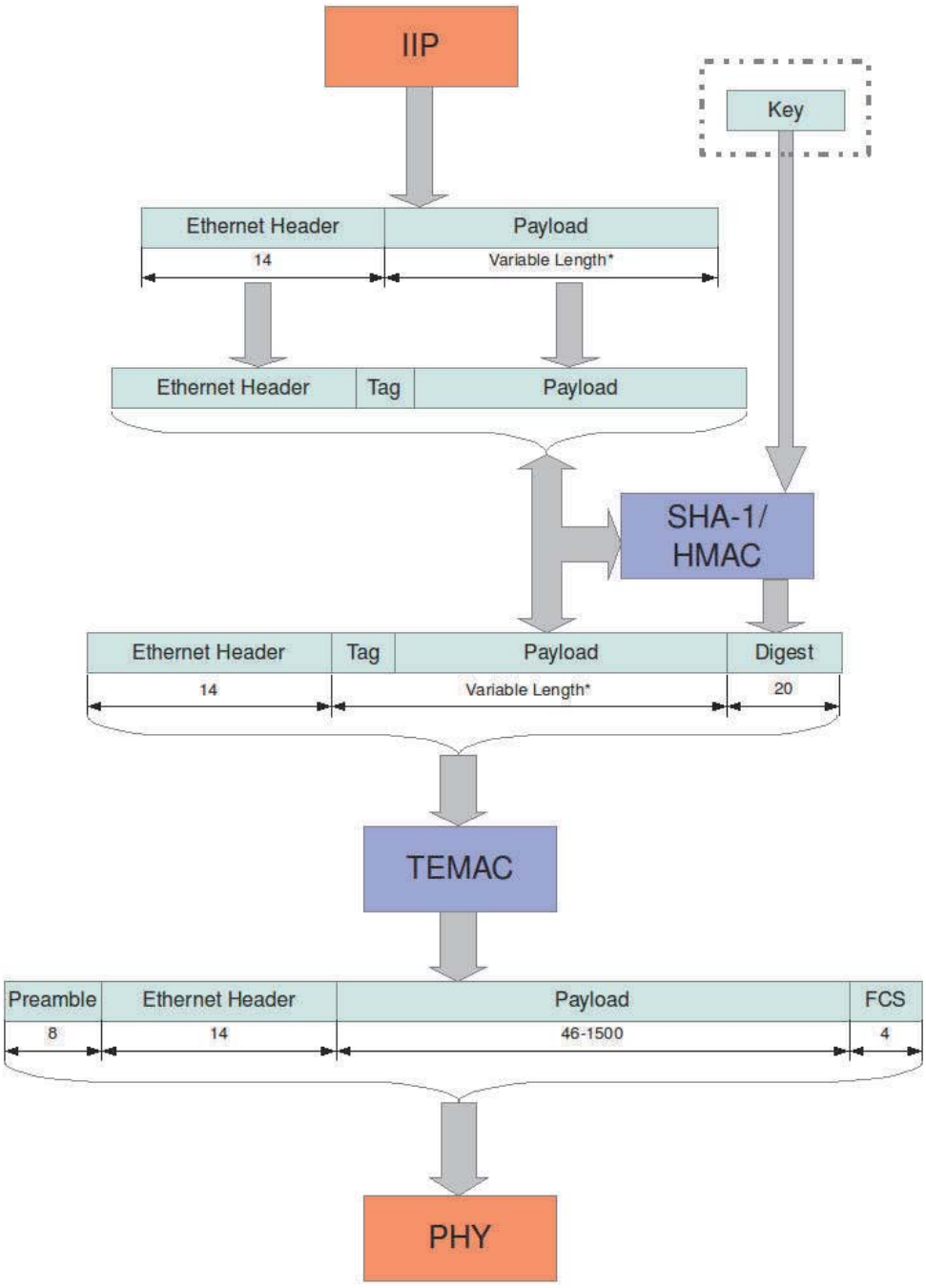
Niniejszy rozdział jest oparty na wczesnych doświadczeniach z integracji wybranych mechanizmów bezpieczeństwa z siecią zarządzania Systemu IIP przeprowadzonej w środowisku PL-LAB. Przedstawiamy wyniki uzyskane w zakresie implementacji i testowania działania wszystkich trzech linii obrony.

2 Implementacja i testowanie modułów HMAC-SHA-1

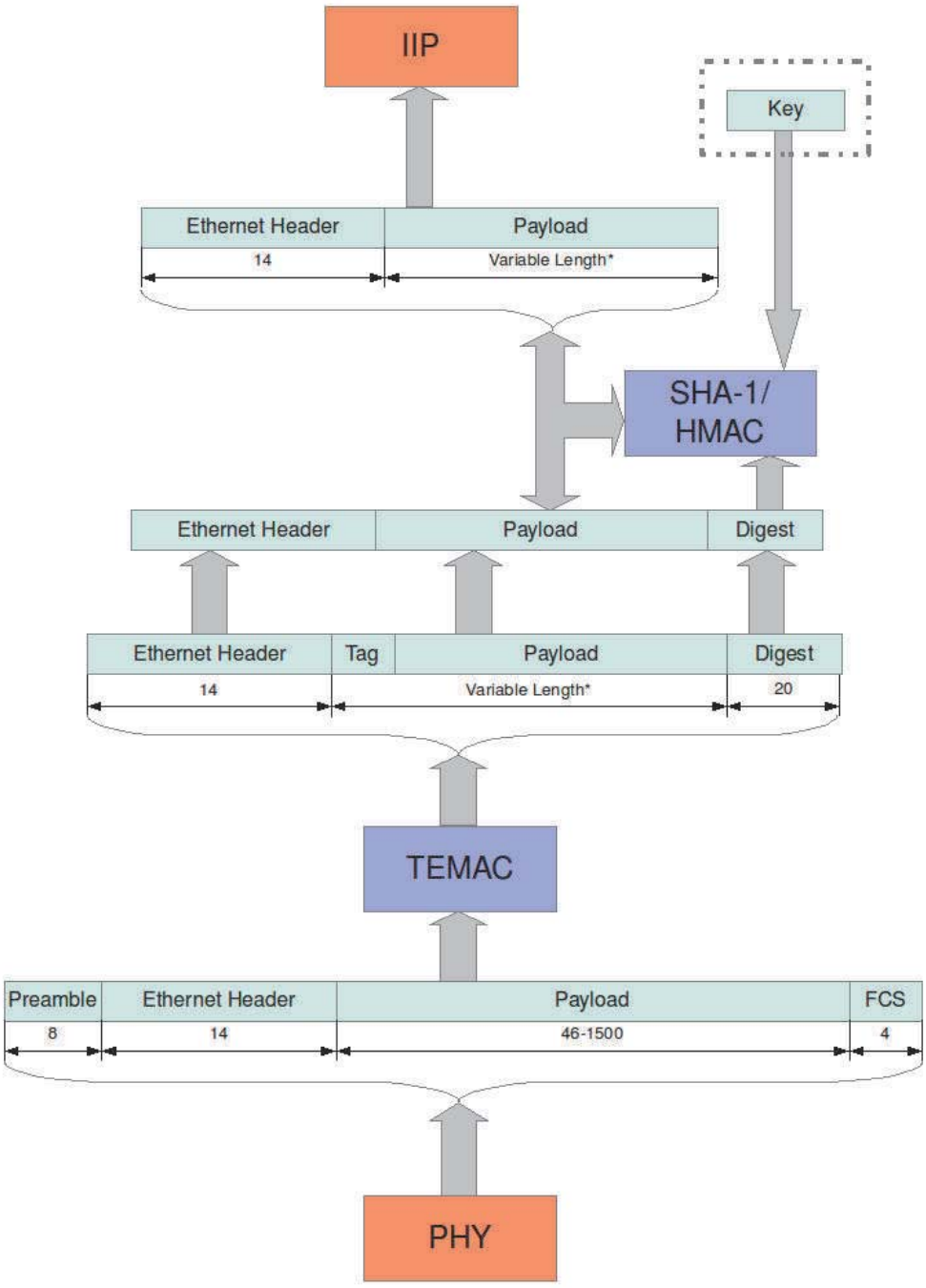
Sprzętowy koder oraz weryfikator HMAC-SHA-1 zapewnia jednolitą ochronę zagregowanego strumienia danych, który powstaje z multipleksacji strumieni danych przesyłanych przez trzy rodzaje sieci IPv6-QoS, CAN i DSS w systemie transmisyjnym IIP. Na poziomie warstwy łącza danych ochronie podlegają ramki IIP, zawarte w strukturze ramki Ethernet. Zaletą omawianego rozwiązania jest unifikująca forma ochrony strumienia danych w systemie. Jest niezależna od rodzaju aplikacji, od których pochodzą przenoszone dane, niezależna od specyfiki Równoległego Internetu oraz niezależna od stosowanych protokołów wyższych warstw. Implementacja zapewnia ochronę integralności ramki Ethernet wraz z nagłówkiem, w tym z polami nadawcy i odbiorcy. W powiązaniu z wprowadzoną w tym rozwiązaniu obsługą numerów sekwencyjnych omawiana implementacja umożliwia wykrywanie niektórych form ataków przeciwko strumieniowi danych, takich jak na przykład wprowadzanie obcego ruchu, fałszowanie przesyłanych danych oraz atak poprzez powtórzenia (*replay*). Przetwarzanie ramek przez moduł HMAC-SHA-1 polega na wytworzeniu sygnatury skrótu HMAC-SHA-1 i numeru sekwencyjnego ramki oraz na dołączeniu tych informacji do ramki po stronie nadawcy, a potem na weryfikacji poprawności tej ramki poprzez ponowne obliczenie sygnatury po stronie odbiorcy i porównanie jej z sygnaturą otrzymaną, a także sprawdzeniu numeru sekwencyjnego, co skrótowo ilustrują rysunki 1 i 2.

2.1 Architektura sprzętowa HMAC-SHA-1

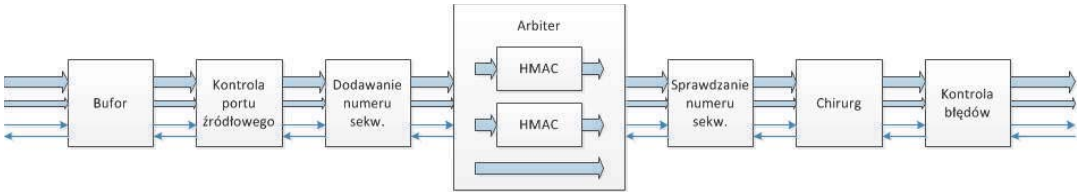
Warstwa sprzętowa odpowiedzialna za przetwarzanie danych została zaimplementowana w postaci rekonfigurowalnej logiki wewnątrz układu Virtex-II Pro znajdującego się na karcie NetFPGA 1G. Składa się ona z modułów napisanych w języku opisu sprzętu Verilog, połączonych szeregowo. Każda ramka Ethernet przesyłana wewnątrz struktury NetFPGA jest opatrzona co najmniej jednym nagłówkiem sterującym, zawierającym informacje niezbędne do prawidłowego przetwarzania tej ramki. Ramki przychodzące z czterech zewnętrznych portów fizycznych o przepustowości nominalnej 1 Gbit/s każdy oraz z czterech portów wirtualnych, dostępnych od strony systemu operacyjnego Linux przez magistralę PCI trafiają do tak zwanej ścieżki danych w układzie FPGA, gdzie znajdują się moduły odpowiedzialne za przetwarzanie ich zawartości. Każdy moduł może być dowolnie modyfikowany lub podmieniany, zależnie od potrzeb projektanta. Spójność całej struktury zapewnia opracowany na Uniwersytecie Stanforda standard komunikacji między modułami wewnątrz struktury NetFPGA. Dane są przesyłane wzdłuż ścieżki danych w 64-bitowych blokach razem z 8-bitowym polem sterującym. Pozwala to na realizację algorytmów o przetwarzaniu potokowym. Aby ułatwić komunikację z oprogramowaniem, karta NetFPGA posiada własny system rejestrów dostępnych z poziomu systemu operacyjnego za pomocą funkcji *ioctl()*. Rejestry definiowane są w plikach *.xml* i generowane za pomocą skryptów w języku Perl. Projektant może dodawać lub modyfikować rejestry na potrzeby



Rysunek 1. Przetwarzanie sygnatury HMAC-SHA-1: generacja



Rysunek 2. Przetwarzanie sygnatury HMAC-SHA-1: weryfikacja



Rysunek 3. Modularna struktura HMAC-SHA-1

swojej aplikacji. Moduł HMAC-SHA-1 wprowadza zarówno rejestry użytkownika jak również wewnętrzne rejestry techniczne.

2.2 Architektura wewnętrzna rozwiązania HMAC-SHA-1

HMAC-SHA-1 składa się z modułów realizujących odrębne operacje na ramce (rysunek 3).

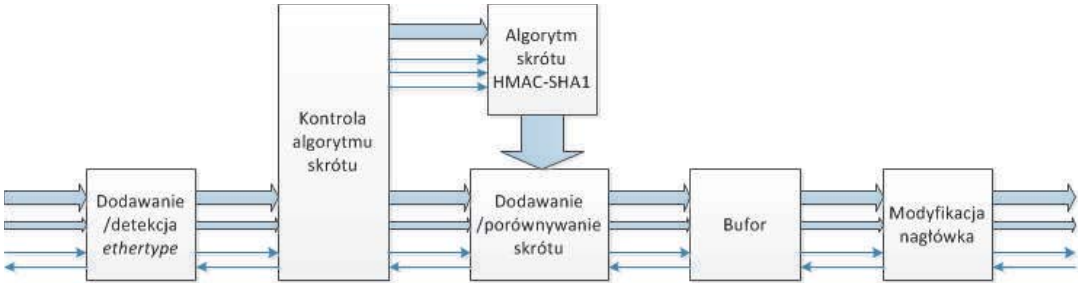
Bufor o pojemności 16kB, zrealizowany z użyciem wbudowanych pamięci BRAM, zapobiega utracie ramek w sytuacji, gdy chwilowy napływ ruchu przekracza możliwości przetwarzania przez modułu HMAC-SHA-1.

