Fending off IoT-Hunting Attacks at Home Networks

Vincentius Martin
Duke University

Qiang Cao
Duke University

Theophilus Benson
Brown University

ABSTRACT

Many attacks target vulnerabilities of home IoT devices, such as bugs in outdated software and weak passwords. The home network is at a vantage point for deploying security appliances to deal with such IoT attacks. We propose a comprehensive home network defense, Pot2DPI, and use it to raise an attacker's uncertainty about devices and enable the home network to monitor traffic, detect anomalies, and filter malicious packets. The security offered by Pot2DPI comes from a synthesis of practical techniques: honeypot, deep packet inspection (DPI), and a realization of moving target defense (MTD) in port forwarding. In particular, Pot2DPI has a chain of honeypot and DPI that collects suspicious packet traces, acquires attack signatures, and installs filtering rules at a home router timely. Meanwhile, Pot2DPI shuffles the mapping of ports between the router and the devices connected to it, making a targeted attack difficult and defense more effective. Pot2DPI is our first step towards securing a smart home.

CCS CONCEPTS

• **Security and privacy** → *Network security*; *Intrusion/anomaly detection and malware mitigation*; *Vulnerability management*;

KEYWORDS

Home network, IoT, Honeypot, Moving target defense

ACM Reference Format:
Vincentius Martin, Qiang Cao, and Theophilus Benson. 2017. Fending off IoT-Hunting Attacks at Home Networks. In *CAN'17: Cloud-Assisted Networking Workshop, December 12, 2017, Incheon, Republic of Korea.* ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3155921.3160640

1 INTRODUCTION

There is a tremendous growth in the use of IoT devices in connected home networks, with studies [12] projecting that by 2020 there will be over 1.1 billion smart home devices shipped to develop smarter communities. However, these devices increasingly become a target of various attacks that hunt for precious assets from them, such as computing and networking resources a device can have [9] and sensitive data a device may hold [17]. Compromised devices can then be turned into fleets of bots and launch large attacks, with the potential to cause record-breaking damage [10].

We investigate the problem of combating the attacks hunting home IoT devices. If we can prevent the enormous number of such devices from being compromised, it will substantially reduce the destroy power of attackers and even make it impossible for them to launch large IoT-based attacks.

While the Internet has long been a target of attacks, the security of home IoT devices is exacerbated by multiple factors. First, home IoT devices are often backed by a manufacturer-operated cloud, e.g., Samsung's TV portal [7], and hence have external dependencies outside home. Under this operation model, IoT devices must open ports, through port forwarding, leaving the home network susceptible to attacks. Besides, devices such as Nest Cam open ports to maintain a service accessible from the Internet, which might lead to a security breach [14]. Second, as part of the current poor device management practice, a large portion of these IoT devices lack updated firmware or rely on default or weak passwords for authentication. The direct effect is that a password's ability to lock out unauthenticated users is substantially weakened. As a result, impersonation and password guessing against devices are less costly and can occur with a simple dictionary and a number of attempts (§2.1). These together constitute an insecure path through which an attacker can jeopardize the integrity of a poorly-secured device and even take over it.

This problem has received attention from both the research community and industry. Several research proposals [19, 27, 29] and commercial solutions [2, 4] adopted the idea of strengthening security through the home network. Since the home network is at a vantage position: interposing between the smart home devices and the Internet, it can send packet traces for inspection by virtualized appliances in the cloud [2, 4, 19] or manage security locally: patching software [27] and/or directing packets to micro network-security functions [29].

We seek to incorporate a few practical security techniques into the home network, with a goal to make attack prevention, detection, and reaction simpler and faster. In particular, a home router manages the connectivity of devices and oversees all their network communication. Our design seeks to encompass the governance from the home router, together with security appliances to form a concerted defense. At the same time, a modern home router is often shipped with a surplus of CPU and memory (e.g., Google OnHub [11]). If we deploy security appliances on the router, we can keep monitoring, detection, and filtering local, and eliminate the need of additional gadget installations and network changes.

