Vacuum Cleaning Security

Pinky and the Brain Edition

Fabian Ullrich
IT Security Analyst @ ERNW GmbH
Heidelberg, Germany

Jiska Classen
Secure Mobile Networking Lab - SEEMOO
Technische Universität Darmstadt, Germany
Motivation

- A vacuum cleaning robot is in your house, has access to your Wi-Fi and knows many of your personal habits.
- Who of you owns a vacuum cleaning robot?
- (Own as it's YOUR robot, neither your neighbor's nor someone's on the Internet!)
Motivation

- A vacuum cleaning robot is in your house, has access to your Wi-Fi and knows many of your personal habits.
- Who of you owns a vacuum cleaning robot?
- Neato is one of the top vacuum cleaning robot models in the US.
- In Germany, Vorwerk has been selling vacuum cleaners forever (founded 1883). Their top model, a robot, is a rebranded Neato.
- Vorwerk won the test comparisons in Germany with their VR300/VR200.
Robots were harmed during our experiments! (Sorry for that...)
No customer data was leaked.
Neato was informed and fixed all issues in time.
Infrastructure & Security Features

UI & USB console
Infrastructure & Security Features

UI & USB console

Manual robot commands
Infrastructure & Security Features

UI & USB console

Manual robot commands

Account information

Setup only

Beehive
Infrastructure & Security Features

- **Nucleo**
  - Robot commands, status responses

- **Beehive**
  - Robot commands / status
  - Account information
  - Setup only
  - Manual robot commands

UI & USB console
Infrastructure & Security Features

- Secure boot
- Encrypted logs
- Signed firmware updates

- HTTPS
- RSA key (robot↔cloud)
- Secret key (robot↔user)

- HTTPS
- Secret key (robot↔user)

- HTTPS
- Secret key (robot↔user)

- HTTPS
- RSA key (robot↔cloud)

- HTTPS
- Secret key (robot↔user)
Contributions

- With all these security features, what could possibly go wrong?
Contributions

- With all these security features, what could possibly go wrong?

- We bypass secure boot on a vacuum cleaning robot to extract its memory.

- Our key findings are...
  - ...key findings!
  - And a QNX side quest.

- We also gained unauthenticated RCE on robots over the cloud.
Secure Boot Bypass

- Custom *AM335x* chip (guessed by size factor).
- **QNX 6.5 image** from *Foundry27* is bootable but crashes.

- Get QNX SDP, modify image, skip hardware initialization, reboot Neato system into custom image for **cold boot** attack, print all **RAM to the serial port**.

- Watchdog started by **Pinky**, 
- Cleaning logic binary started by **Brain**.

---

Jiska found this, got assigned CVE-2018-20785, fixed in Neato BotVac Connected D7 4.4.0-72 in December 2018 and for Vorwerk somewhen after January 2019.
Keys and their Purpose (1)

Secret Key
- Generated when associating a robot with a user account.
- Known by: robot, app and cloud components.
- **Individual** key for each robot/user account relation!
  - Used for authenticating commands to robot.

https://developers.neatorobotics.com/
Secret Key

- Generated when associating a **robot** with a **user account**.
- Known by: robot, app and cloud components.
- **Individual** key for each **robot/user account** relation!
  Used for **authenticating commands to robot**.

Header = Authorization: NEATOAPP [signature]

1. string_to_sign = serial + date + message_body
2. signature = HMAC_SHA256(secret_key, string_to_sign)
Keys and their Purpose (2)

RSA Key
- Robots have to initially send the secret key
  - Has to be authenticated.
- Secret key not that secret
  - Several third parties know it.
  - Cannot be used to authenticate the robot in the cloud.

RSA Key used to authenticate robot to cloud.
Keys and their Purpose (2)

RSA Key

- Robots have to initially send the secret key
  - Has to be authenticated.
- Secret key not that secret
  - Several third parties know it.
  - Cannot be used to authenticate the robot in the cloud.
  
  **RSA Key** used to authenticate robot to cloud.

Header = Authorization: NEATOBOT [serial]:[signature]

1 \[\text{string\_to\_sign} = \text{serial} + \text{http\_method} + \text{URI} + \text{date} + \text{body}\]

2 \[\text{signature} = \text{sign\_rsa\_sha256}(\text{string\_to\_sign}, \text{rsa\_private\_key})\]
Secret Key Entropy Reduction

```
1  rnd = rand();
2
3  time_shift[0:3] = time_now;
4  time_shift[4:6] = 0;
5  time_shift[7] = 16;
6  time_shift[8] = rnd + rnd / 0xFFFF;
7  time_shift[9] = entropy_reducing_math(rnd + rnd / 0xFFFF);
8  time_shift[10:15] = robot_MAC;
```
Secret Key Entropy Reduction

```c
 rnd = rand();

 time_shift[0:3] = time_now;
 time_shift[4:6] = 0;
 time_shift[7] = 16;
 time_shift[8] = rnd + rnd / 0xFFFF;
 time_shift[9] = entropy_reducing_math(rnd + rnd / 0xFFFF);
 time_shift[10:15] = robot_MAC;

 return 454;
```
Secret Key Entropy Reduction

```
1   rnd = rand();
2
3   time_shift[0:3] = time_now;
4   time_shift[4:6] = 0;
5   time_shift[7] = 16;
6   time_shift[8] = rnd + rnd / 0xFFFF;
7   time_shift[9] = entropy_reducing_math(rnd + rnd / 0xFFFF);
8   time_shift[10:15] = robot_MAC;
```

- Entropy relies on **time of robot linkage**.
  - One year = 25 bit
  - One hour = 12 bit

- There are multiple **offline attack** scenarios.