Kontrola portu źródłowego porównuje rodzaj ramki (zabezpieczona albo niezabezpieczona) z konfiguracją portu, przez który została odebrana (port do odbioru ruchu zabezpieczonego albo niezabezpieczonego). Przedmiotem uwagi jest wartość pola *ethertype*. Wartość umowna *custom ethertype* wskazuje, że ramka jest zabezpieczona. Ze względu na to, że na nagłówek sterujący ramki w układzie NetFPGA przeznaczone jest 8 bajtów, a jego standardowe pola informacyjne wypełniają 5 bajtów, pozostała przestrzeń jest wykorzystywana wewnątrz HMAC-SHA-1 do przekazywania dodatkowych informacji o stanie przetworzenia ramki w miarę jej przekazywania pomiędzy modułami. Ramki niewłaściwego typu są znakowane w dodatkowym polu tego nagłówka i ostatecznie wysyłane na port diagnostyczny do analizy. Numer portu diagnostycznego, podobnie jak inne parametry układu są definiowane w pliku konfiguracyjnym.

Dodawanie numeru sekwencyjnego przydziela ramce odpowiedni numer bezpiecznego powiązania (asocjacji) oraz numer sekwencyjny w obrębie tej asocjacji na podstawie wartości licznika ramek.

Arbiter dzieli przychodzący ruch pomiędzy dwa moduły HMAC-SHA-1 pracujące równolegle, a następne scala wychodzący z nich ruch w jeden strumień danych, z zachowaniem właściwej kolejności ramek. Ramki przeznaczone przez moduł kontroli portu źródłowego do odrzucenia omijają przetwarzanie w module HMAC i są kierowane bezpośrednio do wyjścia z modułu arbitra.

Moduł HMAC (rysunek 4) realizuje obliczanie wartości skrótu za pomocą algorytmu HMAC-SHA-1 oraz dodawanie skrótu do ramki albo weryfikację wartości skrótu zawartego w ramce. Moduł dodawania albo detekcji pola custom ethertype decyduje o trybie działania modułu – w przypadku wykrycia obecności charakterystycznego dla Systemu IIP nagłówka ustawiany jest tryb weryfikacji ramki. W przeciwnym wypadku aktywowany jest tryb zabezpieczania



Rysunek 4. Struktura wewnętrzna modułu HMAC

i jest dodawane pole *custom ethertype*. Sprawdzany jest również numer asocjacji przypisany do ramki i na jego podstawie ładowane są stosowne klucze kryptograficzne dla obliczenia skrótu przy użyciu algorytmu SHA-1. Moduł kontrolujący algorytm obliczania skrótu wydziela część ramki przeznaczoną do przetworzenia. Algorytm skrótu HMAC-SHA1 oblicza wartość skrótu SHA-1 z danych wejściowych połączonych ze stosownym kluczem. Wynikowy skrót jest ponownie łączony z kluczem i przetwarzany przez algorytm SHA-1. W zależności od trybu (*protect/verify*), obliczony skrót jest doklejany do niezabezpieczonej ramki, bądź porównywany ze skrótem zawartym w zabezpieczonej ramce. Jeżeli moduł HMAC pracuje w trybie weryfikacji ramek zabezpieczonych, to po wykryciu niezgodności skrótów: obliczonego i otrzymanego w ramce, musi zmodyfikować nagłówek ramki tak, aby została ona wysłana na interfejs diagnostyczny do analizy. Ponieważ skrót jest zawarty w końcowej części ramki, a informacja o jego poprawności musi zostać zapisana w nagłówku ramki, to na czas weryfikacji skrótu ramka jest buforowana. Po stwierdzeniu poprawności ramki modyfikowany jest odpowiedni bit w nagłówku ramki.

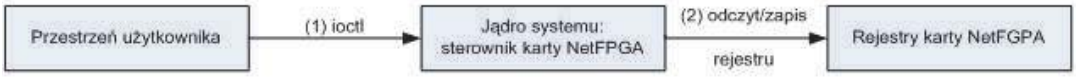
Moduł sprawdzania poprawności numerów sekwencyjnych realizuje algorytm krótko opisany niżej. W przypadku natrafienia na niepoprawny numer ramka zostaje oznaczona jako niepoprawna. Zapisany zostaje oczekiwany numer sekwencyjny, który następnie zostaje przekazany na interfejs diagnostyczny wraz z ramką.

Moduł chirurga realizuje usuwanie numeru sekwencyjnego, pola *custom ethertype* oraz skrótu z ramki zweryfikowanej jako poprawna.

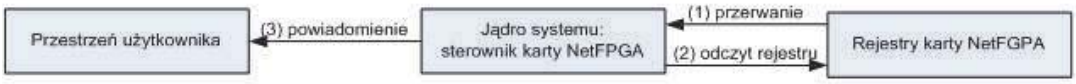
Moduł kontroli błędów analizuje pola błędów w nagłówku NetFPGA. Na ich podstawie zmienia wartości liczników rejestrujących statystyki dotyczące ramek. W przypadku ramki zakwalifikowanej jako błędna (z powodu błędnego numeru portu, niepoprawnej wartości skrótu lub błędu numeru sekwencyjnego) jej pierwsze 64 bajty są przekazywane na port diagnostyczny, wraz z dodatkowymi informacjami o typie błędu.

2.3 Oprogramowanie systemowe

Komunikacja z systemem operacyjnym Komunikacja ta obejmuje komunikację pomiędzy układem NetFPGA a sterownikiem karty w jądrze systemu ope-



Rysunek 5. Schemat komunikacji użytkownika systemu Linux z kartą NetFPGA: mechanizm *ioctl*



Rysunek 6. Schemat komunikacji użytkownika systemu Linux z kartą NetFPGA: mechanizm *Netlink*)

racyjnego Linux a także komunikację pomiędzy sterownikiem a aplikacją w przestrzeni użytkownika. Praca karty NetFPGA realizującej funkcję generatora lub weryfikatora HMAC-SHA-1 może być kontrolowana z poziomu systemu operacyjnego poprzez odczyt lub modyfikacje odpowiednich rejestrów urządzenia, z których część jest przynależna do karty netFPGA, zaś część jest specyficzna dla układu HMAC. Standardowo komunikacja ta odbywa się poprzez magistralę PCI. Rysunek 5 przedstawia zarządzanie rejestrami karty NetFPGA zrealizowaną poprzez systemowe funkcje *ioctl*.

Wadą tego rozwiązania jest brak możliwości komunikacji dwukierunkowej, bowiem inicjatorem komunikacji może być tylko proces z przestrzeni użytkownika w systemie operacyjnym. Od HMAC-SHA-1 wymaga się, aby również karta mogła powiadomić proces użytkownika o stanie w jakim się znalazła – na przykład o zmianach jakie zachodzą w jej rejestrach. W szczególności karta powinna mieć możliwość żądania zmiany aktywnej asocjacji (czyli tzw. bezpiecznego powiązania), a wraz z nim żądania aktywacji nowego klucza kryptograficznego w procesie wymiany klucza gdy bieżące bezpieczne powiązanie wygaśnie ze względu na upływ czasu, bądź z powodu wyczerpania zakresu dopuszczalnych numerów sekwencyjnych. Problem obsługi przerwań od karty rozwiązano rozszerzając sterownik karty NetFPGA w jądrze systemu Linux oraz wykorzystując kontroler przerwań karty.

W pełni dwukierunkową komunikację pomiędzy sterownikiem karty a procesem z przestrzeni użytkownika osiągnięto poprzez wprowadzenie gniazd sieciowych typu *Netlink* [7] - rysunek 6, dzięki któremu proces użytkownika ma możliwość nasłuchiwania zdarzeń pochodzących z jądra systemu, w tym zdarzeń zgłaszanych przez sterownik karty NetFPGA.

Przebieg komunikacji, w której inicjatorem jest karta NetFPGA, jest przedstawiony na rysunku 6. Kolejność zdarzeń jest następująca:

1. z karty NetFPGA do jej sterownika dociera przerwanie,

2. sterownik rozpoznaje typ przerwania poprzez odczyt odpowiedniego rejestru karty; następnie sterownik może odczytać który rejestr uległ zmianie oraz jaka jest aktualna wartość tego rejestru,
3. taki zestaw danych jest przekazywany do procesu z przestrzeni użytkownika, gdzie może zajść dalsza interpretacja danych oraz ich przetwarzanie.

Stosowanie gniazd typu *Netlink* pozwoliło na zdefiniowanie własnego protokołu komunikacji, który dodatkowo można wyposażyć w:

- mechanizmy potwierdzeń poprawnego zapisu danej wartości do danego rejestru karty NetFPGA,
- szczegółowy opis błędu, który wystąpił podczas wykonywania danej czynności przez sterownik karty,
- wiadomości typu multicast do wielu procesów działających w przestrzeni użytkownika.

Podstawowe problemy napotkane podczas implementacji tego schematu komunikacji pomiędzy sprzętem, sterownikiem a aplikacją związane były ze zmianami w mechanizmie *Netlink*, które od wersji jądra 2.6.28 ulegały znacznym modyfikacjom przy równoczesnym braku dokumentacji zmian oraz słabym wsparciu dla kolejnych wersji bibliotek w systemie CentOS. Ostatecznie wybrano nową wersję jądra systemu i zdecydowano się na użycie biblioteki *libnl-genl* będącej rozszerzoną wersją biblioteki *libnl*.

Zarządzanie i sterowanie pracą układu Układ jest w pełni zarządzany. Zapis i odczyt rejestrów może być realizowany lokalnie z wiersza poleceń poprzez interfejs *CLI*. Umożliwia to dostęp do rejestrów typu licznikowego, rejestrów przechowujących klucze kryptograficzne i parametry dla schematu SHA-1 oraz rejestry techniczne. Zarządzanie i sterowanie pracą układu możliwe jest także zdalnie przez protokół SNMP. W tym celu zdefiniowano bazę MIB oraz wyspecyfikowano drzewo identyfikatorów OID. Odpowiednie oprogramowanie wspiera dostęp lokalny i zdalny.

Obsługa numerów sekwencyjnych Ochrona przed atakami *replay* wymaga wprowadzenia numerów sekwencyjnych ramek oraz rozbudowy generatora i weryfikatora HMAC-SHA-1 o obsługę tych numerów. Umożliwia to selektywne akceptowanie tylko tych ramek, które mieszczą się w oknie akceptacji o określonej szerokości. Jako założenie projektowe przyjęto brak współpracy nadajnika z odbiornikiem w zakresie sygnalizacji lub sterowania, która byłaby nieskuteczna na łączach o dużych przepływnościach, lub mogłaby zakłócać swobodny przepływ strumienia danych przez system transmisyjny. Doprowadziło to do znacznej rozbudowy odbiornika (weryfikatora HMAC-SHA-1), który oprócz standardowych sytuacji takich jak przekroczenie zakresu dopuszczalnych wartości numeru sekwencyjnego - "przekręceniełicznika - musi obsługiwać m. in.:

- zagubienie ramki lub bloku ramek,

– odwrócenie naturalnego porządku ramek,
– nadejście ramki ze znacznym opóźnieniem,
– nadejście ramek o wyższych numerach sekwencyjnych niż oczekiwana (wskutek zagubienia jednej lub większej liczby wcześniejszych ramek).

Przypadki te poważnie utrudniają zarządzanie numerami sekwencyjnymi. Wprowadzono okno akceptacji ramek oraz obsługę kilkunastu przypadków szczegółowych.

