International Journal of Security and Its Applications
Vol.8, No.3 (2014), pp. 107-118
http://dx.doi.org/10.14257/ijsia.2014.8.3.12

A Cooperative Intrusion Detection Model for Cloud Computing Networks

Shaohua Teng[1], Chaoyu Zheng[1], Haibin Zhu[2], Dongning Liu[3] and Wei Zhang[3]

[1]*School of Computer Science and Technology*
Guangdong University of Technology, Guangzhou, China
[2]*Collaborative Systems Laboratory, Nipissing University, North Bay, Canada*
[3]*School of Computer Science and Technology*
Guangdong University of Technology, Guangzhou, China
[1]*shteng@gdut.edu.cn, 149nuanye@163.com,* [2]*haibinz@nipissingu.ca,*
[3]*liudn@gdut.edu.cn, weizhang@gdut.edu.cn*

Abstract

While cloud computing provides a convenient and efficient network environment for users to obtain powerful computing resources, it also brings some important security issues about data security and reliable services. One of the major security issues is to deal with malicious attacks. To cope with these attacks in this paper, a collaborative intrusion detection architecture is proposed and the E-CARGO model is used to model this system. According to CIDF (Common Intrusion Detection Frame), the components of the intrusion detection system are defined. Furthermore, we design and clearly describe the behaviors of Agent and their interrelationship. At last, experiments are used to verify our method's effectiveness.

Keywords: *cloud computing; intrusion detection; E-CARGO; collaborative; architecture*

1. Introduction

Cloud computing technology has become one of the most popular topics, and its development has received wide concern [1, 2]. However, the powerful computing resources and huge storage capacities of the Cloud computing environments have great temptation for the intruders, and they can easily become attractive targets. In order to cope with these potential attackers, a cooperative intrusion detection system (IDS) is a viable and effective method [3, 4].

In recent years, attacks have shown increasing sophistication which involves different source hosts and different networks; it is extremely difficult to detect these coordinated attacks since the evidence of the attacks is spread across multiple cloud computing regions [5]. A collaborative intrusion detection system (CIDS) that simultaneously combines the evidences from multiple networks can be formed to detect these attacks [6].

Purdue University proposed an architecture called AAFID (Autonomous Agent for Intrusion Detection), and they firstly used the autonomous agents to build an Intrusion detection system; this system includes four levels of components: agents, filters, transceivers and monitors; agents are used as lower-level element to do data collection and analysis [7].

Chatzigiannakis proposed a Distributed Intrusion Detection model by using Security Agents [8]. It includes three kinds of agents: misuse detection agent, anomaly detection agent and Simple Network Management Protocol (SNMP) query agent; a misuse detection agent is responsible for detecting data from the network by using signature based detection method;

ISSN: 1738-9976 IJSIA
Copyright ⓒ 2014 SERSC

an anomaly detection agent is mainly used to detect denial of service attacks; SNMP query agent queries at the routers of the network and provides the results to the Central IDS Node.

Intrusion detection modeling can be taken as a software engineering problem, and E-CARGO model for Role-Based Collaboration is a promising approach to analyze collaborative systems [9]. Therefore, this model can be used to describe the architecture of the detection model, the components and the relationships among the components.

In E-CARGO, a role (r) is assigned to current and potential intrusion detection agents [10], where the current ones are currently playing a detection role and the potential ones possess the ability to detect but are not currently playing that role; a group (g) is built on a detecting intrusion environment (e); e confines a range limit [l, u] for a role; the role needs the minimum (l) detection agents to play it and can be played by the maximum (u) detection ones.

The rest of this paper is organized as follows. In Section 2, we first describe various coordinated attacks that are common in cloud computing environment. In Section 3, a collaborative intrusion detection architecture is proposed and the components of the architecture are described based on E-CARGO model. In Section 4, we describe the behaviors of agents and their interrelationships. The experiments using our model are presented in Section 5. Finally, the paper concludes the major results and discusses some further ideas of our improvements.

2. Intrusions to Cloud Systems

Availability and security of Cloud resources and services are affected by several common intrusions. In this section we will describe these common attacks.

