Analyzing the Attack Surface of Consumer IoT Devices in the Comcast Network

Executive Summary

Comcast has many subscribers actively using smart devices known as Internet of Things (IoTs). Many such popular IoT devices are shipped with serious security and privacy flaws. These flaws put end users that purchase the devices at risk, create broader security issues for Internet Service Providers (ISPs), and introduce new support and mitigation costs (which are typically passed on to end users).

In this project, we analyze network traffic from Comcast subscribers to better understand the state of security and privacy for consumer IoT devices. Using DNS lookup data and deep packet inspection (DPI), we study the connectivity, security, and privacy of a small subset of smart homes.

We use DNS data to pinpoint homes suspected of having IoTs and then capture and analyze traffic from these homes. Of 200K subscribers, approx 10% had a known IoT device in their homes. We further separate any IoT traffic that we could detect and study its traffic profile. We developed a script to search for keywords through any unencrypted data such as HTTP payloads and found that IoT data traffic is mostly encrypted. We also confirm the absence of malicious traffic by checking traffic destinations against Comcast's threat intelligence framework, and develop a trivial method to pinpoint infected devices in homes if any are detected.

Our work is a first step in analyzing IoT related activity passively from subscribers from a large ISP such as Comcast. We face and solve various challenges such as capturing packets for low-activity IoT and identifying IoT related content at the CMTS level. We concentrate on manual analysis to learn how IoTs behave, and present various solutions and suggestions to secure traffic from smart homes in the future. Our recommendations include setting up a live-stream analysis for confirmed smart homes to verify security and privacy at run time, as well as updating the list of domains that can successfully identify IoTs at the ISP using only DNS queries.

Introduction

Many new devices that are connected to the Internet are devices that are nonPC that nonetheless have Internet connectivity and functionality. This class of devices has generally been described as the Internet of Things (IoT) and refers to a class of Internet connected and may have additional functions as a result of this Internet connectivity, but which are not conventional general purpose personal computer types of devices. Many such IoT devices are shipped with serious security and privacy flaws. These flaws put end users that purchase the devices at risk, create broader security issues for Internet Service Providers (ISPs), and introduce new support and mitigation costs (which are typically passed on to end users).

Many such IoT devices have limited capabilities, may be poorly programmed, or may not follow the right security policies. The growing popularity of these new and untested devices in the Comcast subscriber network has introduced security and privacy risks allowing both malicious hackers and eavesdroppers a large landscape to explore and launch attacks from. A recent incident showed that IoTs such as CCTVs when hacked can easily produce DDoS attack traffic at a scale as large as 400 GBps. Furthermore, an insecure device in a home network may become a gateway to gaining access to all devices in a home and easily accessing private user data or changing the device configuration. This problem becomes very dangerous for IoT devices such as medical equipment, that may result in loss of life if misconfigured by an attacker, or equipment such as security locks and temperature control, that result in loss of money and discomfort.

The purpose of the research is to understand the state of security and privacy of consumer IoT devices in the Comcast network and potentially build an algorithm or ruleset to improve detection capabilities. The research entails analysis of Domain Name System (DNS) and Deep Packet Inspection (DPI) data to investigate the four use cases listed below. DNS provides a list of candidate of homes (CPE IP addresses) that have IoT devices whilst DPI provides lower level details which can be analyzed to understand IoT device security footprint. It should be noted that customer privacy is paramount and the research will aim to preserve it.

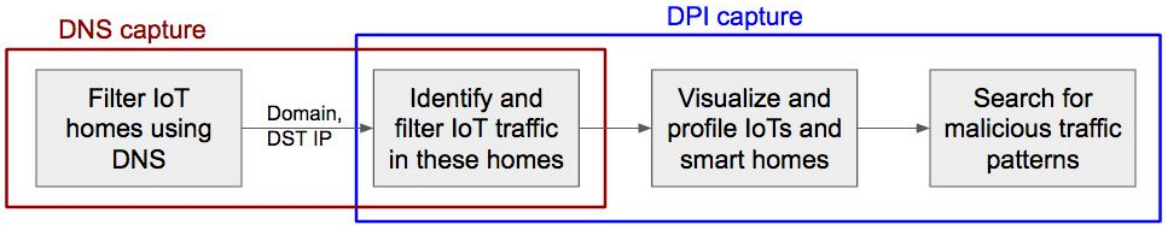
In this work, we analyze network traffic from Comcast subscribers to better understand the state of security and privacy for consumer IoT devices. Using DNS lookup data and deep packet inspection (DPI), we will study the following questions:

