Is Anybody Home? Inferring Activity From Smart Home Network Traffic

Bogdan Copos*, Karl Levitt†, Matt Bishop‡, Jeff Rowe§
Department of Computer Science
University of California, Davis
Email: *bcopos@ucdavis.edu, †levitt@cs.ucdavis.edu, ‡mabishop@ucdavis.edu, §rowe@cs.cdavis.edu,

***Abstract*—As smart home devices are introduced into our homes, security and privacy concerns are being raised. Smart home devices collect, exchange, and transmit various data about the environment of our homes. This data can not only be used to characterize a physical property but also to infer personal information about the inhabitants. One potential attack vector for smart home devices is the use of traffic classification as a source for covert channel attacks. Specifically, we are concerned with the use of traffic classification techniques for inferring events taking place within a building.**

In this work, we study two of the most popular smart home devices, the Nest Thermostat and the wired Nest Protect (i.e. smoke and carbon dioxide detector) and show that traffic analysis can be used to learn potentially sensitive information about the state of a smart home. Among other observations, we show that we can determine, with 88% and 67% accuracy respectively, when the thermostat transitions between the *Home* and *Auto Away* mode and vice versa, based only on network traffic originating from the device. This information may be used, for example, by an attacker to infer whether the home is occupied.

I. Introduction

Smart home devices are becoming increasingly popular in households around the world. Nest Labs, one of the most popular manufacturers of smart thermostats and smoke detectors, is believed to have sold 440,000 smoke detector units over the span of four months in 2014 alone. Smart home devices are designed to help homeowners automate and simplify mundane tasks around their property. However, bringing internet connectivity to household devices has also introduced many security and privacy concerns. At the end of 2015, security researchers discovered a vulnerability in Barbie dolls which would allow attackers to not only steal personal information but also convert a doll into a spying device capable of listening into conversations [6]. In early 2016, security research from Rapid7 found vulnerabilities in Comcast's Xfinity Home Security system that would cause the system to not report when a property's windows and/or doors were compromised [19].

In this paper, we investigate how device-to-device and device-to-cloud smart home network traffic can be used to infer personal information. Specifically, we use traffic analysis techniques on network traffic generated by devices from Nest Labs to learn information about the presence of residents and other events occurring within the property. Traffic analysis is the process of intercepting and analyzing network packets in order to deduce information from patterns in communication. The experiments involve two smart home devices, a smart thermostat and a smart smoke and carbon dioxide detector.

The rest of the paper is organized as follows:

- Section II describes relevant previous work.
- Section III gives a detailed rundown of the devices used in this study and their features and capabilities.
- In Section IV the data collection process is described.
- In Section V, the methodology behind the traffic classification is explained.
- Section VI reports the findings of our analysis.
- Section VII describes how the findings were tested for validity and presents information about the accuracy of our findings.
- Section VIII discusses limitations of our approach.
- In section IX we provide some initial ideas for solutions and list possible future work.

II. Previous Work

Traffic analysis attacks were highlighted in "Attacks of the SSL 3.0 protocol" [16], by Wagner and Schneier who showed the URL of an HTTP GET request is leaked in SSL because cipher-texts fail to disguise the plaintext length.

Later, Cheng and Avnur [3] show that websites can be fingerprinted by performing traffic analysis of SSL encrypted web browsing traffic. Ever since, there have been a number of works [2], [7], [8], [10], [13], [15] exploring traffic analysis attacks using various features including source and destination attributes (e.g. address, port), protocol, packet and connection sizes, and even timing information (e.g. duration of connections, burstiness of transmissions).

Efforts have also been put into developing countermeasures for such attacks [5], [11], [18]. Countermeasure techniques include traffic padding and traffic masking. Another variation is in the implementation, whether server side, client side, or both. Recently, in "Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail" [4], Dyer, Coull et. al. provide the first comprehensive analysis of some of the proposed traffic analysis countermeasures and show why they fail to protect against attacks. The authors argue that there is no efficient solution.

