Energy Smart Grid Cyber-Threat Exposure Analysis and Evaluation Framework

Shahir Majed
Advance Informatics School,
Universiti Teknologi Malaysia,
Shahir.majed@mimos.my

Suhaimi Ibrahim
Advance Informatics School,
Universiti Teknologi Malaysia,
suhaimiibrahim@utm.my

Mohamed Shaaban
Centre of Electrical Energy Systems
Universiti Teknologi Malaysia,
m.shaaban@fke.utm.my

ABSTRACT

This paper introduces a framework for evaluating the threat exposure of a large scale smart grid environment. The framework utilizes a model based on access graphs to determine the system's attack exposure. This method also implements quantitative metrics to evaluate acceptable exposure levels. This evaluation framework also introduces a method to analyze the impact a mitigating security control has on the resulting architecture.

Categories Subject Descriptors
[C.2.0] **Computer-Communication Networks:** Security and Protection
[C.2.1] **Network Architecture and Design:** Distributed Networks
[D.4.6] **Security and Protection:** Access Control
[K.6.5] **Security and Protection:** Physical Security

General Terms
Design, experimentation, performance

Keywords
Vulnerability Assessment, Smart Grid Security, Security Metrics, Critical Infrastructure Protection.

1. INTRODUCTION

The U.S. Department of Energy (DOE) states that attack resilience is a primary requirement of a modern grid[11]. While the smart grid provides considerable benefits to both electricity producers and consumers, the tight coupling of demand response and customer usage information with electrical production produces significant security concerns. Smart grid devices will also spread throughout diverse environments which substantially increases attack exposure. The communications architecture for a large grid implementation also introduces security concerns. This environment wills likely use RF mesh, powerline network, and even cellular communication to support the large number smart meters [9]. The wide array in security features supported by these different communication mediums introduces variable exposure to critical data. In order for system designers to make intelligent decisions based on the security of the smart grid implementations we propose a model which can be used to display trust issues and potential vulnerabilities throughout a large distributed system. This model will detail architecture's resilience to attack by evaluating the required effort by an attacker to access critical resources. In addition, we implement metrics that provide a clear evaluating of the system's security posture.

2. RELATED WORK

There has been little previous work evaluating the exposure of a smart grid architecture. While attack trees and graphs have commonly been used to relay security concern with other systems, they do not span will to a smart grid environment.

1) Attack Trees: An attack tree is a model which enumerates all potential vectors an attacker could use to gain access to some target resource. Each branch in the tree represents a set of intermediate steps the attacker must take prior to gaining access to the target. This produces a useful model to determine various attack vectors and provides an obvious understanding of the steps required to exploit those vectors. While attack trees have been very useful in the understanding the security of many systems, the development of accurate trees is a difficult process. Specifically, tree creation entails the enumeration all particular attack vectors that may provide system access. Unfortunately, the required effort to produce an attack tree on smart grid architecture is intractable. Attack trees also do not accurately model the vulnerabilities in large architectures. Since attack trees can only represent one target goal of an attack,

iiWAS '14, December 04 - 06 2014, Hanoi, Viet Nam Copyright 2014 ACM 978-1-4503-30015/14/12‥$15.00

http://dx.doi.org/10.1145/2684200.2684308

the model does not scale well to a system where different attackers could target different system components. In the smart grid, the number of potential targets could differ based on the attackers' intention. For example, an identity thief might focus on systems hosting privacy data while a more hostile adversary may focus on control system access. The attack tree paradigm would require the development of different models for each potential attack situation.

2) Attack Graphs: Attack graphs take a different approach to modeling security concerns by presenting system vulnerabilities as nodes and using paths to represent the exploitation of a particular vulnerability. Since each node represents vulnerability, the general security of a system can be expressed as the number of vulnerabilities which must be exploited to obtain access to some target resource. While a branch from an attack tree details all required actions that an attacker must perform to gain access to some system, an attack graph typically only displays nodes for known vulnerabilities. Therefore, any errors in the vulnerability assessment process will likely cause inaccuracies in the graph. While an attack graph provides a detailed example of how an attacker can gain access to a system, determining system vulnerabilities is a difficult task. In addition, determining how vulnerabilities are exploited requires substantial experience in computer attacks. This makes the development of an accurate attack graph difficult for individuals without an indepth knowledge of system security. Also, like trees, graphs also model a system with a predetermined end goal for an attacker. While this works for a small system, it will not scale to a system which could have different targets depending on an attacker's goal.

