Analysis of GridGear Solutions' Smart Metering System

December 11, 2016

Group #8

Connie Ma *56047129*
conniemob@gmail.com
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

Derek Chan *33184128*
derek.chan@alumni.ubc.ca
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

Jake Larson *32333122*
larsonja@alumni.ubc.ca
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

Pascal Turmel *19449131*
pcturmel@gmail.com
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

***Abstract*—Our analysis of the GridGear smart metering system comprised of an in-depth review of the company's pre-production web application and backend (software), as well as their metered data aggregation tool on a Raspberry Pi Data-Collector (firmware/hardware). To test the confidentiality, integrity and availability of system data, we made use of brute-force password retrieval techniques on the GridGear web application, man-in-the-middle techniques on the HTTP connection, an arbitrary data modification demonstration, and a proof-of-concept demonstration of spoofing their firmware scripts on an unverified device. In our testing, we discovered several components that were completely unsecured and could potentially allow malicious users and third-party hackers to discretely exploit the integrity of GridGear's data. As power data can only be fully verified by performing on-site consultation of metering hardware, these security flaws may prove to become substantial liabilities that could place GridGear's professional credibility at risk. Based on our findings, we recommended a complete overhaul of GridGear's server and database authentication for software security, a layered approach to the fetching and writing of all data, as well as data encryption and integrity verification of all *in situ* data collected on the hardware and persisted to their servers.**

I. Introduction

A. Problem and significance of the problem addressed by analysis

In the growing sector of clean energy and smart home technology, data granularity shapes the dynamic between its integration into a homeowner's daily life and its vital role in the Internet of Things (IoT). As such, innovative hardware and software such as GridGear's smart meter and data analytics tools become invaluable to utility companies and their end users. However, as aptly demonstrated by the recent compromisations of IoT devices [1], any insecure smart home device with a network connection can become a vulnerable liability to stakeholder assets, and that is the problem we intended to address in our analysis of GridGear's smart metering system.

GridGear's smart meter (SM) technology captured industry-grade energy consumption measurements at high sampling frequencies, and as a result the biggest asset at risk of threat was all **metered data**. A second asset at risk was GridGear's **professional reputation** as a manufacturer of reliable and secure SM products. This is crucial because GridGear's product tailored to utility companies that value data **integrity and availability** in order to deliver correct billing reports to customers.

From the value of the aforementioned assets, we determined that the biggest **risks** from threat agents such as malicious hackers, rival service providers, and unreliable customers lies in the potential compromise of the GridGear database and any unauthorized manipulation of data. In every case, the asset value of GridGear's reputation of reliability and accuracy as a smart metering service will be reduced.

B. Summary of the system and related work on similar systems

The object of our analysis was GridGear's flagship smart metering system, a projected key deliverable to North American utility companies. This system under test (SUT) consisted of a smart meter hardware device, Data-Collector module, associated web application data visualization tool, and backend database for metered data management.

Recognizing that similar smart metering systems existed on the market, we referred to a previous cohort's analysis of the Neurio Home Energy Monitor as a starting point for test synthesis [2]. We referred to an academic paper written by UBC scholars (section III of this report) for proof-of-concept testing of generic smart metering systems depicting applicable adversary models and attack vectors we could employ in our own testing procedure. For technical analysis of GridGear's web application, we referred to various tools and reports documenting key methodologies used in penetration testing. Finally, we referred to a public report from BC Hydro for existing power meter privacy standards [3].

C. *Summary of methodology*

By following these related works and applying them to our own proposed methodologies, we successfully undermined the security of GridGear's system by incrementally examining its various subcomponents. We compromised GridGear customer and administrator accounts by performing an exhaustive password search on the GridGear web application using BurpSuite. We employed ARP spoofing to demonstrate the MITM vulnerabilities of the HTTP web application. We also found plaintext root credentials for GridGear's server on the easily-compromisable Data-Collector filesystem. Finally, we also impersonated GridGear's Data-Collector by running its data aggregation scripts on our own computers, allowing us to receive data from nearby power meters.

