

Combating Cyber Threats on Resource-Constrained IoT Devices: Attack Observations and Stealth Security

資源制約のある IoT 機器におけるサイバー脅威への対策：
攻撃の観測とステルスセキュリティ

By

Aamir H. BOKHARI

September, 2021

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Information Science
to



the Graduate School of Environment and Information Sciences,
Yokohama National University
under the academic supervision of
Professor Tsutomu MATSUMOTO

DEDICATION

I am deeply indebted to my wife and family for encouraging and supporting me during my PhD. I could not have completed this challenge without their inspiration, specially my parents who taught me the value of learning and invigorated me in every aspect of my life.

ACKNOWLEDGEMENTS

The author wishes to record his great indebtedness to Professor Tsutomu Matsumoto under whose supervision this work was carried out, for his insight and encouragement. The author would also like to say special thanks to the co-supervisor, Associate Professor Katsunari Yoshioka, for his kind guidance and valuable comments throughout this research work. This research could not have been completed without the continued support and supervision of my advisors.

The author sincerely acknowledges the review committee members, Professor Matsumoto, Professor Mori, Professor Shikata, Associate Professor Yoshioka, and Associate Professor Shirakawa, for their time and constructive comments that helped in further improving the quality of my dissertation.

The author is also grateful to all the members (past and present) of the Matsumoto Laboratory and Yoshioka Laboratory for their help. Furthermore, the author is also very thankful for the kind support from the secretaries of the labs, Ms. Mio Narimatsu, Ms. Tomoko Ishidate, and Ms. Emiko Kawamura.

ABSTRACT

The digital boom brought empowerment to seamless connectivity by enabling manufacturers to harness the power of the Internet into their products, opening up the world of the Internet of Things (IoT). We are now seeing explosive growth in the IoT market. With the increase in applications designed for mobility, the sensor networks are providing extended coverage for smart use cases, such as in vehicles, wearables, portable medical devices, remote surveillance in farming, goods tracking, and environment monitoring using IP cameras. These smart uses are taking advantage of innovations in the cellular infrastructure via 3G/LTE/5G connectivity, paving the way for cellular IoT. However, such advancements have also brought the side effect of misuse by unscrupulous agents, who scan open ports for services and exploit security weaknesses or vulnerabilities in the system. Due to mobility needs and remote uses, such IoT devices are often small and lack the availability of sufficient resources. Factors such as weaker security, resource constraints, lack of user awareness, and manufacturing priorities have made them an easy target. As a result, cyber-attacks are increasing day by day on such IoT devices.

Cyber peeping (unauthorized access of an IP camera) is one such growing issue that affects many due to privacy concerns. The lack of user awareness and insufficient technical know-how also increases the need for remote management and security patching capability on such resource-constraint IoT devices. But, such capabilities are also under attack as they can provide a direct access to the control of an IoT device. A famous case of the Mirai botnet attack in 2016 has highlighted many security weaknesses. It exploited the above factors for gaining control of millions of IoT devices globally and

then using them for attack purposes. Therefore, in this research we first aim to increase security awareness by examining the dangers associated with IP cameras, as the impact would be significant due to the frequent use of IP cameras in our daily lives. Then, we look at a possible countermeasure that can address the peeping issue (unauthorized access) and the challenges faced by resource-constraint IoT devices.

IP cameras are one of the most widely used devices among the Internet of Things. Existing research on information security for IP cameras has primarily focused on authentication or malware issues, but not on the peeping method itself. How cyber peeping occurs in the real world can further help in strengthening defenses accordingly and spread more awareness about the dangers of IP cameras. For this part of the research, a honeypot was used to observe unauthorized access (peeps) using decoy cameras in two scenarios. The first scenario provided the situation where humans can read background information (handwritten URL and ID/password bait). The second scenario simulated a living room in a home environment. Our experiment resulted in confirming many examples of peeping into the decoy cameras in reality. Besides various attack patterns observed, we also confirmed that intelligence gathering helps an attacker with his attack angle. The results of this study were used in several TV programs of a national broadcasting station to highlight the risks associated with the use of IP cameras. The study results were also directly shared with IP camera vendors, resulting in IP camera security-related improvements. This study published in the Journal of Information Processing (JIP) has also received an award. Therefore, we believe that this study can further help in improving the security and awareness of the dangers associated with IoT devices, such as IP cameras.

The first study showed that knowing what is running or available on the target device

makes it easier to exploit vulnerabilities. Stopping scanning attempts on exposed ports can create more obstacles for an attacker and can be used as a deterrent against unauthorized access. However, the possible countermeasure selection criteria must meet the security challenges due to the lack of management by the end-user, the challenges faced by the manufacturer, and the limited resources available on the device. Existing security options do not meet these requirements fully. To achieve this deterrence without affecting the provided service, we propose a unique but practical method of security called “Port Knocking.” It is a clever technique in which a port can remain hidden (closed) until receiving a predetermined set of knocks (packets) on random ports in a specific order. The device remains hidden from the internet as it does not respond to any query. Due to the nature of this technique, we have defined it as a stealth security feature. Therefore, in the second part of the study, we examined this stealth technology feature and its impact on the CPU and power consumption for securing resource-constraint IoT devices that are growing exponentially. As proof of concept, we have tested it for remote management with the SSH service running on TCP port 22. By enabling secure remote operations and management of an IoT device with this security technique, we can provide an additional security layer and ensure timely patching of security vulnerabilities safely and stealthily. Our experimental results on a resource-constraint IoT device show that port knocking not only secures the device and provides a secure remote management option but also helps in keeping its power consumption low. The results obtained make it an effective security layer for securing resource-constraint IoT devices.

Hence, we were able to achieve the following from our research:

- i. First observational study of IP camera peeping using real IP cameras in various environments that demonstrated detailed

unauthorized access patterns on the insecure IP cameras, including the existence of automated accesses specifically tailored to find IP cameras efficiently. Intelligence gathering was the key to exploiting security weaknesses.

- ii. The research results were published and shared with device manufacturers, broadcasted by national TV, and recognized by a peer-reviewed journal in the form of an award – all show the contribution of this research in increasing the security awareness.
- iii. The empirical study on the proposed stealthy countermeasure revealed that besides blocking all unauthorized access attempts, the proposed stealth security technique could also provide power savings with minimal impact on the CPU consumption.

The lessons learned from the peeping attack (unauthorized access) observations and implementing the proposed countermeasure will enhance the security of the IP cameras and provide the additional benefit of power savings. Furthermore, it can benefit other resource-constraint IoT devices as well.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
TABLE OF CONTENTS	viii
LIST OF FIGURES.....	xi
LIST OF TABLES.....	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Research Questions	3
1.2 Scope and Objectives	5
1.3 Research Methodology.....	5
1.4 Contribution of the Present Work	7
1.5 Dissertation Outline.....	9
CHAPTER 2: LITERATURE REVIEW	12
2.1 Internet of Things (IoT).....	12
2.2 Issues with IP Cameras	13
2.3 Issues with Resource-constrained IoT Devices.....	14
2.4 Summary	18
CHAPTER 3: MAIN OBJECTIVES	19
3.1 Introduction	19
3.2 Research Motivation	21
3.3 Methodology	22
3.4 Research Analysis Flow	24

3.5	Benefits	25
3.6	Summary	26
CHAPTER 4: ATTACK OBSERVATIONS.....		27
4.1	Introduction	27
4.2	Basics of IP Camera	30
4.3	Known Issues with IP Cameras.....	30
4.4	Observational Experiment.....	31
4.4.1	Environment Setup.....	31
4.4.2	Experiment Overview	32
4.5	Analysis and Results	35
4.5.1	Observation Results from URL-reflected Decoy Cameras	36
4.5.2	Observation Results from Living-room Decoy Cameras	38
4.5.3	Peeping Characteristics	39
4.6	Discussion	45
4.6.1	Limitations and Ethical Considerations	47
4.7	Summary	48
CHAPTER 5: COUNTERMEASURES – STEALTH SECURITY.....		49
5.1	Introduction	49
5.2	Perimeter Defense Options.....	50
5.3	Challenges	54
5.4	Stealth Security	56
5.5	Empirical Analysis	59

5.5.1	Test Environment Setup	60
5.5.2	Analysis and Results	63
5.6	Discussion	71
5.6.1	Limitations and Considerations.....	72
5.7	Summary	73
CHAPTER 6: CONCLUSIONS		74
6.1	Concluding Remarks.....	74
6.2	Future Directions.....	76
BIBLIOGRAPHY.....		78
PUBLICATIONS.....		86

LIST OF FIGURES

Figure 1 - Dissertation outline	11
Figure 2 - General IP cameras	20
Figure 3 - IP camera honeypot.	31
Figure 4 - Image from the URL-reflected camera A.	33
Figure 5 - Image from the URL-reflected camera B.	33
Figure 6 - Image from the living-room decoy camera.	34
Figure 7 - Trend of hosts accessing the URL-reflected decoy cameras.	37
Figure 8 - Image acquisition request to camera A using a browser.	39
Figure 9 - Automatic image acquisition request to camera A.	40
Figure 10 - Request for an automatic search of multiple types of cameras.....	40
Figure 11 - Part of the long-term access to camera A.	41
Figure 12 - Image acquisition request to camera C using a browser.....	42
Figure 13 - Automatic image acquisition request to camera C.	42
Figure 14 - Part of the group of accesses appears to be from the same person.	43
Figure 15 - Part of the peeping access using a vulnerability.	44
Figure 16 - Part of the port number change request.	45
Figure 17 - Port knocking mechanism.....	57
Figure 18 - PRNG-CRNG based port knocking.....	58
Figure 19 - Stream-cipher-based port knocking.	59
Figure 20 - Lab experiment.	60
Figure 21 - Test-1 setup.....	61
Figure 22 - Test-2 setup.....	62
Figure 23 - Test-3 setup.....	62

Figure 24 - Test-1 results.....	63
Figure 25 - Voltage stability results.	65
Figure 26 - Power consumption without port knocking on both test devices.	66
Figure 27 - Power consumption and packets received <u>with</u> port knocking (<u>first</u> week).	67
Figure 28 - Power consumption and packets received <u>without</u> port knocking (<u>first</u> week).	68
Figure 29 - Power consumption and packets received <u>with</u> port knocking (<u>later</u> on).	69
Figure 30 - Power consumption and packets received <u>without</u> port knocking (<u>later</u> on).	70

LIST OF TABLES

Table 1 – Observation experiment with the URL-reflected decoy cameras.....	32
Table 2 – Observation experiment with the living-room decoy cameras.	35
Table 3 – Observation results with the URL-reflected decoy cameras.	36
Table 4 – Access analysis of peeping hosts in the URL-reflected decoy cameras.	38
Table 5 – Observation results with the living-room decoy cameras.....	39
Table 6 – Comparison of possible perimeter defense options.	52
Table 7 – IoT test devices.....	60
Table 8 – Port knocking security effectiveness (6 weeks test results).....	64

CHAPTER 1: INTRODUCTION

In recent years, more and more various types of devices are getting connected to the Internet. A common term has been coined to call all such devices as “Internet of Things” (IoT). Due to the technological advancements in wireless connectivity, there is a sharp increase in the number of IoT devices, applications, and uses. Such IoT devices offering various services are attracting a lot of attention. With such attention comes the dangers of being on the internet. The 2016 Mirai botnet was an eye-opener for the IoT industry. Mirai malware recruited millions of IoT devices (DVRs, IP cameras, wireless routers, etc.) connected to the Internet having default/no passwords or weaker security due to unpatched IoT. These compromised IoT devices were then used to launch a distributed denial of service (DDoS) attack on one of the major domain name service (DNS) providers that rendered many popular websites inaccessible for hours [1], [2], [3]. One of the security vendors, Norton expects more sophisticated DDoS attacks as more and more IoT devices get connected to the Internet, fueled by cellular technological developments [4].

Among the thriving IoT market, digital cameras that allow remote viewing and operations over the Internet can be collectively referred to as IP cameras. Besides personal use, they are being used mainly for surveillance, such as remote monitoring, flood monitoring, road conditions, perimeter security, drones, office security, etc. According to the paper [5] in the global marketing insight report, the market size of the IP cameras is expected to grow from more than 8 billion US dollars in 2018 to over 20 billion US dollars by 2025. This sharp rise of 14% compound annual growth rate (CAGR) is attributed to IoT and technological advancements in digital cameras. Due to the nature of IP cameras,

the lack of user awareness about security risks, and the issues related to authentication and vulnerabilities, they are often targeted by a third party to electronically access them without the knowledge and permission of the IP camera owner. Such unauthorized peeping actions by a third person via the Internet can be referred to as cyber-peeping. The misuse of free information posted on websites [6], [7] about accessible streaming video seems to fuel the cyber-peeping appetite, further raising concerns about the security and privacy of the IP cameras. So far, most of the research on unauthorized access to IP cameras has been focused mainly on authentication, changing camera configuration information, and observing malware infections that exploit vulnerabilities [8], [9], [10], [11], [12], [13], [14], [15]. However, there has been no investigation into the actual state of cyber peeping.

The explosive growth of the IoT market is also widening the attack surface, providing more opportunities to exploit IoT vulnerabilities. According to the research done by Schneier [16], the end-users often do not have access to the operating system of the IP camera or an IoT product that they are using. They also do not have adequate technical knowledge to even patch security vulnerabilities without the support of device manufacturer. On the other hand, the device manufacturers are more driven by the competition in the market and the race to put new products first in the market. Security of their products is often compromised as they only see revenue by who captures the market first with user-attractive features, which generally do not include security due to lack of security awareness and knowledge on the buyer side. Few brand conscious manufacturers only pay attention to security vulnerabilities out of the fear of brand damage [17]. Therefore, the products need to be secured at the point of entry, i.e., network level in order to avoid post-production security lapses. The innovations in cellular

connectivity have further opened up the doors for various IoT use cases, all the way from smartwatches to smart vehicles. The resource constraints such as power consumption become challenging when the smart uses require mobility and coverage range. For example, use in agriculture, healthcare, logistics, etc. Conventional security methods, such as firewall or VPN are resource-consuming and do not help with power consumption. The other option is Intrusion Detection System (IDS) or Intrusion Protection System (IPS), but they often have false positives, need signature updates, and can have post-sales maintenance challenges. Cloud-based security is an effective option, but it requires dedication and investment on the manufacturer-side. Most cloud breaches are a result of poor configuration or technical skills. The available options are either good for the enterprise level or with the computing devices that have a lot of resources available. Also, such perimeter defenses usually require technical knowledge to configure and maintain those high-end computing devices with appropriate security rules and patches. In short, the end-users do not have the adequate technical knowledge to fix security issues by themselves, and device manufacturers seem to be more focused on the market than security. The remote management means are also often compromised as they provide a direct path to the management of the device. Therefore, not every security countermeasure can be implemented, and due diligence needs to be exercised very carefully to ensure that the security options do not obstruct the intended use of the IoT device. Hence, the lack of security management and limited resources on IoT devices poses a big challenge.

1.1 Research Questions

In the viewpoint of the above issues, challenges, and lessons learned from existing

research, as well as the lack of security awareness exposed by the increasing cyber-attacks on IoT devices, it is significant to investigate:

1. How can we increase the importance and awareness about the security and privacy among the end-users and manufacturers of IoT devices?
2. To what extent can we overcome the challenges imposed by the limited-resources of IoT devices designed for mobility and coverage with cellular connectivity, in the following context:
 - (a) How much computing power (CPU consumption) would a security feature add to a resource-constraint IoT device?
 - (b) How effective this security feature would be in blocking unwanted access to an essential service when the IoT device is exposed to the Internet directly without other security layers or a firewall?
 - (c) What would be the impact of adding this security feature on the power consumption of the IoT device?

Hence, knowing the peeping techniques can help with further understanding the risks associated with IP cameras and improve awareness among the general public and IP camera vendors from a security and privacy point of view. Furthermore, existing research is missing an important piece of information that can highly impact the intended use of an already resource-constraint IoT device. We must also examine the impact of a security feature on CPU usage and power consumption to ensure that the additional security features do not impact resource-constraint IoT devices negatively. These elements are vital as IoT devices footprint is becoming smaller and smaller along with the diversity in usage.

1.2 Scope and Objectives

To address the above questions and find reasonable answers, we have narrowed our scope to focus on one of the popular IoT devices, i.e., IP cameras. In today's modern society, the use of digital cameras has become a norm and it touches almost every age group. Besides smartphones, the availability of digital cameras with internet connectivity has further broadened the uses, both inside the buildings and outside in the field. For example, nanny cams, pet monitoring, security monitoring, flood monitoring, drones, agriculture, etc. Therefore, by studying the cyber-peeping attacks and sharing the findings, we can divert the attention of all users to the dangers of IP cameras concerning security and privacy. At the same time, we can also provide insight to the IP camera manufacturers about the attack patterns so that they can understand the importance and improve the security of their products.

Furthermore, in this research we have decided to focus on the IoT use cases that involves mobility and greater coverage range, such as flood monitoring, asset monitoring, smart farming, etc. The commonality among these use cases is the resource-constraints due to limited battery sources and connectivity via a cellular connection. Such uses raise not only the power consumption challenges, but also the lack of user technical knowledge and remote management challenges. The point of entry connecting an IP camera or any other IoT device to the internet is the communication module or an IoT gateway. Therefore, we will focus on securing it in such a manner that it can satisfy our objectives.

1.3 Research Methodology

In order to find answer to our first research question in this dissertation, an

observational analysis method was used for increasing user awareness regarding security and privacy risks by exposing cyber-peeping attacks. In order to investigate and analyze the actual situation of cyber peeping in the real world, an IP camera was set up as a honeypot (hereinafter referred to as the “decoy camera”) for conducting an observational experiment. Two types of observation environments were established for the experiment purposes. In the first observation experiment, we prepared two cameras (hereinafter referred to as the “URL reflection type decoy camera”), that displayed a hand-written note for a specific URL and two sets of different ID/passwords (one for each decoy camera) for access confirmation purposes, and assigned 10 IP addresses to each camera for observations. The objective was to study an access by a human element, who can read the background information (reflected URL and ID/password) via peeping into the decoy camera and then use that information to successfully gain access to the reflected URL. On the URL side, we examined what ID/password was entered for determining if humans were attempting access after viewing the video of the decoy camera. Although the first observational study helped in determining human element involvement, it was limited due to the fact that such a set up did not provide continuous peeping interest as the decoy camera was only showing a URL and its related ID/password. Therefore, a second observation experiment was setup where a room was prepared for observation that simulated a living-room of a home in which movement can be expected. Five decoy cameras (hereinafter referred to as the "living-room decoy camera") were installed to show the video of this room. These living-room decoy cameras were then exposed to Internet so that peeping methods could be examined in detail for a longer time. The objective was to attract the peeping entity to engage in more than one peeping event by providing a real life scenario. The details of testing and analysis are provided in Section

4.5.