We propose Pot2DPI, a system brings a line of in-depth defense against IoT-hunting attacks into home network, with assistance from a home router. It essentially embodies a security architecture for incorporating and managing security appliances as modular system units on a modern home router, in spirit similar to [27]. Pot2DPI focuses on appliances to monitor network environment, detect anomalies, fingerprint attacks, and filter out malicious packets. It deploys a lightweight honeypot to collect suspicious probes to the home network, uses DPI (e.g., Bro) to perform inspection on

CAN'17, December 12, 2017, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5423-3/17/12…$15.00
https://doi.org/10.1145/3155921.3160640

incoming packets, and places between them a signature acquisition module that extracts accurate signatures from the honeypot-captured packets. This appliance chain represents a reactive defense that detects and stops attacks using honeypot as a sensor and DPI as an actuator. As part of the concerted defense effort, we build into Pot2DPI another proactive defense that shuffles port mapping on the home router. This is a realization of moving target defense in that it increases the uncertainty on the actual ports an IoT device uses. The effect is multifold: attackers need resources to send out more probes; the home honeypot obtains more packet captures.

We summarize our contributions as follows.

- We design a workflow to monitor and stop attacks using honeypot, DPI, and a novel detection algorithm.
- We identify a way to incorporate the governance role of a home router into a moving target defense.
- We present preliminary evaluation on real malware traces.

2 BACKGROUND

This section provides an overview of IoT operating model, threat model, recent attacks targeting IoT devices, and existing defenses.

2.1 IoT operating model

Home IoT devices. There are broadly two classes of home IoT devices, categorized by the way the devices connect to the Internet: The first, hub-enabled, are IoT devices controlled and orchestrated through a hub, e.g., a home router or a Raspberry Pi. These devices have no direct access to the Internet. The second, IP-enabled, are devices that are directly connected to the Internet and their connectivity can be configured/managed directly. This class of devices often run a version of Linux optimized for embedded systems, e.g. BusyBox [1], and provide a web portal for direct configuration, e.g., CCTV, Nest, SmartTV, and SmartFridge. A hub used by "hub-enabled" devices itself is an "IP-enabled" device, e.g., a home router, or a device, e.g., Raspberry Pi, connected to an "IP-enabled" device. Both classes of devices may receive and send packets from and to the cloud to enable cloud-based orchestration tools and to allow remote access and management.

The IoT problem. Attackers are interested in home IoT devices because of the massive number of devices and the poor management of their security. Like other home appliances, many affordable IoT devices have reached a massive scale under the vision of smart homes. However, the insecure management of these devices turns them into lucrative targets to attackers who hunt for computing cycles, private data, etc. First, open ports for IoT management and external access increase the exposure of home devices to threats. IoT manufacturers, e.g, Samsung, often use a cloud-based approach to manage devices: outsourcing management to a cloud-backed portal to free users from directly managing devices. This requires to open ports in the home network to allow access from external management services. In addition, some home devices such as Nest Cam maintain open ports for external access according to the service model. Unfortunately, these ports are also open to attackers who may exploit them to inject malicious packets. Second, the current poor practice of credential (e.g., passwords) and software management on IoT devices opens up a large attack vector. Vulnerable firmware and software may not always been patched in a timely manner. Default and weak passwords make user authentication ineffective and can significantly weaken system security. For example, a householder may bring up an IoT device online before setting its own password. As a result, the IoT device runs under a default account with a default password. An attacker might craft packets and deliver them through an open port to guess a password. In a Mirai attack [9], an attacker cracks weak credentials using a simple dictionary of 62 pairs of account ID and password.

Home network. A home network is usually a network of devices connected by a home router. The home router acts as a gateway to the home network, interposing itself between all devices (IoT, mobile, and personal computers) and the wide-area network. Thus, the home router manages the network and oversees all network communication from outside to smart devices. The home router runs a software stack to keep devices connected according to the configuration and route traffic for them. The hardware and OS of a home router are trusted and secure.

