2017

Malware in smart grid

Altay Ozen
Iowa State University

Malware in smart grid

by

Altay Ozen

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering (Secure and Reliable Computing)

Program of Study Committee:
Zhenqiang Gong, Major Professor
Thomas E. Daniels
Ahmed E. Kamal

The student author and the program of study committee are solely responsible for the content of this thesis. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

RAS	Remedial Action Scheme
RASc	Remedial Action Scheme Controller
RTDS	Real Time Digital Simulator
RTU	Remote Terminal Unit
CC	Control Center
LAN	Local Area Network
WAN	Wide Area Network
OTC	Operational Transfer Capability
SE	State Estimation
AGC	Automatic Generation Control
WAMPAC	Wide Area Monitoring, Protection, and Control
SCADA	Supervisory Control and Data Acquisition
DOS	Denial of Service
CPS	Cyber Physical System
NERC	North American Electric Reliability Corporation

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Neil Zhenqiang Gong, and my committee members, Dr. Thomas E. Daniels, and Dr. Ahmed E. Kamal, for their guidance and support throughout the course of this research.

I would also like to thank Vivek Kumar Singh for doing this research with me, and his support.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience. I want to also offer my appreciation to those who were willing to participate in my surveys and observations, without whom, this thesis would not have been possible.

ABSTRACT

With the advancement in communication technology of Smart Grid, cyber-attacks are becoming the serious threat. Specifically, the vulnerabilities created due to the successful malware installation in smart grid is a very serious concern since it can be exploited to disable the system along with taking control or damaging the critical infrastructure permanently. The main idea behind this thesis is to explore the malware issue in the remedial action scheme (RAS), widely used for wide area protection, of smart grid. This thesis is concerned mainly on the cyber part of the Smart Grid. The main contribution of the work is divided into two major parts.

In the first part, we modelled the stealthy coordinated cyber-attack with a malware at its core. The purpose of this attack is to damage the grid without getting detected by legitimate users. The attack uses a Trojan Horse malware to get a backdoor access to one of the RAS controllers. Once malware is installed, the attacker gets control of the RAS controller whenever he desires. This includes outside the LAN of the RAS controller as well. Specifically, the malware provides undetectable communication between the attacker and the device, and provides attacker the ability to execute commands in the affected device. Once the malware installation is successful, we perform the coordinate cyber-attacks by replacing the existing RAS controller script with a malicious one which plays with a generator to damage the system. This part is intended to demonstrate the dangers of the malware in Smart grid.

In the second part, the defense scheme against the malware attack is proposed. The main idea is to detect and disable the device operating for RAS controller that is affected by some type of malware. This is done by introducing the one other device called Overseer. The Overseer should not have any access or control over any part of the actual grid (relays, generators, etc.). However, it should be able to communicate with all RAS controllers. RAS controllers are also upgraded so that they will take an extra measurement from a randomly selected generator which is reported to the Overseer with all the other measurements they normally take periodically. The main task of the overseer is to oversee the RAS controllers by taking updates from them. Through the usage of the proposed architecture, the overseer can detect a RAS controller which is acting maliciously. Once the malicious controller is detected, it can disable it using denial of service (DOS) attack on it until the situation is fixed. It is to be noted that the Smart Grid requires RAS controllers to perform corrective action during disturbances in the grid, they are just there to keep track of the grid during normal functioning of the power system. This means that grid does not need RAS controllers to function normally. Another possibility is when the Overseer is infected. Since Overseer has no access/control over the grid, the worst thing an attacker can do is to DOS a RAS controller which, again, will not affect the grid.

CHAPTER 1. INTRODUCTION

Background on Smart Grid

Electric Grid are electric supply networks. Their main function is to distribute electricity to their users. However, electric grid itself is not strong enough for the demands of today [1]. To fix this issue Smart Grid are introduced. These are just electric grid with communication layer added. This extra layer gives the grid ability to be monitored, and automated among other things [1]. This in return increases the reliability, sustainability, and efficiency of the grid [2]. Figure 1 shows a simple overview of a Smart Grid. The added communication layer connects grid to the network which enables the Control Center (CC) to control the grid remotely, and therefore, generally referred as to Supervisory Control and Data Acquisition (SCADA) [4]. This is what makes the Smart Grid a Cyber Physical System (CPS), SCADA communications being the cyber part.

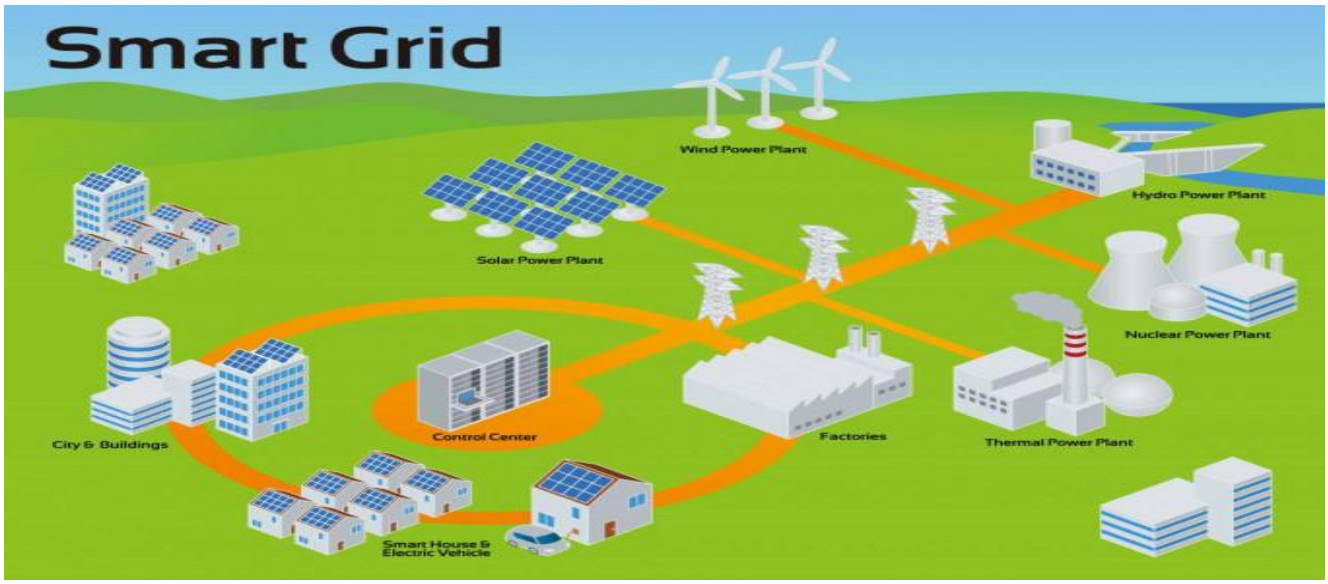

Figure 1: *Simple Smart Grid Overview [3]*

Nowadays, the SCADA communication in grid is mostly essential since it adds many abilities to the electric grid some of which are Automatic Generation Control (AGC), State Estimation (SE), Wide Area Monitoring, Protection, and Control (WAMPAC), and Remedial Action Scheme (RAS) [4], [5]. However, connecting the grid to the network also creates an unintended consequence, it opens the grid to any attack from the network it is connected (Cyber-Attacks).

As with any other network connected device, Smart Grid can be attacked through the network. This includes any network connected part of the grid. Moreover, Smart Grid is actually consisting of legacy parts that are updated with communication capability [7]. This may limit possible defenses of the Smart Grid. Now let's take look into the physical part of the grid.

