2015

Security Frameworks for Machine-to-Machine Devices and Networks

Michael Demblewski
Nova Southeastern University, demblew@nova.edu

NSUWorks Citation

Michael Demblewski. 2015. *Security Frameworks for Machine-to-Machine Devices and Networks.* Doctoral dissertation. Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing. (68) http://nsuworks.nova.edu/gscis_etd/68.

Security Frameworks for Machine-to-Machine Devices and Networks

by

Michael Demblewski

A Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Information Systems

College of Engineering and Computing
Nova Southeastern University

2015

We hereby certify that this dissertation, submitted by Michael Demblewski, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Glen A. Stout, Ph.D.
Chairperson of Dissertation Committee

Date

James D, Cannady, Ph.D.
Dissertation Committee Member

Date

Steven R. Terrell, Ph.D.
Dissertation Committee Member

Date

Approved:

Amon B. Seagull, Ph.D.
Interim Dean, College of Engineering and Computing

Date

College of Engineering and Computing
Nova Southeastern University

2015

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Security Frameworks for Machine-to-Machine Devices and Networks

by
Michael Demblewski
August 2015

Attacks against mobile systems have escalated over the past decade. There have been increases of fraud, platform attacks, and malware. The Internet of Things (IoT) offers a new attack vector for Cybercriminals. M2M contributes to the growing number of devices that use wireless systems for Internet connection. As new applications and platforms are created, old vulnerabilities are transferred to next-generation systems. There is a research gap that exists between the current approaches for security framework development and the understanding of how these new technologies are different and how they are similar. This gap exists because system designers, security architects, and users are not fully aware of security risks and how next-generation devices can jeopardize safety and personal privacy. Current techniques, for developing security requirements, do not adequately consider the use of new technologies, and this weakens countermeasure implementations. These techniques rely on security frameworks for requirements development. These frameworks lack a method for identifying next generation security concerns and processes for comparing, contrasting and evaluating non-human device security protections. This research presents a solution for this problem by offering a novel security framework that is focused on the study of the "functions and capabilities" of M2M devices and improves the systems development life cycle for the overall IoT ecosystem.

Acknowledgements

It is true that the doctoral journey is an adventure of a lifetime and the ultimate in personal academic accomplishment. However, in the words of President Woodrow Wilson, I too "not only use all the brains that I have, but all that I can borrow" so, it is a pleasure to thank those who made this accomplishment possible.

I would like to thank my advisor, Glenn Stout for his inspiration, supervision and support and my committee members, James Cannady and Steven Terrell for all their feedback and guidance.

I am truly indebted and thankful for my management and all my peers at AT&T for always boosting my morale and providing me great information resources, challenges and insight. I am especially appreciative to Ed, Sanjay, Gus, Marc, Nick, Dexter, Scott, Randy and Mike for always listening and providing valuable feedback and wisdom.

It is a great pleasure to thank my classmates Ronda, Russell and Ted for always being there with answers, help and understanding.

I sincerely thank all my friends, whom over the years have never failed to ask about my progress and encouraged me to continue, a very special thanks to Rosa, Dr. John, Judy, Alyce, Sandy, Karen, Alan, Alex and Christopher.

A deeply heartfelt thank you to Jacqui, who has changed my direction and has anchored me many times and for that there are no words that can completely express my love, gratitude and admiration.

This journey could not have been completed without the support and understanding of my family. With love and gratefulness, I thank my Mother and Father for giving me my drive and beliefs and my daughters Lili and Sofie, for their patience, understanding and assistance.

Finally, I thank my daughter, Mickie, my angel, my heart and soul, my constant and reason for existence. Thank you for teaching me how to communicate, for showing me the true path and enlightening me about what is truly important in life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Background

Ecosystems take various forms, including social ecosystems, software ecosystems, and technological ecosystems. According to Jiang and ShiWei (2010), these systems are made up of units consisting of various parts and factors that work together to contribute knowledge, make connections, and follow behaviors to form a community. In recent years, the research community focused on emerging technologies that make up the Internet of Things (IoT) ecosystem.

The Internet of Things (IoT) ecosystem is often labeled by one of the following industry buzzwords, the Machine-to-Machine (M2M) ecosystem, the Industrial Internet, or the Cyber-Physical Systems (CPS) ecosystem. However, industry buzzwords lead to confusion (Casson & Della Giusta, 2014). In the literature, these terms are sometimes used interchangeably, while at other times they are used to describe distinct concepts (Conti, 2006; Atzori, Iera, & Morabito, 2010; Severi et al., 2014). The Internet of Things is the more predominant, broader concept describing where the physical world merges with the digital world.

IoT evolved from M2M and other technologies including the Cellular Networks, Location Based Services and the supervisory control and data acquisition system (SACDA). The IoT is an idea, an architectural framework for where all things are connected over a network. The M2M ecosystem is where the machines communicate with other physical machines as connected and networked devices within the IoT. These machines as devices, are a collection of sensors, smart switches, meters, network gateways, and controllers that are connected together to collect information and transfer that information to a processing point, with little to no human user interaction (Chui, Loffler, & Roberts, 2010). The focus of this research is on the devices that make up the M2M ecosystem.

As explained by Wu et al., (2011), M2M ecosystems can be found in many domains, such as consumer services (smart home technologies, consumer electronics, connected car, etc.), manufacturing, healthcare, energy (smart metering and grids), and transportation (asset tracking and vehicle to vehicle communications). In all, there are 16 critical national infrastructure domains as defined by the U.S. Department of Homeland Security (Vugrin, Warren, Ehlen, & Camphouse, 2010).

The recent growth of M2M attributed to several compounding factors, including global Internet reach, growth and expansion of mobile networks, the growing maturity of ipv6 and the capillary architecture of meshed networks (Marcovici, 2014). Other key factors have been the rapid evolution of smartphones, tablets, and supporting technologies, along with the dropping prices of sensors, actuators, and processors. Technology leaders and research firms such as Cisco, Machina Research, ABI Research, and IBM estimate that the future M2M/IoT connected embedded node-base will number upwards of 50 billion devices within the next 5 to 7 years (Murar & Brad, 2014). Some forecasts estimate more than one trillion devices will be connected by 2025.

It is believed that the M2M ecosystem will evolve through three distinct stages. Stage one involves getting more devices onto the network. Stage one is where we are today. Stage two is true automation among devices without human interventions. Stage three is “building applications on top of connected devices,” so that things interact and are tied together seamlessly (Chang, 2014; Latvakoski et al., 2014). Researchers expect that M2M ecosystems will track and interact with all aspects of life in the future. As a result of the evolution of M2M, its pervasive nature, the convergence of physical and cyber worlds, and the implications of devices taking action without human intervention, M2M security is critical to safeguard our national security as much as our personal security (Rubin, Lynch, Escaravage, & Lerner, 2014).

Understanding Machine to Machine

Machine to Machine or (M2M) is a term that refers to computing devices (sensors, actuators, and gateways) that are interconnected, communicate with each other, supply information to upstream systems, and have the capability to collaborate and act in the physical world, primarily without human intervention (Boswarthick, Elloumi, & Hersent, 2012). As M2M ecosystems are deployed within various domains, there will be a natural evolution for these domains to interact with each other. For example, a connected car system may interact with a smart home system to trigger events within the home as the car pulls into the driveway. The growth of wireless networks, as well as the rise of smartphones, tablets, and supporting technology, will continue to fuel the growth of devices in the wireless domain. However, most M2M communications ecosystems will include a mix of both wired and wireless communication mechanisms and a cluster of Meshed capillary local networks and worldwide reaching long-range networks.

M2M Service Categories

M2M is divided into two service categories: Finished services and Ad-hoc services. A service provider manages “Finished” services from end-to-end over secured transport mechanisms that may or may not be dedicated to the service (Kim, Wei, & Lee, n.d.).

Ad-hoc services, on the other hand, are not formally supported by the provider and are supported by an individual entity, such as a consumer, small business, or open community, and are run over the open Internet (Niyato, Xiao, & Wang, 2011). Due to lack of centralized management, monitoring, and processes in-place, ad-hoc services provide greater security risks.

M2M Ecosystem Samples

There are many types of M2M ecosystems that span a multitude of industry verticals, including energy, manufacturing, consumers, and transportation. Some examples of prevalent M2M ecosystems that illustrate the pervasive nature of its application include the following.

- Connected car: GM OnStar was one of the first applications to use basic voice support services and vehicular troubleshooting. It evolved with new services, such as anti-theft remote engine kill and more sophisticated diagnostics. It is now possible to use smart phones to access the car to check the battery life and locate or unlock the car; entertainment packages are being added as well. In the future, cars will communicate with other cars and traffic control centers for collision avoidance and congestion control. The concept of driverless cars is also fast becoming a reality (Chan, 2011).
- Healthcare: Kim et al., (n.d.) explains that M2M technology has a major impact on the healthcare sector; M2M applications are already in use in hospitals and medical facilities and by individual healthcare professionals. Additional verticals include remote patient monitoring, functionality, maintenance system controls, and patient records storage. Connected devices monitor a patient's condition

worldwide, so that a radiologist in New York can view and diagnose a patient's broken leg in the French Alps and interact remotely with other doctors. Medical equipment with sensors for Smart IVs, monitor and change medicine dosage for a patient, which automatically helps nurses and doctors to provide better patient care. Electronic records are updated in real-time, providing caregivers with up-to-date treatment information.

- Wearable devices: Wearable devices for real-time tracking of the vital signs of self-trackers and medical providers are already available in the market. The next invention is likely to be subcutaneous sensors; any of these devices could easily network with medical devices such as insulin pumps and pacemakers (Seong, Lee, & Kang, 2014).
- Smart Grid: The smart grid technology streamlines energy production and distribution. For example, connected devices in a house are networked with neighborhood hubs that perform centralized processing to manage the local energy use, to reduce usage, to power storage requirements, and to provide real-time availability updates (Sharma & Akhouri, 2014). Intelligence is added to devices that plug into the grid. This allows the electrical grid to anticipate potential supply and distribution issues. Working in concert with smart devices and smart meters, the smart grid can automatically coordinate electrical usage and notify devices to reduce consumption or perform tasks after peak loads subside (López, Moura, Moreno, & Camacho, 2014).
- Smart infrastructure and cities: Sensors added to roadways, bridges, buildings, and transmission towers provide regular updates concerning the state of use and

wear and tear on these structures, thus allowing for maintenance planning and usage control. While humans using sensor data in a traffic control center today manage traffic flow, the natural evolution will be smart roads and bridges that communicate with smart traffic control devices to control traffic flow automatically (Elmangoush, Steinke, Al-Hezm, & Magedanz, 2014).

- Smart home: Consumers have begun to automate day-to-day experiences within the home. Many electric and electronic appliances can now be wired or connected wirelessly to a central control system and gateway. These control systems provide the homeowner with the ability to manage various devices such as alarm clocks, coffee makers and even the toaster. Automation includes window shades that rise on command and thermostats that set temperature based on the occupant’s movement. Home security already leverages M2M; networks of sensors and devices take action based on owner-defined rules. Homeowners use mobile phones to control electronic home appliances and utilities by applying intelligence techniques. These intelligent things learn to connect and communicate without human interaction (Hosek, et al., 2014). This is rapidly evolving to automated tasks or notifications that occur when the user is outside of the home. For example, if you rush out of your home late for work, you might receive a message on your smartphone that you left the doors unlocked. You could take an action remotely to lock the doors and arm the alarm system.
- Asset and cargo tracking: Knowing the real-time location of goods is critical for many industries. Small M2M devices such as RFID tags and GPS modules can be attached to assets and cargo containers to allow suppliers, shippers, and owners

to track their high-value assets across multiple transportation methods, including the shipping containers and even the cargo bay of an aircraft. M2M devices can be attached to a high value asset that enables the end user to track the location of the asset. According to Jordan (2014), automated shipping ports will use container attached M2M devices to track the location of containers, load containers on the appropriate ships and trucks, and automatically redirect containers based on customer needs. In addition, asset tracking supports inventory controls. For example, field equipment such as tractors and mowers are monitored for location and condition. Two mowers placed along the highway communicate with each other to avoid overlapping coverage and collision hazards. For shipping, M2M devices, combined with GPS location, enables updated condition notices, such as refrigerated van temperature. For inventory and vending machine, product expiration, service and maintenance needs are transferred to the distribution center and orders are automatically filled (Jordan, 2014).

Conclusion

M2M ecosystems will continue to evolve from reporting and telematics to a state where collaboration between devices, as well as between M2M ecosystems, will create actions in the physical world that occurs without human intervention. There will be billions of connected devices into which we may or may not have visibility and these systems and devices will be taking on greater and greater responsibilities which will have a significant human impact based on their proper operation. Because of this, it is critical that a sound security methodology is applied to secure the overall IoT Ecosystem.

Problem Statement

The M2M ecosystem is not secured by present information security policy (Lake, Milito, Morrow, & Vargheese, 2014). A research gap is present where current security frameworks fail to address needed protections for next generation systems like M2M. Present frameworks do not adequately consider next-generation threats from a system or device that is non-human driven, which weakens countermeasure implementations and security guideline development (Wash, 2010). There is a lack of sufficient research on the intersection of human interface, wireless, and M2M device risks and security countermeasures (Riahi, et al., 2014). Presently, security for M2M is based on current mobile systems' security frameworks. It is unknown how these frameworks will measure up against emerging threats, unknown vulnerabilities, and new devices in an automated and self-reacting system like M2M (Ennesser, 2012).

Dissertation Goal

The goal of this research was to develop a framework that evaluates Machine-to-Machine (M2M) device security. The purpose was to seek out and solve the stated problem by developing a novel security framework that is focused on validating the study of the "functions" of M2M devices, based on the tested "capabilities" of such devices and then improve the development life cycle processes in the M2M ecosystem. This effort (a) adequately considered restrictions and constraints; (b) identified significant shortfalls; and (c) led to a more thorough and detailed M2M Security Framework. Other existing methodologies do not provide a representation of what or which M2M system components should be secured based on the function they perform (Godfrey et al., 2015). A framework approach that identifies devices, components, and functions that are harmed when left vulnerable helps to direct ecosystem-design principles and generates better security recommendations (Molotsi & Tait, 2013). Weaknesses in any security framework, or the lack of such a framework, threatens the overall protection of the system, because these jeopardize the confidentiality, integrity, and availability of the endpoint device; where the system is most exposed.

This research identified and defined the framework's security controls, which then offers possible improvements to authentication, authorization, data confidentiality, integrity, accountability, session management, and transport security in order to mitigate risk. Without sufficient knowledge of these controls, the functional requirements that are created out of the framework will not satisfy architectural principles and good security practice standards (Godfrey et al., 2015).

This goal was reached by first building a test platform that includes three M2M devices connected to a device-management system for command and control. System's development, research and testing was completed by modifying, adapting, and applying testing techniques as described by the Open Web Application Security Project, Mobile Security and OWASP Internet of Things Top 10 Projects (OWASP, 2014). Although the OWASP project is dedicated to web application security, all testing methods effectively offer guidelines for evaluating the security of computer systems and networks. Utilizing these methods led to the validation and verification of the effectiveness of a security framework for M2M and the development of a new, comprehensive M2M security framework that addresses the uniqueness of this technology.

Relevance and Significance

The Internet of Things (IoT) running M2M devices is completely changing today’s computing environment (Chen, 2013). IoT is best described as the connection of people to people, people to machines, and machines to machines, or as the idea of all objects connected to the Internet (Tan & Wang, 2010). M2M devices are the hardware device component sub-section of IoT and a collection of sensors, smart devices, network nodes, and controllers that are connected together to collect information and transfer that information to a processing point with little to no human user interaction (Chui, Loffler, & Roberts, 2010). Chen explains that these sensors can listen and talk in ways that humans cannot. It is believed that M2M will improve human life by helping to save money, forecast weather, predict events, and will provide an overall safer environment (Chen, 2012). However, there is a lack of study about the effects on security and privacy of humans not monitoring, controlling, maintaining, or policing the ecosystem (Dahl & Holbo, 2012).

Many mobility trends today lack properly designed security and privacy requirements. For example, Bring Your Own Device (BYOD) and the push to move security responsibility to the end user are exposing new issues for security and privacy (Armando, Costa, & Merlo, 2013). According to Armando et al. (2013), corporations are finding that security policies geared toward traditional desktop protections are antiquated and ineffective. A modernized approach is needed to protect the swiftly moving workforce from the environments offering information anywhere, at any time, and with any device (Lu, et al., 2011). If the exchange of information or content is unprotected, confidentiality and privacy are lost. Next-generation devices, such as those used in M2M, will help control connected environments and will provide a link to mobile systems, smart cities, and various sensor networks; therefore, the people using these platforms must protect all the information thereon (Kriesten, Tünnermann, Mertes, & Hermann, 2010). Enterprise IT policies protect corporate information only if the users abide by the rules set in place (Harris, 2009). With the increase of attacks on personal mobile devices, companies have experienced an increased loss of protected data and compromise of company networks. These same vulnerabilities threaten the M2M ecosystem.

According to Constantinos, Coursaris, and Kim (2011), all mobile environments lack an effective way to understand how the user will implement various applications. The researchers state that present mobility requirement frameworks lack functional tasks understandings and how capabilities drive the ability to protect. This lack allows malicious users to subvert existing requirements and presents designs that cannot change when the user defines a new way to apply the technology. Conventional requirements inadequately design mobile interfaces and applications to fit particular contextual settings, and are not flexible (Constantinos, et al., 2011). Requirements of the engineering process in M2M fail because there is no consideration of activities taking place for the development of threatening scenarios. Therefore, risk management and applied testing processes should be incorporated into security frameworks for devices in ecosystems such as M2M (Caruso & Masters, 2014). This study exposes the failing points of present techniques by identifying specific task failures and proves that a new approach is more effective. It explores the device from the view of dynamic factors to reveal how this impact usability may contribute to effective security design, as recommended by Constantinos, et al., (2011).

Wireless carriers force robust security wrappers around devices in order to address the lack of protected access, at the risk of violating the privacy of the owner of the device (Enck et al., 2014). These solutions put the network operator in control of the user owned devices; the operator has access to the device, to personal user data, to user location, and to specific networks and applications. Once on the network, the device's policy and provisioning capabilities are subject to security automation (Enck et al., 2014).

Wireless system security is complex. Identifying behavioral patterns reduces the complexity and exposes common functions, which leads to more effective requirements and reusable design practices. However, in M2M, the access to the device, user personal data, user location, and even specific networks and applications is automated and hidden. User side protections are limited, and thus security frameworks are limited (Saedy & Mojtahed, 2011). IoT and M2M communications make information available through crowd and information base; this calls for a shift toward information controls beyond network policy in order to protect the unaware benefactor (TalebiFard, Nicanfar, Hu, & Leung, 2013).

Barriers and Issues

In order to successfully complete this research, the study had to ensure that the new framework proved to be a value-added process for users and ecosystem owners. The Internet of Things (IoT) is a concept where physical objects all connect to the Internet. Because M2M devices are the physical part of the IoT, one framework does not guarantee that all objects are secure. This barrier required that we narrowed the quickly changing environment down to one specific “thing”, such as a set of devices performing a specific “task”. To mitigate this challenge, the study focused on three prototyped M2M devices that fulfilled the specific tasks and function, such as location and device control required by the driving use cases. For each task, security controls were tested, changed and analyzed, depending on the failure or success of the increased control. These efforts led to the final framework development and validation.

Assumptions, Limitations, and Delimitations

A limitation of this study was that the M2M ecosystem use cases did not clearly defined the required security requirements required by the business logic. Also, the insufficient research did not address how tiny machines will function in various categories and environments. Several delimitations exist regarding the scope of the review and test platform. M2M devices change quickly and commercial devices are limited and untested, the prototyped devices required alteration to comply with the Command and Control Center and testing tools are limited and mostly available for only smartphone evaluation. In addition, all testing was “feasibility” in nature and sought out possible attack points as defined by past research and executed attack scenarios.

Definition of Terms

Autonomic computing. Is where devices and systems are operating in a self-managing computing model.

Actuator. Is a device that performs action and output.

Application. Is typically software that is designed to perform specific task for a desired outcome.

Capability. Is where the ability to do something is recognized and determined.

Controller. Is an object that controls actuators.

Device. Is any embedded electronic computing equipment that collects data for actuators and sensors for communications.

Function. Is where the purpose or activity for which a thing exists and is used is recorded.

Gateway. Is equipment with electronic computing and communication capability is used for transferring information within networks.

Internet of Things (IoT). An idea where objects, animals or people are seen as unique identifiers connected together over a network.

Machine to Machine (M2M). Is the concept and systems design where devices communicate to each other without requiring human-to-human or human-to-computer interaction.

M2M sensor. Is the device that detects and responds to specific inputs such as motion, moisture and pressure.

M2M Service Provider. Is an entity, such as a company, that provides network connectivity.

M2M Ecosystem. Is an area network that provides connectivity between M2M devices.

M2M Communications Network. Is the physical telecommunications used to exchange data between entities, such as devices, gateways and network infrastructures.

RFID (radio frequency identification). Is a technology and devices that uses electromagnetic radio frequency (RF) to output the identity of an object, animal, or person.

Secured Environment. Is where enabling secure execution of functions is place in an ecosystem

S*mart Grid.* Is a generic label for the electricity distribution system.

Thing. Is simply an identifiable element located in an environment that is connected to other things for the purpose of intelligence transfer.

Use Case. Is a model that describes a system function from the point of view an actor.

List of Acronyms

Bring Your Own Device (BYOD)

Global Positioning System (GPS)

Internet of Things (IoT)

Intrusion detection systems (IDS).

Machine-to-Machine (M2M)

Mobile ad hoc networks (MANET)

Radio Frequency Identification (RFID)

Session Initiation Protocol (SIP)

Supervisory Control and Data Acquisition (SCADA)

Wireless sensor networks (WSN)

Summary

This chapter provides an introduction and overview of the current operational M2M ecosystems. It presents the technical definition of the IoT, and then introduces the M2M security problem that needed to be addressed at the time of this research. Also, described are the innovation and novelty of M2M devices; these platforms and applications call for unique security requirements and design. The chapter summarizes some of the challenges of M2M security and gaps in the literature. At its end, the chapter introduces the goal of the research preformed to develop a framework that evaluates Machine-to-Machine device security.

Chapter 2

Review of the Literature

Introduction

Machine-to-Machine (M2M) devices are comprised of inexpensive sensors that are deployed across many different domains. These include smart power grids, vehicular telematics, information management, medical and health services, and smart home networks (Poncela, Moreno, & Aamir, 2014). Because M2M devices will eventually be included in all objects, including books, televisions, bikes, cars, and homes, the large amount of data is a security risk in itself (Shafiq, et al., 2013). The data collected is stored in unknown locations within the cloud and can disclose information about individuals, such as buying pattern, locations, communication activities, and even health data. M2M device costs fall in the region of $1 to $200. They have a wide range of applications for different industry sectors, which makes them an inexpensive option for business technologies. M2M devices are unsupervised and placed in a variety of locations, which provides hackers access to individual devices and exposes them to theft, reuse, and fraud (Chen & Ma, 2014). As the M2M market grows, researchers expect that the number of fraudulent uses of these devices will grow.

Mobile devices such as smartphones, tablets, sensors, and laptop computers have become tools for everyday life. However, because users of these devices are not fully aware of security risks, the devices are used in ways that may jeopardize the user’s safety and personal privacy (Kanuparthi, Karri, & Addepalli, 2013). Attacks against mobile devices have escalated due to an increase of fraud, development of malware specific to mobile devices, and the heightened interest of cyber-crooks (Murynets & Piqueras Jover, 2012). Mobile devices are full of sensitive information about users and the companies that employs them; this vital intelligence can be used to gain access to internal business and personal networks and systems (McAfee, 2011). Mobile devices are no longer just targets for low-level hackers, but are also the targets of criminals seeking to steal personal and business communications and data (Murphy & Murphy, 2013). M2M networks and devices have little or no user interaction, but the same vulnerabilities will threaten these systems (Kim, He, Thottan, & Deshpande, 2014).

The fundamental premise of this shift in technology is not only to integrate data for greater efficiency, but also to develop the means by which to link goods and services to consumers and users in strategic marketing and engagement (Atzori, Iera, & Morabito, 2010). Despite this fundamental premise, there are many additional facets to the purpose of M2M. In essence, the process breaks down the barriers between digital and physical objects (Kortuem et al., 2010). This can allow for greater connections and efficiencies in the sale of such objects, in the use of educational resources, in the development of health care objectives, and so on (Welbourne, et al., 2009).

At the same time, however, M2M introduces the means by which additional challenges may occur. Issues of security, which is the ability of the system itself to remain safe for users and for organizations, and of user control, which is the ability of a user to have power over the user’s own identity and experience, are yet to be solved (Sarma & Girão, 2009). It is evident that not all of the infrastructures for M2M systems have been perfected to the point at which people know, beyond a shadow of a doubt, that their private information is being protected. It is clear that IoT ecosystem and all of the new technological tools that they present create issues for both business and personal consumers when it comes to privacy. This literature review outlines the present challenges for M2M systems within the context of smartphone technology. Because smartphones share security protocols and networks with the M2M ecosystem, it is important to draw on the smartphone literature and its application to newer M2M technologies that have not been investigated as deeply in the existing research. The literature review thus provides an assessment of current challenges within the M2M security framework and examines the solutions proposed by the literature and concludes with an assessment and summary of the literature.

A Definition of Security Frameworks

A security framework is a more comprehensive form of an information system's framework model. According to Alqassem (2014), these models ensure security by examining vulnerabilities and eliminating risk. A complete security framework includes several essential elements in terms of technological applications, people, usages, processes, policies, guidelines, business logic and strategies (Ohki, et al., 2009). To be of value, a comprehensive security framework must include:

- Proper practices and execution of policies,
- Sound controls of people, processes, and technologies,
- Analysis of risk,
- Acceptable options or alternatives,
- Have an implementation guide,
- Provide a method to test compliance against the framework. (Alqassem, 2014)

According to Laya, Alonso, and Alonso-Zarate (2014), the overall body of research on M2M security is not comprehensive; in fact, it is fragmented and largely separated into the different critical infrastructure sectors, as defined by the U.S. Department of Homeland Security (Evans, Hammond, & Shamsuddin, 2014). Work has been accomplished within these sectors, for example, on the communications and security protocols for wireless networks, smart grids, smart home monitoring systems, and health care systems (Laya, Alonso, & Alonso-Zarate, 2014; Joshi et al., 2014; Park, et al., 2014).

M2M ecosystems are pervasive in nature and found across many domains (energy sectors, manufacturing, agriculture, vehicular telematics, information management, medical and health services, smart homes, etc.). They allow for inter-domain communication and capillary networking for federation. Federation takes place when M2M engages in system-to-system collaboration and interfaces together (Lee, Lee, & Rhee, 2014). These systems and their various components interact with each other and perform functions on behalf of humans in the virtual and physical world. As explained by Godfrey et al. (2015), these automated functions and the potential impact of their manipulation or disruption drive the need for new processes and the evolution of security technology in M2M, in terms of device identity management, security capabilities for low powered devices and security visibility tools. To this end, the literature review places a significant focus on understanding the gaps in the literature and drawing connections between current M2M protocols and best practices in related technologies such as Cellular Network Devices.

M2M Devices and Smartphone Technology

It is important to understand the differences between machine-to-human and machine-to-machine devices because the devices have parallel security concerns. A greater depth of research has taken place in machine-to-human technology than in machine-to-machine technology, so it is to our benefit to understand the extent to which machine-to-human research can be applied to security issues of M2M devices.

M2M devices use the same wireless communications networks as smartphones, but also make use of short-range networks and gateways for peer-to-peer messaging. To this end, the development and end-user issues that impact smartphone technology and security also impact M2M device protocols (Gyrard, Bonnet, & Boudaoud, 2014; Poncela, Moreno, & Aamir, 2014).

