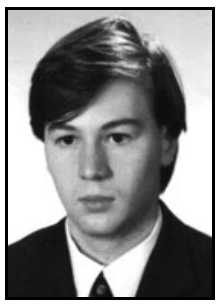
Grzegorz SUŁKOWSKI [2], **Maciej TWARDY** [2], **Kazimierz WIATR** [1,2]
[1] AKADEMIA GÓRNICZO-HUTNICZA
[2] ACK CYFRONET AGH

Weryfikacja reguł bezpieczeństwa wspomagana mechanizmami pamięci podręcznej w sprzętowej implementacji systemu bezpieczeństwa typu firewall

Mgr inż. Grzegorz SUŁKOWSKI



Ukończył elektronikę na Wydziale EAIE Akademii Górniczo-Hutniczej w Krakowie. Od 2006 roku jest Konstruktorem Systemów Obliczeniowych w dziale Archiwizacji i Bezpieczeństwa Danych w ACK CYFRONET AGH. Jego zainteresowania skupiają się wokół algorytmów obliczeniowych oraz ich realizacji w logice reprogramowalnej.

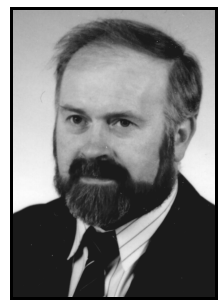
e-mail: Grzegorz.Sulkowski@cyfronet.pl

Mgr inż. Maciej TWARDY

Ukończył elektronikę na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Od 2005 roku kieruje Działem Archiwizacji i Bezpieczeństwa Danych w Akademickim Centrum Komputerowym CYFRONET AGH. Jego zainteresowania związane są z szeroko pojętą informatyką oraz projektowaniem układów cyfrowych w oparciu o logikę reprogramowalną.

e-mail: Maciej.Twardy@cyfronet.pl

Prof. dr hab. inż. Kazimierz WIATR



Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na AGH w Krakowie oraz Dyrektor ACK Cyfronet AGH. Prowadzone prace badawcze dotyczą systemów wizyjnych, systemów wieloprocesorowych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.

e-mail: wiatr@agh.edu.pl

Streszczenie

W niniejszym artykule autorzy dokonują przeglądu istniejących algorytmów klasyfikacji pakietów celem adaptacji najodpowiedniejszego spośród nich dla potrzeb budowanego systemu zabezpieczeń sieciowych klasy Firewall. Równocześnie prezentują koncepcje zwiększenia całkowitej wydajności proponowanego rozwiązania poprzez zastosowanie dodatkowych mechanizmów wykorzystujących m.in. pamięci podręczne, potokowość oraz zrównoleglenie przetwarzania danych.

Słowa kluczowe: systemy bezpieczeństwa informatycznego, układy programowalne, języki opisu sprzętu, firewall, pamięci podręczne, potokowość, przetwarzanie równoległe.

Security rules verification mechanism supported by local cache memory for the hardware Firewall security system

Abstract

In this paper authors present their research into the actual state of the hardware implemented packet classification algorithms for the adaptation into their implementation of the hardware Firewall security system. The paper also describes the idea of enhancing the overall processing efficiency by using additional mechanisms like local cache memory, pipelining and parallel processing.

Keywords: information security systems, programmable logic, hardware description language, firewall, packet classification algorithms, cache memory, pipelining, parallel processing.