2.4 Środowisko weryfikacyjne

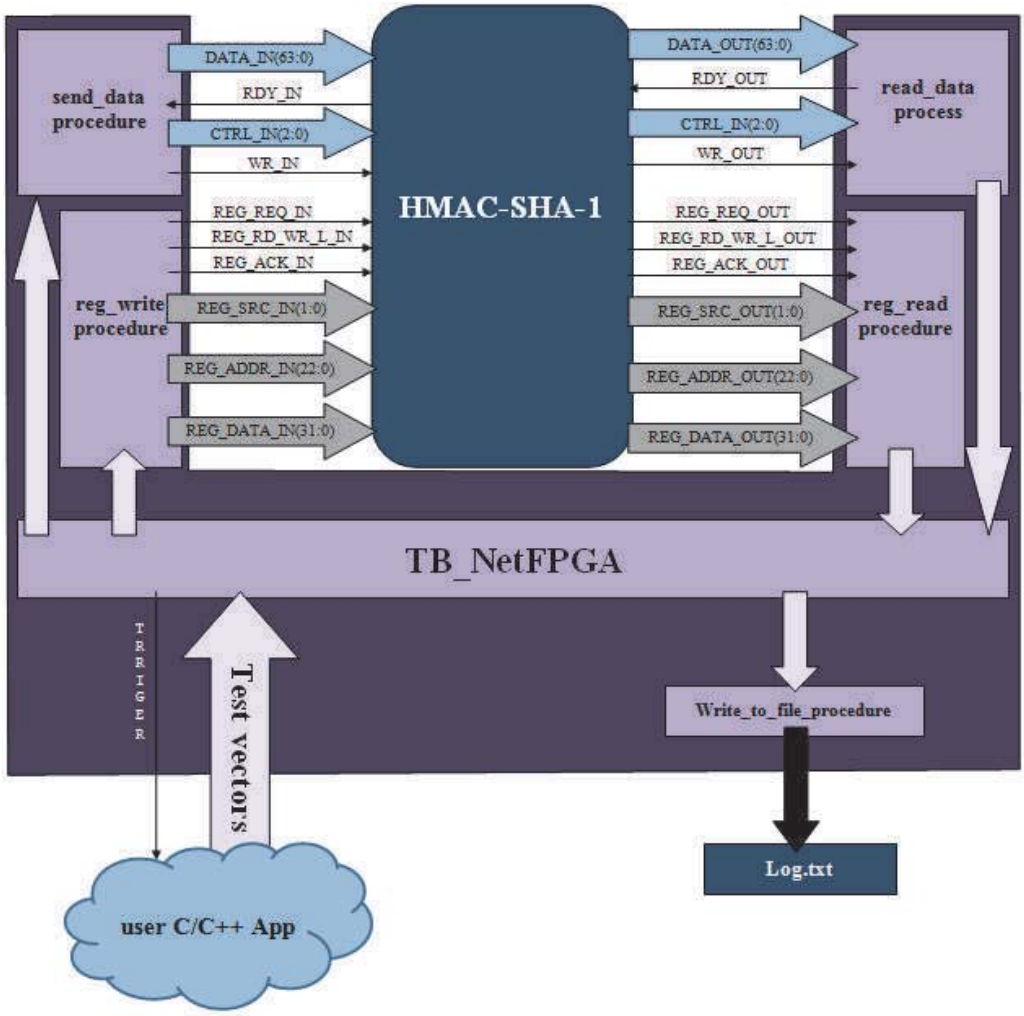
Diagnostyka przy użyciu procedur testowych *TestBench* Dla weryfikacji implementacji sprzętowej modułu HMAC-SHA-1 powstało środowisko testowe – rysunek 7. Środowisko to wysyła odpowiednie dane i porównuje odpowiedź modułu HMAC-SHA-1 z danymi referencyjnymi.

Do generowania wektorów testowych stworzono aplikację w języku C/C++ na bazie ogólnodostępnej biblioteki *CryptoPP* [8]. Aplikacja generuje dwa rodzaje ramek Ethernet: przeznaczonych do zabezpieczenia (wysyłane do modułu) oraz zabezpieczonych (referencyjne). Dane następnie są przesyłane za pomocą interfejsu VHPI (*VHDL Procedural Interface*) [9] do środowiska, które wysyła, odbiera i porównuje ramki, a wynik zapisuje do pliku testowego. Powstało kilkadziesiąt scenariuszy testowych które obejmują między innymi:

– wysyłanie ramek o losowej długości i losowej zawartości,
– wysyłanie ramek o określonej długości w pewnym zakresie z krokiem wynoszącym 1 oktet,
– wysyłanie błędnych ramek, na przykład niosących niepoprawny port źródłowy, niepoprawny numer sekwencyjny, złą sygnaturę skrótu (*hash*),
– wysyłanie kolejnej ramki po przetworzeniu przez moduł poprzedniej ramki,
– wysyłanie paczek ramek z konfigurowalnymi odstępami międzyramkowymi.

Testy były przeprowadzane w trybach: *protect* (zabezpieczanie ramek), *verify* (weryfikacja zabezpieczonych ramek) oraz *mixed-mode* (naprzemienne wysyłanie ramek przeznaczonych do zabezpieczenia i ramek zabezpieczonych). Łącznie przetworzono kilkaset tysięcy ramek. W trakcie 20-minutowego testu przetwarzano około 3000 ramek, a w przypadku wykrycia błędów test był powtarzany. Dla przyspieszenia testów do weryfikacji używano dedykowanej karty HES opisanej w dalszej części tego rozdziału.

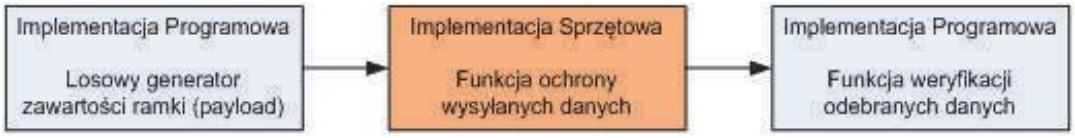
Weryfikacja programowo-sprzętowa Przed docelową implementacją na platformie NetFPGA wykonano programową implementację HMAC-SHA-1. Na taki krok zdecydowano się z trzech powodów. Po pierwsze, implementacja programowa przebiega znacznie szybciej niż implementacja wersji sprzętowej. W przypadku błędów implementacyjnych związanych z logiką algorytmu przetwarzania ramki pozwoliło to zaoszczędzić czas potrzebny na implementację poprzez uniknięcie powtórzenia tych błędów w sprzęcie. Po drugie, rozwiązanie to umożliwiło



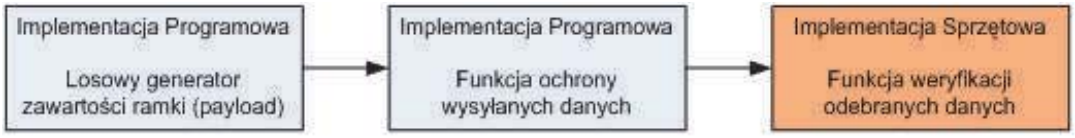
Rysunek 7. Schemat blokowy środowiska testowego HMAC (*TestBench*)

weryfikację implementacji sprzętowej przy użyciu implementacji programowej. Wreszcie potwierdzono, że opracowane wcześniej algorytmy są jednoznacznie rozumiane przez dwa zespoły implementatorów, pracujące niezależnie. Wersję programową również przetestowano pod względem poprawności obliczania sygnatury HMAC-SHA-1 zgodnie z dokumentem RFC 2202 [10]. Testy przewidują dwie konfiguracje (por. rysunki 8 i 9), co pozwala zweryfikować poprawność obliczania sygnatury HMAC, formatowania pól ramki oraz zarządzania numerami sekwencyjnymi.

Konfiguracja przedstawiona na rysunku 8 pozwala określić zdolność poprawnej generacji ramki z sygnaturą HMAC i zarządzanie numerami sekwencyjnymi. Druga konfiguracja, rysunek 9, sprawdza sprzętową funkcję weryfikacji ramki. Weryfikacja polega na obliczeniu sygnatury oraz porównaniu jej z otrzymaną w ramce. Następnie testowana jest funkcja kontroli numerów sekwencyjnych oraz



Rysunek 8. Schemat blokowy układu testowania weryfikacji zawartości ramki: konfiguracja 1



Rysunek 9. Schemat blokowy układu testowania weryfikacji zawartości ramki: konfiguracja 2

przywracanie pierwotnej formy ramki tj. bez pól sygnatury HMAC oraz numerów sekwencyjnych. Środowisko jest monitorowane za pomocą programowego analizatora ruchu sieciowego.

Uruchomienie i diagnostyka Moduł HMAC-SHA-1 został zsyntezowany dla układu *Virtex2P 50FF1152* dysponującego 22464 blokami logicznymi (*slice*) i wykorzystuje 95% jego zasobów. Analiza implementacji poparta eksperymentami pozwoliła na sformułowanie wniosku, że wysoka gęstość upakowania logiki może prowadzić do niestabilnej pracy układu przy wyższych przepływnościach bitowych. Wynika to z ograniczeń platformy sprzętowej Virtex II i jest związane z tym, że program z trudem mieści się w układzie Virtex II. Ze względu na rozbudowaną logikę przetwarzania (np. moduł Arbitra, umożliwiający zrównoleglenie przetwarzania), narzędzia nie są w stanie przeprowadzić implementacji projektu w strukturze rozmieszczając bloki logiczne tak, by można było je taktować domyślnym zegarem karty 125MHz. Ograniczenie to wynika ze specyfiki układu Virtex II i przemawia za przeniesieniem pracy do środowiska Virtex V na kartę *netFPGA10G*, gdzie można będzie w całości wykorzystać możliwości wynikające z architektury rozwiązania HMAC dopuszczającej przetwarzanie równoległe. Automatyczna synteza okazała się nieoptymalna. Aby utrzymać standardowe taktowanie architektury NetFPGA (125MHz) została zastosowana ręczna optymalizacja struktury modułu.

Do testowania wydajności układu HMAC-SHA-1 wykorzystano sprzętowo akcelerowany układ testowania implementacji (karta HES) a w dalszym etapie generator Spirent z oprogramowaniem *Test Center*. Modul HMAC-SHA-1 testowany na karcie HES wykazuje poprawne działanie dla wysokiej częstotliwości taktowania. W połączeniu z generatorem Spirent układ HMAC-SHA-1pracuje stabilnie przy przepustowościach zagregowanych do 750 Mbit/s.

Weryfikacja przy użyciu sprzętowego generatora ruchu Opisane powyżej metody testowania zostały powtórzone w scenariuszach testowych z użyciem generatora ruchu Spirent. Testy badały wydajność implementacji w różnych warunkach pracy oraz testy odporności na błędy i zakleszczenia. Generator był źródłem strumieni ramek o stałych i zróżnicowanych rozmiarach (naprzemiennie długich i krótkich) oraz niebędących wielokrotnością 8 bajtów, o różnej i zmiennej przepływności bitowej, o różnych odstępach międzyramkowych, o różnej zawartości ramek, a także strumieni generowanych w oparciu o referencyjne scenariusze testowe. Doświadczenia wynikające z tych prac stanowią materiał na osobną publikację.

2.5 Wnioski z implementacji

Obliczanie i wprowadzanie do ramki sygnatury skrótu HMAC-SHA-1 oraz numeru sekwencyjnego jest względnie łatwe w implementacji programowej. W przypadku implementacji sprzętowej nakład pracy wymagany do uruchomienia układu oraz na opracowanie procedur testowych, a także nakład pracy wymagany do całościowego przetestowania rozwiązania jest bardzo duży, albowiem lokalizacja i usuwanie błędów są tutaj zdecydowanie trudniejsze. Zastosowanie dobrych praktyk, takich jak układy testbench, interfejs VHPI oraz użycie metody akceleracji sprzętowej HES pomaga w szybszym lokalizowaniu błędów, jednak nie uwalnia od samego poszukiwania błędów. W toku prac projektowych i realizacyjnych przezwyciężono problem niskiej przepustowości układu Virtex i małej dostępności zasobów tego układu. Prace implementacyjne zostały zakończone powodzeniem.

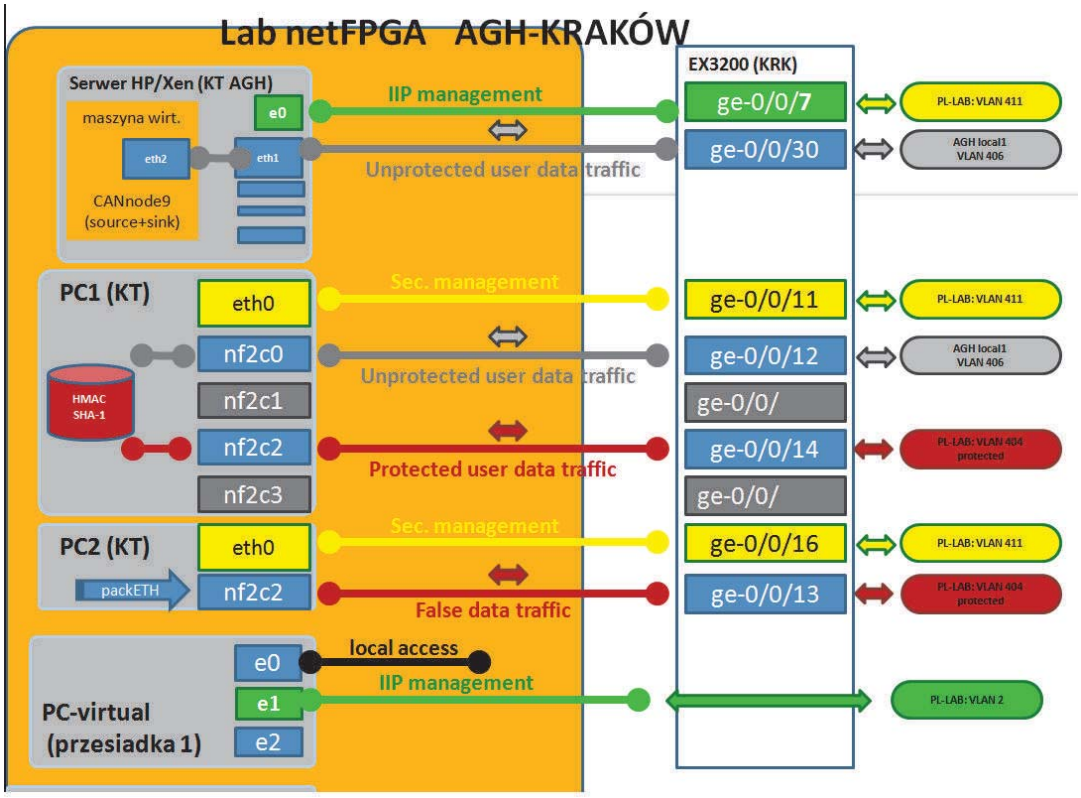
2.6 Integracja w środowisku PL-LAB

Układ został włączony do w sieci PL-LAB w celu zabezpieczenia ruchu generowanego przez Równoległy Internet CAN na łączu pomiędzy Krakowem i Wrocławiem. Schemat konfiguracji programowo-sprzętowej węzła PL-LAB w Krakowie dla potrzeb eksperymentów prowadzonych przez zespół *Security* jest pokazany na rysunku 10.

