I'm the One Who Doesn't Knock

Unlocking Doors from the Network

David Tomaschik
Google Security Team

About Me

- Senior Security Engineer, Google Security Assessments
  - Predominantly Red Teaming
  - Also Breaking IoT

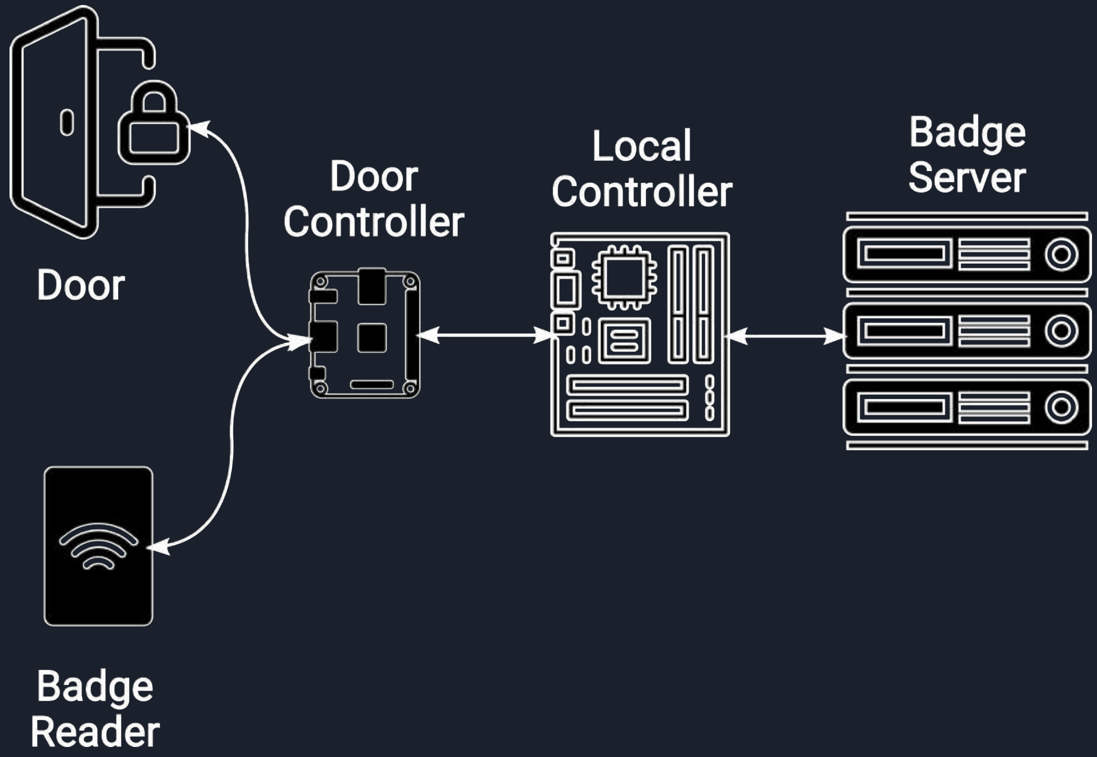
About Me

- Senior Security Engineer, Google Security Assessments
  - Predominantly Red Teaming
  - Also Breaking IoT
- Personal Interests
  - Breaking IoT for Fun
  - Making (Electronic) Things

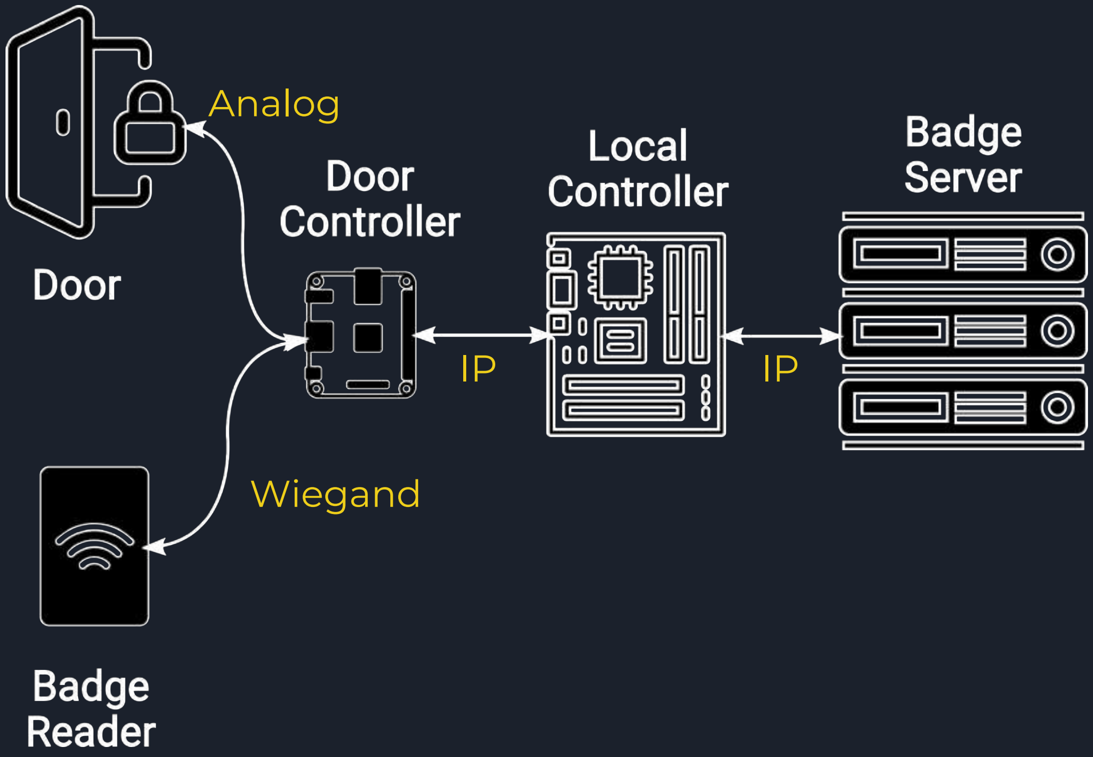
Outline

- Story Time
  - Realizing something is broken
  - Figuring out how broken
  - Figuring out how to exploit
- Discussion
  - How do we fix this?
  - Why is the fix not the same as for client/server applications? (HTTPS, etc.)

