Improving bulk power system resilience by ranking critical nodes in the vulnerability graph

Conference Paper · April 2018

CITATIONS
4

READS
167

3 authors, including:

Md Ariful Haque
Old Dominion University
12 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Sachin Shetty
Old Dominion University
237 PUBLICATIONS **2,039** CITATIONS
SEE PROFILE

IMPROVING BULK POWER SYSTEM RESILIENCE BY RANKING CRITICAL NODES IN THE VULNERABILITY GRAPH

Md Ariful Haque

Department of Modeling, Simulation
& Visualization Engineering
Old Dominion University
5115 Hampton Blvd
Norfolk, VA, USA
mhaqu001@odu.edu

Sachin Shetty

Virginia Modeling Analysis
and Simulation Center
Old Dominion University
1030 University Blvd
Suffolk, VA, USA
sshetty@odu.edu

Gael Kamdem

Virginia Modeling Analysis
and Simulation Center
Old Dominion University
1030 University Blvd
Suffolk, VA, USA
gdeteyou@odu.edu

ABSTRACT

The paper focuses on the critical node identification which can be used to rank the nodes belong to the corporate and control system network of Bulk Power System (BPS). We are proposing here an MADM (Multiple Attribute Decision Making) based ranking algorithm using a multi-layered directed acyclic graph (DAG) model to rank the critical nodes in the network. Ranking of the critical nodes can contribute to the resilience improvement process. Our proposed MVNRank (Multiple Vulnerability Node Rank) algorithm takes into account the exploit and impact scores of vulnerabilities as quantified by CVSS (Common Vulnerability Scoring System) and the severity level of each vulnerability. The algorithm also takes into account the asset value, degree centrality and node's distance from the target network in the vulnerability graph. The pseudocode for the resilience improvement using the node ranking is provided and necessary simulation results are presented to justify the node ranking importance.

Keywords: Bulk Power System, Node Ranking Algorithm, Vulnerability Graph, Cyberattack

1 INTRODUCTION

For decades, researchers have examined the methods and processes to safeguard the critical infrastructures from cyberattacks and other malicious activities. Electric power system such as the substations, smart grids, and ICS are lucrative targets for cybercriminals. In December 2015, a cyberattack incident took place in the power grid of Ukraine which impacted 30 substations and resulted in the loss of electricity for nearly 230,000 people (Greenberg 2016). There is always need to harden the security measures of power system

network. One of the metrics to measure network strength against malicious activities is the resilience. In this paper, we are considering the ICS and the field location devices as the target network. The corporate and control network elements are being considered as the potential nodes to rank. We are using a directed acyclic graph (DAG) model and ranking the critical nodes in the corporate and control network. We don't rank the ICS nodes, but use them to calculate the potential power loss capability of the control center nodes using the assumed ICS connectivity to BPS. Thus this paper is more related to network security of the control and corporate network considering the power system network as the target. The resilience equation that is being used here is a work by Shetty et al. (Shetty, Kamdem, Krishnappa, and Nikol 2018). The resilience computation process is highly time-consuming as it considers all the possible attack paths from the source to the target node which increases in a multiplicative manner based on the number of nodes and vulnerabilities. Thus one of the goals of this paper is to reduce the simulation time to compute resilience. The proposed algorithm itself is not time-consuming. It can facilitate to achieve certain resilience value and provide a list of critical nodes that can be considered sequentially to improve the resilience of BPS.

The rest of the paper is organized as follows. Section 2 discusses some of the current node ranking techniques that have proposed for similar sort of scenarios and their applicability towards our goal to optimize resilience. Section 3 describes the factors that we have considered important to rank the critical nodes. Section 4 discusses simulation setup & results and finally, section 5 discusses our future plans.

2 REVIEW OF RELATED WORKS

Researchers have proposed numerous approaches to deal with the ranking of critical assets in a network. Kijsanayothin and Hewett (2013) have done exploit-based analysis using Markov computational process and ranked nodes in the attack model in order of their likelihoods of compromise. The authors have a similar work in extended form in (Hewett and Kijsanayothin (2015)). Li and Qiu (2012) have proposed an algorithm namely NodeRank which is based on state enumeration attack graphs where the rank values of the nodes show the likelihood of an intruder reaching the states. Both the works have considered most exploitable vulnerability instead of the multiple vulnerabilities possessed by a host. The point missing in those analyses is that a highly exploitable vulnerability may have least impact on the network if successfully exploited and vice versa. Yang et al. (2010) have proposed DBRank for ranking vulnerabilities to patch in computing networks. DBRank prioritizes vulnerabilities based on the diffusibility and benefit of vulnerability exploitation. We have also considered the benefits of the attacker in terms of the adverse impact that can be caused by successfully exploiting a vulnerability. Again, what is missing in that analysis is only ranking the vulnerability individually is not giving a comprehensive idea about the network security status; because an attacker may compromise several vulnerabilities while exploiting a target. Mehta et al. (2006) have proposed a way of ranking attack graphs where the authors have considered nodes as system states and edges as transitions between states. Our approach of node ranking is different from most of the works in the sense that we have taken into consideration the number of vulnerabilities each node possesses; because we never know what vulnerability combinations an attacker may use on its way to exploit a target. Considering only the most exploitable vulnerability may result in producing partial analysis because of the lack of consideration of impact.

