INTRUSION DETECTION SYSTEM BEHAVIOR AS RESOURCE-ORIENTED FORMULA

Ján PERHÁČ, Daniel MIHÁLYI
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 042 00 Košice, E-mail: J.Perhac@tuke.sk, Daniel.Mihalyi@tuke.sk

ABSTRACT

Significant development of the information society in recent years creates increasing pressure on network security. One of the possibilities how to increase network security is the deployment of the Intrusion Detection System in computer network - IDS. This paper deals with formal description of the IDS behavior, through non-traditional resource-oriented logical system. For description of the IDS via logical systems, the Coalgebraic Modal Linear logic for IDS has been introduced, by which the behavioral effects of the IDS has been expressed by formula after simulated ARP spoofing attack in real laboratory environment.

Keywords: *ARP Spoofing, Coalgebraic Modal Linear Logic, IDS, Linear Logic, Snort, TCP Portscan*

1. INTRODUCTION

The logic as formalism is most widely used in computer science, but the expressing power of common used classical logics as a propositional logic based on the Tarski's semantic tradition dealing with true or untrue statement or intuitionistic logic that is based on Heyting's semantic tradition dealing with sense of formulæ is very limited. Interesting breakthrough in this area occurred in the year 1987 by the introduction of linear logic [2], which is a generalization and extension of the above traditional logics.

For exact description of the behavior of a program systems such as intrusion detection system in computer network, the traditional logics as formalisms are often not sufficient. Therefore we came with an idea to create a suitable formalism to describe behavior of the mentioned system which fulfills all demanding requirements. For that, we have created the Coalgebraic Modal Linear Logic for IDS (CMLL) as suitable logical system [5] i.e we introduce its syntax, semantics and proof system. Because the behavioral effects of IDS can be specified by proof, we present deduction rules in Gentzen's sequent calculus.

We illustrate our approach on the open source IDS Snort [8], by its reactions on simulated Address Resolution Protocol (ARP) spofing attack in real laboratory environment. Therefore we present brief introduction of the laboratory computer network topology and basic overview of the IDS Snort. Next we simulate a specific attack on client computer and show how can it be specified by formula of CMLL. Then we construct a proof of formula, from which the real process of algorithm of simulated attack can be seen.

2. COALGEBRAIC MODAL LINEAR LOGIC FOR IDS

By analyzing the current situation in the area of logical systems and consideration of existing solutions for the exact description of the behavior of a program system such as IDS, we have concluded that it is necessary to introduce the new logical system for that. Based on that we have created the Coalgebraic Modal Linear Logic for IDS, which is resulting in generalization of linear logic multiplicative fragment [3] and coalgebraic modal logic [6]. The formulation of this logic is already partially introduced in [5] and its definition came from prior research of the theory of programming research group, which works in our department of computers and informatics. In this paper we present CMLL's syntax and proof system.

Compared to the other logical systems, the significant feature of linear logic is mainly resource-oriented approach of dealing with formulæ [2], which creates a strong expressive power for describing real processes, for example causality, pleonasm or parallelism and many more [4]. These, together with modal operators of Coalgebraic modal logic, create an effective formalism for describing behavior of state-oriented program systems such as IDS.

2.1. Syntax of Coalgebraic Modal Linear Logic for IDS

We formulate syntax of CMLL by following production rule in Backus-Naur form:

$$\varphi ::= a_n \mid \mathbf{1} \mid \bot \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \mid \varphi^{\bot} \mid \Box\varphi \mid \Diamond\varphi \tag{1}$$

All formulæ *(actions)* of CMLL can be constructed by rule (1). The set of all CMLL formulæ can be named as *CMLLForm*. Where:

- a_n means elementary formulæ, where $n = \{1, 2, ...\}$,
- $\varphi^{\bot}$ is a linear negation, which expresses duality between action (φ) and reaction ($\varphi^{\bot}$), in the other words: available and consumed resource,
- $\varphi \multimap \psi$ is *(casual)* linear implication, which expresses that a (re)action ψ is a causal consequence of action φ [4] and after performing this implication, the resource φ became consumed ($\varphi^{\bot}$),
- $\varphi \otimes \psi$ with its neutral element $\mathbf{1}$ is multiplicative conjunction, which expresses the performing of both actions simultaneously,
- $\varphi \parr \psi$ with its neutral element $\bot$ is multiplicative disjunction, which expresses commutativity of duality between available and consumed resources by performing either action φ or action ψ,
- $\Diamond\varphi$ is modal operator expressing possibility of the action,