- Visualize connectivity of a typical smart home
- Analyze the security of IoT devices in subscriber networks and check for malicious end-points
- Study potential privacy leaks from home network data
- Understand what devices are being used by subscribers and what technologies are used by these devices for communication

Our analysis shows that out of 200K subscribers, approx 10% had an IoT device in their homes. We further separate any IoT traffic that we could detect and study its traffic profile. We developed a script to search for keywords through any unencrypted data such as HTTP payloads and found that IoT data traffic is mostly encrypted. We also confirm the absence of malicious traffic by confirming traffic destinations with Comcast's threat intelligence framework, and develop a trivial method to pinpoint infected devices in homes if any are detected.

Method and Data

We divide this work in two parts: First, we will analyze DNS queries to list IoT related requests and the homes that issue them. Secondly, we will filter the DPI data to these homes and analyze the security footprint of these smart homes and IoT devices.



Finding IoTs using DNS data

We capture a list of confirmed IoT-related DNS queries from certain Comcast CMTS sites for analysis [Table 1]. These domains were collected using various active experiments [1] and previous work on IoT related domains [2]. Our previous work shows that the IoTs listed connect to very specific domains that can be easily used to identify the device in a smart home.

Initially, we captured all DNS traffic from sites, however due to the low frequency of IoT DNS queries this causes an explosion in the DNS data captured. Thus, we limit the capture to certain domains on site. This list may be expanded as needed when newer devices with confirmed IoT-related domain queries are detected.

The aim is to list all CPE IPs that we can detect with queries to IoT-domains. Next, these CPE IPs will be monitored for all traffic to analyze IoT related activity, as well as security and privacy.

Table 1: Devices and their filtered IoT-related destination domains

	Device	Destination Domain
1	Nest Thermostat	devices.nest.com
2	Nest Cam	nexus.dropcam.com

3	Samsung Smartthings Hub	api.smartthings.com
4	PixStar Photoframe	api.pix-star.com
5	Amazon Echo	device-metrics-us.amazon.com
6	Sharx Security Camera	sharxsecurity.com
7	August Doorlock	doorbells.august.com
8	Phillips Hue Lightbulbs	bridge.meethue.com
9	Apple Watch	gs-loc.ls-apple.com
10	PlayStation	update.playstation.net
11	Xbox	*.xboxlive.com
12	Nintendo Wii	*.nintendo.net

Limitations and Challenges

The main limitation to the white-listed filtering approach is that we may be missing many IoTs as the list is not exclusive. However, there is also a large chance that a home with one of the listed devices actually has other IoTs that we may be able to capture for our DPI framework. Our results in the next section confirm this as we were able to capture traffic from devices such as SONOS, Samsung SmartTV, and Ring doorbell.

Certain IoTs only query the main domain initially at startup time and follow up only with NTP queries (eg: smartthings hub). Other media related IoTs such as amazon echo might query google.com or spotify.com frequently, which can't really be used to identify IoTs. Thus, it is challenging to select domains that we are absolutely confident in. As we mainly use this stage to filter homes rather than analyzing IoT traffic, we may include domains we suspect as IoTs to analyze the home during the DPI stage.

Data Collection and Characterization

We collected two main datasets: (a) PIMA-detailed that contained all DNS queries during 30 minutes. (b) One day IoT-related DNS filters from the following sites: PIMA, TUSC, MSP, JAX. While PIMA-detailed was a packet capture of all DNS queries, PIMA, TUSC, MSP, and JAX were csv files of only IoT related queries only containing the time, CPE IP, and DNS query, processed at the capture site.