3 MVNRANK ALGORITHM

MVNRank considers all the intermediate nodes in the paths from source to target because an attacker can launch a multi-stage multi-host attack to compromise the security of the ICS system. The factors to formulate MVNRank are presented in the subsections below.

3.1 Asset Value in the Vulnerability Graph

By Asset value, we refer to the importance of the files and data stored in a host or server. This is directly related to how much damage an attacker can make to a network by compromising the asset. For example, if an attacker can compromise a database server; the damage he can make to the network is much greater than the damage by compromising a workstation. Our DAG model network topology is based on the NIST 800-82 CSSP defense-in-depth architecture (Stouffer, Falco, and Scarfone 2013). In this topology, the nodes in control LAN layer are more critical in terms of asset value than the nodes in corporate DMZ layer. If an attacker can penetrate the control LAN system, he would have more access to the application of the power station network consisting of the Remote Terminal Units (RTU), programmable logic controllers (PLC) and other ICS. For determining the asset value of a node, we have considered the concepts similar to PageRank centrality in graph theory. The equation that we have derived for asset value calculation is discussed in the next paragraph.

Let us consider, a graph is denoted as $G(V,E)$, where V is a set of vertices or nodes and E is the set of edges between nodes. Let IN be the set of all the intermediate nodes that fall in the possible paths from the source s to the target t. V is defined as $V = \{v_1, v_2, v_3..., v_{n-2}, v_{n-1}, v_n\}$. If there exist paths from source v_1 to target v_n which pass through all the other nodes, then we can say, intermediate nodes set, $IN = \{v_2, v_3, ..., v_{n-1}\}$ where $v_1 = s \,\&\, v_n = t$. There are some nodes that may not fall in the paths from the specific source node to the target node, hence, $IN \subset V$. The asset value of node i, A_{v_i} is given below in the equation (1):

$$A_{v_i} = N_{PR}(v_i) + \sum_{\forall v_j \in IN(v_i)} \sum_{\forall v_k \in PR(v_j)} \frac{1}{1 + D_{(v_k, v_i)}} \times N_{PR}(v_j) \tag{1}$$

Here, $IN(v_i)$ is the set of all the intermediate nodes through which node v_i is reachable from the source node v_1, $PR(v_j)$ is the set of immediate predecessor nodes of v_j and $N_{PR}(v_j)$ is the number of immediate predecessors of node v_j.

3.2 Reachability and Degree Centratlity

One of the crucial factors to consider for the node ranking is how far or close the attack launch node from the target node. This refers to the attacker's current position in the network and gives an idea about the reachability to the target node. In the graph theory, geodesic distance is being used as a measure of the shortest path length between two nodes. Eccentricity can be thought of as how far a node is from the most distant node to it in the graph. The shortest path distance between two nodes is considered as a measure of the reachability of the nodes. Rather than using the term "reachability", we are using the term "closeness" which we take as the inverse of the shortest path distance between two nodes. For node v_i, distance D_i is equal to the number of intermediate nodes in the shortest path to pass through to reach the target t. Thus, distance D_i=$d(v_i,t)$, where $d(v_i,t)$ is the shortest path length between node v_i and target t using the breadth-first search algorithm. Closeness d_i is defined by the following equation:

$$d_i = \frac{1}{D_i} \tag{2}$$

The degree centrality of a node is the fraction of nodes connected to it. It usually refers to the number of links incident upon a node or the number of connections the node has in the graph. We are considering the degree centrality to distinguish the importance of the firewall nodes in the network because most of the

nodes are communicating between different layers through the firewalls. Degree centrality of node v_i is termed as $C_d(i)$=$deg(v_i)$.

