SOUTHERN ILLINOIS UNIVERSITY Carbondale

A Comparison of Database Insider Attack Monitoring Algorithms using Page Ranks with In/Out Links

Alan Y. Park. Advisor: Sam Chung

School of Information Systems & Applied Technologies, College of Applied Sciences and Arts, Southern Illinois University Carbondale

Summary	PageRank	Weighted PageRank	Comparison Criteria			
The purpose of this research is to compare two database insider attack monitoring approaches that use page rank algorithms: PageRank and	• Defining the Query Transition Probability Matrix K as $K = \{k \mid i \in I, n = I \}$	 The weight of link(v,u) calculated based on the number of in-links. 	PageRank Weighted PageRank			

Weighted PageRank. By calculating the weight of queries, we predict the users' pattern in the database system.

Keywords

- Insider Attack Monitoring,
- Web Mining,
- PageRank Algorithm,
- Weighted PageRank Algorithm,
- Markov Chain

Motivation

Nowadays, an average attacker takes less than ten seconds to hack information systems. Specially, the insider attacks that any malicious attack on the database systems performed by an entrusted group of people having authorized access are more dangerous than the outsider attacks since it has more entry points of malicious attacks than ones from outsider attacks. $K = \{k_{ij} \mid i, j \in I, n = |I|, k_{ij} = 0 \text{ if there is no link between } Q_i \text{ and } Q_j$ $k_{ij} = 1/\text{the number of out links of } Q_i$ $\sum_{j=1}^n k_{ij} = 0|1 \text{ for the fixed } i\}.$



- Building the Query Rank Probability Matrix M as
 - $$\begin{split} &M = \left\{ m_{ij} \mid i,j \in I, n = |I|, \\ &\sum_{j=1}^{n} m_{ij} = 1 \text{ for the fixed } i \right\} \\ &r_i = \sum_{j=1}^{n} k_{ij}, \ \delta = (1-p)/n \\ &m_{ij} = \left\{ \begin{array}{c} pk_{ij} + \delta \ : \ r_i = 1 \\ \frac{1}{n} \ & : \ r_i = 0 \end{array} \right\} \end{split}$$
- $\mathcal{M} = \begin{bmatrix} 0.0375 & 0.4625 & 0.0375 & 0.4625 \\ 0.0375 & 0.0375 & 0.8875 & 0.0375 \\ 0.8875 & 0.0375 & 0.0375 & 0.0375 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{bmatrix}$
- We assume that all queries are ranked equally at the



 The weight of link(v,u) calculated based on the number of out-links.



 Page A has two reference pages p1 and p2. The inlinks of p1 is 2 and p2 is 1, outlinks of p1 is 2 and p2 is 3







 Query 1
 0.333
 0.433

 Query 2
 0.433
 0.232

 Query 3
 0.234
 0.183

 Case 2

Evaluation and Conclusion

Problem Statement

Which approach is better to monitor insider attacks from users who can access database systems with legitimate access controls between using PageRank and Weighted PageRank algorithms?

Approach

By using both PR and WPR algorithm, we calculate the weight of queries and distributes rank scores based on the popularity of the queries. Our case study shows that two algorithms brought different outcomes to our query rank based database insider attack monitoring. From the value of the weight of queries, we are able to select suspicious queries that will not occur regularly on the database system. Based on both the PR and the WPR algorithm, we are able to extract monitoring information from users' past behavior on the database system.

beginning.

formula is modified as

$$\begin{split} \lambda_0 &= \begin{bmatrix} 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{bmatrix} \\ \lambda_1 &= \lambda_0 M^1 = \begin{bmatrix} 0.3031 & 0.1969 & 0.3031 & 0.1969 \end{bmatrix} \\ \lambda_2 &= \lambda_1 M^1 = \lambda_0 M^1 M^1 = \lambda_0 M^2 \\ \lambda &= \lambda_0 M^{17} = \begin{bmatrix} 0.3078 & 0.2138 & 0.2646 & 0.2138 \end{bmatrix} \end{split}$$