In order to find answer to our second research question, we narrowed our scope to focus on a popular raspberry-pi-based IoT gateway device with cellular connectivity, such as that used with IoT sensors in smart cities, smart bicycles, goods tracking, flood monitoring, agriculture monitoring, medical monitoring, etc. Such use cases need cellular IoT in order to ensure mobility and coverage for their intended use. An IoT device was used to test two types of port knocking methods. The first method was based on the python script applying the pseudo-random number generator (PRNG) and the chaotic random number generator (CRNG) algorithms [18]. The second method was also based on the python script, but applying a stream cipher using the Authenticated Encryption with Associated Data (AEAD) algorithms [19]. Due diligence was exercised to ensure we do not have any external influence when measuring the impact on CPU usage and power consumption. Two identical devices were used for testing and comparison purposes. One with the stealth security feature of port knocking enabled and running, whereas the other was without it. Both were then connected to the internet. We initially experimented for a short period (2 weeks) to obtain preliminary results and then later on tested over a longer period (7 weeks) to verify the results. The details of testing and analysis are provided in Section 5.6.

1.4 Contribution of the Present Work

The results of the observational study, empirical analysis, and contributions of this dissertation are briefly summarized below:

As the result of observational study conducted on cyber-peeping attacks, we were able to achieve the following:

- A) First observational study of IP camera peeping using real IP cameras in various environments.
- B) Demonstrated that a peeping problem does exist with a high degree when IP cameras are insecure.
- C) Revealed detailed access patterns on the insecure cameras, including the existence of automated accesses specifically tailored to find IP cameras efficiently.

Moreover, we had conversations with several IP camera vendors and explained the real world risks of insecure IP cameras. One of the vendors now deploy a security mechanism in their IP camera products that enforces users to set their own unique password. Through the publication of the research findings and sharing of the results publically over the NHK national broadcasting channel, we believe that this research has helped in spreading the awareness on security and privacy risks associated with IP cameras. Furthermore, the “JIP Specially Selected Paper Certificate” from the Journal of Information Processing is an endorsement for this research work.

Similarly, through the empirical analysis of the port knocking stealth security feature, we were able to achieve the following:

- D) The CPU consumption with the implementation of the stealth port knocking feature using the stream cipher with AEAD algorithm would only add a maximum overhead of 15% of the CPU power during the key-stream generation, and a maximum of 9% overhead during the knock-sequence update.
- E) The security effectiveness test (by exposing directly to the Internet via 3G without any other security layers) showed that the IoT device with

the stealth security feature protecting the SSH service running on the default port was able to block all unwanted accesses for 42 days, while the unprotected services received 431,142 requests from 5,424 hosts in those 42 days.

- F) The power consumption test showed that the IoT device with the stealth port knocking feature would actually decrease the power consumption compare to the one without such feature because of receiving a lesser number of packets due to the hidden service.

Hence, based on the results of the experiments conducted to measure the effect of using the stealth security feature of port knocking on the resource-constraint IoT device, we can conclude that the experimental results imply that the power consumption overhead by receiving incoming session requests (from scanners/malware on the Internet) without port knocking would easily exceed the power consumption for running port knocking service. Thus, running the stealth port knocking service would be beneficial in terms of not only the security enhancement but also the power consumption on a resource-constraint IoT device.

1.5 Dissertation Outline

This dissertation makes an effort to highlight the security awareness on IoT devices, using IP cameras as an example, and develop a possible security solution for the resource-constrained portable IoT devices. Apart from Chapter 1 (introduction providing the research question, scope, and objectives along with the motivations, contributions, and the organization of this dissertation), each chapter is devoted to one step closer towards

the understanding of combating common attacks on the resource-constrained IoT devices, while progressing ahead logically. A brief breakdown of the rest of the dissertation is as follows:

Chapter 2 – Literature Review

A review of the prominent literature in the study of combating cyber threats on resource-constrained IoT devices. Existing issues and countermeasure to combat with unauthorized accesses are presented in this chapter.

Chapter 3 – Main Objectives

In this chapter, we look at the main objectives of this dissertation. We examine current issues that drive the motivation to understand the problem and the steps taken to resolve them by proposing a possible solution through an observational study and empirical analysis.

Chapter 4 – Attack Observations

This chapter presents a detailed analysis of cyber-attack, such as cyber-peeping using decoy IP cameras as honeypot. Two observational experiments are explained that resulted in insights on motivations behind the human-driven and automation-driven attack patterns, adding to the security and awareness about the dangers of IP cameras.

Chapter 5 – Countermeasures

This chapter concentrates on a proposed stealth security solution that helps by hiding the ports of a resource-constrained IoT device and rendering it invisible to unauthorized attackers. This proposed solution provided power-savings to the IoT device in addition to the security.

Chapter 6 – Conclusions

The concluding chapter comprises a summary of the dissertation and concluding remarks regarding the observed attacks and proposed stealth solution for portable IoT devices. Furthermore, the key points of the future research work, an extension of the present research work, are also included in this chapter.

The outline of this dissertation is systematically shown in **Figure 1** below:

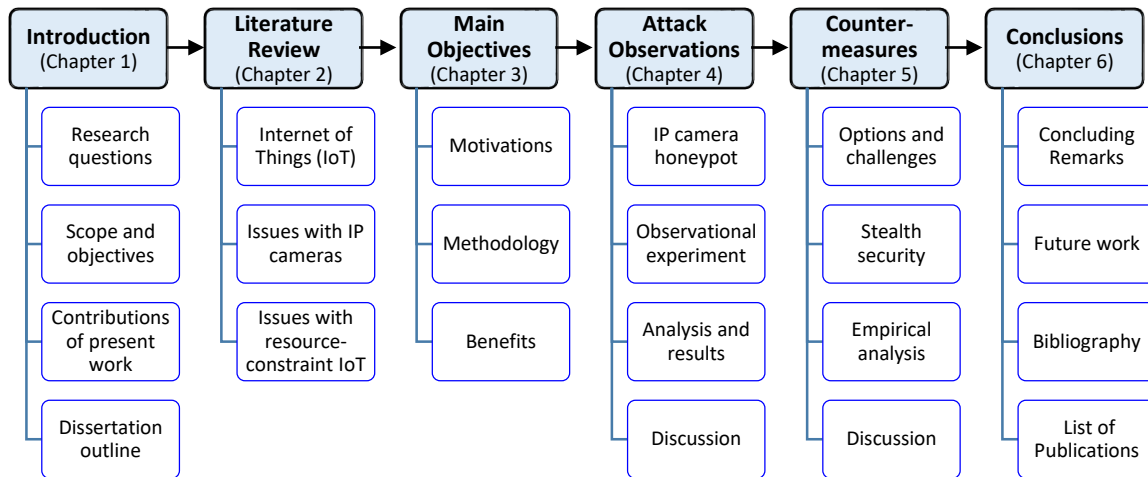


Figure 1 - Dissertation outline

CHAPTER 2: LITERATURE REVIEW

This chapter entails the literature review of previous research works related to the honeypots for observation of cyber-attacks and a stealth security technique called port knocking. In this chapter, a brief literature review is first presented to throw the light on the ongoing advancements in the field of Internet of Things from past to present. Then, we look at the work done for the observation of attacks using honeypots, which is then applied to build our own IP camera honeypot for the purpose of observing cyber peeping attacks. At the end, we examine the current literature on the issues with resource-constraint IoT devices and look at the port knocking feature from stealth security point of view.

2.1 Internet of Things (IoT)

Before the word “Internet of Things” was coined, the general perception was that the Internet connected devices are usually personal computers, laptops, tablets, and smartphones. However, the advancements in digital technology, cheaper processors, and expansion of wireless networks opened up the way to almost connect anything to the internet. IoT technology is now being used to provide smart services to its users where the devices can connect and communicate by themselves without any human interaction, thanks to the cloud and edge computing technologies, artificial intelligence, and machine learning [20]. The 2020 market update from IoT Analytics shows that IoT device connections has surpassed the non-IoT connection by the end of the year 2020, and it is projected to be over 30 billion IoT connections by 2025 [21]. The new norm is smart

devices, such as smartwatches, smart tracking, smart farming, smart bicycles, smart vehicles, smart pills, e-health, wearables, etc. These smart devices are being enabled by the advancements in the connectivity technology, such as cellular networks besides the local wireless and wired networks. 5G advancement in the cellular industry has further fueled the IoT growth. According to the Ericsson mobility report [22], the cellular IoT connections are expected to increase from 1.3 billion IoT connections in 2019 to 5 billion IoT connections by 2025. IoT devices with cellular connectivity are ideal for any use case that requires mobility such as in asset tracking, healthcare, logistics, farming, manufacturing, etc. One of the advantages of using the cellular connectivity is that the infrastructure is already in place and the IoT device only needs a mobile SIM to get connected to the internet, either indoors or outdoors. On the down side, these devices are often constrained by their power consumption and battery life [23]. Hence, resource-constraints are one of the biggest challenges in securing these IoT devices. According to a report on IoT attacks, the first-half of the year 2019 has seen 900% increase in attacks on the IoT devices compare to the first-half of the year 2018 [24]. Similarly, analysis done by Kaspersky using a honeypot showed malwares such as Mirai, Hajime, PNScan, BrickerBot, Gafgyt, etc. exploiting weaknesses in IoT devices due to default/no passwords, vulnerabilities in the firmware that was unpatched, and the open telnet/SSH ports. And, 63% of total attacks were targeted on DVR or IP cameras [25].

2.2 Issues with IP Cameras

IP cameras are one of the mostly used devices among Internet of Things (IoT). In 2018, FBI warned [26] about the dangers of IoT devices, including IP cameras, to be used as “proxies for anonymity and pursuit of malicious cyber activities”. Most of the research

primarily has been focused on vulnerabilities for exploitation of IP cameras. For example, in paper [14] the author has analyzed the security of cloud-based video cameras by focusing on vulnerabilities for exposing potential issues in IP cameras. Also, another paper [13] discusses about the security of smart homes having IP cameras as well, but examines it from systems design and security vulnerabilities point-of-view. In Another article [15] talks about peeping by exploiting default passwords, but not the peeping phenomenon itself. Similarly, server-type honeypots that monitor remote exploit attacks and collect malware specimens, have been researched as honeypots for observing attacks on web services by using general purpose responses to services [27], [28]. In addition, IoT POT [8], [9], [10], [11], Uberpot [29], and SIPHON [12], which are honeypots simulating IoT devices, have been proposed. However, in papers [8], [10], [11], [12], [13], [14], [15], [27], [28] no detailed investigation has been conducted on the peeping technique for acquiring the IP camera video or images. Similarly, in the research paper [9], a successful cyber-attack by an attacker (who viewed the video through the camera that displayed or reflected an ID/password) was confirmed, but the actual method of peeping was not analyzed. Therefore, in this research study, in order to investigate this fact, we experimented with a honeypot of IP cameras (decoy cameras) to observe the peep and analyze the detailed method.

2.3 Issues with Resource-constrained IoT Devices

IP cameras that are connected to a Wi-Fi or wired network can be protected by enabling strong security on the routers that are connecting them to the internet. There are a number of existing security solutions that are designed for perimeter defense. For example, firewalls, Intrusion Detection and Prevention Systems (IDS/IPS),

blacklisting/whitelisting, IP table rules, Virtual Private Networks (VPN), data encryption, etc. The conventional security methods, such as firewall or white-listing require post-sales maintenance of the device to manage policies. VPN slows down the connectivity and need a separate connection for each user. The other option is Intrusion Detection System (IDS) or Intrusion Protection System (IPS), but they often have false positives, need signature updates, and can have post-sales maintenance challenges. Cloud-based security is an effective option, but it requires dedication and investment on the manufacturer-side. Most cloud breaches are a result of poor configuration or technical skills. Therefore, most of existing options are often used at the enterprise level or among the computing devices that have a lot more resources available than the resource-constrained IoT devices. Also, such perimeter defenses usually require technical knowledge to configure and maintain the devices with appropriate security rules and patches. For IoT devices, these security solutions are not well-suited as many layers of security will be need to protect IoT devices, considering available resources and power consumption. In short, the end-users do not have the adequate technical knowledge to fix security issues by themselves, and device manufacturers seem to be more focused on the market than security. Moreover, the remote management means are also often compromised as they provide a direct path to the management of the device. In a recent study by Comparitech researcher using honeypots [31], the intensity of attacks on unprotected devices connected to the internet has increased around 45 times more compared to 2007, and over 70% of attacks were on SSH (port 22) in 24 hours.

Therefore, not every security countermeasure can be implemented, and due diligence needs to be exercised very carefully to ensure that the security options do not obstruct the intended use of the IoT device. Hence, the lack of security management and limited

resources on IoT devices poses a big challenge.

On the other hand, with the advancement of portable IoT the need for mobility and remote connectivity have superseded conventional means of connectivity to the internet. Therefore, applications such as in smart cities, smart bicycles, wearable devices, portable medical devices, agriculture monitoring, flood monitoring, goods tracking, etc. require cellular connectivity in order to ensure mobility and coverage for their intended use. Due to the lack of user knowledge on security, measures that require security configuration changes, security rule settings, or security patching cannot be left upon the users of the IoT devices [30]. That means the available resources, power consumption, and remote management will be an important factor in order to ensure data security on such portable cellular IoT devices.

Research papers [32], [33], [34], [35], [36] talk about a stealthy method of port knocking that is commonly used by the system administrators of large systems for avoiding attacks on the remote management services. With the increase in attacks on IoT devices, the focus has turned towards the port knocking feature and finding the kind of algorithms that can be used in the Internet of Things environment. Due to the popularity of IoT, a lot of research has been directed towards the IoT threats, vulnerabilities, attacks, limitations, authentication, and challenges [1], [19], [37], [38], [39], [40]. Many researchers have also examined the security of IoT and possible countermeasures. For example, in a survey paper [37], the authors examined the issues with IoT devices and various types of IoT attacks along with existing possible countermeasures. As a conclusion, they have ruled out the use of conventional cryptography in small IoT devices or limited its use due to resource constraints. The paper [39] provided a port knocking approach utilizing symmetric key encryption, Message Authentication Code (MAC), and

an encrypted keep-alive system to secure the service port. However, it is limited to TCP ports with static IP configuration only and was tested using a hardware (CPU = Intel Core i7 3770 @ 3.40 GHz; RAM = 8GB DDR3) having plenty of resources. With respect to resources, only physical memory usage and network bandwidth were examined. This paper did not look at the device power consumption, which is more critical in today's resource-constraint IoT devices. In another paper [40], the author has introduced port knocking based on digital certificates for strengthening the authentication between the IoT devices. Another paper [41] proposed an architecture of SSH honeypot based on port knocking and intrusion detection systems for protecting a server, but this study was not focused on the resource constraint IoT devices. The focus was rather on honeypots and attacks on SSH service. However, this paper strengthens the idea that port knocking can be used for security purposes.

According to another paper [42] on security intelligence for the Industry 4.0 revolution, the demand for more direct connectivity with the Internet will also open doors for major security management issues. The author has discussed possible mitigation techniques for raspberry-pi-based IoT devices using port knocking with two-factor authentication as part of their solution. However, in this study the port knocking was implemented on a router and the IoT devices were behind the firewall. In paper [18], the author has introduced an advanced method of port knocking that can reduce attacks by producing difficult-to-guess port knocking sequences based on PRNG and CRNG algorithms. As this solution utilizes both TCP and UDP ports, therefore, it can be a candidate for our purposes. Similarly, another paper [19] suggests a different algorithm based on RFC7539 [43] that pairs the ChaCha20 stream cipher with the Poly1305 authenticator to create an AEAD scheme for use in the TLS protocol for high-speed and

lightweight IoT applications. A security analysis report by KDDI Research Inc. concludes that they could not find a weakness in the AEAD algorithm [44]. Therefore, this method can also be a candidate for our purpose as it can be used for creating random port knocking sequences and key generation.

2.4 Summary

In this chapter, we have provided a brief overview of the existing research in order to understand what work has previously been done and where there is room for improvement. Based on this, we will set our research objectives in the next chapter to understand the current challenges and try to overcome them with proposed solutions. Also, this will help us understand the benefits our research would bring to the research community.

CHAPTER 3: MAIN OBJECTIVES

This chapter focuses on the motivation and methodology of the research done for this dissertation. Our motivation is driven by the challenges raised by the ever increasing cyber-attacks on the resource-constraint IoT devices and a common pattern of lack of security awareness on the user-side. In order to answer our two research questions, we have applied an observational study on cyber-peeping attacks for understanding the dangers associated with security and privacy of IP cameras, as they are almost everywhere in our daily lives. Then, we looked into possible countermeasures for protection from cyber-attacks by analyzing a stealth security technique, called port knocking, which hides the service ports from the internet and opens them for authorized users on receiving a specific sequence of packets (knocks) on various closed ports. Through empirical analysis, we determined its effectiveness against cyber-attacks and resources, such as CPU utilization and power consumption.

3.1 Introduction

The 2016 Mirai botnet malware attack is another example that caused havoc by compromising millions of IoT devices having default/no passwords and unpatched/weaker security. There are still many variants of Mirai malware being used even today. Passwords, telnet, and SSH are one of the main problems of IoT devices as they are open and weak. Among the devices under attack, more than 63% were either digital video recording services or IP cameras [25].

An IP camera is usually connected to the internet via a gateway/router using wired

or wireless connectivity (Wi-Fi, Bluetooth, 3G/LTE/5G, etc.) as shown in **Figure 2**. An IP camera is also an IoT device and IoT devices are often connected to the internet using one of these connectivity options. Due to visual images, IP cameras generate lot of data that add to the consumption of available resources such as power and bandwidth in remote use cases.

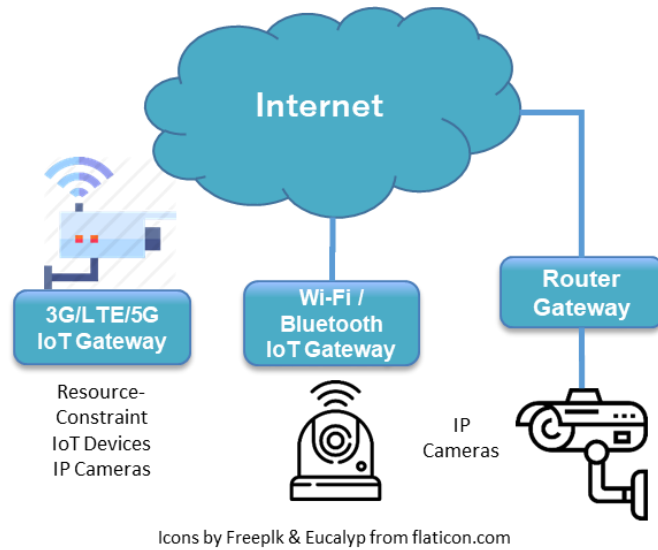


Figure 2 - General IP cameras.

So far, most of the research on unauthorized access to IP cameras has been focused mainly on authentication, changing camera configuration information, and observing malware infections that exploit vulnerabilities [8], [9], [10], [11], [12], [13], [14], [15]. However, there has been no investigation into the actual state of cyber peeping.

