Detection of Employees' Carelessness Leading to Cause Insider Threats through Analyzing Email RDF Data

Sung-Min Kim, Young-Seok Son, and Young-Guk Ha*
Konkuk University, Seoul, South Korea
{allmax100, upsys}@naver.com, ygha@konkuk.ac.kr

Abstract

In recent years, many businesses have been put in a great deal of effort to develop ways to fight against threats posed by insiders of them. Insider threats can be perpetrated for the purpose of doing harm to a business, but in some cases, they are done by mistake. And those mistakes also can be made by carelessness of employees who do not have any malicious purpose to harm their company. In this paper, we suggest the need to watch those employees' actions that might be a clue about their carelessness. Email messages are one of the useful resources to watch because it is the media employees use on a daily basis at work and each message can reflect the writer's feeling or what they are doing at work by words. In this paper, with an experiment, we suggest closely waching some of email behaviors can tell us about clues to which employee is carelessness now and has the potential danger to make mistakes that might lead to a serious damage to the business.

Keywords: Insider Threat Detection, Email, Carelessness, Rdf

1 Introduction

In recent years, many businesses and public departments have put in a great deal of effort to develop ways to fight against threats posed by insiders of them.[13][21][14] Thoes threats, called Insider Threats"[15], involve fraud, the theft of confidential or commercially valueable information, the theft of intellectual property, or the sabotage of computer systems.[23] Such insider threats can be perpetrated for the purpose of doing harm to a business, but in some cases, they are done by an employee's mistake. Workers' mistakes, if not that serious, can end up with only their boss scolding them for their mistake. In some cases, however, their mistakes can lead to serious accidents causing a catastrophic damage to the business. According a recent survey, e-mail breach and database breach starting with insider mistakes have become common news.[10] Employees' mistakes at work usually come from their carelessness-we'll see a few examples in Section 2. In many situations, employees can get careless when they are so busy with too much work and tired or get deeply influenced by their own personal matters, falling to focus on what they are doing or to expect the consequences of their acts. As the situation at work changes, employees' mental states also vary, so any employee can be careless at any moment whatever position he/she is in. If an employer get to know which employees can be careless now and has the potential to cause an accident by his/her carelessness, then the employer can take measures to avoid the potential danger by giving them a warning about their carelessness or other wise methods. One of the possible ways to figure out whether employees are careless or not is to track their acts on computer, which can reflect their mental condition. In the research we did for this paper, we decided that email messages exchanged among employees are the best resource to watch because email is the media that employees always use at work and can contain a wealth of information about their work or mental condition. Some of email behaviors

Research Briefs on Informaiton & Communication Technology Evolution (ReBICTE), Vol. 1, Article No. 8 (January 15, 2015)
*Corresponding author: 1007 Secheonnyeonguan, 1 Hwayang dong, GuangGin gu, Seoul, South Korea, Tel: +82-(0)2-450-3273

of employees can be a useful clue about their carelessness. One of email bahaviors that we decided can be a useful clue is not sending back a reply to an email message whose sender wants to get an reply. The reason a receiver does not reply to an email message might be that he or she forgets to remember to reply the email message, or fails to recognize the need to reply to it , which we suggest can reflect his or her carelessness. For the experiment described in Section 4, we used email data from users who worked for Enron-the details of the data are covered in Chpater 3. And we adopted RDF (Resource Descrption Framework) language[9] for the format of the intermediate experimental data file-we'll cover the data format in Section 4.2- because RDF has a suitable for distributed processing, which opens the possibility of allowing computing systems have scalability for big data computing. The result of the experiment showed us some meaningful figures described in Section 4.5.

2 Unintentional Insider Threats

There are a lot of insider threat cases which happened due to someone's mistake. We call such insider threats Unintentional Insider Threats(UIT)[20]. Here, we introduce some of those cases caused unintentionally. As Figure 1 shows, unintentional causes account for large part of the whole causes of data breaches, one of the typical types of insider threats.