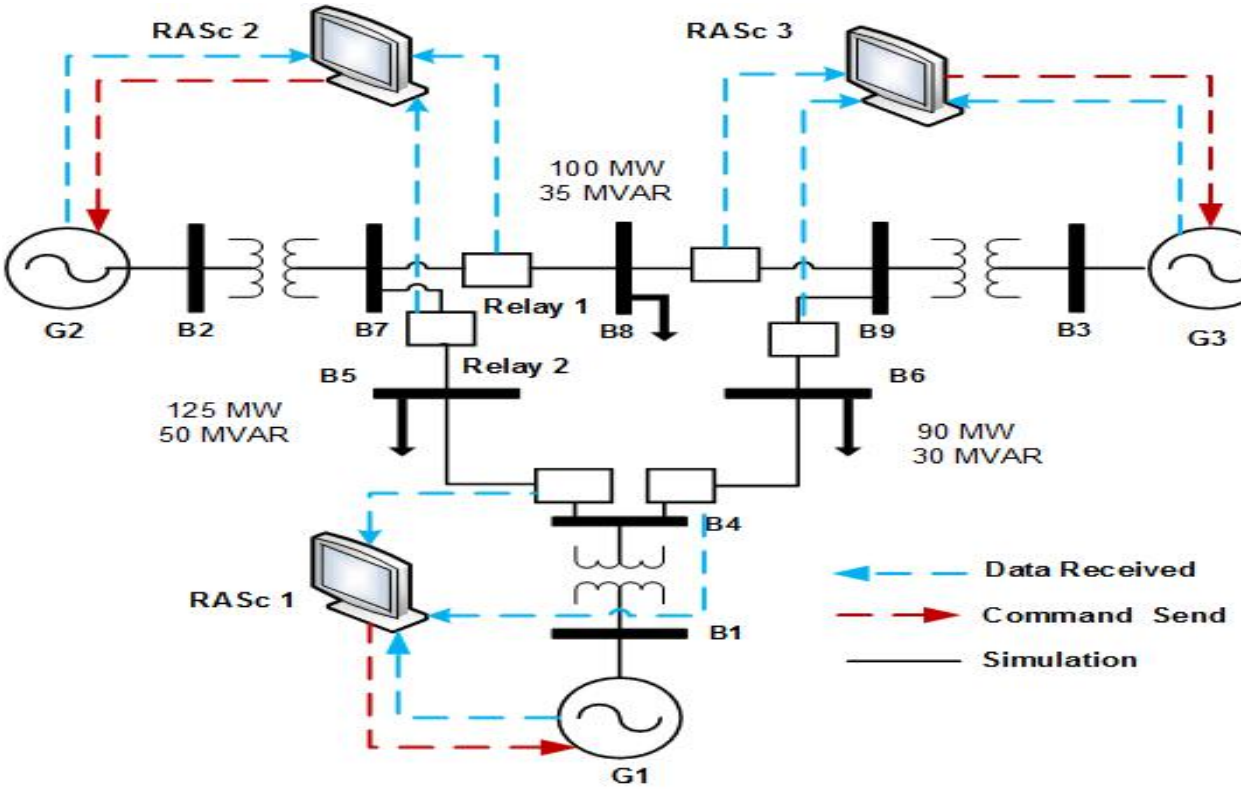

Figure 2. *Distributed RAS enabled IEEE 9 bus system [6]*

Smart Grid networks can be connected in many different ways. This thesis will consider a simple architecture for the gird which is shown in Figure 2. In here the relevant devices for the thesis are generators (named G#), relays (named R#), Remedial Action Scheme Controllers (named RASc #). Both the generators and relays are simple; generators generate power, relays either open the line to cut the distribution or close the line allowing distribution from that line. On the other hand, RASc's are bit more complex.

Remedial Action Scheme (RAS)

RAS is one of the features that makes the grid smart. The main purpose of the scheme is to fix any issue that may arise in the grid without requiring the intervention from humans [8]. As one can guess, this scheme makes the workers' job easier since they do not need to manually intervene each time something goes wrong. The RAS depends on the geographically distributed devices that is called RAS controller (RASc). RASc devices are distributed within each substation, and are responsible for enforcing the RAS in their own allocated sections [8]. Over the years there are multiple RAS's developed, Zhang and others overviews various different versions of these schemes in their paper [8].

In this thesis, only one version of the RAS is considered. This is one of the widely used version of the scheme, and also a NERC standard definition [10]. As far as this definition go, this thesis only focuses on the part shown in Figure 3. In here RASc collects data of their own section periodically, indicated by the blue lines in Figure 2. However, it stays passive until something unexpected happens. In this case, the unexpected thing is a relay getting tripped, meaning it opens up. There could be several different reasons for a relay to trip. Some of these are among

standard operations like a relay may overload and instead of burning, it will trip [11]. Others may be malicious like someone sending trip command to the relay. Note that RASc can also send a trip command but then this won't count as an unexpected change. Once this unexpected trip happens, RASc will become active, and start checking the Operational Transfer Capability (OTC) on each line. Now in the cases where multiple relays are connected to a generator then if one of the relays open then the same generation from the generator will have to go through one less line which increases each of the other relays' load that are connected to that generation which in return may overload other relays. For instance, in Figure 2, if R1 trips then the entire load of G2 will go to the R2. This will then pass the OTC of R2 so R2 will also trip. In this case, the system will lose an entire generators output. Therefore, to fix this, once RASc finds an overloaded line, it reduces the appropriate generators output, to an appropriate measure as shown in the Figure 3. This way generator won't end up out of service. Finally, when the other relay gets fixed then RASc will return the generator to the normal generation. All in all, the grid will end up fixing itself. Moreover, RASc will periodically report to the Control Center (CC) so that CC will know if there is a problem with the RASc.

As you can see to perform as intended, RASc needs to be able to send corrective commands to the electrical grid, shown in Figure 2 as red lines. This means that most of the grid needs to obey the RASc. Even though this is fine from the physical systems point of view, it is actually problematic from the cyber side of the system. On the cyber side, separation of duty, meaning limiting what any one device can do to as small as possible, is always preferred. Coleman and others explain why this is the case, quite well [9]. Among these, one specific

reason needs to be pointed out. In this system, RASc has too much power on the grid which begs the question, what will happen if an attacker takes control of it?

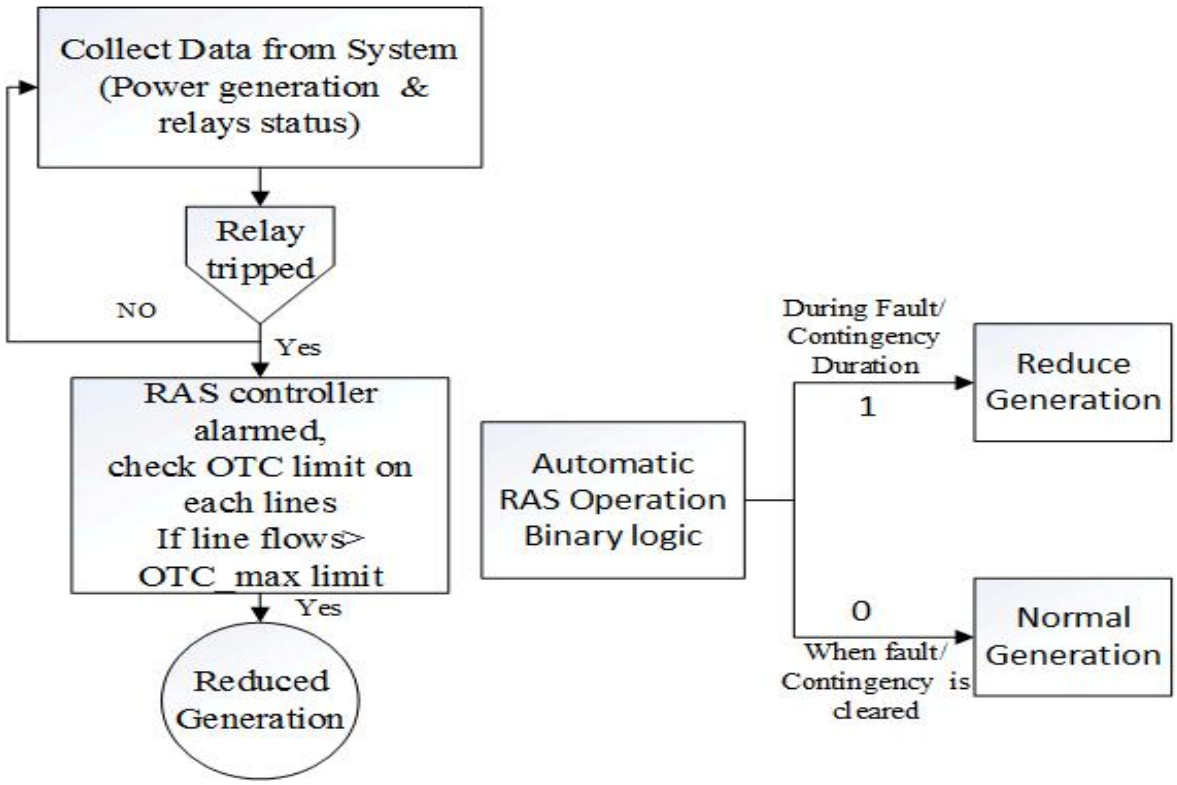

Figure 3. *RAS Considered in this Thesis [6]*

Background on Malware

Cyber Warfare, is a field where change happens relatively faster than other fields. This is because attackers and defenders are generally in a state of playing cat and mouse. For instance, A invents a security protocol, B finds a way to bypass it. Then A fixes the issue but then C finds another way to bypass it, and so on. This may go on forever, or one side may find a solution which is rooted in another field. For instance, most of the current encryptions work when used in the proper way. This is because encryption is done by mathematics therefore, the how to

technique to break it needs to come from mathematics as well; not form cyber side (again this is when it is used correctly). However, even when we put this encryption into Cyber Warfare then we will see that both sides can benefit from it. An attacker can use this to hide its identity meanwhile a legitimate user may use it to hide its credit card number. The point of this paragraph is that there is no sure way to keep attackers out of the system since if someone is determined enough, he'll eventually find a way to get in.

There are different ways for a malicious user infect a device, it depends on many things attacker's knowledge, skills, experience, even in his imagination, and many more. However, in this thesis the focus is on the malware. There are also many different types of malware. Some of the examples are virus, worms, Trojan Horse, spyware (Grayware), adware (Grayware). Note that graywares are like malware but less dangerous [12]. Once in the system these malwares can do many things to hide, and/or take control of the system. Some of the possibilities are using a rootkit to hide, open a backdoor to allow some to bypass standard authentication, using evasions to hide, and can escalate its privilege in the system [12]. More information about these can be found in the Elisan's book [12].

In the Chapter 2 of this thesis, the focus will be on Trojan Horse malware since it is one way to get control of the victim's device. However, at Chapter 3 of this thesis, it does not matter which malware is used as long as a device is completely compromised. Therefore, let's look into what a Trojan Horse actually does.

The main purpose of the Trojan Horse malware is give backdoor access to the system it is installed to the attacker. This means that once it is installed, the attacker can connect to the system from anywhere, and anytime by bypassing the main authentication completely. There are multiple ways for the malware to accomplish this so it depends on the implementation. Unlike

viruses, and worms Trojan Horse does not duplicate itself nor transfers itself to another system. However, it will try to avoid detection which again can be done many possible ways. Once it is operational attacker should be able to get a command line in the victim's system, meaning it can send any malicious script or receive any data his account's privileges in the system will allow. However, once he is in, there are also possible ways to escalate his privileges if needed [12]. This means that attacker will eventually get the full control of the system if he wants to. Then the victim's system can be used for anything malicious. For instance, it may be used as a bot (where attacker attacks another device from victim's devices without victim even realizing it. IN this case, if the attacker to second victim will get caught then first victim will take the blame. Another option would be to selling the victims, device as bot to another person. However, as for this thesis, if attacker takes control of the one of the grid devices, it can then attack the Smart Grid, possibly damaging it permanently.