Similarly, the digital boom in IoT uses, being fueled by the technological advancements in wireless connectivity, has broadened the attack surface [45]. Attackers are taking advantage of weaker or no security on resource-constraint IoT devices due to challenges with limited resources, battery operated devices, and lack of user knowledge [29], [30], [37]. The end-users do not have the adequate technical knowledge to fix

security issues by themselves, and device manufacturers seem to be more focused on the market [16]. Therefore, there is an increasing need to enable IoT devices to be fully patched and secured, but such methods are often under attack as well.

3.2 Research Motivation

In today's modern society, the use of digital cameras has become a norm and it touches almost every age group. Besides the smartphones, the availability of digital cameras with internet connectivity (IP camera) has further broadened the uses, both inside the homes and outside in the field. For example, nanny cams, pet monitoring, flood monitoring, security monitoring, etc. Therefore, we believe that by studying the cyber-peeping attacks we can divert the attention of all users to the dangers of the IP cameras with respect to security and privacy, and at the same time can also provide insight to the IP camera manufacturers about the attack patterns so that they can understand and improve the security of their products.

An IP camera depicts common IoT devices that are exponentially connected to the Internet. When it comes to countermeasures against cyber-attacks on resource-constrained IoT devices, conventional security methods are not so practical to use in such IoT devices due to the limitation of available resources, power consumed, and the lack of user knowledge or manufacturer priority on security. In order to address these challenges, we have focused on those IoT (IP camera) use cases that involves mobility and greater coverage range, such as flood monitoring, asset monitoring, smart farming, drones, etc. The commonality among these use cases is the resource-constraints due to limited battery sources and connectivity via a cellular connection. Such uses increase not only the power consumption challenge, but also the lack of user technical knowledge and remote

management challenges as well. Therefore, a lightweight security solution is required for such IoT applications.

3.3 Methodology

In order to find answer to our first research question in this dissertation, an observational analysis method was used for increasing user awareness regarding security and privacy risks by exposing cyber-peeping attacks. In order to investigate and analyze the actual situation of cyber peeping in the real world, an IP camera was set up as a honeypot (hereinafter referred to as the “decoy camera”) for conducting an observational experiment.

Two types of observation environments were established for the observational experiment purposes. In the first observation experiment, we prepared two cameras (hereinafter referred to as the “URL reflection type decoy camera”), that displayed a handwritten note for a specific URL and two sets of different ID/passwords (one for each decoy camera) for access confirmation purposes, and assigned 10 IP addresses to each camera for observations. The objective was to study an access by a human element, who can read the background information (reflected URL and ID/password) via peeping into the decoy camera and then use that information to successfully gain access to the reflected URL. On the URL side, we examined what ID/password was entered for determining if humans were attempting access after viewing the video of the decoy camera. Although the first observational study helped in determining human element involvement, it was limited due to the fact that such a set up did not provide continuous peeping interest as the decoy camera was only showing a URL and its related ID/password. Therefore, a second observation experiment was setup where a room was prepared for observation that

simulated a living-room of a home in which movement can be expected. Five decoy cameras (hereinafter referred to as the "living-room decoy camera") were installed to show the video of this room. These living-room decoy cameras were then exposed to Internet so that peeping methods could be examined in detail for a longer time. The objective was to attract the peeping entity to engage in more than one peeping event by providing a real life scenario. The details of testing and analysis are provided in Section 4.5.

In order to find answer to our second research question, we narrowed our scope to focus on a popular raspberry-pi-based IoT gateway device with cellular connectivity, such as that used with IoT sensors in smart cities, smart bicycles, goods tracking, flood monitoring, agriculture monitoring, medical monitoring, etc. Such use cases need cellular IoT in order to ensure mobility and coverage for their intended use. Since the entry point that connects an IoT device to the internet is the IoT gateway, therefore, we will examine the stealth security port knocking technique for strengthening the securing in such a manner that it can satisfy our research question. An IoT device was used to test two types of port knocking methods. The first method was based on the python script applying the pseudo-random number generator (PRNG) and the chaotic random number generator (CRNG) algorithms [18]. The second method was also based on the python script, but applying a stream cipher using the Authenticated Encryption with Associated Data (AEAD) algorithms [19]. Due diligence was exercised to ensure we do not have any external influence when measuring the impact on CPU usage and power consumption. Two identical devices were used for testing and comparison purposes. One with the stealth security feature of port knocking enabled and running, whereas the other was without it. Both were then connected to the internet. We initially experimented for a short

period (2 weeks) to obtain preliminary results and then later on tested over a longer period (7 weeks) to verify the results. The details of testing and analysis are provided in Section 5.6.

3.4 Research Analysis Flow

In the first study, the focus was on understanding how peeping (unauthorized access) to an IP camera occurs. In addition to observing various techniques of peeping using human-based and machine-based attacks and publishing results for increasing awareness about the dangers of IP camera, it was also validated that intelligence gathering helps an attacker to select his attack vector. Knowing what kind of services are available on the target device make it easier to exploit vulnerabilities. The attackers mostly launch their attack by scanning ports to know what kind of services are running on the targeted device.

Based on the learning from the first study, we then focused on finding an effective countermeasure. Instead of finding a countermeasure for each attack pattern of each type of the IP camera, it is more beneficial and effective to find a countermeasure that hinders the attacker's attempt for collecting the intelligence in the first place, i.e., using principle of denial. There may be other ways the attacker can still find an angle for an attack, but creating more obstacles to obstruct the attack approach can add to the overall security deterrence.

Fixed IP camera or IoT use cases are common and have the advantage of not facing limitations due to lack of available resources. There are plenty of available security solutions for them. However, due to mobility needs and remote uses such as flood monitoring or surveillance, such IP cameras (IoT devices) are often small and lacks availability of sufficient resources. The point of entry connecting a remote IP camera to

the internet is its communication module or usually an IoT gateway. Therefore, in the second study, we focused on the security deterrence for such resource-constrained IoT devices. If we can make the IoT gateway module/device to respond to authorized requests only, then the intelligence gathering behavior observed in the first study can be obstructed. And, by focusing on the entry-point to the internet, we can add to the security of not only an IP camera but also any other sensor connected behind the IoT gateway device.

Hence, the proposed countermeasure using port knocking stealth technique helps in achieving the above objective of obstructing the intelligence collection by the attackers for exploiting the IP camera or any other service provided by an IoT device. By closing the service port, the attacker will have difficulty in finding what is out there and as a result will not be able to launch the targeted attacks that we observed in the first study. This countermeasure with port knocking stealth technique also provides additional merit of power-saving, which is critical for resource-constrained IoT devices.

3.5 Benefits

- Overall, this research benefits security challenges faced by end-users and resource-constrained IoT device manufacturers.
- Observational study on the dangers of IP camera:
 - Security and privacy awareness.
 - Strengthening of security by the vendor.
- Empirical study on the countermeasure for cyber-attacks using the stealth security port knocking technique:

- A possible reduction in cyber-attacks due to stealth security feature (invisibility) as attackers need to know what ports are open and what services are running.
- CPU resource consumption can be reduced with the significant reduction in the number of incoming packets/traffic when ports are closed (hidden) as the device will not respond to scans or requests.
- This stealth security port knocking technique can also be an effective additional security layer.
- Less CPU consumption can lead to power-savings for the resource-constraint IoT devices.
- Less dependency on the users for security management.
- Less overhead for the IoT device manufacturers.

3.6 Summary

In this chapter we looked at the issues highlighted in the previous chapter 2 (literature review) and highlighted the motivations of this research for handling those issues. Our methodology is based on observational analysis of the cyber-peeping attacks and empirical analysis for finding a reasonable security solution that does not put too much overhead on the limited resources available to IoT devices used for mobility and coverage purposes.

The results from the cyber-peeping can help increase the awareness for security and privacy among the users of IP cameras. Similarly, the proposed countermeasure may help in reducing cyber-attacks on all types of IoT devices. Chapter 4 and 5 dig deeper into these studies and provide more details.

CHAPTER 4: ATTACK OBSERVATIONS

This chapter focuses on the study of attacks on IP cameras from the point of view of cyber peeping, because existing research on information security for IP cameras has been primarily focused on issues with authentication or malware, but not on the peeping method itself. How cyber peeping is conducted in real world can further help in strengthening defenses accordingly and spread more awareness about dangers of IP camera.

In this chapter, we will examine attacks for cyber peeping by setting up a honeypot using decoy cameras in two scenarios, as follows:

- First, where a background information (handwritten URL and ID/password bait) can be read by humans. This will help us differentiate between humans and machine-based attempts. The assumption here is that machine-based programs cannot read background handwritten information.
- Second, by simulating a living-room in a home environment, to make it more close to a real life scenario. This will help us to capture more attacks by cashing in on human curiosity of the attackers.

This chapter is based on a journal article titled “Dangers of IP Camera – An Observational Study on Peeping [J-1].” The Journal of Information Processing awarded this publication as the “JIP Specially Selected Paper” (特選論文).

4.1 Introduction

Among IoT devices, digital cameras that allow remote viewing and operations via

Internet can be collectively referred to as IP cameras. There are many IP camera devices connected to the Internet with vulnerability and authentication issues, and it is, therefore, possible for a third person to electronically peep into them. Such peeping actions via Internet can be referred to as cyber peeping. Another issue is due to the presence of certain websites, like Insecam [6], which provides video images of freely accessible IP cameras for anyone, thereby, creating security and privacy issues.

So far, most of the research on unauthorized access to IP cameras has been focused mainly on authentication, changing camera configuration information, and observing malware infections that exploit vulnerabilities [8], [9], [10], [11], [12], [13], [14], [15]. However, there has been no investigation into the actual state of cyber peeping. Knowing the peeping techniques can help in further understanding the risks associated with using IP cameras and improving awareness among the general public and IP camera vendors from the security and the privacy point of view. Therefore, in this research study, in order to investigate and analyze the actual situation of cyber peeping in the real world, an IP camera was set up as a honeypot (hereinafter referred to as the “decoy camera”) for conducting an observational experiment. Two types of observation environments were established for the experiment purposes.

In the first observation experiment, we prepared two cameras (hereinafter referred to as the “URL reflection type decoy camera”), that displayed a hand-written note for a specific URL and two sets of different ID/passwords (one for each decoy camera) for access confirmation purposes, and assigned 10 IP addresses to each camera for observations. The objective was to study an access by a human element, who can read the background information (reflected URL and ID/password) via peeping into the decoy camera and then use that information to successfully gain access to the reflected URL.

On the URL side, we examined what ID/password was entered for determining if humans were attempting access after viewing the video of the decoy camera.

Although the first observational study helped in determining human element involvement, it was limited due to the fact that such a set up did not provide continuous peeping interest as the decoy camera was only showing a URL and its related ID/password. Therefore, a second observation experiment was setup where a room was prepared for observation that simulated a living-room of a home in which movement can be expected. Five decoy cameras (hereinafter referred to as the "living-room decoy camera") were installed to show the video of this room. These living-room decoy cameras were then exposed to Internet so that peeping methods could be examined in detail for a longer time. The objective was to attract the peeping entity to engage in more than one peeping event by providing a real life scenario.

As the result of this study, we were able to achieve the following:

- A) First observational study of IP camera peeping using real IP cameras in various environments.
- B) Demonstrated that a peeping problem does exist with a high degree when IP cameras are insecure.
- C) Revealed detailed access patterns on the insecure cameras, including the existence of automated accesses specifically tailored to find IP cameras efficiently.

Moreover, we had conversations with several IP camera vendors and explained the real world risks of insecure IP cameras. One of the vendors now deploy a security mechanism in their IP camera products that enforces users to set their own unique password.

4.2 Basics of IP Camera

An IP camera is a digital camera that can be accessed or viewed remotely via a network by connecting to the Internet. It is usually connected to the internet via a gateway/router using wired or wireless connectivity (Wi-Fi, Bluetooth, 3G/LTE/5G, etc.). An IP camera is also an IoT device and IoT devices are often connected to the internet using one of these connectivity options. Due to visual images, IP cameras generate a lot of data that adds to the consumption of available resources such as power and bandwidth in remote use cases. A web browser or manufacturer-specific application software is used to view its video. There are also IP cameras with functions allowing operational control such as directional movement or zooming etc. that can be operated via the browsers and/or an application.

4.3 Known Issues with IP Cameras

Many IP cameras are set to perform authentication via login ID/ password so that privacy can be maintained and images are not viewed by others than the authorized users. However, there are cases in which authentication may not be set or the login ID/password may not have been changed from the initial default setting, allowing unauthorized access from the outside and cyber peeping may be possible [46], [47], [48]. Also, even if the ID/password is set, there are many devices with weak or leaked passwords due to device vulnerabilities that can be peeked by outsiders [47], [49]. Similarly, there are many devices for which security measures may not be sufficient [46], [48].

Furthermore, there are several websites that gather information on cameras and videos that can be viewed without authentication [7], [15], [50], tens of thousands of such

cameras worldwide are posted on the Insecam website [6].

4.4 Observational Experiment

4.4.1 Environment Setup

In this research, the communication environment of the decoy camera was constructed by extending the method of IoTPOT, proposed in paper [11]. In this honeypot, as shown in **Figure 3**, a proxy script is running, and the received communication is transferred to the communication control machine.

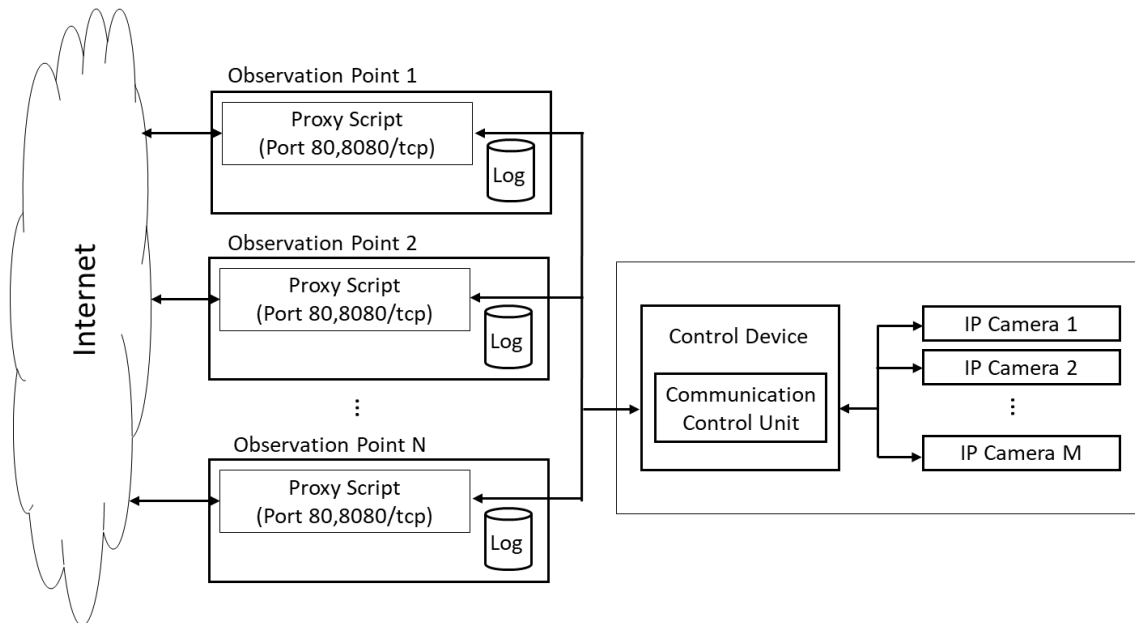


Figure 3 - IP camera honeypot.

In the communication control unit, the communication is transferred to the IP camera corresponding to each observation point, and the response is transferred to the proxy script at each observation point, such that an IP camera operating at each observation point is made to appear to an attacker as directly connected to the Internet. To ensure that an attacker can always connect to the same IP camera, we used a static mapping of a set

of 10 consecutive public IP addresses to each IP camera (except for one camera that was directly connected to the Internet with one public IP address only). For example, observation point 1 proxies a set of 10 consecutive public IP addresses assigned to IP camera 1 with the help of communication control unit, observation point 2 proxies another set of 10 consecutive public IP addresses assigned to IP camera 2 with the help of communication control unit, and so on.

4.4.2 Experiment Overview

In this research, two types of decoy cameras were set up to observe peeping in IP cameras. The first one was a decoy camera (a URL reflection type decoy camera) that displays a hand-written URL address and an ID/password for accessing that URL. In case of paper [9], the observation of access by the decoy camera with an ID/password was conducted, however in our research the URL is also reflected. In this URL, a script for basic authentication was run, and it was set to refuse login with any ID/password combination so that login challenges could be observed. Two cameras were used as URL reflection type decoy camera, each with a different ID/password set. **Table 1** shows the equipment used in this observation experiment.

Table 1 – Observation experiment with the URL-reflected decoy cameras.

IP Camera	Made in ...	Authentication Feature	ID / Password	IP Address	Operational Functions	Observation Period	Observation Days
A	Japan	No	admin/*****	10	Yes	Jun.11 ~ Jul.23, 2017	43 Days
B	Taiwan	Yes		10	No	Jun.11 ~ Jul.23, 2017	43 Days

Camera A was set to allow viewing of camera images without authentication, and Camera B was set to be able to access the video with the camera's default ID/password. Each camera was assigned 10 consecutive IP addresses as the observation points, and the

traffic on 80/TCP and 8080/TCP from these points was set to be relayed to Cameras A and B.

Figures 4 and 5 show the images from the URL reflection type decoy camera A and the URL reflection type decoy camera B, respectively. Both of the URLs are the same, but the ID/password is different, so that it is possible to determine which camera was viewed by the peeping host who input that particular ID/password.

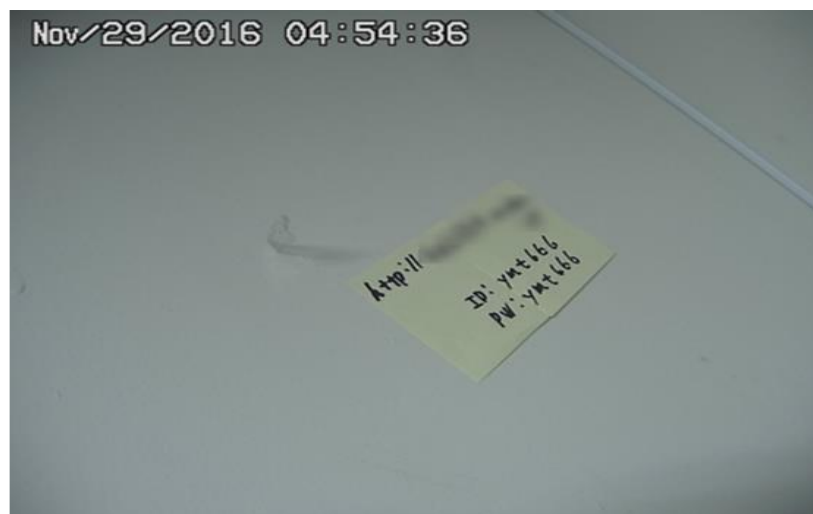


Figure 4 - Image from the URL-reflected camera A.