W konfiguracji wykorzystano sieci VLAN obsługiwane przez system PL-LAB i terminowane na odpowiednich portach przełączników dostępowych *EX3200* ulokowanych w Krakowie oraz we Wrocławiu. Strumień danych użytkowych od pochodził od węzła sieci CAN będącego maszyną wirtualną. Dla przenoszenia ruchu zabezpieczonego (*protected user data traffic*) używano dedykowanego VLAN tak, jak to przedstawia rysunek 10.

3 Wykrywanie anomalii w szeregach czasowych

Zachowanie węzła Systemu IIP może być analizowane w kontekście charakterystyki zmian wartości wybranych atrybutów opisujących stan węzła. Istotnym elementem umożliwiającym wykrywanie stanów nietypowych (anomalii) węzła



Rysunek 10. Schemat blokowy układu testowania weryfikacji zawartości ramki: implementacja programowa

Systemu IIP jest zatem czas obserwacji. Korzystając z tej właściwości, w celu ochrony Systemu IIP została zaadoptowana i zaimplementowana jedna z dostępnych metod wykrywania anomalii w szeregach czasowych. Atrybutami opisującymi działanie węzła Systemu IIP, które mają szczególne znaczenie w kontekście zadania wykrywania zagrożeń są: poziom wykorzystania czasu procesora, poziom zajętości pamięci operacyjnej węzła oraz liczba wysłanych i odebranych ramek dla każdego z Równoległych Internetów. Na przykład nietypowo wysoki poziom obciążenia procesora, czy też nadmiernie duża liczba wysłanych/odebranych ramek w jednostce czasu są charakterystycznymi objawami towarzyszącymi atakom na dostępność usługi (ang. *Denial of Service – DoS*). Takie objawy towarzyszyć mogą również atakom specyficznym dla Systemu IIP, np. polegającym na wstrzykiwaniu ruchu, fałszowaniu ruchu czy też zakłócaniu ruchu poprzez wprowadzanie sztucznych opóźnień. Metodą, która została zaadaptowana na potrzeby zaimplementowanego modułu wykrywania anomalii LAD, jest metoda zaproponowana przez M.Burgessa w pracy [11]. Jedną z istotnych zmian wprowadzonych do oryginalnej koncepcji M.Burgessa było uwzględnienie specyfiki Systemu IIP polegającej na współistnieniu w jednym węźle kilku Równoległych Internetów, które to mogą posiadać odrębną charakterystykę zmienności. Inną istotną modyfikacją istniejącej metody było zintegrowanie procesu wykrywania anomalii

realizowanego przez LAD z działaniami modułu szacującego reputację węzłów Systemu IIP.

Moduł LAD realizuje analizę szeregów czasowych w trzech następujących krokach. Pierwszy krok dotyczy selekcji cechy na potrzeby wykrywania anomalii. Oznacza to, że w tym kroku dokonywany jest wybór najbardziej reprezentatywnego dla potrzeb analizy bezpieczeństwa atrybutu/atrybutów opisującego stan węzła. Krok drugi to szacowanie parametrów. W tym kroku historyczne wartości wybranej cechy porównywane są z wartościami bieżącymi. Ma to na celu oszacowanie poziomu znaczenia zmian wartości obserwowanej cechy dla procesu wykrywania nietypowych stanów węzła. Następnie te dwa kroki są powtarzane wielokrotnie, tak aby ustalić ostateczny model w sposób umożliwiający prawidłowe identyfikowanie stanów nietypowych węzła. Po ostatecznym ustaleniu modelu węzła moduł jest w stanie identyfikować anomalie, które zdefiniowane są jako stany węzła charakteryzujące się znaczącym poziomem rozbieżności pomiędzy bieżącą obserwacją a stanem typowym zdefiniowanym poprzez statystyki wyliczone w fazie ustalania modelu zachowania węzła. Jednym z podstawowych założeń dotyczących konstruowanego modelu jest cykliczność obserwowanych zmian wartości wybranej cechy.

Na potrzeby przedstawienia idei zaimplementowanej metody analizy szeregów czasowych przyjęto, że analizie podlega szereg opisujący liczbę odebranych przez węzeł bajtów danych. Szereg został oznaczony w następujący sposób:

$$X = (x_1, x_2, \ldots, x_l, \ldots), \tag{1}$$

gdzie x_l jest liczbą odebranych bajtów w oknie czasowym o numerze l. Przy czym szerokość okna czasowego jest odrębnym parametrem podlegającym optymalizacji ze względu na dokładność wykrywania anomalii. Wartość ta, podobnie jak inne wartości zmienne algorytmu, może być wyznaczona z wykorzystaniem analizy za pomocą krzywych ROC (ang. *Receiver Operating Characteristic*). Metoda ta jest powszechnie wykorzystywana w zadaniach oceny jakości klasyfikacji i w tym przypadku może być również użyta w celu oceny jakości procesu wykrywania anomalii dla przyjętych różnych wartości progowych.

Następnym istotnym elementem metody jest wyróżnienie dwóch podszeregów z analizowanego szeregu X. Podszeregi te oznaczone zostały odpowiednio X_P i X_T

$$X_{P,l} = \frac{1}{P}\sum_{k=0}^{P-1} x_{l-k}, X_{T,l} = \frac{1}{T}\sum_{k=0}^{T-1} x_{l-kP}, \tag{2}$$

gdzie P oraz T są wartościami wykorzystanymi w celu uśrednienia wartości (wartość P jest nazywana okresem charakterystycznym szeregu X). Szeregi X_P i X_T składają się zatem z wartości średnich obserwowanej cechy, wyliczonych na wartościach szeregu X - odpowiednio szereg X_P z wartości średnich elementów pobranych z kolejnych przedziałów równych co do długości okresowi charakterystycznemu P, a szereg X_T z wartości średnich elementów X znajdujących się w tej samej fazie okresu charakterystycznego szeregu X. Dla obu tak zdefiniowanych podszeregów wyliczane są wartości standardowego odchylenia $\sigma_{P,l}$ oraz $\sigma_{T,l}$

na odpowiednich przedziałach uśredniania dla wartości wyliczonych na podstawie wykładniczej średniej kroczącej $\bar{X}_{P,l}$ i $\bar{X}_{T,l}$. Ostatecznie lokalne odchylenie obserwowanej wartości cechy wyrażone jest poprzez następującą formułę

$$\delta_{P,l} = |x_l - \bar{X}_{P,l}|, \delta_{T,l} = |x_l - \bar{X}_{T,l}|, \tag{3}$$

Jeżeli obserwacja x_l dotyczyła stanu anomalnego, to następnie obliczany jest potencjalny niekorzystny wpływ tego stanu na poziom bezpieczeństwa monitorowanego węzła. Wpływ ten oznaczany jest przez zmienną *severity*, której aktualna wartość jest przekazywana przez LAD do modułu reputacyjnego oraz która jest szacowana za pomocą wyrażenia:

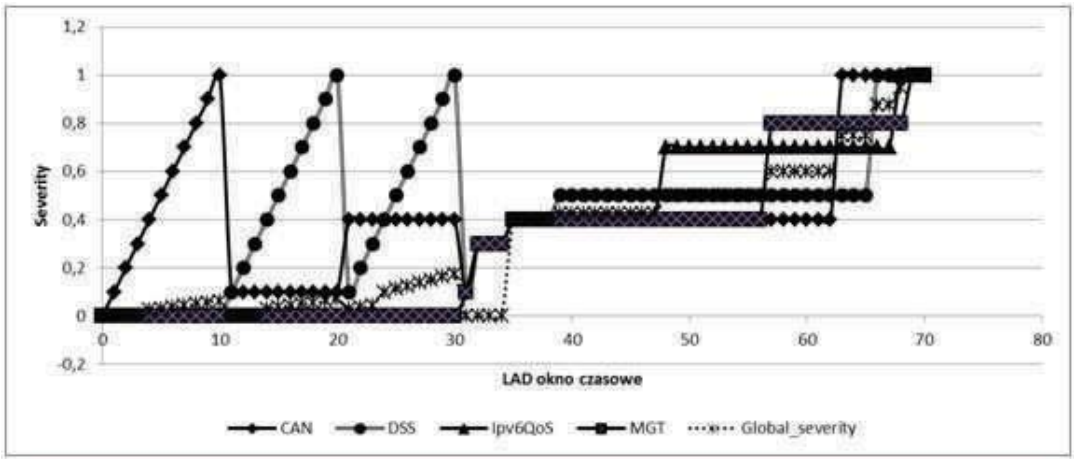
$$c_l = \frac{\sqrt{\left(\delta_{P,l}/\sigma_{P,l}\right)^2 + \left(\delta_{T,l}/\sigma_{T,l}\right)^2}}{3\sqrt{2}}, \tag{4}$$

($c_l = 1$ jeżeli prawa strona równości (4) przekracza 1). Warto zauważyć, iż w przypadku gdy aktualnie zmierzona wartość cechy jest zbliżona do wartości średniej, to jest gdy bieżąca obserwacja koresponduje z doświadczeniem historycznym, to wartość zmiennej *severity* jest bliska 0. Natomiast gdy różnica pomiędzy aktualną wartością cechy oraz jej wartością średnią przewyższa trzykrotnie wartość wyliczonego odchylenia standardowego, to wartość zmiennej *severity* jest maksymalna, równa 1. Drugim, komplementarnym atrybutem określającym poziom zagrożenia wykrytej anomalii jest *intensity*. Wartość tego parametru jest wyliczana jako średnia wartość ze zmiennej *severity* liczona dla charakterystycznego okresu szeregu X.

Specyficzną cechą architektury Systemu IIP jest to, że węzły IIP przekazują cztery, w ogólności niezależne, strumienie ruchu sieciowego związane z czterema Równoległymi Internetami (IPv6 QoS, DSS, CAN oraz Management). Dlatego kolejnym istotnym elementem opracowanej metody wykrywania anomalii była możliwość wykrywania anomalii na ogólnym poziomie analizy, tzn. dotyczących całego węzła Systemu IIP, a nie tylko poszczególnych Równoległych Internetów. Oznacza to konieczność zdefiniowania miar *severity* oraz *intensity* z uwzględnieniem wiedzy dotyczącej wszystkich czterech Równoległych Internetów. Globalna wartość *severity* łączy wartości *severity* wyliczone dla wszystkich Równoległych Internetów i jest wyrażona w następujący sposób:

$$\begin{aligned} c_{l-global} = \frac{\gamma}{4} \left(\beta_{DSS} * c_{l-DSS} + \beta_{CAN} * c_{l-CAN}\right. \\ \left. + \beta_{IPv6QoS} * c_{l-IPv6QoS} + \beta_{MGT} * c_{l-MGT}\right), \end{aligned} \tag{5}$$

gdzie wartości *severity* dla poszczególnych Równoległych Internetów są obliczone w sposób przedstawiony wcześniej z wykorzystaniem wiedzy dotyczącej zmian w ruchu w obrębie danego Internetu. Wartości współczynników występujących w równaniu (5) sumują się do jedności i dają możliwość zrównoważenia wpływu poszczególnych Równoległych Internetów na poziom globalnej wartości *severity*. Wartość parametru jest równa liczbie Równoległych Internetów, dla których wartość *severity* przekroczyła predefiniowaną wartość progową. W przykładzie przedstawionym na rysunku 11 przyjęto ten próg na poziomie $0,3$ oraz

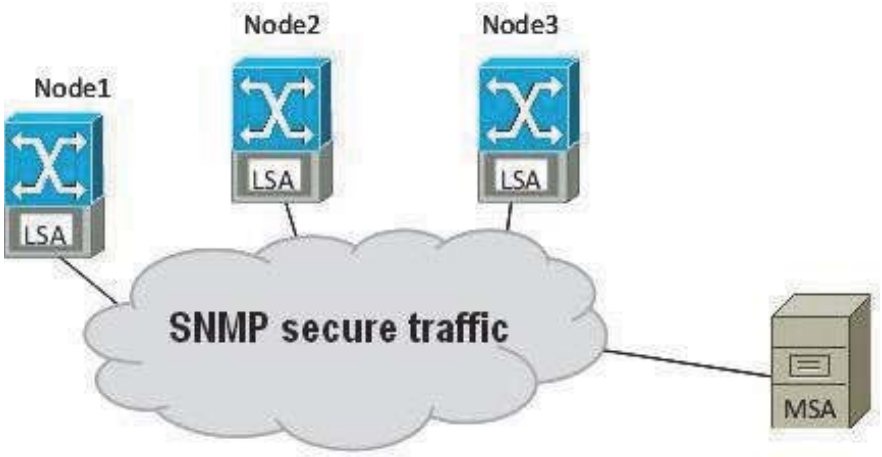


Rysunek 11. Poziom globalnej zmiennej *severity* oraz wartości *severity* dla poszczególnych Równoległych Internetów

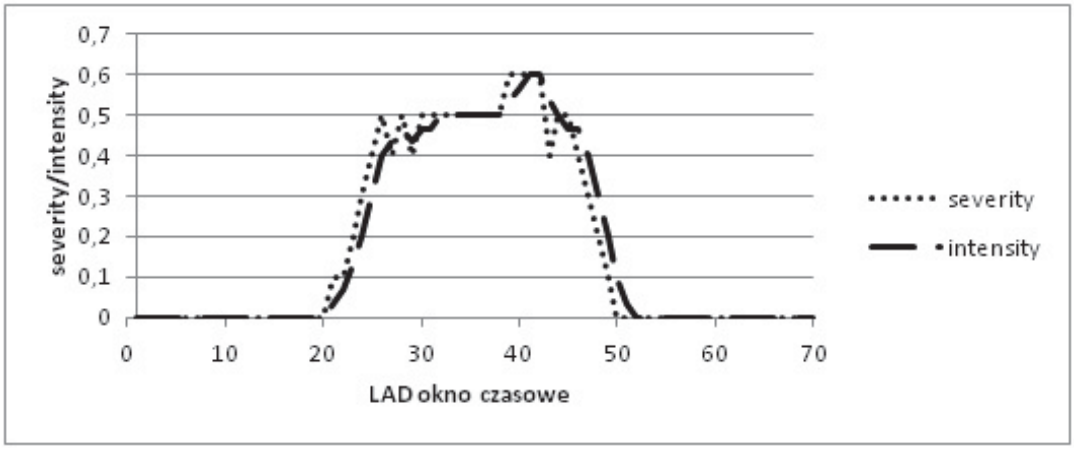
wszystkim parametrom przypisano wartość równą $0,25$. Z postaci wzoru (5) wynika, że nie tylko zmiany aktualnego poziomu *severity* dla poszczególnych Równoległych Internetów mają wpływ na ostateczną wartość globalną. Zależy ona również od liczby Równoległych Internetów, których praca została zaburzona w znaczącym stopniu - im więcej, tym mocniej ich zaburzenia oddziałują na globalną ocenę bezpieczeństwa węzła.

