# DESIGNING INTRUSION DETECTION SYSTEM FOR FPGA TECHNOLOGY

Petr KOBIERSKÝ, Bachelor Degree Programme (3)
Dept. of Computer Systems, FIT, BUT
E-mail: xkobie00@stud.fit.vutbr.cz

Supervised by: Ing. Jan Kořenek

## ABSTRACT

Intrusion Detection System (IDS) is an integrated software/hardware tool capable of detecting unauthorised access to computer systems and malicious network traffic such as viruses, trojan horses and worms. This paper introduce architecture of the hardware accelerated IDS for Field-Programmable Gate Arrays (FPGAs). The approach based on on Nondeterministic Finite Automation (NFA) [1] with optimalization focused to Snort [2] ruleset is used for payload scanning. Proposed architecture was implemented in VHDL and was tested in a hardware.

## 1 ÚVOD

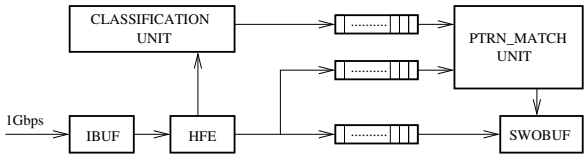
V posledním desetiletí jsme svědky výrazného rozvoje síťových technologií. S rozvojem Internetu se ale také množí virové hrozby a útoky na počítačové systémy. K zajištění bezpečnosti počítačových sítí již nestačí firewaly, ale je nutné použít sofistikovanější systémy pro detekci nebezpečného provozu (IDS). IDS kombinuje funkci klasifikace paketů, která je obsažená v každém firewallu, a funkci vyhledávání vzorů v datech paketů. Tímto umožňuje IDS detekovat daleko více hrozeb šířicích se po Internetu.

IDS musí hledat řádově tisíce řetězců a regulárních výrazu na rychlostech 10 Gb/s a více. To je pro softwarová řešení, která jsou omezena zejména pomalou sběrnici a zpracováním na obecném procesoru, neřešitelná úloha. Běžně používaný volně šiřitelný IDS systém Snort dosahuje v závislosti na počtu IDS pravidel propustnosti v řádech 100 Mb/s, což je pro dnešní gigabitové sítě nedostatečné. Aby bylo možné dosáhnout těchto, ale i vyšších rychlostí, je nutné přesunout časově kriticou operaci vyhledávání řetězců do hardware.

Pro relalizaci rychlého vyhledávání vzorů se jeví jako vhodná technologie FPGA. Ta poskytuje možnost rekonfigurace, čímž umožňuje vytvořit obvod s požadovanými vlastnostmi, jakými jsou rychlost hledání vzorů nebo použitá plocha čipu. Existuje řada přístupu na bázi FPGA pro rychlé vyhledávání vzorů, nejlepších výsledku však dosahuje přístup na bázi NFA publikovaný panem Clarkem [1]. Z tohoto přístupu vycházím i v této práci.

## 2 ARCHITEKTURA

Navržená architektura se skládá z jednotky pro klasifikaci paketů, jednotky pro vyhledávání vzorů a z několika dalších komponent. Blokové schéma architektury je znázorněno na obrázku 1. Paket je přijat ze vstupního rozhraní do vstupního bufferu (IBUF). Pomocí jednotky HFE (Header Field Extractor) jsou TCP/IP položky hlavičky (zdrojová a cílová adresa, porty, protokol...) převedeny na unifikovanou hlavičku (UH), a ta je předána do klasifikační jednotky. Celý paket je uložen do výstupního bufferu (SWOBUF), kde čeká na výsledek analýzy. Klasifikační jednotka porovná UH s pravidly IDS systému a jednotka pro vyhledávání řetězců (PTRN_MATCH) vyhledá vzory vyskytující se v paketu. Pravidla nalezená klasifikační jednotkou se vyhodnotí s výsledkem vyhledávací jednotky. Pokud přijatý paket odpovídá alespoň jednomu pravidlu, je paket exportován do software, v opačném případě je zahozen.