2.1. Coordinated Scanning Attack

Scanning attacks are used to gain the information of the target system, possible versions of the software or the operation system information. For example, an attacker can discovers an opened port upon which services are provided in the Cloud environment.

There are two typical methods of scanning attacks: horizontal scan and block scan [11]. The horizontal scan is used to find an opened service port by scanning a certain range IP addresses. The block one is used to scan a group of services on a range of hosts within a specified area, for identifying the number of services. These attacks can be coordinately accomplished by a great number of computers simultaneously. So without collaboration between detectors, a sing detector is very difficult to detect these attacks

2.2. DDoS Attacks

DDoS (Distributed Denial of Service) is a type of badly security attack, which can disrupt the online service of the cloud computing by trying to make the resources of the target hosts unavailable to their expectative users [12].

The DDoS attack can be roughly split into two stages: recruiting and launching an attack. During the first stage, it controls a set of victim machines on the Internet, and then installs attack tools on these machines. In the second stage, the attacker releases attack commands to puppet machines and launches attacks to the targets.

2.3. User to Root Attacks

U2R (User-to-Root) attacks is a type of attack by illegally elevating user's privileges. An attacker obtains the ordinary user information by sniffing password, and this can make him be

 Copyright ⓒ 2014 SERSC

able to exploit vulnerabilities, for example, obtaining super users access to system. In case of Cloud, It can enable an attacker to acquire root level access to host by obtaining access to legal user's instances.

2.4 Attack Analysis

By analyzing the source and the target host, these attacks can be divided into four kinds of situations, and the detection model can be constituted according to these situations [13].

(1) O2O (one - one): A source host launches mass data packets to a target host, and make the target host can't provide normal services.

(2) O2M (one - multiple): A source host launches mass data packets to multiple destination hosts, and make the hosts can't work normally.

(3) M2O (multiple - one): A source host controls a set of puppet machines, and the puppet machines coordinately attack a destination host.

(4) M2M (multiple - multiple): A source host controls a set of puppet machines, and the puppet machines coordinately attack multiple destination hosts.

3. Collaborative Intrusion Detection Model based on E-CARGO

3.1. The Collaborative Intrusion Detection Architecture

To cope with these kinds of attacks, a collaborative Intrusion detection system (IDS) is a practical solution. The Defense Advanced Research Projects Agency (DARPA) proposed a Common Intrusion Detection Frame (CIDF). In this frame the intrusion detection system is divided into four components: event generator, event detection, response unit and the event database [14]. According to this frame, we propose a framework of collaborative intrusion detection model, and this model is composed of event generators, feature detector, statistical detector, fusion center, and response unit, as shown in Figure 1.

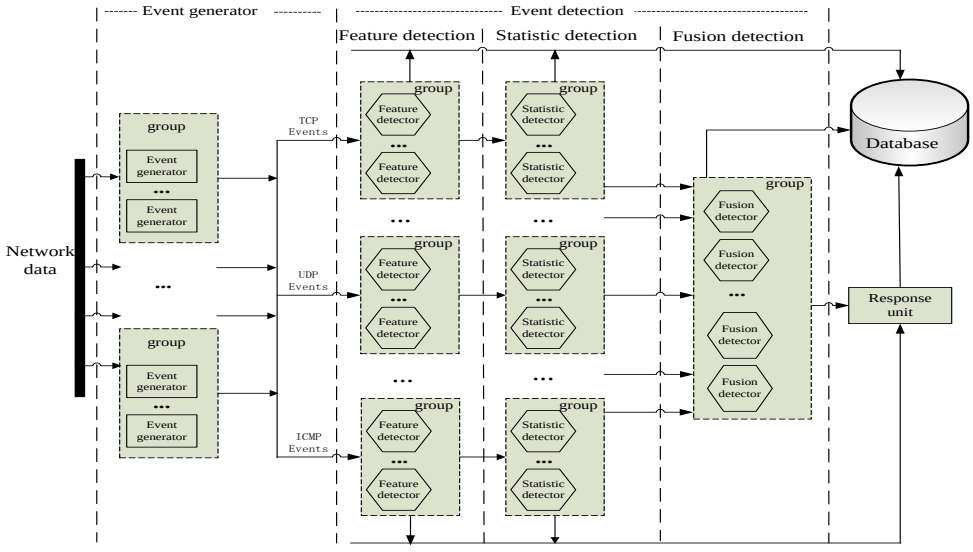


Figure 1. Collaborative Intrusion Detection Architecture

3.2. System Modeling with E-CARGO

Following the E-CARGO model [15], these components are all taken as requirements, which can be defined as roles. Each role r is assigned to agents, where these agents play a task (a role).

