Analysis and Extension of Home IoT Network Segmentation Architectures

Tyrone Davis III
td3@mit.edu

Madeline Wang
mewang22@mit.edu

Tim Zavarella
timzava@mit.edu

Maggie Zhang
mzhang21@mit.edu

1 INTRODUCTION

The number of Internet of Things (IoT) devices in the world is projected to triple by 2030 to 25 billion devices. And of that amount about 60% are from the consumer segment [8]. Less than a month before that projection was published, the US government passed a bill called "IoT Cybersecurity Improvement Act of 2020" which requires government agencies to take specific steps to increase the security of IoT devices connected to their networks [12]. With the increase of IoT devices, there has been a correlated increase in the need to secure such devices to protect both normal consumers and large agencies alike worldwide.

IoT attackers have benefited from the increased attack surface of the ever expanding IoT world and have proven successful at exploiting the lack of sufficient security on such devices. Most devices that monitor or sense a specific physical environment and transfer data from the physical world to the virtual world are often classified as IoT devices. A network that supports these devices include uniquely addressable data communicating and collecting devices, a data transmission network, a computing platform, and customized user applications [15]. The growing diversity of networks and architectures has expanded IoT capability, but has also given attackers several entry points into the larger IoT surface.

Three main entry points are the devices themselves, network protocols, and application software. Outdated update mechanisms, insecure components, memory or firmware are vulnerabilities that can exist on a device. The network that connects different devices can be subject to an attack and affect multiple devices at once such as a DDoS attack. The web applications or software that users interact with on the internet can have vulnerabilities that could compromise user credentials. Due to the wide variety of devices, applications, manufacturers, and price points, most devices are not designed with security as a priority. In the event that devices are compromised, it is difficult for users to realize that it has happened as the devices have very little user feedback. These factors make IoT devices attractive targets for attackers.

2 MOTIVATIONS

In the current IoT landscape, the majority of devices do not guarantee trust, non-revocation, secrecy or verification. The security properties of IoT devices, however, differ from those of traditional network devices. They are often stronger but more difficult to implement especially when considering hardware constraints. In particular, IoT devices are low cost, limited in computational power and small storage capacity. These constraints pose a significant problem to security developers for these devices, as direct control over with whom or what our devices are communicating with gets further out of reach as communication between connected devices becomes increasingly expected.

Furthermore, the behavior and structure of IoT devices themselves makes them vulnerable. IoT devices are usually running 24/7, and thus are always connected to the Internet along with their open ports. In addition, due to their under-powered and small storage nature, it is often infeasible to install anti-malware and advanced security software on the devices. Users are often also unable to monitor the behavior of the devices as there is no such user interface. Finally, most devices come from different manufacturers, so the user lacks one central interface to manage all of their devices.

An example of an attack that crippled the Internet thanks to insecure IoT devices was the Mirai botnet attack. In 2016, Mirai took down major sites like AirBnB, GitHub, Twitter, Netflix, etc. through a distributed denial of service (DDoS) attack. Mirai simply scanned blocks of the internet for IoT devices with open Telnet ports and tried to guess user credentials from hardcoded credential combinations. Compromised devices were then used to take down other servers by overwhelming them with large traffic flow [2]. The attack proved more successful than expected as the number of insecure IoT devices was quite high.

Due to high variance in security policies and designs across IoT devices, we think better network-level security can prevent attacks like Mirai while remaining agnostic to device specifics. We look to Software Defined Networking (SDN) as a tool allowing dynamic network policy configurations that will increase network security. We argue that the above technical constraints and attacks underscore the need to define a more robust security policy such as network segmentation. The benefits of network segmentation include isolation from insecure IoT devices on one segment of the network, better visibility if a threat is detected, the prevention of lateral movement between segments by an attacker and improved performance with regard to isolated congestion bursts.

3 DESIGN GOALS

In this paper, we review existing proposed network segmentation security designs that are applicable to home IoT devices and networks. Network segmentation can reduce attack vectors and limit damage to the overall network from compromised devices by isolating infection to at worst a segment. Home users, however, are likely to be non-technical and might be less interested in securing their devices if it is too time consuming or difficult. It was found that "most security advice simply offers a poor cost-benefit tradeoff to users and is rejected" [7]. Furthermore, as DDoS attacks sometimes do not end up specifically affecting the owners of the devices that were infected, but networks as a whole, users are even

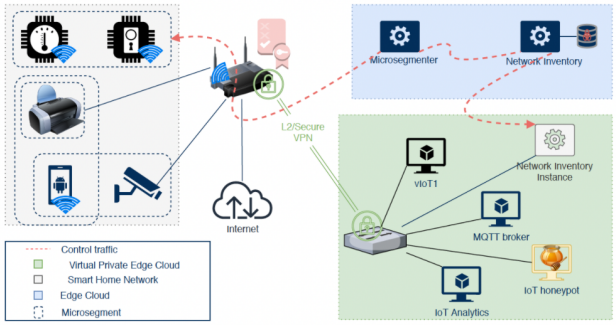

Figure 1: Isolation architecture proposed by "Transparent Microsegmentation in Smart Home IoT Networks" [10]. Red dotted line is the automatic segmentation process

less motivated to take on costs and burden to uphold security in their devices. Thus, since security against attacks depend on many users following security procedures, we will propose a system that prioritizes ease of use as well as secureness.

