Exploiting Known Vulnerabilities of a Smart Thermostat

Mike Moody
School of Computing and Academic Studies
British Columbia Institute of Technology
Burnaby, BC, Canada
Email: mikemoody@gmail.com

Aaron Hunter
School of Computing and Academic Studies
British Columbia Institute of Technology
Burnaby, BC, Canada
Email: aaron_hunter@bcit.ca

***Abstract*—We address security vulnerabilities for a smart thermostat. As this kind of smart appliance is adopted in homes around the world, every user will be opening up a new avenue for cyber attack. Since these devices have known vulnerabilities and they are being managed by non-technical users, we anticipate that smart thermostats are likely to be targetted by unsophisticated attackers relying on publicly available exploits to take advantage of weakly protected devices. As such, in this paper, we take the role of a 'script kiddy' and we assess the security of a smart thermostat by using Internet resources for attacks at both the physical level and the network level. We demonstrate that such attacks are unlikely to be effective without some additional social engineering to obtain user credentials. Moreover, we suggest that the vulnerability to attack can be further minimized by simply reducing the use of remote storage where possible.**

I. Introduction

Smart appliances use communication networks to operate more efficiently and more economically for home users. While the advantages of such appliances are well-known[6], [8], there are also established concerns related to security [3]. Some of the concerns are standard issues in cybersecurity: smart devices can be exploited to set up bot nets or to run malicious software in the home [2]. But there are also less technical concerns, related to the balance of privacy and accessibility of information. In this paper, we look at one particular kind of smart appliance: the smart thermostat. We survey known vulnerabilities, and we attempt to exploit a particular model of thermostat using publicly available scripts and attacks.

We focus explicitly on attacks that are publicly available and posted on the Internet. In the hacker community, an unsophisticated attacker that relies on this kind of attack is known as a *script kiddy* [4]. The term is generally seen as pejorative, as a script kiddy typically does not understand the underlying mechanisms of the attacks that they use. Nevertheless, the damage caused by this kind of unsophisticated attacker is still very real. We argue that script kiddy attacks are a particular problem for smart appliances, where many users rely on off-the-shelf security guarantees. Smart thermostats are currently being purchased and installed by users that are completely unaware of known security flaws, and there is a large community of unskilled hackers that may like to exploit this situation. As such, in this paper, we set out to explore the effectiveness of such attacks.

This paper makes several contributions to existing literature. First, by explicitly focusing on script kiddy attacks, we emphasize the distinct character of security flaws that are likely to be exploited widely for mass-market devices such as smart appliances. Second, in the case of one particular model of smart thermostat, we demonstrate that it is actually quite hard to use known exploits without having enough technical knowledge to tweak the attacks appropriately. Finally, our analysis points to a very simple vulnerability, with a similarly simple solution.