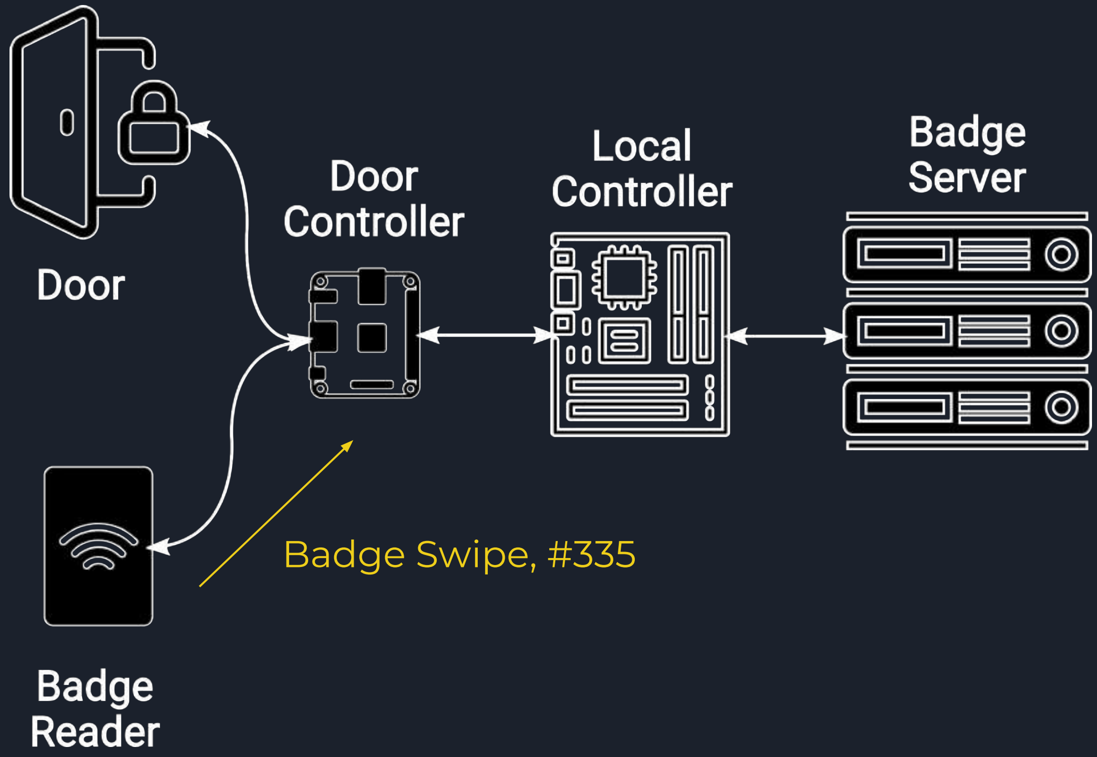
Door Access Control (Typical)



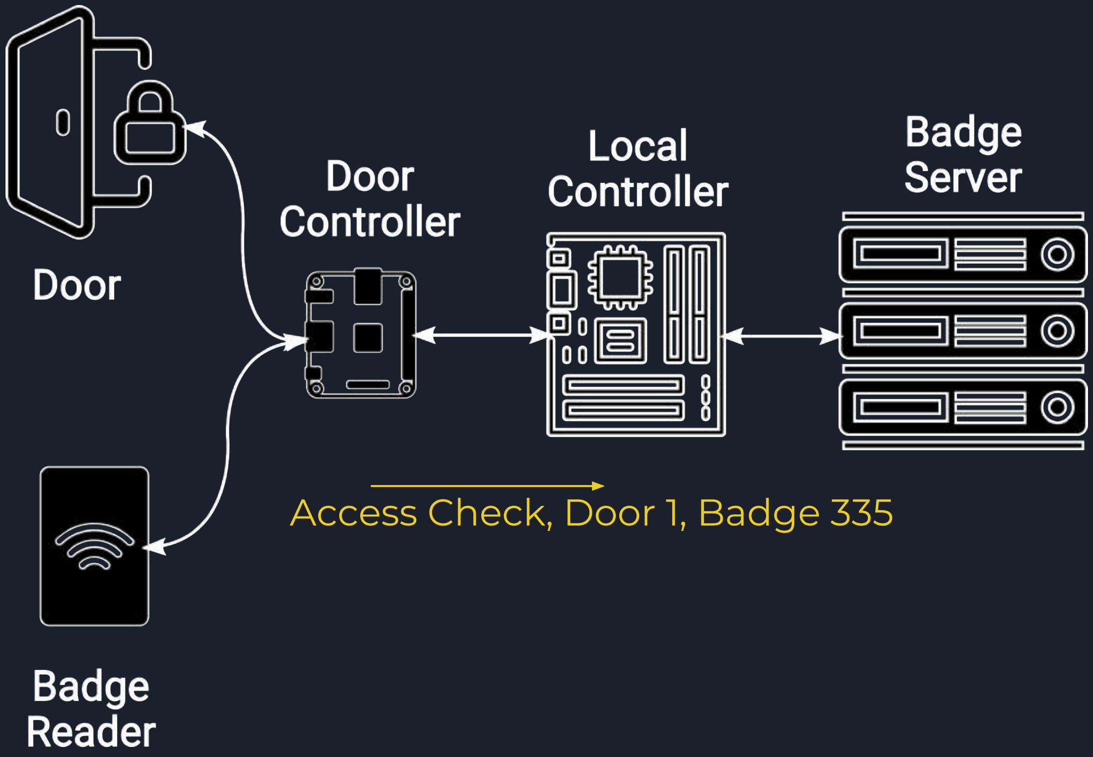
Door Access Control (Typical)