We want our design to be implementation-wise simple to understand and for the system to operate with good functionality and include as much automation as possible so that the user incurs no burden. However, we do not want to compromise security either, so we want to have a segmentation policy and components that enforce security in the system. As botnet attacks such as Mirai have spread very far and quickly, our system aims to protect against them. [1]

To design a system that upholds our design goals, we intend to review paper policies for security and usability flaws which could reasonably lead to vulnerabilities. After highlighting these flaws, we propose our own network segmentation architecture that addresses these flaws and builds on our analysis of the reviewed systems.

4 BASIC ARCHITECTURES

4.1 Microsegmentation

4.1.1 Overview. The first architecture we will focus on is from "Transparent Microsegmentation in Smart Home IoT Networks" [10]. This architecture uses the rapidly growing technology of software-defined networking (SDN) and virtual network functions (VNF) to control network communication and create virtual segments. Through this segmentation, the goal was to prevent potential lateral infection of the network if a device were to become compromised.

4.1.2 Architecture. This architecture uses 2 network domains, the smart home network where the devices lie, and an edge cloud network that hosts 2 virtual network functions, a microsegmenter and a network inventory. The 2 network domains are connected by a SDN-enabled smart home gateway. The home IoT devices connect to the gateway and the gateway speaks to the virtual functions on the cloud. All communication in this network is enabled through OpenFlow, which creates programmable rules that direct and manage traffic.

A focus of this architecture is automatic segment allocation to prevent users from a manual burden when connecting their devices. When a device first enters the network, the gateway propagates that information to the network inventory that identifies and fingerprints the device and scans it for security vulnerabilities (In this design, their implementation uses the Avast Wifi Inspector to scan for security vulnerabilities. However, another option is Nessus, which scans ports, identifies application issues, discovers unpatched software, and attempts default credentials and then categorizes each vulnerability based on risk level [14]). It then sends that information back to the microsegmenter which allocates the device to a segment accordingly. In this way, users do not need to do any extra work when connecting their devices to their network or gateway. This process of scanning devices is periodically repeated and repeated if the gateway is reset to ensure the network inventory is kept up to date.

Segments on this network are mainly defined by the functionality of the device. The functional groups they used were cameras, controllers/hubs, energy management, appliances, and health-monitors [13]. By segmenting based on functionality, they put the devices that have the most inter-communication together. This allows them to be more restrictive on their communication protocols, their OpenFlow rules.

In this design, microsegments are isolated so there is no communication between different microsegments. By removing unnecessary communication, this reduces the potential attack space for lateral movements. Within microsegments, communication is allowed through OpenFlow rules. Devices that need to communicate with the cloud or the internet do so through the gateway. This also helps secure the network by removing open ports or channels to the internet that attackers can sneak in through. By routing all external communication through the gateway, they ensure at least one consistent wall of security between external networks and the smart home network.

4.1.3 Suitability for Our Architecture Goals. In terms of security, this design works relatively well against lateral attacks. The paper analyzes the system on a Mirai topology taken from [13]. From their case study, they found that the attack surface was reduced by at least 65.85% (depending on which device was initially corrupted) [10]. A flaw of the complete isolation of microsegmentation is that it could potentially block necessary communication between devices in different functional groups and thus different microsegments. In the same case study, they showed that 2.16% of the communication without microsegmentation was blocked with their implementation. Although this is seemingly low and could be blocking potentially malicious communication as well, if communication vital to a user's home utility was blocked, this could still be a problem.

In terms of usability for users, the automatic segment allocation is very convenient for user's day to day life. Furthermore, the Avast Wifi Inspector reports back vulnerabilities on devices to a user. Therefore, by implementing the inspector in the network inventory, users will receive notifications and potential next steps for dealing with vulnerabilities on their devices. This helps with the issue of no transparency or having a proper channel for users to understand the status of their devices. However, this design requires the new technology of SDN and virtual functions on an edge cloud to be set up so there is some amount of work that needs to initially be completed in order for a user to enjoy the automatic functions.