1. Wstęp

Spośród wszystkich elementów składowych sprzętowego systemu bezpieczeństwa typu Firewall, największy wpływ na jego docelową wydajność ma blok weryfikujący przetwarzane dane pod kątem zgodności z przyjętą polityką bezpieczeństwa. Narzut czasowy związany z analizą danych nagłówków transmitowanych pakietów, szczególnie uwidaczniający się w wypadku konwencjonalnych sekwencyjnych metod klasyfikacyjnych, stanowi istotne pole do sprzętowej optymalizacji z wykorzystaniem ogromnego potencjału układów reprogramowalnych FPGA (ang. Field Programmable Gate Array) [8]. Przeniesienie takiej funkcjonalności do postaci sprzętowej, w zamierzeniach autorów, ma pozwolić na osiągniecie wysokiego poziomu bezpieczeństwa danych oraz dużej wydajności ich przetwarzania, pozwalającej na zaspokojenie potrzeb wynikających z dynamicznie rozwijających się technologii komunikacji sieciowych. W wypadku standardu Ethernet, aktualnie dostępne na rynku rozwiązania umożliwiają przesyłanie danych z prędkościami do 10Gb/s. Przy takich przepustowościach programowe systemy analizy i klasyfikacji pakietów nie są w stanie zagwarantować wystarczającej wydajności, ze względu na ograniczenia płynące z sekwencyjnego charakteru przetwarzania reguł bezpieczeństwa. Niezbędne w takiej sytuacji staje się sprzętowe wspomaganie procesu analizy danych. Nurt prac badawczych w tym obszarze rozwija się coraz intensywniej. Pojawiają się kolejne opracowania sprzętowych algorytmów klasyfikacji pakietów, oparte głównie o pamięci typu TCAM (ang. Ternary Content Addressable Memory) oraz rozwiązaniach BVT (ang. Bit Vector Trees). Nie rozwiązują one jednak wszystkich problemów; newralgicznym elementem pozostają nadal reguły bezpieczeństwa zawierające definicję portów, gdzie pamięci TCAM, ze względu na swoją budowę, nie nadają się do klasyfikowania zakresów portów postaci np. 40 – 57. W tej sytuacji koniecznym jest poszukiwanie innych, efektywniejszych metod realizacji tej funkcjonalności, przy założeniu wykorzystywania pamięci TCAM w części weryfikującej adresy sieciowe. Sumaryczne opóźnienie, dla tak zdefiniowanego toru analizy pakietów, będzie wypadkową czasu wyszukiwania adresów w pamięci TCAM, zakresu portów przy zastosowaniu dedykowanych algorytmów oraz opóźnień wnoszonych przez dodatkową logikę używaną np. do określania priorytetów reguł.

Proponowana przez autorów architektura pamięci podręcznej nagłówków (ang. cache memory) pozwoli przyśpieszyć analizę pakietów sieciowych w implementowanym w układach FPGA systemie bezpieczeństwa tak, aby wprowadzane opóźnienie przetwarzania danych było konkurencyjne w stosunku do alternatywnych rozwiązań.

2. Istniejące rozwiązania

Zagadnienie sprzętowej implementacji algorytmów wyszukiwania wzorców w układach FPGA, z racji bardzo szerokiego obszaru praktycznych zastosowań obejmujących różnorodne systemy zabezpieczania przetwarzania i przesyłania danych (IDS – ang. Intrusion Detection System, IPS – ang. Intrusion Prevention System, Firewall'e, itp.), stało się przedmiotem zainteresowania wielu ośrodków naukowych na świecie. Wyróżnić można kilka

głównych nurtów, wokół których koncentrują się prace badawcze. Pierwszym z nich jest wykorzystywanie do wyszukiwania wzorców skończonych automatów stanów FSM (ang. Finite State Machine). Spośród wielu prac związanych z tym tematem na szczególną uwagę zasługują badania realizowane przez G.Tripp [7]. Proponuje on rozwiązanie bazujące na zrównolegleniu przetwarzania danych przy wykorzystaniu grupy współbieżnie pracujących automatów stanu. Każdy z pojedynczych FSM'ów analizuje wejściowy strumień danych pod kątem wystąpień zadanego wzorca. Ponieważ operuje on na słowach o szerokości 8 bitów, w celu przyśpieszenia operacji wyszukiwania dane organizuje się w większe bloki np. 32 bitowe. Rozwiązanie takie wymusza odpowiednie zdefiniowanie wzorca, tak by zapobiec sytuacji, w której jest on krótszy niż okno przeszukiwania. W takiej sytuacji, w wypadku wystąpienia przesunięcia względem przeszukiwanego wzorca, analizowany ciąg nie zostanie poprawnie zidentyfikowany. Aby uniknąć opisanego zjawiska, zaproponowano implementację mechanizmów śledzenia historii wystąpień każdego z podwzorców (wynikającego z dekompozycji pomiędzy wszystkie FSM'y) na każdym możliwym słowie okna przeszukiwania, co komplikuje cały algorytm wyszukiwania. Wadą takiego rozwiązania jest fakt, że dla każdego nowo dodanego wzorca należy przekonfigurować wszystkie pracujące automaty stanów, powodując tym samym zatrzymanie pracy całego układu analizującego dane. Uzyskiwane wyniki implementacji [7]: częstotliwość pracy około 149MHz oraz przepustowość wyszukiwania dla wzorca o długości 32 bitów wynoszącą 4,7 Gbps, spełniają wymagania stawiane przez sieci o przepływnościach 1 Gbps. Należy jednak zauważyć, że złożoność czasowa dla procesu wyszukiwania jest postaci *O(n)*, co przy dużej liczbie wzorców wprowadzać znaczne opóźnienia a tym samym degradować wydajność całego systemu.