3) Access Graphs: An access graph differs from attack trees and graphs as they primarily use the trust shared between systems to model security properties. The work done by both Xiao[1] and Ammann[2] creates a graph where edges represent some trust level between systems. This concept provides an useful understanding of system resources and the risk they inherent from potentially compromised related systems. Here the term trust is used to abstract some level of system access based on some pre-condition such as having some access to another system. This graph then assumes that access to another trust can either be obtained through an already established trust relationship or through some exploited vulnerability. While the idea of a access graph based provides a unique way to understand security relationships, the model proposed by Xiao is still heavily dependent on the enumeration of system vulnerabilities and their exploitability. However, with the prevalence of zero day exploits, system vulnerability information can rarely be considered complete [15].

3. SMART GRID ARCHITECTURE

Simple smart grid architecture is proposed for the basis of our analysis. Figure 1 displays this sample smart grid environment. In this figure we have some number of Home Area Networks (HANs), Business Area Networks (BANs), and Industrial Area Networks (IANs) connected to AMI meters. These AMI meters monitor the energy usage of the HAN/BAN/IANs and provide demand response to them. The AMI meters communicate to a AMI headend device which aggregates the information from a large number of AMI devices. Communication of the AMI data is supported by some field area network (FAN). The FAN can utilize RF mesh, powerline broadband and/or cellular technology for the transmission media. The AMI head-end then relays the data to the Meter Data Management System (MDMS) which provides primary control over the AMI architecture. It is likely that these systems will reside on some wide area network (WAN) which may have some access to SCADA control systems. Finally we have some web portal system which provides easy access to account information for smart grid users.

4. THREAT EXPOSURE ANALYSIS AND EVALUATION FRAMEWORK

The development of a secure infrastructure is contingent on the ability to accurately assess current system exposure levels and measure whether a specific security control can reduce the exposure to an acceptable level. This framework introduces a Exposure Enhancement

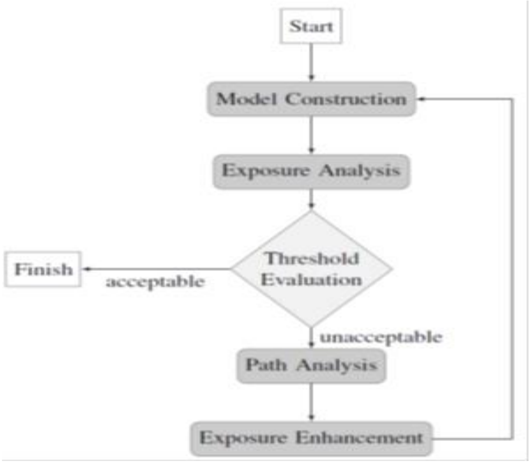

Fig. 2. Smart Grid Threat Exposure Analysis and Evaluation Framework

model to represent potential risk in an architecture and metrics to understand that risk. This paper utilizes an access graph approach to model the smart grid environment. This model is then analyzed to determine the exposure of key system components. The results of this analysis are compared against some configurable threshold to determine if the architecture meets some predetermined security requirement. Finally a framework for analyzing potential security improvements is proposed. Figure 2 introduces a flow chart illustrating the framework.

A. Layered Access Graph Approach

The layered approach to access graph implementation expands the flexibility of the model and also provides the ability to represent various security properties of the network. The layers used in this model are described below

1) Physical Layer - models the physical architecture as a traditional network graph
2) Component Layer - separates physical assets into individual components and models data flow between components
3) Security Layer - introduces specific security features of the components and edges to model system risk

1) **Physical Layer:** The physical layer model directly represents the physical layout of a network architecture. We develop a graph, G = (V, E), where V represents the different hosts and E represents the communication links in the architecture. This layer provides a model of how data flows between systems and how security impacts to WAN one host may affect another. This layer also displays the physical communications links which may be vulnerable to eavesdropping, man in the middle and jamming attacks. Below we display graph representing three systems and two communication links connecting them.

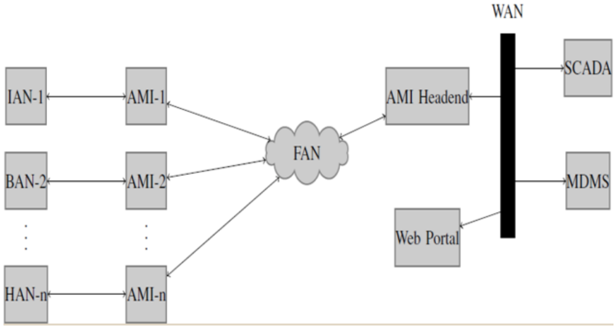

Fig. 3. Example Smart Grid Architecture

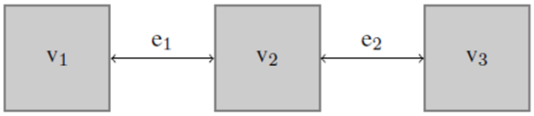

Fig. 3. Physical Layer Graph, G = (V, E)

We could assume Figure 1 represents a very simple environment with the following node and edge definitions.