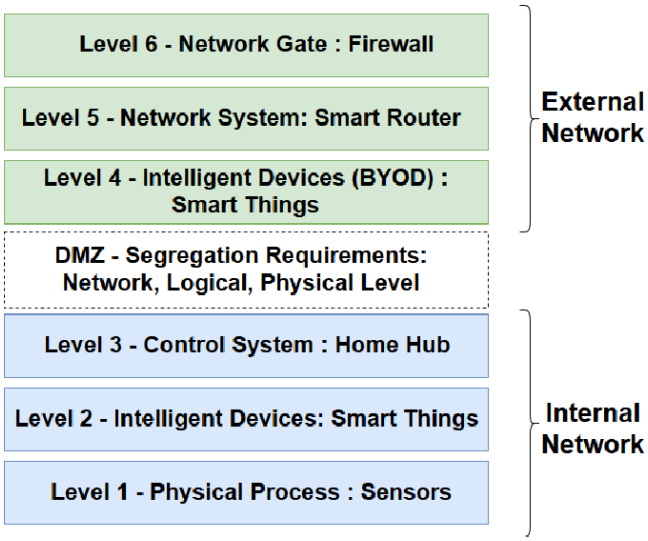

Figure 2: The hierarchy of levels proposed by "A Segregated Architecture for a Trust-based Network of Internet of Things" [6]

Not only would that set up potentially be difficult for an average user, but this technology also isn't currently widespread on most consumer products and thus a user would need to specifically go seek it out.

Finally, a gap the paper does not flesh out is what happens to devices that are found to be vulnerable or malicious. It vaguely states that malicious devices are quarantined and prevented from accessing the wireless network but fails to state how. There is nothing said about how vulnerable but not yet infected devices are dealt with. This would be an important detail we would need to expand on for our own design.

4.2 Segregated Architecture

4.2.1 Overview. The next architecture we reviewed was described in "A Segregated Architecture for a Trust-based Network of Internet of Things." [6] This home network architecture was roughly modeled on how larger scale professionally managed networks are manually configured with several network levels with different security requirements for each. The primary goal of the architecture was to create and maintain a segmented network such that *secure* IoT devices are segregated and can trust that they were only interacting with other secure devices.

4.2.2 Architecture. This is a network segmentation architecture and as such primarily provides North-South isolation through additional physical hardware. The main segments are an external and an internal network as pictured in Figure 2. The external network operates similarly to a standard home network. Computers, smart phones, and other devices are placed on Level 4 in this architecture. IoT devices which meet certain security requirements are allowed access to the internal network by a smart hub which coordinates the devices and limits communication between devices on the internal network. Communication between the internal network and external network (and the internal network and the internet) is allowed through the firewall and DMZ when required for functionality.

The smart hub coordinates devices on the internal network and manages devices' requests to join, stay on, and leave the internal network. In order to join the internal network, the smart hub performs a trust calculation, and if sufficiently high, (as described later) the hub will provide access to the internal network and symmetric encryption keys to use for communication. The hub will also identify other devices which are allowed to communicate with the new devices and will distribute the encryption keys to those devices as well. While devices are on the internal network, the hub will monitor their behaviour for signs of changes or other abnormal behavior. In the event of abnormal behavior, the hub will take one of two actions: quarantine the affected device by limiting any communication between it and other devices on the internal network, and banning the device from the network. In the event that a device leaves the network or is banned and removed from the network, the hub maintains a record of the device's reputation. This prevents devices which are kicked for bad behavior from simply rejoining the network again at a later date.

The smart hub's trust estimation contains several different criteria: if there are any identified vulnerabilities to the device, past behaviour of the device, the importance of the device, and a risk calculation for known attacks. The risk calculation is performed using three additional criteria for each vulnerability/attack: The likelihood L that the risk will happen, the severity S of the damage that could be caused by the risk and the detectability D of the risk. Each criteria is scored 1, 3, or 9 and Risk = LSD. Devices with high risk are not allowed to join the internal network while devices with medium risk are allowed on the network conditionally on a sufficiently high importance

4.2.3 Suitability for Our Architecture Goals. This paper provides an interesting network architecture especially in the trust and risk estimation components. This could provide a robust segregated trusted network for devices which support the architecture. However, there are several components which we believe make this architecture unsuitable for the use case and goals we outlined for this project. The first component is that manufacturer support is required to use the internal network and communicate with the smart hub. As this would not improve the security of existing devices or future devices produced by manufacturers with less concerns over the safety of their products, this does not meet our goal of improving security for existing untrusted IoT devices.

Additionally, we believe that several of the important security features are under-specified which could significantly impact the security guarantees provided by the architecture. One such under-specification was that the paper states that IoT devices in the internal network can only communicate with each other if it is required for functionality. This could provide significant East-West isolation similar to microsegmentation and would be presumably achieved by the smart hub selectively sharing the encryption keys used to communicate. If functionality requirements are determined by user intervention, then users might be inclined to approve everything, even if prompted by a compromised device. Alternatively, if functionality requirements are determined by the manufacturer, then