Door Access Control (Typical)

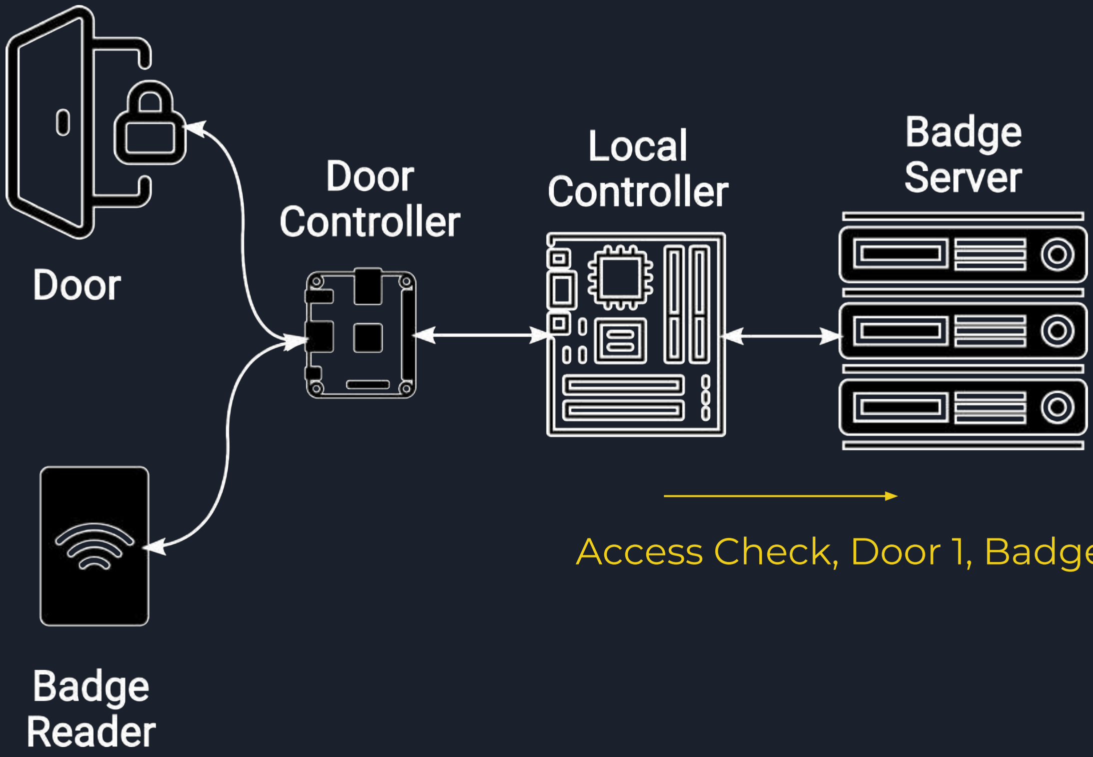


Door Access Control (Typical)



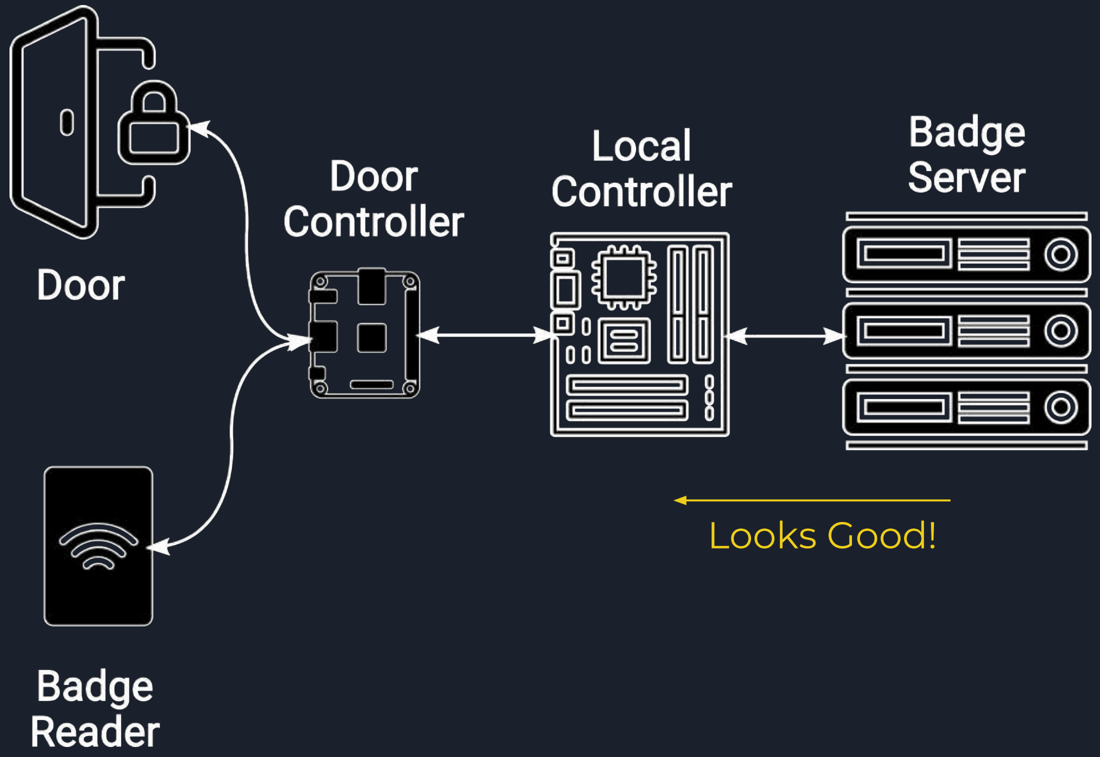
Access Check, Door 1, Badge 335

Door Access Control (Typical)