D. *Summary of obtained results, conclusions from results, and recommended solutions*

In all of our findings, we were merely interested in diminishing the confidentiality of GridGear data. However, we uncovered that it was fairly simple to additionally distort the integrity of data by spoofing it in the database. We concluded from our testing that the GridGear system had significant security flaws in terms of data confidentiality and system integrity. As discussed in following sections, these flaws were non-trivial and required immediate attention from a GridGear web developer or software security administrator, should the product be made shippable to a production environment.

For software, we recommended a redesign of the current GridGear web authentication scheme to follow stricter design principles of psychological acceptance and least privilege in terms of client- and server-side authentication. Specifically we recommended the implementation of HTTPS protocol in all web application communication and a password policy to ensure the use of strong passwords. For hardware, we recommended implementing basic encryption and message authentication of data at the collection level, which will ensure the confidentiality and integrity of data passing through the Data-Collector script to the GridGear servers. We also recommended changing the authentication scheme between the Data-Collector and server to use public keys instead of passwords, and to limit the Data-Collector's privilege in order to follow the principle of least privilege.

E. *List of contributions*

We contributed the following work to make simple improvements to GridGear's system:

- **Improved separation of system privileges**
  We created a "Data-Collector" (*datacollector*) user account on GridGear's server and delegated it write-access to a sole directory on the filesystem. Although this is not an ideal solution (conventionally, no Data-Collector should have access to the filesystem), this is sufficient improvement over the exposition of the GridGear server's root credentials.
- **Addition of asymmetric key credentials**
  To prevent unauthorized writes to the GridGear server, we created a public/private key pair on each Data-Collector and added the public key to the server's `authorized_hosts` file. This provided a scheme for GridGear's administrators to easily add and revoke privileges to deployed Data-Collector devices.
- **Securing Data-Collector credentials**
  We altered the login credentials of the Data-Collector so that these were no longer the Raspberry Pi defaults.

II. Analyzed System

Listed below are distinct stakeholders that interact with GridGear's smart metering system, alongside specific technologies made accessible to each entity:

- **Customers:** Typically utility companies or installers, these entities are direct customers of GridGear that have purchased a subset of GridGear's meters to monitor the electrical power usage of their end users. Customers have access to purchased **smart meters** and Raspberry Pi **Data-Collectors**, in addition to an account with the GridGear **web application** that can request visualizations of their aggregate power data. Smart meters are also typically installed by GridGear's customers.
- **End users:** Entities whose homes or properties consume electrical power provided by GridGear's customers. End users have non-physical access to the **smart meter** device, as it is installed in their homes and properties. End users cannot tamper with the smart meter hardware.
- **Administrators**: GridGear employees. Administrators have are responsible for management of **all GridGear systems and devices**, which includes Data-Colector facilitation and server administration. They have full access to the entire system.

Listed below are the key components that comprise GridGear's smart metering system, followed by an abridged depiction of the dataflow between each component (supplemented by the graphic in Figure 1):

- **Smart meters**: devices that collect power data from an end user's main line in the electrical panel [4]. Live power readings are collected via industrial-grade current transformers, and the data is then transmitted to a utility's **Data-Collector** over radio frequency (RF) when polled by the Data-Collector. *(hardware)*

- **Raspberry Pi Data-Collector**: using a RF transceiver USB dongle, a single Data-Collector will request and receive **smart meter** data, aggregating power data from multiple smart meters. The Data-Collector then periodically uploads its data as CSV files to the **GridGear server** via SFTP. *(hardware, software)*
- **GridGear server**: a single server entity that stores power data of all of GridGear's customers in its database. The server additionally runs GridGear's Django web application and contains Python scripts that generate graphical visualizations of the CSV data received from various **Data-Collectors** to display for authenticated GridGear customers. *(software)*
- **Client computers accessing the GridGear web server**: interface for users and administrators to log into the GridGear web application. Customers can only view their respective meter data. Administrators can view all data and configure the **Data-Collector's** sampling and reporting rates. *(software)*