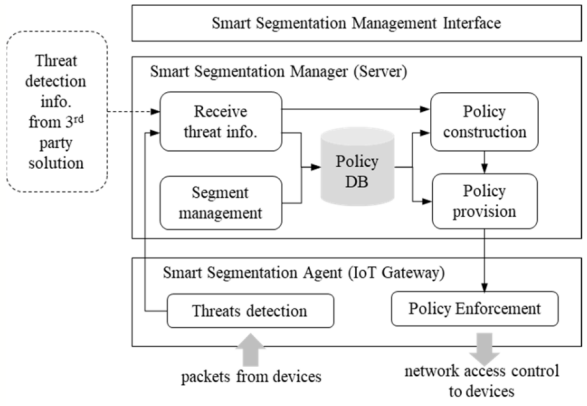

Figure 3: The smart segmentation framework proposed by "Proposal of Smart Segmentation Framework for preventing threats from spreading in IoT." [9]

there is an incentive to be as expansive as possible to allow for future features/interactions between other devices. Similarly, which types of communication are suitable for crossing the DMZ is not specified. If that barrier is sufficiently porous then the benefits of network segmentation are not realized, but if it is too strict, it could limit the functionality of devices on the network.

4.3 Smart Segmentation Framework

4.3.1 Overview. Another system that we analyzed was the smart segmentation framework in "Proposal of Smart Segmentation Framework for preventing threats from spreading in IoT" [9] proposed by researchers in South Korea aiming to create an architecture network resilient against botnet malware attacks like Mirai. This architecture enforces device registration to group the devices into segments, which will be monitored by a manager, who will carry out the security procedure when a spread of a threat is detected within segments.

4.3.2 Architecture. The overall architecture of the smart segmentation framework consists of three parts: a manager that runs on the server, an agent that runs on the IoT gateway where devices are connected, and a management interface that monitors the segmentation statuses as displayed in Figure 3

First, IoT devices are registered and segments are created to divide them. In order for an IoT device to join the system, it must first register, giving its attributes such as device type, vendor, model, OS, etc. Then, the devices will be grouped in segments of three types: a *device segment* consisting of devices with the same attributes, a *network segment* configured by the network unit of a gateway, and a *service segment* which is organized by the type of service the device provides.

During operation, the agent observes the packets passing through the network gateway to look for signs of botnet malware behavior. If it detects this behavior, it passes this information to the manager, which determines the next steps. It looks at the number of threat incidences in this period compared with the average number of threat occurrences in previous analysis periods to see if the spread is spreading, and which segments to take action on. It then creates enforcement policies to mitigate the spread, such as rebooting or shutting down certain devices/segments, and then decides what gateways the policies will be enforced in. These policies are passed back to the corresponding agent, who enforces them as firewall rules. The management interface simply displays the status in the segmentation network, including information like policy enforcement status and threat status for each device/segment to the user through a GUI.

4.3.3 Suitability for Our Architecture Goals. The ideas proposed in this paper appear very promising and robust, as the system was tested against a Mirai malware based DDoS attack, and it performed very well against the attack, with few infected devices. However, not much detail was provided regarding much of the implementation of the architecture, so it is uncertain how the normal user would apply this system idea. In addition, the many different types of segments may make the system costly to maintain. We would use certain ideas and components from this smart segmentation network in our system, such as their idea to detect botnet malware by looking at connection requests outside of reference domain names and their algorithm for detection of infection spread and resulting action. In addition, the manager component idea would create a level of automation in the system to ease the burden on the user as well as provide a central authority to create security for the whole system.

4.4 Zero Trust

4.4.1 Overview. The final system we are looking at in depth is the Zero Trust architecture system proposed in "Securing IoT Devices Using Zero Trust and Blockchain." [5] For the purposes of this paper, we will be focusing on the Zero Trust idea and not really the policies regarding blockchain, although this may be an idea to consider for future work. Zero Trust is based on the idea that nothing on the network should be trusted, and that everything should be verified, which enforces security, especially in networks containing vulnerable components such as IoT devices. The architecture proposed is characterized by a segmented, parallelized, and centralized network in which security protocols are enforced on all entities in the network, access is limited and controlled, and network traffic is unexpected and logged.

4.4.2 Architecture. The Architecture of the Zero Trust framework consists of three key concepts/components. The *segmentation gateway* is the center of the network and provides all security functions, such as firewall, network access control, intrusion detection, etc. It also implements all network security policies and segregates network traffic to secure and parallel network segments. The *microcore and perimeter* (MCAP) creates parallel segmentation and isolates critical network resources. Finally, there is a *centralized, unified, and transparent management* that oversees the microcore and perimeters, as displayed in Figure 4

The paper proposes a segmentation basis for dividing the IoT devices based off of their value and risk to the network by looking at their hardware and software characteristics, as well as their geographic location. Using the device characteristics, volume of generated traffic, and communication protocols, a vulnerability profile for the device is created, which contains a classification of High

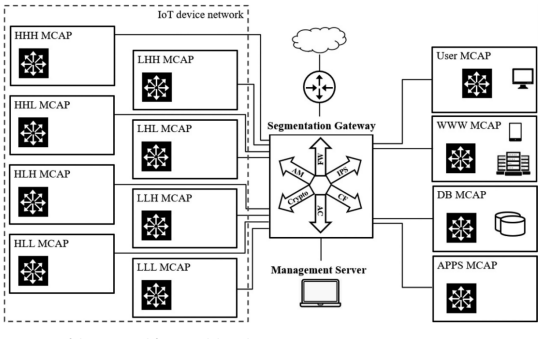

Figure 4: The zero trust framework proposed by "Securing IoT Devices Using Zero Trust and Blockchain." [5]

