A Stuxnet-Style Attack on Layer-Two Organized Smart Grid Systems

Zachary Anderson
Department of Computer Science
Cal Poly Pomona
Pomona, CA, USA
zbanderson@cpp.edu

***Abstract*—Smart grids are a new and emerging way to think about how a power grid allocates and delivers power. Contrasted with the current power grid, smart grids have computers integrated at each of their components, allowing for more efficient power distribution and lower rates of outages. However, the integration with computers also allows for vulnerabilities to cyber attacks. In this report, I demonstrate how a smart grid substation with surprisingly simple architecture is vulnerable to a man-in-the-middle attack that prevents hijacks telemetry data and prevents accurate status data from being transmitted.**

I. Introduction

The current electrical grid, while still a marvel of engineering, is being stretched to its limits with the current population and world power consumption. In order to fix this problem, resources are being allocated to the creation and integration of a Smart Grid. [1] As shown in fig. 1, The current power grid is organized in a top-down structure, in which the power transmission system, substations, distribution lines, and customers are all dependent on one large central generation facility. Under the current system, if any of those systems fail, there is a power outage.

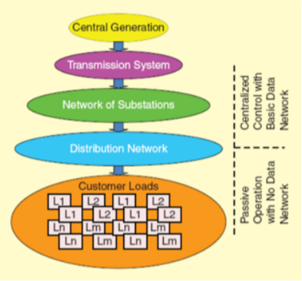


Fig. 1. Abstract layout of the current power grid and its top-down organization

Smart grids are organized in such a way that a complete power grid is composed of multiple micro-grids, each micro-grid having its own power generation, storage, monitoring, and customers it supports. Fig. 2 below shows how a smart grid and its component micro-grids would be organized.

Smart grids come with a whole host of benefits for the users and power companies, including ease of scalability and supporting multiple business models, but the largest and most interesting benefit is the integration of computers into the power grid. [2] This merging of information and energy, as, Stephen Bush puts it, allows the power grid to compensate for inefficiencies and optimize power flow. [3] Each component of the grid, including transformers, meters, and substations, have a computer that sends information back to a SCADA system at the central hub of the grid. Essentially, SCADA is the central brain of the entire grid, and in the case of a smart grid, the status of each component of each micro-grid is sent to SCADA for processing and optimization. SCADA stands for Supervisory Control And Data Acquisition and such systems have been used in a variety of other mass systems. [4] [5]

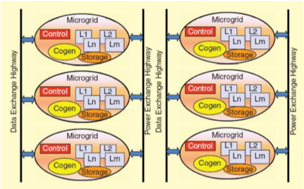


Fig. 2. Abstract layout of the smart grid and its component micro-grids

Smart grids, while they have the potential for greater optimization and performance than the current power grid, are vulnerable to cyberattacks due to the fact that they have heavy computer integration. As the components of the grid are transmitting information back and forth, it is possible for an attacker to hijack and edit the network traffic between nodes on a layer-two (data-link) level. Similar attacks have been done in the past, notably in 2010 with the Stuxnet computer worm, which compromised Iranian centrifuges and made them report incorrect status updates during operation. Additionally, power grids are a very valuable target, as rolling blackouts cause workplace losses, injuries, and disruption, and all of these allow avenues into further cyberattacks. They are also a relatively easy target, as places like Southern California in the Summer are already prone to rolling blackouts due to the increased energy consumption from air conditioners, fans, and refrigerators running as a result of the high heat. Additionally, although most cyberattacks are monetarily-motivated, there are a significant amount of cyberattacks carried out for political reasons, and it is not inconceivable that people might target a power company out of a protest to their policy. This attack can also be done on fairly simple architecture, consisting of only two devices. Figure 3 shows the layout in which this attack would be possible. In this report, I will demonstrate how an