Compared with previous works which focus on identifying websites accessed by a user, our work focuses on home automation devices. Our literature search was able to only identify a single paper [12] in which the authors predicted user behavior from wireless home automation communications. In this work, the authors conduct analysis on two installations of the HomeMatic home automation system. Specifically, without prior knowledge of the HomeMatic installations, they use the content of communications between devices to not only identify the devices within the home, but also user behavior. There are several differences between our work and theirs. Perhaps the biggest distinction is that the authors of that work had access to unencrypted traffic, while the traffic we obtained was mostly encrypted. The only unencrypted traffic we observed was comprised of HTTP requests to a weather service. Such traffic was ignored by our analysis and not used for our inference process.

III. Devices

The devices used for this study are the 2nd generation Nest Thermostat and the 2nd generation Nest Protect Wired. Both devices are equipped with 802.11 radios as well as 802.15.4 radios. However, at the time of this work, the 802.15.4 radios were not used by the devices. The device setup is performed with the use of the mobile Nest application. Through the application, users are asked to provide the SSID and password of their home network. This allows the devices to communicate with the Nest services. Unlike other home automation platforms, in the Nest ecosystem there is no central hub responsible for coordination of the devices. The devices can access the internet directly through the home's Wi-Fi router. Additionally, the Nest Protect devices are capable of communicating with each other, regardless of the presence of a Wi-Fi network, using the Nest Interconnect feature.

The Nest Thermostat has several features designed to optimize energy usage. The most popular features are the ability to learn a user's schedule and preferences, and the capability to detect motion. The Thermostat has three modes of operation: *Home*, *Auto Away*, and *Away*. In the *Home* mode, the temperature is optimized for the user's comfort and it follows the learned schedule and settings of the user. Lack of motion over extended periods of time will trigger the *Auto Away* mode, where the temperature adjusts for energy savings. The *Away* can manually be set by the user (from the device or the mobile application) if the user expects to be away for an extended period of time (e.g. before leaving for a vacation).

The Nest Protect also has some interesting features. It is equipped with a motion sensor and relays the information to the Nest servers to optimize the learning of the owner's schedule. The motion sensor can also be used to activate the Pathlight, a light ring designed to help illuminate dark areas in the presence of an individual. Instances of Pathlight activation are also communicated with the Nest servers and show up in the device's history.

Nest provides device owners with a web interface from which the devices can be accessed. The web interface allows the user to adjust temperature for the Thermostat, adjust the temperature schedule learned by the Thermostat, view historical data about the devices' states and energy usage, as well as inspect and modify various device settings and preferences (e.g. technical information, temperature units, Nest Sense, equipment information, etc.).

IV. Data Collection

Network traffic was collected over the period of a month using a standard netbook device. Because of the extended capturing time and the overhead a GUI tool would impose (e.g. Wireshark), we chose to use *dumpcap* [17], a command-line network traffic dump tool. In order to capture the Nest traffic, the netbook's network adapter was set to monitor mode. Monitor mode allows the network adapter to monitor all of the traffic within a wireless network. In other words, unlike promiscuous mode, monitor mode allows the netbook's network adapter to record traffic originating from and destined to the Nest devices. To limit the packet captures to just traffic to and from the Nest devices, we created a filter based on MAC addresses. Nest devices can easily be identified on the network due to the 3 byte Organizationally Unique Identifier (OUI). We used this knowledge for the generation of the capture filter.

After the traffic was collected, *airdecap-ng* [1] was used to decrypt the packets and remove the radiotap headers. This process decrypts the WPA encrypted traffic, however, it does not affect SSL/TLS encryption. While this slightly simplifies our analysis, we argue that it does not invalidate our findings, since packet size information is available even without decrypting WPA traffic.

The resulting packet capture files were processed using Bro [14] in order to generate connection logs. Bro is a network analysis framework. As part of the framework, Bro comes with a tool capable of parsing packet capture files and aggregating packets into connections, outputting logs describing various characteristics of the connections observed. The analysis was performed on the resulting connection logs.

Examining the connection logs generated after the first 72 hours of capture permitted us to get an idea about the nature and type of the network traffic. As expected, the devices contact a small finite number of services. The Nest Thermostat contacts 14 different hosts, including Google DNS servers, AWS servers, Nest weather service servers and others. Communication protocols include HTTP, NTP, DNS, and most commonly SSL/TLS. HTTP is only used by the Nest Thermostat to obtain weather data from the Nest servers. While the address of some remote services such as DNS and NTP change over time, some of the contacted IP addresses remain the same. The traffic for the Nest Protect device is similar except for the lack of NTP requests and the presence of the WEAVE protocol [9].