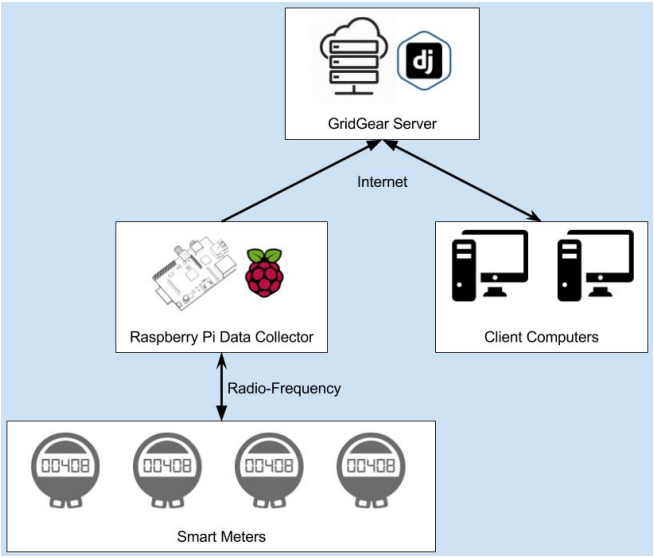

Figure 1: Data flow in GridGear's Smart Metering Network

III. Related Work

A similar system analysis was demonstrated by a previous cohort's project with the Neurio Home Energy Monitor (Neurio) device. Although Neurio's main customers are individual homeowners, the security concerns of Neurio are nearly identical to those of GridGear. In both cases analysis can still be performed on the system by evaluating its handling of data confidentiality, integrity, and availability.

The previous cohort used guessing methods against Neurio's web application to discover unsecured HTTP GET/POST requests that could leak the data of one user to a second unauthorized user. We took a similar approach by also analysing GridGear's web application's vulnerable HTTP protocol. As the firmware-level interfacing was different between a Neurio sensor and GridGear's Data-Collector tool, we took separate measures to compromise the confidentiality and integrity of the system.

In academia, we referred to a paper by UBC's Farid Molazem Tabrizi and Karthik Pattabiraman which proposed key vulnerabilities in an abstract smart meter system [5]. The examples outlined include spoofing metered data with man-in-the-middle vectors and eavesdropping on data traffic between the meter and the server. We applied some of these examples in our own analysis methodology on GridGear's system. Additionally, we also referred to Henrique Danta's work for insights on how to analyze the reliability and robustness of a smart meter device [6]. We considered the techniques described in his work when analyzing the security of all communication channels in GridGear's system.

In industry, we considered BC Hydro's public investigation report by the Information & Privacy Commissioner for BC as a reference for any legal requirements for smart meter data privacy [3].

IV. Analysis Methodology

A. System Analysis & Penetration Testing Methodology

Our analysis of GridGear's system involved investigating the four numbered vectors indicated in Fig. 2.

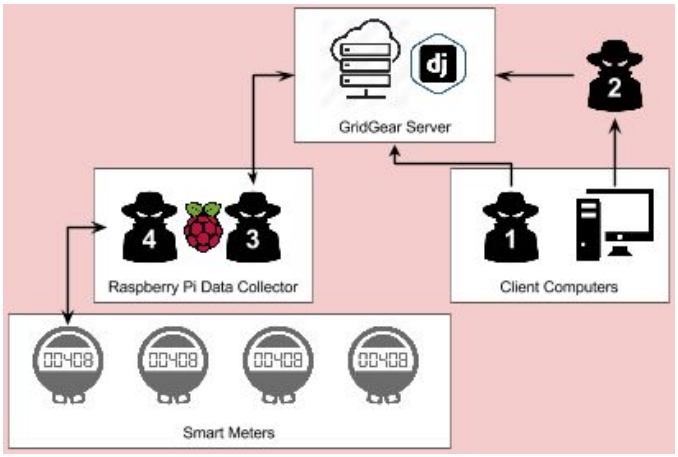

Figure 2: Our team's analysis vectors on GridGear's system

1. *Client Impersonation (Password Guessing):*

GridGear's web application greeted its users with a simple login page. Our inspection of this component revealed that unlimited login attempts could be made. Using BurpSuite as a web proxy in order to bypass the Cross-Site Request Forgery (CSRF) tokens, we automated the login process by drafting a POST request for the login page, filling in the login credentials, and then sending the request to GridGear's server.

Through the use of BurpSuite, we were able to successfully find the login password for a test account. This demonstrated the possibility of a malicious user running the same password cracking attack on GridGear's live system.