According to Zhang et al. (2011), smartphones are machine-to-human interface devices that provide input for communication and applications function. Machine-to-machine devices, on the other hand, are small and inexpensive, and they are designed for automated (rather than human-centered) wired or wireless communications. Both devices rely on the same networks for communications, but M2M devices may offer greater security concerns with respect to confidentiality, integrity, and availability due to bandwidth restraint, authentication, access control limits, and the need for secure identification certificates (Vandikas, et al., 2011; Kim & Hong, 2014). Nonetheless, M2M deployed devices will outnumber smartphones within the next decade (Cruz, Duarte, & Ferreira, 2014). In 2012, there were one billion smartphone users worldwide; by the end of 2015, this number will reach 1.75 billion and, by the end of 2020, the number of connected M2M devices will potentially reach 50 billion (Cruz, Duarte, & Ferreira, 2014).

While smartphones have specific usage and characteristics from a user behavioral perspective (Welsh, Baird, Zhao, & Block-Schachter, 2014), M2M devices are designed for specific tasks and industry functions (Poncela, Moreno, & Aamir, 2014). However, according to Gyrard, Bonnet, and Boudaoud (2014), many M2M devices currently use 2G and 3G embedded modules, which lead to old and new vulnerabilities and security challenges that require new security mechanisms for countermeasures. Given these factors, researchers have proposed various changes to authentication in order to offer built-in authentication and security for easier deployment and network optimization (Hersent, Boswarthick, & Elloumi, 2012; Xu, Liu, Huang, & Zhang, 2014). However, these changes are untested and not standardized.

Although smartphones are capable of more complex tasks than most M2M devices, the functionality of the two types of devices is basically the same. Both are attractive targets for attackers (Aucinas, Crowcroft, & Hui, 2012). Both types of devices can be affected by data integrity issues and, therefore, require data protection assurance (McGrath & Scanaill, 2013). Smartphones hold user privacy-data and M2M devices transport this same data.

For example, smartphone global positioning system (GPS) networks are not only tied to standalone applications, but are also increasingly tied to social media services (Scipioni & Langheinrich, 2010). The location-based software on a consumer’s phone may be engaged in collecting and publishing location information whether or not the consumer is actually aware of these processes, and this data can be linked to smart city

M2M devices as well (Scipioni & Langheinrich, 2010). GPS data can then be used to stamp location information onto digital photographs in order to profile tourists as they travel, which leads to a privacy violation in the examining of the user's movements throughout a city (Gasson, et al., 2011).

Information security best practices provide classification to the data for applying security controls to smartphones and the wireless network elements over which the devices function (Kazmi, Felguera, Vila, & Marcos, 2012). In the M2M ecosystem, on the other hand, automated decisions and business logic create the rules on data transport, which requires the implementation of higher levels of security (Pang, et al., 2013).

Specific Threats to M2M Devices Predicated by Smartphone Challenges

There is evidence that M2M devices have similar areas of vulnerability to smartphones devices in terms of data security. As Christiansen (2011) explains, the different forms of mobile data are collected and used by corporations in the following three ways.

> 1. Collect personal data and aggregate it to sell to third parties, use it internally, or both.
> 2. Collect personal data, keeping personal data within the company but providing the opportunity for advertisers to specify a certain range of traits for target marketing.
> 3. Collect personal data with the intention of selling the information, sometimes including specific profiles or names, to third parties. (p. 509)

Consumers may not be aware of when and the means by which this data is collected through smartphones and M2M devices, nor of when it might be passed on to third parties (Christiansen, 2011; King & Jessen, 2010; Leontiadis et al., 2012; Scipioni & Langheinrich, 2010; Xu et al., 2011). This is because the majority of users do not read the privacy and user agreements they sign when they purchase or download software (Christiansen, 2011). But, as Kiukkonen et al. (2010) explains, consumers do not perceive the connection between these types of information and their usage, especially in an M2M ecosystem where there might not be a user agreement. M2M network owner's need to become more aware of how and why consumers choose to give them access to their own data, and whether or not there is an actual choice taking place, especially when many of the data mining techniques are hidden from the consumer's view on a daily basis. All of this data is thus likely to be open to external scrutiny if the M2M or smartphone system is compromised.

Security researchers have identified ways to bypass device restrictions and install rewritten firmware that creates malicious vulnerabilities within smartphones (Aviv, Gibson, Mossop, Blaze, & Smith, 2010; Park, Choi, Eom, & Chung, 2013; Karim, Shah, & Salleh, 2014). If these same types of attacks were to occur within the context of a medical M2M device instead of within a smartphone environment, this could lead to the harm or even death of a hospital patient (Pérez-Cebollada, Martínez-Ruiz, & Bernal-Agustín, 2014). According to Pérez-Cebollada, Martínez-Ruiz, and Bernal-Agustín (2014), M2M devices that transport medical data are not secure because the memory of the devices is limited. New authentication and secure communications are required to protect the data and the device. Given these factors, both the potentially critical nature of the industries involved as well as the potential threats to property and human safety make M2M security a high priority.

The smartphone air interface threat is also significant to M2M devices because evidence has shown that smartphones and wireless platforms have been attacked by a number of vectors that will compromise M2M devices. Examples include: the “man-in-the-middle” scenario, where an attacker can place a device between the target user and the network (Ukil, Bandyopadhyay, Bhattacharyya, & Pal, 2013); the compromising of authentication where an attack on the “challenge and response pairs”, the cipher keys, overrides the integrity keys of the authentication vector (Cheng, 2011); and eavesdropping, wherein the intruder listens to signaling and data connections associated with users and network elements without the knowledge of users (Arapinis et al., 2012). Also, an attacker can impersonate a user or an entire network by using false signals, user data, or both through the network in an attempt to make the network believe they originate from a “good” user. Furthermore, signals, user data, or both can be sent to a target user to make that user believe they originate from a genuine network (Arapinis et al., 2012).

Nonetheless, the greatest threat to mobile devices comes from malware embedded in applications and the fact that these devices are always connected to a network (Distefano et al., 2010). Mobile application development has exploded, but the capability of sending mobile malware to devices has exploded at the same time. Malware has infected wireless-enabled devices and is capable of propagating itself to other devices, including M2M devices (Liu, Zhang, Yan, & Chen, 2009; Landman, 2010). The wireless industry as a whole is ill-prepared to combat the problem. According to Felt et al.

(2011), malware grew to epidemic proportions from 2009 to 2011. Malware programs exploited all mobile operating systems, and their designs have reached a paramount level of sophistication. It is unknown how quickly Malware will move in an M2M ecosystem.

Data attacks have also been on the increase. Over the past ten years, wireless devices have become more and more data-enabled. M2M devices receive and send data, and access applications and information quickly (Miluzzo et al., 2010). Xie et al. (2010) examined the security services that are provided at the device level and discovered that vendors and developers change security profiles for faster data transfer over open networks. M2M devices lack security tokens but transfer valuable information and do not offer proper security guidelines for open wireless networks, the lack of these guidelines increases security and privacy concerns (Ashley, Hinton, & Vandenwauver, 2001). If infected with a virus, the stored information will be lost or even transferred to an unauthorized user. To promote better security, a new framework approach is needed that supports security design at the component level and that defines best practices for next generation networks (Guo et al., 2013).

Device-to-device attacks have also become more common. Voris, Saxena, and Halevi (2011) explained that devices are capable of functioning as both attackers and victims of an attack. Motivations for such attacks range from simple vandalism to information theft, mobile phone spam, and denial-of-service attacks (Goel, 2011). In this form of malware, mobile bots function as propagation applications that cause excessive charges to customers, deteriorate services, and even cause public relations disasters (Nadji et al., 2011). Chan, Venkataraman, Chaugule, and Campbell (2010) proved that attacks need not be complex and can serve to launch an authentication attack on the operating systems that bypasses access control mechanisms and forces a restart of the systems. Chan et al. (2010) described an attack that used the mobile phone as the vector and allowed for access to files and network connections. Using the M2M device as a vector will compromise relevant public information on various platforms (Owusu, et al., 2012). The sheer increase of the numbers of M2M devices and their wide disbursement will present more opportunities for devices to be used as vectors. The lack of security controls within M2M makes the ecosystem more susceptible to compromise, which could lead to collateral damage to the M2M ecosystem. Federation among M2M systems, including capillary networks using multiple communication protocols (e.g., ZigBee, Bluetooth, Z-Wave and others), offers even more opportunity for compromise of authentication, authorization, and verification.

According to Bahga and Madisetti (2014), the monitoring capabilities and processes that providers use to detect and respond to security incidents within the M2M ecosystem must evolve. Today’s processes and tools for detection of M2M security events including present communication protocols, traffic patterns, and even the potential massive distributed scale of the IoT are immature. According to El-Mahdy (2014), M2M devices are vulnerable to the same attacks as smartphones, as well as new forms of security threats. M2M platforms such as connected cars, sensors, and smart homes are especially susceptible to these same attacks, in addition to denial-of-services attacks and intrusion. For example, the flooding attack is of great concern, according to Liu, Yang, and Liu (2014). In a flooding attack, a network interface is compromised by misconfigured end-devices, which causes an authentication signaling failure on the signaling and user plane of the M2M device. Another worrisome possibility is the attack of a botnet of malicious devices that attempt to flood the network (Jermyn, Salles-Loustau, & Zonouz, 2014). Jung, Kim, and Kim, (2014) warn that the ability to generate and disseminate detailed information on M2M networks facilitates the spread of malware. Specifically designed malware that may infect M2M devices within a group could cause a lack of authentication mechanisms on M2M platforms adjacent to the main device. These platforms include connected car or smart grid gateway sensors (Li et al., 2014). M2M devices are also subject to data modification and manipulation attacks against data (Ren, Yu, Ma, & Ren, 2013). Modification attacks target against the integrity of routing messages to the prioritized devices, which causes a failure in service and can lead to harm or even death in the case of medical systems or connected cars (Jeon, Lee, Park, & Jeong, 2013). Detecting these attacks is a primary concern, especially in smart metering networks, where source authenticity and data integrity changes can harm the power grid (Abdullah, Welch, & Seah, 2013).

In addition, communications with and authentication of a device in an M2M ecosystem is of utmost importance for these devices to successfully function together to protect against legacy attacks (Ren, Yu, Ma, & Ren, 2013). To explain further, consider an M2M device endpoint that sends messages using the Session Initiation Protocol (SIP), which falls victim to the SIP messaging attack. These attacks take advantage of known SIP vulnerabilities and can cause channel eavesdropping attacks, credential compromise attacks, function compromise attacks, and ghost compromise attacks (Ren et al., 2013). SIP messaging is also vulnerable to impersonation compromise attacks (Koh & Kwon, 2014). It is possible that if a spoofed SIP message is sent to a specific M2M device, the collected sensitive subscriber information will be compromised. Also at risk in M2M is theft of service itself. This occurs in a repurposing attack, where sensors in power meters and devices within connected car fuel systems and compromise or services are used without getting charged for them (Obikoya, 2014).

For this research, we broke down the M2M security challenges into four layers: data collection, communications, computing, and action (Gyrard, 2013). Challenges within each of these layers include deployment, maintenance, and measurement, as well as the risk of failure; the complexity and number of hard-to-manage devices on M2M projects make the latter a particular concern (Gyrard, 2013). For example, when a power company deploys a vast number of power meter reading devices to connect to homes, the company runs the risk of overwhelming the wireless communication networks and thereby threatening the security and privacy of the ecosystem (Brahmi, 2014). Chen and Ma (2014) explain that M2M standards are still under development and that existing solutions are fragmented. Solutions are being designed from scratch, but conventional IT design standards lack policy and trust. Furthermore, they do not address the challenges in the M2M ecosystem (Cohen, Money, & Quick, 2014). Present application development tools lack the proper data analytics, data security, and sensor management needed to navigate the complexity within M2M ecosystems. Different network protocols, data formats, incompatible devices, and multiple applications cause major security and privacy concerns on M2M type systems (Das, Borisov, Mittal, & Caesar, 2014)

In M2M, the engineering of solutions that meet end user needs requires modernization and incorporation of end-to-end threats (Chaugule, Xu, & Zhu, 2011). Increasing the functionality of devices also increases the danger of attack to the operator’s network and the end user’s privacy and freedom of use (Sohr, Mustafa, &

Nowak, 2011). Current threat management programs fail to identify threats or provide adequate safeguards to mobile devices (Neumann, 2009). This practice leads to risky behaviors and weak security postures that do not serve to mitigate mobile security risks and threats.

Security and privacy controls for mobile devices are created by mobile application developers, network systems engineers, and security architects, whom are likely to rely on lessons learned from decades of desktop protection policies (Chin et al., 2012). Unfortunately, this practice does not work efficiently or effectively in a self-directing network like M2M (Oberheide & Jahanian, 2010). Requirements for secure coding and system design are unique in the M2M security (Distefano, Grillo, Lentini, & Italiano, 2010).

Oberheide and Jahanian (2010) explain that security and privacy are not considered at the design stage in mobility platforms. This is due to the fast pace of growth within these platforms (Benjamins, 2014). Mobility service providers are pushing more responsibility for security on to the users. However, if the user is the device itself, as in M2M, this responsibility shift increases the security tensions in an already complex user environment, which can lead to a great misunderstanding in the differences in user behaviors and might possibly promote security risks (*Emerging Cyber Threats Report* 2011; 2012).

Security policy focuses on trust, information protection, and access control rules. These policies define the requirements that enforce security policies developed by IT departments using old rules (Chen, 2013). Chen (2013) explains that these policies do not satisfy the challenges of and do not create suitable security policies for M2M.

Security systems are not effective if users' perceived security and privacy requirements are not included at the design stage (Savola, 2009). As environmental and policy changes take place, it becomes necessary to find a solution for detecting system behaviors and specification changes, especially because M2M encompasses different industries with many different types of devices and access technologies (Chen, 2013).

There is a need to look deeply into the practices that govern both the use of M2M devices and their deployment by businesses. According to Rodríguez, Cuéllar, Lilius, and Calvo-Flores (2014), human activity representation and daily human behaviors studies establish the criteria for the evaluation of missing features for both security and privacy. According to Chen (2013), typical information security practices involve assessing processed data, assigning a data classification, and applying countermeasure security controls to network elements. To this end, it is important to move beyond just the data and network element when determining security controls in M2M development, because it is unclear where these devices will be deployed (Chen, 2013; Rodríguez, Cuéllar, Lilius, & Calvo-Flores, 2014).

Chen (2013) explains that machine-to-human interacting systems analyze incoming attack information in a demand attack-defense fashion. This approach leads to only a partial view of the entire attack and requires great human effort to configure and deploy applications. In M2M, the potentially critical nature of the industries involved and the potential threats to property and human safety make security a higher priority (Chen, 2013). When assessing the design and security controls of an M2M ecosystem, the potential impacts of a compromise must be analyzed and understood before one engineers the network and applications. For example, a compromise at a water treatment facility could result in an unsafe water supply and the loss of human life. A compromise of a fleet management system of heavy equipment, on the other hand, may create problems for the owner of the equipment, but most likely will not endanger human life (Bojic et al., 2012).

Challenges Presented for Adequate Management Solutions

Mobile Device Management platforms (MDM) are the default platforms for protecting sensitive information in the event that a smartphone is lost, stolen, or compromised (Redman, Girard, & Wallin, 2011). These platforms can perform a remote device wipe, enhance behind-the-firewall security, and support access and control for thousands of users and applications (Redman et al., 2011). In addition, the devices using these platforms are loaded with small applications, called clients that create enhanced on-device security and security policy enforcement by over-the-air controls (Khan, Khan, Nauman, Ali, & Alam, 2009). MDMs mitigate the privacy and security risks by controlling smartphone security policy through the adjustment of work and private data spaces and segregating of remote device management.

According to Ebersold (2014), the M2M ecosystem is an immature and the rapidly changing landscape is in need of security controls. The M2M domain remains uncontrolled, with distributed locations of devices and end nodes. Because of this, the ecosystem is considered untrusted and lacking in security requirements; they reflect a limited amount of control. Remote device management for M2M is unlike traditional service models, because there are many devices talking to each other and the backend system. These devices perform real world actions, like locking doors and changing the flight direction of an airplane in flight. However, the mass deployment of M2M services is not supported efficiently by present standards. This hinders the control and management of many device functions (Wu, et al., 2011).

Therefore, M2M devices are designed to function as self-monitoring devices. However, the traditional management techniques do not scale up to the growth of M2M networks and services. According to Song, Kunz, Schmidt, and Szczytowski (2014), new MDM functionalities must be developed to manage and control M2M devices. These functionalities include overload control, conflict management, and semantic interworking controls. The systems themselves must learn personalized service recommendations and policy changes (Kamal et al., 2013). Thus, in the management of traffic, adaptive radio resource management could reduce random access delay experienced by the device (Hsu, Wang, & Tseng, 2013). It is very challenging to manage all different types of devices for effective communication with one another because of scalability and interoperability (that is, what works for one platform might not work for another) (Floeck, Papageorgiou, Schuelke, & Song, 2014). M2M devices will very soon encompass a multitude of communication technologies and will connect to other devices and with many different networks. These devices are unattended and difficult to monitor. Lack of overall visibility capabilities into the M2M domain and relevant device capabilities make it difficult to detect when a compromise has occurred. Thus, M2M requires a paradigm shift in how security is designed (Granjal, Monteiro, and Sliva, 2013).

Chen and Chang (2012) describe the problems with M2M intrusion detection. Few studies have investigated M2M specific intrusion detection systems (IDS). However, related work in wireless sensor networks (WSN) and mobile ad hoc networks (MANET) may provide methods to identify vulnerability characteristics for M2M. Chen and Chang (2012) explain that M2M hardware constraint is a challenge mostly because of the unreliability of wireless links between sensor and actuator nodes within the radio frequency spectrum in the low power radio networks. In addition, Anggorojati, Prasad, and Prasad (2013) explain that it is difficult to identify the rational attacker from the defender in M2M ecosystems because the devices are automated.

These known IDS techniques are not fully efficient because it is difficult to characterize the normal behavior of a sensor and then identify the known behavior patterns of non-authorized devices (Khan & Pathan, 2013). Hammoudeh, Mancilla-David, Selman, and Papantoni-Kazakos (2013) proposed a specification-based IDS to solve this problem. The specification-based IDS is a combination solution that detects malicious message transmissions. A Timing Centric IDS that identifies changes in timing and device response might also pose a solution (Kumar & Chilamkurti, 2014).

In order to achieve the goal of this work, we experimented with specification-based intrusion detection to challenge and prove that the new security framework worked functionally and correctly. As security policies changed, data was collected and analyzed to verify that security had increased. This testing focused on the added value of the security framework, by observing when a set of resources changed and how different security values affected the security policy (Anggorojati, Prasad, & Prasad, 2013).

Assessment of the Gaps in the Literature and Business Applications

The findings from this literature review indicate that there is a lack of and need for a method for assessing the overall ecosystem security and identifying applicable security controls for M2M devices. Typical security practices involve assessing the sensitivity of data that is being processed, assigning a classification to the data, and applying appropriate security controls based on the data sensitivity (the more sensitive the data, the more stringent the controls). The use of inexpensive sensors, mobile and wireless communications, short-range networks, and gateways as enablers to M2M systems present unique security challenges.

It is evident that devices will be pervasive throughout an M2M ecosystem and will be performing automated functions on behalf of humans as the field of M2M technology develops. If the function is manipulated or fails, the potential impacts could be wide ranging, depending upon the criticality of the function. The capability to deploy security control on lower power devices, the lack of visibility, and the processes of device management are some of the evident challenges (Fischer, 2014).

The literature demonstrates that data classification alone is insufficient to drive security controls and a secure design for M2M. This is because the data by itself does not address the overall role of a device in an end-to-end M2M function. For example, if one takes temperature data measurements singly, the information is non-critical in nature. However, the role of the temperature sensor in the context of an overall M2M function (such as temperature regulation in a home or business) determines the importance of its measured data. If the temperature sensor fails to operate as intended (through manipulation or other means) in a home environment, consequences will be much different than if the same temperature sensor fails to operate as intended in a nuclear power plant. Intended functions, device roles, environments, and potential impacts are therefore key aspects of deriving M2M functional classification in a way that drives a secure design (Godfrey et al., 2015).

Past research demonstrated the need to devise a new M2M security framework that is practical for use in each of the 16 national critical infrastructure sectors (Rubin, Lynch, Escaravage, & Lerner, 2014). A framework that could compare, contrast, and make quantifiable statements about security is an extremely valuable security asset for all systems (Chin, Felt, Sekar, & Wagner, 2012). Such a framework allows organizations to determine where resources, policies, and procedures should be placed to best secure present and future systems.

The developed framework was tested against present security frameworks to justify the rationale for how well the new framework reaches security goals for M2M. Building a prototype helped obtain a more complete and thorough understanding of the system and framework. However, it was difficult to compare real world systems with lab-based systems because there is a lack of understanding of what security really means in the M2M ecosystem and when the properties of this system are truly secure. So also offered is a new testing approach for security in M2M that overcomes this problem in order to gain knowledge from the development process as defined by Alqassem, (2014).

Summary

The Machine-to-Machine ecosystem is quickly becoming a commercial offering of devices, networks and platforms. The devices are tiny, unsecure components that are hard to command and control. The ecosystem communicates over wireless, wire line, private and public networks. M2M platforms are presently being developed for all things that connect to the Internet. This systems development effort leads to and allows the objects such as device and gateways to function unnoticed by humans and thereby expand end-user concerns about the balances between security and privacy of services, safety and over systems awareness.

Chapter 3

Methodology

Overview of the Research

This research used the systems development research methodology. This methodology has been a applied to systems research for over the past 100 years and draws from foundation research classification schemes, including engineering, developmental, and formative types of research (Nunamaker & Chen, 1990). The methodology combines processes, methods, and tools to conduct an investigative study. The investigation leads to results that contribute to the overall body of knowledge in applied systems development and new approaches to processes and products in the Information Systems domain. At the research methods foundation, the systems development methodology is ideal for investigating, improving and creating new things (Nunamaker & Chen, 1990). This research follows the proven repeatable method of 1) building a test system; 2) observing the behavior of the system; 3) testing the system with tools; 4) using the results to develop a new and better system (Hubbard, 2014). In the case of this study the outlined method was used to research the present day Security Frameworks for mobile systems security and utilize executed testing results to create a better Security Framework that solved the stated problem for Machine-to-Machine (M2M) devices.

Researchers have called for further study of security and privacy in M2M ecosystems, particularly in terms of how the requirements differ from those in present mobile systems such as Smartphone and messaging platforms (Accorsi, Stocker, & Müller, 2013; Alqassem, 2014; Chen, 2013; Chin, 2013). Typical security practices involve assessing the data that is being processed, assigning a classification to the data, and applying appropriate security controls to network elements that handle the data. Studying frameworks and system architectures in this way fleshes out a better policy and improve effectiveness for protecting M2M devices.

Develop Advanced Theory

Chin et al. (2013) accomplished their smartphone research by conducting interviews of users' willingness to use a device for various tasks in order to test the hypothesis that people fear their privacy and security are at risk when smartphones are used and do not trust smartphone applications. This method will not work in M2M, as the end points are devices and not people.

This research hypothesized that the M2M ecosystem is even less understood and trusted by users because users are not aware of the advanced risks (Chen, 2013). A proper study and requirements analysis method are vital for the development of all complex systems; without them, major problems are introduced in the complete system life cycle (Zafar, Arnautovic, Diabat, & Svetinovic, 2014). A better security framework for M2M must offer and provide a trustworthy environment that builds the confidence of technology recipients and enforces security implementation at the design stage (Saeed, Tahir, Mughal, & Khan, 2014).

Research Methods

For M2M security, architects rely on use case development for framework validation, which has created a present framework that is only theoretical (Katt, Gander, Breu, & Felderer, 2013). This research observed and evaluated device changes, characteristics, and categories such as inputs, efficiency, effectiveness, outputs, quality, impact, and the usefulness of safeguards in order to develop a functional M2M security framework that includes and considers the capability if the devices, as supported by Alqassem (2014). Leading to the understanding of the M2M development challenges and supporting evidence that there is a need for an overall IoT security design and testing method. System prototyping was applied and practical applications used to gain knowledge as to how devices are presently protected and communicate with each other (Parkin, Moorsel, & Coles, 2009; Alqassem, 2014). The table below shows the found differences and similarities between present smartphones and M2M devices. These differences are why M2M devices must be secured using a new security framework. However, the similarities provide a foundation for testing and new methods development.

The OWASP Mobile Security and OWASP Internet of Things Top 10 Project Methods (OWASP, 2014) offer a standardized and disseminated mobile, system applications and hardware risk model based on surveyed results. Fortify (2014) reports that there are many techniques for testing individual platforms; however, these do not address the general M2M threat model. The Fortify (2014) report used mobile device testing tools to examine and compare vulnerabilities found on smartphones and matches them to M2M devices vulnerabilities as depicted in Table 1, the below table outlines the comparison results of present smartphone devices and future M2M devices.

Table 1. Device Comparisons

	Smartphones	M2M Devices
M2M Device Limits:		
Always mobile	Yes	No
Strain on network resources	No, mobile networks are engineered to balance these devices.	Yes, the number of devices and yet unknown uses is a contributor.
Support for legacy networks	Somewhat—carriers are seeking to "sunset" the 2G network.	Yes, these devices are engineered for mostly 2G and 3G networks only.
Easy hardware and software upgrade	Yes, full over-the-air support for device management.	No, unknown ways for firmware and SIM updates to no SIM devices.
Full command, control, and conflict management	Yes	No
M2M Device Risks:		
Direct attacks	Yes	Yes
Indirect attacks	Yes	Yes
Infrastructure and data theft	Yes	Yes
3rd party attacks	Yes	Yes
Protocol attacks	Yes	Yes
M2M Device Attack Vectors:		
Server side attacks	No, not from a device user side.	Yes, these devices talk to each other.
Infrastructure attacks	Not on a grand scale.	Yes, due to the federation of new platforms.
RF side interception and eavesdropping	Yes	Yes

Table 1. Device Comparisons continued

Open gateway attacks	No, mostly closed platforms.	Yes, undeveloped standards and open systems.
Low power device side attacks	No	Yes
M2M Device Vulnerabilities:		
Device control	Yes, hackers have remotely powered on and off devices.	Yes, attackers can take control of home networks, thermostats, or other connected devices via unprotected devices.
Encryption	No, strong encryption and authentication on end-to-end systems.	Yes, weak network protection and unencrypted data storage.
Password	No, most devices use signing key and password for firmware updates.	Yes, devices do not validate Secure Socket Layer (SSL) certificates.
API vulnerabilities	No, standards are defined.	Yes, weak standards that are still in development.
Vulnerable protocol	No, mature and proven protocols.	Yes, protocol flaws and untested vulnerabilities.

Note. Table built from Hersent, O., Boswarthick, D., & Elloumi, O. (2014) *M2M communications: A systems approach*. Hoboken, N.J.: Wiley Publishing.

In this research we developed a new approach that addresses the specific threats to M2M devices by studying how the device might function when tested against proven and successful attack scenarios. The results were then used to build the tested and new functional security design framework for M2M devices that is presented in Chapter 5.

Instrument Development and Validation

The test environment resembled a real world M2M network and included all conditions, circumstances, and influences surrounding and affecting M2M devices. The most important requirement for this test environment was the ability to systematically collect data (Bhunia & Mukherjee, 2014). To ensure efficient data collection, this research use a combination of systems including a commercially available development M2M Control Center to implement visibility into the devices and devices side log files. The control center connects to M2M devices using the wireless network and enables devices management for analysis, diagnostics, connection history, and changes in configuration, and the log files report all actions and anomalies on the devices.