V. Traffic Classification

A. Traffic Characterization

To identify potential patterns or characteristics to further investigate, we generated graphs depicting various features of

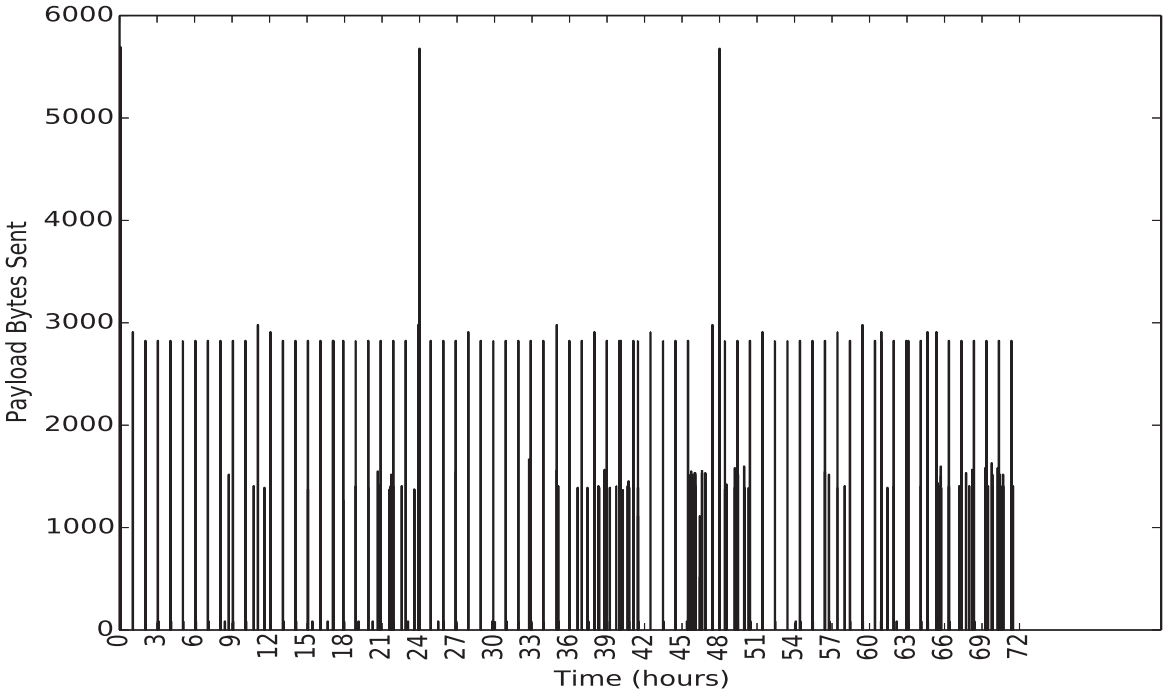

Fig. 1. This figure depicts connections made by the Nest Thermostat over the span of 3 days for IP address 54.204.245.223. Each connection is represented by the total number of payload bytes sent.

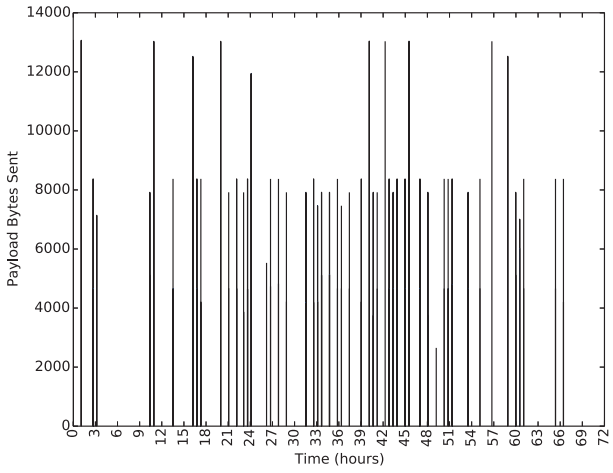

Fig. 2. This figure depicts connections made by the Nest Protect for IP address 107.21.244.221. Each connection is represented by the total number of payload bytes sent.

the network traffic generated by the Nest devices.