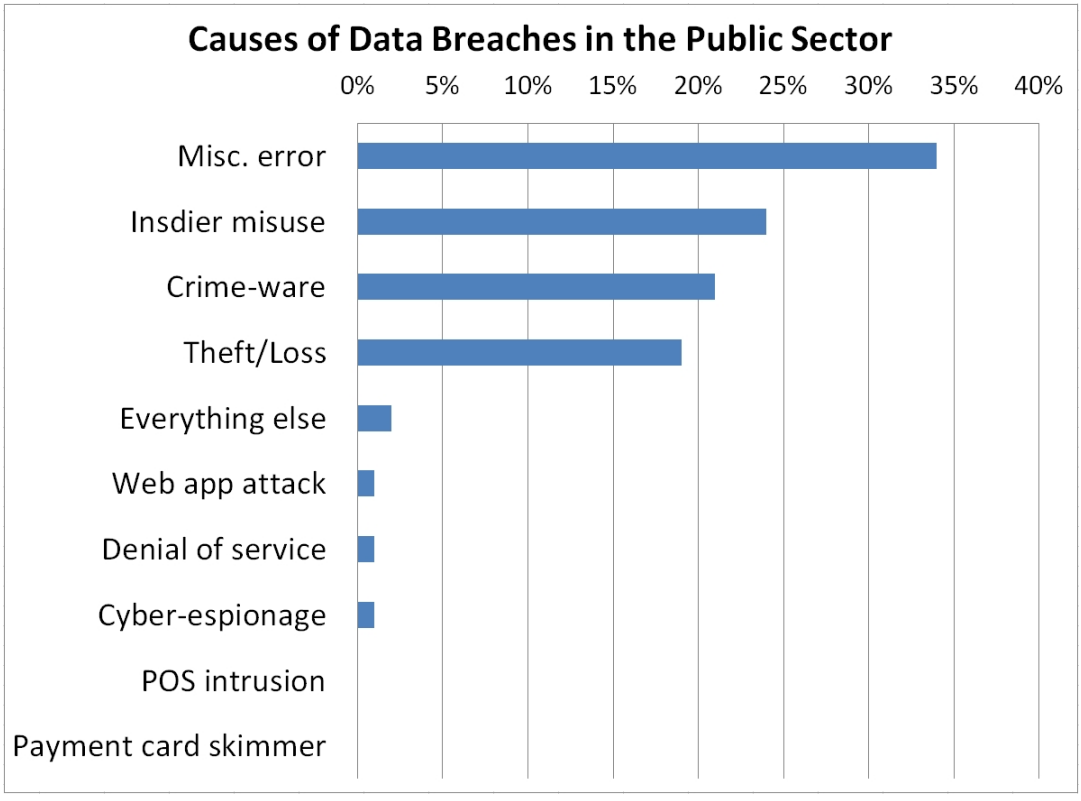

Figure 1: Causes of Data Breaches[16]

2.1 ICO's Sending Details of Adoptive Parents

The Information Commissioner's Office, which is an independent authority in the UK that promotes openness of public information and protection of private information[18], went through a breach where

a council employee sent a letter about an adopted child to the birth mother, and mistakenly included a covering letter giving details of the adoptive parents' home address. The investigation on the incident concluded that the council employee did not have a clear policy[17]. He was definately carelessness.

2.2 Australian Federal Police's Publishing Metadata from Criminal Investigations

The Australian federal police mistakenly published highly sensitive information including metadata connected to criminal investigations. The information had been available online on parliamentary sites even for several years. The published information contained the address of a surveillance target, metadata of criminal cases being investigated at that time, and some officers' information such their names and phone numbers.[8]

2.3 University of Nottingham Sending Personal Details of Job applicats

The University of Nottingham sent confidential details of thousands of job applicants mistakenly via e-mail. It included 4,751 job applicants' name and their job interviews' results.[5]

2.4 California's Posting 14,000's Social Security Numbers

California's Medicaid health-assistance program accidentally published the Social Security numbers of 14,000 individuals affiliated with the program. The published information included the participants' highly personal information such as names, home addresses as well as their social security numbers. The information has published on a government site for nine days until they discovered their mistake on Nov. 14 according to Kaspersky's Threatpost blog report.[4] The breach is not done by a hacker. It happaned by mistake, but the result was disastrous.

2.5 IRS's Posting Social Security Numbers on Website

The IRS mistakenly posted the Social Security numbers of tens of thousands of Americans on a government website. They confirmed the fact the day after the breach occured.[2]

2.6 Patient data breach at Rady Children's Hospital

Employees at Rady Children's Hospital mistakenly forwarded a spreadsheet that contained their private health information to a handful of job applicants by email. The file had protected information about 14,121 patients including their names, dates of birth, primary diagnoses, admittance and discharge dates, medical record numbers, and other information like insurance claim information.[19]

2.7 Conclusion from the Cases

Among the introduced cases, 3 cases are about sending information that were not allowed to be sent. 2.1 2.3 And other 3 cases are about publishing confidential information mistakenly and faling to discover it. Each case shows different aspects of carelessness.