 $PR(u) = (1 - d) + d \sum PR(v)W_{(v,u)}^{in}W_{(v,u)}^{out}$ $v \in B(u)$

Calculation using RStudio

R PStudio					- 1	×
M KStudio					Ľ	~
File Edit Code View Plots Session Build Debug Tools Help						
🔍 🗸 🚭 🖌 🔝 🚔 📄 🍌 Go to file/function 📄 📰 🔹 Addins 👻					萬 Project:	(None) 🔹
Untitled1 ×	-	Environment History				-
🗇 🗘 📃 🔲 Source on Save 🔍 🎽 📲	🕞 Run 🛛 💽 Source 🔹 🚍	🕣 🔒 📑 Import Dataset 🔹 🧹			📃 List	- 6
1		Global Environment 🕶			Q	
		Data				•
		QueryMatrix	num [1:3, 1:3] 0.05 0.05 0.9 0.4	475 0.05 0.0	5 0.475 0.9 0.05	
		result1	num [1, 1:3] 0.333 0.191 0.475			
		result10	num [1, 1:3] 0.389 0.214 0.396			
		result11	num [1, 1:3] 0.387 0.215 0.397			
		result12	num [1, 1:3] 0.388 0.214 0.397			
		result13	num [1, 1:3] 0.388 0.215 0.397			
		result14	num [1, 1:3] 0.387 0.215 0.397			
		result15	num [1, 1:3] 0.387 0.215 0.397			
		result16	num [1, 1:3] 0.387 0.215 0.397			
		result17	num [1, 1:3] 0.387 0.215 0.397			
		Files Plots Packages Help Vi	ewer			-0
		😘 New Folder 🛛 Delete 📑 Rena	ame 🤷 More 🕇			E
		▲ Name		Size	Modified	•
		.RData		3.4 KB	Jul 1, 2016, 10:10 PM	
		.Rhistory		2.3 KB	Sep 17, 2016, 11:07 AM	
1:1 (Top Level) =	R Script 🛊					
Console ~/ 🔗	-0	ASUS				

Consequently, Since the PageRank algorithm only takes the number of out-links as a factor while the WPR algorithm takes number of both inlinks and out-links, we have the different value of the weight of queries.

Future Works

- Applying both algorithms to the database system.
- Visualization of the query weight by using Rstuido.

Reference

 Cheolmin Sky Moon, Sam Chung, Barbara Endicott-Popovsky, Multiple-Criteria Query Statement Probabilities Based Database Insider Attack Monitoring System, University of Washington Tacoma MS Capstone, 2014.
 Larry Page, and Sergey Brin, Rajeev Motwanit, Terry Winograd, "The PageRank Citation Ranking: Bring Order to the Web", Technical report in Standford University, 1998.
 Wenpu Xing and Ghorbani Ali, "Weighted PageRank Algorithm", Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR '04), IEEE, 2004.

> QueryMatrix Q1 Q2 Q3	^		Custom Office Templates
Q1 0.05 0.475 0.475 Q2 0.05 0.050 0.900			G Freemake
Q3 0.90 0.050 0.050 > QueryBegin = c(0.333,0.333,0.333) > QueryBegin %*% QueryMatrix			GitHub
Q1 Q2 Q3 [1,] 0.333 0.191475 0.474525			matrix 2.3 KB Jul 1, 2016, 10:10 PM
<pre>> result1 = QueryBegin %*% QueryMatrix > result2 = result1 %*% QueryMatrix</pre>			Outlook Files
> result2 Q1 Q2 Q3 [1] 0 4532963 0 191475 0 3542288			🧐 R
<pre>> result2 = result1 %*% QueryMatrix > result3 = result2 %*% QueryMatrix</pre>			Virtual Machines
<pre>> result4 = result3 %*% QueryMatrix > result5 = result4 %*% QueryMatrix</pre>			VPProjects
<pre>> result6 = result5 %*% QueryMatrix</pre>	•	_	

ASARS 2016