2. *Server-Client Interception (MITM Analysis):*

GridGear's web application employed HTTP. To demonstrate the potential for a man-in-the-middle (MITM) attack, we used ARP spoofing on a local area network with one router, one MITM computer, and one victim computer [7]. The MITM used two instances of the *arpspoof* Linux program to perform the following [8]:

- To the router, the MITM computer associates its own MAC address to the victim computer's IP address
- To the victim, the MITM computer associates its own MAC address to the router's IP address

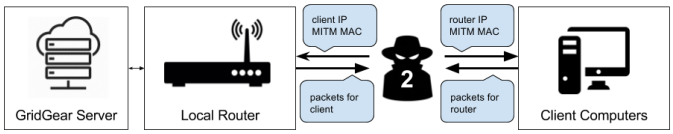

Figure 3: Analysis #2: MITM configuration using arpspoof

The result of this configuration, as shown in Figure 3, was that the MITM computer was able to intercept all network packets sent between the victim and the router. Since HTTP packets encode all information in plaintext, any activity performed by the victim computer on GridGear's web application would have its data compromised by the MITM. We could then extract the GridGear login credentials and the Web Application's session ID with Wireshark, demonstrating this component's insecure handling of data confidentiality.

3. *Data-Collector Impersonation to Server:*

Our analysis of the Data-Collector began with finding two simple methods to access the Data-Collector's filesystem:

A. Removing the Data-Collector's SD card and inserting it onto a computer to freely view the entire filesystem
B. Logging into the Data-Collector with SSH on the default SSH port (22) using the default Raspberry Pi credentials: username *pi* and password *raspberry* [9]

Once we had access to the Data-Collector's filesystem, we were able to locate GridGear's main program, which was a plaintext Python script. Inside the script were the root credentials for GridGear's single server, which were intended to be used with SFTP to upload metered data to the server. We were able to use these same credentials on our own computers to SSH into GridGear's server as the root user. Once we were on GridGear's server, we were able to find the filepath where all customer data was stored. As a proof-of-concept attack, we modified existing smart meter data and viewed our change on the web application, as shown in Figure 4. At this point, it would be trivial to also view and modify the meter data of all other customers, thus compromising the integrity and confidentiality of the system.

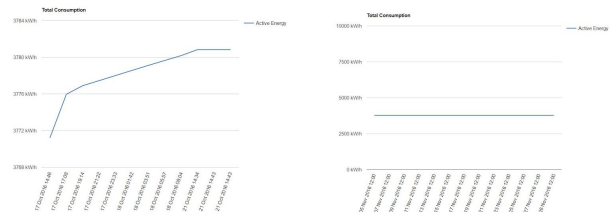

Figure 4: Root access allows arbitrary data modifications. Original (left), modified (right).

4. *Data-Collector Impersonation to Smart Meters:*

The Data-Collector uses a RF protocol implemented in a Python script for communication with the GridGear meters. We simply ran the same script on our own computers and succeeded in requesting and receiving smart meter data.

B. *Ethical Considerations*

The scope of our ethical considerations were mainly focused on the ACM-adapted principles, including elements such as contributing to society, divulging information honestly and responsibly, operating with integrity, honoring confidentiality, and avoiding harm to others (see Appendix A). As GridGear's software product was not yet released to production, we were able to exercise more freedom and creativity in our various threat models and proof-of-concept attack vectors. Our analysis primarily aimed to identify scenarios that may allow future users to exploit current system vulnerabilities. We additionally maintained confidentiality by communicating on private channels for all project-related activities, and honored GridGear's trust by discussing our plan for analysis in person.

C. *Risk Management*

GridGear provided us with a test system (smart meter and Raspberry Pi Data-Collector device), root access to the GridGear server, and test credentials for their web application. Since no production customers were using the SUT and backups of all test data were available at length, we were reassured that our testing would not compromise the data in the database.

Despite having permission to freely tamper with the system, the team discussed the plan for analysis with GridGear to ensure that no administrative sanctions would be incurred, with summary of all discussions documented in confidential email channels. To manage legal risks, all entities involved directly with the GridGear analysis project signed the CPEN 442 Project Authorization Form stating that GridGear would allow the security analysis of their system and responsible disclosure of analyses.