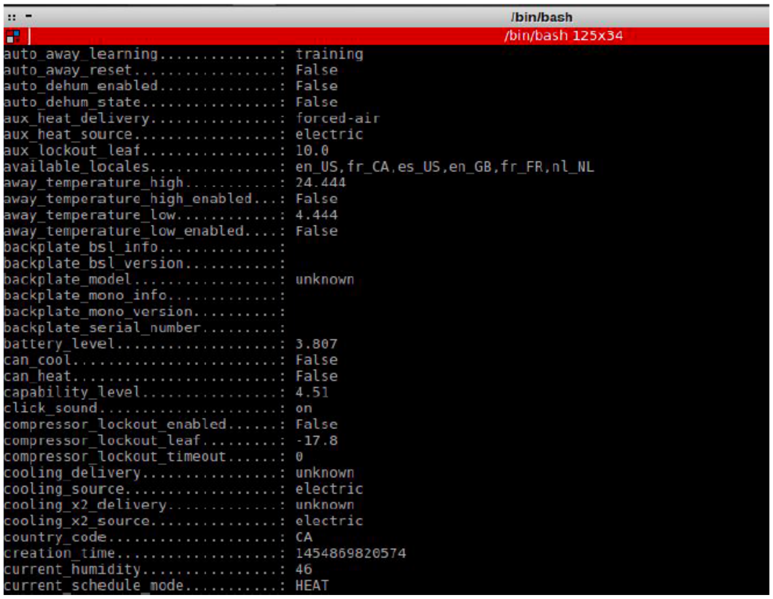
II. Approach

A. The Nest Thermostat

This paper focuses on Google's Nest thermostat. This device can automatically control the temperature of a home, detect usage patterns, and adapt to a user's preferences. Previous studies have provided mixed results in terms of the security of this device. In comparison with other smart thermostats, it has been demonstrated that the Nest uses encryption appropriately to protect data and it is not susceptible to remote access attacks[1]. On the other hand, given physical access to the device, it is vulnerable to exploits that allow an attacker to gain root access and install malicious software [2].

B. Methodology

We take an experimental approach to the analysis of Nest security. We use known exploits and tools for mining Nest data to determine exactly what can be obtained by a novice intruder. We focus on an intruder that must rely on known tools and existing attacks, because there are a large number of such intruders that might attempt to attack individual Nest devices. This is similar to the case of mobile phone security, where it has been demonstrated that indvidual users on GSM networks are actually highly susceptible to attacks that can easily be carried out by so-called script kiddies [5]. Our aim is to determine what an intruder needs to do in order to obtain useful information from a Nest thermostat, and also to provide a straightforward approach to retrieving this information. After this analysis is complete, we step back and take a higher-level perspective. Given the vulnerabilities that we discover, we look at how we can we improve ther overall communication and storage model.



```
auto_away_learning..............: training
auto_away_reset.................: False
auto_dehum_enabled..............: False
auto_dehum_state................: False
aux_heat_delivery...............: forced-air
aux_heat_source.................: electric
aux_lockout_leaf................: 10.0
available_locales...............: en_US,fr_CA,es_US,en_GB,fr_FR,nl_NL
away_temperature_high...........: 24.444
away_temperature_high_enabled...: False
away_temperature_low............: 4.444
away_temperature_low_enabled....: False
backplate_bsl_info..............:
backplate_bsl_version...........:
backplate_model.................: unknown
backplate_mono_info.............:
backplate_mono_version..........:
backplate_serial_number.........:
battery_level...................: 3.807
can_cool........................: False
can_heat........................: False
capability_level................: 4.51
click_sound.....................: on
compressor_lockout_enabled......: False
compressor_lockout_leaf.........: -17.8
compressor_lockout_timeout......: 0
cooling_delivery................: unknown
cooling_source..................: electric
cooling_x2_delivery.............: unknown
cooling_x2_source...............: electric
country_code....................: CA
creation_time...................: 1454869820574
current_humidity................: 46
current_schedule_mode...........: HEAT
```

Fig. 1. Script Output

This is a preliminary exploration and discussion paper. Our goal is to demonstrate at a very simple level the privacy risks and vulnerabilities in order to motivate further exploration, as well as high-level solutions that can easily be implemented.

III. Vulnerability Exploration

A. Overview

We follow a three stage approach to analysing the vulnerabilities of the Nest thermostat. First, we look at attacks where we have physical access to the device. Second, we analyze the security of the wireless data communicated by the device. Finally, we look at vulnerabilitiesin communication with the server.

B. Physical Security

There are well-known exploits that have been run on the Nest thermostat to install malicious code. Our goal is to determine the extent to which a naive intruder can use publicly available scripts to gain access to a Nest device. Towards this end, we use the publicly available Nest DeviceFirmware Upudate (DFU) Attack (https://github.com/gtvhacker/NestDFUAttack). To run this script, we need to have physical access to the Nest through the USB port. The script is purported to give root access to the device by exploiting the device firmware updater. In our trial, the script failed to give root access, presenting instead an error message. The support materials posted with the script instruct the user to try different USB cables, different physical machines, and different versions of Linux. In other words: for a novice intruder, the only solution to a problem with the script is to manipulate the physical configuration until it works. In our case, we could not gain root access, despite twelve attempts with different attack configurations. It is not clear if the Nest firmware has been updated to protect against the known attack, or if there was some other underlying issue. However, the difficulty that we experienced is likely common among novice users attemping to use this particular attack.