Odmienną koncepcję przyjęto w pracy J.W. Lockwood'a [2] przedstawiającej klasyfikator pakietów sieciowych dla systemu IDS implementowanego w FPGA. W proponowanym rozwiązaniu mechanizm wyszukujący podzielony jest na dwa bloki funkcjonalne. Pierwszy z nich, bazujący na pamięci TCAM, analizuje nagłówki pakietów sieciowych pod kątem wystąpienia adresów identycznych ze zdefiniowanymi w regułach filtrujących. Drugi blok, implementowany jako binarne drzewo przeszukiwań, odpowiada za odnajdywanie w danych wejściowych zakresów portów źródłowych i docelowych pasujących do wzorców zdefiniowanych w regułach filtrującym. Ponieważ proponowany mechanizm operuje na dwóch współbieżnych blokach, w projekcie zaimplementowano dodatkową logikę weryfikującą poprawność klasyfikowania pakietów. Zastosowanie pamięci TCAM pozwala na przeszukiwanie reguł filtrujących ze złożonością czasową *O(1)* zapewniając tym samym stały, niezależny od liczby reguł, czas przetwarzania. Nie mniej jednak przeszukiwanie drzewa binarnego charakteryzuje się złożonością obliczeniową *O(lonN),* oraz złożonością $O(N^2)$ zapotrzebowania na pamięć ram, wynikającą z dekompozycji reguł filtrujących zakresy portów z postaci przedziałów na postać prefiksową [2].

Wykorzystanie mechanizmów wyszukujących wzorce dla klasyfikatora pakietów sieciowych w sprzętowej realizacji Firewall'a przedstawione jest w publikacji J. Loinig et al. [3]. Iimplementacja ta operuje na algorytmie liniowego przeszukiwania reguł filtrujących zapisanych w wewnętrznej pamięci RAM układu FPGA. Celem przyśpieszenia procesu klasyfikacji pakietów sieciowych nagłówek pakietu wejściowego przetwarzany jest równolegle w kilku klasyfikatorach jednocześnie z wykorzystaniem dwuportowego dostępu do wewnętrznych pamięci RAM. Aby w tej sytuacji zagwarantować poprawną identyfikację pakietu, wprowadzono dodatkowe mechanizmy znacznikowe. Poszczególnym regułom filtrującym nadano indywidualne priorytety, dzięki czemu możliwe jest odtworzenie hierarchii reguł niezależnie od kolejności ich ulokowania w pamięci RAM . Dla zapewnienia ciągłości potokowego przetwarzania pakietów dla każdego z torów transmisyjnych zostały zastosowane dodatkowe bufory FIFO (ang. First In, First Out). Prototyp klasyfikatora zaimplementowany w układzie FPGA pozwalał na pracę z częstotliwością dochodzącą do 139MHz. Weryfikacja pojedynczego pakietu danych zajmowała 239 cykli zegarowych. Takie wyniki mieszczą się w wymaganiach stawianych dla sieci o przepustowości do 1Gbps (częstotliwość pracy 125MHz). Jednakże wprowadzane opóźnienie, wynikające z implementacji liniowego algorytmu przeszukiwania o czasowej złożoności przeszukiwania *O(n),* może znacząco wpływać na wydajność w sensie QoS.

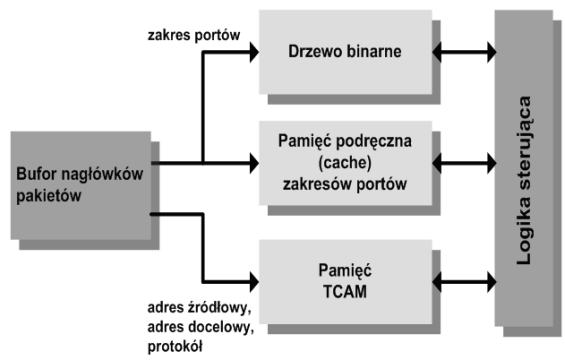
Z przedstawionej analizy wynika, że dobre wyniki dla czasowej złożoności obliczeniowej można uzyskać stosując pamięci TCAM jako podstawowy blok przeprowadzający operacje wyszukiwania wzorców, będących jednocześnie regułami filtrującymi adresację sieciową. Dla filtracji portów i zakresów portów należy zastanowić się nad optymalizacją proponowanych rozwiązań celem zminimalizowania wpływu złożoności czasowej *O(logN)* kosztem zwiększenia zasobów pamięciowych i logicznych. Można to osiągnąć przez zastosowanie pamięci podręcznych typu *cache* wspomagających blok filtracji portowej.