V. Results

As demonstrated from all analysis vectors discussed in the previous section, we concluded from our testing that the GridGear system had very significant security flaws in terms

of data confidentiality, system integrity, and general violation of secure design principles. We found these flaws to be non-trivial and requiring immediate attention from a GridGear web developer or software security administrator before the product be made shippable to a production environment. Given that the most basic of tutorials and tools accessible on the Internet could successfully compromise the security of the GridGear system, we had many observations and recommendations to make, the discussion of which will be covered in the following sections.

VI. Discussion of the Results

A. Interpretation of the Results

A major factor in the security compromises found in GridGear's system stemmed from the fact that the company is a small startup in its early stages of growth. As such, GridGear's priorities were favoured towards marketable features, such as a pleasant web interface, the ability to graph historical data, and the automated collection of power data. Security features required development time, which would take away from producing and enhancing visible user features.

In addition, there were no strict regulations on security implementations for smart meter data. In BC Hydro's analysis of the privacy of their smart meters, the only obligation under the *Freedom of Information and Protection of Privacy Act* was to make "...reasonable security arrangements against such risks as unauthorized access, collection, use, disclosure, or disposal" [3]. The lack of concrete guidelines that defined *reasonable security arrangements* resulted in the weak and non-standard security implementations being employed in novel smart metering systems, such as GridGear's.

B. Adversary Models

From our cited research documents, we narrowed down two specific adversary models against GridGear's system:

Malicious Utility Company Customer:

- **Objective:** modify metered data (eg. reduce power consumption for a lower power bill)
- **Initial capabilities:** network access to web application, potential network access to Data-Collector filesystem
- **Capabilities during the attack:** root privileges to modify any metered data on GridGear's server

Personal Espionage:

- **Objective:** unauthorized viewing of metered data
- **Initial capabilities**: network access to web application, potential network access to Data-Collector filesystem, RF access to smart meters
- **Capabilities during the attack:** MITM attacks on HTTP to compromise user passwords, online exhaustive password search against web application login page, root privileges to view any metered data on GridGear's server, RF interception of smart meter data transmission

C. Principles of Designing Secure Systems

We noted that several principles of secure system design were violated by both GridGear developers and administrators:

- **The principle of least privilege** was violated by GridGear's Data-Collectors, as these devices had root access to the server despite only requiring write access to a specific filepath. As soon as a Data-Collector's filesystem was compromised, the inappropriate privileges allowed us to execute arbitrary operations on the server. Example operations included data manipulation, data mining, malware installation on the server, and functional server reformation.
- **The principle of open design** was violated by the RF dongles used to transmit data between the Data-Collector and the smart meter. These dongles used a proprietary protocol with no encryption. The obscurity of the RF channels was the basis of securing GridGear communications, yet the mechanism lacked official open documentation.
- **The principle of psychological acceptability** was also violated by the use of HTTP and weak password policies. Users are accustomed to having HTTPS used for secure applications, and having minimum password strength requirements when registering. The lack of these basic security measures may make a user mistrust or potentially abuse the system's security.
- **The principle of defense in depth** was violated on GridGear's server, as there was no additional level of defense on the server once an adversary obtained root access. All the data was stored in plaintext in the same directories with obvious names, making it easy to view and alter the data of any GridGear customer.

VII. Recommendations

A. Software Recommendations

For GridGear's software components, we recommended reformatting of the current GridGear web authentication scheme to follow stricter design principles of **psychological acceptance** in terms of client- and server-side authentication.

Specifically, this would entail the implementation of HTTPS protocol in all web application communication for secure data handling, and a stricter password policy with length and character requirements to enforce the use of stronger passwords. Additionally, as a secondary measure, we

recommended the implementation of an API layer to facilitate request timeouts when a user is performing too many login attempts within a fixed period of time. This will prevent or at least delay the success of an automated brute-force password guessing vector akin to the analysis we conducted.

We made these recommendations as they are key to developing a secure web application authentication scheme, and because they are used in similar systems for the better handling of data. We saw no alternatives for forgoing the implementation of the authentication measures, as they are absolutely necessary to the design of a trusted system. We did however mostly limit our recommendations to the application-level, as database-level integrity checks may be too complex for the system at this point in the development process.