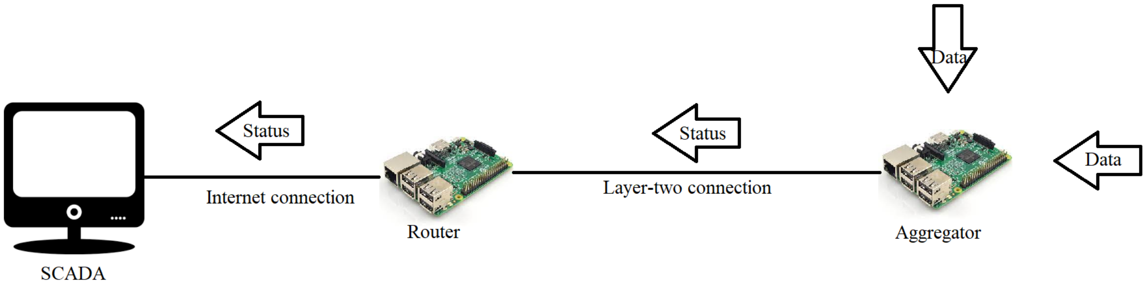


Fig. 3. Network architecture on a smart grid system where this attack would be possible

attacker can tamper with network traffic and prevent accurate data from being reported.

II. Plan

To demonstrate how an attacker can compromise and potentially damage a smart grid substation, one comprised of two major components will be simulated. These components will be a system that simulates a data collector, which in the real world would be some sort of power monitoring device that aggregates all of the data and another device that would be a system that sends the data collected back to the SCADA system. For brevity, these devices will be referred to as the aggregator and the router, respectively. These are joined together on a layer-two system. The SCADA system is represented by a python web server displaying the substation's status on an html page, visible on a third computer on the network, which does not interact with the machines in any way except for an SSH connection to the aggregator and router, in which the only interaction is starting and stopping programs that create the data flow.

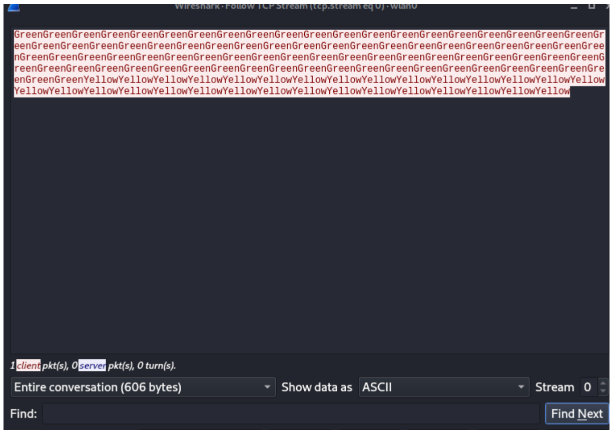
The goal of the attack is to stop accurate data from being sent from the power station system to the SCADA system, and this will be accomplished by creating a man-in-the-middle (MITM) attack that will modify the traffic between the aggregator and the router before being sent off over the internet. Our aggregator and router were both Raspberry Pis running Raspian OS. The aggregator is running a Python program that sends a temperature and a status (either green, yellow, or red) every second over an unencrypted TCP connection to the router. The temperature increases every 2 seconds and the status changes depending on the temperature. In the demonstration, between 70-100 degrees correlates with ”Green,” 101-130 correlates with ”Yellow,” and above 131 correlates with ”Red.” The temperatures and the status demarcations might be different in the real world, but these can be used without loss of generality. Once the status has been sent to the router, it then pings a web server, telling it to change its display to a green, yellow, or red image to reflect the status being sent to the SCADA system. The attacking device was another Raspberry Pi running Kali Linux, which is a Linux distribution developed by Offensive Security and comes with many useful penetration testing tools pre-installed. In order to simulate the way that an attacker could compromise this system with no prior knowledge of its inner components, three further tools will be used in order to analyze the network, devices, and traffic of the system.

A. Tools

The first tool to be used is Nmap, which allows a user to scan a network and find all of the MAC addresses, IP addresses, and hostnames of the devices on the network. [6] For the purposes of the experiment, I assigned the aggregator and router hostnames of ”aggregator” and ”router,” respectively, but it is possible to identify the devices without descriptive hostnames. More detail on that subject will be given further below.