Assumptions

In this thesis there are couple assumptions made to focus on the actual attack/defense part of the Smart Grid. Some of these assumptions also introduces some limits but they are necessary for these to work. Here are the assumptions made on this thesis:

- Malware is already installed, and the attacker has full control of the device.
- There are at least three RASc used in the system.
- Only one device is infected throughout the system.

Thesis Organization

In this chapter 1, thesis was focused on the background information for the different fields. As for the rest of the thesis is going to follow this:

- Chapter 2 introduces Stealthy Malware Based Attack to the RAS scheme. In here, thesis demonstrates the dangers of Malware in Smart Grid.
- Chapter 3 introduces an Attack Resilient RAS which is main purpose is to detect and prevent malware attacks. The focus here will be how to defend against Malware.
- Chapter 4 will give summary of the main points of the thesis and will conclude the it.

CHAPTER 2. A STEALTHY MALWARE BASED ATTACK ON REMEDIAL ACTION SCHEME

Chapter Purpose

The purpose of this chapter is to demonstrate malware based attacks are a big threat to the Smart Grid. Moreover, this chapter will show that malwares can be used to damage the system permanently as well.

Throughout this chapter an instance of a malware based attack is going to be discussed. This particular attack will show what type of damages to the Smart Grid are possible, and realistic.

Overview of the Attack

As it is mentioned in the Chapter 1, even though RASc makes the maintenance of the Smart Grid way easier, it has too much power over the grid. The ability to command any grid device is great ability to have. Therefore, by controlling a RASc attacker can control the entire grid. Therefore, RASc becomes a natural target.

The attack timeline in Figure 4 starts with the installation of the malware. Again, it was the assumption is that this installation part is already done, and now the attacker has full control of the RASc device. This means that attacker can now control the RASc script as well whenever he wants, and he can also stand back and observe the system. In other words, now attacker has full access to local area network (LAN) via infected RASc's device. This means even if the LAN has firewalls towards the internet/wide area network (WAN). It does not matter because malware already gave the attacker a backdoor to RASc which is inside the LAN. Therefore, attacker has

access to the LAN regardless of where he is physically located. As for the malware, there are couple ways it can bypass the firewall without getting noticed anything. The explanation of how it is done in this case will be presented in the Implementation section of this chapter. Before moving on, it needs to be pointed out that full access means attacker can now listen, capture, drop, and/or modify any packets in the LAN, and he can be physically located on anywhere in the world with an internet access.

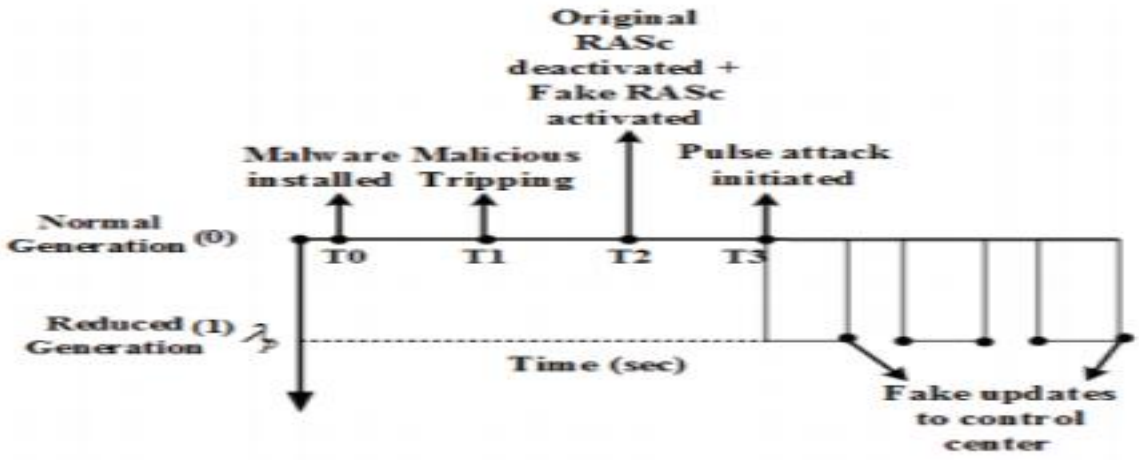

Figure 4. *Attack Timeline [6]*

Now that the attacker has a foothold in the LAN of the Smart Grid, let's move to the time T1 in Figure 4. Recall that RASc stays in the passive role until some unexpected change happens in the system. Since now attacker controls the RASc, the next step is to make this unexpected change so attacker's fake RASc can start giving orders. Now doing this requires attacker to know the trip command. For this attacker may know the communication protocol used which in that case he can just create its own trip message and send it to a relay. Note that these protocols are not really private so it is possible that attacker may have done its research beforehand. Another option here is that attacker can just do a replay attack. The replay attack is an attack where

attacker saves a certain packet during observation of the LAN, and then he sends the copied version of this packet (replays it) to the same device and expects that device to do the same thing. In this case, the captured packet is going to be the trip command which also happens during normal operations as well. Notice that in this latter case attacker did not even needed to know the protocol used. Before moving on it is important to point out that again, that Smart Grid mostly consists of legacy systems [7] so most of the grid devices are simple, they can't do anything complex like some encryption type communication.

The next step is at time T2 in the Figure 4. At this point, it is important to point out that RASc updates the CC periodically because previous trip at T1 activated it. Since the attacker wants to control the RASc, he also needs to be able to figure out this period of update. However, this is not a problem at all considering he can just listen the LAN to quickly get an estimate of the update periods. Once this is done, attacker can simply deactivate the original RASc and then can run his own fake RASc version of it instead. Note that at this point, little variance of the update period by the attacker's switch is tolerable since network can have delays in them too especially when LAN is constantly full with updates. Therefore, small variations on the updates are not uncommon, and it won't arouse suspicion. However, it should be clear that attacker knows the updates times at this point. This means he can time the switch just right to get around this issue as well.

The next, and the final part of the attack is starts at time T3 in Figure 4. At this point attacker controls an active RASc which sends fake updates to the CC. Therefore, nobody knows something bas is happening. Therefore, at this point attacker can do basically anything to the his RASc's section of the grid without getting detected since his RASc sends fake updates to the CC. Moreover, he can give commands to other reachable RASc's sections as well. However, an

attack on an uninfected RASc's section will be detected, and will be fixed by that RASc. Moreover, CC will be alerted by that RASc as well. Therefore, this is not recommended. On the other hand, any attack done to the attacker's RASc's section will not be detected at all since only device that measures it is under the control of attacker. This means attacker can do some attack until a permanent damage to the system occurs. More importantly, he does not even need to rush it. In this thesis, attacker will perform a pulse attack which is where he reduces the generation suddenly for some time and the increases it back suddenly for some over and over again, creating a square wave. Note that reducing/increasing generation packet can be obtained the same ways as attacker did with the tripping packet. Notice that if this attack goes on for a time, which is very likely since not noticeable, then system may suffer great damage.

Implementation

This attack is implemented on Iowa State University's PowerCyber Testbed which simulates the grid [13]. Implementation of the attack is done in this testbed to avoid damaging a real grid system. The entire implementation of the system is with the attack performed on it is shown in Figure 5, and Figure 6 shows the what each step corresponds. Notice that for simplicity sake attacker is already inside the LAN for this implementation therefore, tripping and pulse attack are directly done by the attacker in here. However, these can be easily done from the RASc as well so again attacker does not need to be in the LAN once the malware is installed. Moreover, notice that there both the pulse attack and false report to CC happens at step 4 in Figure 5, and Figure 6. This signifies that they are happening at the same time.

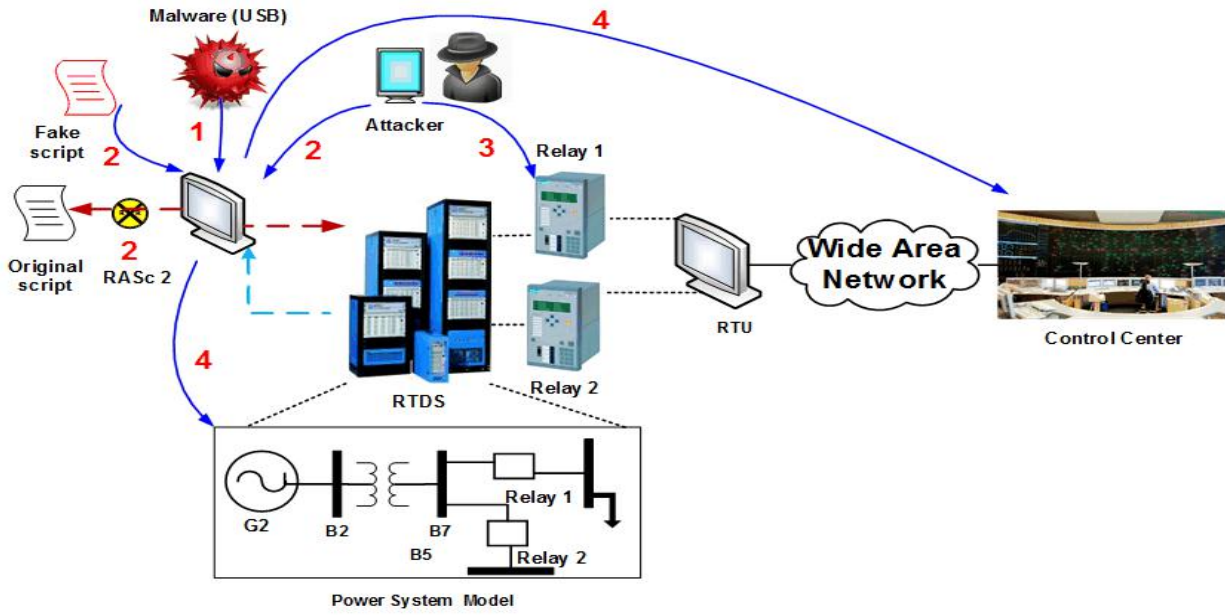

Figure 5. *Attack Implementation [6]*

Step 1: Install malware (Trojan horse) using USB drive, emails and other social engineering skills