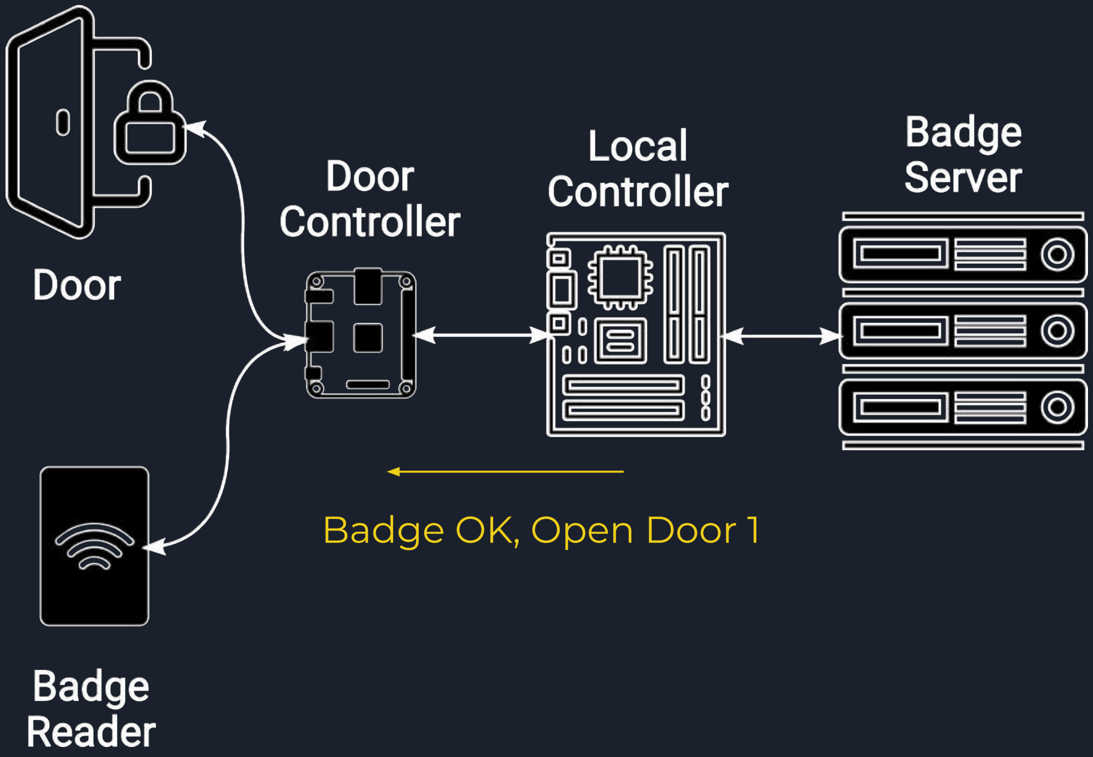
Access Check, Door 1, Badge 335

Door Access Control (Typical)



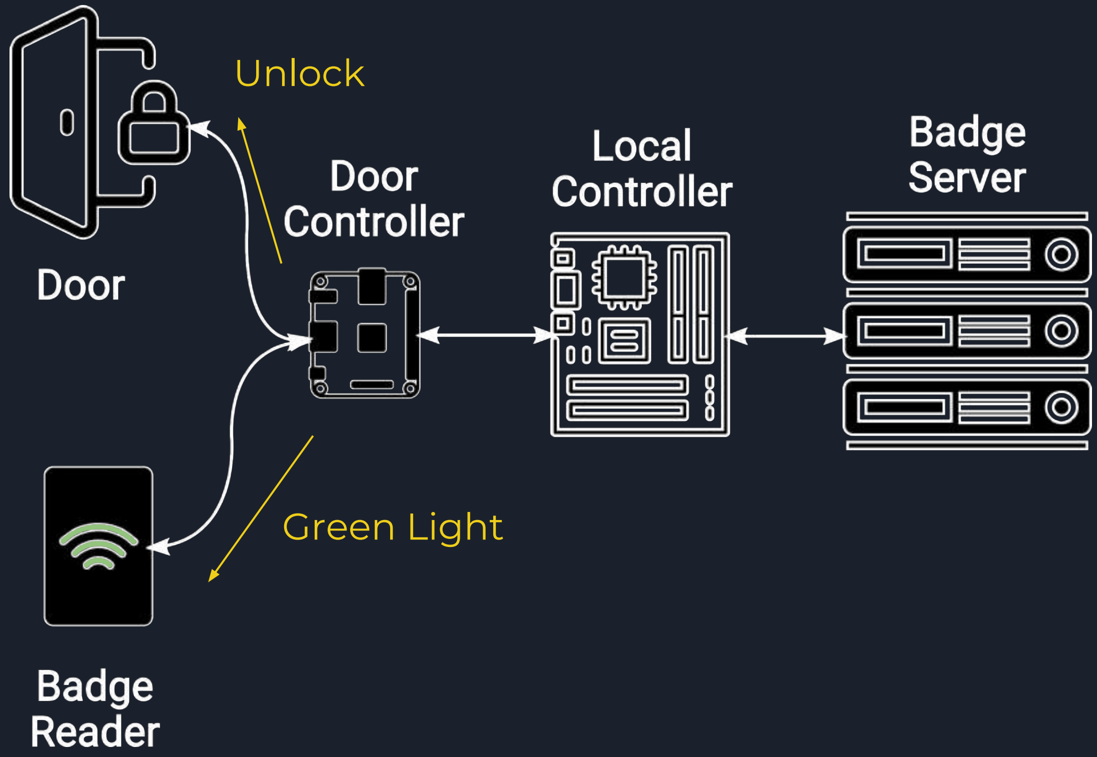
Looks Good!