The second tool is Wireshark, which is a network packet analyzer that allows a user to view unencrypted traffic being sent on the network. [7] One caveat is that since TCP traffic is not broadcast, that is, the network packets are only visible to the destination address, it is impossible to see the network traffic without an ARP poisoning attack via Ettercap or other means. Wireshark also allows the user to view the TCP stream of captured network traffic, an example of which is shown in figure 4.

The last tool that will be used in the attack Ettercap, which is a tool suite allowing users to execute MITM attacks between devices on a network. One of the many attacks Ettercap can perform is an ARP poisoning attack. In a layer-two network, the Address Resolution Protocol (ARP) table is responsible for mapping IP addresses into MAC addresses. ARP poisoning attacks modify the IP/MAC address mapping, which allows a computer to listen in on traffic meant for another computer by pretending that the malicious computer has the destination MAC address. [8] By using an ARP poisoning attack, one can reroute the traffic from the aggregator through the attacking machine. Ettercap also allows people to create filters using a scripting language, allowing us to modify the intercepted traffic. Writing this filter will be possible only by scanning the network traffic using Wireshark, as the filter only allows for substitution or deletion of packets that contain a specific signature. By looking at the traffic, the attacker can see the status' format as it is sent over the network. Using this traffic analysis, the attacker can write a filter that will find the status in the traffic and replace the status with the correct ”bad” data.



```
GreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGr
eenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGree
nGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenG
reenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGreenGre
enGreenGreenYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellow
YellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellowYellow
```

1 client pkt(s), 0 server pkt(s), 0 turn(s).

Entire conversation (606 bytes) Show data as ASCII Stream 0

Find:

Fig. 4. TCP stream of the data sent from the aggregator to the router

B. Method

In a real-world attack, an attacker would most likely want to choose a target that would have maximum impact. In smart grid architecture, where each micro-grid serves different loads and is completely self-contained, the best option would be to target the substation, which is the component of a power grid responsible for taking the power that has just been generated and distributing it to the industries, companies, and houses that it serves. Substations have another quality that makes them easy targets: most of the time they are unmanned. Depending on the location and security level, it might be possible for an attacker to slip into a substation, find an open Ethernet port, and connect an attacking machine to the network. Another possibility is for an attacker to either disguise as a technician to have an easier time bypassing security if there is any present, or even for an attacker to bribe a legitimate technician to plug in a machine on a network and walk away. These last two scenarios especially bring up the necessity of social engineering knowledge and how to recognize it, but as that is beyond the scope of this paper, they will not be discussed.

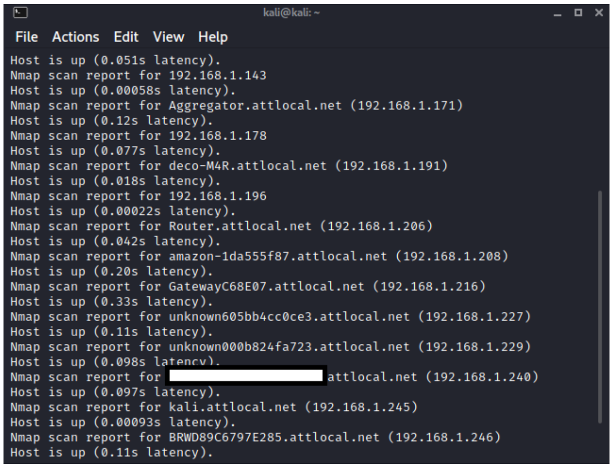
Once the target is chosen, the first step of the attack is to use Nmap to scan the network to find the aggregator and router equivalent devices (as those would most likely not be the exact names in a real-world scenario. If the hostnames are named descriptively, then the job of determining which devices are the aggregator and router is trivial. If not, analyzing the network traffic between devices can shed light on which devices should be looked at. Analysis of each MAC address can also help, as the MAC address indicates what type of system a device is.