Step 2: Perform malicious tripping by replaying the captured packet

Step 3: Close the program running for RASc and execute the fake RASc program

Step 4: Initiate pulse attack on generator operated initially by original RASc

Step 4: Send false updates to the control center close to the previous original RASc updates

Figure 6. *Attack Implementation Steps*

Now in Figure 5, there are two devices that are not mention before. These are RTDS, and RTU. RTDS is Real Time Digital Simulator which is what simulates the grid. As for the RTU it is Remote Terminal Unit which is basically a gateway for this LAN, it provides secure communications with the CC over internet/WAN. In other words, it provides SCADA connectivity.

The most relevant part of the implementation to this paper however, is the implementation of the Trojan Horse malware. The malware was done in Python, and ran on Windows. However, the most surprising thing about it was that even though it can have a big impact on the grid, it actually does very little. Let's begin with the installation, to install it just needed to be ran once if it ran on with admin privileges it will also lists itself to automatic start up in windows. This means windows starts the malware automatically each time it is opened. However, this is not necessary since in Smart Grid in general, all devices should be running constantly.

Once the installation is done, it needed to open a backdoor which essentially means a new connection. This can be done in two ways either as a server which means opening a port in the device and listen it, or as a client which means trying to connect to an already open port of another device. Now there is a problem with the first one where malware is the server. The problem is that being as server means, opening a port which means spending resources which increases the potential of detection. This is not just because someone constantly tracking the resources used in the RASc but because it may slowdown the device itself which may alerting the CC something is up. More importantly, this would be an instant detect if someone does a port scan which is when someone tries connect each of your ports to see which are open [14]. This is generally used by attackers but sometimes system owners do it for testing purposes as well. This

leave us with the second option where malware acts as the client which is what used. IN this case malware can simply try to open a connection with the attacker's device periodically. Most of the time connection is going to be refused since attacker will only open it when he wants to get control of RASc. This means that malware will get connection errors which are dropped to prevent detection. Once attacker wants to connect all he needs to do is open the appropriate port in its own device and wait for the next open connection from malware.

One thing to note here is that malware needs to have attacker's IP hardcoded which may give enough information to the system owner to find the attacker if malware is caught. However, if this is a concern then a compiled language can be sued instead which in that case malware would be binary executable. In this implementation this was not a concern. As for detection wise sending a single connection every once in while would be lot harder to detect then the previous alternative.

Now there is only one part left which is how to get past firewall. Well, the answer is simple we use the internet with encryption. What this means is that usage of port 80, which is reserved for the http/https protocol, to send/receive encrypted backdoor communication packages will hide the backdoor. The reason is simple https is an encrypted protocol which means lots of encrypted packages arrive to this port so if the backdoor packages would use an encryption as well, they should look just as random as https packages [15]. This means firewall cannot distinguish them. Therefore, now malware can send and receive undetected communications as well. Finally, when a connection is established malware simply executes whatever attacker sends in command line and sends back its output which gives the control of the system to the attacker.

In short, all malware does is periodically connect to the attacker through an encrypted channel, once connection is established, it will simple execute whatever attacker sends in a command line of the system that is it.

There are some other things to note in this section. To begin with, the communication between RASc and RTDS were in DNP3. As for the RASc Python Script is used to implement them where decisions of the RASc depended upon the action table in them which also stored what to do as well. Therefore, it was more of a look up table logic. Wireshark [16] is used to capture/sniff packets in LAN, and finally, CryptCat [17] is used to for encrypted communication between malware and attacker.

Experimental Results

The results for this has attack has been gained through by long into the generator while doing pulse attacks with different duty cycles to it. These can be seen on Figure 7, and 8. In these figures Pset2 refers to Load References which is what attacker's RASc command generation to be. However, in reality it takes to increase/decrease generation so we get the graph in P2 which is the actual generation. Rest of the graphs TM2 which is mechanical torque, and W2 which is angular speed are there to show that generators are having hard time, meaning they will permanently break if this continues on. However, this thesis' focus is more on the cyber side. Finally, duty cycles mean in every period of the square generation how much of the time it generation should be lowered. Therefore, 10% duty cycle means 10% of each period the generation will be reduced and 90% of the period it will be normal. Finally, you can notice the initial tripping at the beginning of each graph.

