Exploring the Security of the Nest Thermostat:
Obtaining Root Access and Searching for PUFs

Project Description
New and “smarter” technology allows us to interact with our communities and environments in previously unimaginable ways. Nest, a company recently acquired by Google, has redesigned common household products to be both more autonomous—learning about its homeowner’s habits and making adjustments on its own—and more connected—allowing remote control by other devices via Wi-Fi. However, with more user-aware products comes sensitive data collection. These products work by gathering, storing, and analyzing (e.g., machine learning) user data; access to such data can reveal personal information about a user’s home-life habits and provide real-time tracking. The Nest Thermostat\(^1\), for example, learns about the user’s schedule and preferred temperatures so that it can automatically adjust itself throughout the day. Such a “smart” device allows users to reap significant savings in energy bills; however, security vulnerabilities that have been found in the device indicate that the device itself and sensitive information can be compromised. A smart product is not necessarily a safe product—a company’s failure to provide adequate user data protection puts privacy at risk.

In this semester-long project, I will probe the Nest Thermostat in search of both weaknesses in its security and ways to increase the security in the device. First, I will reproduce published works to hack into the device and obtain root access. I will verify the compromised state by writing, loading, and executing an application on the Nest Thermostat, showing user control over the device. From here, I will search for Physical Unclonable Functions (PUFs), which are

\(^1\) The Nest Learning Thermostat: https://nest.com/thermostat/life-with-nest-thermostat/
physical features of the device (i.e. hardware characteristics) that are almost impossible to replicate but whose values can be obtained fairly easily (e.g., SRAM power-up state has shown some potential in current literature\(^2\)). PUFs have great potential in improving device security, especially as fingerprints to generate encryption keys. Hard-to-break keys will make it more difficult for devices to be compromised. Finally, after finding and verifying a PUF, I will build an application that utilizes the PUF as a proof of concept. The ultimate goal of the project will be to evaluate the security of the Nest Thermostat and to explore the potential of PUFs in improving the security of the device.

For this project, I will be in contact with Professor Stefan Katzenbeisser and André Schaller (PhD) of the Technical University of Darmstadt (TU Darmstadt, Germany) for some of their expertise with PUFs in addition to my advisor, Professor Jakub Szefer (Yale).

**Major Steps and Checkpoints**

1. Obtain root access to the Nest Thermostat by reproducing prior work done by the SSL group at UCF\(^3\) and Exploitee.rs\(^4\).
   a. Load a Linux kernel onto the Nest and add a SSH server.
   b. Verify SSH connection.
2. Modify the kernel or add SFTP/SCP functionality so that files can be uploaded to the Nest.
3. Test the available tools.
   a. Build an application.
   b. Cross-compile it for the Texas Instruments ARM processor on the Nest.
   c. Load and execute the application on the device.

---


\(^3\) The Security in Silicon Laboratory (SSL) group at the University of Central Florida (UCF) exploited a vulnerability in the Nest Thermostat’s boot process to compromise the device and to gain root access. They have published and presented their work at the Black Hat 2014 conference. [https://www.blackhat.com/docs/us-14/materials/us-14-Jin-Smart-Nest-Thermostat-A-Smart-Spy-In-Your-Home-WP.pdf](https://www.blackhat.com/docs/us-14/materials/us-14-Jin-Smart-Nest-Thermostat-A-Smart-Spy-In-Your-Home-WP.pdf)

\(^4\) Exploitee.rs, formerly GTV Hacker, is a group that formed in 2010 to hack Google TV. Since then, it has published instructions on obtaining root access to numerous consumer products, including the Google Chromecast, the Samsung SmartCam, and the Nest Thermostat, on their website. [https://www.exploitee.rs/](https://www.exploitee.rs/)
4. Explore and verify the existence of PUFs on the device.
5. Build an application that uses PUFs.

**Deliverables**

1. Code
   a. Simple application to verify tools and the device’s compromised state.
   b. Application that uses PUFs.
2. Detailed write-up so that others may reproduce the work.
   a. Step-by-step instructions
3. CPSC 490 requirements to be submitted to the CS Department:
   a. Final project report
   b. Webpage
4. Presentations
   a. For the lab group
   b. For the Davenport Senior Mellon Forum (date pending)

---

5 This list merely defines the minimum amount work I hope to accomplish.