Once the aggregator and router are chosen, all the attacker needs to do is note their IP addresses (helpfully provided by Nmap) and start up Ettercap. Then the attacker can start a MITM attack on the correct two devices, which reroutes all traffic between the aggregator and router through the attacking machine. Now that the traffic is being routed through the attacking machine, starting Wireshark will allow the TCP traffic that would normally be unavailable be seen by the attacker. By analyzing this traffic, the attacker can get a sense of what types of status is being sent between the aggregator and router, which will allow us to write the Ettercap filter in the most optimal way.

For the purposes of demonstration, the statuses sent between devices are very simplified, but in a real-world scenario, an attacker would have to constantly look at the traffic and notice patterns in how the aggregator sends status to the router. This would most likely require the attacker to stay connected and monitor traffic over long periods of time, hopefully during varying periods of strain to show the attacker what a ”good” status looks like and what a ”bad” status looks like. Once the attacker has a strong sense of what the traffic looks like, the attacker can then write the filter in such a way that it filters out the ”bad” status and replaces it with a ”good” status. Figure 6 above shows this in more detail.

III. Results

Scanning the entire network was accomplished by one simple Nmap command in a Kali Linux terminal, and the tool returned the IP addresses, hostnames, and MAC addresses of each device currently connected to the network. A few quick scrolls through the list of hostnames returned reveal the aggregator and router, as shown in figure 5.



```
Host is up (0.051s latency).
Nmap scan report for 192.168.1.143
Host is up (0.00058s latency).
Nmap scan report for Aggregator.attlocal.net (192.168.1.171)
Host is up (0.12s latency).
Nmap scan report for 192.168.1.178
Host is up (0.077s latency).
Nmap scan report for deco-M4R.attlocal.net (192.168.1.191)
Host is up (0.018s latency).
Nmap scan report for 192.168.1.196
Host is up (0.00022s latency).
Nmap scan report for Router.attlocal.net (192.168.1.206)
Host is up (0.042s latency).
Nmap scan report for amazon-1da555f87.attlocal.net (192.168.1.208)
Host is up (0.20s latency).
Nmap scan report for GatewayC68E07.attlocal.net (192.168.1.216)
Host is up (0.33s latency).
Nmap scan report for unknown605bb4cc0ce3.attlocal.net (192.168.1.227)
Host is up (0.11s latency).
Nmap scan report for unknown000b824fa723.attlocal.net (192.168.1.229)
Host is up (0.098s latency).
Nmap scan report for [redacted]attlocal.net (192.168.1.240)
Host is up (0.097s latency).
Nmap scan report for kali.attlocal.net (192.168.1.245)
Host is up (0.00093s latency).
Nmap scan report for BRWD89C6797E285.attlocal.net (192.168.1.246)
Host is up (0.11s latency).
```

Fig. 6. Hostnames of the devices and the network and their IP addresses. (One hostname blocked out in white due to it containing personal information)

When scanning the network using Ettercap (which only reveals the IP and MAC addresses and not the potentially enlightening hostnames), it is trivial to cross-reference the IP addresses of the aggregator and router with the IP addresses Ettercap found. Targeting them and choosing an ”ARP Poisoning” MITM will cause the traffic from the aggregator to flow through the attacking machine before going back into the router. Now that the aggregator’s TCP traffic is being routed through the attacking machine, a Wireshark program running on the machine will be able to see the Wireshark traffic. If the hostnames are descriptive and it is clear which IP addresses belong to the aggregator and router, finding the traffic between them is trivial. However, if the hostnames are not descriptive, this is where a brute-force approach will be necessary. By analyzing the traffic between each device, it will be possible, albeit tedious, to determine which devices are the aggregator and router.

By following the TCP stream between the devices, an attacker can see what type of network traffic the aggregator is