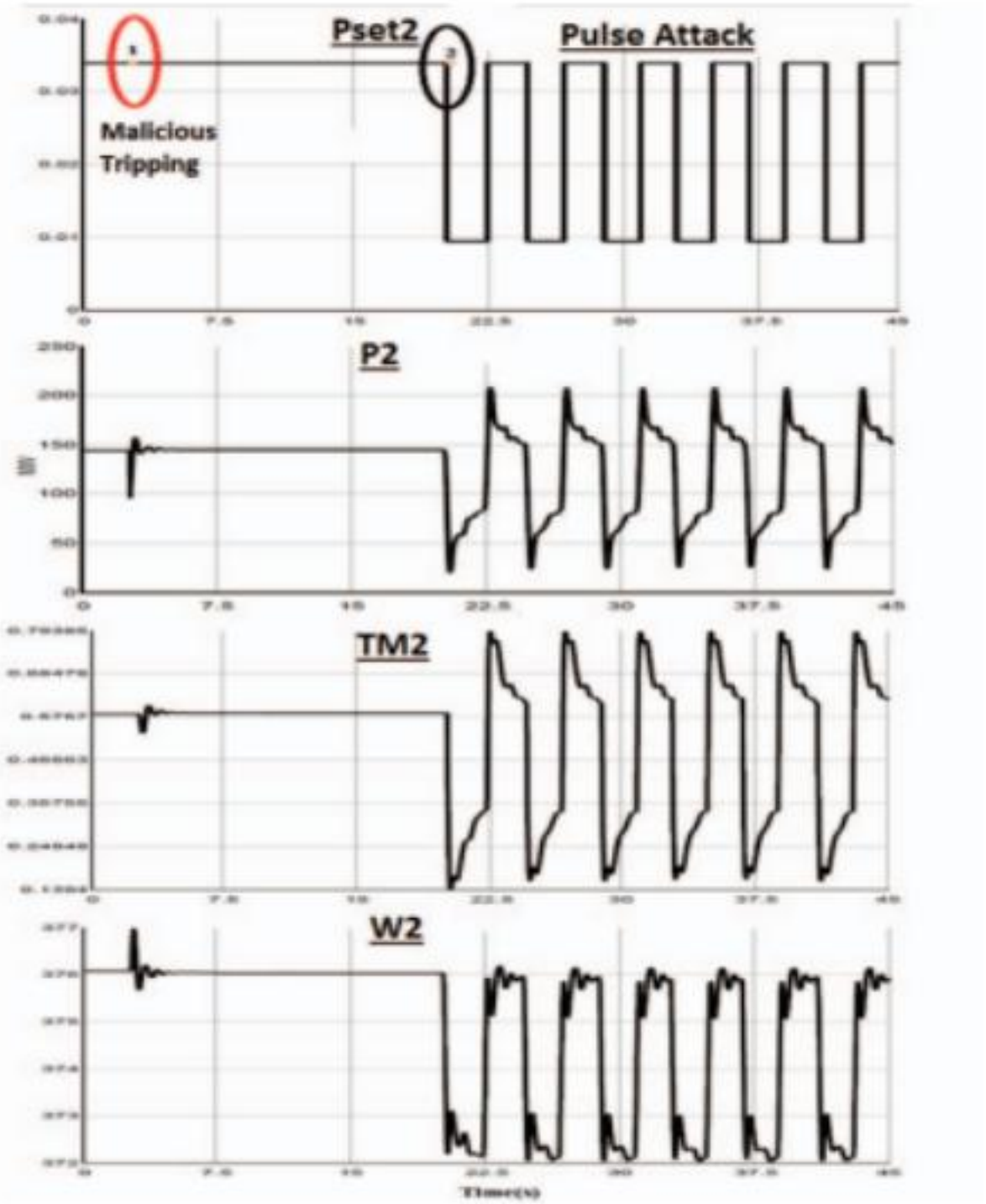

Figure 7. *Attack Result 50% Duty Cycle [6]*

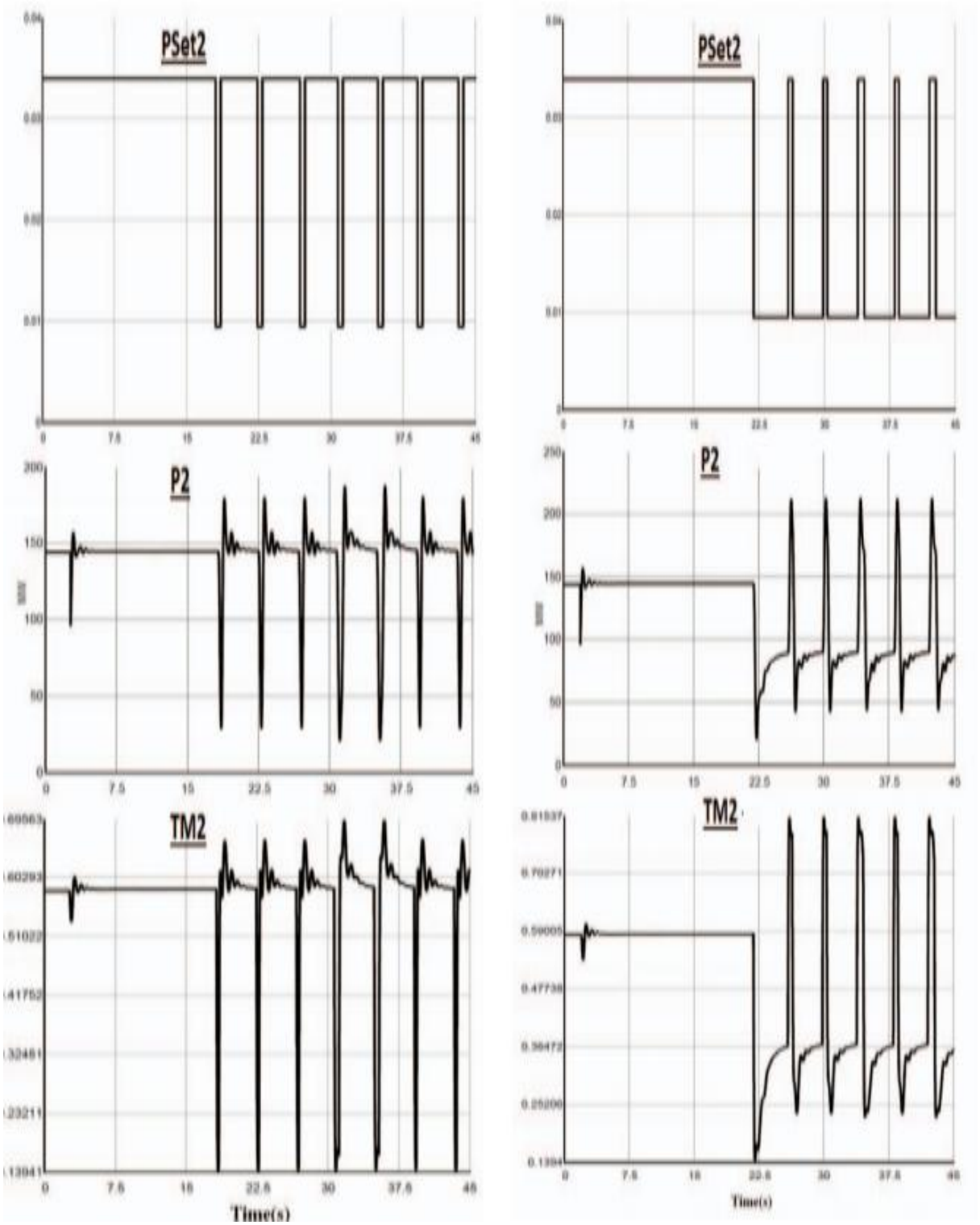

Figure 8. *Attack Results 10% (left), and 90% Duty Cycle [6]*

Possible Mitigation – Digital Signature

A possible mitigation for this particular attack would be implementing a timestamp digital signatures. This would make replay attacks not possible since attacker can't sign packages which means can't modify them. As for the captured packages, their timestamp would expire.

However, there is a problem with this approach. The problem is that most of the Smart Grid devices are legacy and may not be able to support this digital signature (or encryption which will also work). This may be overcome by introducing two new signing devices right in front of each side of the communication. This way everything in the middle will be signed, see Figure 9. More information, including mathematical information, about digital signatures can be found here [18].

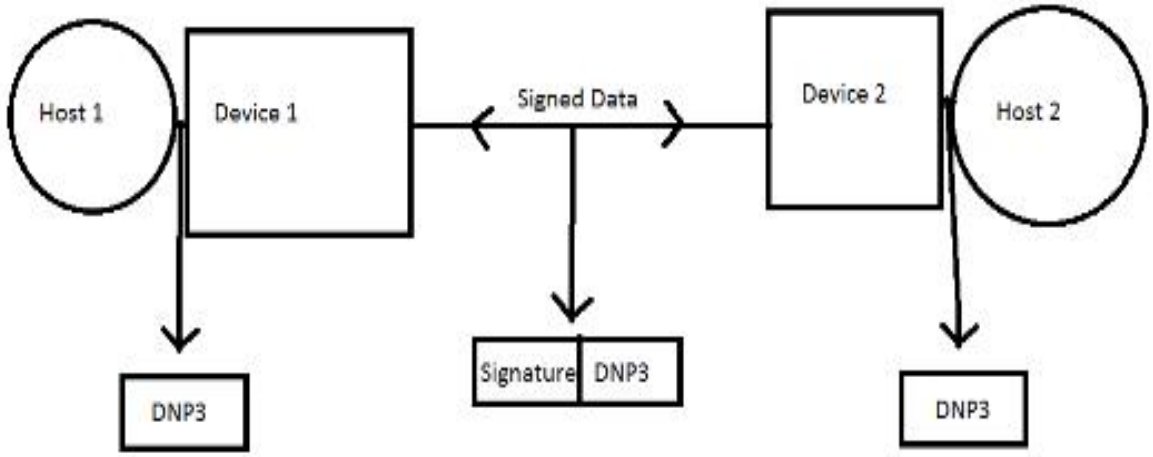

Figure 9. *Digital Signature*

Possible Extensions

The demonstrated attack is just a single instance of what malware can do. This attack can be extended to many other malicious things, even when with the digital signature. The problem is that RASc still has too much power over the grid. Therefore, it can still cause harm. For instance, one simple thing attacker can do is to perform Man-in-the-Middle [19]. Then it can simply drop all packets to cut off the communication (Denial-of-Service).

It is also important to note that malware in Smart Grids in general can cause disasters. This particular instance only considered a single type of malware however, there are other versions as well which may cause more problems to the Smart Grids.

CHAPTER 3. ATTACK RESILIENT REMEDIAL ACTION SCHEME

Chapter Purpose

In this chapter, an improved RAS is presented. The idea behind this version was to be able to continue working even when a RASc is completely compromised. In other words, the purpose is to mitigate malware problem as much as possible. When a RASc is compromised, this new scheme should be able to detect the compromised RASc, alert CC, and disable the infected RASc until CC fixes the situation. In other words, problem will still need a manual intervention however, with this scheme Smart Grid will continue working properly even during the attack. This means that unlike before there won't be any urgency to fix the infected RASc since electric distribution will continue on as expected.

Overview of the New Scheme

In the improved RAS, two new things introduced. First, is that a completely new device which is called Overseer. As the name suggest, its job is to oversee all the RASc. Second, is that now all RASc are upgraded which means they act little differently. The new architecture can be seen in the Figure 10.

There are couple major changes here. To begin with now RASc no longer report their periodic updates to the CC but instead they report it to the Overseer. Moreover, now all RASc take one extra measurement in addition to their standard measurements periodically. Every period each RASc pseudo-randomly chooses a generator, and measures its output. Notice that this random choice excludes the RASc's own generator. Taking other zones' measurements is

possible because all measurements done through the network which entire Smart Grid is connected to meaning every RASc should have access to other zone's devices. Also this was not an issue before as well since if infected RASc attack some other RASc zone then the zone owner RASc will fix the problem, and report it to the CC in previous case, Overseer in this improved scheme.

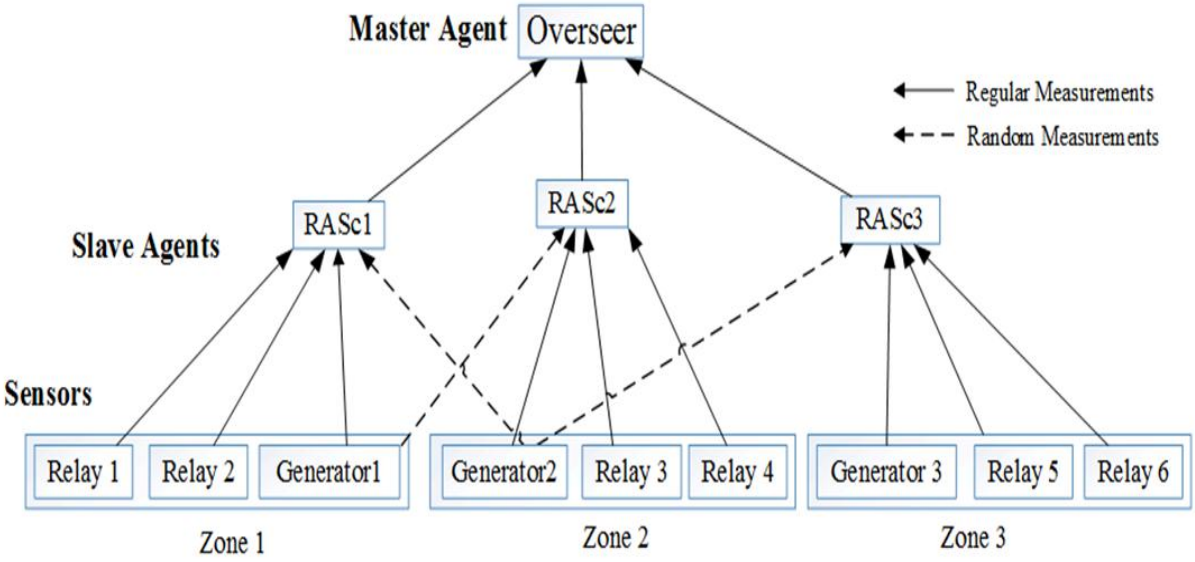

Figure 10. *New Scheme Architecture*

There is one other thing that needs to be mentioned in this architecture that is not really visible in the Figure 10 and yet it plays a critical role. This thing is that Overseer should not have any access to the Sensors, meaning relays, and generators in this case. Therefore, there is a separation of duty here which forces Overseer to only be able to communicate with RASc. Moreover, Overseer is also not allowed to give direct commands to the RASc however, it can give any other commands. Note that in here a direct command is specified as any command that directly affects the sensors. For instance, open Relay 3, and reduce generator 1 are some

examples of direct commands. Again, the idea here is that Overseer cannot do anything to the sensors, even fix a problem within them. In this scheme, this is what separation of duty entails.