Poniżej został przedstawiony scenariusz testowy, który demonstruje możliwości zaimplementowanego w module LAD algorytmu analizy szeregów czasowych na potrzeby detekcji specyficznych ataków poziomu 2 architektury Systemu IIP (IIPS Level 2). Tymi specyficznymi atakami są: wstrzykiwanie ruchu (ang. *traffic injection*) oraz zaburzanie ruchu (ang. *ruffling*). Moduł LAD analizuje szeregi czasowe utworzone z wartości opisujących liczbę bajtów odebranych w ramach danego Równoległego Internetu w badanym węźle Systemu IIP. Wartości te są dostarczane do LAD co 10 sekund przez podsystem monitorowania Systemu IIP za pomocą metody Get dostępnej w ramach protokołu SNMP dla odpowiednio ustalonej wartości OID. Na potrzeby badań ruch był generowany w sposób sztuczny z wykorzystaniem aplikacji D-ITG [12].

Pierwszy eksperyment dotyczył symulowanych ataków, które mają miejsce w obrębie pojedynczego Równoległego Internetu i dotyczyły sytuacji wstrzykiwania ruchu, gdzie atak był realizowany z poziomu węzła Node3 przeciw węzłowi Node2 (rys. 12). W początkowym okresie Node3 generował ruch typowy o charakterystyce odstępów pomiędzy kolejnymi IIPS-PDU zgodnie z rozkładem Pareto, gdzie wartość średnia dla rozkładu została ustalona na poziomie 15 ms a wartość odchylenia standardowego na poziomie 75 ms. Natomiast rozmiar przesyłanej ramki został zdefiniowany jako równy 5000 B. Moduł LAD w węźle Node2 analizuje szereg czasowy utworzony z wartości oznaczających liczbę odebranych bajtów w kolejnych oknach czasowych. Temu okresowi eksperymentu odpowiada wartość 0 dla zmiennej *severity* przedstawionej na wykresie na ry-



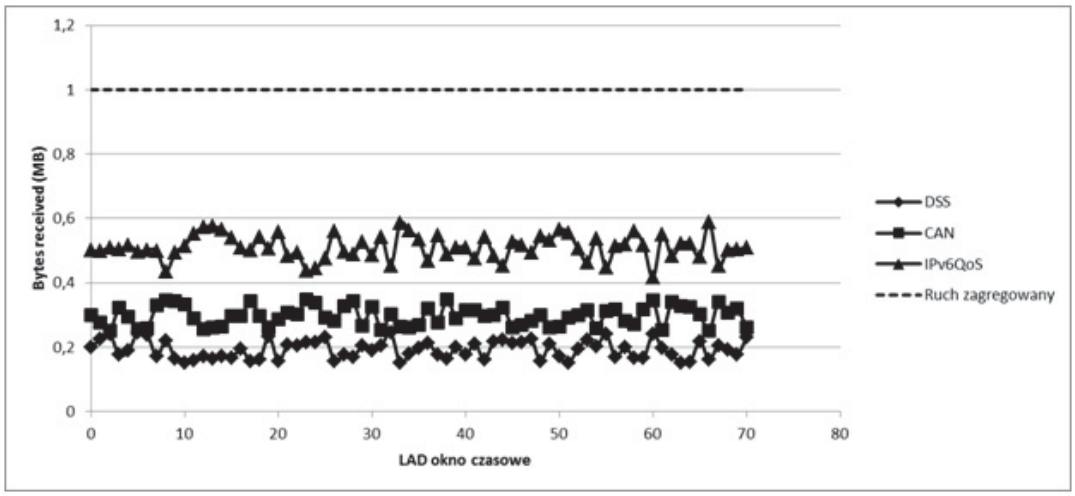
Rysunek 12. Środowisko dla testów integracji systemu bezpieczeństwa i systemu zarządzania uruchomione w Instytucie Łączności w Warszawie



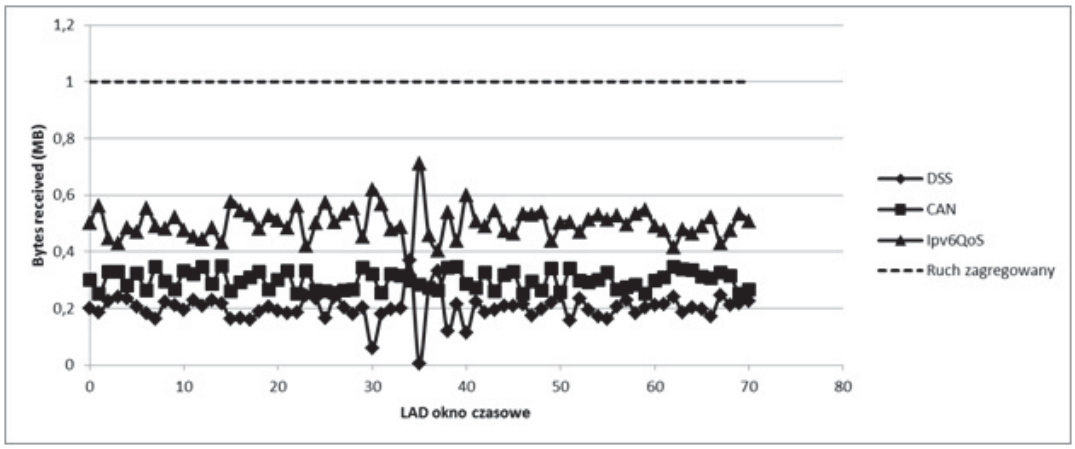
Rysunek 13. Wykrywanie ataków typu *traffic injection*

sunku 13. Wstrzyknięcie ruchu przez węzeł Node3 zostało zasymulowane poprzez wysłanie strumienia dodatkowych ramek z interwałami losowanymi z rozkładu normalnego z wartością średnią równą 20 ms oraz odchyleniem standardowym 5 ms. Dodatkowe ramki miały rozmiar 500 B. Wstrzykiwanie dodatkowego ruchu rozpoczęto po upływie 20 okien. LAD w węźle Node2 aktualizuje swoje obserwacje i w miarę narastania liczby zaburzonych elementów obserwowanego szeregu czasowego zwiększa się również wartość zmiennej *severity*. Na rysunku 13 można również zauważyć, iż po pewnym czasie LAD uczy się nowej charakterystyki obserwowanego ruchu i wartość *severity*/*intensity* ponownie spada do 0.

Następny scenariusz testowy pokazuje różnice pomiędzy wykrywaniem anomalii w zagregowanym ruchu pochodzącym z wszystkich Równoległych Inter-



Rysunek 14. Ruch bez anomalii



Rysunek 15. Ruch z anomaliami w Równoległych Internetach DSS oraz IPv6QoS

netów, a wykrywaniem anomalii w ruchu każdego z Równoległych Internetów z osobna. Na rysunkach 14 oraz 15 średnia liczba bajtów zagregowanego ruchu kierowanego w danym oknie czasowym do węzła IIPS jest stała. Oznacza to, że w obu przypadkach gdy poddajemy analizie ruch zagregowany, nie zostaną wykryte żadne anomalie, chociaż charakterystyka ruchu na poziomie poszczególnych Równoległych Internetów w sytuacjach zilustrowanych na rysunkach 14 i 15 jest różna. Stan ten ulegnie zmianie, gdy będzie rozpatrywany oddzielnie ruch dla każdego z Równoległych Internetów. W tym przypadku zostanie wykryte anomalne natężenie ruchu dla Równoległego Internetu DSS oraz IPv6QoS pomiędzy 30. a 40. oknem czasowym (rysysunek 15). Dzięki temu podana wcześniej miara globalnej wartości *severity* dla węzła będzie w stanie również odzwierciedlić różnicę obu przedstawionych przypadków.

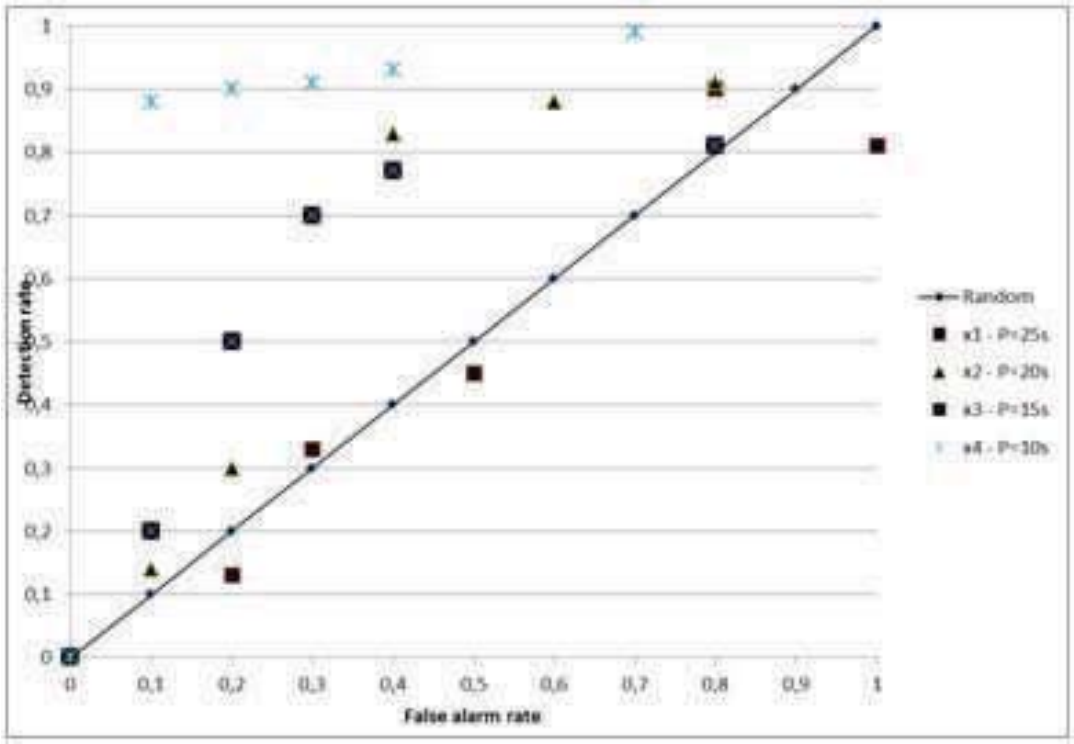
Zanim opisana metoda będzie mogła być wykorzystana w rzeczywistym Systemie IIP, konieczna jest odpowiednia parametryzacja algorytmu. Jedną z częściej wykorzystywanych metod pozwalających na ustalenie optymalnych wartości parametrów algorytmu klasyfikującego zdarzenia jako normalne lub anomalne jest metoda wykorzystująca krzywe ROC (ang. *Receiver Operating Characteristic*) [13]. Krzywe te umożliwiają obrazowe porównanie współczynnika fałszywych alarmów z współczynnikiem wykrywalności zdarzeń nietypowych. Współczynnik fałszywych alarmów jest ilorazem liczby błędnych alarmów i liczby wszystkich anomalii, które miały miejsce w badanym okresie, a współczynnik wykrywalności jest ilorazem liczby prawidłowo wykrytych anomalii i wszystkich anomalii w badanym okresie. Na przykład, aby dobrać odpowiednią wartość okresu charakterystycznego P dla równania (2) zostało przeprowadzonych pięć eksperymentów z różnymi wartościami P. W eksperymentach wykorzystano przedstawiony wcześniej scenariusz ataku poprzez wstrzyknięcie ruchu. Otrzymane krzywe ROC zostały przedstawione na rysunku 16. Uzyskane wyniki wskazują na to, iż odpowiednią dla tego typu zdarzeń wartością P jest okres o długości 10 sekund, gdyż dla takiej wartości okresu charakterystycznego przy niewielkim współczynniku fałszywych alarmów otrzymano najwyższy współczynnik wykrywalności sięgający 90% . Analogiczne eksperymenty powinny być przeprowadzone również dla określenia właściwych wartości pozostałych parametrów zmiennych wykorzystywanych w zaimplementowanej metodzie analizy szeregów czasowych.