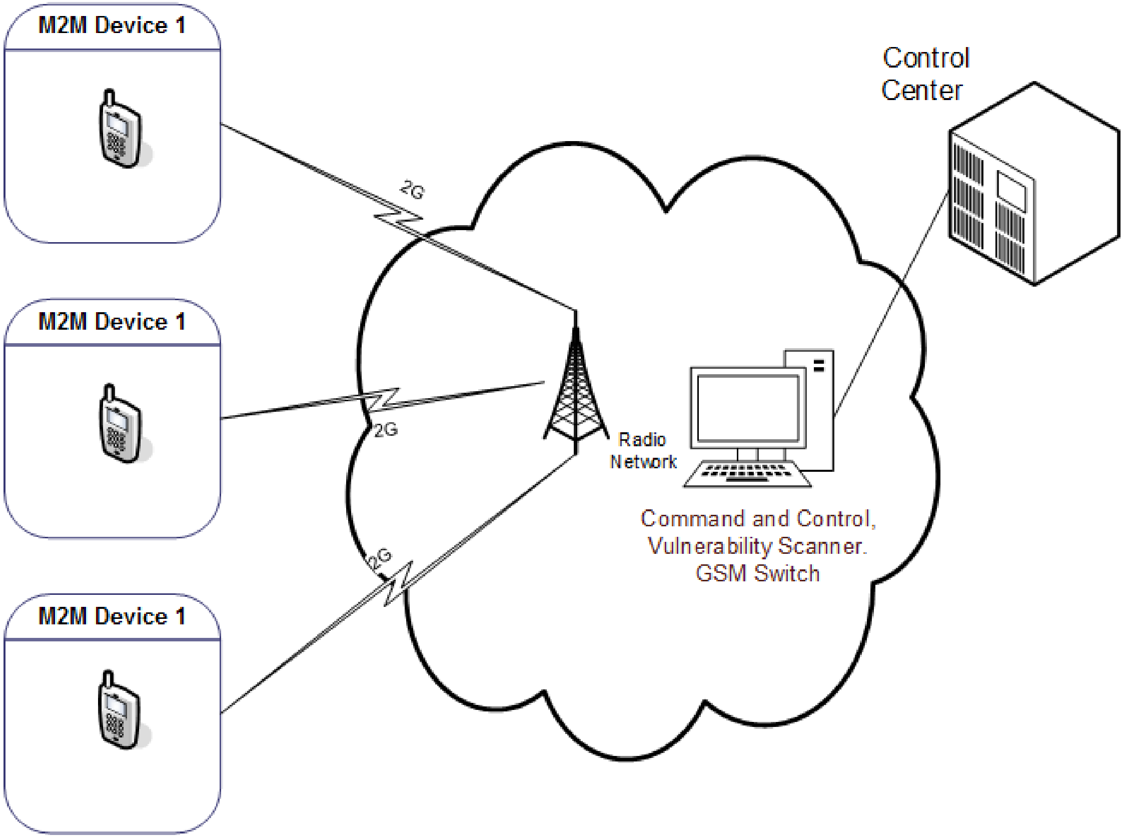


Figure 1. Sample Testbed

The test bed shown in Figure 1, above, is made of GSM-based modules connected over the wireless network to the control center. The Control Center provides device activation tools and allowed provisioning and upgrades to be performed for the device policy for testing and study. This allowed for usage analytics when security policies were changed and product performance, when functions were increased.

Examined Devices

We used three prototype built M2M devices for testing, validation, monitoring, and control provisioning. The devices were built to meet the needs of the Business Logic requirements as defined by the use case examples.

Detailed Procedure

The key to a reliable framework is the understanding and application of the system's rules. According to Yahya, Kamalrudin, Sidek, and Grundy (2014), one must apply detailed use case analysis to reach this understanding, or else the framework fails. For this research, the three below theoretical model example use cases were applied for analysis.

Use Case 1

Request a device that allows a utility company to better manage remote devices without human interaction.

Business Logic. Business logic comprises the rules within the M2M application that define actions. A sensor-controlled meter with real-time feedback and control of grid management devices is in place for Use Case 1. Tasks such as health and status checks can be remotely performed and firmware updating takes place without hands-on access, which results in lower operational costs. In addition, these devices more effectively distribute power by directing the power where it is needed and when, which results in the most efficient use of current assets and lower operational costs. It also decreases outages caused by over-current conditions. Such devices help the utility company to more accurately predict load periods.

Potential Impact. Potential impact observes what occurs if the function is manipulated. In this case, the impact is the loss of operations' controls and updates to the device and severing system.

Device Data. Device Data comprises inputs and outputs to Business Logic. The main data is messaging, in this case. Regulatory takes precedent based on the example Use Case. Data transmission, control features, network wide changes, and updates are vulnerable if attacked.

Supporting Components. Key elements include the sensor-controlled meter, switch, network, and backend.

Use Case 2

Explains a smart device that utility companies will offer to customers so that they can use smartphones, IVR, or web applications to self-service payments for electricity services.

Business Logic. A Smart Grid prepaid electricity device provides location specific data. Prepaid electricity is a fully managed, hosted payment solution that interfaces with smart meters. These devices have a “disconnect service” switch for real-time payment processing.

Potential Impact. Inaccurate or manipulated data might be sent to the backend services. An incorrect disconnect may occur.

Device Data. Data includes meter data, messages from the meter to the backend, backend on-off commands, and commands to meter notifying to send data.

Supporting Components. Components include the meter, capillary-network, GPS, and backend system.

Use Case 3

This device enables the utility to remotely monitor the premise, the work environment, and the health of specific systems.

Business Logic. A Smart Grid Cellular Communication Device with video and location transport strictly designed and developed for Smart Grid backup security. It allows a restart, disconnect functionality, and power shut-off at the premise, which results in a safer operational environment.

Potential Impact. There is the risk of death or harm if the wrong system is shut down or the device fails to report the danger.

Device Data. Data includes automated shut-off messaging, reporting of location data, and picture data.

Supporting Components. Supporting components include the meter, network, GPS, camera, and backend system.

Data Analysis

Most frameworks are built theoretically and conceptually. Theories predict relationships, events, and behaviors. A theoretical framework is an inductive process.

To prove that a framework adds value, theoretical predictions must be observed and evaluated (Liu, et al., 2014). For this research, we observed and evaluated the critical interrelationships among concepts. The following test performances were observed and evaluated.

- The strength of the end-to-end security attributes within the framework.
- The ways the security attributes interact with each basic component.
- The availability to break down each component into functional units and scaled attributes.
- How the defined units affect each security attribute.
- Identification of model components associated with each attribute.
- Comparison and contrast the frameworks for security and the overall value they offer.

According to Alberts, Allen, and Stoddard (2012), some foundational work for security frameworks development has been performed, but has yet to materialize. As a result, decision makers and users lack confidence in the security of emerging systems that have been developed (Buyens, Scandariato, & Joosen, 2009).

According to Flood and Keane (2014), data analysis and systems design encompasses a beginning three-step process: information gathering, static analysis, and dynamic analysis. For this research, the following method was used to extract information out of the use cases, as listed below.

(1) Information Gathering:

- Do the running applications provide security protections? Example: Does the device allow SMS messaging?

- Are the networking interfaces protected? Example: Is there mobile communication only or Wi-Fi wireless, too?
- Are various networks supported? Example: Various networks include 2G, LTE, Wi-Fi, and Bluetooth.
- Do the networking protocols meet industry standards? Example: Are secure protocols used, such as M2M/IoT (XMPP, MQTT)?
- Are transactions performed that require additional security protections? Example: Do transactions include payment information, personal data, or location data?
- Are hardware components exposed? Example: GPS or Camera might be exposed.

(2) Static Analysis:

- Perform a detailed analysis of the device source code based on the Business Logic requirement.
- Review the Operating System security framework.
- Verify that all applications cannot be extracted.
- Outline the permissions for authorized access.
- Analyze configuration files and verify access controls.
- Analyze all points where untrusted data entry may be inputted.
- Outline the user authentication process.
- Identify the functionality of inbound connections from other devices.
- Perform privilege elevation analysis.
- Test and analyze the encryption that is used on the device.

- Seek out device and platform exposed APIs.

(3) Dynamic Analysis:

- Depending on the device vs. Use Case, discover the vulnerability of all Native Mobile Applications running on the device.
- Analyze the Web services in use from the device to the end server.
- Verify the authentication process from the device to serving gateways.
- Determine the access controls for gateways and aggregator devices.
- Analyze the message delivery round-trip time for the M2M device.
- Analyze the results of data manipulation generated by outside sources.

Targets

We identified the following device targets based on the data analysis from the example use cases.

1. Web Applications: Use Cases 1, 2, and 3 are all web interfacing scenarios with numerous opportunities for attackers to inject malicious code. These devices run the risk of becoming weaponized attack tools that can be used for SQL injection or to expose cross-site scripting flaws. These attacks can cause Denial of Service (DoS), XSS and HTML Injection errors, and attacks against web-facing applications. They can compromise sensitive information that is stored on devices (de Ipiña et al., 2005).
2. Authentication: Authentication of a device is of utmost importance within any system. However, traditional authentication schemes always assume that a person is present. M2M devices’ access methods are sometimes limited. Specific security requirements that are based on Use Case demand must meet

the unique needs of the framework and formally model the authentication for the ecosystems (Cha, et al., 2009).

3. Authorization and Insufficient Transport Layer Protection: Use Case examples 2 and 3 call for sensitive data transport. The M2M devices store local data and configuration files with limited resources. Protection of the user's privileges and discovery of bypass methods is required if the device is going to security transmit and store sensitive data; this includes the access to transport messaging and location data (Kothmayr, et al., 2013).
4. Unintended Data Leakage during Session Management: The devices and network management session must try to avoid aggressive transport of short sessions or transport of very long sessions of data to avoid data leakage. The processing power behavior of small devices must reduce signaling and power overhead to protect sensitive information. This includes logging and transition data, overhead messages between other components and devices, and sensitive user data. These devices are low powered and resource limited in processing, so they require smaller encryption keys than other devices like Smartphones, to protect against attack (Song, Kunz, Schmidt, & Szczytowski, 2014).
5. Cryptography: M2M devices are vulnerable to brute force attacks. If, in any of the Use Cases, such an attack is successful, the ecosystem as a whole is placed in danger. These attacks expose applications and data information to the possible reconstruction of encrypted messages and exposure of weak

protocols in a Life/Safety function, which could cause harm or death (Gyrard, Bonnet, & Boudaoud, 2014).

6. Untrusted Inputs and Binary Protections: M2M devices have limited file and data storage capabilities in the cache and drive space. It is imperative for security that no unprotected data is left on the devices and that unencrypted data storage is controlled, because this is where other weaknesses can be exploited (Dye & Scarfone, 2014).

Exact Tests

The exact tests have been outlined in the tables as shown in the Appendix A. section. Described are the tests that were executed to verify device compliance and function capability as required by the defined use cases, the results then were used to build a “Functional Security Design Framework for M2M Devices”.

When the results were shown to be different from expected it proved that the typical security practices require additional assessment and development leading to future research opportunity. This outcome also proved that the devices do not meet the correctly assigned capability required as per the use cases, because the corresponding security controls were not identified during testing. For a functional framework to be “functional,” the strength of the controls employed must be equivalent to the sensitivity level of the data. If the device classification is “only” protected, this means that insufficient security has been applied. The device and the data transported by the device will not address the overall role of a device in an M2M ecosystem. The business logic and the potential impact sections of the use cases should lead to the security of the device and the capabilities will drive the development of the functional M2M security framework. We used these sections to gain the knowledge to determine security controls, formulate better requirements and drive security architecture at the design stage of the device.

When the devices did not meet the requested function, then requirements were added or device components changed until the logic, data type, and potential impact balanced the risk. Simply discovering vulnerabilities does not estimate the associated risk to the business, user, or ecosystem. The Repeatable Method approach was used to allow for the evolutionary process to take place and for the discovery of new countermeasures against potential risks to the business, device, and user of the ecosystem to be realized.

A Repeatable Method approach is required because vulnerabilities that are critical to one use case may not be very important to another use case. We declare that functional frameworks should allow customized changes and retesting for each particular use case. This flexibility helps to develop solid security requirements that satisfy the overall roles and classifications of a device placed within any particular end-to-end M2M function.

Format for Presenting Results

In order to create a framework methodology, the following outline was used as a guide:

1. Define specific functions and operations within M2M services.
    - A remote device control, for example, provides automated environment controls based on business-driven requirements.

2. Look at the properties of the various functions and operations within the service.
   - Does the function involve any of the following data?
     i. Non-sensitive information like error reporting
     ii. Personal private information like legal/ or regulatory statutes or protected data
     iii. Personal confidential, like identifying number addresses or assets
     iv. Payment/financial information
     v. Critical harm or financial loss
     vi. Life and death or could cause harm
3. Identify and classify the functions in the M2M system.

In the case of M2M, it is important to move beyond the data and network elements when determining security controls. Elements in the M2M ecosystem may make automated decisions and take actions based not only on data, but also on associated business logic (Aslam, Gehrmann, & Björkman, 2013). These actions may be referred to as functions that the device is capable of performing. Within some scenarios, the same devices support functions of varying importance. Therefore, a method was needed to determine security controls for M2M that would classify the overall functions and apply proper security controls based on that functional security need as balanced by the device's functional capabilities. An M2M ecosystem will have multiple functions and needs. Each function must be identified and classified to ensure that correct security controls are identified, that security recommendations are made, and that trust exists (Aslam, Gehrmann, & Björkman, 2013).

This research aimed towards a new and better security framework for the devices that depended on real-world device use cases, design-centric data collection, investigation of device changes, and device behaviors when security controls are applied to the M2M ecosystem (Accorsi, Stocker, & Müller, 2013). Studying the foundation framework helped organize ideas and led to the development of a better framework based on efficient device tasks.

Validate Methodology by Executing Within an Actual Production Project

In order to develop a new framework, the following inputs from the theoretical functional classification framework were used:

- Business logic (rules within the M2M application that define actions)
- Device data (inputs and outputs to business logic)
- Potential impact (consequence if the function is manipulated)

These three inputs determine the overall functional classification of the Use Case, which will helped to ensure that security and privacy within requirements can be applied once the functional classification exercise has been completed (Abie & Balasingham, 2012; Godfrey et al., 2015).

M2M devices must be designed to perform a particular function. These functions can be classified into categories ranging from "non-impacting" to "life-threatening" (Godfrey et al., 2015). The final framework takes into consideration that the security controls applied to a particular device must be equal to the applied "classification" of the device. Devices and the components inside these devices often lack fundamental security controls, such as secure boot, authentication and authorization, secure update capabilities, and encrypted communications. Framework rules created with application and device control-based functional classifications described lead to the development and enforcement of new industry standards and security policies (Alqassem, 2014).

Resource Requirements

Three M2M connection kits featuring the GSM/Wi-Fi modules were used to meet this study's goals. Also a M2M test bed for the development and experimental assessment of the framework's security controls was built for the prototype devices. These devices are GSM/GPRS/EDGE devices that are running Unix operating systems and use the Web based API control application for management from a desktop. These kits cost $999 each at the time of this research. All software used was shareware or open source. The M2M Control Center is provided free of charge as part of an innovation developer program. The program offers a set of developer tools and real-time visibility into the device and network behaviors. Also, built for this research was a prototype Wi-Fi test bed system for device scanning and testing and a prototype 2G Cellular network test bed using OpenBTS and USRP radios to simulated the cellular network for security and performance testing.

Summary

Security Frameworks are a set of tools that can be used to develop requirements for devices and systems. These frameworks can also be used to test new business models, seek problem solutions, and forecast functionality. This chapter outlines efforts that had been taken to refine the initial framework into a novel framework by testing new models and developing new solutions to offer a new security framework for M2M devices.

Chapter 4

Results

Introduction

This research focused on the development of a new framework for securing M2M type devices by applying the System Development Methodology. The goal of the analysis phase was to improve the knowledge of present processes, services, and functions. Researchers are concerned with present technology and how this technology may improve and be made more secure (Vaishnavi & Kuechler, 2015). As described by Vaishnav and Kuechler (2015, p 285) new frameworks may be developed using a Model-Driven Approach and drawn out of logical models that are developed from data collected during the prototype testing and analysis phase of the greater method.

The testing approach included both “requirements verification–based testing” and “attack based testing”. The testing followed a pattern-based methodology leading to a step-by-step procedure for repeatable evaluation of M2M type devices. This method provided a baseline for future testing by finding vulnerabilities on “function” oriented devices. The baseline was formed around the air interface on both Wi-Fi and Cellular Wireless vulnerabilities. Industry standard techniques that have revealed results when applied to similar devices such as smart-phones, Wi-Fi based devices and embedded device type systems were applied (Gubbi, Buyya, Marusic, & Palaniswami, 2013). As explained in Chapter 3, Table 2, the OWASP Security Projects theorizes ten risks to various ecosystems. Determined was that the hypothesis that “vulnerabilities are transferred to next technologies” is true when applying the described testing methods.

System and Testing Analysis

The requirements determination phase of the Systems Development Methodology for systems analysis is “prototyping”. As explained by Dehghani and Ramsin, (2015) prototypes take on many forms and systems; process and models can be prototyped. The Prototype Methodology is a technique and supplemental methodology of the Systems Development Methodology. Holmlid and Evenson (2007) explain that prototypes explore future reality by distinguishing and comparing exploration and demonstration. To design the final framework we distinguished and compared prototyped physical testing results with prior literature review results. The modeled theorized security framework is taken under analysis by executing both functional and evolutionary prototyping categorizations for output evaluation (Dehghani & Ramsin, 2015). The functional test verifies actual system functions by using real and known attack scenarios. Once expected and actual results were verified and known, the evolutionary approach was then applied, producing reliable requirements for the better framework and final operational system. The operational system is built from the knowledge gained during the prototype testing which also led to a better understanding of the requirements required to secure M2M devices.

Findings

This research focused on the device hardware, simple task oriented applications and data transport means. Test (1) was the execution of full vulnerably scan to identify obvious attack points (Hager, 2013). This approach focuses on identifying running services of the device such as operating systems and open ports. Once known, next steps to follow are realized and executed. All test were run within private wireless networks consisting of various devices including two smartphones and three prototyped M2M-type devices.

Weak Server Side Controls

Expected and Actual Results

Results proved that there were vulnerabilities on all devices. Baseline vulnerability scanning was applied and directed towards analyzing “Weak Server Side Controls”. This threat usually includes an untrustworthy input to a backend API service, web service, or traditional web server application. When reversed, the process and focus is from the attack direction, originating from server towards the M2M-type device interface. It was found that if an adversary sends malicious inputs or unexpected sequences to a device from a server the devices becomes a vulnerable endpoint and reveal attack vector.

Server-side control attacks are an important security threat when pointed towards the device. Findings proved that when a user acts like a serving system and performs vulnerability scanning this identifies open ports on the devices. Also found during the scan was the assigned IP addresses, the device’s operating system, software, and services that are running on the device. The additional finding showed that when devices are vulnerable to applications and have open communication ports they are exposed to attacks such as DoS, malware infestation and Cross-Site Scripting attacks. Scanning for known vulnerabilities allows for the discovery of access opportunities. Also, discovered was that these test will impact the device communication protocols within the service-layer and the storage of encryption keys for authentication.

Improved requirement

Critical M2M devices should be protected against port scanners by hiding or closing all unused TCP and UDP ports. Implementing Internet Protocol (IP) filtering and other firewall techniques on a device level will close any open connections to active sockets and protect the device from discovery. Also, device applications must ensure that only required ports allow incoming connections and devices required to send content are patched with the latest security updates.

Insecure Data Storage

Testing found that configuration settings impact data storage and expose vulnerabilities to the privacy and security of the messaging data. This is important because, host discovery, port scanning, operation system detection, and service discovery all expose the running applications and stored data located on the device, which can be used to launch other attacks.

Expected and Actual results:

It was expected that the results would show that the standard behavior for each of the events is capable of being identified on each device because there are no additional transport protections implemented. As expected, the actual result proved that various TCP and UDP ports are opened on the M2M devices as well as the other network connected devices. Knowledge gained from this test was that the devices are prototyped and the ports can be turned off or on. However, for the device to communicate seamlessly, applications such as SMS messaging must remain opened. This case showed the device to be vulnerable to various known SMS attacks.

However, being able to scan a device is not a bad thing. Discovering vulnerabilities or configuration errors results in understanding where intrusions can occur and leads to the development of better countermeasures.

Improved requirement

M2M devices that allow for systematic scanning and allow review of process running should be controlled in a known manner by a central gateway or system. Other devices require secure countermeasures against random port scanning. Devices need a method for quarantining the applications and revoking the permissions after the applications are closed. This method will prevent exposed ports, allowing only ports that are required for data transfer to be exposed. In addition, for devices that communicate over the cellular network, updating the device communications module to 3G and placing a 3G or above smart card into the device for authentication and authorization would increase security and strengthens server side controls protections.

Insufficient Transport Layer Protection

Findings proved that insufficient transport layer protections lead to improper session handling. Service-layer keys and the storage of keys on the device is a known device vulnerability as described (oneM2M Partners, 2013). The service layer consists of all the services that the manufacturer makes available on the devices or that is preinstalled by device peripherals. The long-term service-layer consists of keys that may be discovered while they are stored on the devices. If discovered, these keys may be copied and used against the device or gateway during other attacks (Ukil, Bandyopadhyay, Bhattacharyya, & Pal, 2013). The services layer is highly vulnerable to attack because the data within the layer provides the business functionality that allows the devices' supported communications and messaging to and from the gateway. If copied, the long-term service-layer keys may be used to impersonate M2M devices to the gateways or vice versa (Latvakoski et al., 2009). When long-term service-layer keys are stored within the M2M device or M2M gateways, they may be discovered during scanning by unauthorized entities (Minoli, 2015). For example, Transport Layer Security (TLS) uses a cryptographic system with two keys to encrypt data (Hersent, Boswarthick, & Elloumi, 2011). Once the keys are discovered, they can be used for illegitimate purposes, such as false authorization and authentication. There are various methods for discovering open and available keys. Hardware probing methods include the monitoring of internal processes or simply the reading of memory contents. According to Lu, O'Neill, and McCanny (2010), DPA is a widely studied side-channel attack however this attack is outside the scope of this research.

Improved Requirement

It is recommended that Random Delay Insertion (RDI) be deployed as a countermeasure technique. However, M2M devices run weak cryptographic processes and do not have the resources for increased countermeasures that will reduce the risk of DPA attacks.

Expected and Actual results

Findings showed that devices are impacted by the device communication protocols within the service-layer and the storage of encryption keys for authentication.

Also found was that wireless networks and configuration settings are impacted by these found vulnerabilities because the privacy and security of messaging is again at risk. There is no indication that these devices are protected and analysis proved that ports that are open and running on the device which can be exposed, leading to authorization and access to the device by rogue application execution.

We expected to find security vulnerabilities against the open ports and testing proved that there are inherent limitations to resources on these devices. Actual results also showed that data is stored read-only with limited in storage time as required by standard policy. However, the devices tested are prototyped devices and unsecure. So, as expected when testing Layer 2 with port pinging and scanning, open ports and device information, including the TCP IP Address and MAC address was exposed. Testing also exposed that with low processing devices, repeated testing, freezes the devices and takes them off the network. In particular, an Address Resolution Protocol (ARP) request, does determine that the host is alive and provides the MAC address of the devices in the return message. When pinging the device (ICMP echo request) on all open ports using tools for flooding messages, the device fails because the flood message stresses the CPU to MAX usage.

Improved Requirement

A better design for production M2M devices would be to limit or completely shut off of all “Ping” operation and stop Internet Control Message Protocol (ICMP) echo request packets from targeting the device and that all IPv6 security measure be used on M2M devices.

Network Layer Protection

Management and control frames are directly related to service-layer keys and focused on the Layer 3 frames because M2M data is serviced by three present requirements: massive data analysis, real-time data analysis, and deep data analysis (Kitagami, Yamamoto, Koizumi, & Suganuma, 2013). Information stored in the devices can be detected and used by an attacker to compromise these systems. The device is also at risk of being spoofed or turned into an attacking device or for attaching to a fake access point or a fake base station. Management and control frames must be protected to protect the protocol stack (Lin, 2008). Unprotected protocol stacks may lead to denial of service (DoS), Man-in-the-middle (MinM), and similar attacks. This vulnerability can prevent the operation of the overall M2M service.

Expected and Actual results

Test directed towards Network Layer Protection and Insecure Data Storage found that M2M devices are vulnerable to stored management and control frames discovery on Layer 2 as described by Hersent, Boswarthick, and Elloumi (2011). When testing the possibility of locating service-layer keys within the protocol stack and the device, it was found, that these keys could be reused, deleted, or changed. The M2M device tested use AT COMMANDS as input and out messages. These commands are transferred over the service-layer and a compromise will return information that impacts the constraints on the device. The discovery of authentication frames and open data transfer may be used for Layer 3 attacks such as DoS and Man-in-the-Middle attacks. The analysis found that there are weaknesses in the various protocols. These are known vulnerabilities in the GSM stack. As expected, sending different configuration messages to the device on each port exposed the GSM vulnerabilities. Running successful scans against layer 3 and layer 4 discovered that the host is alive. However, the open and closed TCP ports did fail to return an “Acknowledgment” to messaging and did lock the test devices when a particular port was addressed in the execution string.

Improved Requirement

When deploying M2M device communications, each device and gateway should be isolated over a shared network infrastructure. This network should provide management and provisioning using a tunnel type protocol for secure data transfer and access authentication.

Unintended Data Leakage due to Improper Session Handling

The M2M devices’ Open Standards are untested because of lacking transparency and undefined guidelines; the use of open standards makes the device vulnerable to attackers (Torbensen, 2011). Because these devices are usually built from commercial off-the-shelf (COTS) hardware and software, attackers can seek knowledge to expose open vulnerabilities (Igure, Laughter, & Williams, 2006). Many M2M devices run a small version of Linux as an operating system. Components such as cameras and Wi-Fi nodes are run and accessed through files that are usually located in the file systems of the operating system (OS) and these files communicate directly with the kernel driver that communicates with the component hardware; leaving them vulnerable to data leakage attack (Yaoming, 2010).

Expected and Actual results

Unintended data leakage and improper session handling testing indicates that sessions can be easily exposed and, once found, can lead to vulnerability. As environmental conditions change, the RF signal, in or out data, and messaging should all continue to match a specified value as described by the standards. Actual results showed that vulnerabilities are successfully found over Wi-Fi and GPRS, proving that needed information to ensure a successful penetration attack is available and device compromise probable. The knowledge gained was that when using a threat assessment tool the complexity and severity of a single point of failure on the device is identified by open IP addresses, active machine names and opened various port identification points; protocol vulnerabilities. This information leads to the discovery of running services on specific ports for exposure by untrusted inputs. The interfaces are then exposed to attack and directed by security decisions, untrusted inputs, and insecure data storage vulnerabilities. The discovery proves that the devices lack human user interface and tampering resistance notification as described by Hagar (2013).

Improved Requirement

All components must communicate transparently, regardless of their hardware and software (Bernardi, Merseguer, & Petriu, 2013). This is important because the critical infrastructure includes the supervisory control and data acquisition (SCADA) networks such as the natural disaster early warning systems, crime prevention cameras, and a range of vulnerabilities from equipment failures to terrorist attacks that threatens the ecosystems (Igure, Laughter, & Williams, 2006).

Protocol tampering and device repurposing

M2M systems are designed to function without human interaction. However, M2M message traffic, data transfer, and content are influenced by human-based traffic, such as location data, billing data, and personalized content. The M2M device must also be monitored, repaired, and managed by humans (Cha, et al., 2009).

Expected and Actual results

It is found that ability to detect tampering and design flaws in the device is not possible because there is a lack of monitoring systems that allows for remote device-user interaction.

Improved Requirement

M2M devices may be targets of physical tampering, repurposing, or modification and require that failure indicators, such as on/off settings, hardware status, and network control or alarms that transport over alternative networks be in place. These fail indicators will allow an operator to take action on alarms and protect against an attacker performing remote hacking on management or maintenance interfaces or using the device as an attack tool.

Unrestricted File Upload/Download

Findings exposed that testing directed towards unrestricted file "upload/download" reveals the possibility of an unrestricted file upload to the M2M infrastructure from the device (Flick & Morehouse, 2010; Skianis, 2013). The vulnerability affects databases, operating systems, and applications that are developed for internal use and the end severing systems. In the case of M2M ecosystems that are connected to the Internet with unrestricted access, these platforms lack restrictions on the size or number of uploaded files that are sent to and from devices. Files that are too large will consume resources and freeze the device operations (Flick & Morehouse, 2010). When open access to SSH or FTP servers is found the device data storage is at risk. If an attacker uploads or transfers files of dangerous types using an automated processed within the M2M ecosystem, for example, via open FTP ports, the ecosystem does not block the "input file" and actual result show this to be true, making it possible for the device to be weaponized.