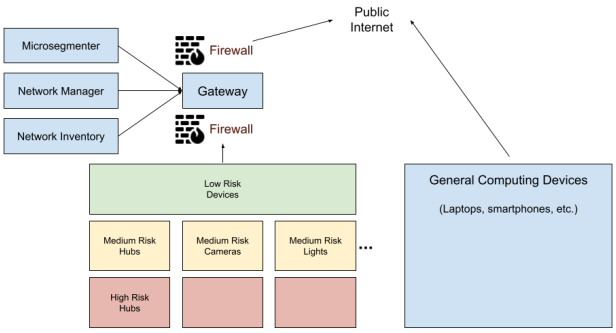

Figure 5: An overview of our proposed system architecture. General computing devices are excluded from the restrictions placed on IoT devices. IoT devices are fingerprinted by the network inventory and placed on segments by the microsegmenter. Segments are assigned based on device risk and functionality as determined by the netowrk inventory. Communication between segments is prohibited except for between medium and high risk of the same functionality through the gateway. The network manager monitors device behavior.

(H) or Low (L) in three categories: network capability (computing power, location, and supported protocol), risk score of the device (threat, vulnerability, business criticality, and estimated loss), and data risk (confidentiality, integrity, and availability). By grouping devices that have the same classification in all three categories, eight different segments are formed (e.g. HHH and HLL).

Consequently, following the idea of Zero Trust, for each segment, there will be separate MCAPs for web traffic, database functions, applications, etc. so that there is no overlap of traffic. The segmentation gateway provides the access control and user authentication services.

4.4.3 Suitability for Our Architecture Goals. From this paper, we see that the idea of Zero Trust is very powerful in enforcing security, as no device/entity has inherent access or trust. However, there is again very little information of what the implementation of the system ideas would look like and what the complexity is. In addition, some functions of components of the system are uncertain, such as what monitoring the network traffic would look like. In our system, we would like to enforce the idea of Zero Trust, with no communication between segments that is not verified, and encrypted messages. The idea of segmenting based off of a 3D vulnerability profile also seems to create robust segments, since IoT devices serve a variety of functions, and as such, we will be considering characteristics such as system importance of devices in our system segmentation.

5 OUR DESIGN

5.1 System Overview

Our system will be primarily based on a micro-segmentation approach. As stated previously, this system is geared towards home networks with IoT devices and largely non-technical users. Furthermore, this approach aligns with the Zero Trust policy that we wish to include as devices come from all different manufacturers and with varying levels of security, and we should not expect devices to be safe [5]. Our architecture is managed by a single central coordinator, and the microsegmentation can be achieved by software-defined networking technologies such as the OpenFlow protocol [10]. Centralizing all of the network management in one device does create a single target for attackers and poses risks. However we believe that this approach is the most feasible for actual adoption. A decentralized approach increases complexity which can harm user understanding and adding additional devices increases costs which would decrease the likelihood of users adopting the architecture. For these reasons we chose a centralized approach.

On the network we will consider two main types of devices: standard computing devices (such as laptops and smartphones) and IoT devices. Standard computing devices are considered trusted in our architecture and are not subject to segmentation or other restrictions beyond those typical of a home network. This is because many of the security flaws of IoT devices such as poor feedback on device state and infrequent updates are less common on general purpose computing devices. Additionally, many of the assumptions which are made in order to devise our segmentation and firewall approach do not hold for general computing devices. As such this approach would be too inconvenient to the user to expect widespread use of the system if it was enforced for general use devices.

Devices which are identified as IoT devices will be moved onto the segregated IoT component of the network. They will then be placed within a microsegment based on their risk and functionality. Complete isolation between devices would likely provide the most security against lateral movement. However, this is impractical though as IoT devices often communicate with each other in order to provide their functionality and disallowing this would unacceptably damage functionality. The selected approach is an attempt to minimize risk while still allowing users to use their devices to the greatest degree possible.

Once an IoT device is placed on the network, it will be monitored by a network manager and certain communication must pass through a whitelist based firewall. The network manager improves the user's ability to detect changes in a device's behavior which

might indicate that it has been compromised. The network manager can alert the user to changes, reclassify devices as higher risk, and if necessary ban suspicious devices [9]. Monitoring behavior is especially important between devices in the same microsegment as there are less preventative measures against lateral movement within the segment. The firewall operates some microsegments, and is between the home network and the public internet. The firewall attempts to stringently enforce a requirement that devices only communicate with other devices as required for original functionality. This prevents compromised devices from performing actions that deviate from the typical actions of the device. The goal of these features is not to provide strict security guarantees, but instead to minimize attack surfaces in a manner that is as unobtrusive to the user as possible. Stricter security guarantees are possible, but if they are too difficult to use or to understand the importance of then they will not be used and ultimately will be less effective than a more limited system.