3.3 Exploit and Impact Score

Each vulnerability is associated with CVSS base metrics (Mell et al. 2007) which are access vector (A_V), access complexity (A_C), access authentication (A_A), confidentiality impact (I_C), integrity impact (I_I) and availability impact (I_A). The quantification of those base metrics are given in Table 1 and Table 2.

Table 1: CVSS Exploit base metrics.

Access Vector (A_V)	Local (L)	0.395
	Adjacent Network (A)	0.646
	Remote Network (N)	1.0
Access Complexity (A_C)	High (H)	0.35
	Medium (M)	0.61
	Low (L)	0.71
Access Authentication (A_A)	Multiple (M)	0.45
	Single (S)	0.56
	None (N)	0.704

Table 2: CVSS Impact base metrics.

Confidentiality Impact (I_C)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660
Integrity Impact (I_I)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660
Availability Impact (I_A)	None (N)	0.0
	Partial (P)	0.275
	Complete (C)	0.660

Each vulnerability represents an edge in the graph with weights. Edge weights are the exploitability and impact as calculated from the CVSS. Edge exploitability and impacts are the most important quantitative parameters that we have in the quantification of the resilience of a network. The exploitability and impact have been calculated in CVSS by the below equations (Mell et al. 2007):

$$ExploitabilityScore, ES = 20 \times AccessVector \times AccessComplexity \times AccessAuthentication$$
$$ImpactScore, IS = 10.41 \times (1 - ((1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImpact)))$$

Both exploit score and impact score are in between 0 to 10. To combine the effects of multiple vulnerabilities, we have used the weighted average (WA) method. And to use WA, we have scaled down the ES and IS to 1 and termed them as exploit factor (EF) and impact factor (IF).

$$\left.\begin{aligned} EF &= \frac{ES}{10} \\ IF &= \frac{IS}{10} \end{aligned}\right\} \tag{3}$$

Different vulnerabilities have different exploit and impact effect on the network. Treating each vulnerability on the same scale would give us an inappropriate result. To combine their effects, we have come up with a weighted average method. Table 3 shows that highly severe vulnerabilities have a weight of 0.5, medium severe vulnerabilities have a weight of 0.3, and low severe vulnerabilities have a weight of 0.2. The summation of this three-category weights is equal to 1.0 as needed.

We use the following two weighted average method formulae to consider the combined effect of the severity levels. The node v_i, has vulnerabilities from 1 to n. Let, EF_i and IF_i are the array of all associated exploit factor and impact factor for all the vulnerabilities of node v_i and they are being defined as:

Table 3: Exploit and impact factor weight.

EF	Severity	Weight	IF	Severity	Weight
0.7~1.0	High	0.5	0.7~1.0	High	0.5
0.4~0.69	Medium	0.3	0.4~0.69	Medium	0.3
0.0~0.39	Low	0.2	0.0~0.39	Low	0.2

$$\begin{cases} EF_i = [EF_{i_1}, EF_{i_2}, EF_{i_3}, ..., EF_{i_n}] \\ IF_i = [IF_{i_1}, IF_{i_2}, IF_{i_3}, ..., IF_{i_n}] \end{cases}$$

Weighted exploitability factor EF_{w_i} of node v_i having n vulnerabilities is calculated as below:

$$EF_{w_i} = \frac{\sum_{j=1}^{n} w_{i_j} \times EF_{i_j}}{\sum_{j=1}^{n} w_{i_j}} \tag{4}$$

Weighted impact factor IF_{w_i} of node v_i having n vulnerabilities is calculated as below:

$$IF_{w_i} = \frac{\sum_{k=1}^{n} w_{i_k} \times IF_{i_k}}{\sum_{k=1}^{n} w_{i_k}} \tag{5}$$

Both EF_{w_i} and IF_{w_i} have a value within 0 to 1.

3.4 Physical Impact Capability

In the section 3.3, we have considered the impact factor based on the software vulnerability. But the same vulnerability may cause different impact when it belongs to the corporate DMZ layer than the control system layer. Mostly in the power system domain, the physical loss is quantified by the amount of power outage that can be caused by the attacker by exploiting a vulnerability. That is why it is important to consider the physical impact of the vulnerability. In the power system domain, people are mostly concerned about the availability impact and integrity impact rather than the confidentiality impact.

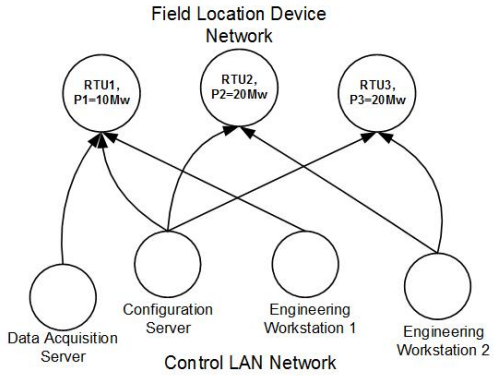

Figure 1: Sample network connectivity between control system LAN and field device network for equation demonstration.