Expected and Actual results

As expected there was the discovery of UDP ports and these ports are open and actual results prove that these ports remain open after message transfer. However, the tools identified that open ports are only open until closed by the communication software. It was observed that it is very easy to identify hosts using discovery against UDP and probing an isolated task and that the ICMP host are unreachable when requesting "responses" to identify live hosts with UDP requests on closed ports was discovered as expected. Additional scanning reported that all open ports for discovery allow for fingerprinting of services. Testing discovered, for example, that port 22 (SSH) is opened on the device but the other ports are closed. SSH is opened at install of the OS and not closed at reboot leaving an open door opportunity for device compromise.

Improved requirement

M2M devices must protect against the discovery of sensitive data. The protection of sensitive data in M2M devices and M2M gateways leads to protection against broken cryptography and lack of binary protections (Pandey, Choi, Kim, & Hong). M2M devices, such as sensors, collect data that is sensitive, including toxic levels of poison, the temperature of machinery, and personal consumer data. The execution of sensitive functions and the storage and transfer of this information must be protected.

Broken Cryptography and Lack of Binary Protections

According to Pandey, Choi, Kim, and Hong (2011), the M2M ecosystem has important characteristics that other ecosystems do not, like, sleeping devices, low power devices, weak signal networks, and low device intelligence. Because of this the ecosystem is vulnerable to automated service discovery. In such an environment automatic execution of software and storage of sensitive data may lead to a higher level of compromise potential (Cha et al., 2009). Needed is a workable level of encryption on all communications between the device and server that takes the device limitation into consideration and protects against the attacking of sensitive functions within the M2M device.

Expected and Actual results

As we expected, it is possible to capture sensitive information that is outbound or sent to the device. However, the data itself is protected by the communication protocol security policy. Actual results proved that data is protected in the cellular network because of the secure communications protocols, access and authentication methods that are provided by the SIM Card. Knowledge gain included that if encryption is turned on then all transfer communication is protected, however, as expected, when off, information is transferred in the clear. Without proper communication encryption the devices are subject to attacks such as eavesdropping.

Improved requirement

It is important and recommended that there is no exposed sensitive data and that protections are in place to prevent unauthorized entities from using this data for illegitimate purposes, transmitted data should be deleted from the device. Eavesdropping of cryptographic resources discloses identities and exposes knowledge of sensitive information. The ease of eavesdropping during prototype testing revealed that poor authorization and authentication and broken cryptography are vulnerabilities with M2M devices over Wi-Fi, Bluetooth and Cellular. It is expected the other radio technologies, such as ZigBee are just as vulnerable but more investigation is recommended as future research. Eavesdropping testing found that the M2M device service-communication protocols that connect the device to the gateway are vulnerable as described (Kylanpaa, Rantala, Merilinna, & Nieminen, 2013). In addition, as reported Ren, Yu, Ma, and Ren (2013), all real-time, wireless communication-oriented overlay networks, like M2M capillary systems and federated systems, operate within “registration based” versus “location lookup quire” architectures. This architecture requires special security requirements because vulnerabilities may result in a breach of security from real-time eavesdropping on data transfer and messaging.

Eavesdropping

Also, results found that the feasibility of eavesdropping on the M2M Service Layer does expose messages between internal components and the available cryptographic resources that may expose confidential or private information.

Expected and Actual results

The network and device tasks are separate from each other by the various inside security systems including the SIM cards and OS security containers. However, when testing over cellular there are known weakness in the GSM protocol stack that is exploitable and makes eavesdropping possible. These protocol weaknesses also may lead to jamming, false base station impersonation and cipher vulnerabilities in GSM.

Improved requirement

Layered security at chip design is required, such as a layered approach will blanket the device and focus on the device as the secure endpoint from the component level. Also, applications should have a separation of functions that prevent tasks from being hijacked, which would make them less vulnerable to eavesdropping.

Jamming

M2M ecosystems connect to various types of radio frequency networks. An attacker may purchase commercially available jamming devices with enough power to jam a radius or radio ranges from a few feet to over a mile.

Expected and Actual results

It is found that jamming attacks do block and degrade the radio channels that connect the M2M device to the base station and Wi-Fi access points. These attacks cause a Denial-of-Service attack towards the device. This is an “over-the-air” attack and is executed by either sending strong signals over the same frequencies used by the device or by directly consuming the channels available at the base station. Actual results prove that the injection of RF amplification, noise, and spurious connections effectively disrupts the communications between the device and the cell tower. In other words, jamming denies service of the radio spectrum to the cellular network within range of the M2M devices and forces the device to seek a signal making the device vulnerable to a false base station attack.

Improved Requirement

M2M devices are low power devices that are vulnerable to both active and passive jamming attacks that degrade or completely block all communications in a prescribed area of operations. It is recommended that in critical communication scenarios, anti-jamming techniques be deployed such as a gateway with hopping signals for RF transport or multi-radio (ZigBee, Wi-Fi, and Cellular) capabilities are used.

Network Impersonation/False Base Station

It is possible to attack a legitimate GSM network with a modified Cellular Base Station (Fake Base Station). The goal is to exploit the weakness in M2M devices when the device seeks the strongest radio signal from the GSM network. Once locked on to the strongest RF signal, the device camps on the false base station. The false base station assigns radio channels that look legitimate to the device. The target device is then out of reach of the authentic carrier’s paging signals. The device registers to that network, believing it is locked on the serving network, such an attack is comparable to radio jamming and is very difficult to counteract effectively in any radio system (3GPP, 2002).

Also, a compromised base station can act as a repeater. In repeater mode, the attack platform functions as a relay for requests to the legitimate network. These systems are located in-between the network and the target user, causing a Man-in-the-Middle scenario. In this case the legitimate service requests and/or paging messages for the target M2M device can be modified or ignored by the attacking system. In the security architecture of most wireless systems, there is no prevention mechanism against the false BTS relaying messages. As previously noted the device is only seeking authentication.

Expected and Actual results

As expected it was found that in the security architecture of 2G/3G cellular, there is no prevention mechanism against the false base station relaying messages. As previously noted the device is only seeking authentication. During actual testing, results proved that transmitted packets could be received at the device leading to the impersonation of the network. This is the capability whereby the intruder type system sends signaling and/or authentication data to the device in an attempt to make the device believe the authentication originate from known good network. However, this is a known problem with Wi-Fi, 2G and 3G cellular networks and leads to a false base station (cellular) or rogue access point (Wi-Fi), once the device is locked onto an attack system the vulnerabilities with in the communication protocols can be targeted (3GPP, 2001).

Improved requirement

M2M device should have firmware that detects and fingerprints all connected network characteristics.

Risky Communication Protocols

Weak communication protocol design leads to session injection (Fuzzing) and exposes vulnerabilities and other weaknesses in the transport layer. Protections against client side injection and security decisions via untrusted inputs were identified during testing and a needed countermeasure for M2M devices is required. Alteration of M2M communication protocols and messaging between devices and gateways were proven to be vulnerable as described by Sheng et al., (2013). This validates, the Chen and Ma (2014), statement that the different communication protocols presently in use within the wireless and wire-lined systems will cause a protocol security gap. This gap may potentially lead to a threat against the M2M ecosystem. Protocol level device attacks include Man-in-the-Middle and Denial-of-Service attacks. Exploitation of network services weaknesses and over-the-air management is also vulnerable. When exploring the transaction layer and its protocols, it was explained that message exchange between the devices and gateway can be altered in the M2M ecosystem by forcing the device off the core network and executing fuzzing attacks as described by Chen and Ma, (2014).

Expected and Actual results

The device to gateway communications defaults to the cellular 3G protocols and authentication method when a 3G SIM is used for access and authentication, protecting against known 2G vulnerabilities. However, actual results proved that the device can then be forced back to 2G, allowing manipulation of the interaction between the device and network. Knowledge gained included that the devices with the GSM/GPRS SIM will register with the open networks. Also, it was verified that the devices are not resistant to

GSM fuzzing when flooded with SMS type messages. However, a full suite of created test cases was not applied for full fuzz testing, this is recommended as future research.

Improved Requirement

It is recommended that protections be placed on the device at each individual component level, a device is not security if all components are secure at the design stage. These added active security protections would protect the device's operational environment and will restrict access to data, protect message transport and secure firmware access.

Messaging Manipulation

Any alteration or manipulation of the messaging protocol may lead to the readability of sensitive information or modification of message content (Chen & Ma, 2014). Replay Messaging facilitates false base station roaming because of improper session handling and lack of transport layer protection (Latvakoski et al., 2014). Latvakoski et al., (2014) adds that the transport data, signaling data, and control data require added security measures for correct transmission of messages. This data passes between the devices and gateways through the physical layer or protocol layers. If the protocol is compromised, all services are affected.

Expected and Actual results

We found that when a replay attack occurs an attacker can copy messages between devices and gateways and use them to defeat authentication. Replay attacks damage transaction information by allowing for the modification, insertion, or deletion of legitimate user data or signaling message structures. Leading to the discovery of the device because of the lack of identity mechanism and lack of protection against deceptive or fraudulent message content. As expected, result concluded that the attack can exploit the lack of protection in the communications service layers and that this lack might lead to replay or playback of network data transmissions. Also, actual results concluded that the devices roam and register with the false base station only when the SIM card is allowed to drop to 2G cellular coverage. We gained the knowledge that M2M devices can be compromised and that if a gateway is used as a base station, the gateway can also act as a repeater. In repeater mode, the attacker may compromise various functions by requesting data and access into the legitimate network allowing a Man-in-Middle type attacks.

Untrusted Inputs

When considering the Man-in-Middle type attacks as another version of unauthorized or corrupted applications of untrusted inputs, these expose vulnerabilities from unauthorized, corrupted or modified messages to and from M2M devices (Ho, et al., 2013). Jeon, Lee, Park, and Jeong, (2013) explained that most unknown replay attacks are not detected at the device level, this will be even more exposed in a mutual cluster authentication or mesh network architecture environment such as those in a capillary M2M ecosystem. Unauthorized devices may run software that authorizes functions to create vulnerabilities that impersonate the network and device management platform which might expose other connected devices (Liu, 2012).

Expected and Actual results

We found that the feasibility of an attacker to impersonate the network and device management platform by temporary failing the input and output communication protocol is plausible. Once completed the attacker could fail the device by reporting fake device consumption, breaching privacy, or reporting confidential information; which would lead to the attacker's control of remote management functions. As expected the communications protocols that are utilized by the device have known weaknesses, which allows for the manipulation of interaction between the device and the end gateway. Actual results proved, that the gateways provide services via the packet switched protocols that may be subjected to attack. Other communications protocols like those between the core-network of the cellular network architecture, outside databases and networks elements like switch routing and management functions, may also be vulnerable (Huber and Huber 2002). We gained the knowledge that in M2M ecosystems that are using primary radio/antenna component for GSM networks, the IMSI catching attack using the false base station transceiver it is possible to attack in the same manner as that of mobile phone attacks. As described, the devices see the base station as a legitimate carrier's network. Per specification and design, the devices seek the "best" power received transmission, once found the device transmits its identifier data, such as IMSI and system interdependencies. The Man-in-the-Middle attack is also possible because of poor authorization and authentication methods (Kim, Jeong, & Hong, 2013). According to Kim, Jeong, and Hong (2013), it is difficult to detect or prevent the Man-in-the-Middle attack in the M2M ecosystem. In IoT, the problem is elevated due to the difficulties of managing or controlling each device independently within the ecosystem and the probability of federated networks and device-to-device communications.

Improved Requirement

It is recommended that new approaches and standards for the ecosystem apply new security requirements that address the resource constraints of present networks and device. This vulnerability is possible because of the device inability to detect replay messaging. The IoT ecosystems lack requirements and methods for protection against these type attacks.

Situational Recognition

According to Jin (2013), M2M ecosystems lack the proper device situational recognition within the systems that certifies the platform and message protections. Today, these systems use identity-based algorithms for situational recognition and a convergence framework to analyze certification technology. In addition, Man-in-the-Middle attacks target integrity and confidentiality from a messaging standpoint, which allows the attacker to take over as the core network by representing the gateway to the device.

Expected and Actual results

As explained and expected, the M2M devices successfully roam to a fake network that is impersonating a gateway because they look legitimate to the device. Actual results prove that the Man-in-the-Middle attack vector exists and can be executed from various devices and tools (i.e. AP or fake AP capability and RF spoofing).

Improved Requirement

M2M devices require a new and higher level of authentication to protect the end-to-end communication and data transmission.

Unintended data leakage

Improper session handling leads to venerable data storage and unintended data leakage. M2M networks have sector to subsector interdependencies that when threaten will lead to cascading impacts across domains (Macaulay & Singer, 2011). Improper message content delivery affects the overall ecosystem environment, because the information that a message contains included the various contexts of interdependencies. For this research, interdependencies are defined as types of intelligence that are actionable, such as asset ownership, location, and device role (Bianchi, 2014). At risk are the underlying systems and resources that may impose many forms of vulnerabilities on interdependencies that directly relate to failures of the ecosystems' critical infrastructures. For this research, interdependencies are defined, as anything that can be used as an attack vector and that shares resources with other applications or devices.

Expected and Actual results

As expect it was found that the principle of least privilege is enforced by the device's operating system and that all system dependencies are secured to the same or higher level of assurance as other programs. Actual testing found that no privilege user information is stored on the devices that were tested. This is also true when reviewing data logged at the control center; the center applies context awareness and only provides device level information such as MEID, IMEI, SIM ID and OS VERSION information.

Improved requirement

It is recommended that context awareness be provided to M2M devices in the same manner as it is deployed on mobile devices. This means that all stored relevant information; including the owner of the device, network authentication keys, and other specific access policies must be protected (Cam-Winget & Didier, 2014). Lack of context-awareness will break applications such as device authentication and key generation (Hagar, 2013). If there is a lack of context awareness it threatens how the device functions over the M2M ecosystems.

Lacking API protections

M2M devices transmit to other devices including sensors, actuator, gateways and end systems. For successful security and scalability a secure application-programming interface (API) must be used to ensure that the protocols are protected.

Expected and Actual results

As expected all the running applications used for external communications to the network functioned as designed. Protection for the authentication and prevention of manipulation of the messaging protocol are in place at the clients and servers. However, actual results did show that the devices expectedly will move to rogue access points and false base stations. Once connected to these networks the device may be accessed using the vulnerabilities in a published API. The device uses a published API during communication with the Control Center. Increased knowledge added that when handling exceptions, such as when the device is moved to a rogue network, there is no error message sent to or from the Control Center.

Improved requirement

A better messaging protocol would apply an “alarm” type message. M2M devices are being deployed as a “one size fits all” box that will function as a tool for other devices, gateways, and applications. This increases the risks associated with security, privacy and data protection.

Buffer Overflows

The application framework of all devices manages the functions that performs various tasks; like resource management and call management. When an erroneous condition, such as a buffer overflow, occurs, the device processing power is stressed beyond the boundaries of the store data buffer limits and may lessen the difficulty of a buffer overflow condition (Hagar, 2013). This condition leads to extra data overwriting and failure of the device processor’s memory locations, which causes the device to fail (Shewale, Patil, Deshmukh & Singh, 2014). A buffer overflow condition attacks corrupt data, crashes the program, or causes the execution of malicious code. As described by Hagar (2013), when sending data and messages to various abbreviations of application, program interfaces (APIs) show no “failed error messages” or device side alarms. All APIs are designed to have length constraints for the utilization of storage, data locations, and code. These constraints define execution space and help to find any vulnerabilities that enable the execution of applications.

Expected and Actual results

As expected the exploitation of buffer overflow vulnerabilities can be exposed. What is important is at what point and the measurement of “ease of action” of the testing.

The expected buffer overflow happens at the point where the message is received and the device tries to store the message. Although, actual results found that no buffer overflow is found on the prototype devices, it is important to note that the serial data transmission is very slow and failed. Knowledge gained was that the communications module on the device has a limited buffer size and the header size fails when bigger files are received; causing a buffer overflow.

Improved Requirement

It is recommended that limiting the buffer size and using a GET method with short answer for all requested and using AT COMMANDS for HTTP communication should be limited based on device function. Increasing the buffer size leads to false data injection opportunity and increases the likelihood of client side injection (Lu, et al., 2012). According to Lu et al. (2012), networks and devices like M2M are seriously threatened by injection attacks.

False Data Injection

These attacks threaten authentication if an attacker can discover the capability of bypassing administrative privileges. Once discover that attacker can view sensitive information and can alter contents stored in the device. If the availability of data is compromised, the authentication process may be defeated. In that case, all sensitive information on the device is at risk. Increasing this risk opens the device to remote command execution techniques that can send mass injections to the device by executing a simple text-based attack and injection vector.

Expected and Actual results

As expected injection attacks can be mitigated by strong authentication process such as 2G/3G authentication. However, the small size, low power, and unattended operations of M2M devices make a false data injection a higher risks because a compromised M2M device can launch other attacks. Actual results found that various forms of side-channel attacks can be performed against encryption/decryption algorithms flaws and vulnerabilities. However, SQL injection attacks against the device produced no failing results. This is because there are no Web applications running on the devices. Test against a client side injection attacks and HTML-5 cross-site scripting attacks also did not fail these devices. However, the fault injection attack, timing attack, EM analysis attack, and power analysis attack are known cryptographic attacks used against mobile devices successful. It is probable that these attacks will fail M2M devices due to the function specific task they perform. Knowledge gained contributed that fault injection freezes the device CPU. The timing attack was inconclusive based on how long the device takes to execute commands when the fault injection script is running, causing the device to reboot. This result is considered a successful DoS attack because it causes the device to fail the session.

Improved Requirement

M2M device should be task driven and limited to receiving messages and perform only function related operations to mitigate attack opportunity.

Lacking Session Management

Session Management failure will lead to improper session handling and broken authentication because of misconfiguration (Lake, Milito, Morrow, & Vargheese, 2013). According to Okugawa, Masutani, and Yoda (2005), M2M networks are self-organizing and composed of scattered small devices that require the survivability of simultaneous communicating endpoints, the identity of the device is a key to managing the moving parts of the ecosystem. If the session management or authentication is broken, all ecosystem functions are at risk of attack (Lake, Milito, Morrow, & Vargheese, 2013). Vulnerabilities in session authentication protocols lead to integrity and privacy issues because of key exchange failures and setup integrity flaws. These flaws fail at session shutdown and within authentication schemes such as logout, account update, and session timeout and device application methods of ensuring key privacy.

Misconfiguration attacks exploit configuration weaknesses that can fail account access protections, expose patching flaws, compromise unprotected files and directories, and grant unauthorized access to the device. Most devices are provided “off-the-shelf” with unnecessary and unsafe features that are enabled by default, including backdoor accounts, special access mechanisms, and incorrect permissions. Researchers have compromised device security configuration by reviewing the unauthorized access to sensitive information security policies. This leads to the compromise of the device and M2M ecosystem by granting unauthorized access to or providing knowledge of devices, accounts, applications, and platform.

Expected and Actual results

As we expected we found that the device environments are easily accessible and exploitable because all applications have permission to run, that based configuration files are not properly locked down, that clear text reveals username and password type data, and that database connection strings are set to default settings in configuration. Additional results found that services and applications can be turned “off and on” because the root login/password is known. Knowledge gained proves that there are design and development-related vulnerabilities.

Improved Requirement

Authentication or session management functions must verify device identity. Also, for message content there should be no exposed accounts, passwords, or session IDs. During open sessions it is expected that no data is visible other than the authenticated data and that data should be visible on the device only after the termination of the session and that all data should be removed from the logs after an inactivity timeout. An additional requirement is the importance that M2M devices are configured to perform as few tasks as possible to prevent security misconfiguration. If a device’s security is misconfigured, then various events can take place that may hinder the device’s performance (Hongsong, Zhongchuan, & Dongyan, 2011).

Useable Cryptanalysis

On-device platforms include local databases and file systems that usually have very limited access control protection. System data and credential are managed and stored in applications. Access to these internal systems must only be granted after explicit confirmation of the requesting entity.

Expected and Actual results

As expected cryptanalysis on the device requires improvements and needed are new methods to protect message context and secure plaintext data transport for low power/battery operated devices. M2M devices lack encryption algorithms that do not require large key sizes and keep data confidential during static events. Actual results found that the mobility characteristic of a wireless system/platform that a mobile application runs on, protection of application, user data and system data is very important in securing the M2M device.

Improved requirement

M2M devices must protect data on the system and must be stored locally, all information should be encrypted in storage using local key store, and file system protection should be in place and secure access to nonvolatile memory protected.

Unauthorized access

Valid input of data is required to ensure content is provided to applications securely (Ellinas, Panayiotou, Kyriakides, & Polycarpou, 2015). It is important that M2M devices and gateways have the functionality to detect and prevent unauthorized access to the ecosystem. In the case of ecosystem federation, there is the possibility to have non-existent communications security between devices, due to lack of requirements requiring protection against invalid input data and parameters that outline qualifier, range, and data fields. For mitigation, stronger message authentication is required and lightweight encryption should be used to provide confidentiality (Shah, Perrig, & Sinopoli, 2008; Awad, 2015). The M2M ecosystem passes communications over various networks and these devices are capable of sending messages and data using protocols such as HTTP or SIP. Cross-site scripting (XSS) attacks target events and may allow for code or data injection that disrupts the communication path (Siewruk, Sredniawa, Grabowski, & Legierski, 2013). As described it is feasible to use forbidden commands to bypass filters on a device using alternate forms of messaging syntax, which will cause the device to fail when processing, this is due to the protocol weakness using a cross-site scripting process by executing arbitrary commands from SMS messages.

Expected and Actual results

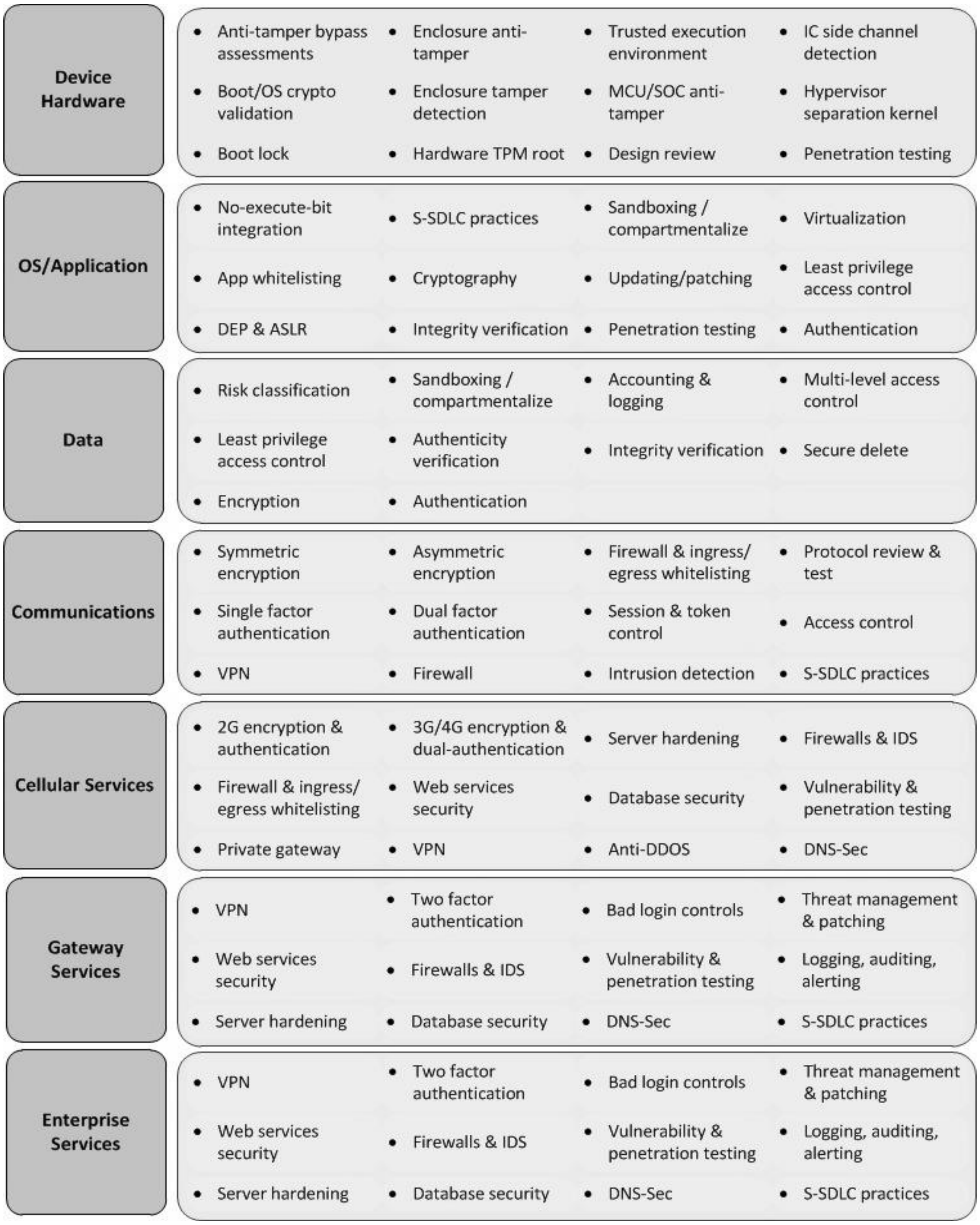
As expected the ability to bypass filters where “scripts” are executed is a prohibited functionality and all input from the server side is validate. The data-input does not fail the device when sent as described by the standards. The device validation application prevents and protects unauthorized input from infecting other on-board applications. Actual results employ automated tools and scripts in a non-reduced time frame and led to the potential cross site scripting, verbose errors and forceful browsing as expected with typically automated tools.

Improved Requirement

Only authorized specific types of data should be sent to and from the device. Maintenance and control messaging should be completed from a protect platform such as a firmware over the air (FOTA) platform, that protects the devices.

Summary

Findings support the hypothesis that the M2M ecosystem is even less understood and trusted because of unknown vulnerabilities and advanced risks. Conclusions are based on expected and actual analysis using a prototype approach. Prototyping is a sub-method within the overall System Development Methodology. The present framework "OWASP Internet of Things Top 10, (2014)" addresses the field of Internet of Things vulnerabilities from a risk point of view and is developed from results of polling industry leaders about the threat landscape. This framework has been offered as a security template for manufacturers to build better secure products and system developers to address requirements for M2M ecosystem security, but is focused on the end-to-end threats as identified in Figure 2. We found that a better Internet of Things security framework exposes five key vulnerabilities that threaten the M2M device. Holmlid and Evenson (2007) explain that prototyping explores future reality. By applying distinguish analysis of testing results, we validated the theoretical framework. Then we improved the framework with additional tested enchantments that addresses the threats as depicted in Figure 2. Using analysis, we validated the prototype testing, using the theoretical framework and "literature reviewed" expected results. Actual results from testing were then extracted and used for the final design of the New Security Framework and the discovery next generation threats facing M2M devices. Additionally, the hypothesis that "vulnerabilities are transferred to next technologies" is supported; based on expected and actual testing results. We found that M2M devices do lack secure authentication, session key exchange schemes, and adequate cryptographic storage for small packet transmission.