5.2 Segmentation Guidelines

Our segmentation guidelines will be based on a mix of the microsegmentation and segregated architecture designs. From the microsegmentation architecture, we will use the idea of complete isolation of microsegments, their communication protocols, and their functional groups: cameras, controllers/hubs, energy management, appliances, and health-monitors [10]. From the segregated architecture design, we shall use their trust estimation calculation to end up with a low, medium, or high risk for each device [6]. We chose to use isolation of microsegments due to our focus on security, especially against lateral attacks. By eliminating that communication, we remove potential attack paths. Given that we chose isolation, we then needed to base our microsegments on functionality in order to allow necessary communication for utility purposes. However, we also wanted to take into account the risks of the devices. By keeping higher risk devices away from lower risk devices, we could further limit the potential attack space at a higher probability and ensure better security for safer devices. We chose to use the calculation from the segregated architecture design because we appreciated its focus on detectability, as usability and transparency are important to us. We also liked the factor of the reputation score since it would make the process more efficient and secure.

Our overall segmentation will be as follows. On device entry, the network inventory fingerprints the device, scans for vulnerabilities, calculates the trust estimation score, and returns the results along with a classification of functionality. The microsegmenter will take that score and functionality and allocate the device accordingly. There will be one segment for all devices with low risk. This is analogous to the internal network from the segregated architecture. We've combined all the low risk devices as those are considered devices we can "trust". However, for the rest of the devices, we will split on both functionality and risk so that we have a group for medium risk cameras, high risk cameras, medium risk hubs, high risk hubs, etc. In this way, we will end up with at most 13 microsegments.

Similar to the microsegmentation communication, we allow devices in the same microsegment to directly communicate with each other and allow communication with external networks through the gateway. Typically, there should be no communication between different microsegments. However, by creating separate microsegments for medium and high risk devices of the same functionalities, we could potentially cause problems in terms of the working functionality of those devices if they need to communicate. To ensure the home IoT devices still work, we will allow communication between medium and high risk segments of the same functionality. However, this communication will go through the gateway just like communication with external networks. It will also be monitored by the network manager to detect any malicious communication. All other communication between different microsegments will be prohibited.

5.3 Network Manager

From the various papers, we concluded that having a network manager in our system will be beneficial, as it provides a central authority in the system that can make decisions as well as provide a level of automation so that the user is not burdened with constantly having to monitoring the network traffic or deciding what actions should be taken when infection is detected. Managers are often used in different systems to provide high level policies and command without the complexity of distributed management. For our purposes, the manager will be located on the segmentation gateway and be able to view all traffic and communication between IoT devices and segments on the network. To detect malware behavior, the manager monitors packets that pass through the gateway. One way of detecting such threats is since botnet malware receives attack commands from a remote Command and Control server, when a device is infected, it must make DNS queries to connect to that server frequently, as its IP address changes to avoid tracking. Therefore, when the manager sees a connection request that is outside of the specified target of each IoT device, it will view this as suspicious and mark it as a threat [9].

Meanwhile, the manager also has an algorithm for determining rate of infection and what areas of the network have been compromised. One method of doing this is as the manager constantly monitors the network for threats, it will store the number of threat occurrences in a period of time (determined based on the system). When detecting a threat, it will compare the number of threat occurrences in the current time period with the average number of threat occurrences in previous time periods. If the number is greater, the manager knows that the infection is spreading. The manager can do this for each segment as well as the whole network to locate where the infection has occurred. Then it will decide whether to reboot the devices in that segment or to shut them down based on the threat level and carry this out automatically. [9]

5.4 Restrictive Firewall

The firewall plays an important part in limiting the attack vectors on the network and minimizing damage in the event of an infection on the network in our architecture. In-bound and out-bound packets to IoT devices have to pass through a restrictive whitelist based firewall. Outbound firewalls are less commonly used as internal devices are typically treated as trusted. We are treating IoT devices as untrusted and the outbound firewall plays an important part in limiting harm done by a compromised IoT device. Additionally,

as described in the segmentation guidelines, since the segregation between medium and high risk segments of the same functionality is relaxed, those communications would also have to pass through the firewall. A whitelist only firewall was chosen as a reactive blacklisting approach would only protect against known bad actors or attack vectors.

When a new device joins the network, it is fingerprinted and the firewall rules are generated using a trust on first use model. Devices are allowed unrestricted access during first startup and a short period of normal operation. Since most IoT devices only contact a small number of distinct domains in order to function properly, [3] this list of domains contacted on joining can be used as the whitelist for that device going forward. Therefore, if devices in other microsegments or on the public internet attempt to communicate with this device, they would be disallowed as they would not be on the whitelist. This can prevent default credentials and many unpatched vulnerability attacks as only a limited number of devices are permitted to communicate with the IoT device. This does not prevent lateral attacks within the same microsegment, however, as traffic within a microsegment does not pass through the firewall. The outbound firewall does limit the value of such an attack though. Attackers can only communicate with the whitelisted domains, so performing DDoS attacks on arbitrary targets or sending the user's private information to arbitrary domains is prevented.