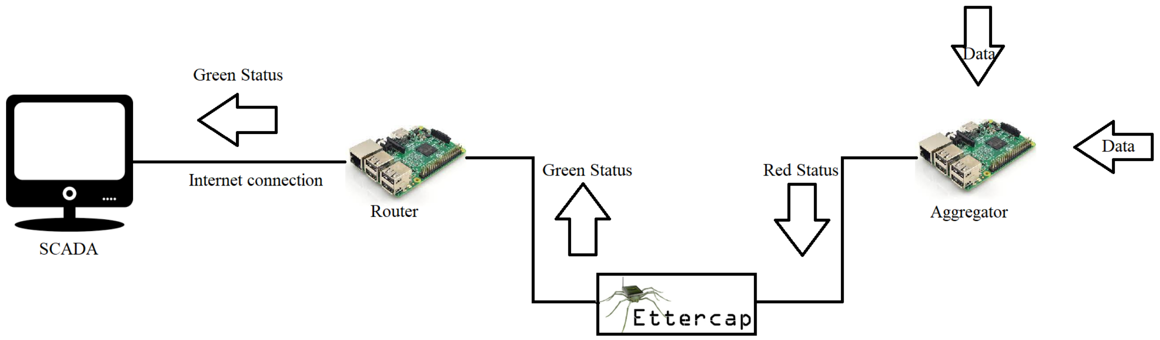


Fig. 5. Network on the subsystem with MITM in place

sending. From here, the traffic and the ports that the two are communicating on can be analyzed. Wireshark shows that the two devices are communicating on port 8000 and the traffic consists of a number, a colon, and either a ”Green,” ”Yellow,” or ”Red” message. This, in the demonstration, corresponds with a temperature and a status. In the real world, this might not be exactly how the status is laid out, but packet analysis will provide insight into how the traffic is organized. This analysis gives us all the information needed to build the Ettercap filter, which will filter out all traffic containing a ”Yellow” or ”Red” status and replace it with a ”Green” status and a temperature of 90 degrees.

Ettercap contains a built-in scripting language that is syntactically similar to C, and that is what will be used to build the filter.

The code for the filter is below:

```
if(ip.proto == TCP && tcp.dst == 8000)
{
    pcre_regex(DATA.data,
    ”^[:digit:]*\:Yellow”,
    ”90:Green”)
    pcre_regex(DATA.data,
    ”^[:digit:]*\:Red”,
    ”90:Green”)
}
```

The top line of the filter checks if the traffic is TCP traffic, as any other traffic should be left alone. The port they are communicating on is shown from the Wireshark scan, which allows us to filter the traffic further. The ”pcre_regex” function searches in the traffic for a regular expression, in this case a number followed by a colon and then followed by either ”Yellow” or ”Red,” and replaces any matching strings with Similarly, the ”replace” function replaces one string with another. Together with the search for a ”Red” status, this Ettercap filter replaces the ”Red” and ”Yellow” strings with ”Green,” hiding any potential bad statuses.

When not running the MITM attack, the aggregator and router were logging the data that it sent and it received, respectively. Figure 7 shows the comparison between the data sent and received.

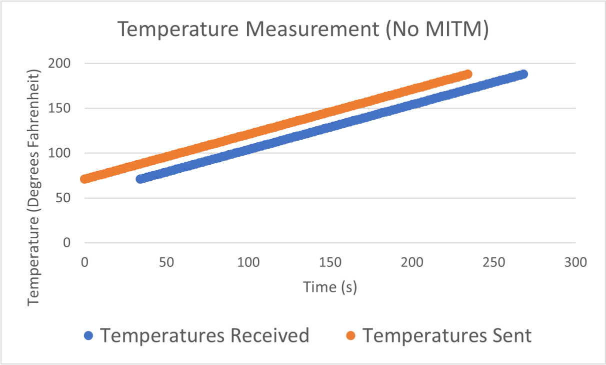


Fig. 7. Graph of the temperatures sent and the temperatures received without the MITM active

After the MITM attack has been loaded onto Ettercap and the filter has been applied, a look at the display showed that the router was receiving ”Green” status messages from the aggregator even when the status on the aggregator was set to ”Yellow” or ”Red.” The temperatures received also differed from the temperatures sent, as can be seen in figure 8.

