CS 7863: Network Theory Final Project: Compliance Graph Analysis

Noah Schrick

May 3, 2022

Contents

1 Introduction

1.1 Attack Graphs

To address the rising risks of computing and threats to cybersecurity, vulnerability analysis modeling is a technique employed by experts to identify weak points in a system or set of systems. One such modeling approach is to represent the system or set of systems through graphical means, with system information encoded into the nodes and edges of the graph. This modeling approach was first utilized in the 1990s in a format called attack trees, and can be seen through the works of the authors of [1] and [2]. These attack trees would later be expanded into attack graphs.

Attack graphs begin with a root node that contains all the current information of the system or set of systems. From this initial root state, all assets in the system are examined to see if any single modification can be made, where a modification is typically a change in system policy or security settings. If a modification can be made, an edge is drawn from the previous state to a new state that includes all of the previous state's information, but now reflects the change in the system. This edge is labeled to reflect which change was made to the system. This process is exhaustively repeated, where all system properties are examined, all attack options are fully enumerated, all permutations are examined, and all changes to a system are encoded into their own independent states, where these states are then individually analyzed through the process.

1.2 Compliance Graphs

Compliance graphs are an alternate form of attack graphs, utilized specifically for examining compliance and regulation statuses of systems. Like attack graphs, compliance graphs can also be used to determine all ways that systems may fall out of compliance or violate regulations. These graphs are notably useful for cyber-physical systems due to the increased need for compliance. As the authors of [3], [4], and [5] discuss, cyber-physical systems have seen greater usage, especially in areas such as critical infrastructure and Internet of Things.

The semantics of compliance graphs are similar to that of attack graphs, but with a few differences regarding the information at each state. While security and compliance statuses are related, the information that is analyzed in compliance graphs is focused less on certain security properties, and is expanded to also examine administrative policies and properties of systems. Since compliance and regulation is broad and can vary by industry and application, the information to analyze can range from safety regulations, maintenance compliance, or any

other regulatory compliance. However, the graph structure of compliance graphs is identical to that of attack graphs, where edges represent a modification to the systems, and nodes represent all current information in the system.

1.3 Difficulties of Compliance Graph Analysis

Analysis of directed graphs is not as simple as their undirected counterparts, and attack and compliance graphs are directed acyclic graphs. The primary contributor to the increased difficulty is due to the asymmetric adjacency matrix present in directed graphs. With undirected graphs, simplifications can be made in the analysis process both computationally and conceptually. Since the "in" degrees are equal to the "out" degrees, less work is required both in terms of parsing the adjacency matrix, but also in terms of determining importance of nodes. The author of [6] discusses that common analysis techniques such as eigenvector centrality is often unapplicable to directed acyclic graphs. As the author of [7] discusses, the difficulty of directed graphs also extends to the graph Laplacian, where the definition for asymmetric adjacency matrices is not uniquely defined, and is based on either row or column sums computing to zero, but both cannot. The author of [7] continues to discuss that directed graphs lead to complex eigenvalues, and can lead to adjacency matrices that are unable to be diagonalized. These challenges require different approaches for typical clustering or centrality measures.

2 Related Works

The author of [8] presents three centrality measures that were applied to various attack graphs. The centrality measures implemented were Katz, K-path Edge, and Adapted PageRank. Each of these centrality measures are applicable to the directed format of attack graphs, and conclusions can be drawn regarding patching schemes for preventing exploits. As an approach for avoiding complex eigenvalues, the authors of [9] present work examining directed, undirected, and mixed graphs using its Hermitian adjacency matrix. Other works, such as that discussed by the author of [7] include mathematical manipulation of directed graph spectra (originally presented by the author of [10]) with Schur's Theorem to bound eigenvalues and allow for explicit computation, which can then be used for additional analysis metrics.