3 Enron Email Dataset

Enron email dataset is widedly used as a resource for researchers wanting to improving current email tools, or understanding how email is currently used.[7] It is from the actualy company, Enron, which was an American enrgy, commodities, and services company based in Houston.[22] The email dataset

contains email messages from 168 uesrs working in senior management of Enron. It contains 619,446 messages, and every message in the dataset is organized into folders.[6] But a lot of the same messages are organized into multiple folders, and some messages are computer generated. After removing duplicated messages, we got about 192,798 distinct messages. The following are the meta properties of each message. A mail's content follows these meta properties in the text file. Figure 2 shows an actual message file.

- Message-ID - an unique ID for each message
- Date - date when the mail was sent
- From - an email address of a sender, who wrote an email message
- To - an email addresses of a receivers, who received an email message
- Subject - a title of an email message
- Mime-Version - The version is 1.0.
- Content-Type - The default content type is 'text/plain'.
- Content-Transfer-Encoding - The default encoding is 7bit.
- X-From - a name of a sender
- X-To - names of a receivers
- X-cc - Carbon Copy
- X-bcc - Blind Carbon Copy
- X-Folder - a path of a folder where a message is
- X-Origin - a name of an user folder where a message is
- X-FileName - file names attached to a message

4 Finding Email Messages Not Replied To

Our experience suggests that not replying to an email message is about faling to discover something that has to be dealt with or forgetting about it. Therefore, we suggest if a recever does not reply to an message, it reflects his or her carelessness which is the reason not to discover the need to reply to the message or to forget to reply to it. However, every message is not for being received a reply from a receiver. Messages can be divided into two types: messages whose sender expects get a reply from the receiver of it, and messages whose sender does not expect any reply. In this section, we are going to explain the experiment where we classified messages into those two types, which is the first step to find messages that are not replied to. Figure 3 indicates the steps of the expriment we conducted.

```
Message-ID: <17796836.1075840054859.JavaMail.evans@thyme>
Date: Thu, 29 Nov 2001 19:35:20 -0800 (PST)
From: dcherry@bpa.gov
To: scranda@ect.enron.com
Subject: NYTimes.com Article: Foundation Gives Way on Chief's Big Dream
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: dcherry@bpa.gov
X-To: scranda@ect.enron.com
X-cc:
X-bcc:
X-Folder: \ExMerge - Crandall, Sean\Inbox
X-Origin: CRANDELL-S
X-FileName:

This article from NYTimes.com has been sent to you by dcherry@bpa.gov.
FYIdcherry@bpa.gov/---------- advertisement
----------\Special Offer to NY Times customers:Spend $100 &
ship free at Starbucks.com
http://www.nytimes.com/ads/starbucks/email.html
\------------------------------------------------------------/Foundation Gives
Way on Chief's Big DreamNovember 29, 2001 By JOHN SCHWARTZ and
RICHARD A. OPPEL Jr. Well before anyone could imagine that Enron might
collapse,Kenneth L. Lay was stumped. In an interview in August, he
dismissed questions about a vague clause in the energycompany's annual
report that hinted at bigger problems ifits stock price or credit rating fell
below certain levels."I just can't help you on that," he said. Pressed
furtheron questions about a bewildering constellation of business
partnerships that involved Enron's former chief financialofficer, he said,
"You're getting way over my head." At the least, Mr. Lay - a man of big
ideas, a crusader forfree markets, a risk taker in the Texas wildcatter
tradition - had taken his eye off the ball. While he wasbusy befriending the
nation's most powerful politicians,erecting one of the tallest buildings in
Houston andpasting Enron's logo on the city's new ballpark, the littlethings
were turning out to be Mr. Lay's big problems. One after another,
disclosures spilled out of his companyover the last month: the partnerships
had hidden billionsin debt; years of Enron's reported profits had been
exaggerated; the government was investigating. Rivals wereshunning
```