Door Access Control (Typical)

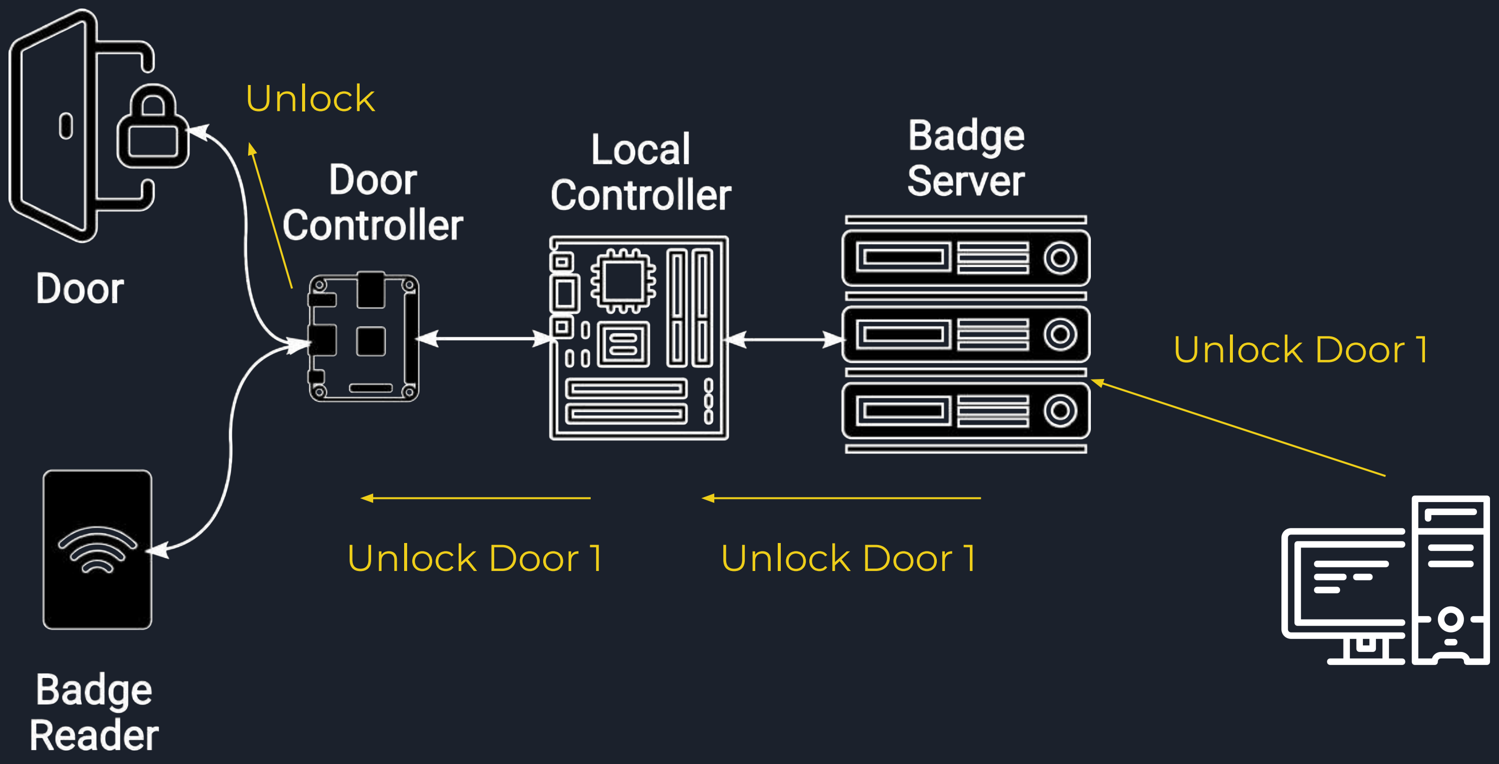


Badge OK, Open Door 1

Door Access Control (Typical)