While the home routers are not as powerful as cloud-grade servers, there are significant resources on a modern home router. For example, a OnHub router [11] boasts a 1GB RAM and a dual-core 1.4GHz processor. These routers are able to run virtualized Docker containers [8], run NFVs (e.g., DPI [3]), and run SDN-based virtual switches.

2.2 Threat model

In our threat model, an attacker seeks to compromise IoT devices connected to home networks. Compromised IoT devices can cause significant damages, such as leakage of sensitive data (e.g., video), disruption of important network services (e.g., DNS) by large attacks like DDoS, etc. An attack can target different vulnerabilities of IoT devices, such as those of outdated software, default or weak passwords, etc. We do not assume any limitation on an attacker's ability to obtain knowledge about vulnerabilities of various devices and attack them, but instead rely on the defense system to stop intrusion attempts. However, an attacker is restricted by the fact that it does not have physical access to a home, without the householder's permission. Therefore, an attacker does not have physical access to a home router or to home IoT devices, nor does it know how many devices are connected to the home network and what are they. The attacker needs to use probes to learn if there is a vulnerable IoT device on a certain port. These probes go through the home router before reaching any device. An attacker can accumulate its knowledge about port forwarding at a home network over time. But attaining such information requires probes and the knowledge can be out of date when the configuration of port forwarding changes.

2.3 Motivating IoT attacks

We use two recent IoT attacks, i.e., Mirai [10] and Persirai [14], to understand their characteristics, in terms of targeted devices, means of penetrating, severity of the subsequent attacks that could be launched from controlled devices. Since Persirai is a successor of Mirai, this also sheds light on how IoT attacks evolve over time.

Mirai focuses on BusyBox-based devices: the Mirai code scans the Internet for devices running open servers on port 23. It then

attempts to log into these IoT devices using a known list of weak passwords and account IDs, and installs the botnet malware on them after login. Compromised devices in turn repeat this process: scanning for other devices and compromising those with weak passwords.

Persirai [14] improves over Mirai. It targets the Universal Plug and Play (UPnP) protocol and exploits it to gain control of a vulnerable IP camera. IP cameras can use UPuP to plug into a home network and open a port (81) on the router, acting as a server. However, through this open port, Persirai instead injects malicious commands to IP cameras. Vulnerable cameras (e.g., suffering from authentication credential vulnerabilities [13]) then take the commands and download and install malware from a specified site.

In both Mirai and Persirai, attackers follow a common path to compromise IoT devices: they first seek for a reachable port opened by a device and then exploit the device vulnerabilities, e.g., password or software vulnerabilities, to take it over. Compromised devices can be further used to take over other devices or join a botnet to launch attacks, such as the DDoS attack against Dyn. The scale and magnitude of the Dyn cyberattack have reached a record, with a reported peak rate at 1.2 Tbps and involving 100,000 bots [10].

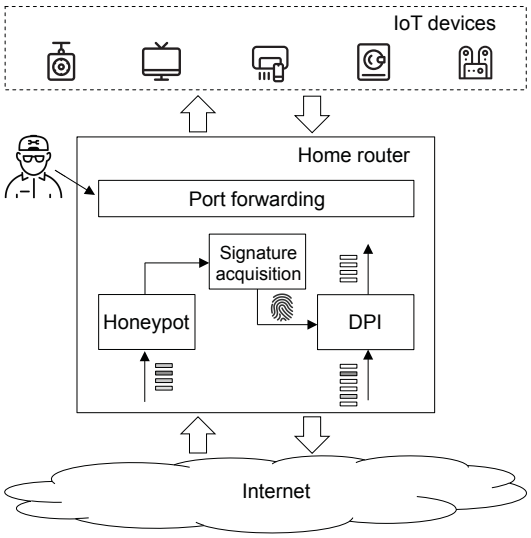

Figure 1: Architecture of Pot2DPI. Shaded rectangles represent packets carrying malicious traffic.