The main idea here is that the detection of the compromised RASc is done by comparing other RASc values inside the Overseer. This is possible because of the extra random measurement that each RASc reports periodically. Once Overseer detects some problem it disables the RASc, asks another RASc to fix the damaged zone, and alerts CC. Therefore, the infected RASc stays disabled until CC manually fixes the issue. However, grid itself does not require RASc to operate. In fact, RASc is only there to double check the grid. Therefore, Smart Grid will continue to operate as normal. However, if RASc for each zone is really desired during the infected RASc's disabled period Overseer can be programmed to ask another RASc to fix the disabled RASc's zone (same way that Overseer did during the alarm raising part) periodically.

Before moving on it is very critical to be note that Overseer needs to be a weak device. This means that it should only be able to disable a single RASc at a time because otherwise, Overseer would have too much power which would make it the new malware target, same issue as before.

One thing to point out here that if these directions are followed then malware infecting the Overseer will cause at most the same damage as the malware which infects the RASc. In both cases at most one RASc will be disabled. This is because Overseer can't affect the sensors anyway, so attacker can't control anything. Moreover, Overseer is only strong enough the disable one RASc at a time, meaning attacker cannot disable more than one RASc. As for the malware affecting RASc case, Overseer will detect it through comparisons of the different RASc measurements and will disable that infected RASc, and ask another RASc to fix its zone.

Therefore, regardless of which device attacker chooses, the worst case outcome is always one RASc being disabled and Smart Grid sensors continue to work as normal.

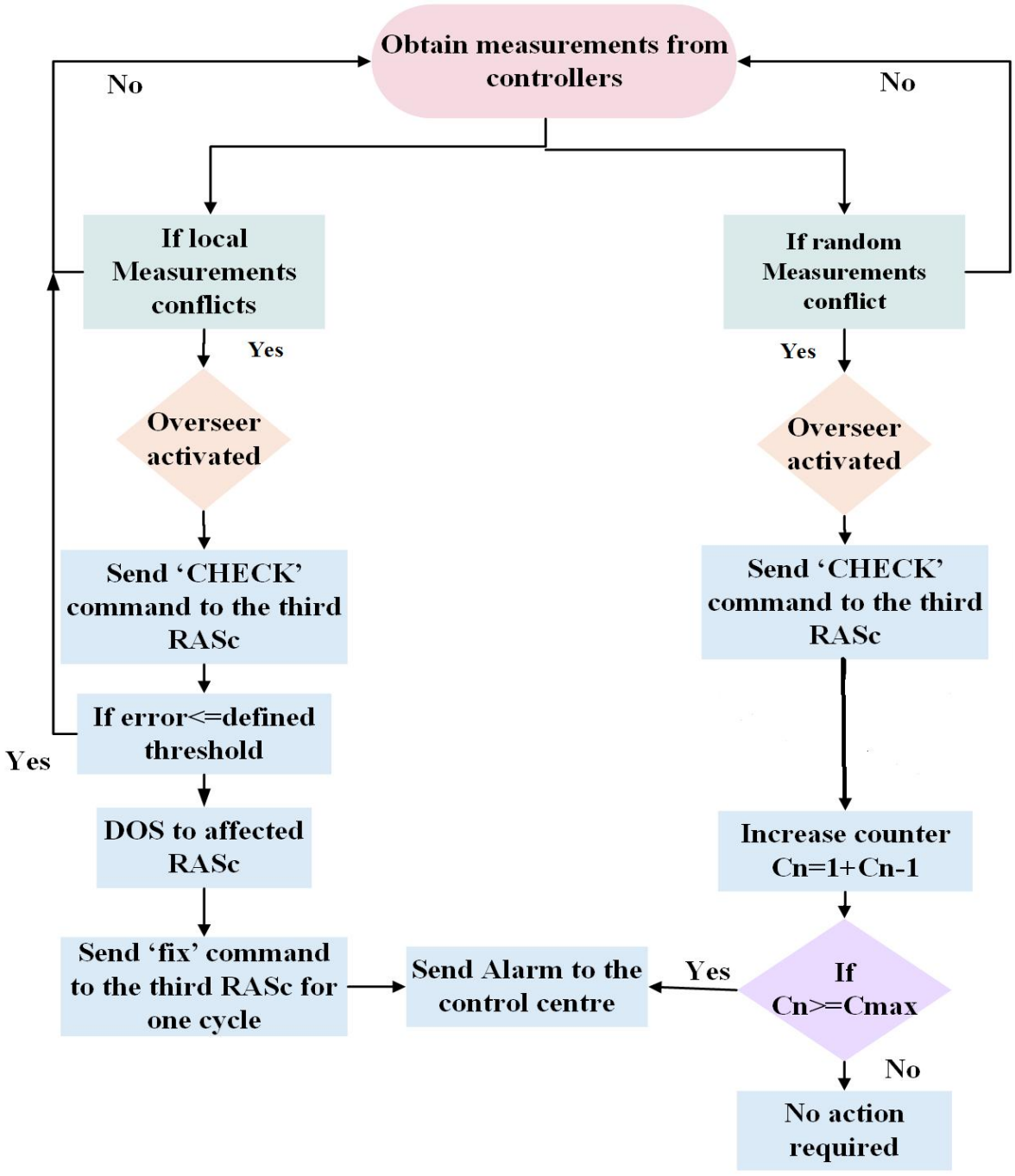

Figure 11. *New Scheme Details*

Now with the general scheme in mind, let's look at the details, as in what really happens during this new scheme. Therefore, this is the place to go over Figure 11. Figure 11 shows two sides of a measurement conflict. The left side RASc who owns the conflicted device and the right part is the RASc whose randomly chosen measured device is conflicted. So Figure 11 begins with Overseer periodically collecting measurements until a conflict happens. At this point all Overseer knows is that two RASc reported different measurements for the same device. This may mean that one of them is compromised or it may simply mean that there might be a synchronization error through network (i.e. a packet dropped) which is normal. Therefore, Overseer asks another randomly chosen RASc to measure this device on the next period to double check it. This done by sending a CHECK command to a randomly selected third RASc. As for the RASc who receives this command, it measures the said device in its next cycle instead of randomly selecting a device. This means it will still report the same amount of measurements. Once Overseer receives next cycles measurements which contains CHECK commands answer this is where right and left part separates.

Now, in the right part of the Figure 11, Overseer can compare this with this cycle's measurement of the same device from the owner RASc. If the difference is smaller than some predetermined error threshold (since devices are not perfect, measurement may have small differences) then everything is normal, this means third RASc back up the original RASc which means previous cycle was an error of sorts. However, if the difference is bigger than error threshold then Overseer sees that two RASc disagreed with the owner RASc. This is interpreted as owner RASc is compromised and is currently modifying its own sensors. Therefore, Overseer disables the said RASc, alerts CC, and sends FIX command to a randomly picked RASc. The RASc who receives this FIX command switches its zone for the next cycle only during which it

performs its standard grid stabilizing duties. At the end of the cycle RASc switches back to its own zone and continues as usual.

As for the left part, for this cycle (the cycle after conflict) this RASc was not forced to choose the same device as its random since otherwise, system ends up being too many RASc forced to choose the same device which guarantees certain device(s) to be not double checked at that cycle, giving attacker a something to attack for that cycle. Therefore, Overseer does not have something to double check this RASc. However, it should be pointed out that if this RASc is compromised at his point Overseer knows for sure that he is not harming the system since otherwise, it would end up in the left part of Figure 11. Therefore, if this RASc is compromised all it is doing to report the wrong random measurements which is harmless to the system because Overseer's double check prevents this raising any alert for the wrong RASc.

However, the problem here is that Overseer is not sure whether this RASc is compromised or not. Moreover, if it is the it is not good idea to let attacker sit in that RASc even if he is doing nothing wrong. This is where the counter part of the Figure 11 comes in place. Since it is not possible for Overseer to double check this RASc's random measurements, instead Overseer keeps a counter per RASc which increments each time a random measurement conflicts. However, it should be noted that this counter will increase in even in the conflicts this RASc turned out to be correct. This is why instead of double check like it did in previous (where only two wrong allowed) Overseer allows more than two wrongs. The exact number of conflicts allowed is a predefined threshold inside the Overseer. Once a RASc's counter passes this Overseer assumes it is compromised. However, in this case it just alerts the CC but does not disable the RASc since disabling RASc is costly (see last three paragraphs of this section), and attacker is not harming the system so no rush. If at some point attacker starts harming the system,

then attacker's RASc will fall in the left category in Figure 11 which makes it end up getting disabled.