Figure 5 - Image from the URL-reflected camera B.

In the second observation experiment, we prepared a room for observation simulating a family living-room where the image is expected to change, because the first type of observational setup did not attract continuous peeping as the URL reflection type decoy camera is not much attractive for peeping toms if the camera is only showing the URL and ID/password. Therefore, another study with a decoy camera setup (living-room decoy camera) was used to simulate a real life living-room view (**Figure 6**).



Figure 6 - Image from the living-room decoy camera.

For the observation room, we used the home network test bed environment proposed in the paper [51]. For the living-room decoy camera, five cameras shown in **Table 2** were used.

Table 2 – Observation experiment with the living-room decoy cameras.

IP Camera	Made in ...	Authentication Feature	ID/ Password	IP Addresses	Operational Functions	Observation Period	Observation Days
A	Japan	No		10	Yes	Oct.06~ Nov.25, 2017	51 Days
C	Japan	No		10	Yes	Oct.06~ Nov.25, 2017	51 Days
D	Japan	No		10	Yes	Oct.06~ Nov.25, 2017	51 Days
E	Japan	No		10	No	Oct.06~ Nov.25, 2017	51 Days
F	China	Yes	admin/***	1	Yes	Sep.21~ Nov.25, 2017	66 Days

The living-room decoy camera A was the same device as the URL reflection type decoy camera A. The living-room decoy cameras A, C, D, and E, were set such that they could be browsed without authentication, whereas the living-room decoy camera F was set to allow login with the default ID/password. A set of 10 consecutive IP addresses was assigned to each living-room camera A, C, D, and E, and the communication addressed to 80/TCP and 8080/TCP was set to be transferred to these cameras. As the living-room decoy camera F did not get traffic transferred to it from the proxy, therefore, it could only be viewed via 1 IP address on 80/TCP port.

4.5 Analysis and Results

A common attack pattern was validated in both observations where the attackers first collected intelligence about the target IP address by scanning for available ports and services. Once they understood what is connected to the target IP address, they then proceeded with different peeping techniques, described in the below sub-sections.

4.5.1 Observation Results from URL-reflected Decoy Cameras

Table 3 shows the number of hosts that sent HTTP requests, the number of hosts that succeeded in authentication, the number of hosts that acquired camera view (peeped), and the number of hosts that operated the camera, as observed by the URL-reflected decoy camera experiment. Since the camera A is set to allow access to the image without authentication, the column for successfully authenticating hosts is not applicable (as marked by a diagonal line). Similarly, as camera B does not have a camera operation function, the number of hosts that operated camera is not applicable (diagonal line).

Table 3 – Observation results with the URL-reflected decoy cameras.

Camera	Request Hosts	Successfully Authenticating Hosts	Camera View Hosts	Camera Operations Hosts
A	27,116		25,585	499
B	1,280	4	0	

Furthermore, the host that peeped in means "the host that sent the request to acquire a video or image of the camera view". In case of Camera A, quite a large number of hosts (over 94%) targeted this camera for peeping, i.e., 25,585 hosts peeped out of the 27,116 hosts that sent the HTTP request. Among these peeping hosts, 1.7% (449 hosts) were observed controlling the operation of the camera. We confirmed this from the fact that Web-UI browser access of Camera A was used in order to access the camera functions. For the camera B, 4 hosts out of the 1,280 hosts that sent the request succeeded in logging in, but there was no host that acquired the camera video, and no peeping attempt aimed at the device was observed.

Figure 7 shows the transition of the number of hosts that sent the HTTP request to

the URL reflection type decoy cameras. Access to Camera B was almost constant, but access increased rapidly on Camera A on the third day of observation. When this case was investigated, many accesses using Insecam as a referrer were confirmed after June 13. Therefore, when we actually accessed Insecam website [6], the video of the camera posted there on June 13 was confirmed that explained the sudden increase in the number of peeping hosts.

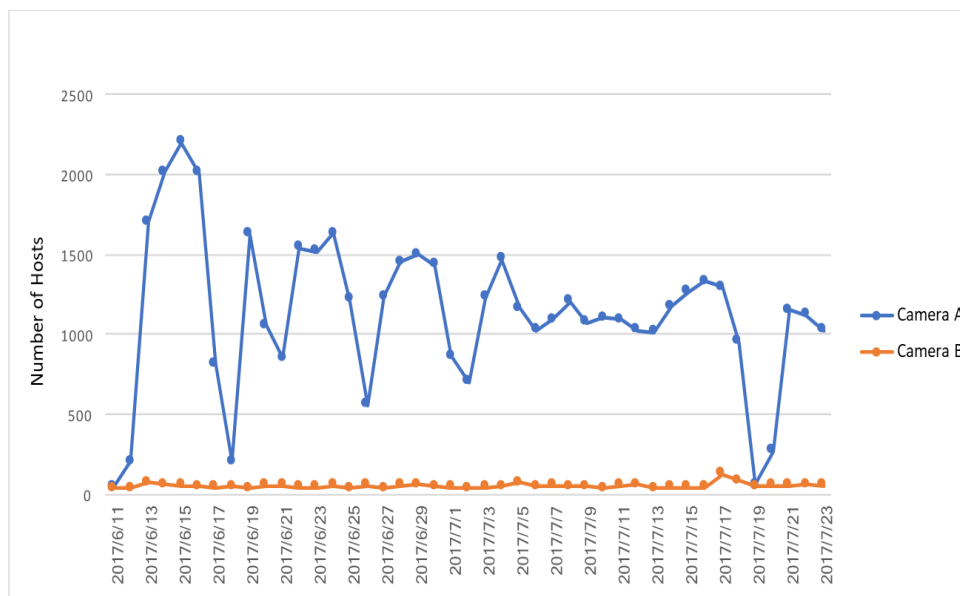


Figure 7 - Trend of hosts accessing the URL-reflected decoy cameras.

Next, **Table 4** shows further peeping attempts for the reflected URL in the URL reflection type decoy cameras by a number of access hosts attempting logins using the domain of the reflected URL rather than the direct input of the IP address. It also shows the number of hosts that entered the ID/password set reflected in Camera A, and the number of hosts that entered the ID/password set reflected in Camera B. Of the 583 hosts that accessed the URL, 422 were the access hosts that used the domain of the URL, that is, the host whose content of the HTTP header of the HTTP request matched the domain in the URL. Since this URL has not been disclosed outside this experiment, it is highly

likely that access using that domain was made with visual observation of the reflected background information from the camera image. In addition, 217 hosts trying to login to the URL and entering the ID/password reflected on the camera A were observed. Therefore, we can say that certain number of people took the next action such as further accessing the reflected URL after reading the reflected ID/password information from the camera A video.

Table 4 – Access analysis of peeping hosts in the URL-reflected decoy cameras.

Number of Hosts that Requested Access	Hosts that Requested Access via URL	Number of Hosts that Attempted Login	Hosts that Requested Access using Info from Camera A	Hosts that Requested Access using Info from Camera B
583	422	235	217	0

4.5.2 Observation Results from Living-room Decoy Cameras

Table 5 shows the number of hosts that sent HTTP requests, the number of hosts that succeeded in authentication, the number of hosts that browsed the camera (peeped), and the number of hosts that operated cameras, as observed by the living-room decoy camera experiment. Similar to Table 3, here also for the cameras that can access the video without authentication, the column for the number of hosts successfully authenticated was not applicable (diagonal line). Similarly, for the camera with no camera operation function, the number of operated hosts was not applicable (diagonal lines) either. Although none of the living-room decoy cameras were listed on Insecam website, still multiple peeping accesses were observed. Further analysis are provided in Section 4.6.

Table 5 – Observation results with the living-room decoy cameras.

Camera	Number of Hosts Requested Access	Hosts Successfully Authenticated	Hosts that Browsed Camera	Hosts that Operated Camera
A	1,755		33	8
C	1,998		66	18
D	1,806		13	1
E	1,749		4	
F	876	51	32	6

4.5.3 Peeping Characteristics

The peeping characteristics observed with each IP camera device are described below. In figures 8 to 16, a part of the character string in the request has been masked for security and privacy purposes.

Camera A

When this device is accessed by a general web browser, the video or image being captured is acquired with the requests as shown in **Figure 8**. We benchmarked this pattern as the normal case when accessed by a general web browser.

```
GET /cgi-bin/xxxx?resolution=yyyyyquality=yy&page=yyyyyy&Language=yy  
GET /cgi-bin/xxxxxxx?resolution=yyyyy&page=yyyyyyyy&Language=yy
```

Figure 8 - Image acquisition request to camera A using a browser.

Among the hosts who peeped into camera A, some hosts acquired images multiple times with the requests as shown in **Figure 9**.

```
GET /cgi-bin/xxxxxx?resolution=yyyy&xxx;quality=yy&xxx; Language= yy  
&xxx;yy  
GET /cgi-bin/xxxxxx?resolution=yyyyquality=yy&Language=yy&COUNTER
```

Figure 9 - Automatic image acquisition request to camera A.

On comparing the two requests, it can be seen that the request shown in Figure 9 is very different from that of the browser access (Figure 8) because it includes the character strings such as amp and COUNTER, and the argument page is not added. From this, it is considered that this access uses a tool or script that acquires images automatically. Hence it can be concluded that there is a host that targets a specific manufacturer's equipment (IP camera) and performs peeping and image acquisition automatically.

Furthermore, as shown in **Figure 10**, in addition to the image acquisition request of the relevant device, a host was observed sending a request to acquire the image of the IP camera of other manufacturers. This host sends similar requests to multiple IP addresses and it seems that many IP addresses are being tried to access for collecting images from various IP cameras.

```
GET /xxxxxx.JPG?COUNTER  
GET /cgi-bin/xxxxx?resolution=yy&quality=yy&Language=yy&COUNTER  
GET /cgi-bin/xxxxx.cgi?chn=yy&login&pwd&q=yy&COUNTER  
GET /mjpg/xxxxxxx.mjpg?COUNTER  
GET /xxxxxxximageyyyyy?COUNTER  
GET /cgi-bin/xxxxxx  
GET /cgi-bin/xxxxxx?fake=yyyy  
GET /cgi-bin/xxxxxx?resolution=yyyyquality=yy&Language=yy&COUNTER
```

Figure 10 - Request for an automatic search of multiple types of cameras.

In addition to above patterns, a host that peeps for a long time was also observed.

The access flow of this host is shown in **Figure 11**.

```
GET /xxxxxx.cgi?login&pwd  
GET /cgi-bin/xxxxxx  
GET /cgi-bin/xxxx?resolution=yyyy&quality=yy&page=yyyyy&Language=y  
GET /xxxxxxxxxJPEG  
GET /cgi-bin/xxxxxx  
GET /cgi-bin/xxxxxx?fake=yyyy
```

Figure 11 - Part of the long-term access to camera A.

First, a request (shown in the first line of Figure 11) was transmitted 18 times in total by changing the value of the argument “pwd”. However, it seems that these requests are targeted to other IP cameras because this device returns “404 Not Found” error message. After 1 minute 36 seconds, the second line request is sent twice, and the captured image of this device is acquired. Since this request is obviously different from the request observed in Figure 8, it is considered to be an automated access by a tool/script or the like. 52 seconds later, the video is acquired with the request on the third line. At almost the same time we observe requests of JavaScript, CSS, image files etc., and since “User-Agent” is also a relatively new version of the actual browser, therefore It seems that some human has accessed using a web browser. After 3 minutes 17 seconds, image acquisition is performed again with the request of lines 4 and 5. 14 seconds later, the request on line 6 was intermittently transmitted at 1 - 3 second intervals with a different “User-Agent” than before. The value entered in the argument fake was different each time, but the intention to add the argument is not clear because the camera display image at the time of transmission is acquired regardless of the value of fake and whether or not it exists. The image acquisition by this request was observed intermittently until reaching 42 hours. When analyzing the access flow of this host, it seems that the requests in lines 1 to 5 are

for searching the camera first, and continuous image collection is started when the camera is found. Moreover, in the camera A, 8 hosts were observed operating the camera in addition to viewing the images. Both hosts include requests for JavaScript, CSS, image files, etc. that are generated by access using a browser, and therefore, it seems that humans accessed the camera with a browser and operated the camera. However, the two hosts sometimes were observed to send automated image acquisition requests before and after an access by a browser. Thus, performed peeping by combining an automated access using a tool and checking the video using a browser.

Camera C

In camera C, it was confirmed that the image was acquired by a request (**Figure 13**) different from the request for image acquisition using a browser (**Figure 12**).

```
GET /xxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&View= xxxxx &Count  
=yyyyyy
```

Figure 12 - Image acquisition request to camera C using a browser.

```
GET /xxxxxJPEG?Resolution=yyyy
```

Figure 13 - Automatic image acquisition request to camera C.

This request is significantly different from the request by the browser in that some arguments are not assigned, and therefore it seems that the request for access is by a tool or the like that automatically collects images. In addition, one host sent the request pattern as shown in Figure 12, intermittently between October 21 and November 19, acquired a total of 2,528 images continuously, and performed long-term peeping access. Of the 66 IP addresses that peeped, 18 IP addresses operated the camera, and 9 IP addresses were

concentrated in comparatively close address range within the same AS [52]. A detailed analysis of the access from these 9 IP addresses showed that the “User-Agent” used to acquire the images was the same, and the iPhone web browser Safari [53] was used. In addition, "GET / apple - touch - iconicon.png" has been observed, and this request is an icon image acquisition request that iPhone uses to create a site shortcut on the home screen, and this request is unique to iPhone or iPad. From these facts, there is a high possibility that access from 9 IP addresses is using the iPhone. A part of such access from these 9 IP addresses is shown in **Figure 14**.

```
Observation Time: 2017/10/08 04:11:29, Access Source IP: xxx.xxx.xxx.004
GET /xxxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&Count=yyyyyy115
Observation Time: 2017/10/08 04:11:32, Access Source IP: xxx.xxx.xxx.004
GET /xxxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&Count=yyyyyy116
Observation Time: 2017/10/08 04:11:36, Access Source IP: xxx.xxx.xxx.004
GET /xxxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&Count=yyyyyy117
Observation Time: 2017/10/08 04:56:25, Access Source IP: xxx.xxx.xxx.016
GET /xxxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&Count=yyyyyy118
Observation Time: 2017/10/08 11:57:49, Access Source IP: xxx.xxx.xxx.007
GET /xxxx/xxxxxxxxxJPEG?Resolution=yyyy&Quality=xxxx&Count=yyyyyy119
```

Figure 14 - Part of the group of accesses appears to be from the same person.

In the case of access by the browser in the camera C, the display image is acquired every 3 seconds by the request shown in Figure 12, and each time the argument Count of the request increases by 1 until the page is reloaded. In Figure 14 with 5 image acquisition requests, the Count is incremented by 1 for each request, and it seems that the access is from the same client. However, the source IP address is different for the 1-3, 4th and 5th requests. Further analysis of this phenomenon shows that there is an interval of about 45 minutes between the 3rd and 4th accesses and about 7 hours between the 4th and 5th

accesses. So in the process of accessing the device using the iPhone, it can be considered that this is due to the fact that the allocation of IP address has changed due to a physical movement or time lapses. In other words, all of these accesses are expected to be by the same person using the same device.

Camera D


We also observed several peeks at camera D. The image acquisition request by the browser of the camera D is the same as the request of the camera C (Figure 12). Camera D observed several peeping accesses, but most of them were accesses to obtain images automatically using the same tools as in Figure 13. In addition, one host was observed operating the camera. This host also operated camera C.

Camera E

Camera E also observed a group of requests (Figure 10) to search for multiple cameras, similar to those observed with camera A. In addition, no automated access or long-term access targeting only the relevant device was observed.

Camera F

In camera F, 32 out of 51 hosts that successfully logged in were observed peeping. This camera had a known vulnerability in which the default ID/password was leaked by sending a specific request to camera F device, and hosts that peeped exploiting such vulnerability were observed. **Figure 15** shows a part of the access flow.



```
GET /xxxxxx.cgi  
GET /xxxxxxxxxxxxx.cgi  
GET /xxxxxxxxx.cgi?login&pwd&streamid=yyy&audio=yy&filexxxx=
```

Figure 15 - Part of the peeping access using a vulnerability.

An ID/password is acquired by sending the request of line 1 and 2. We noticed from the “User-Agent” that these requests are presumed to be accessed by tools etc. other than the browser. After that, this host logged in based on the obtained information and browsed the video using the request shown in line 3. At the time of this access, besides reading CSS and image files etc. similar to the browser access pattern, it was also observed that the “User-Agent” acquired the video using the concerned camera-specific plug-in of Internet Explorer [54]. This indicates that a human accessed via the browser and peeped in using the plug-in.

Moreover, some of the other hosts obtained videos automatically instead of using a browser after obtaining an ID/password, and host peeping at a specific device was also observed. Furthermore, in addition to peeping, we also noticed a request sent (as shown in **Figure 16**) and observed an attack from one host to change the port number for video delivery from 80/TCP to 788/TCP. If the intention was to exclude other intruders, then this action will also obstruct browsing by an authorized user. Therefore, this intention is questionable due to the effect of changing the settings in such a way.

```
GET /set_XXXXXX.cgi?xxx_url=XXXXXX.htm&login&pwd&ipaddr=yy.yy.yy.yy  
&mask=yy.yy.yy.yy&gateway=yy.yy.yy.yy&dns1=yy.yy.yy.yy&dns2=yy.yy.yy.yy&dhc  
p=yy&port=788&rtspXXXX&XXXXport=yyy&onXXXX=yy
```

Figure 16 - Part of the port number change request.

4.6 Discussion

As a result of the first observational study, access to view the video of the URL reflection type decoy camera was observed from 25,585 IP addresses in 43 days. In

particular, when one of the URL reflection type decoy cameras was posted on Insecam two days after starting the experiment, the observed access immediately increased from 1,701 IP addresses per day to more than 20,000 times, with 94% peeping. From this, it can be stated that a large amount of cyber peeping occurred because of the Insecam website. On the other hand, though the other URL reflecting camera was accessed by a few hosts, we did not observe a peeping host acquiring its camera view. Thus we can say that the weakness of login authentication and type of device can make a difference in the number of peeps. Moreover, 217 IP addresses, i.e., 0.8% of total hosts who acquired video of reflected URL decoy cameras attempted login with the reflected information. Though a very smaller number than expected, however, we were able to observe the human element by confirming hosts accessing the URL displayed on the URL reflection type decoy camera A and executing further (after viewing the URL information from the accessed camera) by logging-in with that reflected ID/password.

In the case of the second observational study, even though none of the five living-room decoy cameras got posted on Insecam, we still confirmed that peeping access from multiple hosts appears to be due to the nature of living-room image triggering curiosity. We observed a host that periodically acquires a video image of the camera automatically with a dedicated tool specialized for image acquisition. In addition, access to operate cameras and access for a long time (as much as 42 hours for a single camera) were also observed. Furthermore, we observed an attacker who peeped at the camera after breaking through the authentication and an attack that changed the TCP port number of the user interface for viewing.