3. Architektura klasyfikatora weryfikującego

Ze względu na postawione założenia projektowe [5, 6] oraz wnioski z analizy istniejących rozwiązań, autorzy opracowali ulepszone rozwiązanie klasyfikatora reguł filtrujących, w skład którego wchodzą następujące bloki funkcjonalne:

- filtr adresów sieciowych – pamięć TCAM ,
- filtr portów – binarne drzewo decyzyjne,
- pamięć podręczna portów – pamięć CAM (ang. Content-Addressable Memory).

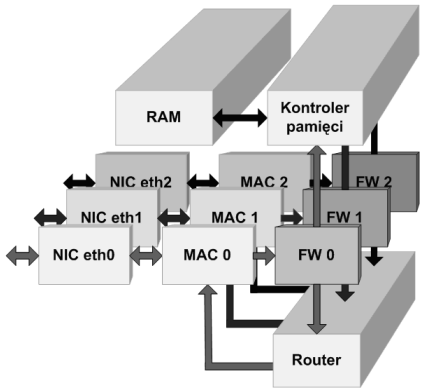
Wymienione powyżej bloki stanowią jądro (kernel) mechanizmu weryfikującego reguły firewall'a. Zastosowanie pamięci podręcznej dla buforowania reguł definiujących zakresy portów, implementowanej jako pamięć CAM, pozwoli zredukować złożoność czasową przeszukiwania do postaci *O(1)*. Należy jednak zaznaczyć, że wynik taki zostanie osiągnięty tylko dla reguł już raz przeanalizowanych, czyli w przypadku, gdy nagłówek pakietu sieciowego się powtórzy. Założenie to jest jak najbardziej słuszne i wynika z wniosków dla statystycznej analizy ruchu sieciowego. Schemat proponowanej architektury kernela (FW) przedstawiony jest na rys. 1.



Rys. 1. Schemat architektury sprzętowego Firewall'a
Fig. 1. The architecture diagram of the hardware Firewall

Blokowy schemat architektury sprzętowego firewall'a, uwzględniający omawiany klasyfikator weryfikujący, moduły warstwy MAC i PHY przedstawiony jest na rys. 2.

Przeszukiwanie reguł filtrujących podzielone zostało stosownie do zaproponowanych bloków funkcjonalnych. Dla zobrazowania mechanizmu dekompozycji reguł filtrujących dla filtru segmentów sieciowych na reguły weryfikujące rozpatrzmy poniższy przykład.



Rys. 2. Schemat architektury sprzętowego Firewall'a
Fig. 2. The architecture diagram of the hardware Firewall

Tablica tab. 1 przedstawia przykładowy zestaw reguł filtrujących systemu firewall, zawierających dwie części - adresową i portową.

Tab. 1. Tablica reguł filtrujących
Tab. 1. Filter rules table

Nr.	Adres źródłowy	Adres docelowy	Protokół	Port źródłowy	Port docelowy
1	*	192.168.0.1	tcp	*	21
2	192.168.2.0/24	192.168.0.0/24	*	*	5000:5010
3	192.168.0.0/24	192.168.0.5	udp	*	53
4	192.168.0.7	192.168.0.6	tcp	*	22
5	*	*	tcp	20	6000:6330

Część adresowa wraz z kolumną określającą typ protokołu implementowana jest w pamięci TCAM pod adresem odpowiadającym numerowi reguły. Zawartość komórki pamięci dla reguły nr 1 będzie miała następującą postać:

1 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx 11000000 10101000 00000000 00000001

Pierwsza cyfra oznacza typ protokołu (tcp=1, upd=0), kolejne 32 znaki x oznaczają wartość dowolną dla adresu źródłowego, następne 32 znaki oznaczają adres docelowy. Podobna notacja jest stosowana dla bloku cache portów. Ostatni blok, filtr portów, jest obecnie w fazie koncepcyjnej. Docelowo implementacja będzie bazowała na algorytmie drzewa binarnego [4]. Zadaniem logiki sterującej jest właściwa ocena wyników weryfikacji otrzymanych z bloków filtrujących, podjęcie decyzji o akceptacji przetwarzanego pakietu ACCEPT lub odrzuceniu DROP i przekazanie jej do warstw końcowych (warstwy kolejkowania i buforowania) bloku FW (rys. 1 i rys. 2), skąd przetwarzany pakiet albo zostanie przeniesiony do kolejki TX modułu MAC (decyzja ACCEPT), albo upuszczony (decyzja DROP).