Ponieważ System IIP na obecnym etapie jest w trakcie ustawicznego rozwoju oraz badań laboratoryjnych, w przedstawionym przykładzie posłużono się ruchem generowanym w sposób sztuczny, za pomocą dodatkowych aplikacji. Docelowo parametryzacja algorytmu powinna wykorzystywać dane pochodzące z rzeczywistego Systemu IIP.

4 Wykrywanie anomalii metodami opartymi na eksploracji danych

Wykrywanie anomalii za pomocą metod eksploracji danych bazuje na analizie rejestru SRE. Wpisy SRE mogą przykładowo zawierać: nieudane logowanie, odrzucenie PDU przez moduł HMAC, próba odczytania poprzez SNMP klucza z bazy MIB, do którego użytkownik nie ma stosownych uprawnień czy też odrzucenie połączenia w sieci zarządzania niezgodnego z polityką bezpieczeństwa. Z każdym z wymienionych zdarzeń, oprócz jego typu i czasu wystąpienia, związane są pewne atrybuty, np. nazwa konta, adresy IP odrzuconego PDU: źródłowy i docelowy, czy OID klucza w bazie MIB. SRE generowane przez różne programy działające pod kontrolą systemu Linux w węźle IIP są logowane za pomocą standardowego podsystemu syslog, skąd moduł LAD pobiera dane do analizy. Takie rozwiązanie umożliwi w przyszłości łatwe uwzględnianie innych zdefiniowanych SRE. Aktualnie moduł LAD przystosowany jest do analizowanie danych z następujących źródeł:

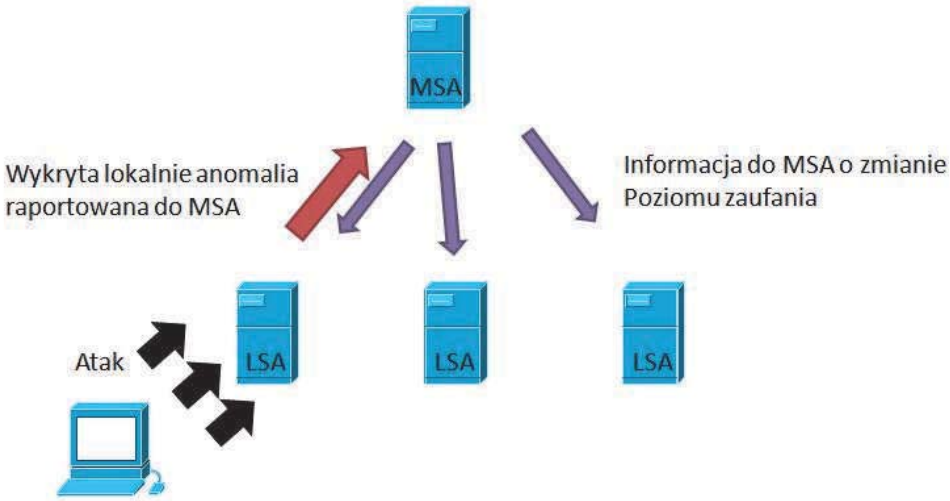
– zapora ogniowa systemu Linux (*ip6tables*),



Rysunek 16. Krzywe ROC dla różnych wartości P

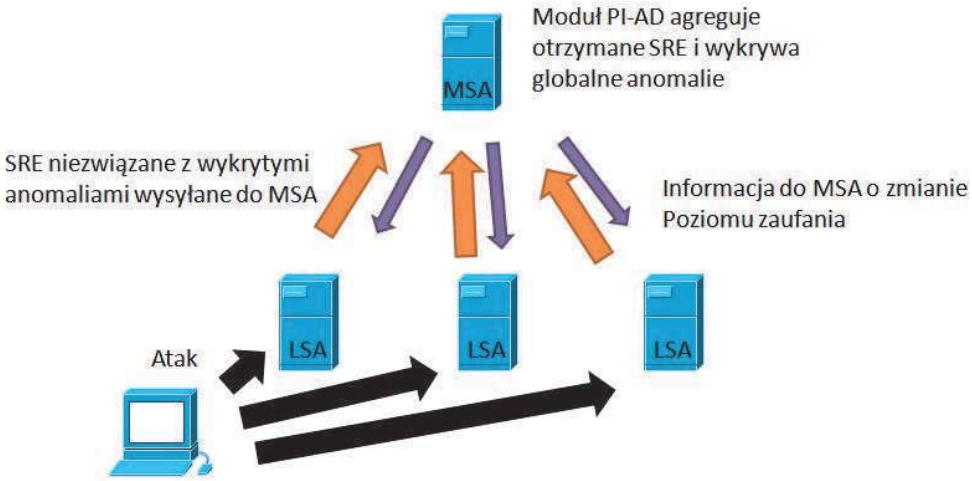
– proxy SNMP opisane szczegółowo w podrozdziale 6, pozwalające uzyskać logi związane z atakami na serwer SNMP,
– interfejs diagnostyczny modułu HMAC,
– logi z serwera SSH.

Proponowana metoda opiera się na wykrywaniu tzw. zbiorów częstych (*frequent sets*), tj. powtarzających się często wzorców zachowań [14]. Jeśli dane do analizy traktujemy jako zbiory odpowiednich atrybutów, to zbiorem częstym nazywamy podzbiór występujący co najmniej *minSup*-krotnie w analizowanych danych. Parametr *minSup*, nazywany minimalnym wsparciem, jest parametrem wejściowym algorytmu wyszukiwania zbiorów częstych. Zbiory częste można wykrywać za pomocą różnych algorytmów. W ramach projektu został zaimplementowany algorytm *a priori*, którego dokładny opis można znaleźć w [5]. Analizując tą metodą SRE dotyczące np. odrzuconych prób dostępu do kluczy bazy MIB można wykryć wzorzec związany z próbą poznania przez danego użytkownika wartości pewnych gałęzi. Jeśli w wykrytym zbiorze częstym będzie znajdowała się informacja o adresie IP, oznaczać to będzie iż ataki były przeprowadzane z jednej maszyny. Brak w wykrytym zbiorze częstym identyfikatora OID świadczy o próbie dotyczy poznania różnych gałęzi. W sytuacji kiedy wykrytym zbiorze częstym będzie atrybut związany z OID, a brak będzie atrybutu związanego z adresem IP, atak był przeprowadzany z wielu maszyn i dotyczył określonego elementu bazy MIB. W wyniku analizy uzyskanych zbiorów częstych, do modułu



Rysunek 17. Działanie modułu lokalnego wykrywania anomalii (LAD)

obliczania reputacji (*reputation calculator*) zostanie przekazana ocena ryzyka (*severity*) związanego z daną anomalią oraz prawdopodobieństwo trafności oceny dokonanej przez moduł LAD. Informacje te wykorzystywane są przez system zarządzania reputacją (szczegółowo omówiony w podrozdziale 5). W celu wykrywania szerokiego spektrum ataków, analiza anomalii z wykorzystaniem metod eksploracji danych wykorzystywana jest na dwóch poziomach - lokalnym poprzez moduł LAD (ang. *local anomaly detection*) oraz globalnie poprzez moduł PI-AD (ang. *Parallel Internet-wide Anomaly Detection*). Moduł LAD zainstalowany na każdym węźle Systemu IIP analizuje wszystkie zarejestrowane SRE i wykrywa w nich zbiory częste. Wykrycie zbioru częstego jest dowodem wystąpienia pewnego powtarzalnego wzorca, który z dużym prawdopodobieństwem może być atakiem, lub inną anomalią, która wymaga interwencji administratora. Jeśli wykryty zbiór częsty posiada element opisujący adres nadawcy, generowany jest komunikat o anomalii i wysyłany za pomocą modułu LSA do modułu centralnego MSA. Na podstawie poszczególnych elementów zawartych w wykrytym węźle, jak i otrzymanego wsparcia wykrytego zbioru częstego obliczane są wymagane przez podsystem obliczania reputacji parametry wykrytej anomalii: jej potencjalne zagrożenie (*severity*) oraz prawdopodobieństwo trafności (*intensity*). Sytuacja tego typu przedstawiona jest na rysunku 17. Działania podejmowane w trakcie ataku powodują wygenerowanie wystarczającej liczby zdarzeń SRE aby wykryć zbiór częsty. Analiza wykrytego zbioru częstego prowadzi do wygenerowania anomalii, która później raportowana jest do modułu centralnego MSA. Na podstawie tych informacji, moduł MSA wylicza poziom zaufania danego węzła, a informacja zwrotnie zostaje wysłana do wszystkich węzłów Systemu IIP. Jednak tego typu podejście nie chroni przed atakującymi stosującymi bardziej finezyjne metody ataków, przykładowo atakowanie wielu maszyn w ten sposób, że żadna pojedyn-



Rysunek 18. Dzialanie modulu globalnego wykrywania anomalii (PI-AD)

czo nie wykrywa ataku. Tego typu zagrożenia wykrywane na poziomie globalnym poprzez moduł PI-AD. Moduł LAD po wykryciu zbiorów częstych dokonuje przeglądu wszystkich zdarzeń SRE, zarejestrowanych w danym oknie czasowym. Wszystkie zdarzenia SRE, które nie wspierają żadnego z wykrytych zbiorów częstych (to znaczy nie są nadzbiorami jakiegokolwiek wykrytego zbioru częstego) zostają wysłane do dalszej analizy przez moduł PI-AD. Moduł ten, umieszczony w węźle centralnym, agreguje otrzymane SRE ze wszystkich węzłów Systemu IIP. Na zagregowanych danych dokonywane jest wykrywanie zbiorów częstych. Ich analiza pozwala wykryć anomalie analogicznie do operacji przeprowadzanych w module LAD. Rysunek 18 przedstawia działanie modułu PI-AD. Atakujący wysyła pewien ruch do wielu węzłów Systemu IIP. Ponieważ ruch jest bardzo mały, żaden z węzłów lokalnie nie wykrywa zagrożenia, natomiast rejestruje pojedyncze zdarzenia SRE. Wszystkie tego typu zdarzenia wysyłane są za pomocą LSA do modułu PI-AD. Tutaj po dokonaniu agregacji wykrywane są zbiory częste. Pojedyncze z punktu widzenia jednego węzła zdarzenia po zagregowaniu ujawniają powtarzalny wzorzec i pozwalają wykryć anomalie. Pozostałe kroki są analogiczne, jak w poprzednim przykładzie. Komunikat o wykrytej anomalii zostaje wysłany do podsystemu obliczania reputacji, który obniża poziom zaufania do atakującego. Uzyskana w ten sposób informacja zostaje rozesłana do wszystkich węzłów Systemu IIP.

5 System reputacyjny

Ujednolicone podejście do problematyki bezpieczeństwa w Systemie IIP stało sie możliwe dzięki wykorzystaniu systemów zaufania i reputacji jako najwyższej warstwy architektury bezpieczeństwa. Reputacja pełni tu funkcję jednolitej miary poziomu bezpieczeństwa węzłów i łączy, uwzględniającej zarówno zagro-