With physical access to the device, an intruder might also attempt to use forensic tools to extract data. For this purpose, we tested two free forensic software packages: AccessData's Forensic Toolkit(FTK) and Autopsy. These particular tools were used as they are freely available, and would therefore be the most likely options for a novice intruder. We found that the Nest thermostat contains a single XML file, which includes static information, such as the device's serial number, MAC address and software version. Without root access to the device, we were unable to extract any information from a new user account. Hence, using existing exploits and standard forensic tools, we essentially found the Nest to be resilient to physical attack without suitable user credentials.

C. Packet Analysis

Using Wireshark and Ettercap, we analyzed the packets across the network. The packets were captured over the course of 4 hours, during which specific events were triggered on the Nest device. Events included activating the motion sensor, changing the temperature manually on the Nest, changing the temperature using the iPhone app, changing the temperature on the Nest website and having the temperature change automatically at a scheduled time. The away mode function was tested by manually triggering the away mode on the iPhone app, website, and device. The auto away feature was tested by remaining outside of the room for over 2 hours where the Nest was installed.

A Wireshark packet capture over the course of four hours only returned a few packets. Each packet was encrypted using 128bit AES encryption, and no useable information was gained from these packets. While capturing packets with Wireshark, Ettercap was used to monitor network traffic by spoofing a router using an Address Resolution Protocol (ARP Poisoning). Several connections were being made to outside services, but these packets were strongly encrypted. The packet analysis did not provide any useful attack.

D. Known Credential Attacks

In the previous sections, we found that the Nest thermostat uses encryption effectively and limits communication in a way that actually makes it reasonably resilient to attack. In this section, we consider the case where an attacker has obtained access to login credentials for a particular user.

For most users, the Nest thermostat is controlled through the use of a mobile phone app that can provide detailed information about the current state of the device. In this study, we used a freely available Python script to retreive this information from the Nest servers, as this allows for greater flexibility and automation. A sample of the information obtained is given in Figure 1. We remark specifically on the final piece of information in the figure, which indicates the current schedule mode; this particular item will also show when the device is in *AWAY* mode, indicating that the owner is not home.

By using the script access approach at repeated intervals, it is quite easy to get an accurate use pattern for a device. Not only will this allow an intruder to accurately predict when a user is at home, but it will also provide a great amount of information about the heating patterns and heating technology in the home. This information could be used by a malcious user to deceive a home owner into granting physical access to the home. With physical access to the device, as well as login credentials or a suitable exploit script, the intruder could then gain control of the Nest for more nefarious purposes. Moreover, it is well-known that physical access to other smart devices running the ANSII protocol is a problem as passwords for such devices may still be stored in plain text [7]. As such, physical access to smart devides as a purported technician is clearly a major problem for home users.

IV. Discussion

A. Technical Issues

In our investigation, the Nest thermostat was resilient to a rooting attack as well as packet analysis. Moreover, the vulnerability of data is protected by login credentials in a standard manner that uses encryption effectively. Of course, we were restricting attention to a particular attack that has been posted online, and that is executable by a script kiddy. It actually appears that the fundamental idea behind the attack may still work. In other words, it may still be possible to exploit the Device Firmware updater; a new script or hack could be developed to take advantage of this. To combat this, device makers need to ensure that hardware level encryption is priority. But, at least for the moment, it appears that an attacker will need to be sophisticated enough to write this attack on their own.

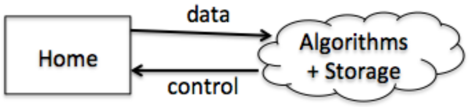


Fig. 2. Nest Communication

B. Practical Issues

Although the nest was reasonably reslilient against the given rooting attack, there are still potential problems with access to data. We noted that an attacker that obtains user credentials can actually learn a great deal of sensitive information. The issue is that information is stored remotely. The basic communication model is given in Figure 2, which shows that the Nest device sends user data to Nest servers in order to use learning algorithms to create a control schedule.