2.4 Existing home network-based approaches

Approaches and tools [2, 4, 19, 27, 29] have been proposed within the context of emerging IoT devices. [19] proposes to outsource security management of a home network to a trusted third party that has appropriate operation expertise. [4] and [2] are security products for a smart home, sharing the idea of outsourcing security with [19], but relying on each fixed service provider. [29] presents a network-centric approach that uses micro middleboxes as security gateways for IoT devices and orchestrates them through a logically centralized controller. [27] introduces an in-hub security manager to deal with software updates and malicious traffic. Pot2DPI extends the previous work in that it focuses on developing a comprehensive defense by combining the power of practical security techniques, i.e., honeypot, DPI, and a realization of moving target defense. The interplay among these techniques turns out to be unique and can significantly raise the bar for the security of home IoT devices.

3 COMPREHENSIVE SMART HOME DEFENSE

3.1 Vision

We envision a home router-assisted defense system guarding IoT devices. The objective of this system is to combat IoT attacks in a manner that makes prevention more effective, detection more accurate, and reaction on time. This is driven by our observations on home IoT devices, their security implications, and the typical home network environments where they are in use.

Identifiable attack patterns. IoT attacks are constrained by the vulnerabilities of the devices they could exploit and the way they exploit them, and thus the attacks are detectable. For example, Mirai (§2.3) exploits weakness of unchanged password setting on an IoT device; and Persirai (§2.3) leverages software vulnerabilities of certain IP cameras. These IoT defects leave the attack vector possible, but at the same time substantially restrict the form of an attack: attacking bots must be carefully implemented so that their behavior conforms to the intent to exploit the target IoT vulnerabilities. These behavior patterns in turn enable calibrated detection methods to discover and catch such attacks. Thus, a defense system with the ability to detect and react (e.g., using DPI) to distinct patterns can be effective in throttling IoT attacks after calibration.

Vantage point for threat monitoring and policy enforcement. Home networks are at a strategic position for providing security to IoT devices. Since IoT devices are often connected to a home network, attack traffic hunting for IoT devices must go through the home network. If we can strengthen the security of home network, we can let the network to filter out attack traffic before it reaches any IoT device. Such a home network-based approach can significantly simplify and speed up attack detection, prevention, and reaction, as most of the work is done locally within the home network. In particular, a householder often uses a home router as a gateway to the home network. The home router manages connectivity between internal endpoints, e.g., devices and laptops, and external hosts from the Internet. It also oversees all of their communication and routes packets for all the connected devices. Thus, the home router is a vantage point to implement security and cope with threats. We use it to interpose on the communication between the home network and the WAN, and bring in security without the cost of extended support from devices or other parties, e.g., ISPs. Pot2DPI also relies on it to control the configuration of port forwarding, implementing a line of moving target defense.

As discussed in §2.1, a modern home router is often shipped with a surplus of hardware resources. This work focuses on using these resources to provision a set of security appliances. In a home network with other devices, such as PC and Raspberry Pi, one can offload security workload to them when resources are available on these devices and latency can be tolerated.

3.2 Challenges

We are facing challenges in incorporating security into home networks. These challenges arise from the heterogeneity of IoT devices, the complexity of the attack vector, and the availability of computation resources at a home network for security.

Heterogeneity of IoT devices. IoT devices targeted by attackers are diverse, including smart lights, thermostats, smart speakers (with online voice services), etc. Honeypots for Pot2DPI need to react to attacking bots that may run different protocols for hunting different devices. Each such interaction can involve multiple rounds of communication between a stateful attacker and a honeypot, with payloads conforming to an application protocol. Honeypots should retain high fidelity while still run efficiently.