Device Hardware	Anti-tamper bypass assessments	Enclosure anti-tamper	Trusted execution environment	IC side channel detection
	Boot/OS crypto validation	Enclosure tamper detection	MCU/SOC anti-tamper	Hypervisor separation kernel
	Boot lock	Hardware TPM root	Design review	Penetration testing
OS/Application	No-execute-bit integration	S-SDLC practices	Sandboxing / compartmentalize	Virtualization
	App whitelisting	Cryptography	Updating/patching	Least privilege access control
	DEP & ASLR	Integrity verification	Penetration testing	Authentication
Data	Risk classification	Sandboxing / compartmentalize	Accounting & logging	Multi-level access control
	Least privilege access control	Authenticity verification	Integrity verification	Secure delete
	Encryption	Authentication		
Communications	Symmetric encryption	Asymmetric encryption	Firewall & ingress/ egress whitelisting	Protocol review & test
	Single factor authentication	Dual factor authentication	Session & token control	Access control
	VPN	Firewall	Intrusion detection	S-SDLC practices
Cellular Services	2G encryption & authentication	3G/4G encryption & dual-authentication	Server hardening	Firewalls & IDS
	Firewall & ingress/ egress whitelisting	Web services security	Database security	Vulnerability & penetration testing
	Private gateway	VPN	Anti-DDOS	DNS-Sec
Gateway Services	VPN	Two factor authentication	Bad login controls	Threat management & patching
	Web services security	Firewalls & IDS	Vulnerability & penetration testing	Logging, auditing, alerting
	Server hardening	Database security	DNS-Sec	S-SDLC practices
Enterprise Services	VPN	Two factor authentication	Bad login controls	Threat management & patching
	Web services security	Firewalls & IDS	Vulnerability & penetration testing	Logging, auditing, alerting
	Server hardening	Database security	DNS-Sec	S-SDLC practices

Figure 2. Threat Framework (Horton, 2014)

M2M devices lack these because the devices have low power requirements, insufficient processing ability, and associated resource constraints. The ability and ease of finding keys, viewing clear text copies of data and accessing channels automatically without decrypting data threatens the secure cryptographic storage. The highest threat to

M2M devices is Denial of Service, which is triggered by weak legacy communication protocols and known radio-side vulnerabilities. As stated, the System Development Methodology used for this research is based on building blocks. The blocks were constructed in 3 Phases. Chapter 4 was the Prototyping phase where the primary and secondary analysis took place. The research performed in the Prototyping phase helped build the required “knowledge” of vulnerabilities and allowed for the “discovery” of problems threatening M2M devices, which then directed the development of the final framework shown in Chapter 5.

Chapter 5

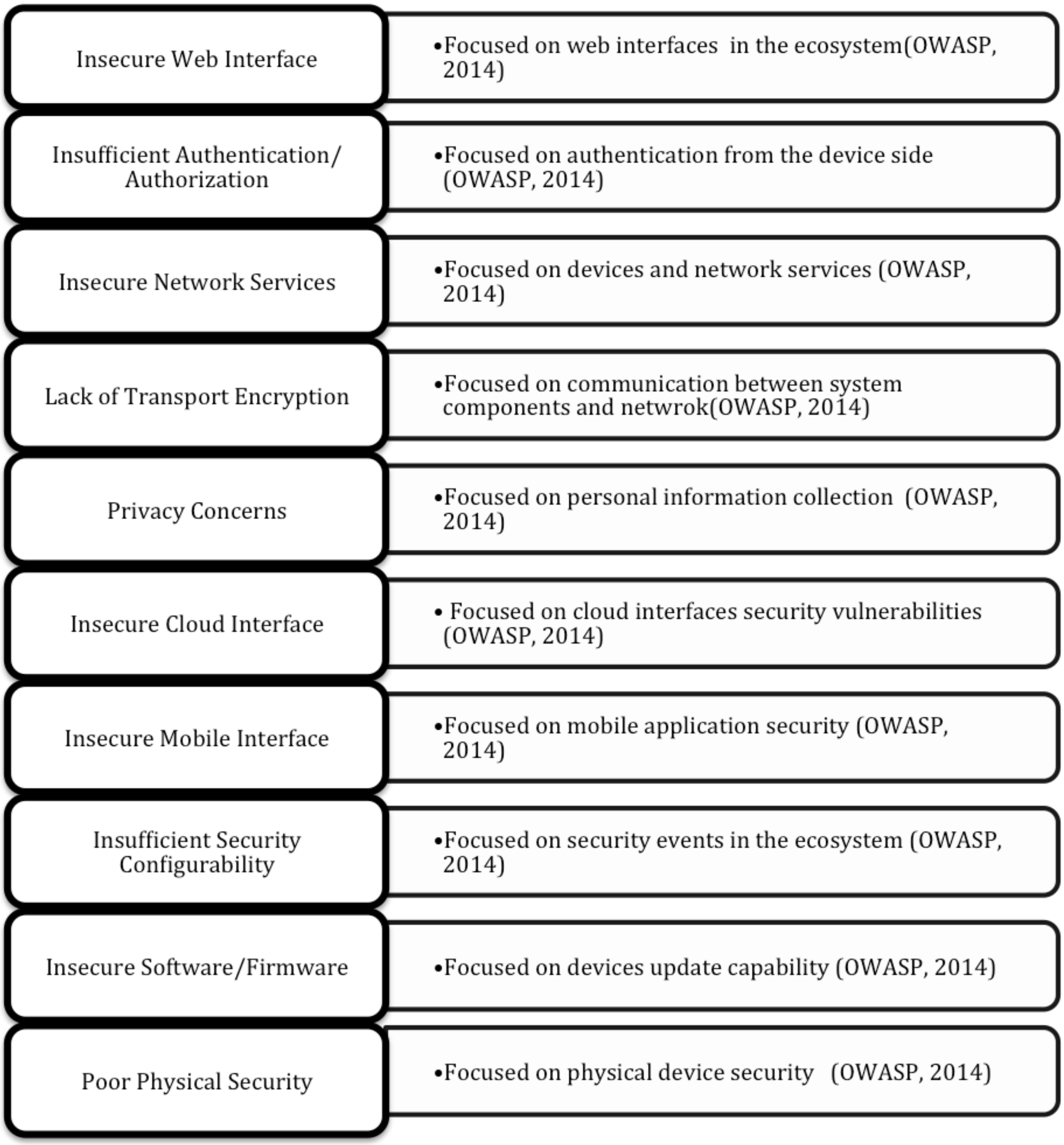
Conclusions, Implications, Recommendations, and Summary

Guided by the systems development research methods, the purpose of this study was to develop a validated and improved security framework for M2M devices that addresses the threat landscape of the Internet of Things. This study included a history of Wireless Systems technology development, a comprehensive review of the literature of the vulnerabilities and threats to these systems and analysis of present security frameworks for IoT. The study also included real-world prototype testing and an analysis approach as a sub-method of the systems development research methodology, this approach identified threats, vulnerabilities and needed requirements for improved security of M2M devices. The approach led to the development of the final security framework that foundationally focuses on the device’s required functions versus actual capabilities of the devices tested and benchmarked within the tested ecosystem.

The literature review and data analysis results enabled the discovery of and provided gained knowledge to draw specific conclusions and directed the development of the improved security framework. The final framework can be replicated and used by M2M device manufactures, IoT systems developers and security architects for guidance in securing the overall IoT ecosystem and the M2M devices. Chapter 5 presents conclusions, implications, and recommendations for future research and concludes with a summary of the research study.

Conclusions

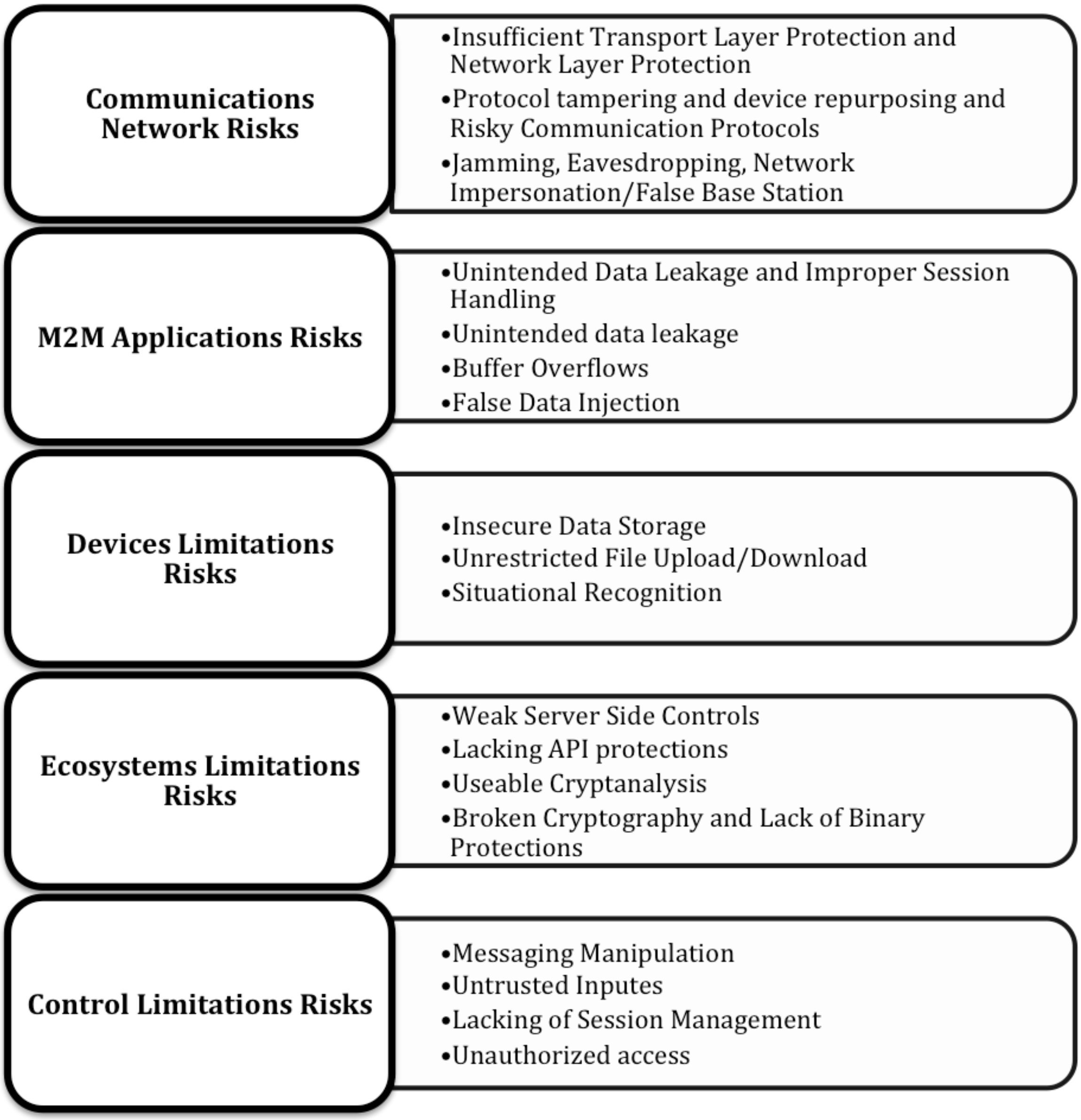
The following conclusions are drawn from the review of the literature and the appropriate results from the systems analysis. Table 2 represents the theoretical model on present risks believed to threaten the IoT ecosystem, followed by a short description of the major categories as described by the OWASP Internet of Things Top 10 Risks (2014).



Insecure Web Interface	•Focused on web interfaces in the ecosystem(OWASP, 2014)
Insufficient Authentication/ Authorization	•Focused on authentication from the device side (OWASP, 2014)
Insecure Network Services	•Focused on devices and network services (OWASP, 2014)
Lack of Transport Encryption	•Focused on communication between system components and netwrok(OWASP, 2014)
Privacy Concerns	•Focused on personal information collection (OWASP, 2014)
Insecure Cloud Interface	• Focused on cloud interfaces security vulnerabilities (OWASP, 2014)
Insecure Mobile Interface	•Focused on mobile application security (OWASP, 2014)
Insufficient Security Configurability	•Focused on security events in the ecosystem (OWASP, 2014)
Insecure Software/Firmware	•Focused on devices update capability (OWASP, 2014)
Poor Physical Security	•Focused on physical device security (OWASP, 2014)

Table 2. IoT Theoretical Top 10 Risks

The above model identifies risks associated across the IoT ecosystems domains. Table 3 below offers an improved model of the Top 5 risks that threaten the M2M devices that will operate within the IoT. This model is functional and is formulated from real-world testing and drawn by content from Chapter 3 and results from Chapter 4.



Communications Network Risks	•Insufficient Transport Layer Protection and Network Layer Protection •Protocol tampering and device repurposing and Risky Communication Protocols •Jamming, Eavesdropping, Network Impersonation/False Base Station
M2M Applications Risks	•Unintended Data Leakage and Improper Session Handling •Unintended data leakage •Buffer Overflows •False Data Injection
Devices Limitations Risks	•Insecure Data Storage •Unrestricted File Upload/Download •Situational Recognition
Ecosystems Limitations Risks	•Weak Server Side Controls •Lacking API protections •Useable Cryptanalysis •Broken Cryptography and Lack of Binary Protections
Control Limitations Risks	•Messaging Manipulation •Untrusted Inputes •Lacking of Session Management •Unauthorized access

Table 3. Top 5 Functional Risks and Vulnerabilities

Table 3 categorizes the five high-level risk domains related to M2M devices and identifies weakness from the M2M device's perspective. This model provides a checklist of concepts that leads to stronger protections for the M2M devices and processes. Highlighted are five key categories where technical vulnerabilities have been found and observed during the prototyping phase.

These categories include:

The "Communications Network Risks" showing that M2M devices are highly vulnerable to legacy wireless network attacks. M2M devices must be capable of gathering information and delivering that information reliably and securely (Bartoli et al., 2011). The present protocol suite exhibits vulnerabilities that hinder performance, and network reliability, these vulnerabilities include the weakness of the transport layer and include, RF jamming and eavesdropping attacks. M2M devices are vulnerable to many of the same "Applications Risks" that threaten Mobile and Smartphone devices. However, the IoT architecture is dependent on many technologies from different domains, this makes the M2M devices less secure because of lacking identification, authentication and authorization for interoperability across capillary networks within the domains. Present technologies do not scale across federated networks and various IoT ecosystems because of lacking standardization efforts and reliable system interfaces (Wu, et al., 2011).

Some M2M devices are designed to perform solo to limited tasks making the devices vulnerable to "Devices Limitations Risks". These risks include low and limited data storage capabilities, low processing ability, limited availability of power resources and little situational recognition. However, these vulnerabilities also exist between devices and gateways in capillary networks where many devices may communicate with each other proving again that lacking interoperability and management capabilities, leads to even greater limitations and risk. The very nature of the physical environment that many M2M devices are deployed within causes a significant threat to the device because harsh conditions lead to worsening resources, new failing points and attack vectors. Plus, protocols for cryptanalysis and API protections in IoT are not strong enough and efficiently tested from the device side (Hue, et al., 2013).

M2M devices lack strong “control” mechanisms and attacks may lead to the malicious takeover of the physical device. These include messaging manipulation from untrusted inputs and unauthorized access from secure session management. Lack of well-defended requirements and standards solutions cause an unsuitable system design and management solution (Foschini, Taleb, Corradi, & Bottazzi, 2011).

The reached goal of this research was to create a better security framework for M2M device security development. Chapters 3 and 4 provide an outline for framework development. The framework illustrated in Figure 3 is created from multiple perspectives included in the system development process. This approach provided a means for developing and applying a security design method for M2M devices by applying the System Development Methodology. For this research, a mollified 3-phase approach was applied. Phase 1 conducted a preliminary analysis and foundation knowledge gathering, Phase 2 conducted the system analysis for developing stronger security requirements recommendations and, Phase 3 developed the new framework. The framework uses “prototyped” observed results supported by results extracted from literature review to identify vulnerabilities and strengthen the device specifications and then provided a formal representation of specifications and requirements in the form of the illustrated frameworks (Figure 2, Table 3, and Figure 3).

The basic building blocks of this framework are the “use cases” as defined in Chapter 2. For this research the “use cases” can be thought of as both the “actor” and “role” in the development process. From the “use cases”, partial knowledge is gained about the role that the M2M device must fulfill. Additional knowledge in gained from calculating the “Business Logic” that must be performed by the M2M Device. The “Business Logic” serves as the “perspective” of the actor. Once the knowledge gained stage is completed and combined, the first “action” is determined and the function of the M2M device was determined. In this framework, the “action” determines the “Functional Classification” of the device role (Godfrey et al., 2015). These steps complete Phase 1 of the System Development Methodology.

In Phase 2 the present frameworks were analyzed, problems found and improved requirements then defined. For this research, the present framework for M2M security that was interpreted and analyzed was the OWASP Internet of Things Top 10 (2014). The test scenarios for diagnosing problems with present technologies were then realized and recommend requirements generated using the prototyping development approach. Prototyping is an experimental process that is suitable for both gaining present systems operational experience and for the discovery of new requirements identification (Vaishnavi & Kuechler, 2015).

Phase 3 includes the design and development of the final framework. The research completed the development and documentation of the system by preparing a framework that contributes to the overall solution of security for next-generation devices.

The framework presented in Figure 3 contributes the missing knowledge needed to fill the gap between the intersection of human interface, wireless, and M2M device risks and security countermeasures (Riahi, et al., 2014). The framework solves the stated problem by developing and focusing on devices "functions and capabilities", this effort (a) adequately considers restrictions and constraints; (b) identifies significant shortfalls; and (c) led to a more thorough and detailed M2M security framework.

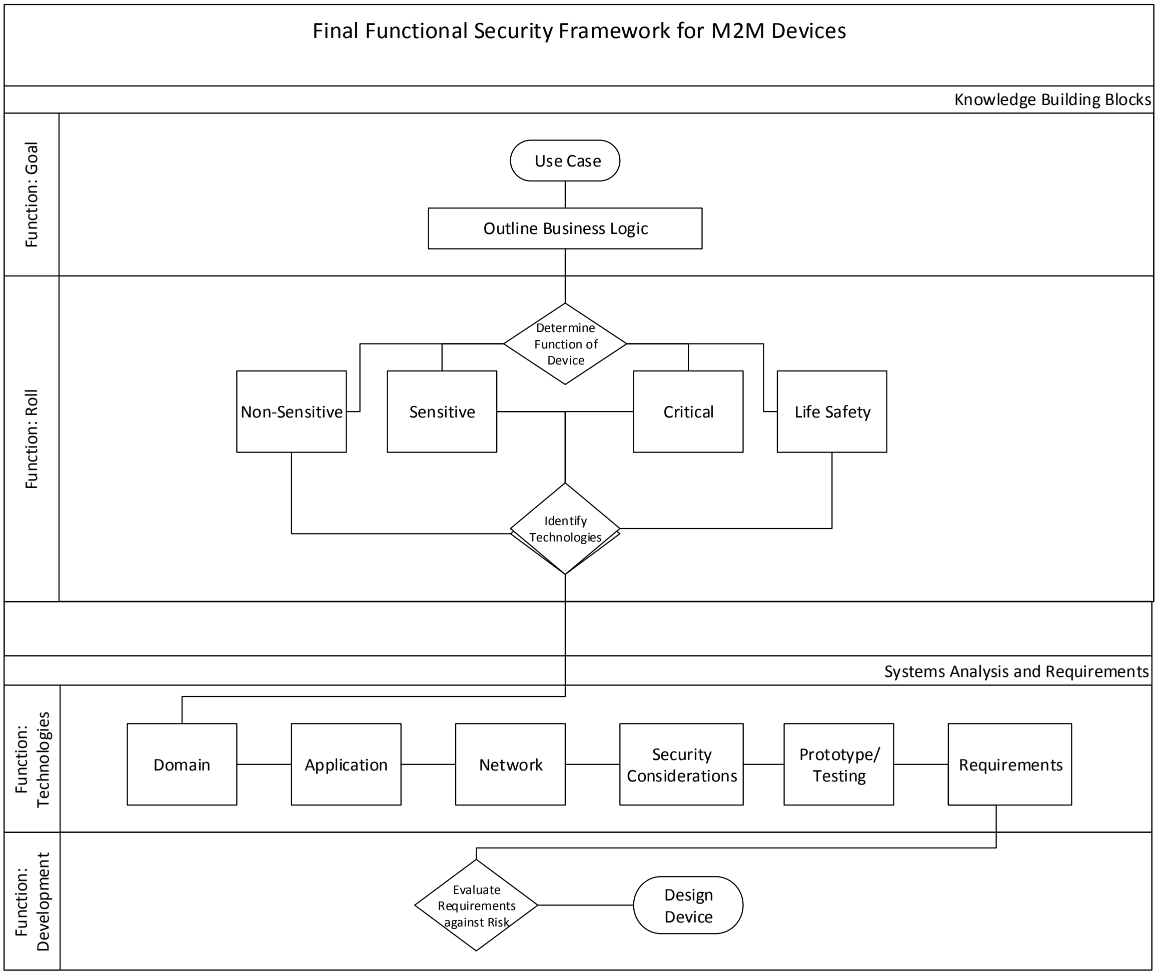


Figure 3. Final Functional Framework

Implications

Figure 3, shown above, illustrates the improved framework. This study helped identify the present vulnerabilities in M2M devices and assist in the development of new security requirements for the IoT ecosystem. The results of the study and the review of the literature guided the design of the new security framework for M2M devices. This new framework provides needed direction and exposes threats that must be taken into consideration for secure device development and ecosystem implementation.

This study also contributed to the body of knowledge of systems design by applying the development research method to address the research problem of proper functional security in next generation technologies. Although the research was focused on the method of development and the creation of a security framework, an additional goal was reached of discovering, restructuring and presenting the M2M device top five risks and threats framework.

Recommendations

Future research could be conducted based on the results outlined in Chapter 4 of this study, many of the tests proved the feasibility of attack. Results found that many of the vulnerabilities are caused by past technology vulnerabilities and known threats. Present communications protocols that will be implemented in future IoT ecosystems are flawed with both security and privacy concerns. The possible development and redesigned or even better, new communications protocols, will prove to have value to the industry. This study also focused on the development of a new framework or risk and research could be conducted on specific vulnerabilities and development of countermeasures address exposed risk. Such research would provide deeper discovery and furnish needed requirements for device hardware and software design.

Summary

At present, there are many security problems with IoT ecosystems and, in particular, the M2M devices that are designed for these systems (Lai et al., 2012). This research fills the gap that exists between the past approaches for security framework development and understanding, by identifying how these new technologies must differ from past and present technologies. The past techniques for developing security requirements do not adequately consider the use of new technologies, which weakens countermeasure implementations. Developed by this research is a security framework designed for requirements development. This research provides a framework design method for identifying next-generation security concerns and processes for comparing, contrasting and evaluating non-human device security protections used in the IoT ecosystem.

Appendix A: Real-World Test

Appendix A

Real-World Test

The below tables outline the test that were executed to verify device compliance, function, and capability as required by the defined Use Cases. The results then were used to build the Risk, Vulnerability and Functional Security Design Framework.

Test 1	Baseline vulnerability scans are directed towards Weak Server Side Controls.
What Is Tested and Analyzed?	Literature explains that vulnerability scans identify open ports. Also reported is the Internet Protocol (IP) addresses, the device operating system, software and services that are running on the device.
Overview	The vulnerability scanner used for testing was a software-based scanner.
Issue	Devices are vulnerable to application failures and open communication ports that lead to DoS attacks. Scanning for known vulnerabilities allowed for the discovery of access opportunities.
Description	We executed scanning to find vulnerable access points on various devices running on an isolated network.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, application security, application environment and security controls, encryption concepts, capabilities of information systems, resource protection, incident response, patch and vulnerability management, disaster recovery process, and internal security (CISSP, 2014).
Affected Stakeholders	Application service providers, manufacturers of devices, the M2M device/gateway, management providers, M2M service providers, network operators, and user/consumers.
Architecture Impact	We found that this test impacts the device communication protocols within the service-layer and the storage of encryption keys for authentication. Also found was that the wireless network and configuration settings are impacted if vulnerabilities are discovered because the privacy and security of messaging is at risk.
Action	For the purpose of this research the vulnerability scanner was considered the server. As the server, the scanner was used to send and scan the device. In particular to this research the events of interest were host discovery, port scanning, OS detection and service discovery.

Expected Result	We expected that the standard behavior for each of the events would be capable of being identified on each device.
Actual Result	We found that normal various TCP and UDP ports were opened on the M2M type devices as well as the other network connected devices.
Knowledge Gained	We gained knowledge that because the devices are prototyped the ports can be turned off or on. However, for the device to communicate applications such as for SMS messaging must remain opened and did leave ports vulnerable to various known SMS attacks. However, being able to scan a device is not a bad thing. Discovering vulnerabilities or configuration errors results in understanding where intrusions can occur and leads to the development of better countermeasures.
New Requirement.	A better-designed device allows for the scanning and review of process. Quarantining the devices applications and revoking the permissions for the applications closes the exposed ports. Only ports that are required for data transfer are exposed. Over the cellular network, updating the device communications module to 3G and placing a 3G or above smart card into the devices increases security and strengthens protection against server side control attacks.

Service-Layer Keys

Test 2	We directed Test 2 towards insufficient transport layer protection and improper session handling.
What Is Tested and Analyzed?	We tested the availability of long-term service-layer keys and the storage of keys on the devices as called for by (oneM2M Partners, 2014).
Overview	The services layer consists of all the services that the manufacturer makes available on the devices or is preinstalled by device peripherals. The long-term service-layer consists of keys that may be discovered while they are stored on the devices. If discovered, these keys may be copied and used against the device or gateway during other attacks (Ukil, Bandyopadhyay, Bhattacharyya, & Pal, 2013).
Issue	The services layer is highly vulnerable to attack because the data within the layers provides the business functionality that allows the devices' supported communications and messaging to and from the gateway. If copied, the long-term service-layer keys may be used to impersonate M2M devices to the gateways or vice versa (Latvakoski et al., 2009).
Description	When "long-term service-layer keys are stored within the M2M device or M2M gateways", they may be discovered during scanning by unauthorized entities (oneM2M, 2014). For example, transport layer security (TLS) uses a cryptographic system with two keys to encrypt data (Hersent, Boswarthick, & Elloumi, 2011). Once the keys are discovered, they can be used for illegitimate purposes, such as false authorization and authentication. There are various methods for discovering open and available keys. Hardware probing methods include

	the monitoring of internal processes or simply the reading of memory contents. According to Lu, O’Neill, and McCanny (2010), DPA is a widely studied side-channel attack however this attack is outside the scope of this research. They recommend that Random Delay Insertion (RDI) be deployed as a countermeasure technique. However, M2M devices run weak cryptographic processes and do not have the resources for increased countermeasures that will reduce the risk of DPA attacks.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, application security, application environment and security controls, encryption concepts, capabilities of information systems, resource protection, incident response, patch and vulnerability management, disaster recovery process, and internal security (CISSP, 2014)
Affected Stakeholders	Application service providers, manufacturer of devices, M2M device/gateway, management providers, M2M service providers, network operators, and user/consumer
Architecture Impact	We found that this test impacts the device communication protocols within the service-layer and the storage of encryption keys for authentication. The wireless network and configuration settings are impacted if vulnerabilities are found, because the privacy and security of messaging is at risk.
Action	1. Use the V scan to discover IP address. 2. Ping open ports, this test’s the security of the communication protocols 3. Establish a wireless connection with which to send input data and functions toward the device by using ARPping and verify working device. 4. Scan again with Nmap to see the device using Wi-Fi Internet connection 5. Observe the tool log for data computation bugs. 6. Document the device performance and responses from the device. 7. Report any failures or open known vulnerabilities that effect policy and enforcement.
Expected Result	This is a deep analysis of the ports open on a running device. The test exposes the device to authorization vulnerabilities. We expected that the tools would report no known security vulnerabilities against the open ports. These devices run limited applications such as GPS location tracking, which only requests location data when needed. There are inherent limitations to resources on these devices; data should be stored read-only and limited in storage time.
Actual Result	We found that because these devices are prototyped devices and are built using the Raspberry Pi version 2. These are unsecure devices. Testing layer 2 with port pinging and scanning showed open ports and

	device information including the TCP IP Address and MAC.
Knowledge Gained	The ARP request determined that the host is alive, also the MAC address of the devices is returned. However the RPi are low processing devices and repeated testing freezes the device and takes it off network. Also ICMP echo request using Scapy flood the device at whatever port is specified in the ping command. This flood message stresses the CPU to MAX usage.
New Requirement.	Detailed in Chapter 4.