For example, Figure 1 depicts connections made by the Nest Thermostat over the span of 72 hours for the most often contacted IP address (Amazon AWS). Each connection is defined by the total number of payload bytes sent by the device throughout the connection duration. In the graph, we can see some patterns, such as the periodic connections of roughly 2800 bytes sent as well as large connections during which the device sent almost 6000 bytes. Additionally, we can see some seemingly random bursts of connections of approximately 1500 bytes. Figure 2 shows similar patterns with connections made by the Nest Protect to a particular IP. This IP also resolved to a Amazon AWS machine.

We also looked at the distribution of the sizes of connections made by the Nest Thermostat, shown in Figure 3. We define the size of a connection as the number of bytes transmitted by the Nest device. The connections are also organized by host (i.e. destination IP address). The figure brings to light several observations. First, it shows how for each host, the connections vary in the number of bytes sent. For example, we can see that for certain hosts, all connections have the same size. This is expected in cases such as NTP requests or for identical DNS requests. At the same time, this diagram allows us to visually identify clusters of connections of similar sizes for a given host. For example, for IP address 54.243.235.111 we can see several connections of approximately 25,000 bytes sent.

Having made such observations, we decide to focus on connection sizes for the rest of our analysis. Specifically, we attempt to determine if there are any associations between certain sized connections and the time of events of interest (e.g. device mode switch, alarm activation, pathlight activation).

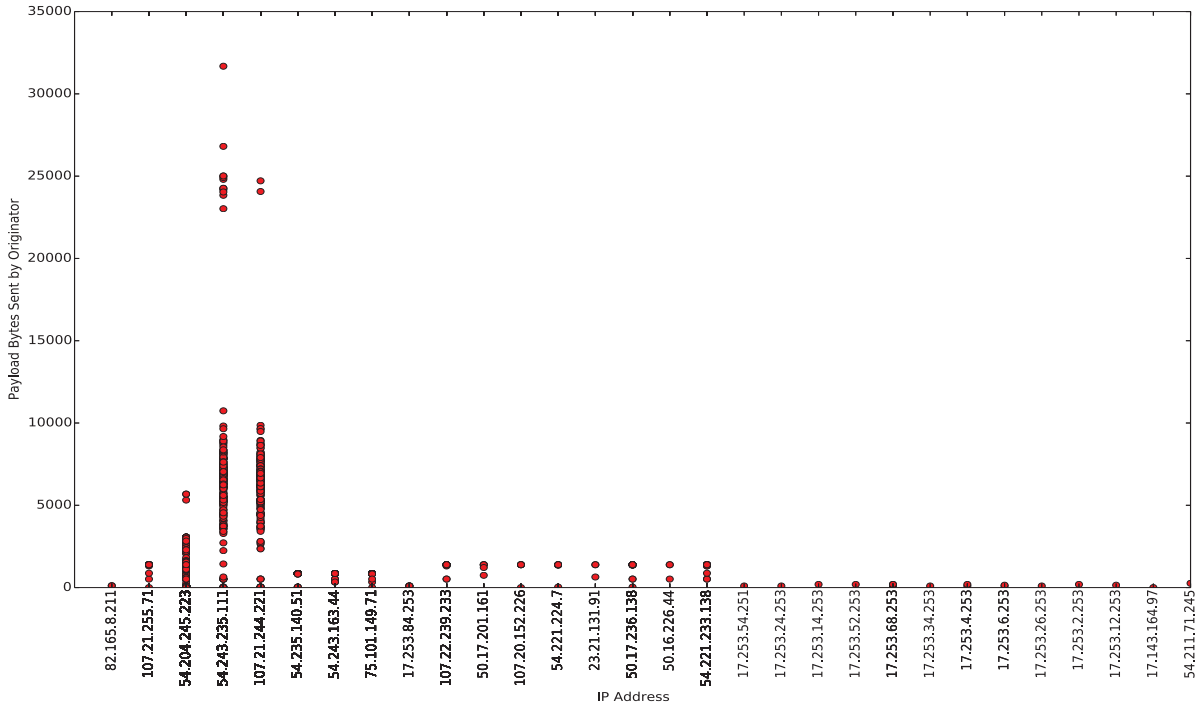

Fig. 3. Figure depicts the distribution of Nest Thermostat connections with respect to payload bytes sent. One can see that some connections are very similar in terms of the number of bytes sent.