This approach does have one significant disadvantage – new features to devices would not be available if it required to connect to new devices/domains. While this is inconvenient for users and manufacturers, we believe the security advantages outweigh this disadvantage. Our architecture's solution to this problem is to require users to manually approve new whitelist rules that are automatically generated based on device connection attempts. This makes it easy for users to modify their network for new functionality. The system would require users to check the system rather than notifying them of these new rules as this would decrease the chance the user would blindly approve a request they didn't understand or expect

Devices such as smart assistants would also be negatively impacted by this approach. These devices can connect to many different domains and share similarities with general computing devices. They are also generally developed by large companies with a stronger emphasis on security. As such our system would not classify these devices as IoT devices solving this problem.

5.5 Analysis

5.5.1 Security. Many current segmentation designs heavily focus on preventing lateral attacks on the network after a device has been compromised as that is an extremely common attack deployed with bots. We recognize that as a major threat as well and have utilized microsegmentation, a firewall, and a network monitor to protect against it. On entry, we segment devices based on risk and vulnerability of devices. Our isolation scheme reduces the possible attack space as it removes unnecessary communication routes between devices in different segments. For the limited communication allowed between segments (of the same functionality), it flows through the gateway which has a whitelist firewall on it and the network manager monitoring traffic. The firewall will ensure that suspicious communication between devices that don't typically communicate is blocked. Finally, if the attack was able to get through both those levels, the network manager adds an additional level of security by monitoring the traffic and computing the risks of infection spreading. If it finds suspicious behavior, it can reset or temporarily shut down a segment to protect the other segments. By ensuring a dynamic system that is regularly scanned by the network inventory and monitored by the network manager, we are constantly protecting against infected devices entering the network and compromised devices spreading the infection.

Aside from lateral attacks, our system goes beyond typical segmentation designs as it also helps prevent a number of attacks on the individual devices. Since all communication with external networks flows through the gateway with the firewall on it, we help reduce the number of open paths attackers can use from outside the network to gain access to a device. The network inventory helps report known vulnerabilities to our users so they can take action. Most importantly, our firewall helps protect our devices against communication from devices not on the whitelist. This will prevent unknown or random malicious devices from being able to communicate with our devices. Most typical bot attacks like default credentials or unpatched software attacks will be protected against with the firewall.

5.5.2 Usability. Our design focuses on accessibility and ease for users by reducing the need for user action and being transparent with any actions they do need to take. One way we do so is through our automatic segmentation lifecycle that makes it seamless to add devices to the network. If there are specific vulnerabilities on those devices, we alert the user so they can choose to address the vulnerabilities. This helps users become aware of the security of their devices. However, we segment accordingly to ensure the network is safe regardless of user input or action. If user action is taken, it will be updated on the next periodic cycle of fingerprinting, scanning, and segmenting the devices.

Other automatic features that help improve security with little burden on the user are the firewall and network manager that automatically monitor the network, filter out risks, and deal with potential infection sources. If there are certain communications that are blocked by the firewall, we notify the user so they can check their systems and manually allow certain communications through the firewall. However, once again, regardless of user action or input, they can rest assured in knowing the system is still protecting against those threats. In this way, we put minimal burden on the user throughout the process but are transparent with threats and risks so they have the choice to take action.

5.5.3 Considerations. One consideration to take in mind is that we use a centralized system that is heavily based on the gateway. All communication externally and limited communication between microsegments flows through the gateway. Furthermore, the network manager acts on the gateway and the gateway communicates with the network inventory and microsegmenter. In other words, if the gateway becomes compromised, much of the network becomes at risk. This is a risk that we take for the functionality of the system we designed. We also take into account the fact that IoT gateway security is known to be of great importance so the security of gateways is typically quite high.

Another consideration is that our implementation is based in SDN and the technology that is still developing. In order for the microsegmentation implementation discussed in [the microsegmentation paper] to work, the user needs to acquire a SDN-enabled gateway and SDN applications. While these exist, they are not as ubiquitous and so there may be a slightly higher barrier to entry. However, as SDN continues to grow and become more common in systems, we believe this will become less of a problem.

Finally, Osman et al. mentions an issue with the scalability of the OpenFlow protocols that are used to direct or restrict traffic between devices [10]. If there are many segments and many devices on each segment, it's possible this could use a nontrivial amount of space. Given scalability was not one of our priorities, we did not address this particular issue. However, we made that decision knowing that home IoT networks typically do not have a significant number of IoT devices.