We remark that the thermostat need not operate in this manner. It would be possible to achieve the same behaviour without actually storing individual user data remotely, as in Figure 3. Historical data could be stored locally on the device, along with the algorithm that is used to learn preferences. The Nest is actually a computing device with enough power to run a simple learning algorithm to create a custom usage profile. Of course, the learning algorithms need to be maintained and updated periodically; this update process could be done through network communication, without actually transmitting sensitive user data. As a matter of policy, we should be considering such alternative cloud architectures for smart appliances.

C. Legal Issues

There are non-technical problems with the nest to be considered here as well. Certainly, in a forensic investigation, there is an advantage to having historical data strored remotely. This would be useful in the investigation of a robbery or a house fire, for example. However, the potential here for misconduct is great. In addition to the risks faced by an intruder with access to the thermostat, there are privacy concerns at play.

Over the last couple of decades, many countries have introduced new privacy laws. Under a transparent reading of applicable privacy laws in Canada, for example, the historical data being stored should be not used for any purpose other than controlling the Nest device. Tracking and storing private home-usage data of this sort without a targetted purpose is problematic; and the targetted purpose can certainly be achieved without remote storage. This is actually a general problem with smart devices in the home; the data being collected is rarely known to the user, and the use of this data is often hidden in privacy policies or term of service [2]. The Nest home thermostat is then another example of this emerging problem.

D. Future Work

One natural direction for future work would be to look at basic attacks on different thermostats, or related smart appliances. However, there is also value in continuing to

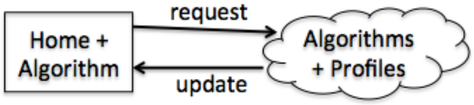


Fig. 3. Alternative Nest Communication

explore the security of the same device. The approach taken in this paper is to imagine an unsophisticated attacker that is reliant on known vulnerabilities and scripts to attack the Nest thermostat. Although we were not able to gain root access by simply using a publicly available script, the vulnerability is still present. In future work, we would like to explore how quickly new versions of this script appear online, as well as how quickly the device is patched to protect against this vulnerability.

V. Conclusion

In this paper, we have considered script kiddy attacks on the Nest thermostat. The underlying motivation has been to consider how vulnerable a popular smart appliance may be to widely-known attacks that can be carried out by a malcious intruder with little technical knowledge. We have seen that it may actually be difficult for a naive attacker to use such scripts effectively, as off-the-shelf performance was not successful even when we tried many different software configuations. The Nest also appears to use encryption appropriately, so that the most obvious vulnerabilities require user credentials. While this may seem like a sufficient guarantee of security, it is still possible to question why so much data needs to be stored remotely, as the potential for server-side attacks is still present.

References

[1] M. Burrough and J. Gill. *Smart Thermostat Security: Turning up the Heat*, (http://www.burrough.org/Documents/Thermostat-final-paper.pdf), accessed 2015.

[2] G. Hernandez, O. Arias, D. Buentello, and Y. Jin. Smart Nest Thermostat: A Smart Spy in Your Home. *Black Hat USA*, 2014.

[3] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin. Cyberphysical systems: A security perspective. *Proceedings of the 20th IEEE European Test Symposium(ETS)*, 2015.

[4] K. Mitnick and W. Simon. The Art of Deception: Controlling the Human Element of Security. Wiley, 2003.

[5] C. Ntantogian , G. Valtas, N. Kapetanakis, F. Lalagiannis, G. Karopoulos, C. Xenakis. Attacking GSM Networks as a Script Kiddie Using Commodity Hardware and Software. *Proceedings of the International Conference on Trust, Privacy and Security in Digital Business*, 2015.

[6] R. Robles and T. Kim. Review: Context Aware Tools for Smart Home Development. *International Journal of Smart Home*, 4 (1), 2010.

[7] J. Rrushi, H. Farhangi, R. Nikolic, C. Howey, K. Carmichael, and A. Palizban. By-design Vulnerabilities in the ANSI C12.22 Protocol Specification. *Proceedings of the ACM Symposium on Applied Computing*, 2015.

[8] R. Yang and M. Newman. Learning from a learning thermostat: lessons for intelligent systems for the home. *Proceedings of the ACM international joint conference on Pervasive and Ubiquitous Computing*, 2013.