B. Correlation Analysis

Our goal is to identify network patterns that allow us to infer information about the state of the devices and/or the state of the house. Specifically, we are interested in identifying patterns for the occurrence of events such as Thermostat operation mode tranisition (i.e. *Home* to *Auto Away* and vice versa), motion detection, smoke alarm activation and Pathlight activation. In an attempt to identify such patterns, we look for sets of one, two, and three connections that occur at the same time as the events of interest. For identifying sets of 2 and 3 connections, we use correlation analysis with a sliding time window. We analyze connections captured over the span of 60 days, only a small percentage of the total captured connections. Specifically, we process the connection logs with a sliding window of 10 seconds and a window displacement of 2 seconds. This means that the sliding window shifts by 2 seconds, causing a 8 second overlap between consecutive time windows. During this process, we generate a correlation matrix. The correlation matrix is a N by N matrix where N is the number of unique connections as defined by their destination IP and payload bytes sent. The matrix describes the number of occurrences of any two connections over the whole observation time. More precisely, the value of a given entry (i, j) in the correlation matrix is equal to the number of instances connections i and j occurred together in a 10 second time window. Connections are defined by the destination IP address and the number of bytes sent by the source (i.e. Nest device). We create a mapping between connections and indices in the correlation matrix. To increase time efficiency, we also generate SHA256 hash of each connection (defined by the destination IP address and bytes sent) for connection equivalence comparisons. A two dimensional correlation matrix allows us to find pairs of correlated connections. The same approach is extend to three dimensional matrix to allow the discovering of sets of three correlated connections.

To minimize the size of the connections analyzed, we apply several filters. First, we ignore connections with protocols other than NTP and SSL/TLS. We determine that correlations between connections with ICMP or DNS protocols are of no importance. Once the correlation matrix has been generated, we also filter out some of the correlated connections. To begin with, we eliminate all connections with low correlation (i.e number of occurences lower than 3). We determine that connections which rarely occur together are insignificant and can be disregarded. We also wish to discount regularly or frequently occurring connections. Such connections may appear to be highly correlated with other connections simply because of the number of occurrences, however the relationship between such connections may have no special significance. Distinguishing between significant and insignificant correlations in such cases is difficult. For example, one of the most frequent connections made by the Nest Thermostat is to IP address 54.204.245.223 with 2826 bytes sent. This connection shows up in the top 12 most correlated connection pairs. To carefully

filter out some of these correlations, we perform two tests. First, to identify regularly occurring correlated connections, we look at the difference between the timestamps of when the connections occur together. If this difference is consistent (plus/minus a small threshold), we ignore these connections. We also look at frequency of each connection and for a given pair of correlated connections, if both connections are very frequent, we ignore the correlation.

While filtering narrows the search space, it does not produce the desired patterns. To learn the patterns, we obtained the exact time of the occurrences of the events of interest from the Nest web interface. Having such information available, we were able to draw associations between connections (or sets of two and three connections) and the occurrence of an event. To finally select between patterns of one, two, and three connections, we choose the patterns with the best accuracy.

VI. Findings

Our analysis was successful in discovering network traffic patterns for Nest Thermostat operation mode switch, Nest Protect smoke alarm trigger, Nest Protect Pathlight activation. Additionally, we were able to make an interesting observation relating to the distribution of NTP requests. The details of each finding are described below independently.

A. Mode Transition

One of the events of interests was the transition between modes of operation in the Nest Thermostat. If the transition between modes was reflected in a unique identifiable pattern in the network communications between the Nest Thermostat and the Nest severs, such a pattern could be used to infer information about the occupancy status of a building. Our correlation analysis discovered a number of sets, of both size two and three, of correlated connections which occur during the Thermostat's transition from *Home* to *Auto Away*. The connections are identified by the destination IP address 54.204.245.223 and sizes 1375, 1391, and 2911. The correlated connection sets were comprised of permutations of these three connections. However, it should be noted that the set of three correlated connections had the best accuracy rate.