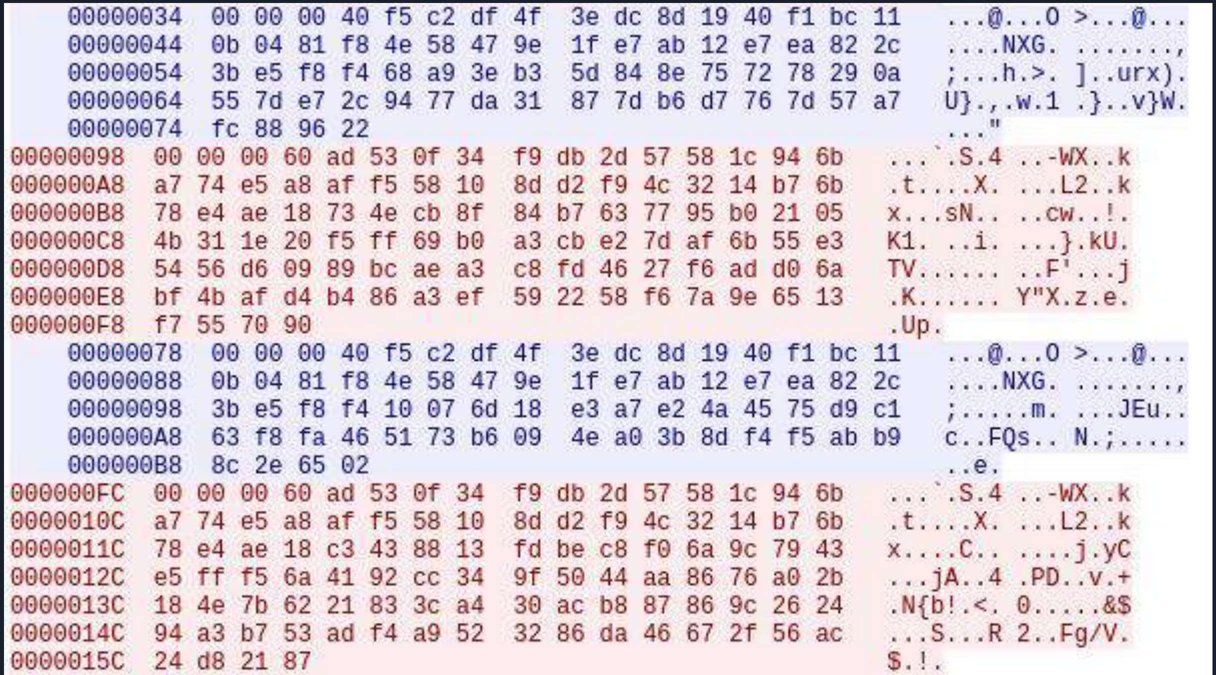
Door Access Control (Remote Unlock)



Once Upon a Time...

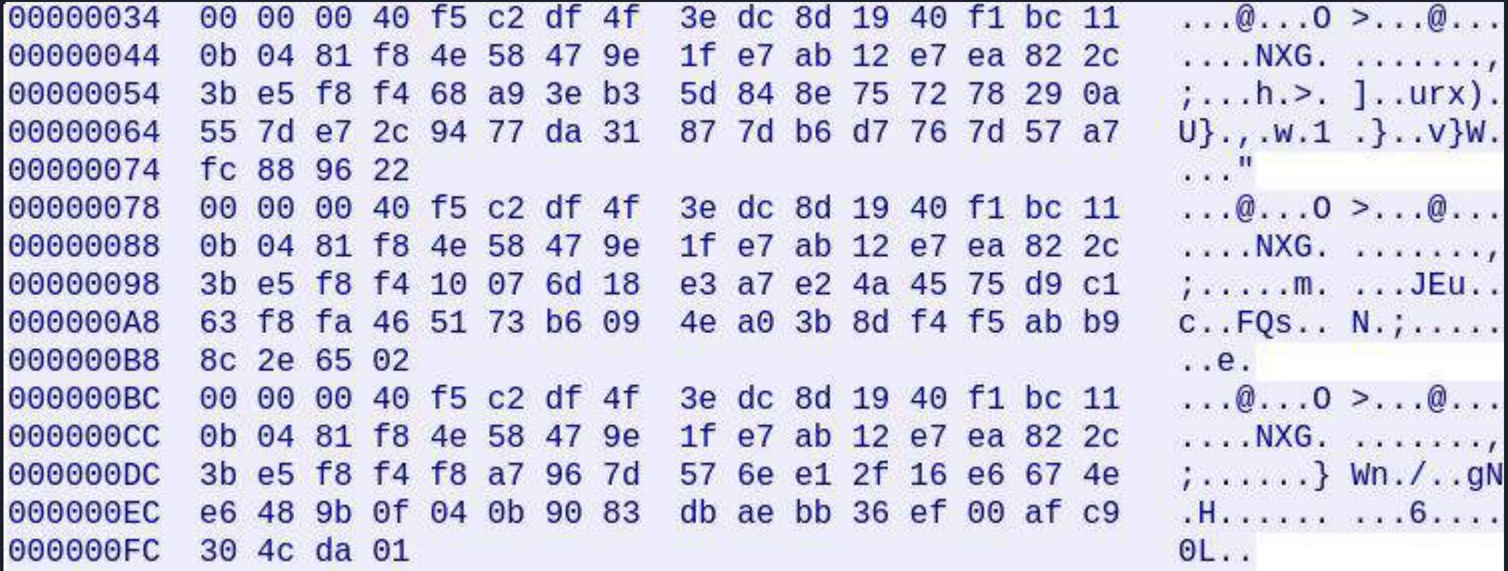
- Executing a Red Team
- Patch panel in area accessible to contractors
- Traced cables to door controllers
- Dumped traffic for later analysis