Network	Nodes	Edges	Connectivity (%)
Car	2491	12968	0.209
HIPAA	2321	8063	0.150
PCI DSS	61	163	4.381

Table 1: Network Properties for the Three Networks Utilized

3 Experimental Networks

The work conducted in this approach utilized three compliance graphs, with their properties displayed in Table 1. Connectivity in this table refers to the mean degree, divided by the number of nodes in the network, multiplied by 100 to get the number in a percentage form. Network 1 is a vehicle maintenance network. This network has one car asset that is deemed "brand new", and has no mileage. This network is examined at its current state, and progresses through time with time steps of 1 month, up to 12 months total. At each time step the car gains mileage and increases its age property, and is reexamined to evaluate its standing in regards to its vehicular regulatory maintenance schedule. Network 2 is an artificial company network that is attempting to maintain HIPAA compliance [11]. This network examines its standing in relation to security properties that are required per HIPAA guidelines, as well as employee cooperation to training and administrative policies. This network is also progressed through time to illustrate the company's standing in relation to yearly audits and trainings that must be followed. Employees are also added and removed through the network at set points during the time progression process. Network 3 is another artificial company network. This company is attempting to maintain PCI DSS compliance [12]. This network generation was static and did not progress through time. This network examined the company and its current state, and examined all changes that could occur. These changes were primarily tied to security properties such as physical break-ins on the property, firewalls being disabled, default system settings, and encryption expiration.

4 Centralities and their Applications to Compliance Graphs

4.1 Introduction

The author of [13] provides a survey of centrality measures, and discusses how various centrality measures have been implemented and brought forth in order to determine node importance in networks. By determining the importance of

nodes, various conclusions can be drawn regarding the network. In the case of compliance graphs, conclusions can be drawn regarding the prioritization of patching or correction schemes. If one node is known to lead to the creation of many other nodes, it may be said that a patch is imperative to prevent further opportunities for compliance violation. This work discusses five centrality measures, and discusses their application to compliance graphs.

4.2 Degree

Degree centrality is a trivial, localized measure of node importance based on the number of edges that a node has. In an undirected graph, the degree centrality is predicated solely on the number of edges. However, in the case of a directed graph, a distinction is drawn with a degree centrality oriented on the number of edges coming into a node, and another measure focused on the number of edges leaving a node. Both of these cases provide useful information for compliance graphs. When a node has a large number of other nodes it points to, this node may be prioritized since it creates further opportunity for violation. When a node has a large number of edges pointing to it, this node may be prioritized since the probability that systems may enter this state is higher due to the increased number of ways that a system could lead to this state.

4.3 Betweenness

Betweenness centrality ranks node importance based on its ability to transfer information flow in a network. For all pairs of nodes in a network, a shortest path is determined. A node that is in this shortest path is considered to have importance. The total betweenness centrality is based on the number of shortest paths that pass through a given node. For compliance graphs, the shortest paths are useful to identify the quickest way that systems may fall out of compliance. By prioritizing the nodes that fall in the highest number of shortest paths, correction schemes can be employed to prolong or prevent systems from falling out of compliance.

Betweenness centrality is given in Equation 1, where i and j are two different, individual nodes in the network, σ_{ij} is the total number of shortest paths from i to j, and $\sigma_{ij}(v)$ is the number of shortest paths that include a node v.

$$\sum_{i \neq i \neq v} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{1}$$

4.4 Katz

Katz centrality was first introduced by the author of [14], and measures the importance of nodes through all paths in a network. Katz centrality varies in that its centrality measure is not limited to solely the shortest path between any two given nodes. The original work by the author defines Katz as seen in Equation 2, where i and j are nodes in the network, n is the total number of nodes in the network, A is the adjacency matrix, and α is an attenuation factor and has a value between 0 and 1. From this, a value of 1 is assigned if node i is connected to node j.

$$C_{\text{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ji} \tag{2}$$

Later works have expanded on the original Katz to include a β vector that allows for additional scaling in the instance that prior knowledge of the network exists. The modified equation can be seen in Equation 3.

$$\vec{x} = (I - \alpha A)^{-1} \vec{\beta} \tag{3}$$

For compliance graphs, Katz centrality represents the total number of paths that exist from a given node to any other downstream nodes, and is scaled based on the attenuation factor as well as the prior knowledge vector β. When the Katz centrality of a given node is high, prioritizing a correction scheme for the node would be useful to prevent opportunity of future compliance violations that may be many steps ahead, but still reachable from the current state.