Obrázek 1: Navrhovaná architektura

## 3 BLOK PRO VYHLEDÁVÁNÍ VZORŮ

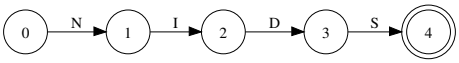
Jádrem navržené architekury je jednotka pro vyhledávání vzorů. Obsah jednotky je generován se zadané množiny řetězců. Při převodu jsou nejprve řetězce a regulární výrazy převedeny na nedeterministický konečný automat (NFA), který lze dále transformovat do ekvivalentní hardwarové reprezentace zachycené na obrázku 4. Pokud je použit NFA, který přijímá 8 bit (1 znak) v jednom hodinovém cyklu a systémová frekvence je 100 MHz, je design schopen podle vzorce $Throughtput = DataWidth * Frequency$ dosáhnout propustnosti 800 Mb/s. Propustnost lze zvýšit převodem NFA na rozšířený automat (ENFA), který přijímá více znaků v jednom hodinovém cyklu. Tím lze podle velikosti hledané množiny řetězců dosáhnout propustnosti až 10 Gb/s. Větších propustností nelze s tímto přístupem dosáhnout vzhledem k nedostatečné kapacitě současných FPGA a problémům s omezenou kapacitou propojovací matice. V budoucnosti je nutné se soustředit na nalezení vhodného kódování stavů a rozdělení NFA na několik nezávislých části.

| Typ řetězců | 0.8 Gb/s | 1.6 Gb/s | 3.2 Gb/s | 6.4 Gb/s | 12.6 Gb/s |
|---|---|---|---|---|---|
| Náhodně[1] | 33 % | 36 % | 61 % | 99 % | 187 % |
| Snort 2.4.3[2] | 38 % | 39 % | 51 % | 85 % | 147 % |

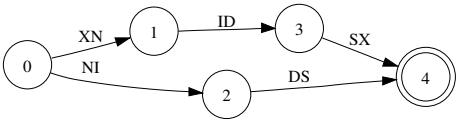
Tabulka 1: Využití plochy čipu 2VP50 pro různé propustnosti

[1] 1024 řetězců s rozložením délky Normal(15,3) - 14 917 znaků
[2] Výchozí nastavení pravidel - 29 925 znaků

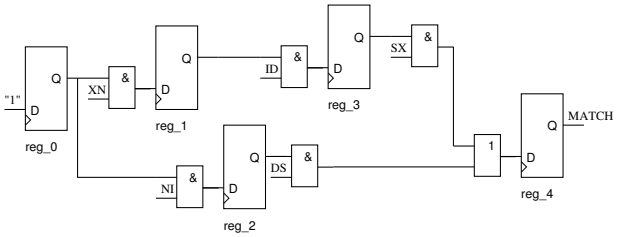
Při tvorbě transformačního programu, který převádí NFA do hardwarové reprezentace popsané v jazyce VHDL, byla snaha minimalizovat počet využitých HW zdrojů. Část optimalizací se provádí při tvorbě NFA, kdy jsou sdíleny prefixy řetězců a odstraněny duplicitní řetězce. Další optimalizací je sdílení logiky přechodu mezi stavy. Rovněž k implementaci sdíleného dekodéru, který převádí vstupní znak na kód 1 z 256, jsem využil BlockRam paměti, které jsou v designu jinak nevyužity. Uvedené optimalizace jsou velmi vhodné pro pravidla systému Snort. V tabulce 1 je uvedena závislost obsazení čipu na celkové propustnosti systému. Lepšího poměru znak/kapacita u řetězců z IDS systému Snort je dosaženo zejména díky uvedeným optimalizacím, které se v náhodných řetězcích neprojevily. Optimalizace řetězců systému Snort byly schopné zredukovat počet stavů NFA na polovinu.




Obrázek 2: NFA pro řetězec „NIDS“




Obrázek 3: Rozšířený NFA (16 bit)




Obrázek 4: HW. reprezentace ENFA

## 3.1 ZÁVĚR

Byla navržena architektura pro hledání řetezců na multigigabitových rychlostech. Architektura byla implementována v jazyce VHDL a výsledná implementace byla ověřena na desce COMBO6X a SFPRO [3]. Navržené řešení monitoruje jeden port na 1 Gb/s. Vyšších propustností lze dosáhnout, ale se zmenšující se množinou vyhledávaných řetězců. Jak pro poměrně velkou množinu náhodně vygenerovaných řetězců, tak i pro celou sadu pravidel Snort se podařilo dosáhnout propustnosti 6.4 Gb/s.

## REFERENCE

[1] Clark Ch., Schimmel D.: Scalable Pattern Matching for High-Speed Networks. In: *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, California, 2004, pp. 249-257.

[2] WWW stránka projektu Snort. http://www.snort.org (únor 2006)

[3] WWW stránka projektu Liberouter. http://www.liberouter.org (únor 2006)