Traffic Analysis



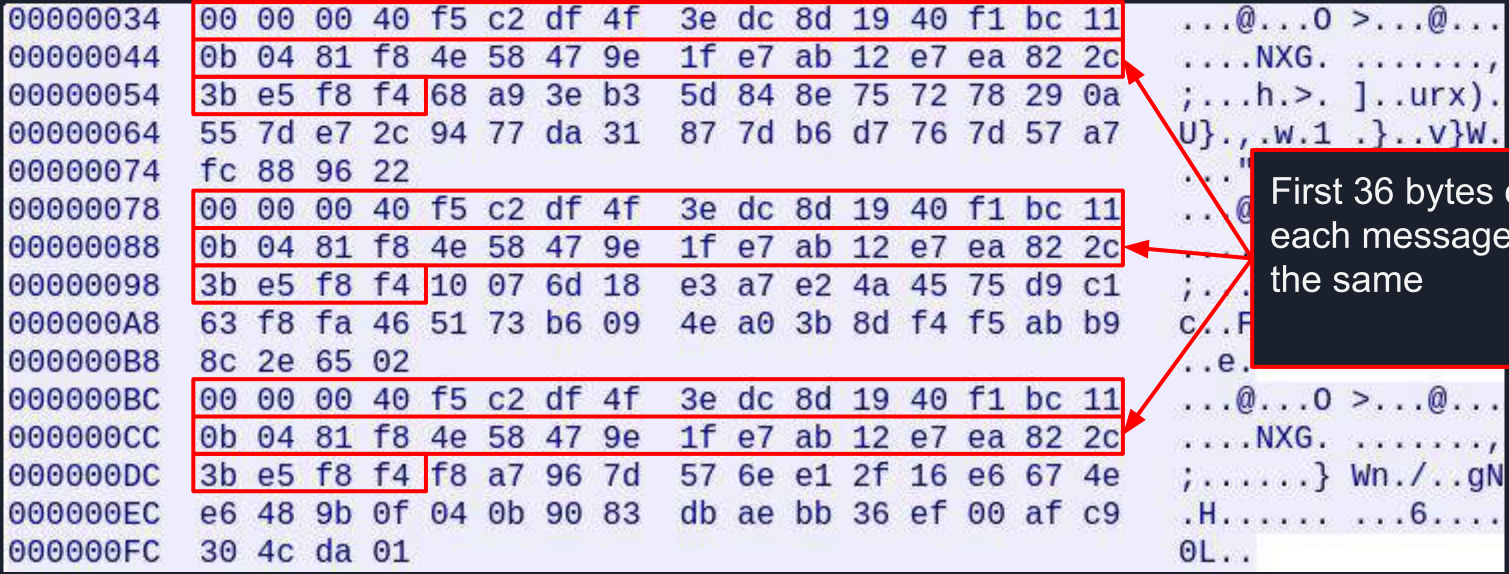
```
    00000034  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
    00000044  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
    00000054  3b e5 f8 f4 68 a9 3e b3  5d 84 8e 75 72 78 29 0a   ;...h.>. ]..urx).
    00000064  55 7d e7 2c 94 77 da 31  87 7d b6 d7 76 7d 57 a7   U}.,.w.1 .}..v}W.
    00000074  fc 88 96 22                                        ..."
000000098  00 00 00 60 ad 53 0f 34  f9 db 2d 57 58 1c 94 6b   ...`.S.4 ..-WX..k
000000A8  a7 74 e5 a8 af f5 58 10  8d d2 f9 4c 32 14 b7 6b   .t....X. ...L2..k
000000B8  78 e4 ae 18 73 4e cb 8f  84 b7 63 77 95 b0 21 05   x...sN.. ..cw..!.
000000C8  4b 31 1e 20 f5 ff 69 b0  a3 cb e2 7d af 6b 55 e3   K1. ..i. ...}.kU.
000000D8  54 56 d6 09 89 bc ae a3  c8 fd 46 27 f6 ad d0 6a   TV...... ..F'...j
000000E8  bf 4b af d4 b4 86 a3 ef  59 22 58 f6 7a 9e 65 13   .K...... Y"X.z.e.
000000F8  f7 55 70 90                                        .Up.
    00000078  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
    00000088  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
    00000098  3b e5 f8 f4 10 07 6d 18  e3 a7 e2 4a 45 75 d9 c1   ;.....m. ...JEu..
    000000A8  63 f8 fa 46 51 73 b6 09  4e a0 3b 8d f4 f5 ab b9   c..FQs.. N.;.....
    000000B8  8c 2e 65 02                                        ..e.
000000FC  00 00 00 60 ad 53 0f 34  f9 db 2d 57 58 1c 94 6b   ...`.S.4 ..-WX..k
0000010C  a7 74 e5 a8 af f5 58 10  8d d2 f9 4c 32 14 b7 6b   .t....X. ...L2..k
0000011C  78 e4 ae 18 c3 43 88 13  fd be c8 f0 6a 9c 79 43   x....C.. ....j.yC
0000012C  e5 ff f5 6a 41 92 cc 34  9f 50 44 aa 86 76 a0 2b   ...jA..4 .PD..v.+
0000013C  18 4e 7b 62 21 83 3c a4  30 ac b8 87 86 9c 26 24   .N{b!.<. 0.....&$
0000014C  94 a3 b7 53 ad f4 a9 52  32 86 da 46 67 2f 56 ac   ...S...R 2..Fg/V.
0000015C  24 d8 21 87                                        $.!.
```

Traffic Analysis



```
00000034  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
00000044  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
00000054  3b e5 f8 f4 68 a9 3e b3  5d 84 8e 75 72 78 29 0a   ;...h.>. ]..urx).
00000064  55 7d e7 2c 94 77 da 31  87 7d b6 d7 76 7d 57 a7   U}.,.w.1 .}..v}W.
00000074  fc 88 96 22                                        ..."
00000078  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
00000088  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
00000098  3b e5 f8 f4 10 07 6d 18  e3 a7 e2 4a 45 75 d9 c1   ;.....m. ...JEu..
000000A8  63 f8 fa 46 51 73 b6 09  4e a0 3b 8d f4 f5 ab b9   c..FQs.. N.;.....
000000B8  8c 2e 65 02                                        ..e.
000000BC  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
000000CC  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
000000DC  3b e5 f8 f4 f8 a7 96 7d  57 6e e1 2f 16 e6 67 4e   ;......} Wn./..gN
000000EC  e6 48 9b 0f 04 0b 90 83  db ae bb 36 ef 00 af c9   .H...... ...6....
000000FC  30 4c da 01                                        0L..
```