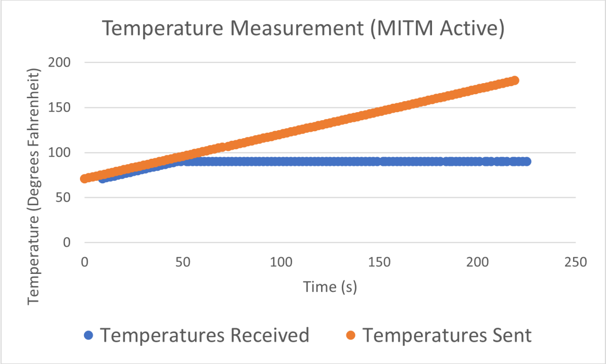


Fig. 8. Graph of the temperatures sent and the temperatures received without the MITM active

After unloading the filter and re-ARPing the aggregator and router however, the machines did not connect back to each other. Restarting the data transfer programs on both the aggregator and router fixed the problem. This problem only occurred when the filter was applied and not during the Ettercap session without the filter. The reason for this is unknown. However, in the context of a real-world attack, this might not be a significant issue. Consider that the goal of the attacker is to take down a substation by hiding its status and preventing protective measures that come from a substation having a bad status. If the substation overheats or fails in some other way, the drop in communication between the aggregator and router will most likely be a small issue compared to the substation's breakdown.

IV. Mitigations

There are a few ways to prevent a MITM such as this one from happening. One of the advantages of the attack presented here is that it does not require the attacker to compromise any devices on the network due to its architecture. One way to make this attack harder would be to have one device that acts as both the aggregator and the router as opposed to having dedicated devices for each. A setup like this would force the attacker to penetrate the network and access the actual device, which could be very difficult depending on the security that the device has, and would also require that the attacker either intentionally add a bug to the code in a malicious update or modify the drivers in some way such that the network traffic is modified in the way that the attacker wants. This attack would be even more reminiscent of the 2010 Stuxnet virus. [9]

Another mitigation option is having the devices connected on at layer-three level. The core of this attack strategy is ARP poisoning, which is only possible on a layer-two network. If the aggregator and router were connected at the layer-three level, or were potentially connected by a layer-three switch, then Ettercap would not be able to poison the ARP cache and the attack would be rendered impossible. However, the simplest and maybe best way to prevent a MITM such as this one is to encrypt the network traffic that the aggregator sends to the router. Wireshark will not be able to view the encrypted traffic, leaving the attacker at a dead end as to how he/she can create the Ettercap filter. Further mitigation techniques have been developed. B. Prabadevi and N. Jeyanthi developed a framework that analyzes timestamps and has abnormal packet detection [10]. Another framework has been developed by researchers at the Kwame Nkrumah University of Science and Technology in Ghana that is based on a machine learning approach. [11] If a smart grid's network architecture is set up in the same way as the experiment in this paper, it would behoove the smart grid company to invest in one of these frameworks or in some sort of encryption to prevent these types of attacks form occurring.

V. Related Work

At a previous REU site for the National Science Foundation, students were able to exploit a system on a virtual smart grid architecture by using Metasploit to hack into the data logs on a smart grid device and tamper with them before putting them back in the database, causing the smart grid's central database to crash. [12] If data is stored locally on a smart grid substation, a modification of this attack could be used to affect the SCADA system. This would be more suitable to a large-scale attack with the plan of taking down a huge number of stations however, as unlike the attack demonstrated in this paper, an attack on SCADA would not be quiet at all. Another group of students at an REU site in 2018 developed an attack called a pseudo-node attack in which they compromise one device on the network on a smart grid and send packets to every other device on the network, flooding it with traffic and slowing down network activity. [13] This attack, again, does not have the advantage of being quiet, although it can be an effective attacking method in situations that do not require much stealth or as a part of a larger attack.

VI. Future Work

In the attack demonstrated, the attacking machine simply caused the temperature readings to plateau at 90 degrees. In a real-world environment, the SCADA system will most likely have some form of anomalous process monitoring, and receiving a constant temperature of 90 degrees with no variation at all will most likely be flagged as anomalous. Integrating a machine learning algorithm to figure out what times and days have different sorts of temperatures and variations could make the attack less detectable. This would also most likely require a more sophisticated MITM tool than an Ettercap filter, as they do not contain the logic and power necessary to follow the advice of a machine learning algorithm.