Weak attack signal. The signal that a home network obtains from captured traffic can be weak. For example, to discover potential vulnerabilities of IoT devices, an attacker may scan a range of network addresses. Because a home network is provisioned with limited network connectivity, it can only observe a small fraction of these scans. In addition, attacks can be conducted in a stealthy way and do not sustain for a period of time. The attack traffic captured at a home may be not comparable to the normal traffic handled by it within the same time frame.

Zero-day attacks. New attack signatures need to be delivered and installed at DPI in a timely manner. Home network is a dynamic environment. Attacks can be launched at any time. Once detected, a new attack signature should be installed on DPI as quick as possible. The system can also install filters based on external knowledge of vulnerabilities, such as CVEs.

Limited resources. A home network is constrained in resources, i.e., with bounded network bandwidth and computation resource. Pot2DPI needs to manage honeypots and DPI for efficient use of the available resources in the system. At the same time, it minimizes the interference with normal use of home devices.

4 DESIGN

Figure 1 shows the architecture of Pot2DPI. Pot2DPI runs on a home router as a defense in depth, using a combination of reactive and proactive subsystems to discover threats, sanitize traffic, and manage network connectivity. It contains four major system components: 1) a local honeypot to interact with intruders and collect their probing packets; 2) a signature acquisition module to extract packet level patterns, e.g., malware fingerprints, for recognition of exploit traffic; 3) a deep packet inspection (DPI) module to install filters for certain patterns and block traffic if pattern matches are identified; 4) a port manager to rotate port mapping between the home router and the IoT devices. The first three components aim to automate the workflow of monitoring, detecting, and reacting to various intrusion attempts. The use of the port manager further increases the difficulty that an attacker can identify an open port. It is a proactive defense that realizes the moving target defense (MTD) strategy (§4.4).

In the rest of this section, we elaborate the design of each module in Pot2DPI and show how they work together to identify threats and prevent intrusion.

4.1 Honeypot

Pot2DPI runs a honeypot to respond to intruders and record their probing packets, such as packets sent for port scan attacks [23]. Like previous work Honeycomb [22], we use the off-the-shelf low-interaction virtual honeypot daemon [5] as a basis for creating efficient and responsive honeypots. The honeypot monitors activities on the ports that are not supposed to receive any packets. Pot2DPI mediates the interaction between the honeypot and the port manager, so that the focus of the honeypot moves to a new set of ports when the port mapping for packet forwarding is rotated.

4.2 Attack signature acquisition

Honeypots capture suspicious packet traces through the network tap. These traces contain important information that characterizes attacks. For example, packets in a port scan attack can reveal the source IP addresses of the attackers; packets from a malware campaign can reveal fingerprints of the malware that is being propagated; packets sent over a botnet's command and control channel can reveal the mission of the attack.

The patterns carried in malicious packets are useful for fast decision making when devices are under attack. Once an attack pattern is identified, the system can instruct DPI to drop subsequent packets from the same flow. We therefore look for signatures from packet traces captured by a honeypot. We call a *signature* as a common pattern extracted from packets involved in an attack. There are a couple of technical obstacles to be addressed before we can acquire quality signatures: 1) because a packet includes a limited number of bits, the content (i.e., bits) of individual packets does not necessarily reflect the complete nature of the involved attack, e.g., a packet used in a malware campaign is not necessarily large enough to include the entire fingerprint of the malware; 2) although only suspicious packets get captured by a honeypot, these packets are not necessarily from a single attack, e.g., in the case of multiple concurrent attacks launched independently.

To compensate the insufficient evidence a single packet carries, we re-construct a flow from a stream of packets sent over the same network connection and set our signature acquisition on the granularity of connections, instead of packets. In particular, we concatenate the payload of each packet from the same connection according to the packet timestamp and use the resulting bit stream as a trace for its connection. To discover common patterns among connections, we employ the longest common subsequence algorithm (LCS) that searches through all connection traces and identifies common subsequences.