Now, let us consider the logical connections among different hosts in control layer and field location devices. For example, the application software, already installed in the configuration server to operate the RTU's remotely, is a logical connection between the server and the remote terminal unit. Based on the logical connections as we have assumed in the Figure 1, data acquisition server has logical connections with RTU1 and RTU1 is controlling 10 MW power distribution unit. Configuration server is logically connected to all three RTU's, engineering workstation 1 is logically connected to RTU1 only and engineering workstation 2 is logically connected to RTU2 and RTU3. RTU2 and RTU3 are controlling each 20 MW power distribution unit. If RTU1 is being compromised, there is the chance of losing 10 MW power. Similarly, if RTU2 and RTU3 are being compromised, then 20 MW of power is possible to be made unavailable by the attacker. So each RTU has a fraction of power loss capability. If data acquisition server is being compromised then by compromising RTU1 a total of 10/(20+20+10) or 20% of the power can be taken out of service. Thus a network element in control System LAN network can have a physical power loss capability ranging from 0 (no power outage) to 1 (maximum power outage). So, the expected fractional power loss can be defined as below:

$$Expected\ power\ loss\ factor, EPLF_i = \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \tag{6}$$

Here, B_{im} is a binary quantity (1,0) which means whether node v_i is having an application or software installed that can access the RTU m i.e., if node v_i has logical connection to reach RTU m. If the application is in place already, there is a logical connection between node v_i and RTU m and then $B_{im} = 1$, otherwise $B_{im} = 0$. $\sum_{m=1}^{U} P_m$ is the total power delivery capacity of the substation by all the generator units. $\sum_{m=1}^{u} P_m$ is the summation of the power for all generators controlled by RTU m only, U is the total number of RTU's available in the physical power system network.

Using the integrity impact and availability impact, the impact factor is being modified for the case of physical power loss as below:

$$IF_{PL} = 1.1307 \times (1 - (1 - I_I) \times (1 - I_A))) \tag{7}$$

The factor 1.1307 comes from the calculation to make IF_{PL} equal to 1.0 by excluding the effect of confidentiality impact. The maximum value of I_I and I_A is 0.660. Considering the physical power loss and using the same weighted average approach as before as in Table 3, we can compute the weighted average power loss impact factor for node v_i as below:

$$IF_{PLw_i} = EPLF_i \times \frac{\sum_{l=1}^{n} w_{i_l} \times IF_{PL_{il}}}{\sum_{l=1}^{n} w_{i_l}} \tag{8}$$

or,

$$IF_{PLw_i} = \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \times \frac{\sum_{l=1}^{n} w_{i_l} \times IF_{PL_{il}}}{\sum_{l=1}^{n} w_{i_l}} \tag{9}$$

3.5 MVNRank Algorithm Formula

By combining all the factors discussed in the previous sections, we can derive the node ranking (R_i) value of node v_i by using the below equation:

$$R_i = A_{v_i} \times d_i \times C_d(i) \times N_i \times EF_{w_i} \times (IF_{w_i} + IF_{PLw_i}) \tag{10}$$

or,

$$R_i = A_{v_i} \times d_i \times C_d(i) \times N_i \times \frac{\sum_{j=1}^{N_i} w_{i_j} \times EF_{i_j}}{\sum_{j=1}^{N_i} w_{i_j}} \times \left[\frac{\sum_{k=1}^{N_i} w_{i_k} \times IF_{i_k}}{\sum_{k=1}^{N_i} w_{i_k}} + \frac{\sum_{m=1}^{u} B_{im} P_m}{\sum_{m=1}^{U} P_m} \times \frac{\sum_{l=1}^{N_i} w_{i_l} \times IF_{PL_{il}}}{\sum_{l=1}^{N_i} w_{i_l}} \right] \tag{11}$$

Here, N_i is the total number of vulnerabilities of node v_i. For the IT domain network except for the control system LAN, the nodes would not have the second impact factor component in equation (11), because they don't have any potential power loss, they have a 0 value for the IF_{PL_i}. Only the control system LAN network nodes have this physical impact factor, so those nodes have non zero $IF_{PL_{w_i}}$ which give them some priority over the other nodes. Maximum node rank value, $R_{max} = \max_{\forall i | \exists v_i \in IN(v_t)} [R_i]$, where $IN(v_t)$ is the set of all the intermediate nodes of target node v_t. Relative criticality is found by normalizing the node rank value. Thus, relative criticality (RC_i) of node v_i is found by dividing each R_i value by the R_{max}.

$$RC_i = \frac{R_i}{R_{max}} \tag{12}$$

Here, $0 \leq RC_i \leq 1$. Based on RC_i, the nodes are ranked and identified as the most important to the least important. Figure 2 shows the node ranking algorithm psuedocode.