Figure 2: Enron Email Message File Format

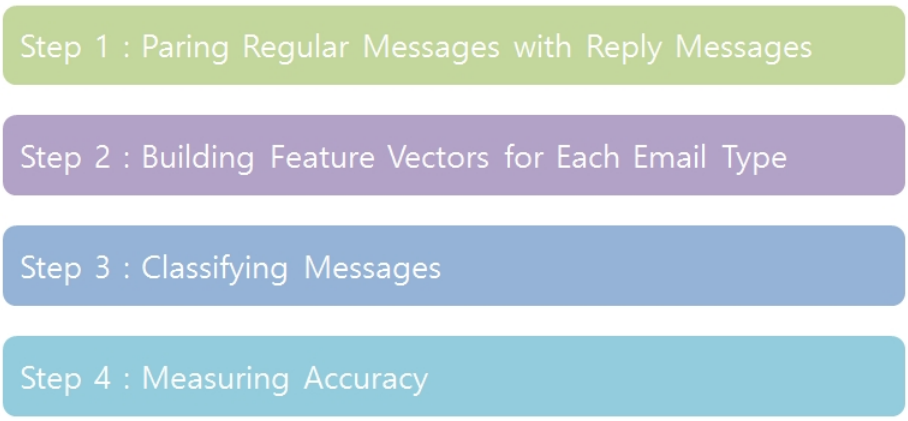

Figure 3: The Steps of the Experiment

4.1 Paring Regular Messages with Reply Messages

For every mail whose subject does not start with 'RE:' or 'FW:'-we call it a regular message, we treid to find another message whose subject starts with 'RE:'-we call it a reply message- and is the same as the regular message's subject if 'RE:' is removed. In addition to considering the subjects of messages, we also considered other factors like email addresses, and the time when those messages were sent. After these procedures, we got 1,897 regular messages each of which has at least 1 reply message paring with the retgular message.

4.2 Intermediate Result Data Format

For the data format of the result file after the paring process, which is shown in Figure 4, we adopted RDF (Resource Description Framework) language, which is suitable for distributed processing using frameworks such as Hadoop[1]. For a paried message, there are four properties: replymsg, subject, from, to. For sending a message to a receiver, four lines of text are generated. If as message was sent to 2 receivers, then 8 ilnes of text are written. The following are the properties of each messages.

- replymsg - a name of a file that contains a reply of a message
- subject - a subject of a message that a file has.
- from - a sender of a file
- to - a reiever of a file

```
<msg:157842.txt> <prop:replymsg>  <msg:153270.txt>
<msg:157842.txt> <prop:subject>   "NUI Energy Brokers, Inc. financial trades"
<msg:157842.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:157842.txt> <prop:to>        <address:tanya.rohauer@enron.com>
<msg:174409.txt> <prop:replymsg>  <msg:150658.txt>
<msg:174409.txt> <prop:subject>   "Chase Master Agreement - "no more confirms""
<msg:174409.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:174409.txt> <prop:to>        <address:tanya.rohauer@enron.com>
<msg:157642.txt> <prop:replymsg>  <msg:153622.txt>
<msg:157642.txt> <prop:subject>   "amending ISDA Masters"
<msg:157642.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:157642.txt> <prop:to>        <address:tanya.rohauer@enron.com>
<msg:156191.txt> <prop:replymsg>  <msg:153199.txt>
<msg:156191.txt> <prop:subject>   "Bank Brussels Lambert (acquired by ING)"
<msg:156191.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:156191.txt> <prop:to>        <address:tanya.rohauer@enron.com>
<msg:156334.txt> <prop:replymsg>  <msg:153199.txt>
<msg:156334.txt> <prop:subject>   "Bank Brussels Lambert (acquired by ING)"
<msg:156334.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:156334.txt> <prop:to>        <address:tanya.rohauer@enron.com>
<msg:157889.txt> <prop:replymsg>  <msg:153199.txt>
<msg:157889.txt> <prop:subject>   "Bank Brussels Lambert (acquired by ING)"
<msg:157889.txt> <prop:from>      <address:sara.shackleton@enron.com>
<msg:157889.txt> <prop:to>        <address:tanya.rohauer@enron.com>
```

Figure 4: Intermediate Data Format for the Experiment

4.3 Building Feature Vectors for Each Email Type

After the paring process, we found that there are 1,897 messages having at least 1 reply message and other 191,475 messages not having any reply. We used the contents of the messages having reply messages to make a feature vector of the type of messages requiring a reply-we call the type 'Type R', and other messages' contents to make a feature vector of the type of messages not requiring any reply-we call the type 'Type N'. For the feature vectors, we adopted bag-of-words model[11]. Every element of a feature vector for a type is the number of a boolean frequency of words-how many messages among the type's messages have the word- divided by the number of the all messages belonging to the email type.