To cope with multiple independent attacks, we employ a preprocessing based on behavioral clustering to provide quality input for signature acquisition. Previous work on behavioral clustering targets http-based application protocols [26]. Our approach does not rely on the payload structure specific to http, but applies clustering analysis to the re-constructed connection traces. The output of this analysis is a set of clusters each of which contains a group of connections that are highly likely involved in the same attack.

4.3 Signature-based filtering

We use the detected attack signatures to stop malicious traffic by installing corresponding filters on the home router, e.g., via Bro.

Pot2DPI relies on deep packet inspection to find pattern matches and relies on filters to block unwanted traffic. One could discretionarily filter traffic based on packet origin, such as the source IP address. DPI such as Bro provides a script language to install filters for signatures, where signatures are in a generic form, i.e., regular expressions.

4.4 Port re-mapping

IoT devices may have open ports configured statically and can become constant targets of attacks, such as port-scan attack. If an attacker knows the open ports, it may seek to minimize the malicious traffic it needs to send out by carefully targeting the victim ports. By doing so it decreases the risk of being detected. We adopt the moving target defense (MTD) strategy and develop a proactive countermeasure. The objective of MTD in Pot2DPI is to re-map ports on the home router to those of the connected IoT devices. This raises the degree of uncertainty an attacker faces and forces it to use significantly more probes if it still hopes to find an open port. As a result, it not only imposes an increased cost on an attacker, but also makes such attacks less stealthy to any detection because more malicious traffic is exposed to the radar of the detection engine.

Port re-mapping does not need any changes to an IoT device, nor changes to the software stack running on it. Instead, we configure a home router with a set of NAT rules to implement a random mapping of ports. Additional management for port re-mapping in Pot2DPI involves notifications of a re-assigned open port to clients or applications that are outside the home network and need the access. Software updates and patches from outside are often pulled by an IoT device and thus the delivery remains unchanged with a known remote distribution server. Device communication within the home network is unaffected under port re-mapping.

5 EVALUATION

In this section, we present the results from preliminary evaluation of Pot2DPI, focusing on attack detection accuracy using real malware pcap traces.

Experiment setup. We implemented a hierarchical clustering algorithm that clusters captured packets based on the similarity of their payloads. It uses Levenshtein editing distance to quantify the difference between payloads. Each cluster of packets is then a basic unit for pattern extraction, where Pot2DPI uses the longest common subsequences algorithm (§4.2) to acquire the common subsequences as attack signatures. We then apply the obtained signatures to packet inspection. We use four malware pcap traces [15, 16] and one normal pcap trace collected from our IoT testbed for evaluation. These traces contain a varying number of packets and connections, as shown in Table 1. We use each pcap trace for signature extraction and use a mix of this trace and the normal trace for packet inspection. We report the number of packets and connections successfully detected by Pot2DPI for each trace.

Detection results. Table 1 shows the results on true positives when we use Pot2DPI to scan the four malware pcap traces and the normal trace from our testbed. We can see that accuracy is quite high for the first three malware traces. For malware Torrentlocker, Pot2DPI missed attack packets and connections (false negatives). When we test with a mix of malware and normal traces, we found false positives with malware Cerber's signatures.

Table 1: Pot2DPI detection results on malware traces.

Malware trace	Alman-trojan	Cerber	Fereit	Torrentlocker
Total pkts	49	1102	92	43
Detected pkts	48	1101	92	21
Total conns	26	1091	51	7
Detected conns	26	1091	51	2

We also compare to Honeycomb [22], a honeypot with signature extraction for intrusion prevention. We implement Honeycomb's horizontal analysis and use it for comparison. Honeycomb's horizontal analysis resulted in more false negatives on malware Alman-trojan and Fereit.