PIMA-detailed collected DNS queries from 200k homes, and we confirmed that 17,577 had IoTs with domains from table 1. Combining results with PIMA, we found that the most popular devices detected were Amazon Echo and Xbox. More than 10k homes had only 1 IoT related query throughout the day, whereas 184 homes had 3 or more queries. We detected a maximum of 6 devices in a home.

Table 2: Summary of DNS analysis

	JAX	TUSC	MSP
Homes with IoTs	29041	24082	12190
Homes with 3+ IoTs	200	606	216
Most popular overall	Amazon Echo and Playstation	Amazon Echo and XBox	Amazon Echo and Apple Watch
Most popular (3+)	Amazon Echo (179); Apple Watch (159)	Apple Watch (518); Xbox (514)	Amazon Echo (159); Apple Watch (115)

The least popular devices in the dataset are the Nest Cam, August Doorlock, and PixStar Photoframe. No homes were detected using the Sharx Security IP camera.

Analyzing IoT security using DPI

We use the CPE IPs filtered from our DNS analysis to set up captures for smart homes in the DPI framework. These homes are confirmed to be using one of the devices listed in table 1, and may have more IoTs that we haven't listed. Our aim is fourfold:

1. Analysis: Extract IoT traffic and study traffic pattern and encryption.
2. Privacy: Search for data leakage in IoT traffic in unencrypted packet payloads or URIs.
3. Security: Analyze if smart home is communicating with malicious destinations.
4. Connectivity: Visualize destinations and URIs IoTs connect to.

For the analysis stage, we capture traffic from filtered smarthomes throughout the day and separate IoT related traffic using DNS queries and responses. For example, all nest thermostat data and logging traffic goes to the following domains: log-rts-*.nest.com; devices-log-*.nest.com. We retrieve the IP address of these destinations via DNS queries that are captured in our samples, and filter all traffic to these IP addresses for the respective device. By using DNS, we were also able to successfully identify more IoTs and their respective domain queries: Ring, Sonos, Samsung TV.

To analyze privacy, we create a short script that accepts searchable keywords in a text file. This script can be run over the captured or filtered traffic to mark instances of data leakage. For demonstration purposes, we only included certain keywords such as 'account', 'address', 'user'.

This list can be expanded as required. Currently, the script can search HTTP payloads for unencrypted packets, as well as URIs from HTTP requests.

For security analysis, we use Comcast's threat intelligence system to query destination IP addresses that smart homes are communicating with. The elastic search engine allows us to perform multiple queries at a time listing all instances of a malicious IP address and suspected faults. Currently, we perform a security analysis of the packet capture. However, in the future this should be turned into a live system for smart homes that can raise alerts on the fly for new destinations these homes communicate to. Using this system, in the next section we present a trivial method to pinpoint suspicious devices that might be infected with malware in Comcast subscribers.

For connectivity we use DNS query and responses, as well as reverse IP lookups to summarize websites and countries the smart home is communicating with. For the current report, we merge the results of connectivity and analysis sections as we concentrate on only IoT-related destinations.

Limitations and Challenges

The main challenge of capturing IoT related traffic is that IoT traffic is very sparse, but IoTs themselves are very diverse. Initially, we tried capturing traffic from as many homes as possible, however the amount of traffic generated makes this practically impossible. Thus, to get captures over a longer duration, we concentrated on the top homes with most number of IoTs from our DNS analysis. However, we also observed that the intersection between DNS and DPI datasets is low. For example, from the top 10 smart homes detected by DNS analysis for PIMA, only 2 homes were capturable in the DPI framework. In fact, the TUSC DNS site had no IPs available for capture in DPI. Thus, our analysis concentrates on a few homes available in both DPI that were detected with multiple IoTs using DNS.

To reduce the amount of traffic captured for a long duration (~20 hours), we use service identification and neglect known traffic such as video and SSL. We set up captures to remove the following services: Netflix, YouTube, Amazon Video, Hulu, Google Play, iTunes Store, SSL, Snapchat, Instagram, GRE, Amazon (website), BitTorrent, Xbox Downloads, PS4 Downloads, Facebook, PCGames, HTTP Live Streaming, MPEG, Comcast.