- $\Box\varphi$ is modal operator expressing necessity of the action,
- $\nabla\Phi \equiv \Box(\parr\Phi) \otimes (\bigotimes(\Diamond\Phi))$ is modal operator called *resource oriented coalgebraic modality* introduced in [5], and
  - $\Phi = \{\varphi_i \mid i = 1,2,\ldots,n\}$ is a finite set of formulæ,
  - $\Diamond\Phi = \{\Diamond\varphi \mid \varphi \in \Phi\}$,
  - operator $\parr\Phi$ means $\parr\Phi = \varphi_1 \parr \varphi_2 \parr \ldots$,
  - operator $\bigotimes$ is expressing possible infinite multiplicative conjunction of formulæ: $\bigotimes\Diamond\Phi = \Diamond\varphi_1 \otimes \Diamond\varphi_2 \otimes \ldots$.

In our approach we formulate the following De Morgan laws for CMLL:

$$\begin{aligned}
\mathbf{1}^{\perp} &\equiv \perp \\
\perp^{\perp} &\equiv \mathbf{1} \\
(\varphi^{\perp})^{\perp} &\equiv \varphi \\
(\varphi \otimes \psi)^{\perp} &\equiv \varphi^{\perp} \parr \psi^{\perp} \\
(\varphi \parr \psi)^{\perp} &\equiv \varphi^{\perp} \otimes \psi^{\perp} \\
\varphi \multimap \psi &\equiv \varphi^{\perp} \parr \psi \\
\Diamond\varphi &\equiv \neg\Box\neg\varphi \\
\Box\varphi &\equiv \neg\Diamond\neg\varphi
\end{aligned}$$

2.2. Proof system of Coalgebraic Modal Linear Logic for IDS

After analyzing options for a proof system that is suitable for our purposes, we have decided to define the proof system of CMLL in Gentzen's Double Side Sequent Calculus (GDSC), which, compared to Hilbert-style proof form, has only one axiom and many inference rules. Moreover, the creation of the proof is fundamentally simpler and, more important, the proofs in GDSC show real process which is described by formulæ. The inference rules for double side Gentzen's sequent calculus for CMLL have following form:

$$\underbrace{\Gamma}_{\varphi_1,\ldots,\varphi_n} \vdash \underbrace{\Delta}_{\psi_1,\ldots,\psi_m} \tag{2}$$

where Γ,Δ are finite sets of formulæ. Notation $\Gamma \vdash \Delta$ means

$$\varphi_1 \otimes \cdots \otimes \varphi_n \vdash \psi_1 \parr \ldots \parr \psi_m \tag{3}$$

which could be read as "the multiplicative disjunction of formulæ on the right side is provable from the multiplicative conjunction of formulæ on the left side of the sequent".

Defined inference rules are:

1. Identity rule is axiom i.e. is the only rule which has no assumptions. It expresses tautology: from action φ you can prove reaction φ.

$$\frac{}{\varphi \vdash \varphi}(id)$$

2. Structural rules are cut rule and exchange rules:

$$\frac{\Gamma \vdash \varphi \quad \Delta, \varphi \vdash \psi}{\Gamma, \Delta \vdash \psi}(cut)$$

Exchange rules express commutative property of logic by allowing permutation of formulæ on both sides of the sequent.

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}(ex_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \psi, \varphi, \Delta}(ex_r)$$

3. Logical rules deal with logical connectives:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}(\mathbf{1}_l) \qquad \frac{}{\vdash \mathbf{1}}(\mathbf{1}_r) \qquad \frac{}{\perp \vdash}(\perp_l) \qquad \frac{}{\Gamma \vdash \perp, \Delta}(\perp_r)$$

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \otimes \psi \vdash \Delta}(\otimes_l) \qquad \frac{\Gamma \vdash \varphi, \Delta \quad \Phi \vdash \psi, \Sigma}{\Gamma, \Phi \vdash \varphi \otimes \psi, \Delta, \Sigma}(\otimes_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \multimap \psi \vdash \Delta, \Sigma}(\multimap_l) \qquad \frac{\Gamma, \varphi \vdash \psi, \Delta}{\Gamma \vdash \varphi \multimap \psi, \Delta}(\multimap_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \parr \psi \vdash \Delta, \Sigma}(\parr_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \parr \psi, \Delta}(\parr_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}((\,)^{\perp}_l) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \varphi^{\perp}}((\,)^{\perp}_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \Box\varphi, \Delta}(\Box_r) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \Box\varphi \vdash \Delta}(\Box_l)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \Diamond\varphi, \Delta}(\Diamond_r) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \Diamond\varphi \vdash \Delta}(\Diamond_l)$$