***Definition 1**: **event generator**.* $r_1 ::= <n, I, N_a, N_o, e_t, e_s, \psi>$ where,

- n is the identification of event generator;
- $I ::= < M_{in}, M_{out} >$ denotes a set of messages, wherein, M_{in} expresses the incoming messages to event generator, M_{in}= {network data}. M_{out} expresses a set of outgoing messages to event detector, M_{out} ={TCP events, UDP events, ICMP events};
- N_a is a set of identifications of agents that are playing event generator;
- N_o is a set of identifications of objects including network data, event generator, event detector and database that can be accessed by the agents playing event generator;
- e_t and e_s are used to respectively express how many units of free time and how many units of space event generator required;
- ψ is the required credits for an agent to play event generator.

The event generators collect data from the networks, and generate suspicious intrusion events. They submit the suspicious intrusion events to the feature and statistical detection agents. According to the network protocol, these suspicious intrusion events are divided into TCP events, UDP events and ICMP events [16].

***Definition 2**: **event detector**.* $r_2 ::= <n, I, N_a, N_o, e_t, e_s, \psi>$ where,

- n is the identification of event detector (it may be feature detection and statistical detection or the fusion center);
- $I ::= < M_{in}, M_{out} >$ denotes a set of messages, wherein, M_{in} expresses the incoming messages to event detector, M_{in}= {TCP events, UDP events, ICMP events}. M_{out} expresses a set of outgoing messages to response unit or database, M_{out} ={normal user behavior, attack behavior, suspicious intrusion events}*;*
- N_a is a set of identifications of agents that detect the suspicious intrusion events;
- N_o is a set of identifications of objects including suspicious intrusion events, event detector, event generator, response unit and database that can be accessed by the agents playing event generator;
- e_t and e_s are used to respectively express how many units of free time and how many units of space event detector required.
- ψ is the required credits for an agent to play event detector.

***Definition 3**: **response unit**.* $r_3 ::= <n, I, N_a, N_o, e_t, e_s, \psi>$ where,

- n is the identification of response unit;
- $I ::= < M_{in}, M_{out} >$ denotes a set of messages, wherein, M_{in} expresses the incoming messages to response unit, M_{in}= {attack behavior}. M_{out} expresses a set of outgoing messages, M_{out} = { reporting, intrusion prevention, storing to database }*;*
- N_a is a set of identifications of agents that play response unit;
- N_o is a set of identifications of objects including attack behaviors, event detector, response unit and database that can be accessed by the agents playing event generator;
- e_t and e_s are used to respectively express how many units of free time and how many units of space response unit required;
- ψ is the required credits for an agent to play response unit.

***Definition 4**: **agent**.* $a ::= <n, c_a, s, r_c, R_p, N_g, e_t, e_s, \psi, u >$, where

- n is the identification of the agent which can play event generator, event detector or

 Copyright ⓒ 2014 SERSC

response unit;

- c_a is a special class that describes the common properties of agents;
- s is a data structure whose values are called attributes, properties, or states;
- r_c means a role that agent is currently playing. If it is empty, then this agent is free;
- R_p means a set of roles that the agent has potential ability to play ($r_c \notin a.R_p$);
- N_g is the identification of group that the agent belongs to;
- $<e_t, e_s>$ expresses the processing capacity for an agent, where e_t expresses how many units of free time it has and e_s expresses how much memory space it has. Based on the performance of an agent's situations, $<e_t, e_s>$ can be reset;
- ψ expresses the past performance or credits of serving others;
- u to expresses the workload of the agent.

An agent a can be an event generator, a detector or a response unit. A denotes the set of all agents.