6 FUTURE WORK

Our proposed design is only one step towards increasing network-level security for IoT devices. Future work will include implementation and conducting an empirical analysis of our proposed design within a test network that is vulnerable to an attack such as Mirai [10]. There will be about 30 devices (IoT and non-IoT) placed at random in the network topology. Test groups will include: with micro-segmentation on functional groups, no segmentation, and all devices isolated. We will evaluate the system based on vulnerability scores [11]. This will quantify our security guarantees with regard to a tested attack.

In the future we would also like to investigate other approaches to developing a restrictive firewall which is easily configured and understood by end users. Historically firewalls have fallen out of favor for their difficulty and inconvenience despite their applicability to this problem. [4] User studies should be performed to determine if real world usage patterns follow the assumptions we made in structuring our firewall (mainly that devices contact only a small static number of domains). If this does not hold, or if this approach is not sufficient, it could be supplemented with other approaches. For instance, approaches that receive firewall rules from the manufacturer or crowd-source firewall rules present interesting design options that were not considered in this paper.

7 CONCLUSION

As IoT devices are both widely propagated and low security, this makes them vulnerable to attack and thus makes entire networks susceptible. One proposed method of reducing network risk is network segmentation, which partitions IoT devices into groups to prevent infection of the whole network. We have read and analyzed a number of papers proposing strategies to carry out network segmentation for IoT devices with various designs and components. There were different segmentation group divisions, systems with managers, systems with trust and zero trust, etc. However, a common problem in all these proposals was the lack of implementation details and importance placed on user usability. Many IoT security measures are simply not used because they are too complicated for the user to carry out. The goal of our system was to take ideas from the papers to create an understandable system that had both strong security and usability based on zero trust. We group devices on both their risk and functionality to create isolated segments that will improve network secureness as well as keep performance, and have a manager overseeing the network to increase automation. We provide a general idea of implementation in order to help the user understand the system and how to use it.

ACKNOWLEDGMENTS

This work was supported by the MIT 6.857 staff. Special thanks to Ron Rivest, Yael Kalai, Michael Specter and all the Spring 2021 TAs.

REFERENCES

[1] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the mirai botnet. In *26th {USENIX} security symposium ({USENIX} Security 17)*. 1093–1110.

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[3] David Barrera, Ian Molloy, and Heqing Huang. 2017. IDIoT: Securing the Internet of Things like it's 1994. arXiv:1712.03623 [cs.CR]

[4] David Barrera, Ian Molloy, and Heqing Huang. 2018. Standardizing IoT Network Security Policy Enforcement.

[5] Suparna Dhar and Indranil Bose. 2020. Securing IoT Devices Using Zero Trust and Blockchain. *Journal of Organizational Computing and Electronic Commerce* (2020), 1–17.

[6] Davide Ferraris, M. Gago, Joshua Daniel, and J. López. 2019. A Segregated Architecture for a Trust-based Network of Internet of Things. *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)* (2019), 1–6.

[7] Cormac Herley. 2009. *So Long, And No Thanks for the Externalities: The Rational Rejection of Security Advice by Users* (nspw ed.). Technical Report MSR-TR-2009-46. https://www.microsoft.com/en-us/research/publication/so-long-and-no-thanks-for-the-externalities-the-rational-rejection-of-security-advice-by-users/ NSPW.

[8] Arne Holst. 2021. *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030.* Statista. Retrieved May 18, 2021 from https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[9] Jaedeok Lim, Seongyoung Sohn, and Jeongnyeo Kim. 2020. Proposal of Smart Segmentation Framework for preventing threats from spreading in IoT. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 1745–1747.

[10] Amr Osman, Armin Wasicek, Stefan Köpsell, and Thorsten Strufe. 2020. Transparent Microsegmentation in Smart Home IoT Networks. In *3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 20)*. USENIX Association. https://www.usenix.org/conference/hotedge20/presentation/osman

[11] Josh Payne, Karan Budhraja, and Ashish Kundu. 2019. How Secure Is Your IoT Network?. In *2019 IEEE International Congress on Internet of Things (ICIOT)*. 181–188. https://doi.org/10.1109/ICIOT.2019.00038

[12] Robin L. Rep. Kelly. 2020. *H.R.1668 - IoT Cybersecurity Improvement Act of 2020.* U.S. Congress. Retrieved May 18, 2021 from https://www.congress.gov/bill/116th-congress/house-bill/1668

[13] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2019. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (Aug 2019), 1745–1759. https://doi.org/10.1109/TMC.2018.2866249

[14] Ryan Williams, Emma McMahon, Sagar Samtani, Mark Patton, and Hsinchun Chen. 2017. Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 179–181. https://doi.org/10.1109/ISI.2017.8004904

[15] Congyingzi Zhang and Robert Green. 2015. Communication Security in Internet of Thing: Preventive Measure and Avoid DDoS Attack over IoT Network. In *Proceedings of the 18th Symposium on Communications & Networking* (Alexandria, Virginia) *(CNS '15)*. Society for Computer Simulation International, San Diego, CA, USA, 8–15. https://doi.org/10.5555/2872550.2872552