B. Hardware Recommendations

For GridGear's hardware components, we recommended better exercise of the principle of **least privilege** by reducing the Raspberry Pi's server privileges. This was already discussed at length in the section of this report detailing our project contributions, but to reiterate we created a separate Linux user group for the Data Collectors with limited privileges to one specific data directory on the GridGear server.

We also recommended some form of basic encryption and message authentication of data at the collection level, which would serve to verify and preserve the confidentiality and integrity of data communication between the Data-Collector and GridGear's servers. To be more specific, this would demand that the meter data pushed by the Data-Collector to the server be encrypted such that, even if the Data-Collector is compromised, it will not prevent the data from being corrupted—a necessary level of **defense in-depth**.

Lastly, we recommended changing the authentication scheme between the Data-Collector and server to use public keys instead of passwords, to further enact the principle of **psychological acceptability**. Furthermore, we *highly* recommended that the root username and password be removed as hard-coded values from the Pi's on-board script. The lack of security in such an action spoke for itself.

Again, these recommendations were chosen because they were fairly necessary to ensure at least the most basic level of security without introducing any added complexity to the usability of the system. Ease of implementation also influenced these recommendations, as these Data-Collector recommendations would be much simpler to implement over making changes to the physical smart meter's security components. We also considered recommending more open documentation at the hardware level to practice the principle of open design, but decided that was more of a measure to be handled at the discretion of GridGear and their partners.

VIII. Conclusion

In the growing sector of clean energy and smart home technology, innovative hardware and software such as GridGear's smart meter and data analytics tools will become increasingly relevant to both homeowners and utility companies. During product development, incorporating secure design principles and evaluating security metrics with regards to data security will place GridGear Solutions at an advantage above its competitors. With the interest of GridGear's future impact of technology on society in mind, our analysis provided a conclusive evaluation of all risks to software and hardware security that we found during our various tests. In this report we have duly identified many of GridGear's system vulnerabilities, their associated risks, the priority of each risk in terms of impact and cost, as well as recommended solutions for mitigation and prevention in the context of secure design principles.

We extend our thanks to the staff of GridGear for providing us with pieces of test hardware and select access to components of GridGear test software, as well as our CPEN 442 instructor and teaching assistants for valuable advice on conducting our analysis project.

References

[1] J. Wagstaff and J.R. Wu, "After cyber attacks, Internet of Things wrestles with making smart devices safer," in *Reuters Technology News*, Singapore/Taipei, 8 Nov 2016.

[2] Neurio Technology Inc., "Neurio Home Energy Monitor," http://neur.io/products/, 2016.

[3] E. Denham, "INVESTIGATION REPORT F11-03", BC Hydro, Vancouver, BC, December 2011. Available: https://www.oipc.bc.ca/investigation-reports/1244

[4] GridGear, "Single Channel Solid State kWh Meter for Submetering," http://gridgear.ca/images/GG1-S%20Brochure%202015.pdf, 2016.

[5] F. Tabrizi and K. Pattabiraman, "A model for security analysis of smart meters", *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, 2012.

[6] H. Dantas, "Vulnerability Analysis of Smart Meters," M.S. thesis, Dept Comp. Eng., Delft Univ. of Tech., Delft, Netherlands, 2014.

[7] occupytheweb, "How to Conduct a Simple Man-in-the-Middle Attack," Wonder HowTo, 2015. Available: http://null-byte.wonderhowto.com/how-to/hack-like-pro-conduct-simple-man-middle-attack-0147291/

[8] D. Song, "arpspoof(8)", Linux man page. Available: https://linux.die.net/man/8/arpspoof

[9] L. Clay, "Linux Users," Raspberry Pi Documentation, 2016. Available: https://www.raspberrypi.org/documentation/linux/usage/users.md

Appendix A: Project Code of Conduct

In order for our analysis procedure to remain ethically sound, the team adhered to several guidelines throughout our pen-testing, which were adapted from the principles of ethics illustrated in the ACM Code of Conduct.

Herein the term *stakeholder* will refer specifically to any individual belonging to entity groups of either: **(A)** our project group, **(B)** GridGear Solutions, **(C)** the current cohort and staff of our CPEN 442 course not directly involved in the analysis project, or **(D)** industry professionals invited by Dr. Beznosov to the end-of-term project competition. The term *we* will refer exclusively to members of stakeholder entity **(A)**, unless otherwise stated.