As a result of analyzing these accesses, we were able to study the real-world situation of different methods of peeping, such as automated access by hosts that search for

cameras and acquire images or hosts that automatically obtain video targeting specific devices. We also observed hosts where a human seems to conduct further peeping after reading the reflected hand-written information and then log in to the peeped URL successfully. Moreover, cyber peeping by exploiting the known vulnerabilities of cameras such as default or weak authentication and operating them was also confirmed. Besides peeping method, we also confirmed that unauthorized access usually starts with intelligence collection through scanning (by humans or automated), even in websites like “Insecam”. Often vulnerabilities in the security of IP cameras are then exploited due to weak/default/no passwords or unpatched systems.

4.6.1 Limitations and Ethical Considerations

Although websites like “Insecam” have succeeded to demonstrate there are many insecure IP cameras on the Internet that have no password set, our study further revealed that there are indeed a considerable number of unwanted accesses to such insecure cameras. In order to improve the situation, we tried to inform two main stakeholders: end-users and IP camera vendors. We consider that publishing our work is one of the main channels to inform the end-users. It is worth noting that the work acquired some media attention and the experimental results were introduced in several TV programs and news by the national broadcasting station [55], [56], from which we believe that we have somewhat contributed in improving public awareness on the risks and dangers of IP cameras. Moreover, we had conversations with several major IP camera vendors to inform them of the increasing cyber threats. One of the vendors now adopts an improved security mechanism. In order to minimize the possible harm to the vendors of the IP cameras, we anonymized the vendor names and tried to redact identifiable information as much as possible from the analysis results. Though we have taken precautions, we believe that the

benefits this study brings would exceed the harm that it might have caused.

4.7 Summary

This chapter has provided us valuable insight on various attack methods (involving humans and automation tools) for the purpose of peeping into cameras connected to the Internet. An IP camera depicts common IoT devices that are exponentially connected to the Internet. Through this study, we can understand what triggers an increase in illegal access and how the attackers obtain information from the devices providing a service over the Internet. Based on the response from the device, they further explore or exploit. This study has also highlighted the importance of strengthening security parameters in IP cameras so that they can avoid unnecessary exposure to websites collecting and showing images of easily accessible IP cameras. Furthermore, through this study, we were able to observe several automated attack patterns for cyber discovery and peeping purposes. As IP cameras or sensors need to connect to the internet, so blocking the intruder's discovery process or scan for collecting information about the IoT device can help us in mitigating unauthorized access. In the next chapter, we will examine a possible deterrence against such unwanted attacks on an IoT device.

CHAPTER 5: COUNTERMEASURES – STEALTH SECURITY

In the previous chapter, we performed an observational study on the cyber-peeping attacks that showed how easily data can be stolen with default/no passwords due to a lack of user awareness or unpatched systems. This chapter focuses on the security challenges due to lack of knowledge, limited resources, and remote management challenges. We then conduct an empirical study on our proposed security countermeasure for protecting resource-constrained IoT devices and reducing the user dependency for managing the IoT devices.

This chapter is based on a journal article titled “Empirical analysis of security and power-saving features of port knocking technique applied to an IoT device [J-2]¹.”

5.1 Introduction

In the 21st century, the Internet of Things (IoT) has opened up a new horizon for entrepreneurs and hackers. According to a well-known security company “Norton,” the number of IoT devices is estimated to reach 21 billion by 2025 [4]. The booming 5G technology means billions of IoT devices can connect directly to the Internet using the 5G speeds over the cellular networks [45], which would make them more susceptible to direct Internet attacks [4]. The current situation in the year 2020 is that Gartner expects over 25% of known attacks to involve IoT, whereas the IT security budgets for IoT would be less than 10% [57]. The IoT device resources are also getting scarce due to small sizes designed for portable use. This may result in lesser or almost no traditional security

¹ The material in this chapter was presented in part at [T-1] and the figures are being reused.

features due to the resource constraints in the IoT devices. These issues will increase the opportunities for hacking [37], [38]. Furthermore, botnets are using self-propagating malware, such as “Mirai” for attack purposes. The first large scale attack on a single enterprise was witnessed in the year 2016 that used millions of compromised IoT devices by the “Mirai” botnet malware, the code of which is now available on the Internet for anyone to use or modify [1], [2], [3], [4], [25]. Therefore, with the growth of the IoT market, the attack field is equally broadening and the cyber-attacks exploiting IoT vulnerabilities in the network services and inadequate security are on the rise. On the other hand, the end-users do not have adequate technical knowledge to fix security issues by themselves. Due to market competition, manufacturers are more interested in being first to market than adding security measures that may require time, resources, and additional costs. The lack of security management and limited resources on the IoT devices has, therefore, become a big challenge.

5.2 Perimeter Defense Options

One possible approach is to keep the security of IoT devices up-to-date using the remote management feature. But, the research shows that such features are also subject to common attacks on well-known services, such as telnet, ftp/tftp, ssh, and web/http [1], [4], [18], [38], [41]. These remote management services are among the top 20 most scanned ports and often are an ideal target for exploitation as they provide direct access to the system with escalated privileges [58]. Securing the remote management capability would require incorporating additional security measures on the already resource-constrained IoT devices. The resource-constraints, such as power consumption becomes challenging when the smart uses require mobility and coverage range. For example, uses

in agriculture, healthcare, logistics, etc.

According to the research done by Schneier [16] and Henke [30], the end-users often do not have access to the operating system of the IP camera or an IoT product that they are using. They also do not have adequate technical knowledge to even patch security vulnerabilities without the support of device manufacturer. On the other hand, the device manufacturers are more driven by the competition in the market and the race to put new products first in the market. Security of their products is often compromised as they only see revenue by who captures the market first with user-attractive features, which generally do not include security due to lack of security awareness and knowledge on the buyer side. Few brand conscious manufacturers only pay attention to security vulnerabilities out of the fear of brand damage [17]. Therefore, the products should be secured at the point of entry, i.e., network level in order to avoid post-production security lapses.

The insights obtained from the first study of attack observations also confirm the lack of security focus from the manufacturers that left IP cameras unpatched and with vulnerabilities. Similarly, the lack of user awareness and technical knowledge was observed in Mirai attack and validated in our testing with IP cameras, where they were easily accessed by unauthorized users when left without changing default passwords or configured with weaker passwords.

With the advancements in communication technology, the usage of IP cameras has also gone beyond the static limitations. Mobility factor has pushed them out into more challenging scenarios where the availability of resources is also limited. For example, environmental monitoring, animal movements, agriculture, drones, remote visual sensor networks, goods transportation, security surveillance, etc. With mobility comes new challenges of limited available resources and power. Energy savings and real-time

performance is critical in IP cameras or visual sensor networks because of the amount of image data that put constraints on the available resources due to both processing and transmitting large image data [59].

Hence, all these factors need to be taken into consideration when selecting an appropriate countermeasure for the IP camera or an IoT device in the second part of the study [14], [16], [17], [30], [37], [59], [60]. **Table 6** provides the selection criteria for available options based on the factors highlighted above:

Table 6 – Comparison of possible perimeter defense options.

Perimeter Defense Options	Limitations for resource-constraint IP camera / IoT device	
	Pros	Cons
Firewall	<ul style="list-style-type: none"> ➤ Commonly used security option for protecting networks 	<ul style="list-style-type: none"> ➤ Resource-centric, no power-savings ➤ Need technical knowledge and management that end-users often do not have ➤ Susceptible to zero-day threats
VPN	<ul style="list-style-type: none"> ➤ Commonly used security option for remote connectivity 	<ul style="list-style-type: none"> ➤ Resource-centric, no power-savings ➤ Need client and technical knowledge ➤ Need to establish separate VPN for each user, ➤ Slows connectivity ➤ Zero-day threats
IDS / IPS	<ul style="list-style-type: none"> ➤ Covers known threats/patterns only ➤ Each packet is analyzed in depth ➤ Provides threat and security visibility 	<ul style="list-style-type: none"> ➤ Resource-centric, no power-savings ➤ Need technical knowledge and management that end-users often do not have, ➤ Device manufacturer need to embed an agent program in each IoT device for visibility ➤ False positives ➤ Susceptible to zero-day / unknown threats

Cloud-based security	<ul style="list-style-type: none"> ➤ Cloud-based security enables the device and users to connect through an IoT cloud rather than running services on the device ➤ It is easier to secure a cloud than each IoT device for patching vulnerabilities and management ➤ Plenty of resources available for security in the cloud than on the IoT device ➤ End-users do not need any technical knowledge or management skills 	<ul style="list-style-type: none"> ➤ Manufacturer / service provider need to develop and maintain the backend cloud system as long as the service is provided to the end-user ➤ Cloud misconfigurations - most of the cloud breaches occur due to poor configuration or technical skills. So all device manufacturers will need extra resources (cost, knowhow, etc) and focus on cloud management / maintenances ➤ All users immediately lose service when the cloud is down or out of service ➤ Cloud-based security is though managed by the service provider, but it is a shared responsibility between customers and service provider
Port Knocking	<ul style="list-style-type: none"> ➤ Allows IoT device to work alone without any external support / cloud ➤ Minimum security management required by manufacturer ➤ Effective against zero-day threats also as the IoT device is hidden from the Internet ➤ Can provide power-savings that is critical for resource-constraint IoT devices ➤ End-user do not need any technical knowledge or management skills 	<ul style="list-style-type: none"> ➤ Device manufacturer need to integrate the port knocker client into its applications ➤ Device manufacturer need to manage the secret keys for synchronizing port knocking client and server

Therefore, not every security countermeasure can be implemented and due diligence need to be exercised very carefully to ensure that the security options do not obstruct the intended use of the IoT device. Hence, conventional security methods are not so practical

to use in IoT devices due to the limitation of available resources, power consumed, management overhead, and the lack of technical knowledge. Thus, a lightweight, secure solution for remote management is required in IoT devices.

5.3 Challenges

Port knocking is one such technique in which a port can be configured to remain hidden (closed) until receiving a predetermined set of knocks (packets) on different ports in a specific order. The literature research shows that studies have been conducted on various port knocking methods often used for the remote management of large systems and importing them into the IoT world as well [19], [39], [61], [62] [63]. However, such studies have been mainly focused on the port knocking algorithms, authentication, and various attacks or complexity of an algorithm being used for knocking and its impact on the performance in terms of protocols, physical memory, or network bandwidth. Also, the security testing has often been only in a controlled environment using penetration testing and not exposing the IoT device with port knocking technique directly to the Internet over the cellular network. Existing research is missing an important piece of information that can greatly impact the intended use of an already resource-constraint IoT device. We must also examine the impact of a security feature on CPU usage and power consumption to ensure that the additional security features do not impact negatively on the resource-constraint IoT devices. These elements are vital as IoT devices footprint is becoming smaller and smaller along with the diversity in usage. Therefore, to the best of our knowledge, the below research questions remain unclear:

Q-2 To what extent can we overcome the challenges imposed by the limited-resources of IoT gateway devices designed for mobility and coverage with cellular connectivity, in the following context:

- (a) How much computing power (CPU consumption) would a security feature add to a resource-constraint IoT device?
- (b) How effective this security feature would be in blocking unwanted access to an essential service when the IoT device is exposed to the Internet directly without other security layers or a firewall?
- (c) What would be the impact of adding this security feature on the power consumption of the IoT device?

In order to find answers to the above research questions, we narrowed our scope to focus on the popular IoT gateway devices with cellular connectivity, such as those used in smart cities, smart bicycles, goods tracking, flood monitoring, agriculture monitoring, medical monitoring, wearables, etc. Such use cases need cellular IoT in order to ensure mobility and coverage for their intended use. We initially experimented for a short period (2 weeks) to obtain preliminary results and then later on tested over a longer period (7 weeks) to verify the results. An IoT device was used to test two types of port knocking methods. The first method was based on the python script applying the pseudo-random number generator (PRNG) and the chaotic random number generator (CRNG) algorithms [18]. The second method was also based on the python script, but applying a stream cipher using the Authenticated Encryption with Associated Data (AEAD) algorithms [19]. The details of testing and analysis are provided in Section 5.6.

Hence, based on the results of the experiments conducted to measure the effect of using the stealth security technique of port knocking on the resource-constraint IoT device,

we can conclude that the experimental results imply that the power consumption overhead by receiving incoming session requests (from scanners/malware on the Internet) without port knocking would easily exceed the power consumption for running port knocking service. Thus, running the stealth port knocking service would be beneficial in terms of not only the security enhancement but also the power consumption on a resource-constraint IoT device.

5.4 Stealth Security

According to the Oxford dictionary, the word “stealth” means “the fact of doing something in a quiet or secret way” [64]. Similarly, the Cambridge and American dictionaries define the word “stealth” as “movement that is quiet and careful in order not to be seen or heard” or “the quality of carrying an action out secretly, so that people do not know it is happening” [65]. Therefore, based on the definition of the word “stealth” and comparing it with how the port knocking technique cleverly hides the ports on an internet-connected device, we have defined port knocking as a “stealth security” feature. It allows the IoT device to avoid detection by the scanners and enable a service for the authorized user without letting others know.

Port Knocking

The ability of this technique to hide the service port by not responding to unauthorized requests or scans makes it stealthy in nature. Port knocking is not a new concept as it has been used by the system administrators to manage the servers remotely. However, its application on IoT devices is relatively new. A common method of attack is

to first look for open service ports using a port scanner. In the case of port knocking, the service port is closed by default and opens for service only for the requester that sends the right sequence of packets to the firewall for authentication, as shown in **Figure 17** that was created by referring to the paper [66].

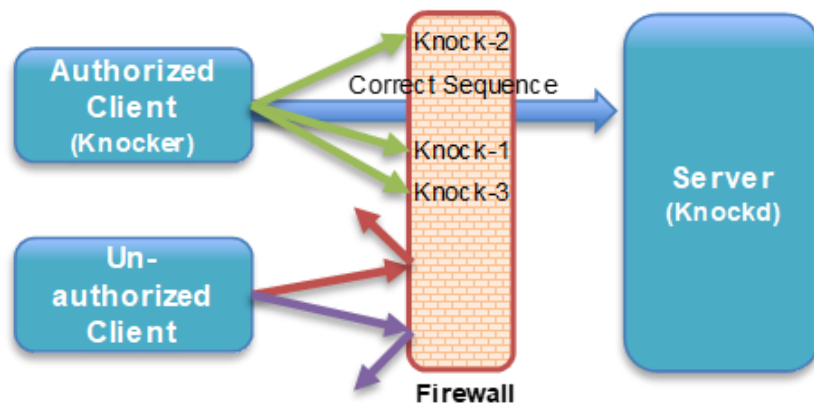


Figure 17 - Port knocking mechanism.

In the case of Linux-based devices, a “knockd” daemon process (server) monitors the knock sequence and open/close ports via the “iptables”. This knock sequence must be randomly generated and synchronized between the client and the server in order to avoid replay attacks or man-in-the-middle (MITM) type of attacks. With the service port closed, the target port cannot be confirmed during scanning, and therefore, an attacker cannot target it [66]. This stealthy feature of port knocking also helps in putting up deterrence against zero-day attacks [42].

Based on the research work in paper [18] and [19], we selected two approaches for our port knocking test on the IoT device because they are lightweight, can be used in the IoT environment, and can generate completely random numbers using two sets of completely different algorithms as follows:

PRNG-CRNG based Port Knocking Daemon

In this approach, the port knocking mechanism (python script-based “knockd” server) is set up by a Python script that produces pseudo-random port numbers using the system time as a seed with a PRNG algorithm, and then again combining the result with the CRNG algorithm for producing random order of assigning protocol (TCP=0 and UDP=1) selections, as shown in **Figure 18** that was created by referring to the paper [18].

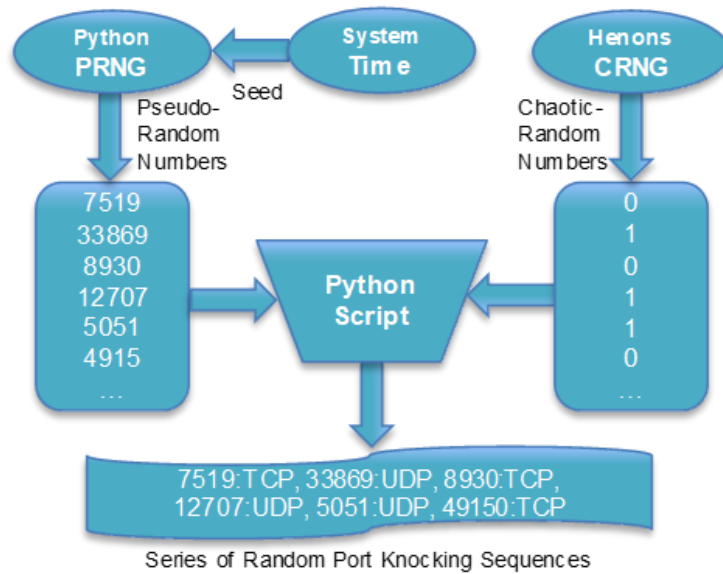


Figure 18 - PRNG-CRNG based port knocking.

This combination creates a system that always generates the same knock sequence based on the same algorithms running on the server and the client side. Due to the pseudo-random nature of the algorithms and stealthy way of hiding the service ports, this method of port knocking mitigates denial of service (DoS), playback, and MITM attacks as well.

Stream-Cipher-based Port Knocking Daemon

The second method is also python-based “knockd” server, but it uses ChaCha20 stream cipher with Poly1305 authenticator [19]. A 256-bit secret key is shared between

server and client, generating a 512-bit key stream from the secret key and time using the “Authenticated Encryption with Associated Data” algorithm, as defined in IETF’s RFC 8439 [67]. The key-stream is split into 10 knock sequences (1 sequence = 3 ports = 48bits). The frequency for updating the knock sequence is set at every 60-sec intervals. The key-stream is regenerated every 600-sec (10 min) intervals [68], as shown in **Figure 19**.

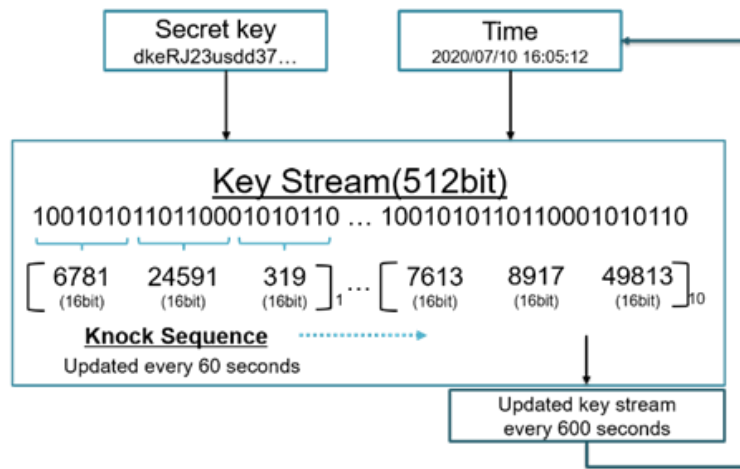


Figure 19 - Stream-cipher-based port knocking.