4.5 K-Path Edge

K-path edge centrality, as discussed by the authors of [15], is predicated on information passing through a network as a means of generalizing k-path centrality. With K-path edge centrality, importance is based on the edges of the network. One difference from betweenness centrality, is that as discussed in Section 4.3, betweenness centrality is global and counts all nodes in a the shortest path. K-path edge centrality is localized, and is constrained by k steps from a given node. Equation 4 displays the centrality measure for K-path edge centrality, where m is a given edge in the network, N is the total number of nodes in the network, $\delta_n^{(K)}$ is the number of K-paths from node n, and $\delta_n^{(K)}(m)$ is the number of K-paths from node n that include edge m.

$$L^{(K)}(m) = \sum_{n=1}^{N} \frac{\delta_n^{(K)}(m)}{\delta_n^{(K)}} \tag{4}$$

For compliance graphs, K-path edge centrality is useful to identify a short chain of changes that may result in a compliance violation. If a node has a high K-path edge centrality and it is likely that the system will be put into that node, then a series of changes could occur that could then put the system in a different states. Prioritizing nodes that have a high K-path edge centrality could be useful in deterring a short chain of changes that could cripple the system further. It is also useful to prevent states where the system is near a compliance violation.

4.6 Adapted Page Rank

The original PageRank algorithm was first designed by the authors of [16] for the Google prototype for ranking web pages. The authors of [17] later introduced an Adapated PageRank that was designed to measure both the number and quality of connections specifically for an urban network. Equation 5 displays the PageRank algorithm, where γ is a damping factor with a value between 0 and 1, n is the total number of nodes in the network, A is the adjacency matrix of the network, i and j represent the row and column of the adjacency matrix, x is a given node in the network, and k is the row sum out degree. Since the Adapted PageRank algorithm measures the quality of connections, there is increased application to directed networks such as compliance graphs. As seen in Equation 5, the k_j term is a penalizing factor. Importance is based on the in degree of a node, with a penalty for the out degree. If many nodes point to a given node, then that node is said to be important due to its accessibility.

$$x_i = \frac{1-\gamma}{n} + \gamma \sum_{j=1}^{n} \frac{A_{ij}}{k_j} x_j \tag{5}$$

The adapted PageRank algorithm includes additional data that may be present in an urban network, such as geographical position, resource availability, and proximity to facilities. This data is user-defined, and may not be present in the network. Equation 6 displays the Adapated PageRank algorithm in matrix form where D is the user-defined data matrix, I is the identity matrix, and $\mathbb{1}$ is a column matrix comprised of 1s.

$$(I - \gamma AD)\vec{x} = \frac{1-\gamma}{n}\mathbb{1} \tag{6}$$

For compliance graphs, the Adapted Page Rank algorithm is useful for a few reasons. First, it is able to include user-defined data regarding the network. This could include scaling certain nodes to have greater weight, such as those known to be a compromised state. Second, since nodes are penalized for pointing to other nodes, this algorithm is useful for determining nodes that are likely to be visited. If a state has a greater in degree, it may need prioritization since the system has a higher likelihood of being placed in this state.

5 Transitive Closure

5.1 Introduction and Application

Transitive closure represents a transitive relation on a given binary set, and can be used to determine reachability of a given network. Figure 2 [1] displays an example output when performing transitive closure. In context of compliance graphs, it is useful to consider that an adversary (whether an internal or external malicious actor, poor policy execution by an organization, accidental misuse, or any other adversarial occurrence) could have no time constraints. That is, for any given state of the system or set of systems, an adversarial act could have "infinite" time to perform a series of actions. If no prior knowledge is known about the network, it can be assumed that all changes performed on the systems are equally likely. In practice, specifying a probability that a change can occur has been performed through a Markov Decision Process, such as that seen by the authors of [18] and [19]. When under these assumptions, it is useful to then consider which nodes are important, assuming they have 1-step reachability to any downstream node they may have a transitive connection to. As a result, a transitive closure was identified for all networks described in Section 3, and this transitive closure was then analyzed through the five centrality methods discussed in Section 4. Results and a discussion of the results can be seen in Section 7.

[1] Image origin can be located at: https://commons.wikimedia.org/wiki/File:Transitive-closure.svg, and this image has been licensed under the terms of the GNU Free Documentation License.