Stored Management and Control Frames

Test 3	Test 3 was directed towards Network Layer Protection and Insecure Data Storage.
What Is Tested and Analyzed?	The importance of deleting stored management and control frames on Layer 2 of M2M devices, as described by Hersent, Boswarthick, and Elloumi (2011)
Overview	Management and control frames are directly related to service-layer keys and focused on the Layer 3 frames because M2M data will service three requirements: massive data analysis, real-time data analysis, and deep data analysis (Kitagami, Yamamoto, Koizumi, & Suganuma, 2013). In many cases, open source software such as SQLite and R will be used to pass data in small- and medium-sized M2M service systems. Information stored in the devices can be detected and used by an attacker to compromise these systems. The device is also at risk for being spoofed or turned into an attacking device, a fake access point, or a fake wireless bridge. Management and control frames must be protected to protect the protocol stack (Lin, 2008).
Issue	Literature explains that unprotected protocol stacks may lead to denial of service (DoS), Man-in-the-middle (MinM), and similar attacks. This vulnerability can prevent the operation of the overall M2M service.
Description	Test the possibility of locating service-layer keys within the protocol stack and the device. We tried to discover if these keys could be reused, deleted, or changed. All systems have management commands that can be used and attack vectors that can perpetrate key-storage functions of M2M devices and M2M gateways.
Impacted Use Cases	All
Affected Security Domain	Authentication, application security, and network security
Affected Stakeholders	Application service providers, manufacturer of devices and gateways, service providers, network operator, and user/consumer
Architecture	If long-term keys that are transferred over the service-layer experience

Impact	compromise of storage, this impacts the constraints on the device.
Action	1. Used ping tools to ping and to send ICMP echo request 2. This should indicate that the host corresponding to the address is alive 3. Generate the test inputs. These are random messages with test strings of characters. 4. Send various messages to the device over the air using both the GSM/GPRS and Wi-Fi networks 5. Test the Protocols IP, ICMP, ARP, & RIP and uses Routers as its device flood test the ICMP against the device not router 6. Using hping3 to perform layer 3 discovery 7. Use Scapy to discover layer 4 User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) transport protocols 8. Document the error logs from the control center and "real" errors that relate to device management and architectural controls.
Expected Result	We expected to discover that the authentication frames and open data transfer may be used for Layer 3 attacks such as DoS and Man-in-the-Middle attacks. These tests should find a weakness in the various protocols. There are known vulnerabilities in the GSM stack. It is expected that sending different configuration messages to the device on each port will expose the GSM vulnerabilities.
Actual Result	We found that running hping3 does successfully scan layer 3 and layer 4 and reports the device IP. Using Scapy to perform layer 4 discovery reported that the host is alive. However, only open and not closed TCP ports reported an ACK to messaging.
Knowledge Gained	The Scapy test fails/locks all 3 RPi devices when a particular port is addressed in the execution string.
New Requirement.	Detailed in Chapter 4.

Software-to-Hardware Signal Interface

Test 4	We directed Test 4 towards unintended data leakage and improper session handling.
What Is Tested and Analyzed?	M2M device software-to-hardware signal interface (Bernardi, Merseguer, & Petriu, 2013).
Overview	Many M2M devices run a small version of Linux as an operating system. Components such as cameras and Wi-Fi nodes are run and accessed through files that are usually located in the `/dev` directory of the OS. These files communicate directly with the kernel driver that is in current communication with the component hardware. All

	components must communicate transparently, regardless of their hardware and software (Bernardi, Merseguer, & Petriu, 2013).
Issue	Literature review explains that these files are easy to expose and, once found, can lead to vulnerability (Igure, Laughter, & Williams, 2006).
Description	We note that M2M devices' use open standards that are tested for transparency. The use of open standard makes the device vulnerable to attackers. Because these devices are usually built from commercial off-the-shelf (COTS) hardware and software, attackers seek knowledge to expose open vulnerabilities so that they can disable the fail-safe mechanisms (Igure, Laughter, & Williams, 2006).
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, application security, disaster recovery process, and internal security (CISSP, 2014)
Affected Stakeholders	Service provider, manufacturer, device management system, network operator, and user/consumer
Architecture Impact	The critical infrastructure includes Supervisory Control and Data Acquisition (SCADA) networks like natural disaster early warning systems and crime prevention cameras. A range of vulnerabilities threatens them: from equipment failures to terrorist attacks (Igure, Laughter, & Williams, 2006).
Action	1. First discover the OS using Nmap 2. Test input, output, and measurement data considerations. 3. Use p0f to analyze a Wireshark capture file. 4. Identify input devices with ranges and resolutions of values. 5. Identify output devices with ranges and resolutions of values. 6. Define the full range of input disturbances (unexpected system inputs). 7. Send messages Nmap to the device input ports to discover Service fingerprinting. 8. Observe possible output disturbances that occur when unexpected system inputs are received. 9. Review and analyze device performance. 10. Read the error log from the control center and document errors.
Expected Result	We expected that the OS is reported correctly and all device port that are open and closed are disclosed. As environmental conditions change, the RF signal, in or out data, and messaging should all continue to match a specified value as described by the standards.
Actual Result	We found that the tests run successfully over Wi-Fi and GPRS showed the required and needed information for ensure a successful penetration test.
Knowledge Gained	We gain the knowledge that from the reported IP addresses, active machines, and open ports are identified from the target devices. The services running on specific ports do ensure successfully routing. Using

	the open source threat assessment tool the complexity and severity of a single point of failure of the device can be identified as explained by Igure, Laughter, and Williams, (2006).
New Requirement.	Detailed in Chapter 4.

Interface Attack

Test 5	We directed Test 5 towards the security decisions via untrusted inputs and insecure data storage.
What Is Tested and Analyzed?	We looked for the discovery of human user interface attack and tampering resistance notification as described by Hagar (2013).
Overview	Literature explains that the M2M systems are designed to function without human interaction. However, M2M message traffic, data transfer, and content are influenced by human-based traffic, such as location data, billing data, and personalized content. The M2M device must also be monitored, repaired, and managed by humans (Cha, Shah, Schmidt, Leicher, & Meyerstein, 2009).
Issue	We understood that M2M devices may be targets of tampering, repurposing, or modification. There is a need for failure indicators, such as on/off settings, hardware status, and network control or alarms. These false indicators will cause an operator to take action on this information. If such an attack takes place, an attacker can then perform remote hacking on management or maintenance interfaces and fully compromise the device and platform.
Description	Test the ability to discover tampering and flaws in the monitoring system that could lead to vulnerabilities in the device-user interaction.
Impacted Use Cases	All
Affected Security Domain	Access control, application security, and device security and control platform
Affected Stakeholders	Application service provider, manufacturer, service provider, system administrator, network operator, and user/consumer
Architecture Impact	M2M service infrastructure and device
Action	1. Use the Control Center to apply inputs and verify that the devices is connected correctly. 2. Make error messages that inform the human user of an alarm. 3. Identify hosts that are discovered by UDP probes 4. This sets up a place to insert overflow input to the buffers when with false messages by sending messages from the control center to the device.

	5. Define the inputs that create these outputs. 6. Test these input/output combinations. 7. Determine what the outputs are and then attempt to force invalid human interaction.
Expected Result	We expected that the UDP port that is open will be discovered.
Actual Result	We found that the port reported open is 2165. Running amap identifies that only port 22 is open until closed.
Knowledge Gained	We gained knowledge that it is very easy to identify hosts using discovery against UDP and probing an isolated task. Hping3 uses ICMP host unreachable responses to identify live hosts with UDP requests. An additional scan using Nmap report all open ports. This discovery allows for fingerprinting of services. Testing discover that port 22 (SSH) is opened on the device but the other ports are closed. SSH is opened at install of the OS and not closed at reboot.
New Requirement.	Detailed in Chapter 4.

Unrestricted File Upload/Download

Test 6	We directed Test 6 towards improper session handling and unrestricted file upload/download.
What Is Tested and Analyzed?	Unrestricted file upload to M2M infrastructure from the device equipment (Flick & Morehouse, 2010; Skianis, 2013).
Overview	Literature explains that the vulnerability affects databases, operating systems, and applications that are developed for internal use and in M2M devices that are connected to the Internet with unrestricted access. These platforms lack restrictions on the size or number of uploaded files. Files that are too large will consume resources and freeze the device operations (Flick & Morehouse, 2010). When open access to SSH of FTP servers and found device data storage is at risk.
Issue	We understand that when an attacker uploads or transfers files of dangerous types using an automated processed within the M2M ecosystem via open FTP ports this allows device compromise.
Description	We explains that attacker may use a compromised device to upload or transfer dangerous type files that can be used to hurt the ecosystem.
Impacted Use Cases	All
Affected Security Domain	Access control, communication channels, network attacks, application security, application environment and security controls, encryption concepts, and capabilities of information systems (CISSP, 2014)
Affected	Application service provider, manufacturer of device, network service

Stakeholders	provider, system administrator, network operator, and user/consumer
Architecture Impact	M2M service infrastructure
Action	1. Validate vulnerabilities using HTTP interaction 2. Create an "input file" to send and retrieve from the device. 3. Observe the network with scanner tools like "Wireshark". 4. Send data file from control server to the device. 5. Validate vulnerabilities with HTTP interaction 6. Review and closely analyze the results, looking for obvious crashes.
Expected Result	We expected that the FTP server will block the "input file".
Actual Result	Described in Chapter 4.
Knowledge Gained	Detailed in Chapter 4.
New Requirement.	Offered in Chapter. 4 & 5.

Discovery of Sensitive Data

Test 7	We directed Test 7 towards broken cryptography and lack of binary protections.
What Is Tested and Analyzed?	The "discovery of sensitive data in M2M devices or M2M gateways" according to (Pandey, Choi, Kim, & Hong, 2011; oneM2M, 2014).
Overview	M2M devices, such as sensors, collect data that is sensitive, including toxic levels of poison, temperature of machinery, and personal consumer data. The execution of sensitive functions and the storage and transfer of this information must be protected.
Issue	According to Pandey, Choi, Kim, and Hong (2011), M2M has important characteristics that other ecosystems do not. For example, M2M offers sleeping devices, low power devices, weak signal networks, and low device intelligence. Automated service discovery and an environment for the automatic execution of software and storage of sensitive data leads to a higher level of compromise potential (Cha et al., 2009).
Description	Literature explains that the level of encryption on all communication between the device and server is vulnerable by attacking the execution of sensitive functions within the M2M device. Verify the exposed sensitive data and ensure that protection is in place to prevent unauthorized entities from using this data for illegitimate purposes.
Impacted Use Cases	All
Affected	Integrity, confidentiality, availability, access control, communication

Security Domain	channels, application security, application environment and security controls, encryption concepts, capabilities of information systems, resource protection, incident response, and internal security (CISSP, 2014)
Affected Stakeholders	Application service provider, manufacturer of devices, network operator, and user/consumer
Architecture Impact	The device storage capability for sensitive data and the functions used by the device to send this data to nodes and gateways
Action	1. Examine target device sensitive information using device forensic tools such as the Oxygen Forensic Tool Kit. 2. Use the tools to communicate with the device remotely. 3. Perform intensive caches analysis. 4. Browse the cache and retrieve all information. 5. From the network side, place a sniffer into the communication path between the device and server. 6. Use open source tools to explore the SSL, SSH, and SCP type protocol. These tools detect and sniff information from the network. 7. Review the targeted application-received information from the server, such as dynamic updates, applets, and scripts. 8. Verify that sensitive information is encrypted and protected.
Expected Result	We expected to find that it is possible to capture sensitive information that is outbound and sent to the device. However, the data itself was be protected by the protocol security policy.
Actual Result	Detailed in Chapter 4.
Knowledge Gained	Explained in Chapter 4.
New Requirement.	Offered in Chapter 4.

Eavesdropping

Test 8	We directed test 8 towards poor authorization and authentication and broken cryptography.
What Is Tested?	Eavesdropping on M2M device service-communications protocols that connect the device to the gateway (Kylanpaa, Rantala, Merilinna, & Nieminen, 2013)
Overview	According Ren, Yu, Ma, and Ren (2013), all real-time, wireless communication-oriented overlay networks, like M2M based capillary systems, might operate within a registration based versus location lookup quire architecture. This architecture has special security requirements because vulnerabilities may result in a breach of security

	from real-time eavesdropping on messaging and data transfer.
Issue	Literature explains that eavesdropping of cryptographic resources discloses identities and exposes knowledge of sensitive information.
Description	We examined the feasibility of “eavesdropping on M2M Service Layer” (oneM2M, 2014). Expose messages between components and the available cryptographic resources must protect confidential or private information.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, and application security (CISSP, 2014)
Affected Stakeholders	M2M service provider, devices manufacturer M2M device/gateway management entities, network operator, and user/consumer
Architecture Impact	Radio network and device controls
Action	1. Set up open source software (OpenBTS and FreeSWITCH) and the Ettus Transceiver to route voice and SMS traffic through a private GSM network. 2. Hook up the Raspberry Pi to a radio interface for device testing. 3. The Raspberry Pi runs: OpenBTS: GSM mobile phone standards network FreeSWITCH: call routing tool Python: for programming scripts 4. Force device to attach to the GSM network. This takes advantage of weaknesses in the GSM protocol stack. 5. Eavesdrop on the following devices: M2M devices to the M2M gateway M2M gateway to M2M devices 6. Test the protection in communications protocols by sending authentication messages. 7. Take advantage of weaknesses in the GSM protocol by performing unexpected actions, such as sending false identity data. 8. Manipulate messaging protocols. 9. Look at logs to see if Detailed in Chapter 4 can read sensitive information or modify message content.
Expected Result	We expected that the network and device tasks would be separated from each other by various systems. There are known weakness in GSM that can be exploited. However, M2M devices and applications should have a separation of functions that prevent tasks from being hijacked, which would make them less vulnerable to eavesdropping as explained by 3GPP (2002).

Actual Result	We found that packets can be downloaded leading to the impersonation of a user this may be caused by the intruder sends signaling and/or user data to the network, as a fake network.
Knowledge Gained	We gained the understanding that the intruder may be able to eavesdrop on signaling and data connections associated with other users. However, this is possible over various radio technologies.
New Requirement.	Detailed in Chapter 4.

Communication Protocols (Fuzzing)

Test 9	We directed Test 9 towards transport layer protection, client side injection and security decisions via untrusted inputs.
What Is Tested and Analyzed?	Alteration of M2M communication protocols and messaging between devices and gateway, as described by Sheng, Yang, Yu, Vasilakos, McCann, and Leung (2013).
Overview	The M2M ecosystem has complex deployment characteristics. New approaches and standards are required for the system to meet security requirements due to the resource constraints of present networks. Any alteration or manipulation of the messaging protocol may lead to the readability of sensitive information or modification of message content (Chen & Ma, 2014).
Issue	According to Chen and Ma (2014), the different communication protocols presently in use within the wireless and wire-lined system will cause a protocol security gap. This gap may potentially lead to a threat against the M2M ecosystem. Protocol device attacks include Man-in-the-Middle attacks and denial-of-service (DoS) attacks, exploitation of network services weaknesses, and over-the-air management attacks.
Description	Explore the transaction layer and its protocols that handle message exchange between the devices and gateway by altering the M2M device and forcing the device off the core network.
Impacted Use Cases	All
Affected Security Domain	Availability, access control, communication channels, and network attacks (CISSP, 2014).
Affected Stakeholders	Management entities, service provider, network operator, and user/consumer
Architecture Impact	Radio network and device controls.
Action	1. Set up open source software (OpenBTS and FreeSWITCH) and the Ettus Transceiver to route voice and SMS traffic through a private GSM network. 2. Hook up the Raspberry Pi to a radio interface for device

	testing 3. The Raspberry Pi runs: OpenBTS: GSM mobile phone standards network FreeSWITCH: call routing tool Python: for programming scripts 4. Force device to attach to GSM network. This takes advantage of weaknesses in the GSM protocol stack. 5. Eavesdrop on the following: The M2M devices to the M2M gateway The M2M gateway to M2M devices 6. Test the protections in the communication protocols by sending authentication messages. 7. Take advantage of weaknesses in the GSM protocol by performing unexpected actions such as sending false identity. 8. Manipulate messaging protocols to fake a known network to the device. 9. Look at logs to see if Detailed in Chapter 4 can read sensitive information or modify message contents.
Expected Result	As we expected the device to gateway communications defaulted to the cellular 3G protocols and authentication method. This will protect against known 2G vulnerabilities. However, if the device can be forced back to 2G, then weakness-allowing manipulation of the interaction will be seen.
Actual Result	We found that the Scapy tool is able to scan the device for open ports and determined that potential target services like GPS/Location and control center communication ports are open.
Knowledge Gained	Knowledge gained showed that tested devices with GSM/GPRS SIM do register with the OpenBTS tool and we verified that the devices are not resistant to GSM fuzzing. Sending test SMS message to device does not fail the communication module however a full suite of created test cases has not been applied for full fuzz testing.
New Requirement.	Detailed in Chapter 4.

Replay Messaging (false Base Station)

Test 10	We directed Test 10 towards Improper session handling and transport layer protection
What Is Tested and Analyzed?	Replay messaging between devices and gateways as described by (Latvakoski et al., 2014)
Overview	The user’s data, signaling data, and control data require security measures for correct transmission. This data passes between the devices

	and gateways through the physical layer or protocol layers. If the protocol is compromised, all services are affected.
Issue	Literature contributes that the replay attack occurs when an attacker can copy messages between devices and gateways and use them to defeat authentication. Replay attacks damage transaction information by allowing for the modification, insertion, or deletion of legitimate user data or signaling message structures.
Description	We tested the feasibility of capturing messages. In order to discover if there is a possibility of any tracking mechanisms that allow for the identity of deceptive or fraudulent message content.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, and network attacks (CISSP, 2014)
Affected Stakeholders	Service provider and network operator
Architecture Impact	Radio network and device controls
Action	1. Set up open source software (OpenBTS and FreeSWITCH) and the Ettus Transceiver to route voice and SMS traffic through a private GSM network. 2. Hook up the Raspberry Pi to a radio interface for device testing. 3. The Raspberry Pi runs: OpenBTS: GSM mobile phone standards network FreeSWITCH: call routing tool Python: for programming scripts 4. Force device to attach to the GSM network. This takes advantage of weaknesses in the GSM protocol stack. 5. Eavesdrop on the following devices: M2M devices to the M2M gateway M2M gateway to M2M devices 6. Test the protection in communications protocols by sending authentication messages. 7. Take advantage of weaknesses in the GSM protocol by performing unexpected actions, such as send false identity data. 8. Manipulate messaging protocols. 9. Look at logs to see if Detailed in Chapter 4 can read sensitive information or modify message contents. 10. Send back captured messages using Python script messages to change data content. 11. Check the receipt of the repetition messages. 12. Log the capability of attack.

Expected Result	As we expected the attack may exploit the lack of protection in the communications service layers and that this lack will lead to replay or playback of network data transmissions thus leading to a successful false bases attack over GSM.
Actual Result	The devices roam and register with false base station.
Knowledge Gained	We gained the knowledge that a compromised BTS can act as a repeater against the M2M devices. In repeater mode, the attack platform functions as a relay for requests to the legitimate network. These systems are located in between the network and the target user. Good service requests and/or paging messages for the target device are modify or ignore by the attacking system. In the security architecture of 3G there is no prevention mechanism against false BTS relaying messages the device is only seeking authentication.
New Requirement.	Detailed in Chapter 4.

Unauthorized or Corrupted Applications

Test 11	We directed Test 11 towards security decisions via untrusted inputs and lack of binary protections.
What Is Tested and Analyzed?	"Unauthorized or corrupted applications or software in M2M devices" as explained by (Ho, Jacobs, Meissner, Meyer, Monjas, & Segura, 2013)
Overview	Software does not properly anticipate or handle exceptional conditions in a manner that is required to provide a safe exchange of information. It is important that applications authenticate successfully to establish secure channels (Jeon, Lee, Park, & Jeong, 2013).
Issue	Literature explains that it is unknown how to detect replay attacks or mutual cluster authentication within a federated M2M ecosystem. Unauthorized devices may run software that authorizes functions to create vulnerabilities that impersonate the network and device management platform (Liu, 2012).
Description	Furthermore, these test showed that the feasibility of an attacker to impersonate the network and device management platform by temporary failing the single input and output communication protocol. If successful, the attacker could fail the device by reporting fake device consumption, breaching privacy, or reporting confidential information, which would lead to the attacker's control of remote management functions and systems.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, application security, application environment and security controls, and internal security (CISSP, 2014)

Affected Stakeholders	M2M application service provider, manufacturer of M2M devices and/or M2M gateways, M2M device/gateway management entities, M2M service provider, and user/consumer
Architecture Impact	M2M service provider’s domain, M2M devices, and M2M gateways
Action	1. Set up open source software (OpenBTS and FreeSWITCH) and the Ettus Transceiver to route voice and SMS traffic through a private GSM network. 2. Hook up the Raspberry Pi to a radio interface for device testing. 3. The Raspberry Pi runs: OpenBTS: GSM mobile phone standards network FreeSWITCH: call routing tool Python: for programming scripts 4. Force device to attach to GSM network. This takes advantage of weaknesses in the GSM protocol stack. 5. Eavesdrop on the following devices: M2M devices to the M2M gateway M2M gateway to M2M devices 6. Test the protection in communications protocols by sending authentication messages. 7. Take advantage of weaknesses in the GSM protocol by performing unexpected actions, such as sending false identity data. 8. Manipulate messaging protocols. 9. Manipulate an authentication protocol. 10. Seek older protocol vulnerabilities. 11. Look at logs to see if Detailed in Chapter 4 can read sensitive information or modify message contents. 12. Send back captured messages using Python script messages to change data content. 13. Manipulate client-server interactions. 14. Check the receipt of the repetition messages. 15. Log the capability of attack.
Expected Result	We expected to find that the communications application running on the device utilized a secure protocol that does not have weaknesses, which allows for the manipulation of interaction between the device and the end gateway.
Actual Result	We found and proved that the packet switched protocols do have known weakness that will disturb M2M device in this environment. This could mean that in 2G/3G legacy cellular communications systems that the relationship between the HLR, VLR, AC and other databases and networks elements might provide increased vulnerability when dealing with a mix of domains and crossover protocols (Huber and Huber

	2002).
Knowledge Gained	Knowledge gained included verification that the primary radio/antenna component on the GSM network is the Base Station Transceiver (BTS) allows for IMSI catching using a False Base Station Transceiver. The devices see this base station as a legitimate carrier’s network. This is because the BTS transmits at a higher power level then the legitimate BTS. Per specification and design, all Mobile Devices seek the “best” power received transmission, once found the device transmits its identifier data, such as IMSI.
New Requirement.	Detailed in Chapter 4.

System Interdependencies

Test 12	We directed Test 12 towards improper session handling insecure data storage and unintended data leakage.
What Is Tested and Analyzed?	M2M system interdependencies threats and cascading impacts (Macaulay & Singer, 2011)
Overview	Proper message content delivery affects the overall ecosystem environment and the information that a message contains includes all interdependencies. Interdependencies are defined as types of intelligence that are actionable, such as asset ownership, location, and device role (Bianchi, 2014).
Issue	Literature explains that the underlying systems and resources may impose many forms of vulnerabilities on interdependency that directly relate to failures of the ecosystems’ critical infrastructures (oneM2M, 2013)
Description	We gathered and tested the effects of external interdependencies on the M2M endpoints from the perspective of the device to the M2M gateway. For this research, an interdependencies threat is anything that can be used as an attack vector and shares resources.
Impacted Use Cases	All use cases
Affected Security Domain	Application environment and security controls, resource protection, incident response, patch and vulnerability management, and disaster recovery process (CISSP, 2014)
Affected Stakeholders	Device/gateway management entities and M2M service provider
Architecture Impact	Principle of least privilege, internal system, and control center
Action	1. Load a malicious resource into the device. 2. Use a bootstrap program to add a program, like J2EE applications, to simulate malware.

	3. Verify the application is running and that other component based applications, such as GPS, are running correctly. 4. Modify the path variable to read and write data to the same local file store as the component based applications. 5. Include malicious resources that can be transmitted from the gateway and sent to the device. 6. Observe whether commands can unwittingly be executed on the device by a remote message or sent application.
Expected Result	As we expected that the device's operating system and that enforced the principle of least privilege all system dependencies are secured to the same or higher level of assurance as other programs.
Actual Result	Nothing found.
Knowledge Gained	We gained the knowledge that no privilege user information is on the device to manipulate the radio or device parameter outside of acceptable use. However the screen only shows MEID, IMEI, SIM ID and VERSION information but this should be acceptable for simple management.
New Requirement.	Detailed in Chapter 4.

Context Awareness

Test 13	We directed Test 13 towards lack of binary protections
What Is Tested and Analyzed?	M2M device application security with context awareness and the ability of the device to store relevant information, including the owner of the device, network authentication keys, and other specific access policies as described by (Cam-Winget & Didier, 2014)
Overview	Context-awareness aims to break applications (Hagar, 2013) such as device authentication and key generation.
Issue	Literature explains that there is a lack of context awareness with the M2M ecosystems and how they will function. Most M2M devices are being deployed as a "one size fits all" box that will function as a tool for other devices, gateways, and applications. This increases the risks associated with security.
Description	Apply deep packet inspection related to what has been completed on mobile smart devices. As more and more applications are developed in the M2M ecosystem, there is a need to consider the other attack possibilities, such as software test attacks, device-to-device attacks, and cognitive machines attacks. These attacks take advantage of weaknesses in the client-server relationship and various protocols that allow for device communications.
Impacted Use Cases	All use cases

Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, and application security (CISSP, 2014)
Affected Stakeholders	Application service provider
Architecture Impact	M2M service provider's domain, M2M devices, and M2M gateways
Action	1. Gather and list the functions and links between applications and communications protocols. 2. List and map all application tasks and functions. 3. Identify use scenario as based on Use Case requirements. 4. Note the different user types as well as common and uncommon usages of running applications. 5. Define valid data or input for application and data transmitted. 6. Define any invalid or valid data as well as input options for each application. 7. Build a matrix of invalid data inputs and outputs, including specific data points or values that are common and different. 8. Determine what inputs and outputs expose failures. 9. Test the end-to-end functional tests of the application. 10. Test the app against expected functionality. 11. Compare the results against standard requirements. 12. Identify risk of failure based on Use Cases.
Expected Result	As we expected all the running applications used for external communications to a network did function and fail as defined. For example, it should not be possible for the authentication protocol to be manipulated or for the messaging protocols to be spoofed by an application or other clients or servers.
Actual Result	However, actual results showed that applications do close unexpectedly and fail the devices.
Knowledge Gained	We gained knowledge that when handling exceptions, an application should provide an error message.
New Requirement.	The device should present a message that is relevant to the context of the application failure.

Man in the Middle

Test 14	We directed Test 14 towards poor authorization and authentication.
What Is Tested and Analyzed?	The possibility of Man-in-the-Middle attack (Kim, Jeong, & Hong, 2013)
Overview	According to Kim, Jeong, and Hong (2013), it is difficult to detect or prevent the Man-in-the-Middle (MitM) attack in the M2M ecosystem. The problem is elevated due to the difficulties of managing or controlling each device independently within the system.
Issue	According to Jin (2013), M2M ecosystems lack proper device situational recognition within the systems that certifies the platform and message protections. Today, these systems use identity-based algorithms for situational recognition and a convergence framework to analyze certification technology that reply on keys and other sensitive information. In addition, MitM attacks target integrity and confidentiality from a messaging standpoint, which allows the attacker to take over as the Core Network by representing the gateway to the device.
Description	We executed a Man-in-the-Middle on the device by intruding into a controlled network. We tried to detect basic flaws by observing the device’s network traffic, protocol design and application, and the server configuration. The goal is to expose the lack of security protections of data in transit over the communication pipe by gaining unauthorized possession of the device.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, and application security (CISSP, 2014)
Affected Stakeholders	Application service provider, M2M gateways, gateway management entities, network operator, and user/consumer
Architecture Impact	Radio network
Action	1. Set up open source software (OpenBTS and FreeSWITCH) and the Ettus Transceiver to route voice and SMS traffic through a private GSM network. 2. Hook up the Raspberry Pi to a radio interface for device testing. 3. The Raspberry Pi runs: OpenBTS: GSM mobile phone standards network FreeSWITCH: call routing tool Python: for programming scripts 4. Force device to attach to GSM network. This takes advantage of weaknesses in the GSM protocol stack. 5. Eavesdrop on the following devices: M2M devices to the M2M gateway

	M2M gateway to M2M devices 6. Test the protection in communications protocols by sending authentication messages. 7. Take advantage of weaknesses in the GSM protocol by performing unexpected actions, such as sending false identity data. 8. Manipulate messaging protocols. 9. Manipulate an authentication protocol. 10. Seek older protocol vulnerabilities. 11. Look at logs to see if Detailed in Chapter 4 can read sensitive information or modify message contents. 12. Send back captured messages using Python script messages to change data content. 13. Manipulate client-server interactions. 14. Check the receipt of the repetition messages. 15. Log the capability of attack.
Expected Result	As we expected we successfully caused the device to roam to a fake core network by impersonating a gateway that looks legitimate to the device (3GPP, 2002).
Actual Result	We found that known MITM attack vectors do capitalize on (**i.e.** AP or fake AP capability) and that the Proxy tool or sniffer is required to determine if vulnerable and OpenBTS roaming.
Knowledge Gained	We gained the understanding that the capability of an intruder to put itself in between the target user and is a genuine threat in M2M as described by past research. The ability to eavesdrop, modify, delete, re-order, replay, and spoof signaling and user data messages is possible (Kim, Jeong, & Hong, 2013).
New Requirement.	Detailed in Chapter 4.