---

Fabian found and analyzed this beautiful piece of code! Got assigned CVE-2018-19441.
RSA Keys for Robot Authenticity (1)

- Encrypted RSA keys in /var/keys.
- \texttt{vendorPrivateKeyProduction} sounds promising!
- Let’s do some string de-obfuscation!
RSA Keys for Robot Authenticity (1)

- Encrypted RSA keys in /var/keys.
- `vendorPrivateKeyProduction` sounds promising!
- Let’s do some string de-obfuscation!

- **RSA key** is the **same** for all robots.
RSA Keys for Robot Authenticity (1)

- Encrypted RSA keys in /var/keys.
- `vendorPrivateKeyProduction` sounds promising!
- Let’s do some string de-obfuscation!

- **RSA key** is the same for all robots.

![Image of a robot and a hacker]
RSA Keys for Robot Authenticity (2)

- We are able to **impersonate arbitrary robots**.
  - Allows for multiple other attacks.
  - For example: **Leak victim’s smartphone IP**
RSA Keys for Robot Authenticity (2)

- We are able to **impersonate arbitrary robots**.
  - Allows for multiple other attacks.
  - For example: **Leak victim’s smartphone IP**

```plaintext
Nucleo

Robot location?

Robot location?

Does no longer get requests

Hacker

```

helio

I'm Testi


Share the moment

Robot location?
We are able to **impersonate arbitrary robots.**
- Allows for multiple other attacks.
- For example: **Leak victim’s smartphone IP**
We are able to **impersonate arbitrary robots.**
- Allows for multiple other attacks.
- For example: **Leak victim’s smartphone IP.**
We are able to **impersonate arbitrary robots**.
- Allows for multiple other attacks.
- For example: **Leak victim’s smartphone IP**

![Diagram showing the process of impersonating a robot through RSA keys.](image-url)
We ♥ QNX < 6.6

- Power plants, cars, and other critical applications run on QNX < 6.6. Today.
- The current version is 7.x, but many vendors might not have updated it so far.
- Default settings are no ASLR, no DEP :).
- The robots run on QNX 6.5 ... **Side quest: How secure is it?**

Fabian and Johannes created a PoC for privilege escalation and got assigned CVE-2019-8998. [http://support.blackberry.com/kb/articleDetail?articleNumber=000057178](http://support.blackberry.com/kb/articleDetail?articleNumber=000057178)
We ❤️ QNX < 6.6

- Power plants, cars, and other critical applications run on QNX < 6.6. Today.
- The current version is 7.x, but many vendors might not have updated it so far.
- Default settings are no ASLR, no DEP :).
- The robots run on QNX 6.5 ... Side quest: How secure is it?

![Diagram showing privilege escalation]

Fabian and Johannes created a PoC for privilege escalation and got assigned CVE-2019-8998. [http://support.blackberry.com/kb/articleDetail?articleNumber=000057178](http://support.blackberry.com/kb/articleDetail?articleNumber=000057178)
Unauthenticated RCE

- **Buffer overflow** in Nucleo cloud connection daemon.

Fabian found the crash and performed analysis, Jiska decrypted the coredump (CVE-2018-17177), Johannes did further analysis and implemented a PoC (CVE-2018-19442).
Unauthenticated RCE

- **Buffer overflow** in Nucleo cloud connection daemon.
- Can be triggered with requests to https://nucleo.neatocloud.com:4443/vendors/neato/robots/[robot_serial]/messages.
- The overflow is within parsing the authentication header, which means that we found an unauthenticated RCE!
- All services run as root.

- Fix: Authentication headers are validated on Nucleo.

Fabian found the crash and performed analysis, Jiska decrypted the coredump (CVE-2018-17177), Johannes did further analysis and implemented a PoC (CVE-2018-19442).
Security Implications (1)

- **IoT** product at home? **Keep it offline!**

- As a **customer:**
  - **Update** your robot.
  - **Hide your robot’s serial number!**
Security Implications (2)

- Connected ecosystem developers:
  - Using RSA, RNG, hashing, secure boot, encrypted logs, signed firmware updates sounds good...
  - Review cryptographic key components and root of trust assumptions.
  - **Dissecting one of your products** should not compromise security of the other products, i.e., similar keys.
  - Test your security relevant code in practice to uncover issues like the static secret key “random” function—check entropy before hashing.
Q&A