<u>Layer-1 nodes and edges</u>

v1	HAN
v2	AMI meter
v3	MDMS
e1	802.11 link
e2	dedicated link

The physical layer graph provides a representation of the physical architecture but does not help understand any potential security concerns. The transition to the trust layer graph will begin model security properties.

2) Component Layer: Similar to the work done by Xiao, we also model trust relationships between hosts. Here we define a trust relation as a tuple < host, privilege, interface > where host is the host of the relationship, privilege is some system privilege level and interface is some method which other hosts use to interact with the system. Typically will typically be some network protocol, but might also be some inter-process communication method. For example, an AMI device may have two distinct interfaces, one which interacts directly with some HAN, one which interacts with the MDMS. These two components could either run under the same shared privilege or have a unique privilege levels specially tailored to their needs. This model relies on this relationship between systems privileges and components to model the robustness of system architectures. For example if two components share the same privilege level and one component is compromised, the attacker will also gain access to the next component. However, if these two components each have distinct privilege levels then one exploited component will not provide access to another. Access the second component would likely require that an attacker perform a second attack. We use this idea to separate hosts into various components and use those components to model the flow of information through the system.

In this model all component level communication must be performed through a logical connection. A logical connection can either be represented by a physical network between components on different hosts or as inter-process communication between components on the same host. Below we provide a graph H = (C, L), where C represents all components and L represents all logical connections. Figure 4 provides an example of a component layer graph.

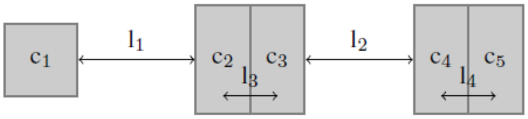

Fig.4. Component Layer Graph, H = (C, L)

The component layer in this example splits the nodes n2 and n3 into multiple components and models the communication between those components with new logical connections. The table below provides a description of all components and logical connections in *H* .

Layer-2 components and logical connections

c1	HAN
c2	smart meter(HAN interface)
c3	smart meter(management interface)
c4	MDMS system
c5	MDMS backend database
l1	network connection between c 1 , c2
l2	network connection between c 3 , c4
l3	inter-process communication between c 2 , c3
l4	inter-process communication between c 3 , c4

3) Security Layer: The security layer integrates properties of a network which can be useful in modeling its security posture. We can then use this final graph to perform analysis of the architecture to understand how resilient the architecture is to attack. The primary method to model the security is through the use of edge weights. Edge weights present a cost of traversing a particular edge. We will use weights to represent the difficulty or cost to an attacker that the exploitation of some node or edge presents. In Figure 5 we provide an example security layer graph which has weights to all edges. This figure shows that the logical connection between c2 and c3 has a weight of l3 which represents the difficulty faced by an attacker to span this connection though some privilege escalation attack.

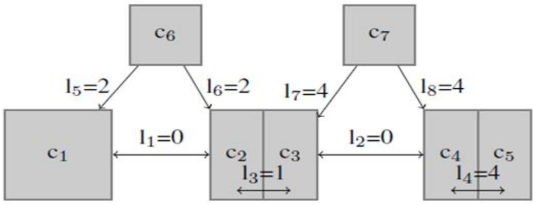

Fig.5. Security Layer Graph, H = (C, L)

The next step in this layer is to create new source edges. A new source edge should be created for each physical edge in the network. These source edges will be used to represent the level of effort an attacker must impose to gain access to the physical link. We connect this node through directional edges which only allow flow out of these source nodes. In Figure5 there are two new source nodes, c 6 and c7 , one for the edge (c1 , c2) and one for the edge (c 3 ,c4). In this example the weights for edges (c6, c1) and (c6 , c2) represent the level of effort required to compromise that edge. Since compromising an edge will provide the attacker with any data traversing it, we can assume it would provide access to those components connected to an edge. For example, if (c 3,c4) is a encrypted wireless channel which uses strong encryption it will have a large weight associated with it. However, if an attacker can compromise the wireless link, the data on the link will provide them with access to both the c1 and c2 component.

Since all edge weights should be assigned based on attack difficulty. A weight assignment function should be utilize security relevant features to weight edges appropriately. For link weight assignment we devise a touple *S*l =< *medium, encryptionstrength, keystrength* >. In this weight model wired connections are considered more secure and are provided a higher weight. We also provide higher weights based on encryption strength and key lenght. The security layer graph in figure 5 utilizes the following weights.