4 SIMULATION RESULTS AND ANALYSIS

In this section, we present the simulation network setup and the results & analysis.

4.1 Simulation Setup

Figure 3 shows the network that we have considered for our simulation setup & design. We have used different product models corresponding to their functionality. We have considered last three years vulnerabilities from the NVD database for simulation purposes. We have used python NetworkX module and a product vulnerability database to simulate the results. No real network setup has done for this experiment. The work is based on the simulation only.

4.2 Simulation Output and Analysis

One of the major concerns of the simulation of this type of network is that each node may have millions of incoming paths combinations from the original entry point. For example, in our simulation Email Server1 has only 30 incoming paths, but Security Server1 has 129600 incoming paths which are the combination of different exploitable vulnerabilities from the origin node. Again, Control Firewall2 has 8294400 (8.29M) paths and Application Server1 has 248832000 (248.8M) paths. The more the total number of paths, the more it takes to simulate the resilience of that node. In Figure 5, we have found that the resilience computation time is sharply increased when the number of paths are increased. While Figure 4 shows the simulation

Algorithm 1 Resilience Improvement Pseudocode

```
1:  Input : G(N, e), Source, Target, R_Target
2:  procedure ResilienceImprove
3:      R_Node ← Value Calculate Using Resilience Formula
4:      IntermediateNodes ← List of Nodes between Source and Target
5:      i ← 0, R_max ← 0
6:      Node_len ← len(IntermediateNodes)

7:      Initialize: R[Node_len] = [ ], RC_NodeList[Node_len] = [ ], RC_dict = { }
8:      for i < Node_len:
9:          R[i] ← Value Calculate Using MVNRank
10:         RC_NodeList[i] ← IntermediateNodes[i]
11:         RC_dict ← {R[i] : RC_NodeList[i]}
12:         i ← i + 1
13:     end for
14:     R_max = max(R)
15:     i ← 0
16:     for i < Node_len:
17:         RC_dict ← {R[i]/R_max : RC_NodeList[i]}
18:     end for
19:     RankMatrix = sort(RC_dict)
20:     j ← 0
21:     for j < len(RankMatrix):
22:         do:
23:             Node = RankMatrix[j]
24:             Node_product ← least vulnerable product
25:             Node_vendor ← least vulnerable vendor
26:             G(N, e) ← New Vulnerability Graph
27:             Calculate R_Node
28:             print Node, R_Node
29:             j ← j + 1
30:         while (R_Node < R_Target)
31:     end for
32: end procedure
```

Figure 2: Resilience improvement pseudocode using node ranking.

time for resilience and paths computation upto number of paths 0.13M, Figure 5 shows the simulation time for the same upto 248.8M number of paths. Application Server1 has 248.8M paths and it takes 2537.342 seconds (42.23 min) to compute the resilience and 387.924 seconds (6.47 min) to compute the total number of paths in vulnerability graph. The simulation time comparison is given here, because one of our objectives of this ranking algorithm is to reduce the simulation time to reach to the target resilience value.

Figure 6 shows the paths reduction over simulation trials for the three selected nodes and Figure 7 shows the resilience value increment over simulation trials where nodes are being selected as per rank order. As after 3rd trials no paths exist from source to the target node, so the resilience reaches to 1.0. For the number of paths, we have used the log scale for proper demonstration and actual value is being shown in (x,y) co-ordinate format. Although after 3rd trials, the value of paths become 0, but due to the use of log scale we have taken it as 1.