Definition 5*: *environment. $e ::= < n, R_e, \circledS, B>$ *where*

- n is the identification of the detecting intrusion environment;
- R_e is a finite set of roles;
- $\circledS$ is the shared object for R_e;
- B is a finite set of tuples consisting of roles and their ranges, i.e., < r, q >, where $r \in R_e$. The role range (also called cardinalities) q is expressed by <l, u> and tells how many agents must (l) and may (u) play r in this environment.

***Definition 6*: *group*.** $g:: = <n, e, J>$ where

- n is the identification of the group;
- e is a detecting intrusion environment for the group to work;
- J is a set of 3-tuple of identifications of an agent and role, i.e., $J = \{<n_a, n_r, n_o> | \exists q, n_o (n_o \in N_o) \wedge (< n_r, q, N_o > \in e.B)\}$.

Agents that perform the same tasks are formed into a group g. The group synthesizes the results detected by the agents of the same group. The group synthesizes the results of the agents which perform the same tasks.

4. Collaborative Intrusion Detection Roles and their Relationships

In order to gain high speed of detection and reduce the size of data to be analyzed, a four layer detection model is proposed, and they are respectively performing the roles of generating events, feature detection, statistical detection and fusion analysis.

4.1. Event Generator

As Definition 1 states, an event generator collects the network packets, and generates suspicious intrusion events by filtering.

The network card should be set to the promiscuous mode for purpose of collecting every data packets through the network card. A filter is used to filter data according to some filtering approaches from the lower frame (MAC) to the top object (transport layer port) step by step, which is shown in Figure 2.

In order to ensure the intrusions promptly and effectively been detected, reducing the amount of data for senior agents is an effective way. Therefore, MAC address eliminates the data sent by an illegal MAC address, and filters out the invalid frames. Then the network layer only receives the remaining frames.

Copyright ⓒ 2014 SERSC

The IP address filtering module firstly eliminates the network packets if they meet the some given conditions, for example, the source address of the datagram is the same as the target address; the source address of the datagram is a hacker address or in the blacklist; the source address of the datagram is trust network address; or the source address of the datagram is local address. Then the remaining data will be processed according to the protocol of the network layer. If it is ICMP protocol, it will generate ICMP event which will be send to the corresponding detection agent. Otherwise it will be send to the upper module.

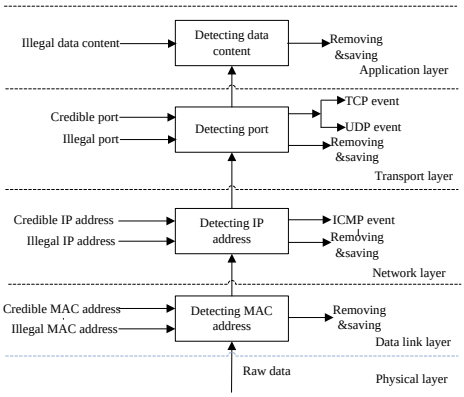


Figure 2. The Event Generator

The data that conform to the following situations will be eliminated by the port filtering module, and the rest of the data will generate TCP events or UDP events according to the protocol type. (1)The fragmentation of two IP packet are overlapped; (2)The IP address and port number of source hosts are respectively equal to the target ones'; (3)Some distinct illegal packets, such as the packet whose destination port address has not opened yet.

4.2. Feature Detector

An agent playing the feature detector role receives events from agents which play the role of event generator. The events can be divided into the following three types according to network protocol: TCP events, UDP events and ICMP events, and they are respectively detected by different feature detection agents. In order to explain the behaviors of feature detector in collaborative intrusion detection, we take TCP events detection as an example, and the other evens detection are similar.