The attack also caused the two devices to no longer communicate with each other after the attack finished. While it is still likely that this would not be a big issue in the context of a legitimate attack, integrating this attack with sending TCP reset packets or somehow flushing the ARP cache of the aggregator and router after the attack could allow the attacker to exit the network quietly.

VII. Conclusion

With power consumption growing as populations increase and remote work and online activity grow, smart grids and the technologies they utilize offer a multitude of advantages in today's world. However, with all new technologies, it is important to consider the roles that computers have and the attack vectors they offer to bad actors. Computers will, by definition, always introduce cyberattack vectors, and in a developing industry like this one, protections against them might not be available or widely-known. Smart grids have the ability to make power distribution much more efficient and effective, but only if proper precautions are taken against all attacks possible. The experiment in this paper demonstrated an attack that anyone with a Raspberry Pi, a few tools, and some background knowledge could do, and while there are mitigations against this attack available, mitigations mean nothing if they are not implemented, turned on, or taken seriously by system administrators and system architects.

VIII. Acknowledgements

Firstly, I would like to credit the National Science Foundation for sponsoring this REU site. I would also like to thank Dr. Tingting Chen for managing the entire REU site and being able to answer any and all questions about general research. I would also like to thank Dr. Sean Monemi for his insight, advice, and knowledge of smart grid technologies and the data, research, experiments, and help he shared. I would also like to thank him for reading and editing my paper as I was drafting and writing it. Additionally, I would like to thank Philip Askew for advice on layer-two architecture and ARP spoofing, pointers on networking and general man-in-the-middle advice and theory, and for help in debugging my experiment.

REFERENCES

[1] United States Department of Energy, "*The Smart Grid*," December 2019. [Online]. Available: https://www.smartgrid.gov/the_smart_grid/smart_grid.html

[2] R. Rylatt, J. R. Snape, P. Allen, B. Mahdavi Ardestani, P. Boait, E. Boggasch, D. Fan, G. Fletcher, R. Gammon, M. Lemon, V. Pakka, C. Rynikiewicz, M. Savill, S. Smith, M. Strathern, and L. Varga, "Exploring smart grid possibilities: A complex systems modelling approach," *Smart Grid*, vol. 1, pp. 1–15, 08 2015.

[3] S. F. Bush, *Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid*. Hoboken: John Wiley & Sons, 2014.

[4] J. Tautz-Weinert and S. J. Watson, "Using scada data for wind turbine condition monitoring - a review," *IET renewable power generation*, vol. 11, no. 4, pp. 382–394, 03 2017.

[5] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water scada systems—part i: Analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1963–1970, 2013.

[6] G. Lyon, "Nmap: the network mapper." [Online]. Available: https://nmap.org/

[7] W. Foundation, "Wireshark: Go deep." [Online]. Available: https://www.wireshark.org/

[8] S. Y. Nam, D. Kim, and J. Kim, "Enhanced arp: preventing arp poisoning-based man-in-the-middle attacks," *IEEE Communications Letters*, vol. 14, no. 2, pp. 187–189, 2010.

[9] D. Kushner, "The real story of stuxnet," accessed 8-4-2021. [Online]. Available: https://spectrum.ieee.org/the-real-story-of-stuxnet

[10] B. Prabadevi and N. Jeyanthi, "Tscba-a mitigation system for arp cache poisoning attacks," *Cybernetics and Information Technologies*, vol. 18, no. 4, pp. 75–93, 2018.

[11] J. J. Kponyo, J. O. Agyemang, and G. S. Klogo, "Detecting end-point (ep) man-in-the-middle (mitm) attack based on arp analysis: A machine learning approach," *International Journal of Communication Networks and Information Security*, vol. 12, no. 3, pp. 384–388, 2020.

[12] S. Monemi, D. Kamand, R. Thayi, S. Luong, and T. Venrick, "Smart grid cyber test bed development," 2016.

[13] S. Monemi, R. Meguerdijian, P. Lam, E. Portillo, D. Marantz, C. Flores, T. Lam, and T. Kim, "Pseudo-node attack on smart grid," 2018.