Separating IoT traffic from large captures is challenging. This is further complicated by the fact that IoT related destinations are not permanent IP addresses, but service servers spawned for a device. Our current approach is to use DNS to identify the IP destination for an IoT, and then filter the flow to that destination. However, we may miss IoT traffic in case there was no corresponding query captured throughout the trace, or a query was only performed at the start up phase and never again. Queries to certain common media domains from IoTs are missed as we can't successfully identify if traffic was generated from an IoT device or not (for example, Amazon Echo and a laptop might both send traffic to spotify.com). Thus, we get only a

conservative estimate of IoT traffic. Furthermore, this method also requires manual analysis for new IoTs.

The chance of finding malicious activity in such traces is almost negligible. Thus we suggest a live stream malware identification for smart homes instead of offline analysis.

Data Collection and Characterization

For capturing DPI data we concentrate on traffic from the PIMA site. Our captures consist of two datasets: (a) PIMA-TRY1: 20 hours of packet capture from a CPE IP known to have at least 5 IoTs; (b) PIMA-TOP: 20 hours of packet capture from the top CPE IPs with more than 3 IoTs.

Through DNS analysis, we know that the smart home used for capturing PIMA-TRY1 is known to have the following devices: Philips Hue smart bulbs, Amazon Echo, Nintendo Wii, Xbox, Nest Thermostat. The 18 homes in PIMA-TOP includes this CPE IP. The most popular device is Amazon Echo, in 14 of the 18 homes, followed by Playstation (9), Nest Thermostat (7), and Xbox (6). Only 2 homes have a Philips Hue and Nintendo Wii each, and only 1 home has a Smartthings hub and an Apple Watch.

Findings

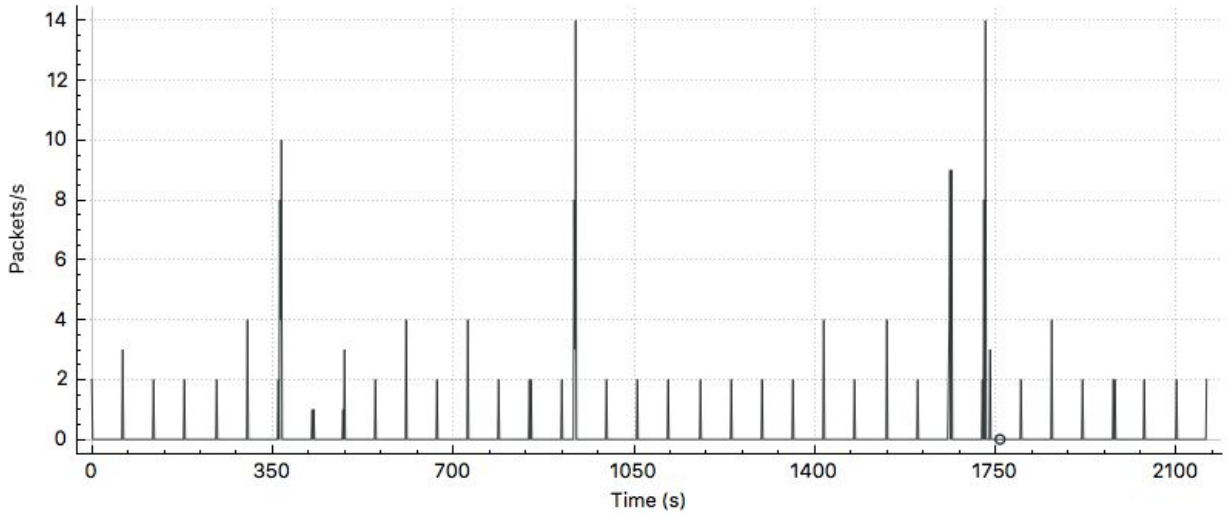
Connectivity and Analysis

Initially, we use wireshark to manually analyze and separate IoT activity. For PIMA-TRY1, we find three more IoT devices that were not included in table 1: Ring, Sonos, Samsung TV. Ring was detected via queries to es.ring.com; Sonos was detected by queries to www.sonos.com; and Samsung Smart TV was detected by frequent queries to samsungacr.com. We also find queries to ooma, a voip phone service, however we are not sure if this is an IoT device.