Buffer Overflows

Test 15	We directed Test 15 towards security decisions via untrusted inputs and lack of binary protections.
What Is Tested and Analyzed?	The difficulty of a buffer overflow condition as described by (Hagar, 2013).
Overview	The application framework of all devices manages the functions that perform various tasks, like resource management and call management. When an erroneous condition, such as a buffer overflow, occurs, the device processing power is stressed beyond the boundaries of the store data buffer limits. This condition leads to extra data overwriting and failure of the device processor's memory

	locations, which causes the device to fail (Shewale, Patil, Deshmukh & Singh, 2014).
Issue	Literature states that buffer overflow condition attacks corrupt data, crash the programs, or cause the execution of malicious code.
Description	As we understand the description by Hagar (2013), we tried to cause a strange error messages by sending data and messages to various abbreviations of application program interfaces (APIs) of the devices. All APIs are designed to have length constraints for the utilization of storage, data locations, and code. These constraints define execution space. We tried to find any vulnerabilities that enable the execution of applications without proper authentication by exploiting the buffers, requesting additional header handling, and overflowing the authentication handling of the device.
Impacted Use Cases	All
Affected Security Domain	Access control, application security, application environment, and security controls (CISSP, 2014)
Affected Stakeholders	Application service provider
Architecture Impact	Application security framework
Action	1. Start the wireless interface in monitor mode on the specific API channel or port. 2. Test the injection capability of the wireless device by sending data to the API. 3. Using open source injection tools such as airodump, aireplay, and aircrack, test and confirm the API can be injected prior to proceeding. If it cannot be, change API ports. 4. Use correct authentication credentials in the messages for baseline results. 5. Use fake authentication credentials in the messages document results. 6. In a requested replay mode, inject packets. 7. Increase packet injection until device fails. 8. Collect error messages.
Expected Result	We expected that the exploitation of buffer overflow vulnerabilities do at some point exposed vulnerabilities. What is important is at what point and the measurement of "ease of action" of the testing. It was also expected that the buffer overflow will happen at the point the message is received and the device tried to store the message.
Actual Result	We found that no buffer overflow was found or actually detected.

Knowledge Gained	Knowledge gained concluded that there is a possibility that the communications module used has a limited buffer and the header size and bigger files will cause a buffer overflow.
New Requirement.	Limiting the buffer size and using a GET method with short answers for all request and using AT commands for HTTP communication might correct this vulnerability.

False Data Injection

Test 16	We directed Test 16 towards poor authorization and authentication, client side injection and security decisions via untrusted inputs.
What Is Tested and Analyzed?	The difficulty of false data injection as described (Lu, Lin, Zhu, Liang, & Shen, 2012)
Overview	According to Lu et al. (2012), networks and devices like M2M are seriously threatened by false data injection attacks. These attacks threaten authentication if they discover the capability of bypassing administrative privileges, can discover and view sensitive information, and can alter contents in a database.
Issue	We found that the issue is that if the availability of data is compromised, the authentication processes may be defeated. In that case, all sensitive information on the device is at risk.
Description	We researched the extent to which one may use the remote command execution technique to send mass injections to a device by sending simple text-based attacks as an injection vector. The goal was to understand the risk associated with injection flaws sent to and from untrusted application and devices.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, and application security (CISSP, 2014)
Affected Stakeholders	M2M application service provider, manufacturer of M2M devices and/or M2M gateways, M2M service provider, and user/consumer
Architecture Impact	CSE, Mca-reference point, and Mcc-reference point
Action	1. Start the wireless interface in monitor mode on the specific wireless GSM channel. 2. Use NowSMS as a tool to send a simple, text-based injection from the command line. 3. Test the injection capability of the wireless device from the gateway. 4. Use command line messaging to send fake authentication credentials.

	5. Send simple, text-based syntax to the device interpreter. 6. As an injection vector, use untrusted data to seek injection flaws. 7. Collect all new unique errors from both Wireshark and NowSMS. 8. Make various API requests in replay mode to insert injected packets within application. 9. Collect results and evaluate risk.
Expected Result	We expected that injection attacks would be mitigated by simple authentication. However, it is unknown: a) if the small size, low power, and unattended operations make a false data injection a higher risk and b) what other attacks can be launched by a compromised M2M device.
Actual Result	We found that various forms of side-channel attacks can be tested against the devices.
Knowledge Gained	We gained knowledge that after running several Python script that should have cause fault injection they freeze the RPi CPU. The timing attack however is inclusive based on the how long the device takes to execute commands when the Fault Injection script is running.
New Requirement.	Detailed in Chapter 4.

Session Management

Test 17	We directed Test 17 towards poor authorization and authentication and improper session handling.
What Is Tested and Analyzed?	Tested is session management and broken authentication as described by (Lake, Milito, Morrow, & Vargheese, 2013)
Overview	According to Okugawa, Masutani, and Yoda (2005), M2M networks may be self-organizing and composed of scattered small mobile devices that require the survivability of simultaneous communicating endpoints. The identity of the device is a key to managing the moving part. If the session management or authentication is broken, all ecosystem functions are at risk of attack (Lake, et al., 2013).
Issue	The issue is that vulnerabilities in session authentication protocols lead to integrity and privacy issues because of key exchange failures and setup integrity flaws. These flaws fail at session shutdown and within authentication schemes such as logout, account update, and session timeout.
Description	We tested the device methods of ensuring key privacy and verified that exploitation is of at least average difficulty and that “leaks or flaws in the authentication or session management functions” can be addressed by verifying that there are no exposed accounts, passwords, or session

	IDs as described by Lake, et al., (2013) and oneM2M, (2013; 2014).
Impacted Use Cases	All
Affected Security Domain	Integrity, availability, access control, communication channels, network attacks, encryption concepts, and resource protection (CISSP, 2014)
Affected Stakeholders	Manufacturer of M2M devices, M2M gateways, device/gateway management entities, M2M service provider, and user/consumer
Architecture Impact	Key management and protocol architecture
Action	1. Using tools like Backtrack, target the device over an IP session. 2. Monitor the wireless interface normal session. 3. Capture the messaging traffic sent to the gateway. 4. Intercept all data that can be used to execute the attack. 5. Get the session ID. 6. Inspect the logs for any session tokens. 7. The device should be using an expiration timeout that can be located in the session token. Verify that the token is cryptographically protected from tampering. 8. Using the injection test to confirm injection.
Expected Result	We expected that no data would be visible other than the authenticated data that should be visible on the device after the termination of the session. The same results were expected on the gateway side after session termination, when all data is removed from the logs after an inactivity timeout, complete reboot.
Actual Result	We found that the man in the middle attacks and false base station proves vulnerability when using known network authentication.
Knowledge Gained	We gained the knowledge that the main function of the communication module is to authentication, send and receive data. As know the GSM software network comments show the same probable security functions for mobility and session management as commercial network, thus have the same vulnerabilities.
New Requirement.	Detailed in Chapter 4.

Security Misconfiguration

Test 18	We directed Test 18 towards security decisions via untrusted inputs, transport layer protection and improper session handling
What Is Tested and Analyzed?	The security misconfiguration results on the device as described by (Hongsong, Zhongchuan, & Dongyan, 2011).

Overview	If a device's security is misconfigured, then various events can take place that may hinder the device's performance. Distributed values and functions may be directly affected such as device health, remote control, management, and the embedded applications' restrictive security model. M2M services have to support evolving requirements and dynamically involve activities such as unsigned applications permission, unprotected APIs, and non-protected registry keys (Drira, 2010).
Issue	We found that the issue is that misconfiguration attacks exploit configuration weaknesses that can fail account access protections, expose-patching flaws, compromise "unprotected files and directories, and grant unauthorized access to the device" (oneM2M, 2014). Most devices are provided "off-the-shelf" with unnecessary and unsafe features that are enabled by default, including backdoor accounts, special access mechanisms, and incorrect permissions.
Description	We researched the device security configuration by reviewing the unauthorized access to sensitive information security policies. Attempt to compromise the device and M2M System by gaining unauthorized access to or knowledge of device accounts, applications, and platform.
Impacted Use Cases	All
Affected Security Domain	Availability, access control, application environment and security controls, encryption concepts, capabilities of information systems, and patch and vulnerability management (CISSP, 2014)
Affected Stakeholders	M2M application service provider, device, and user/consumer
Architecture Impact	Security policies, including policy execution, default value protection, and application roles
Action	1. Test the security policies framework. 2. Research indicative security vulnerabilities from the outside in by examining application binaries for unprotected conditions. 3. Test and verify that the device interface controls automatically logout of all sessions. 4. Test session termination after a given amount of time without activity (session timeout). 5. Document results.
Expected Result	We expected that the accessible device environments are easy to exploit, that all applications have permission to run, that based configuration files are not properly locked down, that clear text reveals username and password type data, and that database connection strings are set to default settings in configuration.
Actual Result	We found that because the devices are built prototypes, services and application can be turning off and on because the root logon/password is known. This might not be the situation on a commercial device.
Knowledge Gained	We gained knowledge that several design and development-related vulnerabilities can be found if services are on and not secure. And that

	the vulnerabilities of misconfiguration and administration errors, should show in the logs.
New Requirement.	Detailed in Chapter 4.

Insecure Cryptographic Storage

Test 19	We directed Test 19 towards broken cryptography.
What Is Tested and Analyzed?	The possibility of and damage from insecure cryptographic storage as described by (Hussen, 2013)
Overview	According to Hussen (2013), M2M devices lack secure authentication, session key exchange schemes, and adequate cryptographic storage for small packet transmission. M2M devices lack these because the devices have low power requirements, insufficient processing ability, and associated resource constraints.
Issue	We find that the issue is that M2M devices have weak cryptographic algorithms and lack the power to decipher cipher text, which is what makes cryptanalysis successful.
Description	We tested the ability and ease of finding keys, viewing clear text copies of data, and accessing channels automatically without decrypting data. We wanted to understand how the device is encrypting data, how it is generating and storing safe keys, and what algorithms are deployed for secure cryptographic storage.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, application security, application environment and security controls, and encryption concepts (CISSP, 2014)
Affected Stakeholders	Service provider, network operator, and user/consumer
Architecture Impact	Cryptographic architectural framework
Action	1. Use scanning suites and packet sniffers to analyze protocol messaging. 2. Execute cryptanalysis on applications via error logs and on messages via scanner. 3. Separate plaintexts from any cipher texts. 4. Try to find the secret key and key storage area. 5. Break down and distinguish each algorithm’s output. 6. Test the functionality of equivalent algorithms for encryption and decryption. 7. Review information from all error messages and other

	descriptive messages. 8. Observe message patterns. 9. Perform formal analysis of protocols. 10. Document the outcome and disclosure of all sensitive information.
Expected Result	We expected that cryptanalysis would prove the need for new methods to protect message context and secure plaintext data transport. The devices lack encryption algorithms that do not require key sizes, and these are important for keeping data confidential.
Actual Result	We found that due to the mobility characteristic of devices they run mobile application that carry the same vulnerabilities known to past researcher, requiring protection of the application, the user data and system data.
Knowledge Gained	The removable storage device is not secure.

Invalid Input Data

Test 20	We directed Test 20 towards security decisions via untrusted inputs.
What Is Tested and Analyzed?	The validation of input data as required to ensure content in order to provide proper applications security (Ellinas, Panayiotou, Kyriakides, & Polycarpou, 2015)
Overview	It is important that M2M devices and gateways have the functionality to detect and prevent unauthorized access to the ecosystem. In the case of ecosystem federation, there is the possibility to have non-existent communications security between devices, due to lack of requirements requiring protection with mutual authentication so add protections against invalid input data and strict parameters that outline qualifiers, range, and data fields is required.
Issue	We found the issue to be that the injection of specific exploits such as buffer overflows, SQL injections, and cross-site scripting will grant access to the device and gateway’s functionality and privilege escalation.
Description	We tested the ability of using invalid input data to gain control of the device. Analyze the difficulty to impose a Denial of Service, bypass authentication, and escalate privileges by accessing unintended functionality, executing remote code, and stealing data.
Impacted Use Cases	All
Affected Security Domain	Integrity, confidentiality, availability, access control, communication channels, network attacks, application security, application environment and security controls, encryption concepts, capabilities of information systems, resource protection, incident response, patch and vulnerability management, disaster recovery process, and internal security (CISSP,

	2014)
Affected Stakeholders	M2M application service provider, user/consumer
Architecture Impact	Messaging architecture, security architecture
Action	1. Use a remote access tool such as NowSMS. 2. Validate input vulnerabilities by causing a device crash or DoS attack. 3. Send an SMS message to the device with execution scripts that are embedded with other exaction code, such as “location request.” 4. Note and verify that the device OS and application require sufficient privileges to execute a script. 5. Note and verify that the applications are protected against malicious written scripts that include rogue strict type characters and lack encoding enforcement. 6. Ensure that all message content is delivered to the device and then sanitized against unacceptable content specification. 7. Perform input validation by reviewing all error logs. 8. Perform output validation by reviewing all error logs. 9. Verify that session tokens function correctly. 10. Document that all privilege constraints are authorized and in policy.
Expected Result	As we expected the input data sets do at some point fail the device and that sending misleading data to the device might cause a DoS.
Actual Result	We found that sensitive data from applications such as passwords and account information can be stored on an external card without OS warning the user of this occurring, this should be protected or prohibited on commercial devices. Also the investigation of AT COMMANDS used as invalid data should be further studied.
Knowledge Gained	Detailed in Chapter 4.
New Requirement.	Presented in Chapter 4. Note: For mitigation, stronger message authentication is required and lightweight encryption should be used to provide secrecy (Shah, Perrig, & Sinopoli, 2008; Awad, 2015).

Cross Scripting

Test 21	We directed Test 21 towards improper session handling, client side injection and transport layer protection.
What Is Tested and Analyzed?	Cross scripting towards a device as described by (Gyrard, Bonnet, & Boudaoud, 2014)

Overview	The M2M ecosystem passes communications over various networks. The devices are capable of sending messages and data using protocols such as HTTP or SIP. Thus, cross-site scripting (XSS) attacks may target events and allow for code or data injection that disrupts the communication path of the device (Siewruk, Sredniawa, Grabowski, & Legierski, 2013).
Issue	Literature explains the issues as the feasibility to use forbidden commands to bypass filters on a device using alternate forms of messaging syntax, which will cause the device to fail when processing.
Description	We tested for protocol weaknesses by using a cross-site scripting process by executing arbitrary commands from SMS messages and AT Commands.
Impacted Use Cases	All
Affected Security Domain	Communication channels, network attacks, application security, application environment, and security controls (CISSP, 2014)
Affected Stakeholders	Service provider, M2M gateways, and user/consumer
Architecture Impact	Protocols and the method of implementations of applications.
Action	1. Using browser technologies that allow client side scripting, create various file submissions that will be sent from the control center to the device. 2. Design and utilize characters type changes, such as SMS messaging, to test coding security enforcement. 3. Deliver to the device to prove that the present protocols are functioning correctly against acceptable content specification. 4. Ensure that all content coming from the device uses the correct encoding for message reply. 5. Document the application response and delivery for timing, structure, and content correctness. 6. Using the control center device management tools, apply messaging filtering. 7. Rerun messaging test. 8. Perform input validation for all remote content that includes automated remote and user-generated content. 9. Document the output validation for all content.
Expected Result	As expected we found that ability to bypass filters where “scripts” are executed is a prohibited functionality. All input from the server side do validate. The data-input did not fail the device. The device validation application did prevent and protect unauthorized input from infecting other on-board applications.

Actual Result	We found that the detection of potential cross site scripting, verbose errors and forceful browsing are aspects and typically identified with automated tool, however not a good test against devices. .
Knowledge Gained	When looking at cross-site scripting (XSS) and how this attacks a web application with data send malicious code must be use. When testing this in the form of a browser side script, the flaws allow attacks to input data and the user may see the output if it is generated without validating, further testing and planning for this type of attacked is required for devices. But for M2M devices there is no end user to view the output.
New Requirement	Only specific types of data may be authorized to be sent to and from the M2M devices.

References

Abdullah, M., Welch, I., & Seah, W. (2013). Efficient and secure data aggregation for smart metering networks. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (pp. 71–76). IEEE.

Abie, H., & Balasingham, I. (2012). Risk-based adaptive security for smart IoT in eHealth. In *Proceedings of the 7th International Conference on Body Area Networks* (pp. 29–275). Brussels, Belgium: ICST.

Accorsi, R., Stocker, T., & Müller, G. (2013). On the exploitation of process mining for security audits: The process discovery case. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 1462–1468). New York, NY: ACM. doi=10.1145/2480362.2480634

Alberts, C., Allen, J., & Stoddard, R. (2012). Risk-Based measurement and analysis: Application to software security. Retrieved May 30, 2014, from the Software Engineering Institute of Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=10067

Alqassem, I. (2014). Privacy and security requirements framework for the Internet of Things (IoT). In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 739–741). New York, NY: ACM. doi=10.1145/2591062.2591201

Anggorojati, B., Prasad, N. R., & Prasad, R. (2013). An intrusion detection game in access control system for the M2M local cloud platform. In *Proceedings of the 2013 19th Asia-Pacific Conference on Communications* (pp. 345–350). IEEE.

Aslam, M., Gehrmann, C., & Björkman, M. (2013). Continuous security evaluation and auditing of remote platforms by combining trusted computing and security automation techniques. In *Proceedings of the 6th International Conference on Security of Information and Networks* (pp. 136–143). New York, NY: ACM. doi=10.1145/2523514.2523537

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805.

Aucinas, A., Crowcroft, J., & Hui, P. (2012). Energy efficient mobile M2M communications.

Awad, I. A. (2015). Security and privacy. [PowerPoint slides]. Retrieved from: http://www7.cs.fau.de/de/wp-content/uploads/sites/2/2014/08/lect12_security.pdf

Aviv, A. J., Gibson, K., Mossop, E., Blaze, M., & Smith, J. M. (2010). Smudge attacks on Smartphone touch screens. *Workshop of Offensive Technologies, 10*, 1–7.

Bahga, A., & Madisetti, V. (2014). Internet of Things: A hands-on approach. VPT.

Bartoli, A., Hernández-Serrano, J., Soriano, M., Dohler, M., Kountouris, A., & Barthel, D. (2011). *Secure lossless aggregation over fading and shadowing channels for smart grid M2M networks.* Smart Grid, IEEE Transactions on, 2(4), 844-864.

Batalla, J., & Krawiec, P. (2014). Conception of ID layer performance at the network level for Internet of Things. *Personal Ubiquitous Computing, 18*(2), 465–480. doi=10.1007/s00779-013-0664-0

Beigi Mohammadi, N., Mišić, J., Mišić, V. B., & Khazaei, H. (2014). A framework for intrusion detection system in advanced metering infrastructure. *Security and Communication Networks, 7*(1), 195–205.

Ben-Asher, N., Kirschnick, N., Sieger, H., Meyer, J., Ben-Oved, A., & Moller, S. (2011). On the need for different security methods on mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (pp. 465–473). doi= 10.1145/2037373.2037442

Benjamins, R. (2014). Big data: From hype to reality? *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics, 2,* 2. doi=10.1145/2611040.2611042

Bernardi, S., Merseguer, J., & Petriu, D. C. (2013). *Model-Driven dependability assessment of software systems*. Springer.

Bianchi, E. (2014). Critical attacks: How economy could be saved by cyber insurance. Retrieved from Lecture Notes Online website: http://tesi.eprints.luiss.it/13069/2/bianchi-eleonora-sintesi-2014.pdf

Bhunia, S. S., Pal, J., & Mukherjee, N. (2014). Fuzzy assisted event driven data collection from sensor nodes in sensor-cloud infrastructure. In *Proceedings of the 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 635–640). IEEE.

Brahmi, H. (2014). Towards efficient data collection in WSNs. In *Proceedings of the 2014 Workshop on PhD Forum* (pp. 11–12). New York, NY: ACM. doi=10.1145/2611166.2611172

Bojic, I., Jezic, G., Katusic, D., Desic, S., Kusek, M., & Huljenic, D. (2012). Communication in machine-to-machine environments. In *Proceedings of the Fifth Balkan Conference in Informatics* (pp. 283–286). New York, NY: ACM. doi=10.1145/2371316.2371379

Boswarthick, D., Elloumi, O., & Hersent, O. (Eds.). (2012). *M2M communications: A systems approach.* John Wiley & Sons.

Buyens, K., Scandariato, R., & Joosen, W. (2009). Measuring the interplay of security principles in software architectures. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (pp. 554–563).

Cam-Winget, N., & Didier, P. (2014). Stateful contextually-aware access control for ICS. [PDF] Retrieved from Lecture Notes Online website: https://www.usenix.org/legacy/event/woot10/tech/full_papers/Aviv.pdf

Casson, M., & Della Giusta, M. (2014). Buzzwords in business and management studies. In *Handbook of research on small business and entrepreneurship* (p. 38).

Cha, I., Shah, Y., Schmidt, A. U., Leicher, A., & Meyerstein, M. V. (2009). Trust in M2M communication. *IEEE Vehicular Technology Magazine*, *4*(3), 69–75.

Chan, C. Y. (2011). Connected vehicles in a connected world. In *2011 International Symposium on VLSI Design, Automation and Test* (pp. 1–4). IEEE.

Chang, S. G. (2014). A structured scenario approach to multi-screen ecosystem forecasting in Korean communications market. Technological Forecasting and Social Change. [PDF]. Retrieved from Lecture Notes Online website: http://www.sciencedirect.com/science/article/pii/S0040162514001267

Chen, S., & Ma, M. (2014). Security issues in machine-to-machine communication. In *Security for Multihop Wireless Networks* (p. 401).

Chaugule, A., Xu, Z., & Zhu, S. (2011). A specification based intrusion detection framework for mobile phones. In *Proceedings of the 9th International Conference on Applied Cryptography and Network Security* (pp. 19–37).

Chen, D., & Chang, G. (2012). A survey on security issues of M2M communications in cyber-physical systems. *KSII Transactions on Internet and Information Systems, 6*(1), 24–45.

Chen, S., & Ma, M. (2014). Security issues in machine-to-machine communication. In *Security for Multihop Wireless Networks* (p. 401).

Chen, Y. K. (2012). Challenges and opportunities of Internet of Things. *Proceedings of the Design Automation Conference, 2*, 383–388. doi=10.1109/ASPDAC.2012.6164978

Chin, E., Porter Felt, A., Sekar, V., & Wagner, D. (2012). Measuring user confidence in Smartphone security and privacy. *Proceedings of the Eighth Symposium on Usable Privacy and Security, 1*, 16. doi=10.1145/2335356.2335358

Choi, Y., Doh, I., Park, S. S., & Chae, K. J. (2013). Security based semantic context awareness system for M2M ubiquitous healthcare service. In *Ubiquitous information technologies and applications* (pp. 187–196). Netherlands: Springer.

Christiansen, L. (2011). Personal privacy and internet marketing: An impossible conflict or a marriage made in heaven? *Business Horizons, 54*, 509–514.

Chui, M., Löffler, M., & Roberts, R. (2010). The Internet of Things. *McKinsey Quarterly, 2*, 1–9.

Clarke, G. R., Reynders, D., & Wright, E. (2004). *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. Newnes.

Cruz, B., Duarte, A. M., & Ferreira, R. (2014). Impact of M2M communications on cellular telecommunications networks. In *Proceedings of the ICDS 2014, The Eighth International Conference on Digital Society* (pp. 86–92).

Cohen, S., Money, W., & Quick, M. (2014). Improving integration and insight in smart cities with policy and trust. *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics, 57*, 9. doi=10.1145/2611040.2611091

Constantinos, K., Coursaris, & Kim, D. (2011). A meta-analytical review of empirical mobile usability studies. *Journal of Usability Studies, 6*, 3–58.

Conti, J. P. (2006). The Internet of Things. *Communications Engineer, 4*(6), 20–25.

Dahl, Y., & Holbø, K. (2012). Value biases of sensor-based assistive technology: Case study of a GPS tracking system used in dementia care. In *Proceedings of the Designing Interactive Systems Conference* (pp. 572–581). ACM.

Das, A., Borisov, N., Mittal, P., & Caesar, M. (2014). Re3: Relay reliability reputation for anonymity systems. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security* (pp. 63–74). New York, NY: ACM. doi=10.1145/2590296.2590338

Dehghani, R., & Ramsin, R. (2015). Methodologies for Developing Knowledge Management Systems: An Evaluation Framework. Journal of Knowledge Management, 19(4).

Distefano, A., Grillo, A., Lentini, A., & Italiano, G. F. (2010). SecureMyDroid: Enforcing security in the mobile devices lifecycle. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research* (pp. 1–4). doi=10.1145/1852666.1852696

Dye, S. M., & Scarfone, K. (2014). A standard for developing secure mobile applications. *Computer Standards & Interfaces*, *36*(3), 524–530.

Ebersold, K. (2014). *The Internet of Things*. Honors Projects in Computer Information Systems. Paper 4. Retrieved from http://digitalcommons.bryant.edu/honors_cis/4

Ellinas, G., Panayiotou, C., Kyriakides, E., & Polycarpou, M. (2015). Critical infrastructure systems: Basic principles of monitoring, control, and security. In *Intelligent monitoring, control, and security of critical infrastructure systems* (pp. 1–30). Berlin, Heidelberg: Springer.

El-Mahdy, A. (2014). *M2M technologies: The state-of-the art.* [PowerPoint slides]. Retrieved from http://labs.ejust.edu.eg/pcl/images/m2m-public.pdf

Elmangoush, A., Steinke, R., Al-Hezmi, A., & Magedanz, T. (2014). On the usage of standardised M2M platforms for smart energy management. In *2014 International Conference on Information Networking* (pp. 79–84). IEEE.

Emerging Cyber Threats Report 2012. (2011). Georgia Tech Information Security Center, & Georgia Tech Research Institute.

Ennesser, F. (2012). Smart cards in M2M communication. In *M2M communications: A systems approach* (pp. 273–294).

Evans, R. P., Hammond, V. B., & Shamsuddin, S. A. (2014). Inherently secure next-generation computing and communication networks for reducing cascading impacts. In *Wiley Handbook of Science and Technology for Homeland Security*.

Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011). A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices* (pp. 3–14). doi=10.1145/2046614.2046618

Flick, T., & Morehouse, J. (2010). *Securing the smart grid: Next generation power grid security*. Elsevier.

Floeck, M., Papageorgiou, A., Schuelke, A., & Song, J. (2014). Horizontal M2M platforms boost vertical industry: Effectiveness study for building energy management systems. In *2014 IEEE World Forum on Internet of Things* (pp. 15–20). IEEE.

Flood, J., & Keane, A. (2014). A framework to improve threat vector analysis through the use of gamification. In *9th International Conference on Cyber Warfare & Security* (p. 247). Academic Conferences Limited.

Fischer, S. (2014). Challenges of the Internet of Services. In *Towards the Internet of Services: The THESEUS research program* (pp. 15–27). Springer International Publishing.

Foschini, L., Taleb, T., Corradi, A., & Bottazzi, D. (2011). *M2M-based metropolitan platform for IMS-enabled road traffic management in IoT.* Communications Magazine, IEEE, 49(11), 50-57.

Gasson, M., Kosta, E., Royer, D., Meints, M., & Warwick, K. (2011). Normality mining: Privacy implications of behavioral profiles drawn from GPS enabled mobile phones. *IEEE Transactions*, *41*, 251–262.

Granjal, J., Monteiro, E., & Silva, J. S. (2013). Security issues and approaches on wireless M2M systems. In *Wireless Networks and Security* (pp. 133–164). Berlin, Heidelberg: Springer.