According to [7], it is not necessary to introduce inference rules for logical connective $\Diamond$, due to the existence of De Morgan laws defined in chapter (2.1), but we have introduced the rules in the above form for logical connective $\Diamond$ which are based on already defined rules for logical connective $\Box$. Proof for the left side based rule:

$$\cfrac{\cfrac{\cfrac{\cfrac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}\,((\,)^{\perp}_{\mathrm{l}})}{\Gamma, \Box(\varphi^{\perp}) \vdash \Delta}\,(\Box_{\mathrm{l}})}{\Gamma \vdash \Box^{\perp}(\varphi^{\perp}), \Delta}\,((\,)^{\perp}_{\mathrm{r}})}{\Gamma \vdash \Diamond\varphi, \Delta}\,(\equiv)$$

And for the right side based rule:

$$\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \varphi^{\perp}, \Delta}\,((\,)^{\perp}_{\mathrm{r}})}{\Gamma \vdash \Box(\varphi^{\perp}), \Delta}\,(\Box_{\mathrm{r}})}{\Gamma, \Box^{\perp}(\varphi^{\perp}) \vdash \Delta}\,((\,)^{\perp}_{\mathrm{l}})}{\Gamma, \Diamond\varphi \vdash \Delta}\,(\equiv)$$

3. SNORT - THE INTRUSION DETECTION SYSTEM IN COMPUTER NETWORK

The intrusion detection system is a software application or a hardware device that monitors a computer network [9]. It serves as a protection and preparation against possible attacks and suspicious network activity. That is achieved by collecting information from various systems and network resources and their subsequent analysis for potential security threats.

Snort is an open source intrusion detection system in computer network, developed by Sourcefire, Inc.. According to the project's homepage [8], Snort is currently, with millions of downloads and nearly four hundred thousands of registered users, the worldwide most used IDS.

We illustrate our approach of coalgebraic modeling of IDS behavior by introduced Coalgebraic Modal Linear Logic for IDS in chapter (2). For that purpose, we have designed laboratory environment using real devices, where we have simulated an attack on a client computer with Snort installed.

3.1. Designed Laboratory Environment

For our purposes, the designed laboratory environment consists of two clients and one router, which serves as a gateway to the internet and also by which the computers are connected in local network. Due to demonstration of the functionality in real conditions we have used the real devices instead of the standard option of machine and network devices virtualization. Let a designed network has the topology shown in Fig. 1:

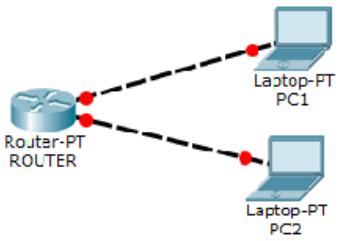

Fig. 1 Designed network topology

where:

- **ROUTER** has assigned IP address with subnet mask: `192.168.1.1/24`,
- **PC1** is a client and has assigned IP address with subnet mask: `192.168.1.102/24` and is playing the role of a victim,
- **PC2** is a client and has assigned IP address with subnet mask: `192.168.1.105/24` and is playing the role of an attacker.

3.2. ARP Spoofing Attack

ARP spoofing is a type of attack which is called "Man-in-the-Middle", where the attacker exploits ARP, so the attacker can pretend to be another computer on the local network [1]. The principle of the attack is based on deceiving the devices on local network into sending data, by fake ARP response to a request for MAC address of specific IP address. In this case, the attacker will always answer the ARP request with its MAC address, to which whole communication intended for another device will be sent. The principle of operation of this type of attack is shown in figure (2):

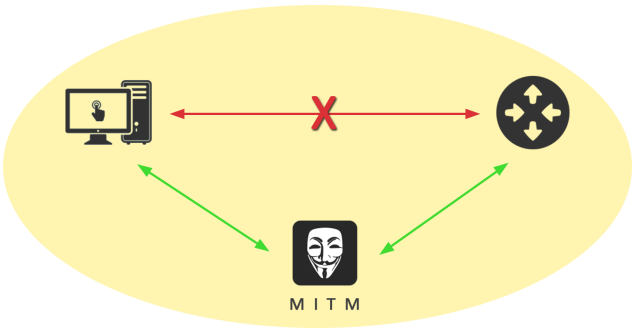

Fig. 2 Principle of the "Man-in-the-Middle" attack

The above mentioned attack can be realized by the following steps:

1. Analysis of the network topology, for example by using tool nmap:

```
root@attacker:~# nmap -F 192.168.1.1/24

Starting Nmap 6.47 ( http://nmap.org )
at 2015-04-15 22:44 UTC
Nmap scan report for 192.168.1.1
Host is up (0.0036s latency).
Not shown: 97 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
80/tcp open  http
MAC Address: XX:XX:XX:XX:XX:XX
(Tp-link Technologies CO.)

Nmap scan report for 192.168.1.102
Host is up (0.030s latency).
All 100 scanned ports on 192.168.1.102
are closed
MAC Address: YY:YY:YY:YY:YY:YY
(Liteon Technology)

Nmap scan report for 192.168.1.105
Host is up (0.000036s latency).
All 100 scanned ports on 192.168.1.105
are closed

Nmap done: 256 IP addresses
(4 hosts up) scanned in 4.10 seconds
```

Remark 3.1. *Since we are using real devices, we have censored the hardware MAC addresses of the network devices for security reasons.*

Snort will detect this step as a potential first phase of a possible attack, and creates an appropriate log:

```
[root@victim ~]# cat /var/log/snort
/portscan.log
Time: 04/13-18:01:39.536820
event_ref: 0
192.168.1.105 -> 192.168.1.102
(portscan) TCP Portscan
Priority Count: 9
Connection Count: 10
IP Count: 1
Scanner IP Range: 192.168.1.105:
192.168.1.105
Port/Proto Count: 10
Port/Proto Range: 25:8080
```

Based on the first step of the attack, we have chosen appropriate victim and in our case it is a computer with IP address and subnet mask:

```
192.168.1.102/24
```

2. Through ARP Spoofing attack we have transferred the communication between the router and victim computer via attacker computer by *arpspoof* tool. First we transfer communication from the victim to the router:

```
root@attacker:~# arpspoof -i wlan0
-t 192.168.1.102 192.168.1.1
YY:YY:YY:YY:YY:YY XX:XX:XX:XX:XX:XX
0806 42: arp reply 192.168.1.1 is-at
YY:YY:YY:YY:YY:YY
```

Finally we transfer communication from the router to the victim:

```
root@attacker:~# arpspoof -i wlan0
-t 192.168.1.1 192.168.1.102
XX:XX:XX:XX:XX:XX YY:YY:YY:YY:YY:YY
0806 42: arp reply 192.168.1.102 is-at
XX:XX:XX:XX:XX:XX
```

Snort recognizes this event as an attack and after trying to display web page `https://student.tuke.sk/` (which has IP address `147.232.3.210`) the log about intrusion will be created:

```
[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-23:59:47.579311
192.168.1.105 -> 192.168.1.102
ICMP TTL:64 TOS:0xC0 ID:57454
IpLen:20 DgmLen:522
Type:5  Code:1
REDIRECT HOST NEW GW: 192.168.1.1
** ORIGINAL DATAGRAM DUMP:
192.168.1.102:56031
-> 147.232.3.210:443
TCP TTL:63 TOS:0x0 ID:44869
IpLen:20 DgmLen:494 DF
Seq: 0xAD179471
(466 more bytes of original packet)
** END OF DUMP
```

3.3. Coalgebraic Modeling of the IDS Behavioral Effects by CMLL

We have introduced the CMLL of formal description of the IDS behavioral effects in chapter (2), by which the process of ARP Spoofing attack can be described as a formula of Coalgebraic Modal Linear Logic for IDS:

$$(((P \otimes Ia_1) \multimap \Diamond At) \otimes ((A_1 \otimes A_2) \otimes Ia_2)) \multimap \Box At \qquad (4)$$

which can be read as "vertical scanning of ports and action of IDS by creating a log implies a possible attack reaction and transferring of communication between the victim and the router via attacker by ARP spoofing implies that the attack necessarily happened".

We are using the following notations for a transparency of formula and proof:

- vertical scanning of ports as P,
- IDS reaction on vertical scanning of ports by creating a log about potential attack as Ia_1,
- transferring communication via attacker from victim to ROUTER as A_1,
- transferring communication via attacker from ROUTER to victim as A_2,
- IDS reaction on transferring communication by creating a appropriate log about potential attack as Ia_2,
- attack as At.

The proof of the formula is constructed by inference rules introduced in chapter (2.2) in form of Gentzen's sequent calculus. Contexts in proof contain:

- $\Gamma = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp}), A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}, \Diamond(At^{\perp})\}$,
- $\Sigma = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp}), A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}\}$,
- $\Delta = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp})\}$,
- $\Psi = \{A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}\}$.