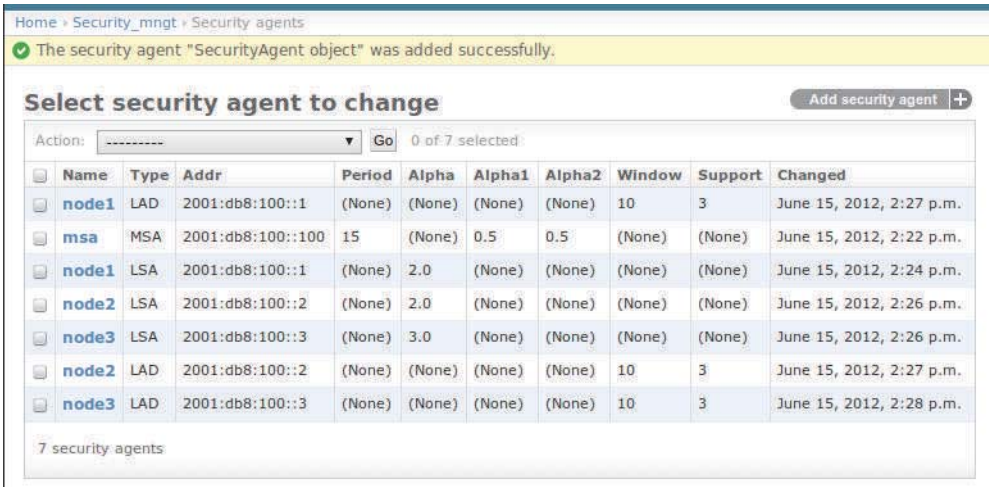
żenia "twarde", przeciw którym stosowane są zwykle kryptograficzne metody ochrony (w naszym przypadku: HMAC, zabezpieczenia protokołu SNMP oraz SSH), jak i "miękkie", wykrywane metodami detekcji anomalii. Kluczową decyzją poprzedzającą propozycję nowej architektury bezpieczeństwa dla Systemu IIP było rozstrzygnięcie, czy wartość reputacji węzła ma być ustalana na zasadzie rekomendacji, czy też obliczana na podstawie obserwacji ruchu pochodzącego z danego węzła przez węzły sąsiednie. W pierwszym przypadku (model rekomendacyjny) ustalanie poziomu zaufania i reputacji danego węzła następuje na podstawie subiektywnych ocen dokonywanych okresowo przez węzły sąsiednie (w naszym przypadku: LSA rezydujące w tych węzłach). Oceny te wykorzystują doświadczenie węzłów komunikujących się z węzłem ocenianym nabyte w ustalonym czasie lub w całej historii pracy sieci. W modelu tym bardzo często jest uwzględniana informacja uzyskana pośrednio, pochodząca od innych węzłów albo z niezależnych źródeł (np. ogólne informacje o zagrożeniach w sieci), por. [15]. W drugim przypadku (model walidacyjny) poziom zaufania i reputacji są obliczane jedynie na podstawie obserwacji zagrożeń Systemu IIP dokonywanych przez LSA w poszczególnych węzłach. Z punktu widzenia całego Systemu IIP wyniki te mają więc charakter subiektywny, jednak oparte są na bezpośrednich danych pomiarowych, czyli danych pochodzących z pierwszej ręki. W zaproponowanej architekturze bezpieczeństwa przyjęto walidacyjny model obliczania zaufania i reputacji i zaadaptowano go w scentralizowanej strukturze zaufania i reputacji obejmującej wszystkie węzły Systemu IIP. W naszym modelu poszczególne LSA raportują swoje, obliczone na podstawie obserwacji ruchu, oceny węzłów sąsiednich do centralnego MSA, który z kolei konsoliduje pozyskane w ten sposób dane i oblicza globalny poziom reputacji wszystkich węzłów. Spośród wielu znanych metod identyfikacji i konsolidacji danych reputacyjnych (por. np. [16]) przyjęto probabilistyczny model bayesowski, w którym na ocenę danego węzła składają się ważone opinie jego sąsiadów (ich wagi sa proporcjonajne do ich aktualnego poziomu reputacji), a płynność zmiany poziomu reputacji jest zagwarantowana przez uwzględnienie czynnika historycznego. Tak zaprojektowany system reputacyjny stanowiący element podsystemu bezpieczeństwa dla IIPS ma dwie podstawowe zalety. Po pierwsze, jest odporny na liczne ataki, na które podatne są systemy oparte na rekomendacji, por. [15]. Po drugie, pozostawia on LSA rozlokowanym w węzłach pewną autonomię przy obliczaniu wartości zaufania i reputacji ich sąsiadów, pozostawiając w kompetencji eksperta zarządzającego centralnie siecią LSA jedynie ustalanie parametrów wykorzystywanych w procedurze obliczania poziomu zaufania (zakres uwzględniania historycznych wartości reputacji, długość okna próbkowania, wartości *severity* i *intensity* dla poszczególnych zagrożeń, itp., por. [4], [5]. W efekcie pozwala to na sprawne i płynne obliczanie poziomu zaufania i reputacji w lokalnym otoczeniu każdego z węzłów oraz niemal natychmiastowa reakcje na dynamicznie zmieniający się poziom zagrożeń w sieci.

System reputacyjny zaprojektowany dla ochrony Systemu IIP jest przeznaczony do wykrywania zagrożeń i ataków w warstwie 2 (Level 2). Korzysta z oryginalnych modułów programowych i sprzętowych oraz ze standardowych metod

ochrony zaimplementowanych w podsystemie bezpieczeństwa Systemu IIP, to znaczy: modułów LAD (wykrywających ataki pochodzące z węzłów Systemu IIP i skierowane przeciwko innym węzłom Systemu IIP), sprzętowego modułu zabezpieczającego HMAC (umożliwiającego wykrywanie ataków pojawiających się na łączu między dwoma sąsiednimi węzłami Systemu IIP), a także z zabezpieczeń kryptograficznych protokołu SNMP i tuneli realizowanych z wykorzystaniem protokołu SSH (chroniących komunikację zarządzania Systemu IIP, podsystemu bezpieczeństwa Systemu IIP i komunikację innych protokołów, np. sygnalizację Równoległego Internetu CAN). Reasumując, zaprojektowany system reputacyjny pozwala oceniać poziom zaufania zarówno samych węzłów, jak i połączeń międzywęzłowych warstwy 2 (Level 2), a także dodatkowo poziom zaufania węzłów i połączeń systemu zarządzania.

Przyjęta scentralizowana architektura systemu reputacyjnego sprawia, że każdy LSA obserwuje ruch przychodzący od węzłów z nim sąsiadujących i przewiduje wystąpienie zagrożeń estymując wartość prawdopodobieństwa (*severities*) wystąpienia wykrytych anomalii, tak jak to już zostało wcześniej przedstawione tym rozdziale (por. także [5]). Wyniki uzyskane w takiej procedurze są okresowo przesyłane do MSA, a tam podlegają procedurze konsolidacji w celu obliczenia wartości poziomu zaufania i reputacji wszystkich węzłów Systemu IIP. Scentralizowana struktura systemu reputacyjnego umożliwia efektywne obliczanie tych wielkości oraz nadzorowanie zachowania wszystkich węzłów Systemu IIP w skali globalnej, co pozwala wykrywać ataki niezauważalne z lokalnej perspektywy jednego lub kilku sąsiadujących węzłów. Wadą takiej scentralizowanej architektury jest fakt, iż MSA staje się naturalnym punktem podatnym na atak odmowy usługi (np. DoS lub DDoS), awarii lub fizycznego odcięcia połączenia. Zagrożeniu temu można jednak przeciwdziałać proponując scenariusze awaryjne korzystające z faktu, iż wszystkie węzły lokalnie dysponują zgromadzona wiedzą na temat poziomu zaufania i reputacji swoich sąsiadów i w razie awarii MSA nadal prowadzą obserwację i bieżącą ocenę swoich sąsiadów. Można zatem zastosować procedurę odtworzenia MSA, używają np. podejścia zaproponowanego w pracy [17]. Wymaga ona w pierwszej kolejności wyboru węzła, w którym zostanie umieszczony nowy MSA, a następnie przesłania lokalnych informacji reputacyjnych zgromadzonych w każdym z LSA umieszczonych w węzłach Systemu IIP do MSA. W krótkim czasie nowy MSA jest w stanie zrekonstruować swoje globalne rekomendacje reputacyjne dla węzłów Systemu IIP poprzez konsolidację danych uzyskanych z LSA.

Jak to zostało przyjęte w architekturze bezpieczeństwa Systemu IIP, każdy LSA estymuje poziom zaufania do wszystkich swoich sąsiadów wykorzystując dwa podejścia: weryfikację ramek Systemu IIP (IIPS-PDU) poprzez sprawdzanie poprawności HMAC oraz wykonywanie algorytmów prowadzących do wykrywania anomalii w lokalnym module LAD. Konsolidując dane uzyskane w weryfikacji wykorzystującej HMAC oraz z poszczególnych algorytmów wykrywania anomalii można uzyskać liczbowe wartości poziomu zaufania połączeń międzywęzłowych. W szczególności, na podstawie takiej analizy można ustalić, czy źródłem wykrytych anomalii ruchu przychodzącego z sąsiedniego węzła jest niewła-



Rysunek 19. Panel do konfiguracji MSA, LSA i LAD w architekturze bezpieczeństwa Systemu IIP

ściwe funkcjonowanie węzła (spowodowane na przykład atakiem na ten węzeł), czy też zagrożenie występujące na trasie przesyłanych pakietów (występujące w pewnym miejscu ścieżki między węzłami). Inną przesłanką wystąpienia ataku mogłoby być zaobserwowanie nietypowego ruchu w sieci (na przykład, zalew nieoczekiwanym strumieniem danych) przy równoczesnym braku wskazania przez lokalny LAD niewłaściwego działania któregokolwiek sąsiedniego węzła. Przeciwnie, jeśli źródłem zaobserwowanych zaburzeń jest jeden z węzłów Systemu IIP, to analiza skonsolidowanych rekomendacji przeprowadzona przez MSA prowadzi do potwierdzenia hipotezy o niepoprawnym działaniu węzła. Taka sytuacja ma zazwyczaj miejsce, gdy negatywne rekomendacje (niskie wartości poziomu zaufania) nadchodzą od wszystkich sąsiadów komunikujących się z ocenianym węzłem.

System reputacyjny wyposażony został w interfejs zarządzania w postaci panelu dostępnego poprzez aplikację WWW (por. rysunek 19). Panel ten daje administratorowi możliwość ustawienia wszystkich parametrów zarówno indywidualnego LSA jak i modułu centralnego (MSA), wpływając tym samym na ustawienia globalne systemu reputacji. Dla poszczególnych typów modułów (LSA-/MSA) parametry te to długość okresu próbkowania, parametry formuły obliczenia reputacji oraz adres sieciowy węzła, por. [5]. Dodatkowo panel zarządzający ma możliwość konfiguracji parametrów podsystemu LAD, wpływając na częstość wykrywania oraz ocenę zdarzeń przez ten system. W obecnej implementacji wszystkie powyższe parametry mogą być ustalane za pomocą dołączonego do systemu interfejsu graficznego. Ustalanie wartości parametrów następuje eksperymentalnie, zgodnie z wiedzą ekspercką oraz długofalową obserwacją funkcjonowania systemu, w szczególności jego reakcji na dane klasy naruszeń bezpieczeństwa. Dzięki tym wartościom można uczynić system reputacyjnym mniej lub bardziej czułym na zidentyfikowane klasy ataków, natomiast poprzez

ustalenie czasu próbkowania możliwe jest sterowanie czasem reakcji systemu. Prezentowany interfejs WWW dla architektury bezpieczeństwa Systemu IIP, został zaprojektowany jako przyjazny użytkownikowi końcowemu oraz intuicyjny w obsłudze, przy jednoczesnym minimalnym wpływie działania interfejsu na strukturę węzłów w sieci. Aplikacja może być umiejscowiona praktycznie w dowolnym punkcie sieci. Została stworzona przy użyciu pakietu *Django* [18], który jest wysokopoziomowym, otwartym i darmowym frameworkiem przeznaczonym do tworzenia aplikacji internetowych. Zarówno *Django*, jak i *pySNMP* [19] (pakiet użyty do implementacji bezpiecznego systemu wieloagentowego) napisane są w języku Python, stąd ich kompatybilność jest wysoka, a integracja oraz eksploatacja ułatwione. Główne okno opisywanej aplikacji, prezentuje dostępne w systemie moduły LSA, MSA oraz LAD wraz z wyszczególnionymi ich nazwami, adresami sieciowymi, parametrami oraz czasem ostatniej ich zmiany. Z uwagi na dynamiczny charakter sieci, w każdym momencie administrator systemu może dodać lub usunąć węzeł z panelu zarządzania. Aplikacja w prosty oraz naturalny sposób umożliwia zmianę parametrów węzłów architektury bezpieczeństwa Systemu IIP.

Gdy wartość wybranego parametru zostanie zaktualizowana przez administratora, aktualna konfiguracja musi zostać przesłana do danego węzła. W tym celu *host*, na którym zainstalowana jest aplikacja WWW, wysyła wiadomość aktualizująca do węzła systemu bezpieczeństwa, którego konfiguracja uległa zmianie. Węzeł po otrzymaniu tej wiadomości aktualizuje swoją konfigurację. Powyższa komunikacja odbywa się z pomocą bezpiecznego kanału, zrealizowanego przez zaszyfrowane oraz uwierzytelnione wiadomości *Inform* protokołu SNMPv3. Następnie aktualna konfiguracja zapisana zostaje w lokalnej (z punktu widzenia aplikacji WWW) bazie danych. Obecna implementacja korzysta z relacyjnej bazy danych, zarządzanej za pomocą systemu *SQLite* [20]. Dostęp do opisywanego systemu zarządzania WWW zabezpieczony jest w podstawowy sposób, tj. za pomocą uwierzytelnienia użytkowników na podstawie ich nazw (login) oraz haseł, jednak w przyszłej, rozszerzonej wersji planowane jest zabezpieczenie tego dostępu za pomocą protokołu *HTTPS* (certyfikowany SSL).

6 Zabezpieczenie systemu zarządzania siecią i zarządzania systemem bezpieczeństwa

Jak wspomniano wcześniej (por. też [5]), elementy systemu reputacyjnego komunikują się wykorzystując protokół SNMPv3. Do zabezpieczenia procesu komunikacji systemu bezpieczeństwa (głównie systemu reputacyjnego), jak również pozostałej komunikacji ruchu zarządzania, został przedstawiony moduł *SNMP-PROXY*. Jest to nowy element architektury bezpieczeństwa Systemu IIP, który został zintegrowany z częścią systemu zarządzania IIP. Funkcjonalność realizowana przez *SNMP-PROXY* zaimplementowana została dla osiągnięcia dwóch celów. Pierwszym z nich jest dostarczenie usług bezpieczeństwa dla ruchu zarządzania Systemu IIP. Drugim celem jest udostępnienie podsystemu logowania

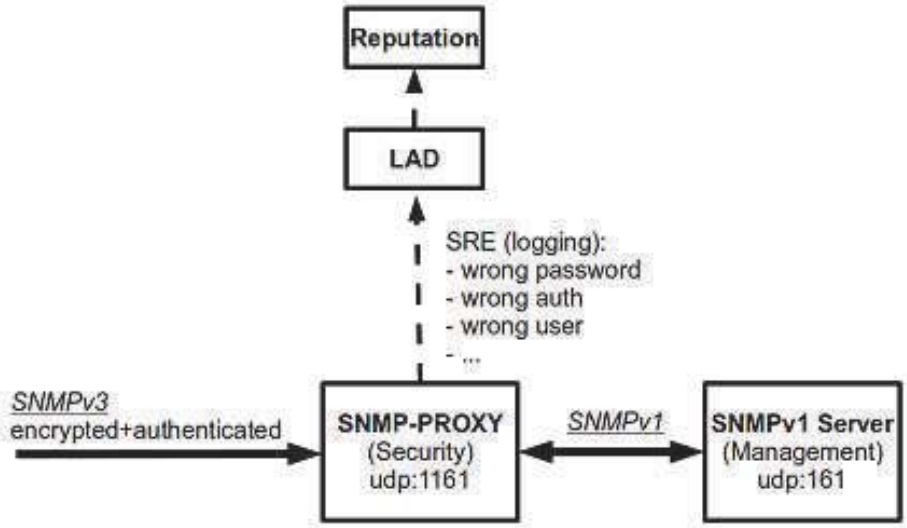
(logowanie SRE), który związany jest z ruchem SNMP. Architektura *SNMP-PROXY* zaprezentowana została na rysunku 20.