Using DNS we separate IoT traffic to their respective destinations for both PIMA-TRY1 and PIMA-TOP dataset. We observe that the IoT data separated consists of either TCP SYN/ACK packets to reset connections, or TLSv1.2 packets when data is involved. Note that we are limiting ourselves to only packets going to IoT related domains we've listed, however there might be unencrypted IoT traffic that we were unable to separate due to the limitation of our DNS based approach.

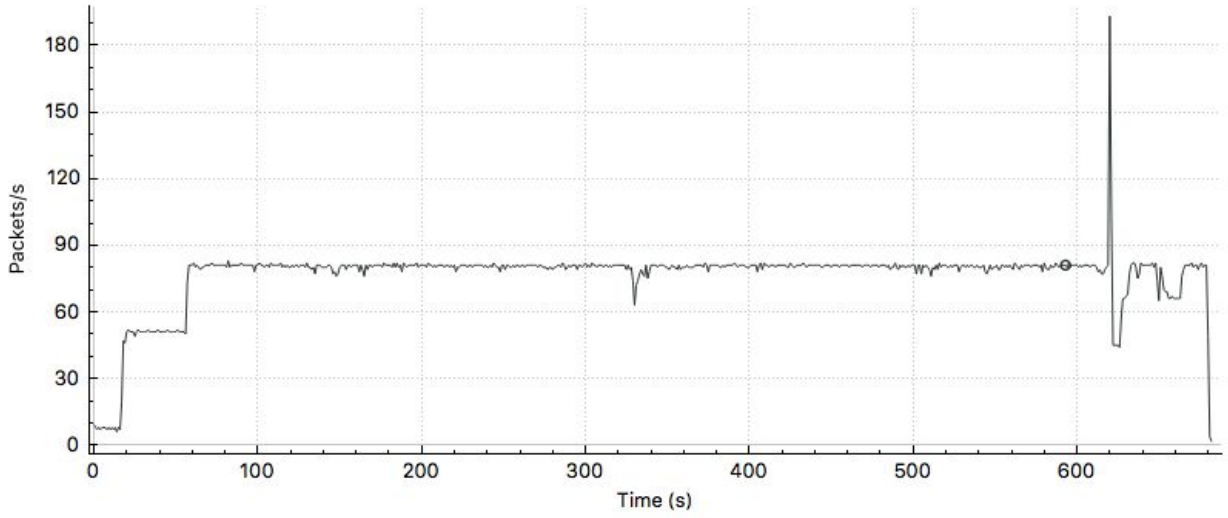
The following plot shows activity of a Nest thermostat in PIMA-TRY1 for 35 minutes in packet per second. The large spikes are periods when there was TLS data being communicated, the rest of the time is keep-alive messages every 60 seconds, if there was no TLS activity. This data was extracted by filtering traffic to the IP address returned for the query to *-front01-iad01.transport.home.nest.com.

Fig1: Nest traffic PIMA-TRY1



Using a similar approach to separate traffic for Ring shows intermittent activity with only 5 exchanges in a half-hour period. The IP address is an Amazon EC2 server. For sonos, we see TCP and SSL handshakes every 20 seconds throughout the 35 minutes but no data exchange. In contrast to the above IoTs, media-related IoTs such as the Samsung SmartTV generates a heavy GSVP protocol stream when in use as shown below. This stream was filtered by protocol rather than DNS as the DNS queries were only to the ACR server whereas we suspect that the actual data stream was using the high definition streaming protocol GSVP.

Fig 2: Samsung Smart TV stream from PIMA-TOP in a capture from 7:30pm to 8:30pm.



We have generated similar plots and analysis for the devices we could filter using DNS from PIMA-TRY1 and PIMA-TOP captures. From a detailed analysis of PIMA-TOP we have found the following new IoT devices:

Table 3: New devices and their corresponding IoT-related destination domains

	Device	Destination Domain
1	Wemo Light Switch	api.xbcs.net
2	Belkin Socket	www.belkin.com
3	Honeywell Alarm	lyric.alarmnet.com
4	Ring Doorbell	*.ring.com
5	Sonos Speakers	*.ws.sonos.com
6	Solar City Energy Monitor	solarcity.devicecloud.com
7	Samsung Smart TV (ACR)	log-ingestion.samsungacr.com
8	Nexia Home System	*.mynexia.com

Using reverse IP lookups, we confirm that all of the above IoT-servers are in the US.