The proof tree is depicted in the figure (3). It is constructed from the root (in the bottom of the tree) where the behavioral formula of the attack which we are proving is, up to the leafs until they will become the individual identities. We are using an appropriate inference rule from chapter (2.2) in every step of deduction. Leafs are axioms, which implies that Gentzen's style proof is constructed correctly, therefore we can say that formula in the root is proved. We can observe behavior of the IDS during specific attacks from the proof tree depicted in figure (3).

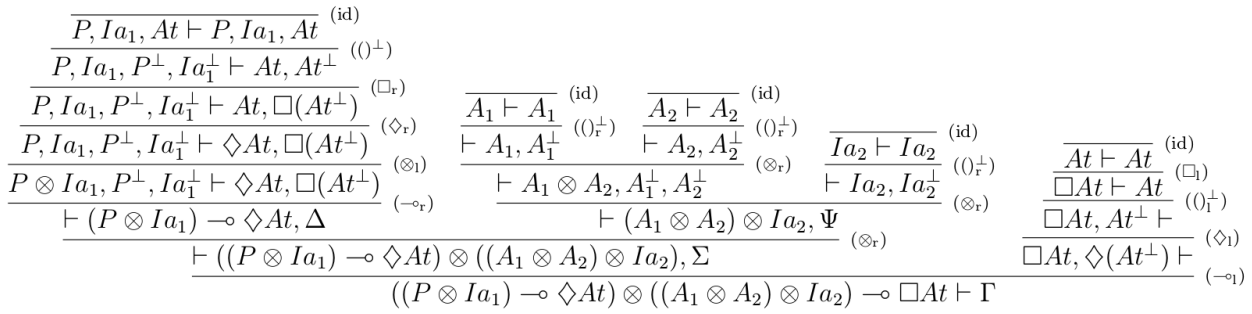

Fig. 3 Proof tree

4. CONCLUSION

In this contribution, we show how the resource-oriented logical system can be used as a formalism for describing real processes of a behavior of non-trivial program system. For that, we have chosen intrusion detection system in computer network - Snort. We have expressed its behavior by a formula of the newly introduced logical system and illustrated how its proof describes real process of IDS behavior.

Another approach of a formal description of program systems are for example Petri nets, where one can see many similarities like modeling cause-effect situations or nondeterministic choices. In our resource oriented approach we can manipulate with logical time and space, too. It means, that in the future we would like to extend our approach by expanding Coalgebraic Modal Linear Logic for IDS with time-spatial calculus from Girard's Ludics theory.

REFERENCES

[1] GIBSON, S.: ARP Cache Poisoning, Gibson Research Corporation, Laguna Hills, CA, USA, 2005 `https://www.grc.com/nat/arp.htm`

[2] GIRARD, J.-Y.: Linear logic. Theoretical Computer Science 50, Elsevier Science Publishers Ltd. Essex, UK, 1987

[3] GIRARD, J.-Y.: Linear Logic: its syntax and semantics, Cambridge University Press, Laboratoire de Mathmatiques Discrètes, Marseille, 1995

[4] MIHÁLYI, – D., NOVITZKÁ, V.: Princípy duality medzi konštruovaním a správaním programov, Equilibria, Košice, 2010

[5] MIHÁLYI, – D., NOVITZKÁ, V.: Towards to the Knowledge in Coalgebraic model IDS, Computing and Informatics, 33, 1, pp. 61-78, 2014, ISSN 1335-9150

[6] MOSS, L.: Coalgebraic logic, Annals of Pure and Applied Logic, Volume 99, Issues 13, Department of Mathematics, Indiana University, Bloomington, pp. 241-259, USA, 1997

[7] ONO, H.: Proof-theoretic methods in nonclassical logic, An Introduction, Theories of Types and Proofs, Mathematical Society of Japan Memoirs 2, pp. 207-254, 1998

[8] ROESCH, M.: Project Snort home page, The Snort Team, 2015 `https://www.snort.org/`

[9] ROZENBLUM, D.: Understanding Intrusion Detection Systems, SANS Institute InfoSec Reading Room, 2001

Received September 9, 2015, accepted November 6, 2015

BIOGRAPHIES

Ján Perháč was born in 1991 in Svidník, Czechoslovakia. In 2015 he graduated (MSc) at the department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice. He defended his master's thesis in the field of Informatics. Currently, he is a PhD student at the same department. He is concerned with GNU/Linux operating systems, network security, non-traditional logical systems and category theory.

Daniel Mihályi has worked as a researcher at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice and later as Assistant Professor. In 2009 he defended PhD. thesis "Duality Between Formal Description of Program Construction and Program Behaviour". The main area of his research includes applications of category theory in informatics and using source-based logical systems for a formal description of the program systems behavior.