Figure 8 and Figure 9 show a comparison of the required simulation time to reach to the resilience value of 1.0 for some of the selected nodes in control DMZ layer and control system LAN layer. Some nodes take less time to reach 1.0 value resilience and some other nodes take a long time to reach 1.0 resilience value based on their number of attack paths. The figures compare cumulative simulation time using the rank order

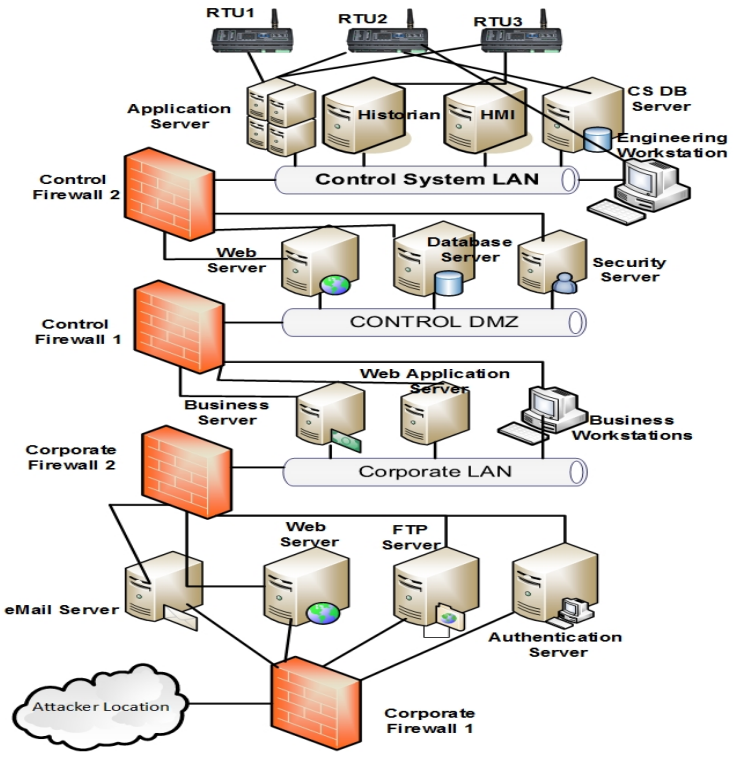

Figure 3: Simulation network setup.

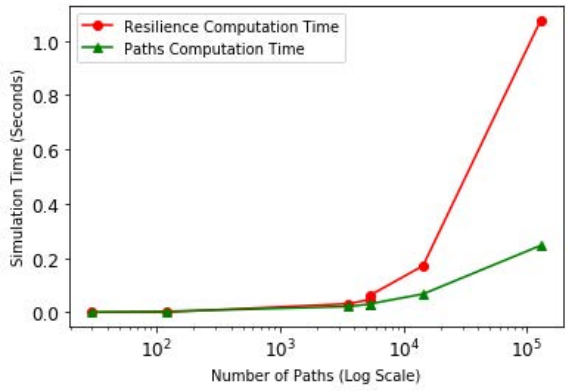

Figure 4: Simulation time (maximum attack paths number $\leq$ 129600).

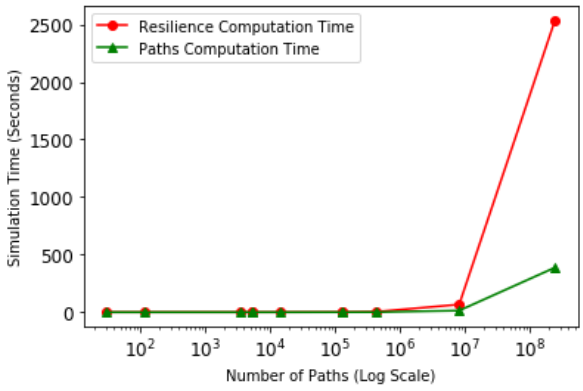

Figure 5: Simulation time (maximum attack paths number $\leq$ 248832000).

and without using the rank order. Figure 8 shows the simulation time of the 100 Monte Carlo simulation average and Figure 9 shows the simulation time of the 20 Monte Carlo simulation average . For both cases, it is found that the simulation time using the ranking order is reduced almost by half than using the random order.

To find the relationship between the simulation time using the rank order and using the random order, we have performed a regression analysis as shown in Figure 10. It is being found that the use of random orders take almost double time to compute the resilience than using the ranked order of nodes. It means the use of the ranked order can save nearly 50% of the simulation time.

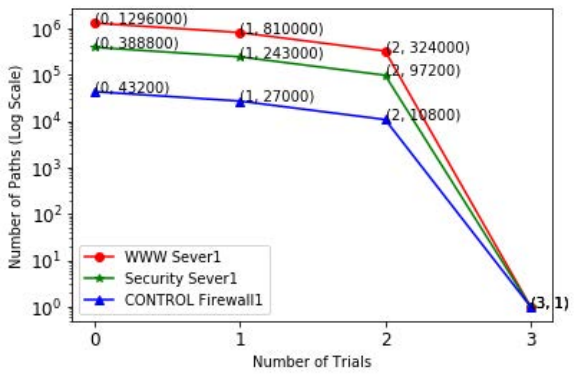

Figure 6: Number of attack paths reduction over simulation trials (for selected nodes in control DMZ layer).