5.5 Empirical Analysis

In order to test the stealth port knocking technique for our research, we selected commonly available off-the-shelf IoT device from the Japanese manufacturer “Plat’Home” [69], [70]: a raspberry-pi hardware configuration with a 3G option for a direct Internet exposure using the cellular network connectivity, as shown in **Table 7**. Raspberry-pi is one of the popular off-the-shelf hardware, supporting numerous IoT uses and applications, including as an IoT gateway for connecting various kinds of sensors with related applications.

Table 7 – IoT test devices.

		VX2	BX1
CPU	Model	Intel Atom E3805	Intel Atom® Processor
	Speed	1.33 GHz	500 MHz
	Cache	1024 KB	1024 KB
Memory		2 GB	1 GB
Storage		32 GB	4 GB

5.5.1 Test Environment Setup

The test equipment used and the way the experiment was conducted in the lab can be seen in the photo, as shown in **Figure 20**. Various combinations of test setups were used to make sure we can minimize any external influences on the power consumption measurements and can obtain maximum possible accuracy.



Figure 20 - Lab experiment.

5.5.1.1 Test-1: Port knocking effectiveness in terms of CPU usage

First, we need to select a port knocking method that does not put too much stress on the CPU of the IoT device. We installed and tested both approaches (PRNG-CRNG based port knocking and stream cipher-based port knocking) on the same IoT test device one-by-one, targeting port 22/TCP for SSH service with port knocking. We used the “dstat” command while connected to the Internet via a 3G line for measuring the CPU usage in each case using a log analyzer, as shown in **Figure 21**.

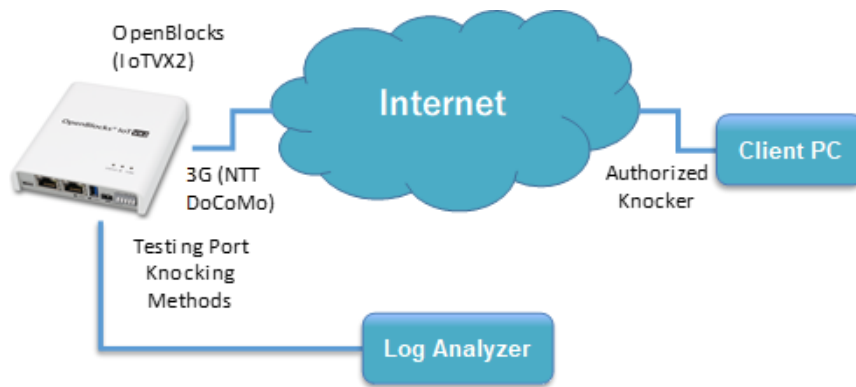


Figure 21 - Test-1 setup.

5.5.1.2 Test-2: Port knocking effectiveness in terms of security

Once we have identified an efficient port knocking method, then we examined how effectively it could reduce the unauthorized SSH login attempts on a default port 22/TCP by setting up the test as shown in **Figure 22**.

We used two IoT devices of the same model and specifications. We installed the stealth port knocking feature on one of them to hide the SSH service running on the default port. We kept the other device running without the port knocking feature so that we can compare the difference when both are exposed directly to the Internet with the same 3G network provider.

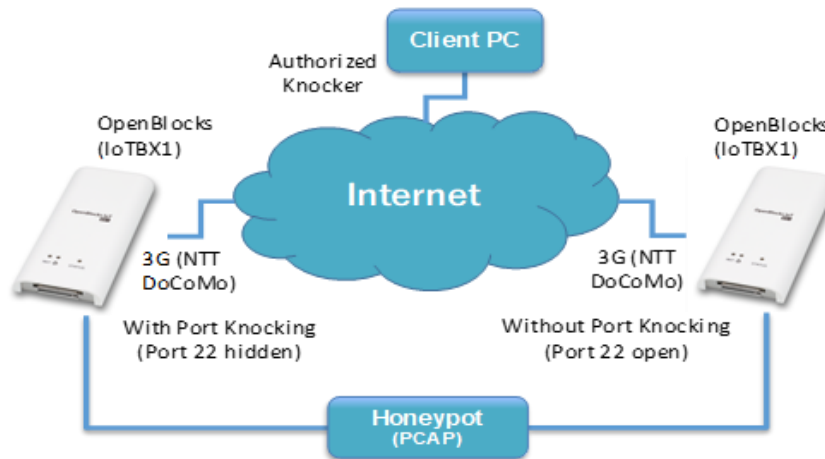


Figure 22 - Test-2 setup.

5.5.1.3 Test-3: Port knocking effectiveness in terms of power consumption

In order to examine the power consumed with and without the port knocking feature, we used the same two identical IoT BX1 devices that were running on the same software and hardware. USB testers were connected to each IoT device, and these testers were then connected to a self-powered USB hub. A note PC was also connected with the USB hub for observational purposes, as shown in **Figure 23**.

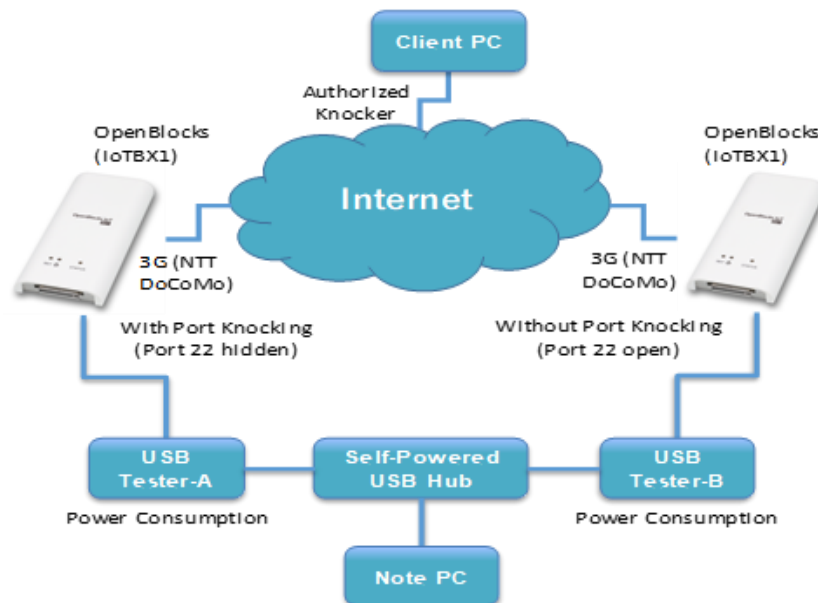


Figure 23 - Test-3 setup.

We then calibrated the USB tester without connecting the two IoT devices for 24 hrs.

and calibrated the two IoT devices without running the port knocking feature for 24 hrs. After the benchmarking, the stealth port knocking feature was enabled on the IoT device connected to the USB Tester-A. Whereas the USB Tester-B was used to record the power consumption of the IoT device without the port knocking feature. Both were directly connected to the Internet over the 3G cellular connection from the same service provider. This way, we ensured that we could take measurements at the same time on both devices for comparison purposes.

5.5.2 Analysis and Results

5.5.2.1 Test-1 Analysis

In the case of test-1 for finding out the effectiveness of port knocking in terms of CPU usage, we found that the PRNG-CRNG based port knocking method mostly consumed 50% of the CPU resources while updating the knock sequence and generating the key. Whereas in comparison, the stream-cipher-based port knocking method only consumed 9% of the CPU when updating the knock sequence, and the CPU usage was only 15% when generating the key, as shown in **Figure 24**.

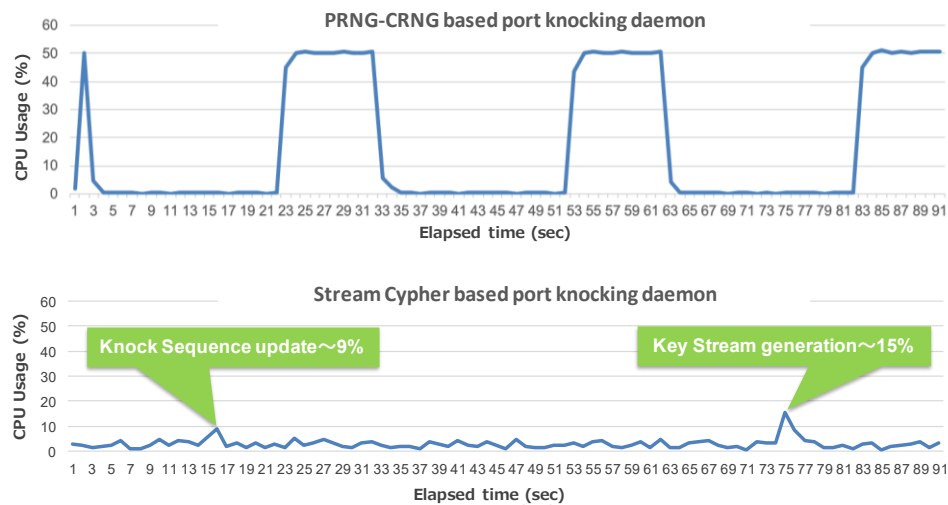


Figure 24 - Test-1 results.

Therefore, further testing was done based on the second method (stream cipher) to confirm the effectiveness of the stealth port knocking feature by hiding the SSH service running on the default port 22/TCP (when not in use but exposed to the Internet) and the power consumption. For further testing, we used the IoTBX1 model as it is more resource constraint than IoTVX2.

5.5.2.2 Test-2 Analysis

In the case of test-2, the stream-cipher-based port knocking method was further tested for 6 weeks (42 days) from October 18, 2020 to November 29, 2020. The results showed that this method of port knocking is quite effective from the security point of view as it stealthily used the SSH service running on the port 22/TCP without any issue while the IoT device was directly exposed to the Internet through the 3G cellular connection without any additional security elements for 6 weeks. During this time, we observed not a single unauthorized SSH login attempt on the IoT test device running the stealth port knocking feature. This is due to the fact that port 22/TCP was hidden and did not respond to any unauthorized SYN packets it received. In comparison, the device with no port knocking feature had 431,142 SSH login attempts from 5,424 unique IP sources (hosts) due to its visible SSH service running on the default port, as shown in **Table 8**.

Table 8 – Port knocking security effectiveness (6 weeks test results)

	With Port Knocking	Without Port Knocking
SSH Login Attempts	0 Times	431,142 Times
Source IP Addresses	0 Hosts	5,424 Hosts

This demonstrates that the stream-cipher-based port knocking was able to reduce the attack surface significantly, adding to the security of the IoT device.

5.5.2.3 Test-3 Analysis

Next, we analyze power consumption with and without the port knocking feature.

Voltage Stability:

In order to benchmark and make sure we do not observe any other influence on the voltage being measured by the USB testers, we connected only the two USB testers to the USB hub and measured voltage for 24 hours on July 22-23, 2020. Both came out to be stable around 5.14 volts, as shown in **Figure 25**.

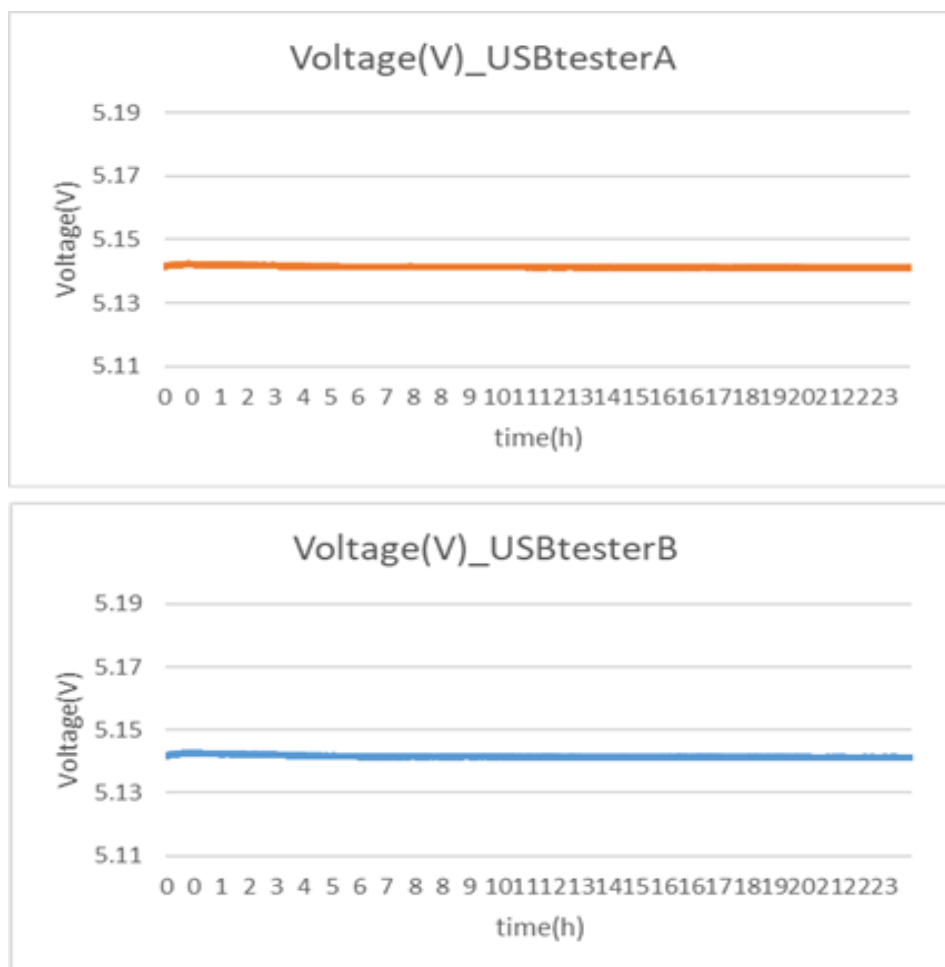


Figure 25 - Voltage stability results.

Confirming power consumption without port knocking:

Next, we established the baseline (benchmark) by measuring the power consumption of both IoT devices without port knocking for 24 hours on July 24-25, 2020. This way, we can observe how these IoT devices are consuming power with and without the stealth port knocking feature running on any of them. We observed almost similar readings (1.38W on Tester-A connected IoTBX1 and 1.37W on Tester-B connected IoTBX1), as shown in **Figure 26**.

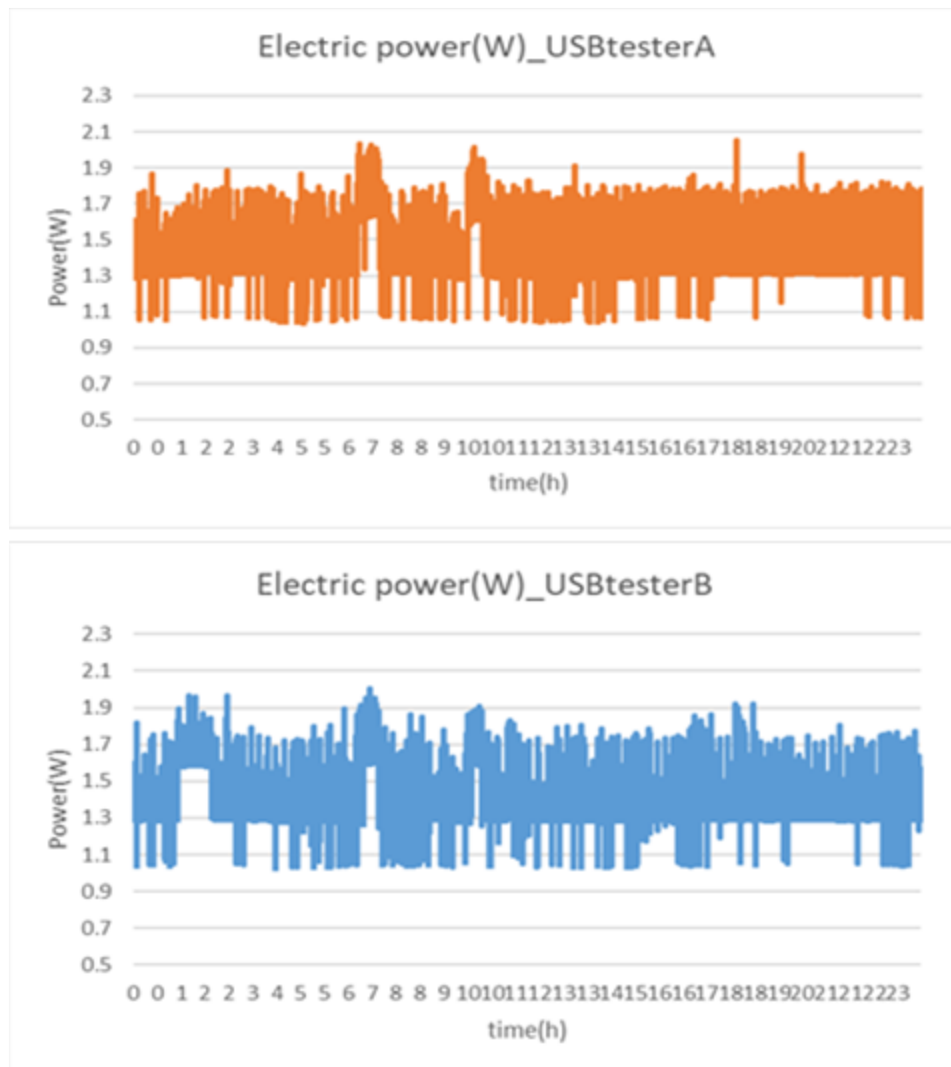


Figure 26 - Power consumption without port knocking on both test devices.

Confirming power consumption with port knocking:

After ensuring we have stable readings without the port knocking, we then enabled stream-cipher-based port knocking on the IoT device connected to the USB Tester-A only. We measured the power consumption of both IoT devices for over seven weeks (52 days), from August 3 thru 10, 2020 and from October 16 thru November 30, 2020. We also observed the total number of packets received by each IoT device by running the “netstat–statistics” command every hour. The sample of data collected in the first week is shown by the graphs in **Figures 27 and 28**.