4.4 Classifying Messages

With the feature vectors of the two types, we tried to classify all the 192,798 messages into the two types again. The following are the procedure to classify the messages.

1. Make a list of words occurring at least once in each message's content except numbers, punctuation marks.
2. Sum Type R's feature vector's values of the words which the list contains.
3. Sum Type N's feature vector's values of the words which the list contains.
4. Compare the two sum values and decide which sum is bigger.
5. If the sum of Type R's values is bigger than that of Type N's, the message is classified into Type R, otherwise it is classified into Type N.
6. Mesaure the accuracy of the result by counting the number of messages that were replied to in Type R messages, and the number of messages that were not replied to in Type N, then you get the precision and the recall[12].

4.5 Experiment Result

With the procedures we introduced, we got 2,473 messages classified into Type R and other 190,325 messages classified into Type N. Among the messages that were classified into Type R, 603 messages were the ones which actually had reply messages paring to each of them. Therefore, for the messages classified into Type R, the positive predictive value (precision) is about 24.38%. And considering that the number of all messages having at least one reply was 1,897, the sensitivity (recall) is about 31.79%. As for the messages classified into Type N, there were 189,031 messages not having any reply among 190,325 messages classified into Type N meaning the precision was about 99.32%. And the number of all messages having no reply was 190,901 meaning the recall was 99.02%. Given the percentage of the messages having replies which is less than 1% (about 0.98%), those values are meaningful. Figure 5 shows the overall experiment statistics.

4.6 Finding Messages Not Being Replied To

The experiment we conducted gave us the precision of 24.38%, which means the 24.38% messages of Type R truned out to have at least a reply. But we assume other about 75% messages' senders were more likely to expect to recevie a reply to their messages than those of the 190,325 Type N messages but did not receive any reply. In other words, the classification claims that those classified into Type R but not having any reply might fail to be replied to. The higher the recall rate we get, The more reasonable the assumption will become.

All Messages	192,798 messages		
Having Reply	1,897 messages	Having No Reply	190,901 messages
Classified Type R	2,473 messages	Classified Type N	190,325 messages
Having Reply in Type R Messages	603 messages	Having No Reply in Type N Messages	189,031 messages
Precision for Type R	24.38% (603/2,473)	Precision for Type N	99.32% (189,031/190,325)
Recall for Type R	31.78% (603/1,897)	Recall for Type N	99.02% (189,031/190,901)

Figure 5: The Results of the Experiment

5 Conclusion

In this paper, we suggested email messages exchanged among employees in a business are a useful clue to the employees' carelessness and it could lead to unintentional insider threats. Through the case study, we found some actual examples that caused threats like data breach by mistake. For the experiment, we used enron email dataset that contains about 150 users who worked for Enron, and about 500,000. Removing some computer generated files, we got about 190,000 distinct messages. With the pre-processed dataset, we tried to found 1,897 messages that received at least one reply. They were used to build a feature vector of Type R meaning a type of messages that usually receive a reply. Ohter messages were used to build a feature vector of Type N meaning a type of messages that don't receive a reply. With the two feature vectors, we tried to classify all the messages into Type R or Type N. As a result, for Type R, we got the precision of 24.38%, and the recall of 31.79%. As for Type N, the precision was 99.32% and the recall was 99.02%.

5.1 Future Work

For the future work, we will try to improve those precisions and recalls of the experiment by adopting more advanced techniques like tf-idf[24] or Singular Value Decomposition[3] for the classification process. And we will also focus on other email behaviors. Sending an email message for wrong addresses is one of the interesting email behavior, which also reflects senders' carelessnesss, and could lead to serious insider threats like data breach like the ones introduced in Section 2. For handling more big email dataset, we plan to use Hadoop framework so that we can get better performance and use more advance machine learning techniques that are time-consuming for a single machine.

5.2 Acknowledgments

- This work was supported by the IT R&D program of MSIP/IITP. [2014-044-024-002 , Developing On-line Open Platform to Provide Local-business Strategy Analysis and User-targeting Visual Advertisement Materials for Micro-enterprise Managers]

References

[1] What Is Apache Hadoop? http://hadoop.apache.org/.

[2] Group: IRS mistakenly posted thousands of Social Security numbers on website, 2013. http://www.foxnews.com/politics/2013/07/09/group-irs-mistakenly-posted-thousands-social-security-number-on-website/.

[3] K. Baker. Singular Value Decomposition Tutorial, 2005. http://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf.