It is also should be noted that after every so many cycles (exact number is count reset threshold defined inside Overseer) all counters will reset to zero. The reason for this is that if counters never reset then each RASc will eventually pass the alert threshold since networks is not perfect. To put this in perspective if count threshold is twenty and count reset threshold is thirty. Then Overseer says that for every thirty cycle if a RASc reports twenty conflicting random measurements then I will assume it is compromised. However, without the count reset this turns out to be if a RASc random measurements conflict twenty time for the entire lifetime I will assume it is compromised. This does not work since for instance, twenty conflict in a year won't indicate attack.

Finally, there is one last part to all of this new scheme which is how exactly Overseer disables a RASc. Now this part is the only problem part since disabling any fully compromised device is a problem itself. If a device is fully compromised, then it is impossible to disable it through software since attacker will intercept it. This eliminates signaling the device type of things. Other devices can be warned to ignore the infected devices but when it comes to Smart Grid most devices are legacy, meaning they usually can't selectively ignore commands. Moreover, since the goal is to disable the device until manual intervention then any manual disabling methods, like plugging it out, will be out of options. Therefore, only other possible solution that comes to mind is to prevent the device from sending anything.

In this thesis this is done through a Denial of Service (DOS) [20] attack from Overseer as seen on Figure 11. However, it should be pointed out that this introduces its own problems. If going this route, then DOS speed must be modified so that it won't DOS the LAN as well which

is highly likely. Although even in that case grid should still continue to work since communication layer is for the extra features which can be regained once manual intervention from CC happens. Also, again, Overseer should be only strong enough to do one DOS at a time otherwise, it would have too much power over the grid which turns us back to the old scheme's problem. The honest recommendation here would be for the reader to find a better way to disable a fully comprised device, only use DOS if no other alternative is found. However, for the purposes of this thesis DOS will be used.

There is also another alternative that can be done here which is handling the disabling in network level. One way to do this is to put each RASc to one port of a network switch, and then give Overseer the control of this network switch. In this case, Overseer can simply turn of the corresponding port of the RASc. However, this itself creates more issues than solutions. To begin with it is extremely wasteful to use one device per port. To put this into a perspective a general switch has five ports. Therefore, for small system maybe doable but for a real life Smart Grid this will going to be costly, requiring lots of network switches. Moreover, for this to work the entire network of the Smart Grid needs to be readjusted completely which is also very hard to do. Finally, even with all this done, it is still coming back to the same issue, Overseer gets too much power over Smart Grid again. In other words, in this case, if malware affects the Overseer then attacker can disconnect the entire Smart Grid through switches. Therefore, so far this does not seem to be a viable option.

Experimental Results of the New Scheme

In this scheme there are three different experiments has been done. These are pulse attack, ramp attack, and how long it takes to detect compromised RASc after measurement differences passes the error threshold.

Let's begin with pulse attack this where attacker tries to perform the same attack as in the previous chapter (essentially makes generation a square wave). Figure 12 shows how many cycles it takes to detect the pulse attack after it is started. Note that a cycle is the frequency of updates. In other words, every RASc gets measurements from sensors, and sends update per cycle. The cycle can differ from system to another. It can even be less than a half second depending on the system. In other words, five cycle means five measurements sent to the Overseer. As you can see as the error threshold increases number cycles it takes the pass the threshold also increases.

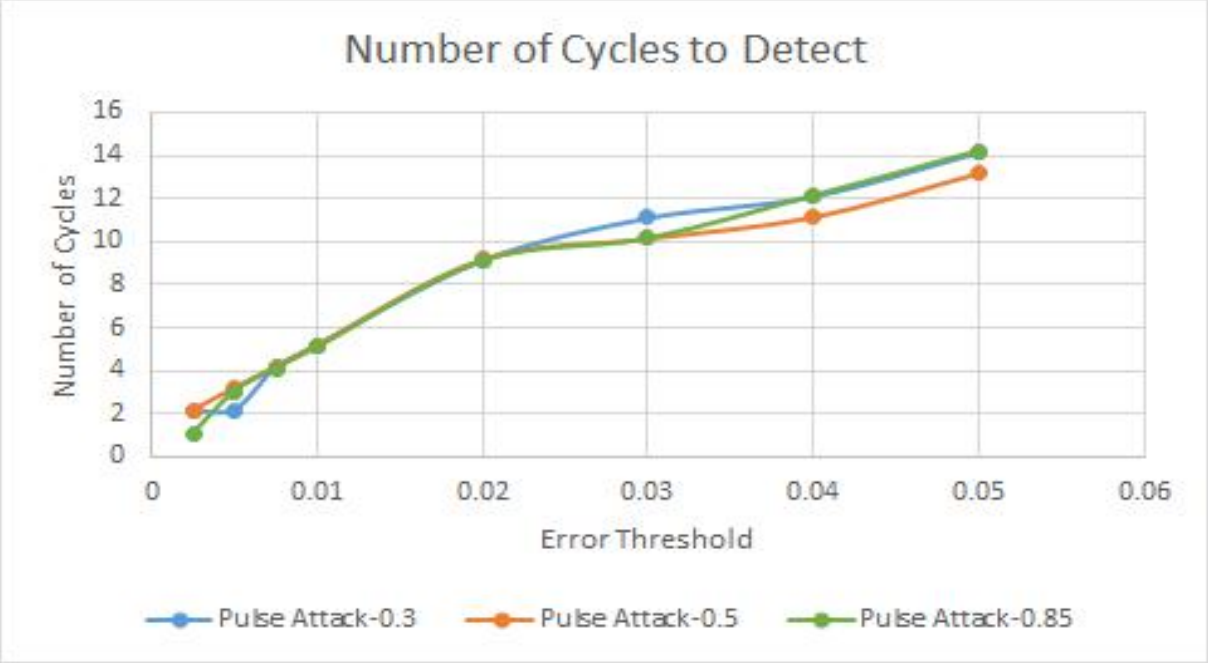

Figure 12. *Detection Cycles for Pulse Attack*

In Figure 12, the numbers per pulse attack shows duty cycles of the attack which is again, how much of the period the generation was reduced. However, as can be seen from the graph duty cycles does not really matter since ramp attack suddenly increase/decreases, meaning it suddenly passes the error threshold regardless of the duty cycle. Of course it still takes more time to pass the threshold when the threshold itself is bigger which explains the increase in the figure 12.

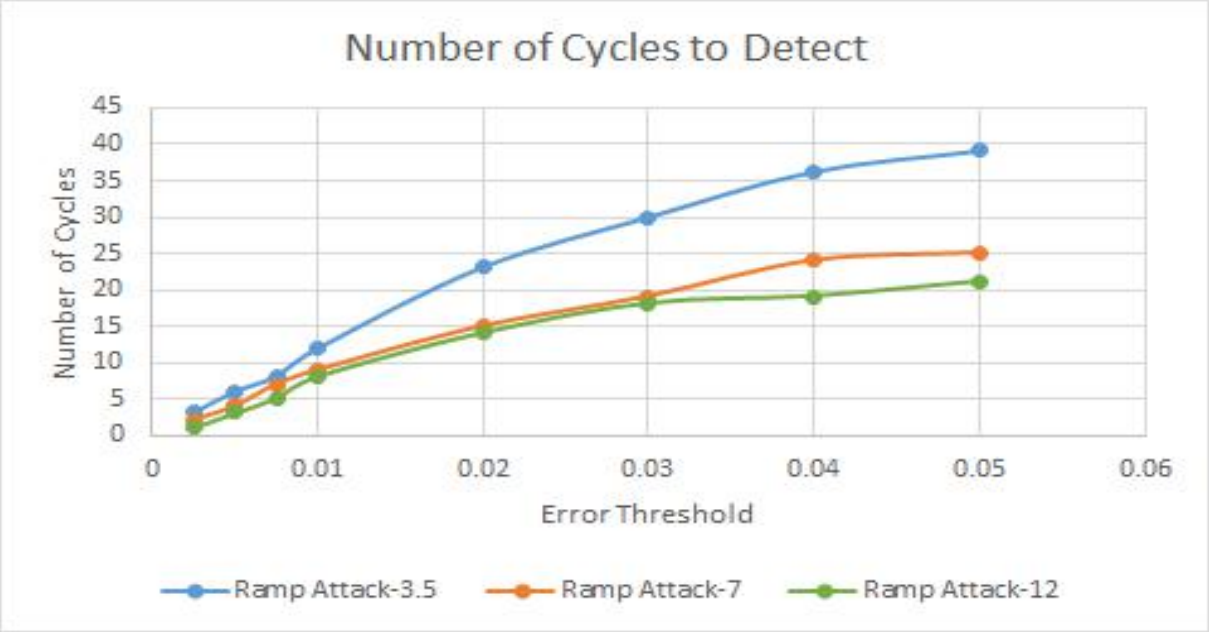

Figure 13. *Detection Cycles for Ramp Attack*

In Figure 13, the numbers are the step size per Ramp Attack. A Ramp attack is when the generation increased/decreased per step size ever cycle. So unlike pulse attack it slowly increases and decreases depending on the step size (bigger step size, faster increase/decrease). This creates a triangular signal instead. Therefore, as you can see number cycle it takes to detect is bigger than in pulse attack since it takes a while for difference to pass the error threshold. Also step size matter since bigger step size faster it will pass the error threshold meaning less cycles to detect as

seen in the Figure 13. Also, again, it takes more time to pass the error threshold when error threshold increases and therefore, the increase in the Figure 13.

