Aim

- Identify IoT devices in DNS data from homes
- Analyze DNS patterns (space and time) to create profiles for homes with IoTs
- Use above to identify malicious activity or behavior

Data/Vantage point

1. Active collection from home router
2. DNS data from campus edge
3. DNS capture from edge of ISP

- containing { time, source IP, destination IP, DNS query/response }
- Data characteristics and challenges
    - At Comcast CMTS, 1 minute of DNS data packet capture was 677 MB
    - Of 2527293 queries, 502648 were IPv6
    - Of 260841 home IPs, approx 10% contained known IoT-related DNS queries
    - Found 11767 unique DNS servers

Main idea

Identifying IoT devices using DNS data

- Input: DNS traffic traces from a large number of houses, all devices within a house share the same IP address (behind NAT).
- Goal: Detect IoT devices
- Data: (domain, homeIP)
- Our definition of IoT (based on the hypothesis below): Internet connected device that performs certain specific DNS queries consistently in the background. Currently should not exist in many homes.
- Hypothesis:
    - Hypothesis 1: There will not be many homes with IoT related DNS queries, i.e. spatial prevalence of IoT domains is lower in aggregate data and vice versa.
        - Spatial prevalence of popular non-IoT domains is high, i.e. most popular domains across homes in dataset will be non-IoT related and can be filtered out using a popularity threshold (to be learned)
        - Count second-level domains across homes---IoTs should be in the filtered tail: `out1=in.map(p⇒(domain, homeIP),1).reduceByKey().filter(T1).map(p ⇒ (domain, homeIP))`
    - Hypothesis 2: Homes with IoTs will have similar query profiles, i.e. spatial correlation of IoT domains among filtered unpopular domains will be higher.
        - Remaining filtered homes with low frequency domains will contain domains (second level) that are either similar across homes, or completely different and individual

        - Clustering homes that have similar domains should give us IoT related domains
        - As false positive some domains that are visited by homes containing IoTs but not from the device will leak through
        - As true negative we will miss unpopular devices that only very few homes (maybe just 1 home?) have in the dataset
        - `out2=out1.map(p⇒(domain,1)).reduceByKey().filter(T2).map(domain)`
    - Hypothesis 3: Each IoT will perform at least 1 regular DNS lookup (heartbeat)
        - IoTs may be performing regular background data and DNS lookups, as well as specific DNS lookup that are only performed during activity
        - Counting the frequency of complete domain query (not second-level domain) will give us time based pattern of background DNS lookups across homes
        - These background lookups can be clustered and there should be a certain frequency of lookup
        - Activity based lookups should not have a consistent frequency
        - `iot_domains = in.filter(p.domain in out2).map(p ⇒ (domain,ts)).fold(inter_request_interval).map(p ⇒ (domain,interval)).reduce(count_frequency_of_intervals).filter(T3).map(domain)`
        - `home_2_iot_devices = in.filter(p.domain in iot_domains).map(p ⇒ (homeIP,f(domain))).groupByKey()`
- After filtering queries using above three, we may be able to identify new IoT related queries from multiple homes perform by checking the queried domain

Analyze DNS patterns (space and time)

- Data { domain, space (home IP count), time (frequency) }
- From above, we can create space (count of homes) and time (frequency of lookups) based pattern for second-level domain name queries that are potentially IoT related

Data Analysis

Active DNS Data

- Show DNS patterns for some lab devices in bg and active mode
- Frequency extraction of domain per IP
- Results: http://www.cs.princeton.edu/~sgrover/reports/Active-dataset-DNS-analysis-May2017.pdf

Feature Extraction + ML (OIT/DNS)

- Combine with hostdb (VM/host/OS) and DHCP (mac address/manufac)
- A records only, remove princeton.edu
- (list of features per sourceIP)
- Dealing with frequency per domain as feature for each sourceIP [challenge]
- Filtering mechanism (decision tree like)
    - Filtering: http://www.cs.princeton.edu/~sgrover/reports/Summaryof1nightanalysisforfitbitdevice-April2017.pdf
    - Feature Extraction: http://www.cs.princeton.edu/~sgrover/reports/FeatureExtractionforML-April2017.pdf
- ML algorithm like DBSCAN for non-number based features [challenge]
- Deal with complete database in Spark

System

- Use domain tree as a basic data structure
    - A real-time stream of DNS queries can be arranged as a domain tree for each home in a particular time window
    - In a particular window, all homes are children of the root, and all domains are arranged as leaf nodes with depth based on domain level
- Apply H1, H2, H3 to the domain tree structure
    - H1 and H2 will remove many nodes based on spatial correlation between each home tree - these nodes and their subtrees should be non-IoT related
    - H3 will let us check temporal correlation between multiple trees for the same home across time windows. This will further remove nodes that do not satisfy our definition of IoT as a device that performs particular queries in the background with a known frequency
    - We can use 'known devices' to find false positives/true negatives and also analyze reduction in size based on the three hypothesis
    - This tree should basically exist on disk and not in memory as it will be very large if we end up getting data from an ISP etc.
- The leftover tree structure contains IoT devices
    - Common domains between homes and across windows can now be separated out by correlating nodes and subtrees. These should be IoT related trees and can be maintained and updated with the real-time stream.
    - IoT trees can also maintain other information at each node such as time frequency or count in a particular window
    - Now if a change is observed during real time on any IoT-related subtree, we can perform further queries on the main tree to see how that change occurred. These changes could include the following examples:

        - An addition of new nodes and edges due to an update or a malware that is infecting many devices of the same kind.
        - A significant change in frequency/count could indicate many users actively using the IoT device in a correlated manner suddenly.

Identifying malicious behavior (Future Work/Apply on ISP data)

- Time profile based abnormalities
    - Maintain time based profile of IoT-related DNS queries per home
    - Say, we suddenly see a new domain pop up for a set of these IoT homes
    - We can identify the common device between these homes for which the new domain might have popped up at around the same time
    - This new domain might be a system update if it remains consistent and is not malicious, or it might be a malware domain/botnet that infects a particular device
    - If we have non-DNS packet capture data (headers only), we can create similar profiles for destination IPs of these filtered IoT-related homes and identify new destination IPs that might suddenly pop-up.

Related Work

- Sarthak Privacy Con ppt
- Yacin NANOG June 2016
- Comcast Summer Work
- TODO: find DNS analysis for clients from ISP vantage point
- TODO: find DNS/IoT related work
- Multiclass Classification Problem Basics:
    - Understanding Machine Learning: From Theory to Algorithms
- Spatial Decision Trees:
    - Focal-Test-Based Spatial Decision Tree Learning
- Device Fingerprinting:
    - Remote physical device fingerprinting
    - Study on OS Fingerprinting and NAT/Tethering based on DNS Log Analysis
- DNS-based classifications:
    - Building a Dynamic Reputation System for DNS
    - EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis

===