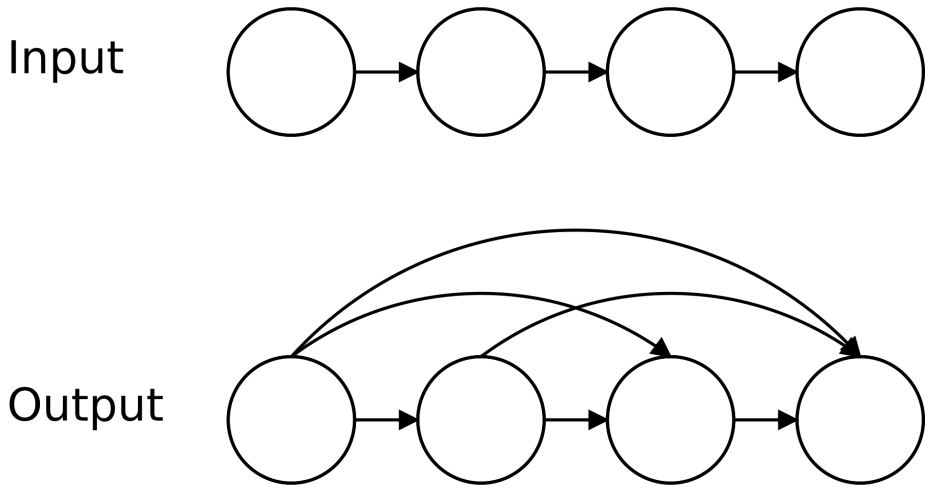

Figure 1: Example of Transitive Closure

6 Dominant Tree

6.1 Introduction and Application

Dominance, as initially introduced by the author of [**?**] in terms of flow, is defined as a node that is in every path to another node. For instance, if a node i is a destination node, and every path to i from a source node includes node j, then node j is said to dominate node i. Figure **??** [2] displays an example starting network. With node 1 being the source node, it is evident that node 2 immediately dominates nodes 3, 4, 5, and 6, since all messages from node 1 must pass through node 2. By definition, each node must also dominate itself, so node 2 also dominates node 2.

7 Results and Result Analysis

8 Conclusions and Future Work

[2] Image origin can be located at: https://commons.wikimedia.org/wiki/File:Dominator_control_flow_graph.svg, and this image has been released into the public domain for use for any purpose, unless such conditions are required by law.