The transition in the opposite direction is identified by connections of to the same destination IP address. However, in this case, the connections have different sizes, specifically 1663, 1631, 1711, 1786, and 1819. In contrast to the transition from *Home* to *Auto Away*, in this case the single connections which occur at the time of the mode transition represent the mode transition the best. In other words, there are no sets of two or three correlated connections which occur during this mode transition.

It is important to note that our analysis also showed that these connections do not appear every time the device transitions between the two modes, which leads to false negatives.

B. Pathlight Activation

Another event of interest was that of the Nest Protect Pathlight activation. As with the mode transitions, if the event has an observable network pattern associated with it, an attacker could use this information to learn about the presence of individuals in a building.

Using the correlation analysis, we were able to identify a set of SSL connections of certain sizes (with respect to payload bytes sent by the device) which are observed together only when the device senses motion and the Pathlight activates.

C. Smoke Alarm Triggering

We also wanted to see if we could determine the triggering of the smoke alarm by observing the network traffic origination from the Nest Protect device. Our correlation analysis showed that two SSL connections of sizes 805 and 662 (with respect to payload bytes sent by the device) are observed together only when the device detects smoke and triggers the smoke alarm. Manual verification showed that there were no False Positives or False Negatives in our pattern recognition.

D. NTP requests

When looking at characteristics of the network traffic, we were surprised to observe NTP packets. Unfortunately, the correlation analysis wasn't able to identify any relationship between NTP requests and the mode of the Thermostat. However, manual investigation revealed that there is a discrepancy in the frequency of NTP requests generated between when the Nest Thermostat is operating in *Home* mode and when it is in *Auto Away* mode. Figure 4 show the occurrences of NTP requests during two days.

As mentioned earlier, one of the features of the Nest Thermostat is the ability to learn a user's schedule for energy optimization. Our hypothesis is that the device updates the Nest servers with motion (or lack of) activity, which includes a timestamp. To guarantee the accuracy of the timestamp, the Nest Thermostat will use NTP to synchronize its clock. For example, the thermostat will report to the server that the user is home or that the user is not home in comparison with activity from the previous day.

VII. Evaluation

The correlation analysis allowed us to identify certain patterns, as discussed above, which we manually checked. However, to further verify their validity and obtain accuracy measurements, we automatically test the remaining 21 days of network traffic for the presence of the discovered patterns. The results of this test were compared against the ground truth, obtained from the Nest web interface. As mentioned earlier, a user can login to the Nest web interface and obtain a log of events (e.g. mode change, smoke alarm) and their associated time of occurrence, as recorded by the devices.

A. Mode Transitions

For the transition between *Home* and *Auto Away* modes, our analysis resulted in 67% accuracy. For the transition in the opposite direction, from *Auto Away* to *Home*, the accuracy was 88%. However, it should be noted that in both cases, there were no false positives. In fact, further manual analysis showed