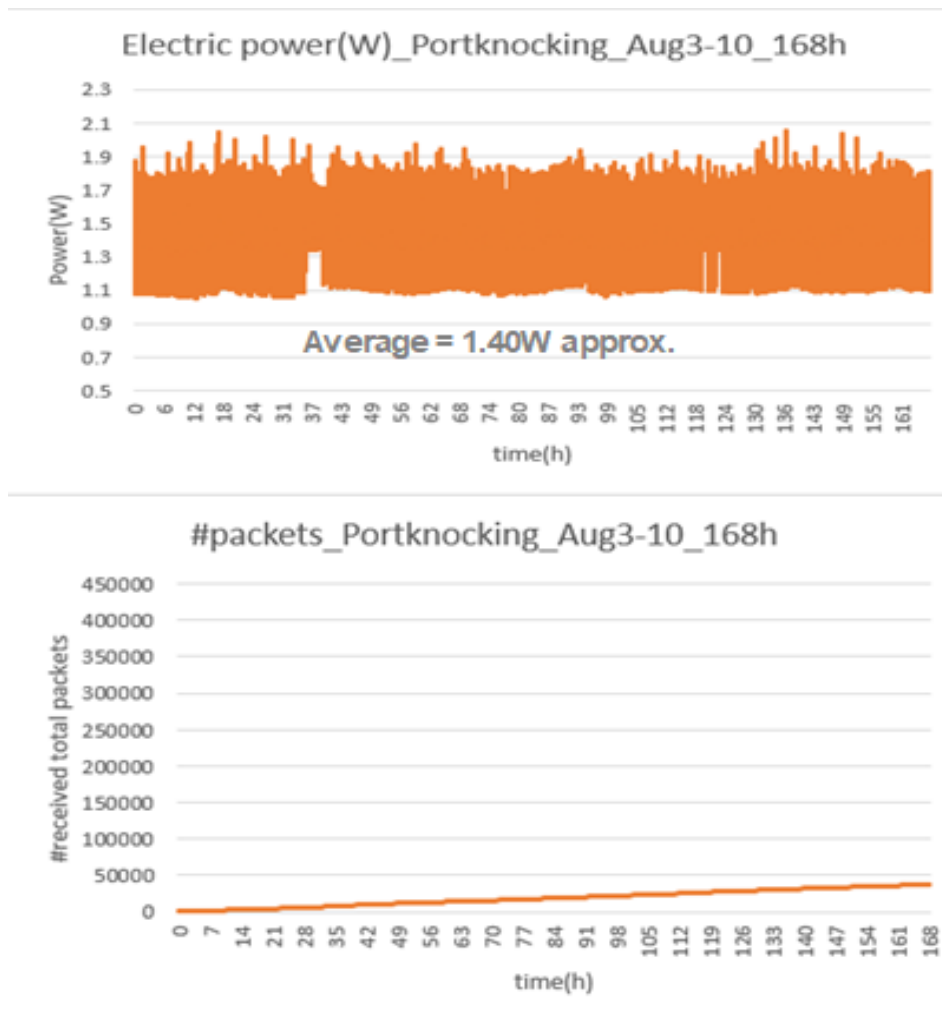


Figure 27 - Power consumption and packets received with port knocking (first week).

The first week of data collected showed that the IoT device without the port knocking feature had an increase of 0.17W in its power consumption (1.52W) from its baseline. Whereas the one with the port knocking feature had only a minimal increase of 0.02W (1.40W).

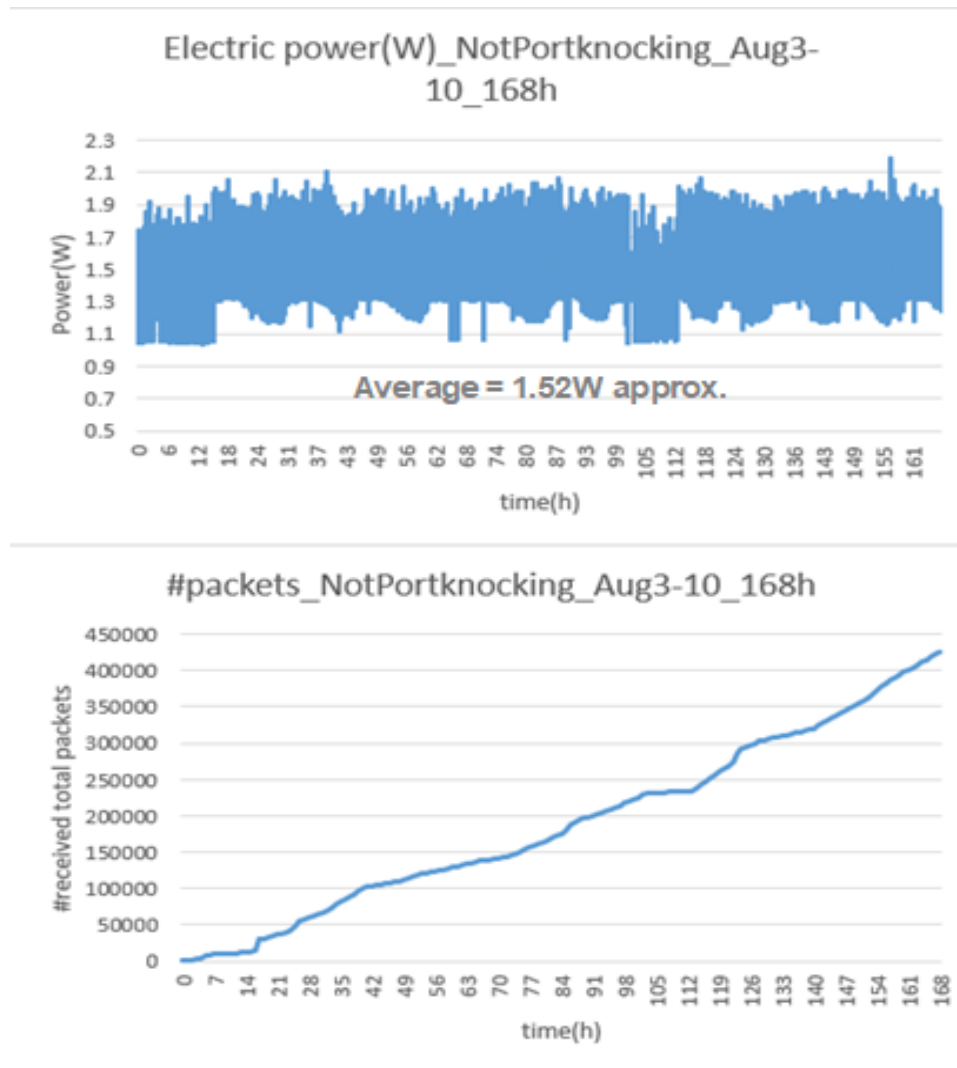


Figure 28 - Power consumption and packets received without port knocking (first week).

The results showed that the power consumption is directly proportional to the packets received. Since the IoT device connected to the USB Tester-A was running the port knocking feature, therefore, the number of packets received on its port 22/TCP was

significantly less compared to the device without the port knocking feature.

Another data sample collected in later weeks shows the same trend of receiving a very high number of packets on the exposed port compare to a significantly lesser number of packets on the stealthily hidden port, as shown in **Figures 29 and 30**.

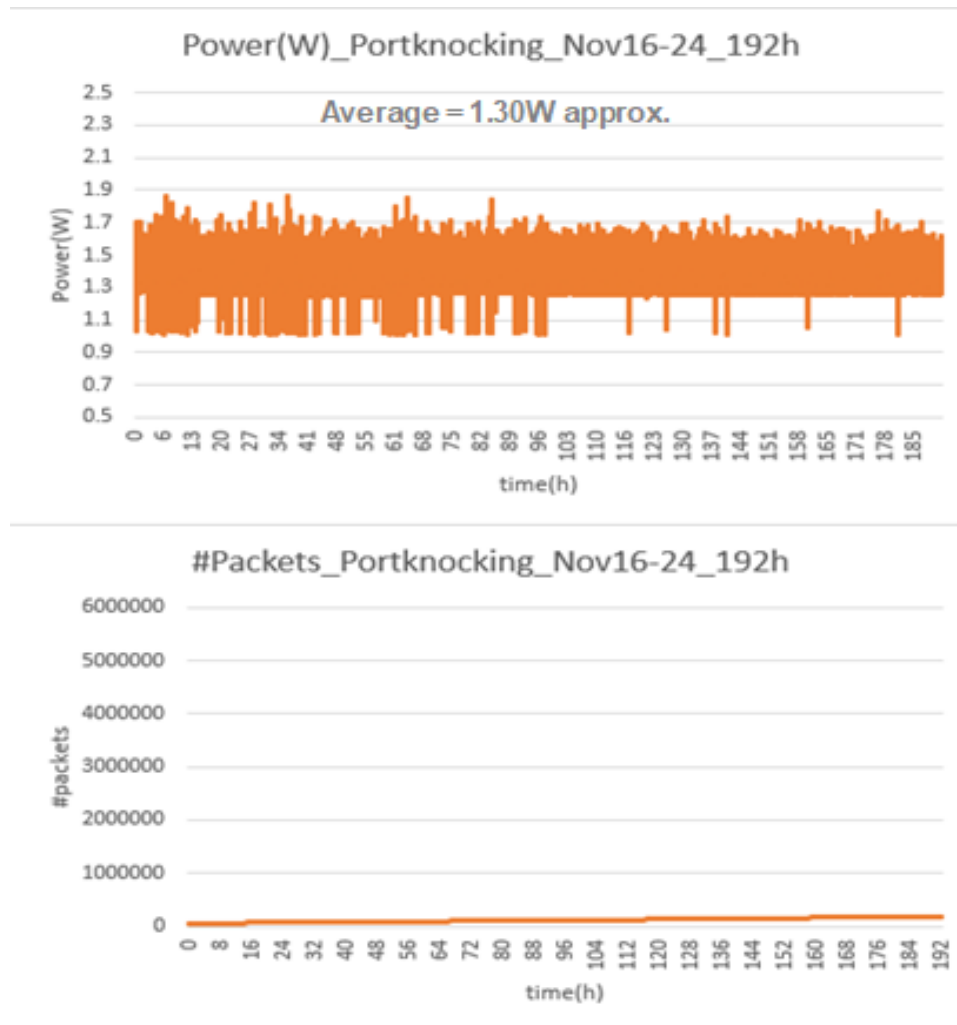


Figure 29 - Power consumption and packets received with port knocking (later on).

The power consumed by the IoT device without the port knocking was around 1.50W. Whereas on the one with the port knocking security feature, it was decreased to 1.30W over the same period of time. This shows a difference of approx. 0.20W between them. The maximum difference observed so far has been an average of 0.25W.

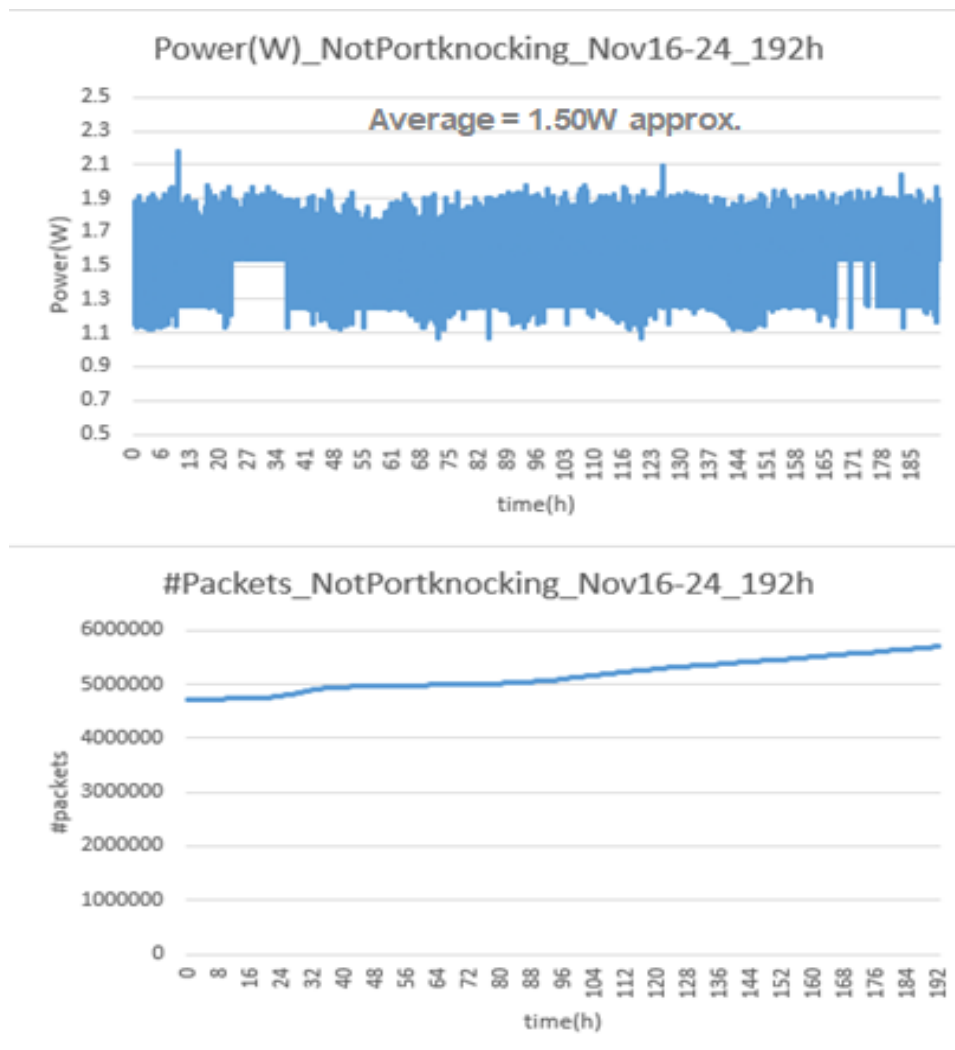


Figure 30 - Power consumption and packets received without port knocking (later on).

Hence, the power consumption observed was always less on the IoT device with the stealth port knocking feature than the one with no such feature. The gap between them slowly increases with time as we observed an average of 0.12W difference in the first week of testing (Figures 27 and 28), but then it grew to an average of 0.25W in the later weeks (Figures 29 and 30) due to the longer exposure of visible SSH default port to the Internet.

This indicates that the longer we use the IoT device with a port knocking feature, its power consumption decreases further compare to the one without the port knocking

feature. The stealthy nature of the port knocking method keeps the default SSH port hidden from the Internet and makes the IoT device not respond to any scans or inquiries for that service port unless the correct knock sequence is received. As a result, the IoT device receives a lesser number of packets than the one with an open service port exposed to the Internet.

5.6 Discussion

Based on our test results, we can summarize the answers to our research questions as follows:

- (a) The CPU consumption test showed that when the stealth port knocking feature was implemented using the stream cipher with AEAD algorithm on the IoT device, it would add a maximum overhead of 15% of the CPU power. Whereas, the PRNG-CRNG algorithm would add an overhead of 50% of the CPU power.
- (b) The security effectiveness test (by exposing directly to the Internet via 3G without any other security layers) showed that the IoT device with the stealth feature protecting the SSH service running on the default port was able to block all unwanted accesses for 42 days, while the unprotected services received 431,142 requests from 5,424 hosts in these 42 days.
- (c) The power consumption test showed that the IoT device with the stealth port knocking feature would actually decrease the power consumption compare to the one without such feature because of receiving a lesser number of packets due to the hidden service. The gap between them

slowly increases with time as we observed an average of 0.12W difference in the first week, but then it grew to an average of 0.25W in the later weeks due to longer exposure of visible SSH default port to the Internet.

In the case of port knocking, the stream-cipher-based algorithm was much practical to use on resource-constrained IoT devices. It not only kept the CPU usage to a very reasonable level and hid the default service port effectively but also helped in greatly reducing the unauthorized traffic coming to that service port. Thereby it helps the resource-constraint IoT device in consuming less power. This could provide a secure remote management option for authorized users without causing much overhead on existing resources of the IoT device. On the other hand, without the stealth port knocking feature, the visible port 22/TCP responded to all the incoming SYN packets that allowed the scanning hosts to follow up with other packets for completing the TCP handshake, enabling attempts to establish or exploit the SSH service. Thereby, receiving more packets to process and consuming more power than its counterpart that was running the stealth port knocking security feature.

Hence, we can conclude that the experimental results imply that the power consumption overhead by receiving incoming session requests (from scanners/malware on the Internet) would easily exceed the power consumption for running port knocking service. Therefore, running the stealth port knocking feature would be beneficial in terms of not only security enhancement but also power consumption.

5.6.1 Limitations and Considerations

The current study was carried out by testing the port knocking feature only for the SSH service using key-authentication on the default port 22/TCP. We have tested it on the

raspberry-pi hardware platform for determining the effectiveness of hiding port from unauthorized traffic (from scanners/malware on the Internet) and its effect on the IoT device's power consumption, with and without the stealth port knocking security feature. Our scenario is applicable to those IoT devices that are directly connected to the Internet using high-speed cellular networks. Careful considerations were given to ensure we benchmark and calibrate measuring devices before collecting the data to avoid any external influence. Though this proposed solution has shown encouraging results, however, we have not tested it for other services and non-default ports. Also, when considering the defense-in-depth approach, what other security methods can be applied and the choice of security layers must take into account the available resources, as we saw around 50% CPU utilization in the case of PRNG-CRNG based port knocking solution with our resource-constrained IoT device.

5.7 Summary

This study has provided us some promising results for using the port knocking security concept with which we can provide a secure channel for the remote management of an IoT device using the SSH service without exposing it to unwanted traffic. This method also decreases the total number of packets received by the IoT device on the hidden ports compare to the device without the port knocking feature, which helps in maintaining a lower power consumption. This feature is also being tested in a real-life scenario under the EU-Japan M-Sec project for securing smart cities [W-1].

CHAPTER 6: CONCLUSIONS

In this dissertation, we examined the issues related to the IoT devices and highlighted one of the key issues of lack of user awareness about security and privacy management when it comes down to the IoT devices under their use. The observational study helped in demonstrating this through cyber-peeping experiment. The results were shared through publication and TV broadcasts in order to raise the awareness. A similar challenge existed with lack of user knowledge and management on resource-constrained IoT devices, for which we proposed a countermeasure based on the stealth security port knocking feature. The conclusions from this dissertation and possible further directions for future work are presented in the following sections.

6.1 Concluding Remarks

- An IP camera depicts common IoT devices that are exponentially connected to the Internet. Through this study, we were able to understand what triggers an increase in illegal accesses and how the attackers obtain information from the devices providing a service over the Internet. Based on the response from the device, they further explore or exploit.
- In the observational study, two scenarios for observation of peeps were tested. First one by two URL reflecting type decoy cameras and second by five living-room cameras. The results of this study provided us a better understanding on peeping methods through IP cameras, both via automation and human involvement. The techniques helped us in

understanding the risks and dangers of using IP cameras with no/default or weak access authentication.

- This study also showed how public websites showing easily accessible IP cameras can drastically increase number of peeps into those IP cameras in the real world.
- Similarly, we were able to confirm that secondary information in viewable areas in front of IP cameras or reflected background information (URL and ID/password as bait via the decoy camera in our case) in IP camera images can be used by a peeping tom to further exploit and gain additional access, thereby exposing the dangers of using IP cameras.
- In the case of stealth security port knocking feature as a countermeasure, the stream-cipher-based algorithm was much practical to use on the resource-constrained IoT devices. It not only kept the CPU usage to a very reasonable level and hid the default service port effectively, but it also helped in greatly reducing the unauthorized traffic coming to that service port. Thereby it helps the resource-constraint IoT device in consuming less power. This could provide a secure remote management option for authorized users without causing much overhead on existing resources of the IoT device.
- On the other hand, without the stealth port knocking feature, the visible port 22/TCP responded to all the incoming SYN packets that allowed the scanning hosts to follow up with other packets for completing the TCP handshake, enabling attempts to establish or exploit the SSH service. Thereby, receiving more packets to process and consuming more power

than its counterpart that was running the stealth port knocking security feature.