According to TCP/IP, a three-way handshake protocol, we can conclude that the TCP scanning attacks mostly adopt deformity connection, and their flags of the packages are distinct from the normal ones'. Therefore, it is very easy to recognize these types of TCP scanning attacks by comparing abnormal flags with the known scanning attacks. If the flags of a data packet match up with the already known scanning attacks', it indicates that the target system is under attack by this type of scanning attack. So we can adopt the following rules to detect TCP scanning attacks which are similar to Snort [17] rules.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN ACK"; flags: A;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; flags: S F;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS"; flags: F U;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN MAIMON"; flags: A F;)
```

 Copyright ⓒ 2014 SERSC

4.3. Statistical Detector

An agent playing the role of statistical detector accepts the messages from agent playing the feature detector role. The principle of statistical detection is adopted as follows: it is considered to an attack event if the number of obtained packets is greater than a predefined threshold in a given time span. It is classified as Trojan horse attack if internal process of system sends messages to external process, otherwise it is denial of service attack or scanning attack.

In order to perform statistical detection, we should design a base table to store the header information of network packet, and the structure of the table is given in Table 1.

Table 1. The Structure of the Base Table

Field	Type	Description
SAT_Time	time	time
Src_IP	char	Source IP address
Src_Port	char	Source port number
Dst_IP	char	Destination IP address
Dst_Port	char	Destination port number
TTL	int	Time to live
Size	int	Length
Protocol	char	Internet protocol

Here, we take 02M (one - multiple) situation for example and the others are similar. The number of the network date which have the same target host is counted in a given time interval. If this value exceeds a predefined threshold, the suspicious attack event can be detected. Let Pre_1, st_time, end_time respectively represent the preset threshold, the start time of detection, the finish time of detection. The SQL query is as follows:

```
Select Src_IP
From Statistical_Table
Where SAT _Time >= st_time and SAT _time <= end_time
Group by Src_IP
Having count (*) >= Pre_1
```

The above roles regulate agents to detect all kinds of scanning attacks whose target host or port have regular changes. However, for the slow scanning attacks which are carefully designed by multiple hosts, it is very difficult to detect. In order to resolve this situation, the suspicious intrusion events should be reported to fusion detector to do further detection.

4.4 Fusion Analysis

The fusion center is constituted of a group of agents playing different roles, which respectively perform data preprocessing, fusion on time, fusion on space, fusion on content and general analysis. Agents and their relationships are depicted in Figure 3.

Copyright ⓒ 2014 SERSC

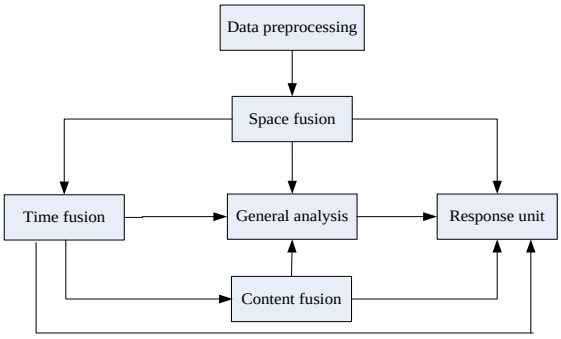


Figure 3. The Structure of Fusion Center

Data preprocessing: it accepts messages from statistical detector, and these data are divided into different groups according to their space information, for example the type of protocol, source IP address, destination IP address, and etc. Then it sorts the data in an ascending order. Finally, it generates the suspicious events and reports the messages to relevant detection agents.

Space-time fusion: It firstly eliminates redundant suspicious intrusion events which are detected by different detection agents at the same time. Then it does statistical and correlation analysis. For example, the process of an attack can be divided into several steps, the previous stage make preparations for the later stage, thus there is an intrinsic link between these stages.

Content fusion: The suspicious intrusion events will be implemented correlation and association analysis in this module. For example, when an attacker launches scanning attacks to multiple networks, the number of suspicious intrusion events which are detected in a single network is lower than a predefined threshold. But by this process it can detect this kind of scanning attacks.

5. Experiments

Experiment 1: This experiment is used for detecting slow scanning attacks on two network regions. Attackers randomizing scan different networks to make UDP echo request of the same network, and adopt the abnormal target addresses, so the attack can evade a traditional scanning detector.

10:13:27.163867 192.168.7.6.3066 > 192.168.134.53.echo: udp 6
10:18:10.581092 132.17.27.24.3066 > 10.16.72.1.echo: udp 6
10:20:30.179362 212.68.38.52.3066 > 10.16.163.152.echo: udp 6
10:31:20.560266 92.134.74.29.3066 > 192.168.61.15.echo: udp 6
10:38:31.791267 152.56.29.34.3066 > 10.16.33.41.echo: udp 6
10:50:13.320882 136.48.46.25.3066 > 192.168.3.122.echo: udp 6
10:58:19.202492 166.81.74.49.3066 > 10.16.47.13.echo: udp 6

Experiment 2: Cooperative analysis is used to detect DDos on three network regions. One region launches SYN flood to the other regions. It can detect SYN flood attack but it can’t identify the attacker's IP address because he uses a fake IP address. From the monitor, we can see that network traffic significantly increases during attacking.

 Copyright ⓒ 2014 SERSC

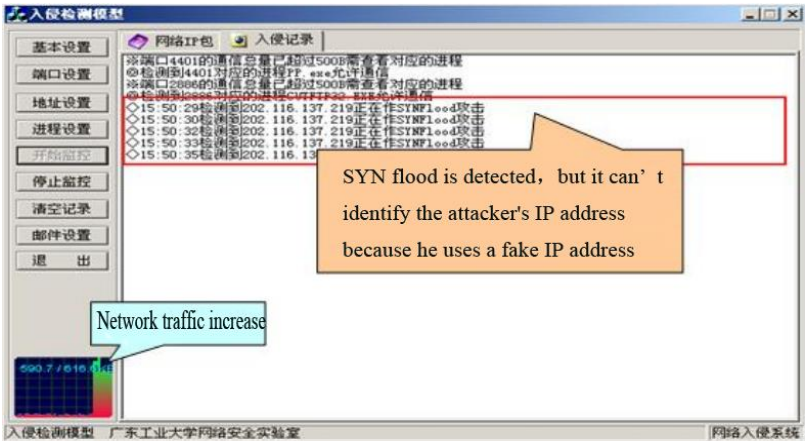


Figure 4. The Detection Result

6. Conclusion and Future Work

The security of clouding computing network is very important to investigate. Several common coordinated intrusions of the cloud computing network are analyzed. According to CIDF, we propose the cooperative intrusion detection architecture, and introduce the E-CARGO model to describe this architecture. The results of experiments show that our proposed method can detect slow scanning attacks and DDos which verify the validity of our model. As future work, we will study how to combine cooperative computing with intrusion detection to deal with the real world problems.

Acknowledgment

This work was supported by Guangdong Provincial Natural Science Foundation (Grant No. 9151009001000007, 10451009001004804, S2012010010570), Guangdong Provincial Science & Technology Project (Grant No. 2012B091000173), Guangzhou City Science & Technology Project (Grant No. 2012J5100054, 2013J4500028), and Key Laboratory of the Ministry of Education project (Grant No. 110411).

References

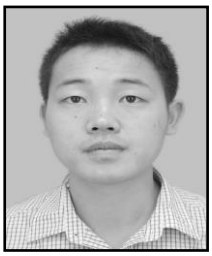
[1] Subashini, S., and V. Kavitha, A survey on security issues in service delivery models of cloud computing [J], Journal of Network and Computer Applications, vol. 34, no. 1, **(2011)**, pp. 1-11.
[2] CHEN Dan-wei, HOU Nan, SUN Guo-zi, Novel Cloud Computing Intrusion Detection Model Based on Improved Manifold Learning [J], Computer Science, vol. 37, no. 10, **(2010)**, pp. 59-62.
[3] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M, A view of cloud computing [J], Communications of the ACM, vol. 53, no. 4, **(2010)**, pp. 50-58.