TODO: Add a paragraph on automated IoT traffic separating by blacklisting popular domains and going through the list of domains

Privacy

To analyze privacy, we create a short script that searches for keywords in unencrypted HTTP packets. If a keyword is found in the payload or the URI, it prints the data. The script accepts a text file containing a list of searchable keywords.

We run our script on IoT separated captures from PIMA-TRY1 and PIMA-TOP and find that the separated IoT traffic uses TLS for the domains we've used. Our previous work [1] has shown that this is not true for all devices, for example, the PixStar Photoframe uses unencrypted packets to send information such as email addresses in clear text. However the homes we are monitoring do not have any such device.

Security

Comcast's threat intelligence servers contain daily reports on suspicious IP addresses from many databases. We poll the dambala-compromised QA database for IP destinations of the smarthomes. We use a script to generate all IP destinations for the 18 homes in PIMA-TOP, and run elasticsearch queries to the threat intelligence framework. The results return whether the IP address was marked and why. Our analysis shows that destination IP addresses were safe as

expected. Surprisingly, certain popular domains such as the open DNS server 8.8.8.8 returned a positive marking, but the reason for this was because it has been used in DNS redirect attacks. As this does not directly insinuate that the IP 8.8.8.8 is malicious, we have excluded results for redirect attacks.

Our approach of trying to find malware offline via packet capture is a challenging problem, and we expected no positive hits for this. However, it motivated us to present a trivial method for finding and pinpointing malicious devices if we can perform the above queries over a live stream for smart homes. Using DNS and offline analysis, we can list the homes and which IoTs they are expected to have. In case a queries show malicious destinations, we can filter the homes that were suspected. If there are any common devices between these suspected homes, there is a chance that the device is compromised. In that case Comcast may do a passive analysis by either capturing traffic, or an active analysis by testing the suspected device.

Lastly, during detailed data analysis, we observed that home number 14 from PIMA-TOP has repeated DNS queries towards xboxlive.com, microsoft.com, samsungacr.com, and hbo.com. The amplification attack is limited to only DNS queries and responses to/from the DNS server 75.75.76.76. Queries to 8.8.8.8 are only sent once, even when repeated, and queries for netflix are also only sent once. However, queries for *.xboxlive.com are sent at a rate of 14 packets per second, followed by queries to *.mp.microsoft.com after approx 200 seconds. We speculate that this is caused by a single misconfigured device, possibly the xbox. To confirm this, we would have to look at captures from this home when the Xbox was off.

Conclusion and Recommendations

Our work is a first step in analyzing IoT related activity passively from subscribers from a large ISP such as Comcast. We've face and solve various challenges such as capturing packets for low-activity IoT and identifying IoT related content at the CMTS level. We concentrate on manual analysis to learn how IoTs behave and present various solutions and suggestions to secure traffic from smart homes in the future. Using this approach, we find a potentially misconfigured device in one of the homes performing repeated queries to certain domains. To identify the cause, we would need to capture traffic while confirming that the suspected device is powered off.

We would like to recommend an ongoing DNS list for known IoTs so that Comcast can keep track of subscribers with smart homes as well as new IoTs that become popular in the network. This list would need to be manually updated on analyzing queries from homes for IoTs that we have not seen before. Furthermore, if we can check IP destinations from these smart homes frequently for threats, we might pinpoint IoT devices with malware.

A live streaming analysis framework for these known smart homes will also give us the capability of searching for privacy leakage in sampled HTTP packets to IoT destinations.

Niksun's NetVCR suite [3] or similar products would be the best to set up such a system for real time monitoring and analysis.

TODO: Add connectivity visualization

References

1. S. Grover, 'The Internet of Unpatched Things', PrivacyCon 2016
2. Y. Nadji, 'Passive DNS-based Device Identification',Nanog 2016
3. Niksun - NetVCR, https://www.niksun.com/product.php?id=110