4. Podsumowanie oraz wyniki implementacji wybranych modułów

W pracy przedstawiono pomysł poprawionego rozwiązania klasyfikatora pakietów sieciowych wykorzystującego mechanizm buforowania podręcznego dla sprzętowo implementowanego firewall'a. Wstępne wyniki syntezy dla modułów pamięci podręcznej (tab. 2) oraz filtru adresów sieciowych (tab. 3) w zależności od liczby implementowanych reguł filtrujących. Należy zauważyć, że częstotliwość pracy uzyskana dla pamięci TCAM nie będzie miała decydującego wpływu na szybkość przetwarzania, która głównie będzie zależała od szerokości szyny przetwarzania (obecnie 32bity) oraz zastosowania pipelining'u. Ostatnim elementem, nad którym trwają obecnie prace jest dobranie odpowiedniej architektury do implementacji binarnego drzewa przeszukiwań, tak by zminimalizować zapotrzebowanie na przestrzeń adresową.

Tab. 2. Wyniki syntezy dla bloku pamięci podręcznej implementowanej jako pamięć CAM w zależności od liczby reguł filtrujących
Tab. 2. Synthesis results for cache memory implemented as CAM in dependences on number of filter rules

	32 reguły	64 reguły	128 reguły	256 reguły	512 reguły
Slices	164	278	506	973	1886
Filp-Flops	74	107	172	301	558
LUTs	299	517	945	1829	3525
BRAMs	4	8	16	32	64
Max. Freq.	155MHz	155MHz	138MHz	125MHz	118MHz

Tab. 3. Wyniki syntezy dla filtru adresów sieciowych implementowanego jako pamięć TCAM w zależności od liczby reguł filtrujących
Tab. 3. Synthesis results for network address filter implemented as TCAM in dependences on number of filter rules

	32 reguły	64 reguły	128 reguły	256 reguły	512 reguły
Slices	784	1380	2597	5025	9803
Filp-Flops	86	88	90	92	94
LUTs	998	1640	2969	5631	10801
Max. Freq.	104MHz	97MHz	77MHz	72MHz	70MHz

Praca naukowa finansowana ze środków na naukę w latach 2006-2008 jako projekt badawczy.

5. Literatura

[1] 802.3 IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks, 2002.
[2] J.W.Lockwood, H. Song: Efficient Packet Classification for Network Intrusion Detection Rusing FPGA. International Symposium on Field-Programmable Gate Arrays (FPGA'05), Monterey, CA, Feb 20-22, 2005.
[3] J. Loinig, J. Wolkerstorfer, A, Szekely: Packet Filtering in Gigabit Networks Using FPGAs. Austrochip 2007 - Proceedings of the 15th Austrian Workshop on Microelectronics, ISBN 978-3-902465-87-0, Oct 2007.
[4] M. Á. Ruiz-Sánchez, E.W. Biersack, W. Dabbous: Survey and Taxonomy of IP Address Lookup Algorithms. IEEE Network, March/April 2001.
[5] G.Sułkowski, M. Twardy, K. Wiatr: Implementacja systemu bezpieczeństwa typu Firewall dla potrzeb sieci Ethernet w oparciu o układy reprogramowalne FPGA, Konferencja KNWS'07, Kwartalnik Pomiary, Automatyka i Kontrola, Warszawa, nr 5, 2007, s. 114-116
[6] G.Sułkowski, M. Twardy, K. Wiatr: Implementacja standardu sieci Ethernet IEEE 802.3 w układach FPGA na potrzeby systemu bezpieczeństwa typu Firewall, Konferencja: Reprogra-mowalne Układy Cyfrowe 2007, Kwartalnik Pomiary, Auto-matyka i Kontrola nr 7, s.
[7] G.Tripp: A parallel "String Matching Engine" for use in high speed Network intrusion detection systems. J Comput Virol (2006) 2:21–34, Springer-Verlag France 2006.
[8] K. Wiatr: Akceleracja obliczeń w systemach wizyjnych, Wydawnictwa Naukowo-Techniczne, Warszawa 2003.

Artykuł recenzowany