[4] LO, Chi-Chun, HUANG, Chun-Chieh; KU, Joy, A cooperative intrusion detection system framework for cloud computing networks [C], In: Parallel Processing Workshops (ICPPW), 2010 39th International Conference on. IEEE, **(2010)**, pp. 280-284.
[5] D. G. Feng, M. Zhang, Y. Zhang and Z. Xu, “Study on cloud computing security”, Journal of Software, vol. 22, no. 1, **(2011)**, pp. 71-83.
[1] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, “A survey of intrusion detection techniques in cloud”, Journal of Network and Computer Applications, vol. 36, no. 1, **(2013)**, pp. 42-57.
[2] E. H. Spafford and D. Zamboni, “Intrusion detection using autonomous agents”, Computer networks, vol. 34, no. 4, **(2000)**, pp. 547-570.
[3] V. Chatzigiannakis, G. Androulidakis and B. Maglaris, “A distributed intrusion detection prototype using security agents”, HP Open View University Association, **(2004)**.

Copyright ⓒ 2014 SERSC

[4] H. Zhu and M. Zhou, "Efficient Role Transfer Based on Kuhn-Munkres Algorithm", IEEE Transactions on Systems, Man, and Cybernetics [J], Part A: Systems and Humans, vol. 42, no. 2, **(2012)**, pp. 491-496.
[5] H. Zhu and M. Zhou, "M-M Role-Transfer Problems and Their Solutions", IEEE Trans. on Systems, Man and Cybernetics [J], Part A: Systems and Humans, vol. 39, no. 2, **(2009)**, pp. 448-459.
[6] C. V. Zhou, C. Leckie and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection", Computers & Security, vol. 29, no. 1, **(2010)**, pp. 124-140.
[7] S. H. Teng, "A Study on Object-Monitoring-based Distributed and Collaborative Intrusion Detection", The degree of Doctor of Philosophy, Guangdong University of Technology, China, **(2008)**.
[8] D. Dash, B. Kveton, J. M. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach and A. Newman, "When gossip is good: Distributed probabilistic inference for detection of slow network intrusions", Proceedings of the 21th national conference on artificial intelligence (AAAI), vol. 21, no. 2, **(2006)**, pp. 1115-1122.
[9] S. Staniford-Chen, B. Tung and D. Schnackenberg, "The common intrusion detection framework (CIDF)", Proceedings of the information survivability workshop, **(1998)** October.
[10] H. Zhu and M. C. Zhou, "Role-based collaboration and its kernel mechanisms, Systems, Man, and Cybernetics, Part C: Applications and Reviews", IEEE Transactions, vol. 36, no. 4, **(2006),** pp. 578-589.
[11] H. S. H. Teng, N. Wu, W. Zhang and J. Su, "A Cooperative Network Intrusion Detection Based on Fuzzy SVMs", Journal of Networks, vol. 5, no. 4, **(2010)**, pp. 475-483.
[12] W. Zhang, S. H. Teng and X. F. Fu, "Cooperative network intrusion detection based on data fusion", Journal of computer applications, vol. 29, no. 1, **(2009)**, pp. 284-290.

Authors

Shaohua Teng is a Professor of Guangdong University of Technology in China. He was born on Jan in 1962. He is responsible for teaching data mining in School of Computer Science and Technology. He is engaged in education and technology transfer on knowledge discovery issues, and his research interests include network security, cooperative computing and data mining. Dr. Teng earned a Ph.D. in Industry Engineering at Guangdong University of Technology. He has published 150 papers on computer magazines and international conferences and 2 books.



Chaoyu Zheng received his BS in Luoyang Normal University in 2011, China. He is currently pursuing his MS in computer science and technology at the School of Computer Science and Technology, in Guangdong University of Technology, China. His research interest is cloud computing security.

Haibin Zhu is Full Professor of the department of Computer Science and Mathematics, Nipissing University, Canada. He has published five books, one book chapter and over 130 research papers. He is a senior member of IEEE and a member of ACM. He is serving and served as co-chair of the technical committee of Distributed Intelligent Systems of IEEE SMC Society, editorial board member for journals, organization chairs and PC members for many conferences. His research interests include adaptive collaboration, role-based collaboration and distributed intelligent systems.

 Copyright ⓒ 2014 SERSC

Dongning Liu was born on Jan in 1979, and received his PhD in Sun Yat-Sen University. He is an associate Professor of Guangdong University of Technology in China. His research interests include artificial intelligent, procession of temporal information.

Wei Zhang is an associate Professor of Guangdong University of Technology in China. She was responsible for teaching data mining in School of Computer Science and Technology. She is engaged in network security, cooperative computing and data mining. Mrs. Zhang earned a M.S. in Software Engineering from the South China University of Technology. She has published 100 papers on computer magazines and international conferences.

Copyright ⓒ 2014 SERSC

Copyright ⓒ 2014 SERSC