6 DISCUSSION

Regular expressions as a generic form of signatures. Payload of malicious packets can exhibit a large variance, even if these packets pertain to the same attack or campaign. Previous work Polygraph [24] showed that one needs multiple substrings to fingerprint worms, because worms can employ polymorphism and render a single-subtring signature ineffective. We advocate regular expressions as a standard way to represent an attack signature. First, regular expressions are more expressive than subsequence/substring signatures, as they can represent a superset of patterns than a combination of multiple substrings does. Yet, the expressiveness of regular expressions do not come at the cost of usability. Filters based on such signatures can be easily installed with DPIs such as Bro [3] and Snort [6]. Second, signatures with regular expressions can be efficiently extracted using existing approaches, such as methods in text mining. For example, machine learning techniques could be appropriately applied to packet traces for discovering regular-expression signatures [18].

Sharing packet traces and signatures. Suspicious packet traces collected by honeypots are valuable resources for uncovering threats. As we discussed in §3.2, the attack signal captured by a honeypot is weak, especially in face of stealthy attacks. One way to improve the utility of the captured packet traces is to share them among home networks. This enables collaborative attack detection and can overcome the limitations imposed by the narrow view of a single honeypot. For example, in order to get high efficacy, an attacker may program controlled bots to probe to many home networks. Sharing and correlating packet traces collected from different homes can reduce ambiguity at detection and improve the effectiveness of the resulting signatures. Further, sharing signatures can be a means of getting security warnings and vulnerability assessment at an early stage. However, the downside is traces or signatures may reveal sensitive information and sharing them can raise privacy concerns. The problem of private information leakage could be alleviated if we retain in the householders' hands tight control over what to share, e.g., choosing which traces or signatures to share with whom and even specifying policies that regulate what sensitive information to be removed before data release.

Mitigation of DoS attacks. Resources on a home router are not unlimited. Pot2DPI consumes non-negligible memory and CPU

cycles to run a honeypot, fingerprint attacks, and inspect incoming packets. Attackers might target Pot2DPI itself to launch DoS attacks by sending an extraordinary number of packets that hit the honeypot and trigger the execution of the entire pipeline. To deal with such attacks, we can rate-limit the amount of traffic that can reach the honeypot. One could simply drop excessive packets or sample packets on a per-flow basis so that the total number of packets a honeypot sees is kept below a pre-set threshold.

7 RELATED WORK

This section describes related work we have not covered and we address their pertinence to this work.

Honeypot. Existing work, such as Honeycomb [22], IoTPOT [25], and others, takes the honeypot approach for collecting traces and getting attack signatures. We use a novel algorithm for extracting patterns and use clustering analysis to deal with independent attacks. By controlling the configuration of port forwarding and shuffling the port mapping, Pot2DPI ensures that the packet traces obtained by its honeypot can cover a wide range of attacks.

DPI/IPS in a dynamic environment. Security approaches [28, 29] have been proposed to handle a dynamic context, such as an enterprise or home network with BYOD devices (bring your own device). These approaches enable the use of contextual policies in DPI/IPS. Unlike them, our policies (attack signatures) are extracted from a honeypot and installed in IPS/DPI.

Botnet detection. Previous work on botnet detection, such as BotMiner [20] and BotHunter [21], monitors traffic at an aggregate close to victims so that the analysis engine can correlate packets from a large number of bots involved in an attack. Pot2DPI uses a honeypot. All packets it collects are malicious. This simplifies attack detection, and lets it focus on signature extraction. In addition, instead of being close to DDoS victim(s), Pot2DPI works at a home network, close to IoT devices.

Traditional host-centric solutions. Existing consumer techniques, e.g., Box or Norton, let a host send traffic to the cloud for anti-virus inspection. In contrast, Pot2DPI passively captures traffic via honeypot and detects attacks; network monitoring and attack detection are local.

8 CONCLUSION

The wide use of home IoT devices and their vulnerabilities make these devices an attractive target of many attacks. In this work, we take the home network as a vantage point and develop a network-based comprehensive defense to combat home IoT-hunting attacks. Specifically, our system Pot2DPI composes a few practical security techniques–honeypot, DPI, and a realization of moving target defense –to achieve a combined power that traps targeted attacks, forces exposure of malicious traffic, simplifies detection and signature extraction, and accelerates delivery of responses. We anticipate Pot2DPI would be a practical element of a secure smart home.