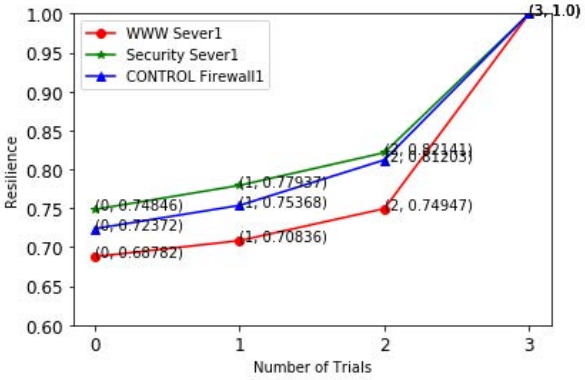

Figure 7: Resilience improvement over simulation trials (for selected nodes in control DMZ layer).

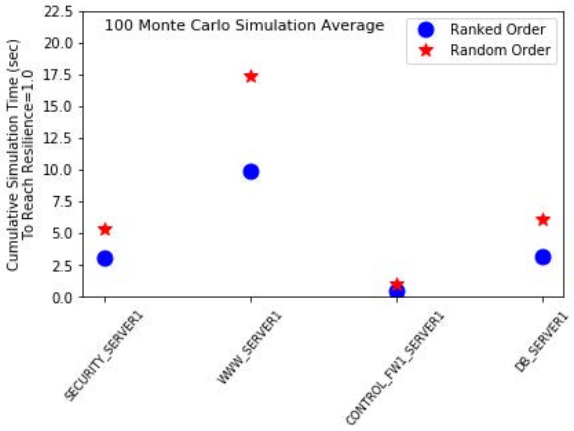

Figure 8: Comparison of cumulative simulation time (in sec) to reach 1.0 resilience value, nodes belong to control DMZ layer.

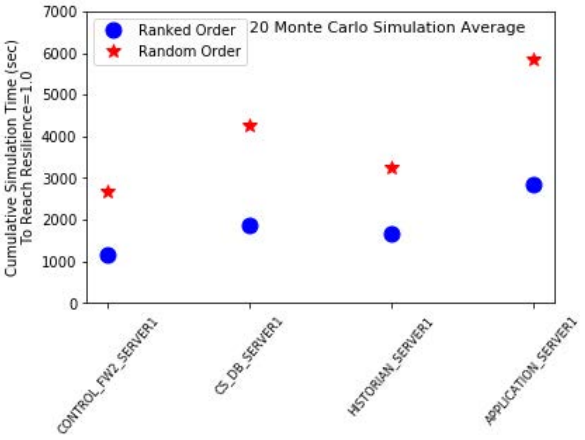

Figure 9: Comparison of cumulative simulation time (in sec) to reach 1.0 resilience value, nodes belong to control system LAN layer.

There may be questions regarding the ranking computation time. In Figure 11 we have presented the simulation time required to compute the rank and the resilience to demonstrate a comparison between these two timing requirements. Here, Y-axis is in log scale. The figure shows that the rank computation time is much less than the resilience computation time for the same node. We have done the simulation for nodes starting from 20 to 200 with an increment of 20 nodes in each time. This time, we have considered only 1 vulnerability between two nodes throughout the network. This is used to demonstrate the scalability of the ranking algorithm.

5 LIMITATIONS AND FUTURE WORK

In this work, we didn't consider the firewall policies, we have only considered the product vulnerabilities. Also, we didn't consider the physical bus systems to compute the power damage capability of each RTU. In future, we have plans to include the firewall policies and the physical bus system based calculation of potential power damage of RTU to resemble real & robust cyber-physical network scenarios.

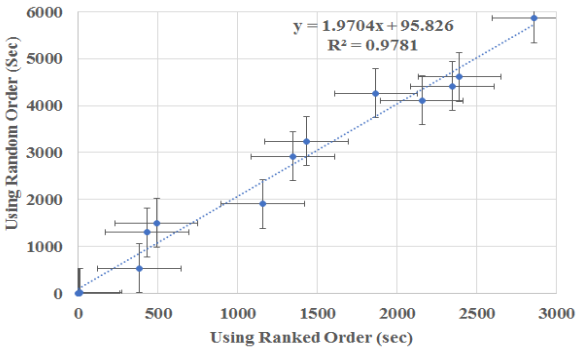

Figure 10: Regression analysis of required simulation time by using rank order and random order.

