Finding security vulnerabilities in an IoT device

Adnan Benchaaboun
Mentored by Mr. Patrick Cross

Introduction

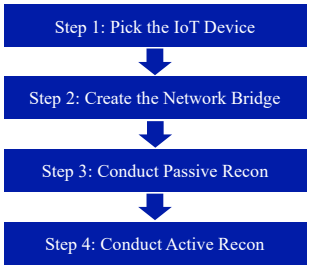
With the growth in popularity of Internet of Things (IoT) devices companies focused more on meeting the demand of the consumer rather than developing the security of these devices. The main reason for the rise in popularity of IoT devices is that they are everyday objects that can connect to the internet and send and receive data (Wood et al., 2017). IoT devices communicate by collecting data from sensors and uploading it to the cloud through either the internet or communications with other devices (Seralathan et al., 2018). These devices capture/receive data through packets, smaller chunks of larger data. This is done to minimize data loss. The device chosen for this project was the Go For IoT Smartphone Garage Door Opener Controller (Figure 1).

Figure 1 (above): The Smartphone Garage Door Opener Controller manufactured by Go For IoT.

One of the main benefits of the controller is that it can be accessed from anywhere. This is done by using a concept called port forwarding which changes the private IP address (i.e., home or workplace network) to a public IP address that can be accessed from anywhere.

One of the main issues with IoT devices is companies' negligence when it comes to security. For example, companies may implement their own encryption scheme instead of heavily tested and researched schemes. This lack of security allows many attackers to exploit vulnerabilities in victim's networks, gain access to victim's sensitive information and/or gain physical access to victim's homes. One faulty device could lead to the whole network being compromised. The main purpose of the project was to raise awareness of the many security concerns in IoT devices by determining vulnerabilities in the Garage Door Opener Controller and to give consumers the knowledge of the danger that comes with using these devices.

Materials and Methods



The project created a four-phase plan to determine vulnerabilities in the device (Figure 2). The goal of the first phase was to research devices that fit the budget and was highly susceptible to security vulnerabilities. The controller was chosen due to its low price and preconceived high susceptibility to vulnerabilities.

Figure 2 (left): Flow chart detailing the four-phase plan used for this project.

Materials and Methods (cont.)

The second phase is explained in Figure 3.

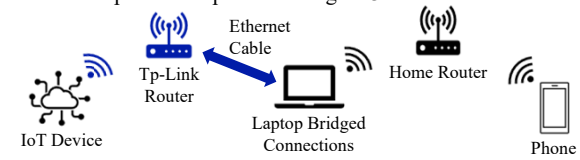


Figure 3 (above): This diagram explains the second phase of the project. The second phase created a network bridge that enabled a computer to capture every packet sent between the phone and the device.

The goal of the third phase was to capture and analyze packets sent between the phone and the IoT device to find vulnerabilities. To capture and analyze packets, an older version of the Wireshark software (v 1.10.8) was used. Once Wireshark was opened, the ethernet interface was selected to capture packets. This interface isolated all communications so that only packets between the phone and the device were observed.

The goal of the fourth phase was to conduct a more aggressive approach to find vulnerabilities through a software called Packet Sender. Two attacks can be performed using this tool: A replay attack on captured packets where packets are modified, by deleting chunks of the password, changing hex values in different fields, changing the length of different fields to find any vulnerabilities, then resent to induce change in the IoT device; and a brute force attack which tests every combination of hex values (0–9, A–F) until a success is achieved.

Results

The full packet for the open command is in Figure 4 and close command in Figure 5. Through passive recon, it was determined that the only difference in each packet sent was the command and command length fields. This meant that no standard encryption scheme was used in the device such as Transport Layer Security (TLS). All standard encryption schemes require that each packet sent has a different set of randomized characters. This device instead sends the same packet every time without anything being randomized.

In active recon, one of the packets captured in Wireshark was

```
http://10.0.0.244:17117/GoForIoTArgs?
command=E1CDE6A980A786B5E983421EF2A1DC0B&
cmdlen=10&
id=94CB4CCB6584F3726D995CDE&
idlen=5&
pwd=C6268ECC279A996B859B44667CE4FFB764F64F10&
pwdlen=15&
os=IOS&
inputchange=0&
category=&
device=Device&timezone=05&
type=INPUT&key=GoForIoT@2017
```

Figure 4 (above): Open Command

```
http://10.0.0.244:17117/GoForIoTArgs?
command=A09E6433BCC0B2D08F7D830BA8081017&
cmdlen=11&
id=94CB4CCB6584F3726D995CDE&
idlen=5&
pwd=C6268ECC279A996B859B44667CE4FFB764F64F10&
pwdlen=15&
os=IOS&
inputchange=1&
category=&
device=Device&timezone=05&
type=INPUT&key=GoForIoT@2017
```

Figure 5 (above): Close Command

Results (cont.)

modified with the Packet Sender software (Figure 6). When chunks of the password and/or id were changed, the device responded with "INVALID ID/PWD". Once the correct credentials were reinputted, the device responded "OK" and opened/closed the garage. If parts of the command field were changed, the device responded "FAIL" and no action was taken. When the length fields were modified, the device still responded "OK". When the packet was sent from any other device connected to the same network as the phone, the device still responded "OK".

Figure 6 (left): Picture of the packet sender GUI during one of the tests. The password field has chunks of it removed and is not the normal length. The device would than respond to this request with "Invalid ID/PWD".

Conclusions

This project successfully created an isolated network for the Go For IoT Smartphone Garage Door Opener Controller and found security vulnerabilities through passive and active recon. The brute force attack method was confirmed to be not applicable since the password is too long and would take too long to crack. The replay attack method was confirmed to work as the same packet was modified then resent successfully from another device. Attackers can tamper with the victim's garage door if they get access to the victim's network. This provides a less secure entry point into the network that can be exploited by attackers. Further research may include accessing more IoT devices on the victim's network to take control over the entire smart home, and most importantly, providing viable and simple solutions for dealing with the vulnerabilities mentioned above. This project is adaptable, and the methods may apply for any project that tests for vulnerabilities in IoT devices created in the future.

References

Seralathan, Y., Oh, T. T., Jadhav, S., Myers, J., Jeong, J. P., Kim, Y. H., & Kim, J. N. (2018). IoT security vulnerability: A case study of a web camera. *IEEE*, 172–177. https://dio.org/10.23919/ICACT.2018.8323686

Wood, D., Apthorpe, N., & Feamster, N. (2017). Cleartext data transmission in consumer IoT medical devices. *IoT S&P'17*, 7–12. https://doi.org/10.1145/3139937.3139939