REFERENCES

[1] 1999. BusyBox. (1999). https://busybox.net/about.html
[2] 1999. F-Secure. (1999). https://www.f-secure.com/en/f-secure
[3] 1999. The Bro Network Security Monitor. (1999). https://www.bro.org
[4] 2001. Bitdefender. (2001). https://www.bitdefender.com
[5] 2007. Developments of the Honeyd Virtual Honeypot. (2007). http://www.honeyd.org/
[6] 2007. Network Intrusion Detection & Prevention System. (2007). https://www.snort.org/
[7] 2011. Register and link Samsung smark TV account online. (2011). http://www.samsung.com/global/article/articleDetailView.do?atcl_id=5
[8] 2013. Docker. (2013). https://www.docker.com/
[9] 2016. Breaking Down Mirai: An IoT DDoS Botnet Analysis. (2016). https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html
[10] 2016. DDoS attack that disrupted internet was largest of its kind in history. (2016). https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
[11] 2016. Google OnHub. (2016). https://on.google.com/hub/
[12] 2016. How IoT & smart home automation will change the way we live. (2016). http://www.businessinsider.com/internet-of-things-smart-home-automation-2016-8
[13] 2016. Multiple vulnerabilities found in Wireless IP Camera. (2016). https://pierrekim.github.io/blog/2017-03-08-camera-goahead-0day.html
[14] 2017. 120,000 IoT cameras vulnerable to new Persirai botnet. (2017). http://www.zdnet.com/article/120000-iot-cameras-vulnerable-to-new-persirai-botnet-say-researchers/
[15] 2017. Malware traffic analysis. (2017). http://www.malware-traffic-analysis.net/
[16] 2017. Pcap analysis. (2017). http://www.pcapanalysis.com/
[17] Nigel Davies, Nina Taft, Mahadev Satyanarayanan, Sarah Clinch, and Brandon Amos. 2016. Privacy Mediators: Helping IoT Cross the Chasm. In *Hot Topics in Mobile Computing (HotMobile)*.
[18] Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, New York, NY, USA.
[19] Nick Feamster. 2010. Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks*. ACM, 37–42.
[20] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. 2008. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-independent Botnet Detection. In *Proceedings of the 17th Conference on Security Symposium*.
[21] Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee. 2007. BotHunter: Detecting Malware Infection Through IDS-driven Dialog Correlation. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*.
[22] Christian Kreibich and Jon Crowcroft. 2004. Honeycomb: Creating Intrusion Detection Signatures Using Honeypots. *SIGCOMM Comput. Commun. Rev.*.
[23] C. B. Lee, C. Roedel, and E. Silenok. 2003. *Detection and Characterization of Port Scan Attacks*. Technical Report. Univeristy of California, Department of Computer Science and Engineering.
[24] James Newsome, Brad Karp, and Dawn Song. 2005. Polygraph: Automatically Generating Signatures for Polymorphic Worms. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*.
[25] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2015. IoTPOT: Analysing the Rise of IoT Compromises. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*.
[26] Roberto Perdisci, Wenke Lee, and Nick Feamster. 2010. Behavioral Clustering of HTTP-based Malware and Signature Generation Using Malicious Network Traces. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*.
[27] Anna Kornfeld Simpson, Franziska Roesner, and Tadayoshi Kohno. 2017. Securing vulnerable home IoT devices with an in-hub security manager. In *International Workshop on Pervasive Smart Living Spaces (PerLS)*.
[28] Tianlong Yu, Seyed K Fayaz, Michael Collins, Vyas Sekar, and Srinivasan Seshan. 2017. PSI: Precise Security Instrumentation for Enterprise Networks. In *Proceedings of the 24th Network and Distributed System Security Symposium (NDSS'17)*.
[29] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. 2015. Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks (HotNets-XIV)*.