- Therefore, this study has provided us some promising results for using the port knocking stealth security concept with which we can provide a secure channel for the remote management of an IoT device using the SSH service without exposing it to unwanted traffic. This method also decreases the total number of packets received by the IoT device on the hidden ports compare to the device without the port knocking feature, which helps in maintaining a lower power consumption.

Hence, as a result of observation through IP camera honeypot, many examples of peeping into the decoy cameras were confirmed in reality and were shared publically, within ethical considerations for the camera manufacturers, emphasizing the dangers associated with IP cameras. This study can further help in improving the security and awareness on the dangers associated with IP cameras. Similarly, we can conclude that the empirical results imply that the power consumption overhead by receiving incoming session requests (from scanners/malware on the Internet) would easily exceed the power consumption for running port knocking service. Therefore, running the stealth port knocking feature would be beneficial in terms of not only security enhancement but also power consumption.

6.2 Future Directions

- The observational study utilized different devices in two types of environments to study the actual state of peeping by humans and automated

accesses. Therefore, it will be interesting to further broaden this study by utilizing same devices in multiple observational environments.

- Additionally, since the observational study was limited to confirming access attempts (by rejecting access and logging attempt only) to the reflected-URL web server with the login info from the peeped decoy cameras, it would be interesting to further carry out a detailed behavioral study for observing the actions of peeping hosts after successfully accessing a reflected URL.
- The stealth security port knocking feature can also be coupled with other lightweight security options to provide a layered defense (defense-in-depth) approach for the resource-constraint IoT devices.
- Newly developed port knocking algorithms should be tested to confirm their impact on CPU usage.
- Expanding the effectiveness of this stealth port knocking feature with other services running on the non-default TCP/UDP ports as well as with multiple services at the same time is recommended to provide more tested options available for the effective use of the port knocking feature.
- As we have tested the concept of IoT devices being directly exposed to the Internet via the high-speed 3G cellular connection (instead of being behind a router/gateway firewall), therefore, future verification options can include the forthcoming smart cellular IoT devices having a direct 5G high-speed connectivity to the Internet.

BIBLIOGRAPHY

- [1] Jaramillo, L.E.S.: Malware detection and mitigation techniques: Lessons learned from Mirai DDOS attack, *Journal of Information Systems Engineering & Management*, Vol.3, No.3, Article No.19 (2018), available from <<https://doi.org/10.20897/jisem/2655>> (accessed 2020-07-14).
- [2] Perrone, G., Vecchio, M., Pecori, R. and Giaffreda, R.: The Day after Mirai: A survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices, *Proc. 2nd International Conference on Internet of Things, Big Data and Security (IoT BDS'17)*, pp.246-253 (2017).
- [3] Loshin, P.: Details emerging on Dyn DNS DDoS attack, Mirai IoT botnet, TechTarget Network (2016), available from <<http://searchsecurity.techtarget.com/news/450401962/Details-emerging-on-Dyn-DNSDDoS-attack-Mirai-IoT-botnet>> (accessed 2020-07-14).
- [4] Symanovich, S.: The future of IoT: 10 predictions about the Internet of Things, Norton Inc. (2019), available from <<https://us.norton.com/Internetsecurity-iot-5-predictions-for-the-future-of-iot.html>> (accessed 2020-10-21).
- [5] Wadhwani, P. and Gankar, S.: IP Camera Market Share Forecast 2025 - Industry Size Report, Global Market Insights Inc. (2019), available from <<https://www.gminsights.com/industry-analysis/ip-camera-market>> (accessed 2020-03-30).
- [6] Insecam, available from <<http://www.insecam.org>> (accessed 2019-02-25).
- [7] Shodan, available from <<https://www.shodan.io>> (accessed 2018-11-11).
- [8] Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T. and Rossow, C.: IoT POT: Analyzing the Rise of IoT Compromises, *Proc. 9th USENIX Workshop on Offensive Technologies (WOOT'15)* (2015).
- [9] Suzuki, S., Pa, Y.M.P., Ezawa, Y., Tie, Y., Nakayama, S., Yoshioka, K. and Matsumoto, T.: Improving IoT POT for Observing Various Attacks Targeting

Embedded Devices, *IEICE Technical Report, ICSS2015-47*, Vol.115, No.488, pp.1-6 (2016).

- [10] Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T. and Rossow, C.: IoTPOT: A Novel Honeypot for Revealing Current IoT Threats, *Journal of Information Processing*, Vol.24, No.3, pp.522–533 (2016).
- [11] Ezawa, Y., Tamiya, K., Nakayama, S., Tie, Y., Yoshioka, K. and Matsumoto, T.: An Analysis of Attacks Targeting WebUI of Embedded Devices by Bare-Metal Honeypot, *Proc. Computer Security Symposium 2017 (CSS2017)*, pp.211-217 (2017).
- [12] Guarnizo, J.D., Tambe, A., Bhunia, S.S., Ochoa, M., Tippenhauer, N.O., Shabtai, A. and Elovici Y.: SIPHON: Towards scalable high-interaction physical honeypots, *Proc. 3rd ACM Workshop on Cyber-Physical System Security (CPSS'17)*, pp.57–68 (2017).
- [13] Stelma, J.: *Securing the Home Network*, Master thesis, Eindhoven University of Technology (2015), available from <<https://pure.tue.nl/ws/files/47037062/799535-1.pdf>> (accessed 2019-03-18).
- [14] Bogaard, C.V.: Security analysis of cloud-based video cameras (2017), available from <<https://pdfs.semanticscholar.org/17cb/8f89320c9ca31d93b0e9e8ede68b6d03ff74.pdf>> (accessed 2020-03-30).
- [15] Smith, M.: Peeping into 73000 unsecured security cameras thanks to default passwords, CSO Online (2014), available from <<https://www.csoononline.com/article/2844283 /peeping-into-73-000-unsecured-security-cameras-thanks-to-default-passwords.html>> (accessed 2019-03-18).
- [16] Schneier, B.: The Internet of Things is wildly insecure and often un-patchable (2014), available from <<https://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem>> (accessed 2020-03-30).
- [17] Witkowski, W.: Intel admits vulnerability, but plays down effects; stock slides, amid gains (2018), available from <<https://www.marketwatch.com/story/intel->

stock-headed-for-worst-day-in-more-than-a-year-amd-pops-on-chip-design-flaw-report-2018-01-03> (accessed 2020-03-30).

- [18] Andreatos, A.S.: Hiding the SSH port via smart Port Knocking. *International Journal of Computers*, Vol.11, pp.28-31 (2017).
- [19] Santis, F.D., Schauer, A. and Sigl, G.: ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications, *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE '17)*, pp.692-697 (2017).
- [20] Zulick, J.: Latest Advances and Applications in the IoT Technology. Compare the Cloud (2019), available from <<https://www.comparethecloud.net/articles/latest-advances-and-applications-in-the-iot-technology>> (accessed 2020-03-30).
- [21] Lueth, K.L.: State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time. IoT Analytics (2020), available from <<https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time>> (accessed 2020-03-30).
- [22] Jonsson, P., Carson, S., Blennerud, G., Shim, J.K., Arendse, B., Hussein, A. and Ohman, K.: Ericsson Mobility Report, Mobility Reports (2019), available from <<https://www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf>> (accessed 2020-03-30).
- [23] EMnify: Cellular IoT: What Business Leaders Should Know, IoT & M2M Connectivity Management Platform, EMnify (2020), available from <<https://www.emnify.com/en/resources/cellular-iot>> (accessed 2021-05-25).
- [24] Staff, D. R.: IoT Attacks Up Significantly in First Half of 2019, Dark Reading (2019), available from <<https://www.darkreading.com/attacks-breaches/iot-attacks-up-significantly-in-first-half-of-2019/d/d-id/1336096>> (accessed 2020-03-30).
- [25] Kuskov, V., Kuzin, M., Shmelve, Y., Makrushin, D. and Grachev, I.: Honeypots and the Internet of Things, Securelist by Kaspersky (2017), available from

<<https://securelist.com/honeypots-and-the-internet-of-things/78751>>
(accessed 2020-03-30).

- [26] Seals, J.: Cyber Actors Use Internet of Things Devices as Proxies for Anonymity and Pursuit of Malicious Cyber Activities, Internet Crime Complaint Center (IC3) (2018), available from <<https://www.ic3.gov/media/2018/180802.aspx>> (accessed 2019-03-21).
- [27] Glastoph, Web Application Honeypot, available from <<https://github.com/mushorg/glastopf>> (accessed 2018-11-21).
- [28] Dionaea, Low Interaction Honeypot, available from <<https://github.com/rep/dionaea>> (accessed on 2018-11-21).
- [29] Demeter, D., Preuss, M. and Shmelev, Y.: IoT: a malware story, Securelist by Kaspersky (2019), available from <<https://securelist.com/iot-a-malware-story/94451>> (accessed on 2018-11-21).
- [30] Henke, C.: IoT attacks, hacker motivations and recommended countermeasures, IoT & M2M Connectivity Management Platform, EMnify (2020), available from <<https://www.emnify.com/en/resources/mirai-stuxnet-silexiot-attacks-hacker-motivations-and-recommended-countermeasures>> (accessed 2021-05-25).
- [31] Bischoff, P.: Hackers attack unsecured computers 70 times per minute: what happens when a computer is left unsecured on the internet [web log], Information Security, Comparitech (2021), available from <<https://www.comparitech.com/blog/information-security/honeypot-computer-study>> (accessed 2021-05-25).
- [32] Sel, D., Totakura, S.H. and Carle, G.: sKnock: port knocking for masses, *Proc. IEEE 35th Symposium on Reliable Distributed Systems Workshops (SRDSW'16)*, pp.1-6 (2016).
- [33] Seidel, U.: TCP stealth hides open ports, ADMIN - Network and Security (2015), available from <<https://www.admin-magazine.com/Archive/2015/26/TCP-Stealth-hides-open-ports>> (accessed 2020-08-19).

- [34] deGraaf, R., Aycock, J. and Jacobson, M.: Improved port knocking with strong authentication, *Proc. 21st Annual Computer Security Applications Conference (ACSAC'05)*, pp.451-462 (2005).
- [35] Al-Bahadili, H. and Hadi. A.H.: Network security using hybrid port knocking, *International Journal of Computer Science and Network Security (IJCSNS)*, Vol.10, No.8, pp.8-11 (2010).
- [36] Mehran, P., Reza, E.A. and Laleh, B.: SPKT: secure port knock-tunneling, an enhanced port security authentication mechanism, *Proc. IEEE Symposium on Computers & Informatics (ISCI'12)*, pp. 145-149, (2012).
- [37] Ramakrishna, C., Kumar, G.K., Reddy, A.M. and Ravi, P.: Survey on various IoT attacks and its countermeasures, *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, Vol.5, No.4, pp.143-150 (2018).
- [38] Deogirikar, J. and Vidhate, A.: Security attacks in IoT: A survey, *Proc. International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC'17)*, pp.32-37 (2017).
- [39] Sathyadevan, S., Vejesh V., Doss, R. and Pan, L.: Portguard - an authentication tool for securing ports in an IoT gateway, *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'17)*, pp.624-629 (2017).
- [40] Mahbooba, B. and Schukat, M.: Digital certificate-based port knocking for connected embedded systems, *Proc. 28th Irish Signals and Systems Conference (ISSC'17)*, pp.1-5 (2017).
- [41] Arifianto, R., Sukarno, P. and Jadied, E.: An SSH honeypot architecture using port knocking and intrusion detection system, *Proc. 6th International Conference on Information and Communication Technology (ICoICT'18)*, pp.409-415 (2018).
- [42] Yutanto, H.: Security Intelligence for Industry 4.0: Design and implementation, *Theoretical & Applied Science*, Vol.9, No.65, pp.228-243, ISSN 2308 – 4944

(2018).

- [43] ChaCha20 and Poly1305 for IETF Protocols (RFC 7539), available from <<https://tools.ietf.org/html/rfc7539>> (accessed 2020-06-15).
- [44] KDDI Research Inc.: Security analysis of ChaCha20-Poly1305 AEAD, *Cryptography Research and Evaluation Committees*, pp.32-33 (2016), available from <<https://www.cryptrec.go.jp/exreport/cryptrec-ex-2601-2016.pdf>> (accessed 2020-07-14).
- [45] Zaidi, A., Branneby, A., Nazari, A., Hogan, M. and Kuhlins, C.: Cellular IoT in the 5G era, Ericsson (2020), available from <<https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-iot-in-the-5g-era>> (accessed 2020-11-14).
- [46] How safe are home security systems? An HP study on IoT security (2015), available from <https://craigsmith.net/wp-content/uploads/2016/05/IoT_Home_Security_Systems.pdf> (accessed 2019-05-18).
- [47] Vivotek IP Cameras - RTSP Authentication Bypass: HelpSystems (2013), available from <<http://www.coresecurity.com/advisories/vivotek-ip-cameras-rtsp-authentication-bypass>> (accessed 2019-02-17).
- [48] Donohue, B.: Urban surveillance camera systems lacking security [web log], Kaspersky (2015), available from <<https://blog.kaspersky.co.jp/urban-surveillance-not-secure/7781>> (accessed 2019-02-25).
- [49] CVE-2017-5674: National vulnerability database, NIST (2017), available from <<https://nvd.nist.gov/vuln/detail/CVE-2017-5674>> (accessed 2018-12-20).
- [50] Censys, available from <<https://censys.io>> (accessed 2019-01-25).
- [51] Yang, Z., Xiong, J., Tie, Y., Tamiya, K., Nishida, S., Yang, D., Fujita, A., Yoshioka, K. and Matsumoto, T.: Observation and Analysis of Cyber Attacks in Home Network Testbed, *Proc. Computer Security Symposium 2017 (CSS2017)* (2017).
- [52] AS number assignments: Japan Network Information Center (JPNIC), available from <<https://www.nic.ad.jp/ja/ip/asnumber.html>> (accessed 2018-11-11).

- [53] Safari - Apple Support, available from <https://support.apple.com/ja_JP/downloads/safari> (accessed 2018-11-14).
- [54] Internet Explorer - Microsoft Download Center, available from <<https://www.microsoft.com/ja-jp/download/Internet-explorer.aspx>> (accessed 2018-11-14).
- [55] Home Gadgets at Risk: NHK Documentary, Season 2018, Episode 4 (2018), available from <<https://www6.nhk.or.jp/special/detail/index.html?aid=20171126>> (accessed 2019-03-18).
- [56] Home Gadgets at Risk: Science ZERO, NHK Educational TV, (2017), available from <<https://www2.nhk.or.jp/archives/chronicle/pg/page010-01-01.cgi?recId=0001000000000000%400000000000000000000000%2D50%2D21%2D4800000000000000000000>> (accessed 2019-03-18).
- [57] Hung, M.: Leading the IoT, Gartner Inc. (2017), available from <https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf> (accessed 2020-07-14).
- [58] Borges, E.: Top 20 and 200 most scanned ports in the cybersecurity industry (2019), available from <<https://securitytrails.com/blog/top-scanned-ports>> (accessed 2020-10-21).
- [59] Soro, S. and Heinzelman, W.: A Survey of Visual Sensor Networks, *Advances in Multimedia*, pp.1–21 (2009), available from <<https://doi.org/10.1155/2009/640386>> (accessed 2020-06-24).
- [60] Shin, L.: A Security Guide to IoT-Cloud Convergence, *Security News* (2020), available from <<https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/a-security-guide-to-iot-cloud-convergence>> (accessed 2021-05-25).
- [61] Ali, F.H.M., Yunos, R. and Alias, M.A.M.: Simple port knocking method: Against TCP replay attack and port scanning, *Proc. International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec '12)*, pp.247-252 (2012).
- [62] Vasserman E.Y., Hopper N., Laxson J. and Tyra J.: SilentKnock: practical,

provably undetectable authentication (2007), available from <https://doi.org/10.1007/978-3-540-74835-9_9> (accessed 2020-08-19).

- [63] Khan, Z.A, Javaid, N., Arshad, M. H., Bibi, A. and Qasim, B.: Performance evaluation of widely used port knocking algorithms, *Proc. IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC'12)*, pp.903-907 (2012).
- [64] Oxford Advanced Learner's Dictionary: Stealth, stealth_1 noun - Definition, pictures, pronunciation and usage notes, available from <https://www.oxfordlearnersdictionaries.com/definition/english/stealth_1?q=Stealth> (accessed 2021-05-25).
- [65] Cambridge University Press: Stealth, Cambridge English Dictionary, available from <<https://dictionary.cambridge.org/dictionary/english/stealth>> (accessed 2021-05-25).
- [66] Kereki, F.: Implement port knocking security with Knockd, Linux Journal (2010), available from <<https://www.linuxjournal.com/magazine/implement-port-knocking-security-knockd>> (accessed 2020-06-24).
- [67] ChaCha20 and Poly1305 for IETF Protocols (RFC 8439), available from <<https://tools.ietf.org/html/rfc8439>> (accessed 2020-06-22).
- [68] Tex2e/ChaCha20-Poly1305 – GitHub, available from <<https://github.com/tex2e/chacha20-poly1305>> (accessed 2020-06-25).
- [69] OpenBlocks IoT VX2: Plat'Home (2019), available from <<https://www.plathome.co.jp/product/openblocks-iot/vx2>> (accessed 2020-06-30).
- [70] OpenBlocks IoT BX1: Plat'Home (2019), available from <<https://www.plathome.co.jp/product/openblocks-iot/bx1>> (accessed 2020-06-30).

PUBLICATIONS

Peer-reviewed Paper in Journals

- [J-1] Tamiya, K., Bokhari, A. H., Ezawa, Y., Nakayama, S., Tie, Y., Tanabe, R., Fujita, A., Yoshioka, K, and Matsumoto, T.: Dangers of IP camera – An observational study on peeping. *Journal of Information Processing*, Vol.28, pp.502-510 (2020), available from <[https://doi.org /10.2197 /ipsjjip.28.502](https://doi.org/10.2197/ipsjjip.28.502)> (accessed 2021-03-14).
- [J-2] Bokhari, A. H., Inoue, Y., Kato, S., Yoshioka, K, and Matsumoto, T.: Empirical analysis of security and power-saving features of port knocking technique applied to an IoT device, *Journal of Information Processing*, Vol.29 (2021).

Technical Paper

- [T-1] Inoue, Y., Kato, S., Bokhari, A. H., Yoshioka, K, and Matsumoto, T.: Empowering Resource-constraint IoT Gateways with Port Knocking Security, *Proc. Computer Security Symposium 2020 (CSS2020)*, pp.362-367 (2020).

Workshop

- [W-1] Bokhari, A.H.: Multi-layered Security Technologies for Hyper-connected Smart Cities, *YEIS International Forum* (2018).

Awards

- 2020 [J-1] “Specially Selected Paper” (特選論文) Certificate – *Journal of Information Processing* (September, 2020).
- 2021 [J-2] “Specially Selected Paper” (特選論文) Certificate – *Journal of Information Processing* (September, 2021).