Zabezpieczenie ruchu zarządzania Systemu IIP osiągnięte jest za pomocą użycia protokołu SNMPv3 w miejsce protokołów o niższych wersjach (które są w stanie zagwarantować znacznie niższy poziom bezpieczeństwa). *SNMP-PROXY* oczekuje na połączenia, wymuszając na komunikujących się podmiotach użycie bezpiecznego protokołu wraz z uwierzytelnieniem za pomocą nazw użytkowników oraz haseł. Cały ruch SNMP zostaje wówczas zaszyfrowany oraz uwierzytelniony. Po udanym nawiązaniu połączenia otrzymane zapytanie SNMP przekazywane jest do właściwego serwera zarządzania SNMP, gdzie jest przetwarzanie oraz obsługiwane. Komunikacja między *SNMP-PROXY* a właściwym serwerem SNMP jest komunikacją lokalną i może odbywać się za pomocą niezabezpieczonych wersji protokołu. Jednak kluczowym aspektem dla funkcjonowania opisywanej architektury jest wymuszenie tego, by cały ruch kierowany do serwera SNMP przechodził przez opisywany moduł. *PROXY-SNMP*, z uwagi na wykorzystanie protokołu SNMPv3, może wymagać maksymalnie trzech parametrów bezpieczeństwa: nazwy bezpieczeństwa (najczęściej utożsamianej z nazwą użytkownika) i dwóch haseł używanych odpowiednio do szyfrowania oraz uwierzytelniania komunikatów. W naszej obecnej implementacji połączenie jest ustanowione wtedy i tylko wtedy, gdy wszystkie te parametry maja właściwie ustawione wartości. Moduł *SNMP-PROXY* został zaimplementowany z wykorzystaniem biblioteki *pySNMP* [19], która zezwala na użycie wybranych metod kryptograficznych. Jako algorytm kodów uwierzytelniających wiadomości dostępny jest schemat HMAC bazujący na funkcjach skrótu MD5 lub SHA-1. Natomiast do realizacji funkcji szyfrowania może zostać użyty jeden z następujących szyfrów: DES, 3DES oraz AES (z kluczami o wybranej długości bitowej: 128, 192, 256).

W celu dostarczenia informacji o zdarzeniach bezpieczeństwa dla ruchu SNMP (logowanie SRE), *SNMP-PROXY*, przetwarzając zapytania SNMP, loguje wszystkie podejrzane aktywności, które następnie są przekazywane do modułu wykrywania anomalii. Zdarzeniami, które są logowane i mogą być następnie rozpatrywane jako ataki są: błędna nazwa użytkownika, błędne hasło szyfrowania lub uwierzytelnienia, wersja protokołu inna niż SNMPv3, próba transmisji nieszyfrowanej lub nieuwierzytelnionej oraz inny zestaw modułów kryptograficznych niż ustalony przez administratora. Zdarzenia bezpieczeństwa logowane są w ustalonym formacie za pomocą funkcji systemowej *syslog()*. Pojedynczy wpis zawiera informacje o czasie zdarzenia, adresie źródłowym, porcie źródłowym, typie błędu wraz z dostępnym kontekstem protokołu SNMP (takim jak np. żądane zasoby, które nieuprawiony użytkownik chciał uzyskać).

7 Integracja podsystemu bezpieczeństwa w laboratorium PL-LAB

Opisane w poprzedzającej części tego rozdziału elementy podsystemu bezpieczeństwa zostaną zintegrowane z innymi elementami Systemu IIP w środowisku

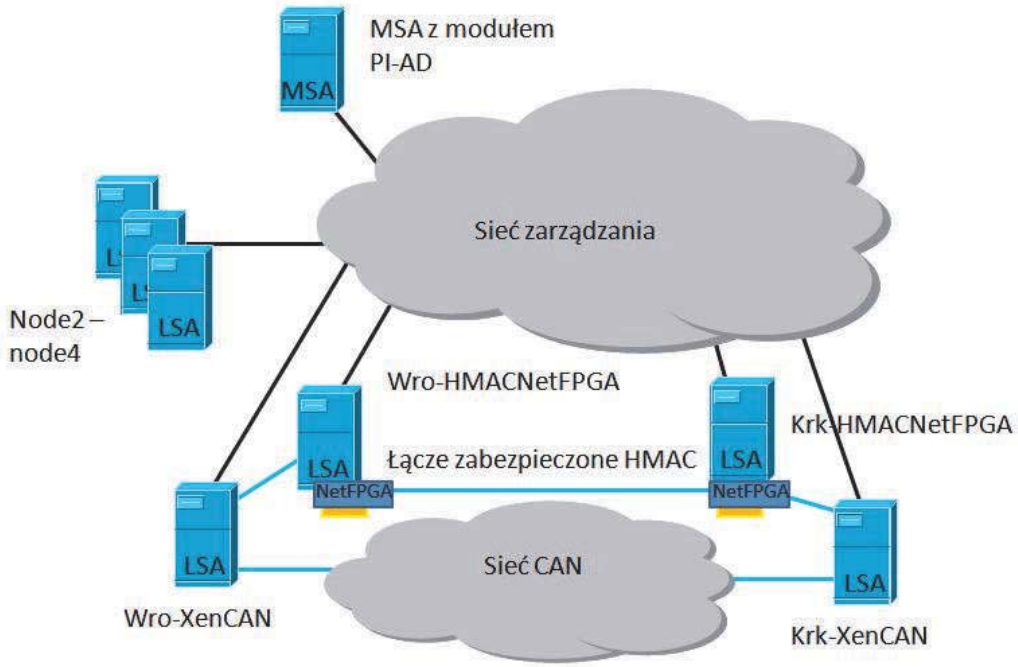


Rysunek 20. Architektura zabezpieczenia SNMP-PROXY

zdalnego laboratorium PL-LAB. Rysunek 21 przedstawia proponowaną topologię połączeń oraz wykorzystywane elementy. Zaproponowana topologia pozwoli na integrację wszystkich elementów podsystemu bezpieczeństwa, jak również integrację z innymi elementami Systemu IIP. Pozwoli to na przeprowadzenie nowych eksperymentów, podczas których zostaną dobrane parametry opisywanych metod tak, aby wykrywać istotne z punktu widzenia Systemu IIP anomalie.

Najważniejszym elementem zaproponowanej integracji jest połączenie dwóch węzłów Systemu IIP łączem zabezpieczonym za pomocą modułu HMAC. Węzłami terminującymi zabezpieczony ruch będą węzły Systemu IIP zrealizowane w technologii Xen uruchomione we Wrocławiu oraz Krakowie. Moduł HMAC, zaimplementowany na kartach NetFPGA zostanie uruchomiony w obu lokalizacjach na dedykowanych maszynach. Dodatkowo w lokalizacji Wrocław zostanie uruchomiony w środowisku wirtualnym centralny węzeł zarządzania z oprogramowaniem MSA i PI-AD oraz dodatkowo trzy węzły z modułami LSA i LAD. Podjęto już prace zmierzające do uruchomienia na węzłach Xen maszyn internetu CAN oraz podłączenia ich do już uruchomionej w laboratorium PL-LAB sieci CAN. Pozwoli to wygenerować na łączach tych maszyn ruch zgodny ze specyfiką Systemu IIP.

Na aktualnym etapie prac wszystkie siedem (poza MSA) węzłów posiada zainstalowane oprogramowanie systemu reputacyjnego (agenta LSA) oraz moduły wykrywania anomalii LAD, w oparciu o obie metody. W celu umożliwienia pobierania danych przez modułu LAD na tych węzłach zostały zainstalowane i odpowiednio skonfigurowane demony rsyslog oraz snmpd. Na ósmym węźle został uruchomiony moduł obliczania reputacji MSA, jak też moduł globalnego wykrywania anomalii - PI-AD. Dodatkowo prowadzone są prace zmierzające do uruchomienia oprogramowania HMAC na kartach NetFPGA oraz integracji in-



Rysunek 21. Topologia sieci eksperymentalnej integrującej wszystkie podsystemy bezpieczeństwa w sieci PL-LAB

terfejsu diagnostycznego HMAC z modułem LAD. Wszystkie przedstawione na rysunku 21 węzły są podłączone do podsieci zarządzania laboratorium PL-LAB.

8 Wnioski

Przedstawiono aktualny zarys prac implementacyjnych nad architekturą bezpieczeństwa zorientowaną na przeciwdziałanie zagrożeniom na poziomie 2 Systemu IIP. W toku prac wykonano i wstępnie przetestowano programową oraz sprzętową realizację modułu generującego oraz weryfikującego sygnaturę skrótu HMAC-SHA-1, dołączaną do ramki System IIP w celu ochrony przed atakami przeciwko systemowi transmisyjnemu Systemu IIP, a także implementacje modułów wykrywania anomalii i systemu reputacyjnego. Wykorzystując niewielkie środowisko testowe z systemem komunikacyjnym zorganizowanym w oparciu o protokół SNMPv3 dokonano szeregu eksperymentów mających na celu weryfikację skuteczności mechanizmów bezpieczeństwa w obecności wybranych metod ataków. Zwrócono uwagę na niektóre aspekty implementacji mogące mieć wpływ na dalsze prace integracyjne w obrębie Systemu IIP. W obecnej chwili trwają końcowe prace nad integracją mechanizmów bezpieczeństwa z Systemem IIP poprzez ogólnopolską sieć testową PL-LAB.

Literatura

1. Cabaj K. i in., Implementation and testing of Level 2 security architecture for the IIP System, Przegląd Telekomunikacyjny - Wiadomości Telekomunikacyjne, Vol.85(81), No.8-9, pp.1426-1435, 2012.
2. Konorski J. i in., Implementacja sprzętowa modułu HMAC-SHA-1 do ochrony komunikacji w Systemie IIP, Przegląd Telekomunikacyjny - Wiadomości Telekomunikacyjne, Vol.85(81), No.8-9, pp.1418-1425, 2012.
3. Burakowski W., Tarasiuk H., and Beben A., System IIP for supporting „Parallel Internets (Networks)", FIA meeting, Ghent 2010, fi-ghent. fiweek.eu/files/2010/12/1535-4-System-IIPFIA-Ghent-ver1.pdf
4. Cabaj K. i in., Architektura bezpieczeństwa Systemu IIP na poziomie wirtualizacji zasobów, Przegląd Telekomunikacyjny - Wiadomości Telekomunikacyjne, Vol.84(80), No.8-9, pp.846-851, 2011.
5. Cabaj K. i in., Architektura bezpieczeństwa Systemu IIP, rozdział w: W. Burakowski, P. Krawiec [Eds.], Inżynieria Internetu Przyszłości, Vol.1, pp.43-60, Warszawa 2012.
6. Konorski J. i in., A Virtualization-Level Future Internet Defense-in-Depth Architecture, in: Communications in Computer and Information Science (CCIS), vol.335, Recent Trends in Computer Networks and Distributed Systems Security, Part 1, pp. 283-292, Springer-Verlag, Berlin Heidelberg 2012.
7. Netlink Protocol Library Suite (libnl). http://www.infradead.org/ tgr/libnl/.
8. Crypto++® Library 5.6.1. http://www.cryptopp.com/.
9. IEEE. 1076-2008 IEEE Standard VHDL Language Reference Manual. 2009. ISBN:978-0-7381-5800-6.
10. RFC 2202. Test Cases for HMAC-MD5 and HMAC-SHA-1", September 1997.
11. Burgess M., Two dimensional time-series for anomaly detection and regulation in adaptive systems, Proc. IFIP/IEEE 13th Int. Workshop on Distributed Systems: Operations and Management, DSOM 2002, pp. 169-185
12. D-ITG Distributed Internet Traffic Generator, http://www.grid.unina. it/software/ITG/
13. Green D.M., Swets J.A., Signal detection theory and psychophysics, John Wiley and Sons, New York 1966.
14. Agrawal R., Srikant R., Fast algorithm for mining association rules, In: J.B. Bocca, M. Jarke, and C.Zaniolo, [ed]. Proceedings of 20th International Conference on Very Large Databases, pp. 487-499, 1994.
15. Konorski J., Orlikowski R., DST-Based Detection of Noncooperative Forwarding Behavior of MANET and WSN Nodes, Proc. 2nd Joint IFIP WMNC., Gdansk, Poland, 2009.
16. Ciszkowski T. et al., Towards Quality of Experience-based Reputation Models for Future Web Service Provisioning, Telecommunication Systems, Vol.51, No.4, pp.283-295, 2012.
17. Oryńczak G., Kotulski Z., Notary-based self-healing mechanism for centralized peer-to-peer infrastructures, Annales UMCS Informatica AI XII, 1, pp.97–112, 2012.
18. https://www.djangoproject.com/
19. http://pysnmp.sourceforge.net/
20. http://www.sqlite.org/