Godfrey, J., Horton, M., Demblewski, M., Bearden, T., Morovitz, J., (2015). *Securing the M2M Ecosystem.* Unpublished PowerPoint slides. AT&T, Dallas.

Goel, S. (2011). Cyberwarfare: Connecting the dots in cyber intelligence. *Communications of the ACM*, *54*(8), 132–140. doi= 10.1145/1978542.1978569

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645-1660.

Gyrard, A. (2013). A machine-to-machine architecture to merge semantic sensor measurements. In *Proceedings of the 22nd International Conference on World Wide Web Companion* (pp. 371–376). Geneva, Switzerland: International World Wide Web Conferences Steering Committee, Republic and Canton.

Gyrard, A., Bonnet, C., & Boudaoud, K. (2014). An ontology-based approach for helping to secure the ETSI machine-to-machine architecture. [PDF]. In *IEEE International Conference on Internet of Things 2014.* Retrieved from https://hal.archives-ouvertes.fr/hal-01017945/document

Hagar, J. D. (2013). *Software test attacks to break mobile and embedded devices.* Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series. Taylor and Francis.

Hammoudeh, M. A., Mancilla-David, F., Selman, J. D., & Papantoni-Kazakos, P. (2013). Communication architectures for distribution networks within the Smart Grid Initiative. In *Green Technologies Conference* (pp. 65–70). IEEE.

Hamida, S., Ben Hamida, E., Ahmed, B., & Abu-Dayya, A. (2013). Towards efficient and secure in-home wearable insomnia monitoring and diagnosis system. In *2013 IEEE 13th International Conference on Bioinformatics and Bioengineering* (pp. 1–6). IEEE.

Hersent, O., Boswarthick, D., & Elloumi, O. (2011). The Internet of Things: Key applications and protocols. John Wiley & Sons.

Heu, A. B., Heu, P. G., Cea, A. O., Cfr, M. R., & Stefa, J. (2013). Internet of Things Architecture.

Hubbard, D. W. (2014). *How to measure anything: Finding the value of intangibles in business.* John Wiley & Sons.

Hussen, H. R., Tizazu, G. A., Ting, M., Lee, T., Choi, Y., & Kim, K. H. (2013). SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network. In *2013 Fifth International Conference on Ubiquitous and Future Networks* (pp. 246–251). IEEE.

Hsu, Y. H., Wang, K., & Tseng, Y. C. (2013). Enhanced cooperative access class barring and traffic adaptive radio resource management for M2M communications over LTE-A. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (pp. 1–6). IEEE.

Ho, E., Jacobs, T., Meissner, S., Meyer, S., Monjas, M. A., & Segura, A. S. (2013). ARM testimonials. In *Enabling Things to Talk* (pp. 279–322). Berlin, Heidelberg: Springer.

Hongsong, C., Zhongchuan, F., & Dongyan, Z. (2011). Security and trust research in M2M system. In *2011 IEEE International Conference on Vehicular Electronics and Safety* (pp. 286–290). IEEE.curity

Horton, M., (2014). *M2M Device Security Considerations*, Data retrieved from www.m2isf.com.

Hosek, J., Masek, P., Kovac, D., Ries, M., & Kröpfl, F. (2014). IP home gateway as universal multi-purpose enabler for smart home services. *Elektrotechnik und Informationstechnik, 131*(4–5), 123–128. Retrieved from: http://link.springer.com/article/10.1007/s00502-014-0209-x#page-1

Igarashi, Y., Joshi, K., Hiltunen, M., & Schlichting, R. (2014). Vision: Towards an extensible app ecosystem for home automation through cloud-offload. In *Proceedings of the Fifth International Workshop on Mobile Cloud Computing & Services* (pp. 35–39). New York, NY: ACM. doi=10.1145/2609908

Igure, V. M., Laughter, S. A., & Williams, R. D. (2006). Security issues in SCADA networks. *Computers & Security, 25*(7), 498–506.

Jeon, Y. B., Lee, K. H., Park, D. S., & Jeong, C. S. (2013). An efficient cluster authentication scheme based on VANET environment in M2M application. *International Journal of Distributed Sensor Networks*.

Jermyn, J., Salles-Loustau, G., & Zonouz, S. (2014) An analysis of DoS attack strategies against the LTE RAN. *Journal of Cyber Security, 3*(2), 159–180.

Jin, B. (2013). An introduction: Context-aware computing for secure message transmission system in M2M. *SmartCR, 3*(6), 416–424.

Jones, S., Hara, S., & Augusto, J. (2014). eFRIEND: An ethical framework for intelligent environment development.

Jordan, B. (2014). Catalyst for the future: The emergence of mobile corporate real estate technologies. *Corporate Real Estate Journal, 3*(3), 184–198.

Joshi, S., Joshi, A., Jabade, S., Jathar, A., Gajbhiye, B. E., Bhoyar, V., & Vyas, J. (2014). M2M communication based wireless SCADA for real-time industrial automation.

Jung, S., Kim, D., & Kim, S. (2014). Cooperative architecture for secure M2M communication in distributed sensor networking [PDF]. Retrieved from: http://ijrat.org/downloads/april-2014/paper%20id-24201457.pdf

Katt, B., Gander, M., Breu, R., & Felderer, M. (2013). Enhancing model driven security through pattern refinement techniques. In *Formal Methods for Components and Objects* (pp. 169–183). Berlin, Heidelberg: Springer.

Kamal, R., Lee, J. H., Hwang, C. K., Moon, S. I., Hong, C. S., & Choi, M. J. (2013). Psychic: An autonomic inference engine for M2M management in *Future Internet*. In *2013 15th Asia-Pacific Network Operations and Management Symposium* (pp. 1–6). IEEE.

Kanuparthi, A., Karri, R., & Addepalli, S. (2013). Hardware and embedded security in the context of Internet of Things. In *Proceedings of the 2013 ACM Workshop on Security, Privacy, & Dependability for Cyber Vehicles* (pp. 61–64). New York, NY: ACM. doi=10.1145/2517968.2517976

Karim, A., Shah, S. A. A., & Salleh, R. (2014). Mobile botnet attacks: A thematic taxonomy. In *New Perspectives in Information Systems and Technologies, 2,* 153–164. Springer International Publishing.

Kazmi, Z., Felguera, T., Vila, J. A., & Marcos, M. M. (2012). TASAM—Towards the smart devices app-stores applications security management related best practices. In *2012 5th International Conference on New Technologies, Mobility and Security* (pp. 1–5). IEEE.

Khalil, D. (2010). Model-based management of ubiquitous and autonomic M2M service architecture. *Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems, 10,* 3. doi=10.1145/1868433.1868446

Khan, S., Khan, S., Nauman, M., Ali, T., & Alam, M. (2009). Realizing dynamic behavior attestation for mobile platforms. In *Proceedings of the 7th International Conference on Frontiers of Information Technology* (pp. 1–6). doi=10.1145/1838002.1838008

Khan, S., & Pathan, A. S. K. (2013). *Wireless networks and security: Issues, challenges and research trends.* Springer.

Kim, Y., He, K., Thottan, M., & Deshpande, J. (2014). Self-configurable and scalable utility communications enabled by software-defined networks. In *Proceedings of the 5th International Conference on Future Energy Systems* (pp. 217–218). New York, NY: ACM. doi=10.1145/2602044.2602074

Kim, K. J., & Hong, S. P. (2014). The study on the methods of secure communications on mobile device for intelligent services. *International Journal of Risk Assessment and Management*, *17*(4), 283–290.

Kim, J., Jeong, H., & Hong, H. (2013). A study of privacy problem solving using device and user authentication for M2M environments. *Security and Communication Networks*.

Kim, S., Kim, J., Kim, S., & Cho, H. (2011). A new shoulder-surfing resistant password for mobile environments. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication* (pp. 1–8). doi=10.1145/1968613.1968647

Kim, J., Wei, Y., & Lee, J. (2014). A scenario of machine-to-machine (M2M) health care service. *KNOM Review, 15*(2). Retrieved from: http://www.knom.or.kr/knom-review/v15n2/5.pd

Kitagami, S., Yamamoto, M., Koizumi, H., & Suganuma, T. (2013). An M2M data analysis service system based on open source software environments. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops* (pp. 953–958). IEEE.

King, N., & Jessen, P. (2010). Profiling the mobile customer: Is industry self-regulation adequate to protect consumer privacy when behavioral advertisers target mobile phones? *Computer Law and Security Review, 26,* 595–612.

Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., & Laurila, J. (2010). Towards rich mobile phone datasets: Lausanne data collection campaign. In *Proceedings of ICPS.* Lausanne, Switzerland.

Kriesten, B., Tünnermann, R., Mertes, C., & Hermann, T. (2010). Controlling ambient information flow between smart objects with a mobile mixed-reality interface. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services* (pp. 405–406). doi=10.1145/1851600.1851687

Koh, Y. M., & Kwon, K. H. (2014). A new lightweight protection method against impersonation attack on SIP. In *Advances in Computer Science and Its Applications* (pp. 273–277). Berlin, Heidelberg: Springer.

Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., & Carle, G. (2013). DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Networks, 11*(8), 2710–2723.

Kortuem, G., Kawsar, F., Fitton, D., & Sundramoorthy, V. (2010). Smart objects as building blocks for the Internet of Things. *IEEE Internet Computing, 14*(1), 44–51.

Kumar, N., & Chilamkurti, N. (2014). Collaborative trust aware intelligent intrusion detection in VANETs, computers, & electrical engineering. Retrieved from: http://www.sciencedirect.com/science/article/pii/S004579061400024X

Kylanpaa, M., Rantala, A., Merilinna, J., & Nieminen, M. (2013). Secure communication platform for distributed city-wide surveillance systems. In *2013 Fourth International Conference on Information, Intelligence, Systems and Applications* (pp. 1–4). IEEE.

Lai, M. Y., Chee, D., Makaya, C., Lin, F. J., Hori, K., & Yoshihara, K. (2012). U.S. Patent Application 13/650,701.

Lake, D., Milito, R., Morrow, M., & Vargheese, R. (2013). *Internet of Things: Architectural framework for eHealth security.* [PDF]. Retrieved from: http://riverpublishers.com/journal/journal_articles/RP_Journal_2245–800X_133.pdf

Landman, M. (2010). Managing smartphone security risks. In *2010 Information Security Curriculum Development Conference* (pp. 145–155). doi=10.1145/1940941.1940971

Latvakoski, J., Alaya, M. B., Ganem, H., Jubeh, B., Iivari, A., Leguay, J., ... , & Granqvist, N. (2014). Towards horizontal architecture for autonomic M2M service networks. *Future Internet, 6*(2), 261–301.

Laya, A., Alonso, L., & Alonso-Zarate, J. (2014). *Is the random access channel of LTE and LTE-A suitable for M2M communications?* A Survey of Alternatives. doi= 10.1109/SURV.2013.111313.00244

Lee, C., Lee, G., & Rhee, W. (2014). Smart ubiquitous networks for future telecommunication environments. *Computer Standards & Interfaces, 36*(2), 412–422.

Leontiadis, I., Efstratiou, C., Picone, M., & Mascolo, C. (2012). Don't kill my ads! Balancing privacy in an ad-supported mobile application market. *HotMobile'12 February 28–29.* San Diego, CA.

Lin, C. (2008). *Channel access management in data intensive sensor networks.* Retrieved from ProQuest Digital Dissertations.

Liu, C. H., Yang, B., & Liu, T. (2014). Efficient naming, addressing and profile services in Internet-of-Things sensory environments. *Ad Hoc Networks, 18,* 85–101.

Liu, G., Wang, R., Liu, J., Shen, J., & Shi, J. (2014). Research based on the information system framework building of "the new three mutual." In *Applied Mechanics and Materials 602* (pp. 3375–3378).

Liu, L., Zhang, X., Yan, G., & Chen, S. (2009). Exploitation and threat analysis of open mobile devices. In *Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems* (pp. 20–29). doi=10.1145/1882486.1882493

Li, Q., Gesbert, D., Gresset, N., Yin, H., Mehmeti, F., & Spyropoulos, T. (2014). Joint precoding over a master-slave coordination link. *Intel Technology Journal, 18*(1).

López, G., Moura, P., Moreno, J. I., & Camacho, J. M. (2014). Multi-Faceted assessment of a wireless communications infrastructure for the green neighborhoods of the smart grid. *Energies, 7*(5), 3453–3483.

Lopez de Ipiña, D., Vázquez, I., Ruiz de Garibay, J., Sainz, D., Lopez de Ipina, D., Vazquez, J. I., ... & Sainz, D. (2005). GPRS-based real-time remote control of {microbots} with {M2M} capabilities. In *The Fourth International Workshop on Wireless Information Systems, 53*, 145–150.

Lu, R., Li, X., Liang, X., Shen, X., & Lin, X. (2011). GRS: The green, reliability, and security of emerging machine to machine communications. *IEEE Communications Magazine, 49*(4), 28–35.

Lu, R., Lin, X., Zhu, H., Liang, X., & Shen, X. (2012). BECAN: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems, 23*(1), 32–43.

Macaulay, T., & Singer, B. L. (2011). *Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS*. CRC Press.

Marcovici, M. (2014). *The Internet of Things: This revolution could unlock $14.4 trillion in business value.* BoD–Books on Demand.

McLaughlin, S., Holbert, B., Fawaz, A., Berthier, R., & Zonouz, S. (2013). A multi-sensor energy theft detection framework for advanced metering infrastructures. *IEEE Journal on Selected Areas in Communications, 31*(7), 1319–1330.

McGrath, M. J., & Scanaill, C. N. (2013). Sensor network topologies and design considerations. In *Sensor Technologies* (pp. 79–95). Apress.

Miluzzo, E., Cornelius, C. T., Ramaswamy, A., Choudhury, T., Liu, Z., & Campbell, A. T. (2010). Darwin phones: The evolution of sensing and inference on mobile phones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (pp. 5–20). doi=10.1145/1814433.1814437

Minoli, D. (2015) *M2M Developments and Satellite Applications, in Innovations in Satellite Communications and Satellite Technology, The Industry Implications of DVB-S2X, High Throughput Satellites, Ultra HD, M2M, and IP*, John Wiley & Sons, Inc, Hoboken, NJ, USA. doi: 10.1002/9781118984086.ch6

Molotsi, K., & Tait, B. (2013). UMS-dev-sec: A proposed framework to address security concerns of UMS devices. In P. Machanick and M. Tsietsi (Eds.), *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 72–76). New York, NY: ACM. doi=10.1145/2513456.2513487

Murar, M., & Brad, S. (2014). Monitoring and controlling of smart equipments using Android compatible devices towards IoT applications and services in manufacturing industry. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics* (pp. 1–5). IEEE.

Murphy, D., & Murphy, R. (2013). Teaching cybersecurity: Protecting the business environment. In *Proceedings of the 2013 on InfoSecCD '13: Information Security Curriculum Development Conference* (pp.). New York, NY: ACM. doi=10.1145/2528908.2528913

Murynets, I., & Piqueras Jove, R. (2012). Crime scene investigation: SMS spam data analysis. In *Proceedings of the 2012 ACM Internet Measurement Conference* (pp. 441–452). New York, NY: ACM. doi=10.1145/2398776.2398822

Mukundan, N. R., & Sai, L. P. (2014). Perceived information security of internal users in Indian IT services industry. *Information Technology and Management, 15*(1), 1–8.

Nadji, Y., Giffin, J., & Traynor, P. (2011). Automated remote repair for mobile malware. In *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 413–422). doi:10.1145/2076732.2076791

Neumann, P. G. (2009). Risks to the public. *Software Engineering Notes, 34*(3), 16–29. doi=10.1145/1527202.1527205

Niyato, D., Xiao, L., & Wang, P. (2011). Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine, 49*(4), 53–59.

Nunamaker Jr, J. F., & Chen, M. (1990, January). Systems development in information systems research in System Sciences, 1990, *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences (3),* 631-640. IEEE.

Oberheide, J., & Jahanian, F. (2010). When mobile is harder than fixed (and vice versa): Demystifying security challenges in mobile environments. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems Applications* (pp. 43–48). doi=10.1145/1734583.1734595

Obikoya, G. D. (2014). Design, construction, and implementation of a remote fuel-level monitoring system. *EURASIP Journal on Wireless Communications and Networking, 1*, 76.

Ohki, E., Harada, Y., Kawaguchi, S., Shiozaki, T., & Kagaya, T. (2009). Information security governance framework. In *Proceedings of the First ACM Workshop on Information Security Governance* (pp. 1–6). New York, NY: ACM. doi=10.1145/1655168.1655170

Okugawa, T., Masutani, H., & Yoda, I. (2005). A home network service environment for wide-area communications. In *2005 Asia-Pacific Conference on Communications* (pp. 14–18). IEEE.

oneM2M-TR-0001. (2013), Technical Report "oneM2M Use Case collection", v0.0.5, Sep, 2013. [Online] retrieved from http://www.onem2m.org/

oneM2M-TS-0002. (2013), Technical Report "oneM2M Requirements Technical Specification" v0.6.2, Oct. 2013. [Online]. retrieved from http://www.onem2m.org/

oneM2M-TS-0008. (2124), Technical Report "oneM2M Requirements Technical Specification" v0.6.2, Oct. 2014. [Online]. retrieved from http://www.onem2m.org/

OWASP. *Mobile security project.* Retrieved December 2014 from website: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project. Online.

Owusu, E., Han, J., Das, S., Perrig, A., & Zhang, J. (2012). ACCessory: Password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems Applications* (pp. 1–6). doi= 10.1145/2162081.2162095

Pandey, S., Choi, M. J., Kim, M. S., & Hong, J. W. (2011). Towards management of machine to machine networks. In *2011 13th Asia-Pacific Network Operations and Management Symposium* (pp. 1–7). IEEE.

Pang, Z., Chen, Q., Tian, J., Zheng, L., & Dubrova, E. (2013). Ecosystem analysis in the design of open platform-based in-home healthcare terminals towards the Internet-of-Things. In *2013 15th International Conference on Advanced Communication Technology* (pp. 529–534). IEEE.

Park, M. W., Choi, Y. H., Eom, J. H., & Chung, T. M. (2013). Dangerous Wi-Fi access point: Attacks to benign smartphone applications. *Personal and Ubiquitous Computing* (pp. 1–14).

Park, R. C., Jung, H., Shin, D. K., Kim, G. J., & Yoon, K. H. (2014). M2M-based smart health service for human UI/UX using motion recognition. *Cluster Computing* (pp. 1–12).

Parkin, S. E., Moorsel, A. V., & Coles, R. (2009). An information security ontology incorporating human-behavioural implications. In *Proceedings of the 2nd International Conference on Security of Information and Networks* (pp. 46–55). doi=10.1145/1626195.1626209

Pérez-Cebollada, E., Martínez-Ruiz, I., & Bernal-Agustín, J. L. (2014). Challenges of M2M technologies for eHealth. In *Future Information Technology* (pp. 249–253). Berlin, Heidelberg: Springer.

Poncela, J., Moreno, J. M., & Aamir, M. (2014). *Analysis of M2M capabilities in 4G*. [PDF]. Retrieved from: http://dspace.uma.es/xmlui/bitstream/handle/10630/7483/Analysis%20of%20M2M%20Capabilities%20in%204G.pdf?sequence=1

Redman, P., Girard, J., & Wallin, L. (2011). *Magic quadrant for mobile device management software.* Gartner. Retrieved from http://www.gartner.com/id=1632331

Ren, W., Yu, L., Ma, L., & Ren, Y. (2013). How to authenticate a device? Formal authentication models for M2M communications defending against ghost compromising attack. *International Journal of Distributed Sensor Networks*,.

Rodríguez, N., Cuéllar, M., Lilius, L., & Calvo-Flores, M. (2014). A survey on ontologies for human behavior recognition. *ACM Computing Surveys, 46*(4), 33. doi=10.1145/2523819

Rubin, D., Lynch, K., Escaravage, J., & Lerner, H. (2014). Harnessing data for national security. *SAIS Review of International Affairs, 34*(1), 121–128.

Saedy, M., & Mojtahed, V. (2011). Ad hoc M2M communications and security based on 4G cellular system. In *Wireless Telecommunications Symposium* (pp. 1–5). IEEE.

Saeed, M. Y., Tahir, A., Mughal, S., & Khan, M. N. A. (2014). Insight into security challenges for cloud databases and data protection techniques for building trust in cloud computing.

Sarma, A. C., & Girão, J. (2009). Identities in the future Internet of Things. *Wireless Personal Communications, 49*(3), 353–363.

Savola, R. M. (2009). Current and emerging security, trust, dependability and privacy challenges in mobile telecommunications. In *Proceedings of the 2009 Second International Conference on Dependability* (pp. 7–12). doi= 10.1109/depend.2009.9

Siewruk, G., Sredniawa, M., Grabowski, S., & Legierski, J. (2013). Integration of context information from different sources: Unified communication, Telco 2.0 and M2M. In *2013 Federated Conference on Computer Science and Information Systems* (pp. 851–858). IEEE.

Scipioni, M., & Langheinrich, M. (2010). I'm here! Privacy challenges in mobile location sharing. In *Second International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use.* Helsinki, Finland.

Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A., & Leung, K. K. (2013). A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities. *IEEE Wireless Communications, 20*(6), 91–98.

Shewale, H., Patil, S., Deshmukh, V., & Singh, P. (2014). Analysis of Android vulnerabilities and modern exploitation techniques. *ICTACT Journal on Communication Technology*, *5*(1).

Skianis, C. (2013). Radio versus backhaul bottlenecks: An integrated quality of service provisioning approach for small cell gateways. *International Journal of Communication Systems.*

Seong, K. E., Lee, K. C., & Kang, S. J. (2014). Self M2M based wearable watch platform for collecting personal activity in real-time. In *Big Data and Smart Computing, 286*, 15–17.

Segura, L. (2011). U.S. patent application 13/028,093

Severi, S., Sottile, F., Abreu, G., Pastrone, C., Spirito, M., & Berens, F. (2014). M2M technologies: Enablers for a pervasive Internet of Things. In *2014 European Conference on Networks and Communications* (pp. 1–5). IEEE.

Shafiq, Z., Ji, L., Liu, A., Pang, J., & Wang, J. (2013). Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM Transactions on Networking, 21*(6), 1960–1973. doi=10.1109/TNET.2013.2256431

Shah, A., Perrig, A., & Sinopoli, B. (2008). Mechanisms to provide integrity in SCADA and PCS devices. In *Proceedings of the International Workshop on Cyber-Physical Systems-Challenges and Applications.*

Sharma, N., & Akhouri, S. (2014). Enabling distributed meter data management using mediation system. In *ENERGY 2014, The Fourth International Conference on Smart Grids, Green Communications and IT Energy-Aware Technologies* (pp. 1–6).

Sohr, K., Mustafa, T., & Nowak, A. (2011). Software security aspects of Java-based mobile phones. In *Proceedings of the 2011 ACM Symposium on Applied Computing* (pp. 1494–1501). doi=10.1145/1982185.1982506

Song, J., Kunz, A., Schmidt, M., & Szczytowski, P. (2014). Connecting and managing M2M devices in the future Internet. *Mobile Networks and Applications, 19*(1), 4–17.

Sousan, W., Gandhi, R., Zhu, Q., & Mahoney, W., (2011). Using anomalous event patterns in control systems for tamper detection. *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, 26*(1), . doi=10.1145/2179298.2179326

Tan, L., & Wang, N. (2010). Future internet: The Internet of Things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering, 5*, 5–376.

Tao, F., Zhang, L., Venkatesh, V. C., Luo, Y., & Cheng, Y. (2011). Cloud manufacturing: A computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. doi=0954405411405575.

TalebiFard, P., Nicanfar, H., Hu, X., & Leung, V. (2013). Semantic based networking of information in vehicular clouds based on dimensionality reduction. In *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications* (pp. 69–76). doi=10.1145/2512921.2512925

Third Generation Partnership Project, (2001), Technical Specification Group SA WG3; A guide to Third Generation security (3G TR 33.900 version 1.2.0).

Torbensen, R. (2011). On the emergence of pervasive home automation (Doctoral dissertation, University of Aalborg).

Unisys Security Index TM: GLOBAL SUMMARY 30 March, 2012 (Wave 1H’12)

Vaishnavi, V. K., & Kuechler, W. (2015). Design science research methods and patterns: innovating information and communication technology. CRC Press.

Vandikas, K., Liebau, N. C., Dohring, M., Mokrushin, L., & Fikouras, I. (2011). M2M service enablement for the enterprise. In *2011 15th International Conference* on *Intelligence in Next Generation Networks* (pp. 169–174). IEEE.

Vugrin, E. D., Warren, D. E., Ehlen, M. A., & Camphouse, R. C. (2010). A framework for assessing the resilience of infrastructure and economic systems. In *Sustainable and Resilient Critical Infrastructure Systems* (pp. 77–116). Berlin, Heidelberg: Springer.

Wang, X., Vasilakos, A., Chen, M., Liu, Y., & Taekyoung Kwon, T. (2012). A survey of green mobile networks: Opportunities and challenges. *Mobile Networks and Applications, 17*(1), 4–20. doi=10.1007/s11036-011-0316-4

Wash, R. (2010). Folk models of home computer security. In *Proceedings of the Sixth Symposium on Usable Privacy and Security* (pp.). New York, NY: ACM. doi:10.1145/1837110.1837125

Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., et al. (2009). Building the Internet of Things using RFID: The RFID ecosystem experience. *IEEE Internet Computing, 13*(3), 48–55.

Welsh, B., Baird, T., Zhao, J., & Block-Schachter, D. (2014). Web app design to implement travel behavioral nudging using “moves.” In *Transportation Research Board 93rd Annual Meeting* (pp. 14–4111).

Wiesmaier, A., Horsch, M., Braun, J., Kiefer, F., Hhnlein, D., Strenzke, F., & Buchmann, J. (2011). An efficient mobile PACE implementation. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (pp. 176–185). doi=10.1145/1966913.1966936

Wu, G., Talwar, S., Johnsson, K., Himayat, N., & Johnson, K. D. (2011). M2M: From mobile to embedded internet. *IEEE Communications Magazine, 49*(4), 36–43.

Wu, L., Du, X., & Fu, X. (2014). Security threats to mobile multimedia applications: Camera-based attacks on mobile phones. *IEEE Communications Magazine, 52*(3), 80–87.

Xie, L., Zhang, X., Seifert, J.-P., & Zhu, S. (2010). pBMDS: A behavior-based malware detection system for cellphone devices. In *Proceedings of the Third ACM Conference on Wireless Network Security* (pp. 37–48). doi=10.1145/1741866.1741874

Xu, H., Luo, X., Carroll, J., & Rosson, M. (2011). The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. *Decision Support Systems, 51,* 42–52.

Xu, N., Zhang, F., Luo, Y., Jia, W., Xuan, D., & Teng, J. (2009). Stealthy video capturer: A new video-based spyware in 3G smartphones. In *Proceedings of the Second ACM Conference on Wireless Network Security* (pp. 69–78). doi=10.1145/1514274.1514285

Xu, Y., Liu, X., Huang, Y., & Zhang, L. (2014). U.S. patent no. 8,713,320. Washington, DC: U.S. Patent and Trademark Office.

Yahya, S., Kamalrudin, M., Sidek, S., & Grundy, J. (2014). Capturing security requirements using essential use cases. In *Requirements Engineering* (pp. 16–30). Berlin, Heidelberg: Springer.

Yaoming, C. (2010). A smart gateway design for WSN health care system.

Yang, S. W., & Chen, Y. K. (2013). The M2M connectivity framework: Towards an IoT landscape. In *2013 IEEE and Internet of Things, IEEE International Conference on Green Computing and Communications and IEEE Cyber, Physical and Social Computing* (pp. 572–579). IEEE.

Zafar, N., Arnautovic, E., Diabat, A., & Svetinovic, D. (2014). System security requirements analysis: A smart grid case study. *Systems Engineering, 17*(1), 77–88.

Zhang, Y., Yu, R., Xie, S., Yao, W., Xiao, Y., & Guizani, M. (2011). Home M2M networks: Architectures, standards, and QoS improvement. *IEEE Communications Magazine*, *49*(4), 44–52.