- weight(wired, NA, NA) = 4
- weight(wireless, WPA, 128-bit) = 2
- weight(wireless, WEP, 128-bit) = 1

Similar weights are devised to model component security. Since most components already have some access, weights definition is based heavily on component privilege level. If component ci is running with high privileges and is exploited, an attacker will gain those high privileges which can be used to gain access to other components on that system. However, if that component is running with low privileges, the attacker will be forced to perform another privilege escalation attack to gain high level privileges. Since we are requiring to the attacker to perform two different attack we can assume this is at least twice as hard as preforming one. We present a tuple for component security as S_c =< *privilege, shared-components, enforcement* > and provide weights below.

- weight(service, none, virtual machine) = 4
- weight(service, none , OS privileges) = 3
- weight(admin, ci, OS privileges) = 2
- weight(admin, ci, web application) = 1

model. From the graph we can see that an attacker starting at c1 can access c5 with a weight of 5, while the attacker could also start from c6 or c7 , the resulting weight is higher. Figure 6 provides an example of a security layer graph for our example smart grid architecture.

5. EXPOSURE ANALYSIS

The security layer graph in figure 6 models the level of difficulty for an attacker to reach various spots on the networks. We utilize this model to compute metrics which detail the security exposure of the system. This is then evaluated against a security threshold to determine whether the architecture meets a required security level.

A. Exposure Determination

Previous work on attack graphs use shortest path analysis to determine the easiest path an attacker can follow to access some target[6]. This paper uses the shortest path algorithm in a similar manner. Since the security level graph presents paths an attacker can take to traverse through the architecture, we use the shortest path problem to determine the level of effort an attacker must expend to access a particular critical resource. Unlike the attack graph shortest path problem, the security layer graph allows the computation of path weights from may potential sources and destination nodes.

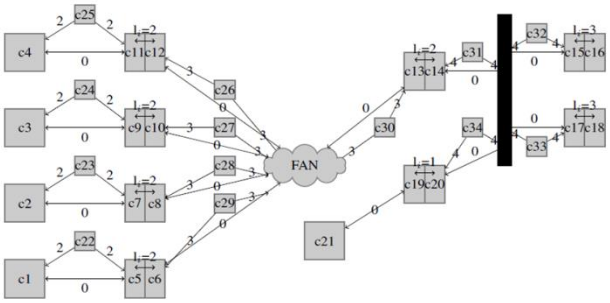

Fig.6. CPS Security layer graph for the example smart grid architecture

```
Input S : Set All Source node
Input T : Set All Critical resource node
    begin :
    $current\_weight \Leftarrow \infty$ ;
    for each $t \in T$  do
        for each $s \in S$ do
            $new\_weight \Leftarrow$ BellmanFord(s, t) ;
            if $j < i$ then
              $current\_weight \leftarrow new\_weight$ ;;
            end
        end
    end
```

Algorithm 1: Determine shortest path to critical resources

This algorithm utilizes Bellman-Ford to solve the single source shortest path problem. While the Bellman-Ford algorithm has a runtime of O(V E)[13], our algorithm must computer must be run for |T | target nodes. Therefore, the actual runtime for the algorithm will be O(|V||E||T |).The shortest path problem computes the difficulty for an attacker to reach some critical resource. The set $T = \{t_1, t_2, ..., t_k\}$ represents all the critical assets in thesmart grid environment. Set $S = \{s_1, s_2, ..., s_l\}$ represents all potential attacker entry points. The shortest path algorithm returns an exposure level E(ti) for each (t_i, s_j) pair. This exposure level represents the easiest vector an attack could use to access some resource. For some critical assets ti we can obtain the overall exposure $E(t_i)$. This requires finding the shortest of all the paths from *l* potential source nodes.

$$E(t_i) = min(E(t_i, s_j)), \; j = 1, \ldots., l \quad (1)$$

B. Threshold Evaluation

The exposure for each critical component should be compared to some configurable exposure threshold R. This threshold is considered the minimal acceptable exposure that any particular host should maintain. We use this threshold to compute λ which is defined as the total exposure of the entire architecture.

$$\lambda = \sum_{i=1}^{k} (R - E(t_i)) \quad (2)$$

The system exposure level, λ, remains acceptable as long as λ = 0. When λ > 0 the system must undergo a security enhancement to reduce exposure.

7. SECURITY ENHANCEMENT ANALYSIS

We perform security enhancement analysis based on the sample security layer graph proposed in figure 6. The shortest path algorithm is computed for sets *T* and S as defined below. The set *T* will contain the MDMS, SCADA component, and web portal since these are all critical assets in the smart grid. The set S contains the customer area networks, web client and wireless man in the middle sources.