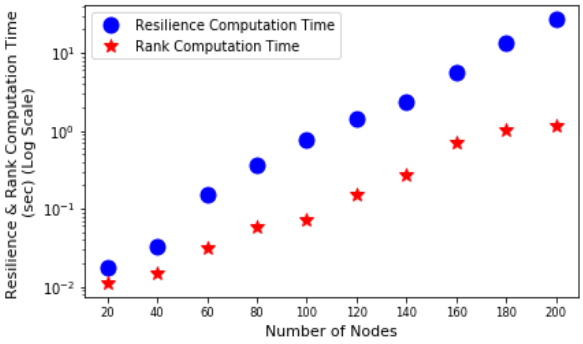

Figure 11: Comparison of resilience computation time and ranking computation time using different number of network nodes.

6 CONCLUSION

Improving resilience is not a straightforward process. It accounts for the combination of the effects by changing different parameters of the intermediate nodes as can be ranked by MVNRank. The factors we have considered in calculating the criticality are important in network security analysis perspective. As the network grows larger, the computation time to improve the resilience can be very high. Using the ranking algorithm, the resilience computation time can be reduced by almost half. MVNRank considered the possibility of potential electric power loss that the attacker can cause to the network, which distinguishes the importance between traditional corporate network elements and control system elements. This node ranking algorithm can be useful in bulk power system network cyber resilience optimization process.

ACKNOWLEDGMENT

This paper is based upon work supported by the Department of Energy under Award Number DE-OE0000780.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

APPENDIX

This paper doesn't derive the equation for the resilience computation. We have used the resilience equation from the work by (Shetty, Kamdem, Krishnappa, and Nikol 2018) (under publication). The equation is given

below:

$$R = \frac{1}{|E|} \times \sum_{e \in E} \frac{c(e)}{c_{max}} \times \left[\frac{1}{T} \int_0^T K_e(t) dt \right] \tag{13}$$

where, E is the set of all possible attack paths from source to target, $|E|$ is the cardinality of E, $c(e)$ is the exploitable path cost, c_{max} is the upper limit of the path exploit costs and $K_e(t)$ is the critical functionality over time.

REFERENCES

Greenberg, A. 2016. "'Crash Override': The Malware That Took Down a Power Grid". *Wired*.

Hewett, R., and P. Kijsanayothin. 2015. "Ranking Intrusion Likelihoods with Exploitability of Network Vulnerabilities in a Large-Scale Attack Model.". *IJ Network Security* vol. 17 (4), pp. 383–394.

Kijsanayothin, P., and R. Hewett. 2013. "Exploit-based analysis of attack models". In *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, pp. 183–186. IEEE.

Li, P., and X. Qiu. 2012. "NodeRank: An Algorithm to Assess State Enumeration Attack Graphs". In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on*, pp. 1–5. IEEE.

Mehta, V., C. Bartzis, H. Zhu, E. Clarke, and J. Wing. 2006. "Ranking attack graphs". In *RAID*, Volume 4219, pp. 127–144. Springer.

Mell, P., K. Scarfone, and S. Romanosky. 2007. "A complete guide to the common vulnerability scoring system version 2.0". In *Published by FIRST-Forum of Incident Response and Security Teams*, Volume 1, pp. 23.

Shetty, S., G. Kamdem, B. Krishnappa, and D. M. Nikol. 2018. "Cyber Resilience Metrics for Bulk Power System". Risk Analysis Journal.

Stouffer, K., J. Falco, and K. Scarfone. 2013. "Guide to industrial control systems (ICS) security". *NIST Special Publication* vol. 800, pp. 82.

Yang, X., S. Shunhong, and L. Yuliang. 2010. "Vulnerability ranking based on exploitation and defense graph". In *Information Networking and Automation (ICINA), 2010 International Conference on*, Volume 1, pp. V1–163. IEEE.

AUTHOR BIOGRAPHIES

MD ARIFUL HAQUE is an MS student in the Department of Modeling, Simulation and Visualization Engineering at Old Dominion University and Graduate Research Assistant in the Cyber Security Lab in VMASC. His email address is mhaqu001@odu.edu.

SACHIN SHETTY is an Associate Professor in the in the Department of Modeling, Simulation and Visualization Engineering at Old Dominion University. He has authored and coauthored over 80 research articles in journals and conference proceedings and two books. His email address is sshetty@odu.edu.

GAEL KAMDEM is a postdoctoral researcher at Virginia Modeling Analysis and Simulation Center (VMASC). His email address is gdeteyou@odu.edu.