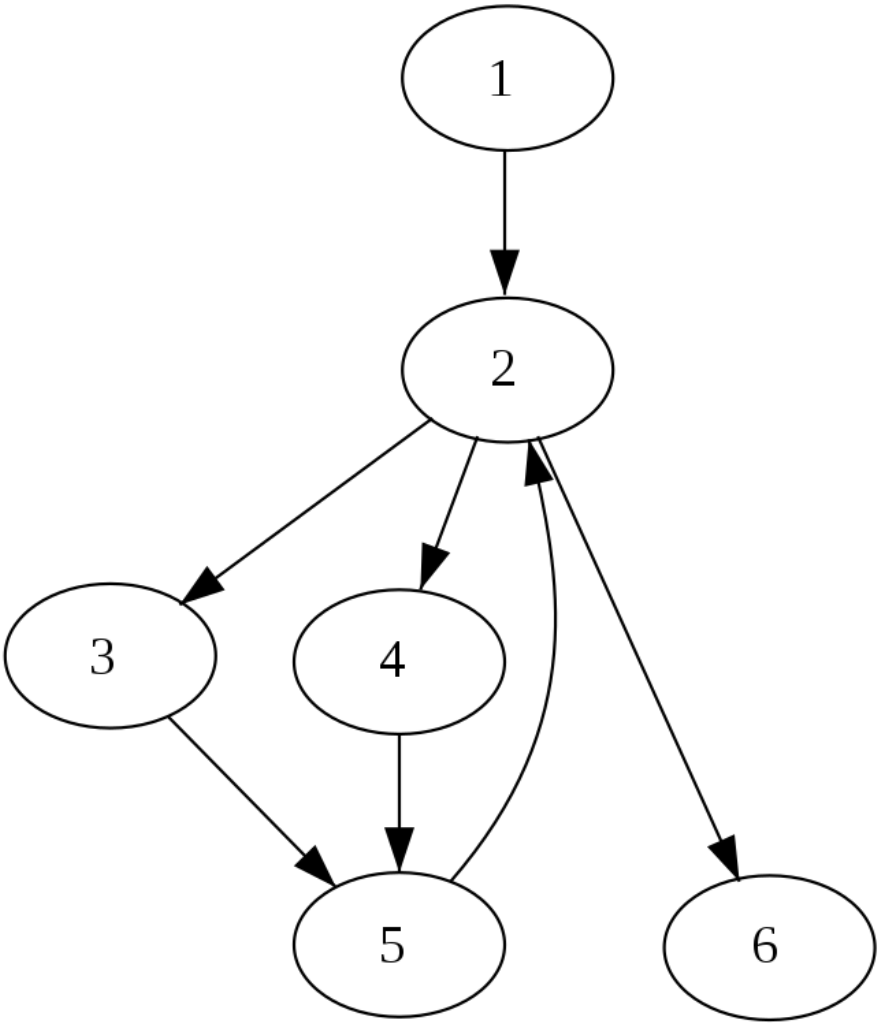

Figure 2: Example Network for Illustrating Dominance

References

[1] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," *Proceedings New Security Paradigms Workshop*, vol. Part F1292, pp. 71–79, 1998. doi: 10.1145/310889.310919.

[2] B. Schneier, "Modeling Security Threats," *Dr. Dobb's Journal*, 1999. vol. 24, no.12.

[3] J. Hale, P. Hawrylak, and M. Papa, "Compliance Method for a Cyber-Physical System." U.S. Patent Number 9,471,789, Oct. 18, 2016.

[4] N. Baloyi and P. Kotzé, "Guidelines for Data Privacy Compliance: A Focus on Cyberphysical Systems and Internet of Things," in *SAICSIT '19: Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019*, (Skukuza South Africa), Association for Computing Machinery, 2019.

[5] E. Allman, "Complying with Compliance: Blowing it off is not an option.," *ACM Queue*, vol. 4, no. 7, 2006.

[6] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[7] P. V. Mieghem, "Directed graphs and mysterious complex eigenvalues," 2018.

[8] M. Li, *A System for Attack Graph Generation and Analysis*. PhD thesis, The University of Tulsa, 2021.

[9] K. Guo and B. Mohar, "Hermitian adjacency matrix of digraphs and mixed graphs," *Journal of Graph Theory*, vol. 85, 2017.

[10] R. A. Brualdi, "Spectra of digraphs," *Linear Algebra and its Applications*, vol. 432, pp. 2181–2213, 2010.

[11] "Health Insurance Portability and Accountability Act of 1996." Pub. L. No. 104-191. 1996 [Online]. Available: https://www.govinfo.gov/content/pkg/PLAW-104publ191/html/PLAW-104publ191.htm.

[12] P. S. S. Council, "Payment Card Industry (PCI) Data Security Standard," May 2018. Available: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf.

[13] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari, "A systematic survey of centrality measures for protein-protein interaction networks," *BMC systems biology*, vol. 12, p. 80, July 2018.

[14] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, pp. 39–43, March 1953.

[15] P. D. Meo, E. Ferrara, G. Fiumara, and A. Ricciardello, "A novel measure of edge centrality in social networks," *Knowledge-Based Systems*, vol. 30, pp. 136–150, jun 2012.

[16] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107–117, 1998. Proceedings of the Seventh International World Wide Web Conference.

[17] T. Agryzkov, J. L. Oliver, L. Tortosa, and J.-F. Vicent, "An algorithm for ranking the nodes of an urban network based on the concept of pagerank vector," *Appl. Math. Comput.*, vol. 219, pp. 2186–2193, 2012.

[18] M. Li, P. Hawrylak, and J. Hale, "Combining OpenCL and MPI to support heterogeneous computing on a cluster," *ACM International Conference Proceeding Series*, 2019.

[19] K. Zeng, "Cyber Attack Analysis Based on Markov Process Model," 2017.