[4] N. Ben. California Accidentally Posts 14,000 Social Security Numbers, 2012. http://www.nbcnews.com/id/50186718/ns/technology_and_science-tech_and_gadgets/#.VFGnufmsVR4.

[5] M. Bookcock. UK: University of Nottingham Mistakenly Sends E-mail Containing Personal Details of Job Applicatns, 2014. http://www.nottinghampost.com/University-Nottingham-mistakenly-sends-e-mail/story-21114748-detail/story.html.

[6] Y. Y. Bryan Klimt. Introducing the Enron Corpus, 2004. http://ceas.cc/2004/168.pdf.

[7] W. W. Cohen. Enron Email Dataset, 2009. https://www.cs.cmu.edu/~./enron/.

[8] P. Farrel. Federal police mistakenly publish metadata from criminal investigations, 2014. http://www.theguardian.com/world/2014/aug/28/federal-police-mistakenly-publish-metadata-from-criminal-investigations.

[9] J. J. Graham Klyne. Resource Description Framework (RDF): Concepts and Abstract Syntax, 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[10] I. Information Shiled. The Insdier Threat - Security Policies to Reduce Risk, 2011. http://www.informationshield.com/papers/Security%20Policies%20Address%20the%20Insider%20Threat.pdf.

[11] jemdoc. Bag-of-words representation of text. https://inst.eecs.berkeley.edu/~ee127a/book/login/exa_vecs_bag_of_words_rep.html.

[12] R. Jizba. Recall and Precision, 2007. https://www.creighton.edu/fileadmin/user/HSL/docs/ref/Searching_-_Recall_Precision.pdf.

[13] N. B. Johnson. DHS cyber effort shifts to insider threats, 2013. http://www.federaltimes.com/article/20131216/DHS/312160014/DHS-cyber-effort-shifts-insider-threats.

[14] T. Lombardo. SECNAV launches plan to battle 'insider threats', 2013. http://www.navytimes.com/article/20130907/NEWS/309070005/SECNAV-launches-plan-battle-insider-threats-.

[15] C. G. Matt Bishop. Defining Insider Threat, 2008. http://nob.cs.ucdavis.edu/bishop/papers/2008-csiiw/definsider.pdf.

[16] L. Mirani. The single biggest cause of government data breaches is "oops", 2014. http://qz.com/201699/the-single-biggest-cause-of-government-data-breaches-is-oops/.

[17] I. C. Office. Adoptive Parents' Details Mistakenly Sent to Birth Family in Council Data Breach, 2013. http://ico.org.uk/news/latest_news/2013/adoptive-parents-details-mistakenly-sent-to-birth-family-05062013.

[18] M. Rouse. Information Commissioiner's Office (ICO), 2008. http://searchstorage.techtarget.co.uk/definition/Information-Commissioners-Office-ICO.

[19] P. Sisson. Patient data breach at Rady Children's, 2014. http://www.utsandiego.com/news/2014/jun/17/rady-breach-data-patients-childrens-hospital/.

[20] T. C. I. T. Team. Unintentional Insider Threats: A Foundational Study, 2013. http://www.sei.cmu.edu/reports/13tn022.pdf.

[21] J. Vijayan. DARPA launches insider threat detection effort for military, 2010. http://www.computerworld.com/article/2515328/security0/darpa-launches-insider-threat-detection-effort-for-military.html.

[22] Wikipedia. Enron Email Dataset, 2014. http://en.wikipedia.org/wiki/Enron.

[23] Wikipedia. Insider Threat, 2014. http://en.wikipedia.org/wiki/Insider_threat.

[24] Wikipedia. tf-idf, 2014. http://en.wikipedia.org/wiki/Tf%E2%80%93idf.

Author Biography

Sung-min Kim received a B.E degree in computer science from Konkuk University in 2013. He is currently working as a researcher at UCLab(Ubiquitous and Cloud Computing Laboratory) and studying for a master's degree at Konkuk University.

Yeong-seok Son is currently working as a researcher at UCLab(Ubiquitous and Cloud Computing Laboratory) and studying for a master's degree at Konkuk University.

Young-guk Ha received the PhD. degree from KAIST(Korea Advanced Institute of Science and Technology). He worked as a senior researcher at ETRI (Electronics and Telecommunications Research Institute) from 1995 through 2008. He was a professor at UST(University of Science and Technology). He is currently a professor at Konkuk University, and directing UCLab(Ubiquitous and Cloud Computing Laboratory).