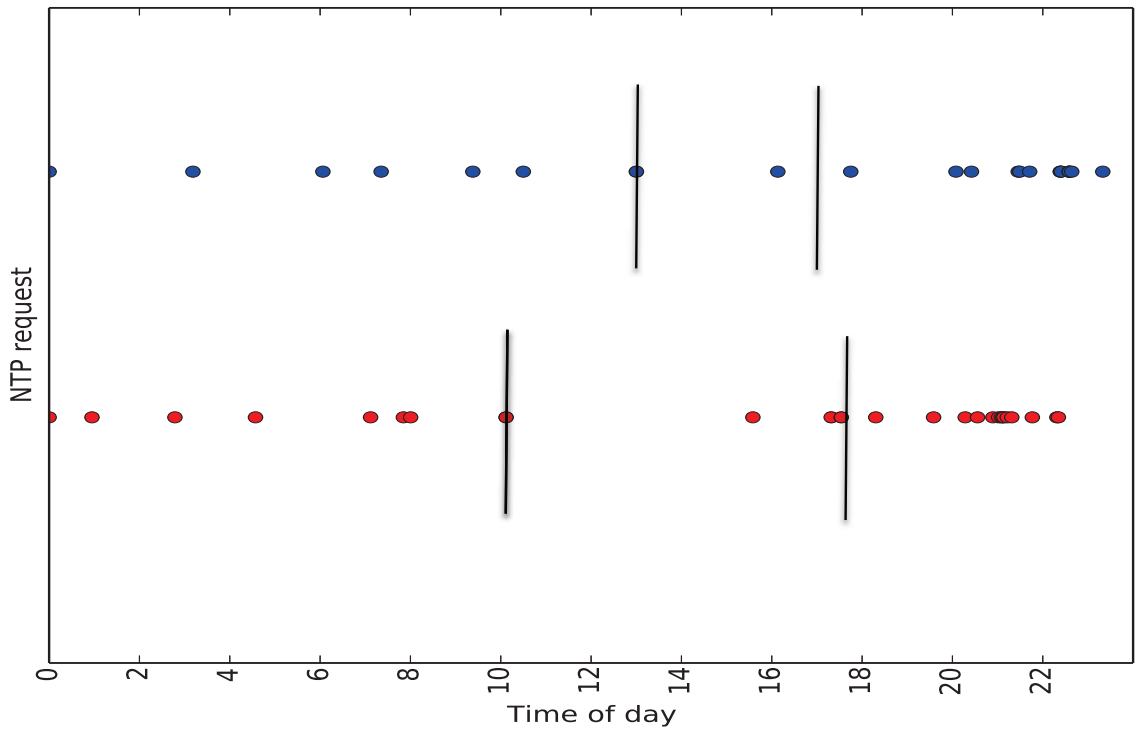

Fig. 4. This figure depicts the frequency of NTP requests made by the Nest Thermostat during two days. Each day is represented by a different color. The vertical bars represent the beginning and end of the *Auto Away* periods.

that in cases where there are multiple transitions throughout a single day, transitions after the second or third instance sometimes do not produce the identified pattern.

B. Smoke Alarm

When checking the validity of the network pattern associated with the smoke alarm triggering, our tests showed 100% accuracy. The analyzer was able to correctly identify all 5 instances of the smoke alarm being triggered, with zero False Negatives and zero False Positives.

C. Protect Pathlight Activation

Validation testing showed 50% accuracy (100% sensitivity). The sensitivity is a measure of the true positive rate. In other words, it expresses the proportion of positives that are correctly identified. This implies that the 50% accuracy rate is due to only False Positives (i.e. no False Negatives). Manual investigation shows that all of the False Positives occur due to the fact that the same unique connections repeat exactly 30 minutes after the initial occurrence. We are unable to explain why this occurs.

D. NTP requests

To test the validity of this observation, we used a simple Support Vector Machine (SVM) approach. Specifically, connection logs were split into non-overlapping one hour periods. To build the feature vector, for each period, the number of NTP requests during that period was obtained. Periods between hours of 12 AM and 6 AM were ignored due to lack of user activity and random distribution of NTP requests. Each hour period used for the learning process was also labeled as 0 or 1 (i.e. 0 means *Home* whereas 1 represented the device being in *Auto Away* mode).

The learned model resulted in 81% accuracy. Our testing approach did result in false positives (i.e. device was identified to be in *Auto Away* mode when it was not). To improve accuracy one could build a model where confidence in the classification adjuts according to the observations made. It should be noted that the start and end times of the *Auto Away* modes varied.

VIII. Limitations

We define connections by their destination IP address and the number of bytes sent by the Nest device. However, this only helps us find correlated connections with exactly the same size. It does not permit any flexibility. It is easy to imagine that packets containing semantically similar information may vary slightly in size (e.g. by a few bytes). Our approach fails to group such similar connections together, which may impact the correlation analysis results. At the same time, grouping connections together without ground-truth knowledge of their semantic similarity may lead to false positives in the correlation analysis and consequently in pattern recognition.

As mentioned earlier, the analysis was performed on unprotected (i.e. no WPA/WEP encryption) wireless traffic. This allowed us to not only aggregate packets into connections but

also uniquely identify connections by the destination IP and number of bytes sent. While we agree that Wi-Fi encryption would increase the complexity of such a covert channel attack, we argue it would not completely eliminate the attack. MAC address and size information can still be extracted from IEEE 802.11 packets. In fact, we manually verified that we could still identify NTP requests in WPA encrypted traffic captures using packet size information.