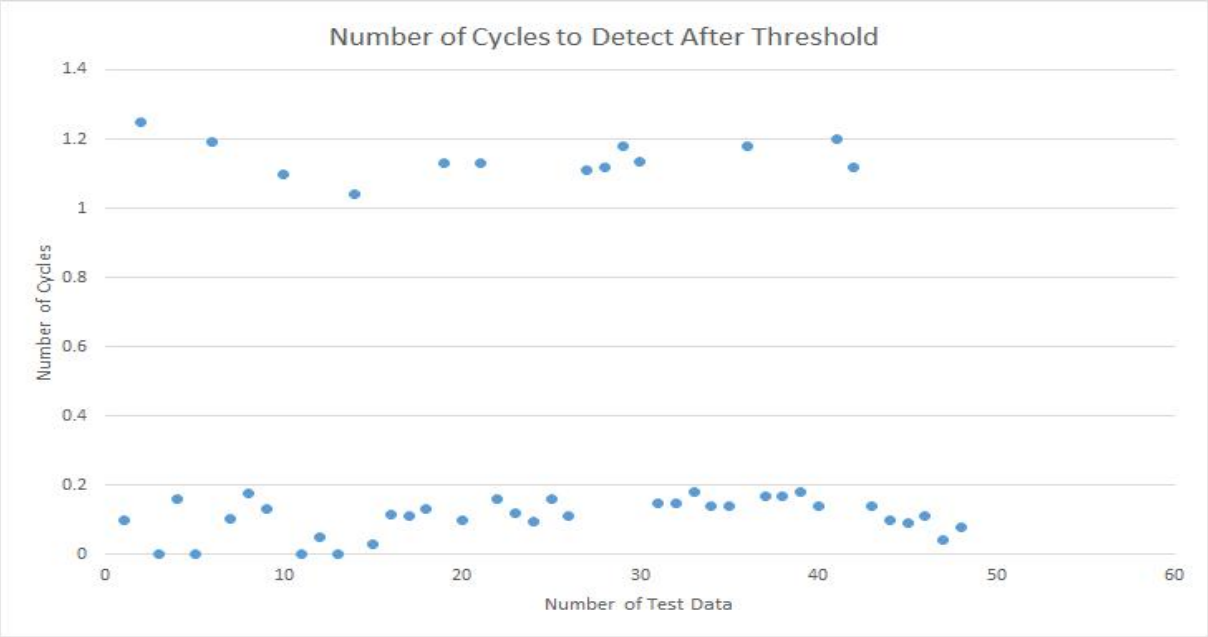

Figure 14. *Detection Cycles After Threshold*

Lastly, Figure 14 shows how much cycles it takes to detect after error threshold for the measurement difference is passed. For this part recall that in order to detect malware one of the other RASc needs to randomly choose one of the infected RASc's sensors to measure. In other words, Figure 14 shows how long it takes for a RASc to randomly select a faulty RASc's sensors after the measurement is detectably different. In this test Overseer either detected it at the same cycle that threshold is passed, or the one after either way it happened really fast. However, this graph is dependent on the architecture of the grid.

Limitations of the Scheme

The biggest one is the usage of DOS since it affects the network as well. Moreover, if DOS is used the system also cannot be extended to prevent multiple compromised devices as well since it will completely convolute the network.

Another limitation is that scheme depends on at least three RASc per Overseer to be present since overseer depends on the assumption of majority being right. This introduces a problem when majority of the RASc are compromised.

Finally, data synchronization may be an issue. Since comparisons depends on the updates, out of synch data may result with false positives. This is especially bad when combined with DOS as the disabling method.

Future Work

This scheme can be improved to work on cases where multiple RASc are infected. However, this is not an easy task, need to account for the cases like where randomly chosen third RASc is also an infected one. At this point it becomes really hard to figure out which RASc is infected or not. On the other hand, maybe statistics can be used to make best guesses about RASc.

Another direction would be finding an alternative to the DOS as the means of disabling. It might be worth noting that maybe infected RASc's internet can be cut out through firewall, meaning firewall may have configured to drop every packet from the RASc, which will end dropping out the attacker's communications as well. This also may worth putting time into.

CHAPTER 4. SUMMARY AND CONCLUSIONS

Summary

In the first part of the thesis, an instance of a malware based stealthy attack is explained. This specific attack simply introduced Trojan Horse malware to the RASc which opened an undetectable backdoor for the attacker. Then attacker used this door to get inside the LAN. Once in the LAN attacker first activated RASc by making a malicious tripping. Then attacker switched the RASc with his own fake version. Finally, attacker initiated a pulse attack to the grid all the while reporting the fake measurements to the CC. This attack shows the vulnerability of the Smart Grids to a Malware.

The mitigation scheme on the second part focused on how to detect the malware attack. The proposed scheme introduced a new device named Overseer, and also upgraded the RASc to take one extra measurement from randomly chosen device. Essentially, every cycle all RASc reports their own measurements, and the random measurement to the Overseer. At this point Overseer compares the measurements and detects any conflicts. Once a conflicts detected Overseer asks to a third RASc to double check it. Depending on the third RASc measurement Overseer detects which RASc has the problem.

Conclusions

In this thesis impact of malware to the RAS has been explored. In overall this issue has been covered in two parts. First, the seriousness of the malware attack on Smart Grids has been demonstrated. This part focused on what is possible, and how the Smart Grids exploited. Second,

an attack resilient RAS has been proposed. This part mostly focused on what can be done about the malwares.

The malware is a real problem in the Smart Grid. That can cause severe permanent damage to it while being completely hidden. The same malware can be used to do multiple different attacks to the system. Therefore, it is really important to take preventative measurements to protect Smart Grid system.

As for the proposed Attack Resilient Scheme, it showed promising results in the testbed for detection. However, the schemes need a better way to disable the infected RASc. Other than this, the scheme still has some room to improve.

REFERENCES

[1] USA. Department of Energy. “The smart grid: an introduction”. Washington, D.C.: U.S. Dept. of Energy, 2008. Print.

[2] M. Govindarasu and P. W. Sauer, "Smart grid & security [Guest Editorial]," IEEE Power and Energy Magazine, vol. 10, pp. 17-17, 2012.

[3] Lehnhoff, K. "Will Smart Communities Enable the Smart Grid?" EBV NEWS Network. N. p., 15 Dec. 2015. Web. 28 Mar. 2017.

[4] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, “Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid,” Smart Grid, IEEE Transactions on, vol. 4, no. 2, pp. 847–855, 2013.

[5] A. Ashok, A. Hahn, and M. Govindarasu, “Cyber-physical security of Wide-Area Monitoring, Protection and Control in a smart grid environment” Journal of Advanced Research, vol.5, pp.481-489, 2014.

[6] V. Kumar Singh, A. Ozen and M. Govindarasu, "Stealthy cyber attacks and impact analysis on wide-area protection of smart grid," 2016 North American Power Symposium (NAPS), Denver, CO, 2016, pp. 1-6.

[7] Mayne, D. "Merging Legacy Systems and the Smart Grid." Merging Legacy Systems and the Smart Grid | DigiKey. N.p., n.d. Web. 17 Apr. 2017.

[8] Zhang, Y., Raoufat, M. E. and Tomsovic, K. 2016. Remedial Action Schemes and Defense Systems. Smart Grid Handbook. 1–10.

[9] Coleman, K. Separation of Duties and IT Security. 2008, August 26 Retrieved April 17, 2017, from http://www.csoonline.com/article/2123120/it-audit/separation-of-duties-and-it-security.html

[10] NERC, “Remedial Action Scheme” Definition Development. Project 2010-05.2 – Special Protection Systems. 2014, August

[11] Abdelmoumene, A., Bentarzi H. “A review on protective relays’ developments and trends” May 2014

[12] C. Elisan. *Malware, Rootkits & Botnets A Beginner's Guide*. McGraw Hill Professional. 2012, September. pp. 10–. ISBN 978-0-07-179205-9.

[13] Krishnaswamy, S. "Accessible Remote Testbed for Cyber-Physical Systems Security of the Smart Grid." Order No. 10167723, Iowa State University, 2016, https://search.proquest.com/docview/1845053025?accountid=10906 (accessed April 17, 2017).

[14] Bradley, C. T. (n.d.). Wondering How Port Scanning Works? Here's the Answer. Retrieved April 17, 2017, from https://www.lifewire.com/introduction-to-port-scanning-2486802

[15] Cao, Z. (2012). New Directions of Modern Cryptography. doi:10.1201/b14302

[16] A. Orebaugh, G. Ramirez, and J. Beale, Wireshark & Ethereal Network Protocol Analyzer Toolkit. Rockland, MA, USA: Syngress, Feb. 2007.

[17] "CryptCat Project-standard netcat with encryption", 2013[Online]. Available: http://www.cryptcat.sourceforge.net.

[18] R. Kaur and A. Kaur, "Digital Signature," *2012 International Conference on Computing Sciences*, Phagwara, 2012, pp. 295-301. doi: 10.1109/ICCS.2012.25

[19] Stricot-Tarboton, S., Chaisiri, S., & Ko, R. K. (2016). Taxonomy of Man-in-the-Middle Attacks on HTTPS. *2016 IEEE Trustcom/BigDataSE/ISPA*. doi:10.1109/trustcom.2016.0106

[20] B. T. Wang and H. Schulzrinne, "Analysis of denial-of-service attacks on denial-of-service defensive measures," *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, 2003, pp. 1339-1343 vol.3. doi: 10.1109/GLOCOM.2003.1258456

[21] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," Smart Grid, IEEE Transactions on, vol. 5, no. 2, pp. 580–591, march 2014.

[22] Michael Vaughn, et.al ,Idaho Power Company, "Idaho Power RAS: A Dynamic Remedial Action Case Study".

[23] A. Srivastava et al. "Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information" Smart Grid, IEEE Transactions on, vol. 4, no. 1, pp. 235–244, 2013.