T(sink)	*S*(source)
t1 = c16	s{1,...,4} = c{1,...,4}
t2 = c17	s{4,...,8} = c{22,...,26}
t3 = c20	s{9,...,12} = c{26−29} s13 = c21

Using the shortest path the exposure level from the various sources to targets is provided below.

Path Exposures		
Target	Source	Exposure
t_1	s{1,...,4}	7
	s{4,...,8}	9
	s{9,...,12}	8
	s13	4
t_2	s{1,...,4}	4
	s{4,...,8}	6
	s{9,...,12}	5
	s13	1
t_3	s{1,...,4}	4
	s{4,...,8}	6
	s{9,...,12}	5
	s13	1

In this example we assume that $R = 5$. Our example architecture would then have a total exposure of $T = 14$. This is a result of the short paths web portal and MDMS to the AMI meters and the web client.

A. Computing Security Enhancement

We can compute the benefit of some security improvement by recomputing path weights considering the improvement. This first involves finding all paths which used a path containing the improved component. After all involved paths are determined and the a security improvement has been selected the shortest paths for the critical hosts involved in those paths should be recomputed. Once this has occurred the difference between the previous exposure, $E(t_i)$, and the current exposure, $E(t_i)$, should be computed. This resulting value β represents comprehensive benefit of the enhancement.

$$\beta = \sum \ E'(t_i, s_j) - E(t_i, s_j) \ (3)$$

The benefit calculation provides the ability to compare the multiple different enhancement options. This allows for the selection of a security control which provides the maximum benefit to the architecture.

8. CONCLUSION

The implementation of a smart grid presents unknown risk to power systems. This paper proposes an efficient framework for modeling the cybersecurity exposure of smart grid architecture. The model specifically highlights the system's resilience to targeted cyber-attacks and heavy reliance on trust between components. The exposure level and acceptable threshold provide concrete metrics that can be used to evaluate the security posture and make efficient system improvements.

9. ACKNOWLEDGEMENT

This research is funded by the Research University grant of Universiti Teknologi Malaysia (UTM) under the Vot no. 03H74. The authors would like to thank the Research Management Centre of UTM and the Malaysian ministry of education for their support and cooperation including students and other individuals who are either directly or indirectly involved in this project.

10. REFERENCES

[1] Xiao, Xiaochun; Zhang, Tiange; Zhang, Gendu. Using Access Graph to Analyze Network Vulnerabilities IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008.

[2] Ammann, P; Pamula, J Using Access Graph to Analyze Network Vulnera- bilities Proceedings of the 21st Annaul Computer Security Applications Conference (ACSAC), 2005.

[3] C.-W. Ten, C.-C. Liu, and G. Manimaran. Vulnerability assessment of cybersecurity for SCADA systems using attack trees IEEE Power Engineering Society General Meeting , 2007.

[4] S. Harari, et al. Impact Analysis of Faults and Attacks in Large-ScaleNetworks IEEE Security & Privacy , 2003.

[5] Jha, S.; Wing, J.; Linger, R.; Longstaff, T. Survivability Analysis ofNetwork Specifications. IEEE Dependable Systems and Networks, 2000. [6] Lippman, R.P.; Ingols, K.W. An Annotated Review

of Past Papers onAttack Graphs. Project Report, Lincoln Laboratory, 2005.

[7] LeMay, M; et al. Unified Architecture for Large-Scale Attested Metering. Hawaiian International Conference on System Sciences, 2007.

[8] Davis, M. SmartGrid Device Security, Adventures in a new medium. Black Hat USA, 2009.

[9] Lee, A. Smart Grid Cyber Security Strategy and Requirements, Draft NISTIR 7628. National Institute for Standards and Technology, 2009. [10] AMI System Security Requirements. V1.01. AMI Security AccelerationProject (ASAP), 2008.

[11] A Systems View of the Modern Grid.U.S. Department of Energy (DOE) National Energy Technology Laboratory (NETL), 2007.

[12] Kleinberg, J; Tardos, E. Algorithm Design. Pearson, Addison Wesley,2006.

[13] Paul E. Black Bellman-Ford algorithm. Dictionary of Algorithms and Data Structures, National Institute for Standards and Technolog, 2005.

[14] Christey, S; Martin, R.Vulnerability Type Distributions in CVE. The MITRE Corporation, 2007. http://cwe.mitre.org/documents/vuln-trends/index.html.

[15] US-CERT Quarterly Trends and Analysis Report. Volumne 1 Issue 2. United State Computer Emergency Readiness Team (US-CERT). 2006