1. *Contributing to society and human well-being*
   Understanding the innovation of the GridGear product and its impactful contribution to the clean energy and sector, we performed all of our testing with the goal of betterment of the SUT in mind. Betterment of the system advances the technology and its usefulness to society and human well-being.
2. *Avoids harm to others*
   We actively demonstrated in testing how necessary it is to maintain the confidentiality and integrity of GridGear system data, specifically where it concerned the privacy of homeowners and the integrity of the utility companies that serve these homeowners. Integrity of smart meter data also protects the professional reputation of GridGear as a reliable product developer and credible service provider.
3. *Is honest and trustworthy*
   We conducted our testing transparently and exercised responsible disclosure in all information transactions with stakeholders in the project, specifically by engaging in private email discussions, phone conversations, and face-to-face meetings at established locations (either the GridGear facilities or at UBC).
4. *Is fair and takes action not to discriminate*
   Each member of our group contributed equally to the analysis and gave thorough input in all proceedings of the project, and we fairly considered all points and questions raised in formal caucus with individuals in stakeholder entity **(C)**.
5. *Honors property rights, including copyrights and patents*
   As we are analyzing GridGear's copyrighted product, we make the priority of the analysis to be one that benefits the company, and respectfully refer first and foremost to the developers and creators of the GridGear of stakeholder entity **(B)** for information dissemination of their smart metering system and any confidential materials and data we are provided with in the process of our analysis.
6. *Gives proper credit when using the intellectual property of others*
   For each third-party tool that we used in our analysis, we performed due diligence on credibility of the resource and cited its creators and source websites in our project references. We additionally disclosed our technical methodologies in detail to all stakeholders at regulated intervals.
7. *Respects other individuals' rights to privacy*
   As touched upon in (5), we prioritized the confidentiality agreement that we signed before we undertook the analysis of the system, and recognized any private information that was disseminated to us by stakeholder entity **(B)**. This specifically included any private application usernames, passwords, private keys, hardware devices, private email addresses and phone numbers, code repositories, and information in the GridGear database.
8. *Honors confidentiality*
   We exercised responsible disclosure and proper use of private resources that were given to us in the dissemination of information to groups **(C)** and **(D)**. See Appendix B for further details.

Appendix B: Responsible Disclosure

A. Summary of disclosure contents

To exercise responsible disclosure, we submitted to GridGear selected sections from our formal report verbatim; namely, the portions of the report pertaining to our **testing methodology, subsequent conclusions to our findings, and recommendations** that we have made for obtaining more secure practices. A verbatim submission is ideal because it does not introduce any ambiguities to the text, and if questions arise between any stakeholders we will be able to provide a uniform explanation.

Redacted portions of this report from our submission to GridGear were sections pertaining to the **introduction of the project, GridGear system analysis, related systems analyses, conclusion of the project, and any appendices**. This is because the information provided in those sections is fairly extraneous in its usefulness to GridGear. This information was however presented in full to Dr. Beznosov and the current CPEN 442 cohort, in addition to the industry professionals invited by Dr. Beznosov to the end-of-term project competition.

B. Summary of debrief with GridGear

Our main point of contact at GridGear is with the CEO of the company, Ilya Radetski, who may be independently reached at ilya@gridgear.ca (office tel. +1.888.512.1392). Through email and in-person correspondence, he has disclosed to us that unfortunately the original system owner/main software developer is no longer with the team at GridGear, and that he is actively seeking a replacement software engineer to undertake many of the security recommendations given to him as a result of this analysis project.

As part of the debriefing, we notified Ilya of the redacted portions of the report and ensured that the entire contents of the report are available should he wish to examine it. The rest of the debriefing was purely technical and consisted of all details covered in the methodology portion of our report, with emphasis on how to reproduce our adversary models in demonstrating the flaws that we found in the system.

C. Timeline of disclosure

We met with Ilya for a debriefing before November 21st, which was 2 weeks before details of the analysis were published during the mini-conference (December 5). During this meeting, we discussed a plan of action to develop countermeasures to the discovered vulnerabilities, and provided contributions from the team to patching the system.

The outcomes of the debriefing were outlined in an email sent to Ilya, with the forthcoming exchange to be CC'd to Dr. Beznosov.