Furthermore, without more knowledge about the contents of the packets transmitted or source code of the software running on these devices, it is difficult to draw conclusions about the source of the False Positive and False Negative rates in our study. Early in our experiments, there was an instance where the Nest Thermostat lost connection with the remote servers. This could be due to incompatibility with the wireless routers, a publicly known issue. After resetting the Thermostat, we did not observe any more issues but without constant supervision it is difficult to say whether the connectivity issue resurfaced. Any such issue would cause the packet captures to be incomplete, which may increase the number of False Negatives and decrease the False Positives rate in our results.

The traffic analysis performed tries to correlate device events/actions with network activity at the time of the event/action. More specifically, our traffic analysis is unable to handle cases in which the devices cache the results of event/action and wait a period of time before transmitting the data to the cloud. While making traffic analysis more difficult, such delays do not make traffic analysis attacks infeasible especially if the delay period is not random. It should be noted that in certain scenarios, low latency is essential for device-to-device or device-to-cloud communications and delays may not be used as a defense mechanism.

IX. Conclusion

We have analyzed two of the most popular home automation devices, the Nest Thermostat and the Nest Protect and we have shown that home automation devices can leak sensitive information about what's happening inside any given property. We show that even if such devices use encryption to communicate, traffic analysis can be used on the network traffic for inference of information. Specifically, we show that features such as destination IP contacted, the numbers of bytes sent, the "burstiness" of packets sent can be used to fingerprint device activity that can be used to infer activity occurring within a home.

To ameliorate such traffic analysis attacks, there are several options. We argue that sending fixed amount of data at set time intervals is not a viable option. One reason is because such an approach would deplete the resources of such devices. Another reason is that these devices often need "fresh" data and withholding the transmission of data for a period of time may affect the quality of service provided by these devices. One solution is to pad packets to the same length and make all connections the same size. Additionally, instead of sending the packets to various remote servers, all packets would be destined to a single server. This server would be a proxy between the application servers and the smart home devices. Such an approach would make covert channel attacks more difficult since it disguises both destination and size information. However, the proposed approach does not protect against side channel attacks which exploit "burstiness". To make covert channel attacks even more difficult, devices could be designed to make randomly occurring deceptive connections.

For future work, we would like to determine what the appropriate balance between privacy and utility is. Another idea is to introduce a policy system, allowing users to specify the risks and cost (with respect to privacy) they are willing to accept.

References

[1] Aircrack-ng. airdecap-ng.

[2] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine. Privacy vulnerabilities in encrypted http streams. *Lecture notes in computer science*, 3856:1, 2006.

[3] H. Cheng and R. Avnur. Traffic analysis of ssl encrypted web browsing. *URL citeseer. ist. psu. edu/656522. html*, 1998.

[4] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 332–346. IEEE, 2012.

[5] X. Fu, B. Graham, R. Bettati, W. Zhao, and D. Xuan. Analytical and empirical analysis of countermeasures to traffic analysis attacks. In *Parallel Processing, 2003. Proceedings. 2003 International Conference on*, pages 483–492. IEEE, 2003.

[6] S. Gibbs. Hackers can hijack wi-fi hello barbie to spy on your children, November 2015.

[7] D. Herrmann, R. Wendolsky, and H. Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42. ACM, 2009.

[8] A. Hintz. Fingerprinting websites using traffic analysis. In *Privacy Enhancing Technologies*, pages 171–178. Springer, 2003.

[9] N. Labs. Nest weave.

[10] M. Liberatore and B. N. Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263. ACM, 2006.

[11] X. Luo, P. Zhou, E. W. Chan, W. Lee, R. K. Chang, and R. Perdisci. Httpos: Sealing information leaks with browser-side obfuscation of encrypted flows. In *NDSS*, 2011.

[12] F. Mollers, S. Seitz, A. Hellmann, and C. Sorge. Short paper: Extrapolation and prediction of user behaviour from wireless home automation communication. In *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless 38; Mobile Networks*, WiSec '14, pages 195–200, New York, NY, USA, 2014. ACM.

[13] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114. ACM, 2011.

[14] V. Paxson. Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[15] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 19–30. IEEE, 2002.

[16] D. Wagner, B. Schneier, et al. Analysis of the ssl 3.0 protocol. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, pages 29–40, 1996.

[17] Wireshark. dumpcap.

[18] C. V. Wright, S. E. Coull, and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *NDSS*, 2009.

[19] K. Zetter. Xfinity's security system flaws open house to thieves, January 2016.