Traffic Analysis



```
00000034  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
00000044  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
00000054  3b e5 f8 f4 68 a9 3e b3  5d 84 8e 75 72 78 29 0a   ;...h.>. ]..urx).
00000064  55 7d e7 2c 94 77 da 31  87 7d b6 d7 76 7d 57 a7   U}.,.w.1 .}..v}W.
00000074  fc 88 96 22                                        ..."
00000078  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@
00000088  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....
00000098  3b e5 f8 f4 10 07 6d 18  e3 a7 e2 4a 45 75 d9 c1   ;...
000000A8  63 f8 fa 46 51 73 b6 09  4e a0 3b 8d f4 f5 ab b9   c..F
000000B8  8c 2e 65 02                                        ..e.
000000BC  00 00 00 40 f5 c2 df 4f  3e dc 8d 19 40 f1 bc 11   ...@...O >...@...
000000CC  0b 04 81 f8 4e 58 47 9e  1f e7 ab 12 e7 ea 82 2c   ....NXG. .......,
000000DC  3b e5 f8 f4 f8 a7 96 7d  57 6e e1 2f 16 e6 67 4e   ;......} Wn./..gN
000000EC  e6 48 9b 0f 04 0b 90 83  db ae bb 36 ef 00 af c9   .H...... ...6....
000000FC  30 4c da 01                                        0L..
```

First 36 bytes of each message the same

From the Product Brief

AES-256 network encryption

I’m not a cryptographer, but I’m pretty sure they’re doing it wrong.

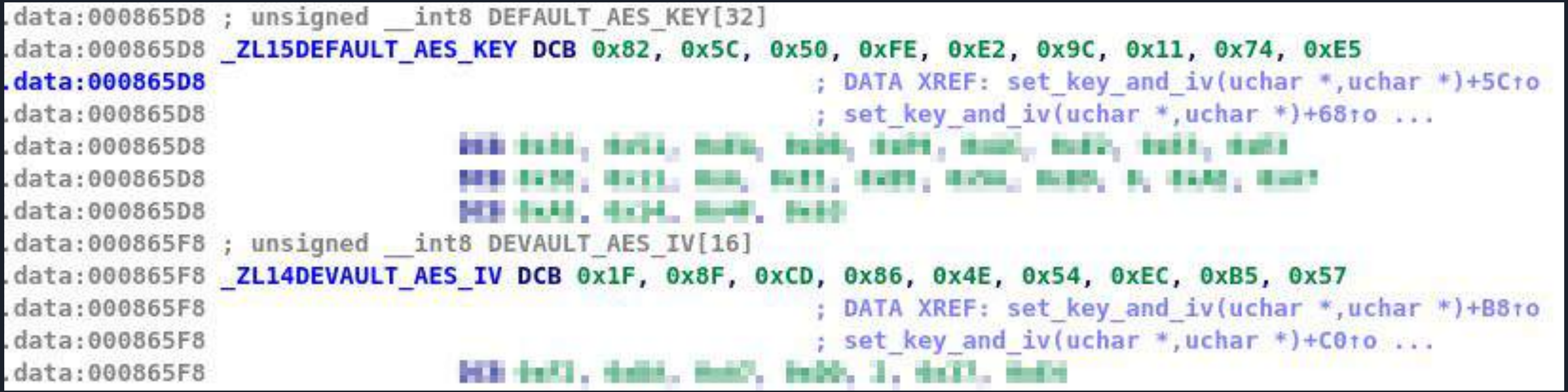
Binary Analysis: Local Controller Firmware

- ARM Device running GNU/Linux
  - Some sort of Debian Derivative
  - Firmware Supplied as deb packages
  - Numerous Binaries, Libraries and Scripts

Binary Analysis

- Shared objects provide (some) symbols by necessity
- Found correct binary & shared objects by “strings” and ldd
  - Need ldd for the armeabi
- If it’s stupid and it works, then it’s not stupid :)

A Wild Key Appears!



```
.data:000865D8 ; unsigned __int8 DEFAULT_AES_KEY[32]
.data:000865D8 _ZL15DEFAULT_AES_KEY DCB 0x82, 0x5C, 0x50, 0xFE, 0xE2, 0x9C, 0x11, 0x74, 0xE5
.data:000865D8                                          ; DATA XREF: set_key_and_iv(uchar *,uchar *)+5C↑o
.data:000865D8                                          ; set_key_and_iv(uchar *,uchar *)+68↑o ...
.data:000865D8                [redacted]
.data:000865D8                [redacted]
.data:000865D8                [redacted]
.data:000865F8 ; unsigned __int8 DEVAULT_AES_IV[16]
.data:000865F8 _ZL14DEVAULT_AES_IV DCB 0x1F, 0x8F, 0xCD, 0x86, 0x4E, 0x54, 0xEC, 0xB5, 0x57
.data:000865F8                                          ; DATA XREF: set_key_and_iv(uchar *,uchar *)+B8↑o
.data:000865F8                                          ; set_key_and_iv(uchar *,uchar *)+C0↑o ...
.data:000865F8                [redacted]
```

Default?

default; noun

> *a preselected option adopted by a computer program or other mechanism when no alternative is specified by the user or programmer*

Technically correct -- the programmer did not specify an alternative.

Will it decrypt?

- Decrypted values looked more structured
- Larger numbers of null bytes (typical of decrypted data)
- Lower entropy
- Without a MAC, no way to know for sure at this stage

Decoding the Plaintext

- Plaintext is useless without meaning
- Some custom binary protocol
- Binary Analysis lead to partial understanding

Decoding the Plaintext

- Badge Reads with Correct Badge Numbers
- Correlate Door Unlock Messages
- Door Status Messages
- Still Many Unknown ¯_(ツ)_/¯

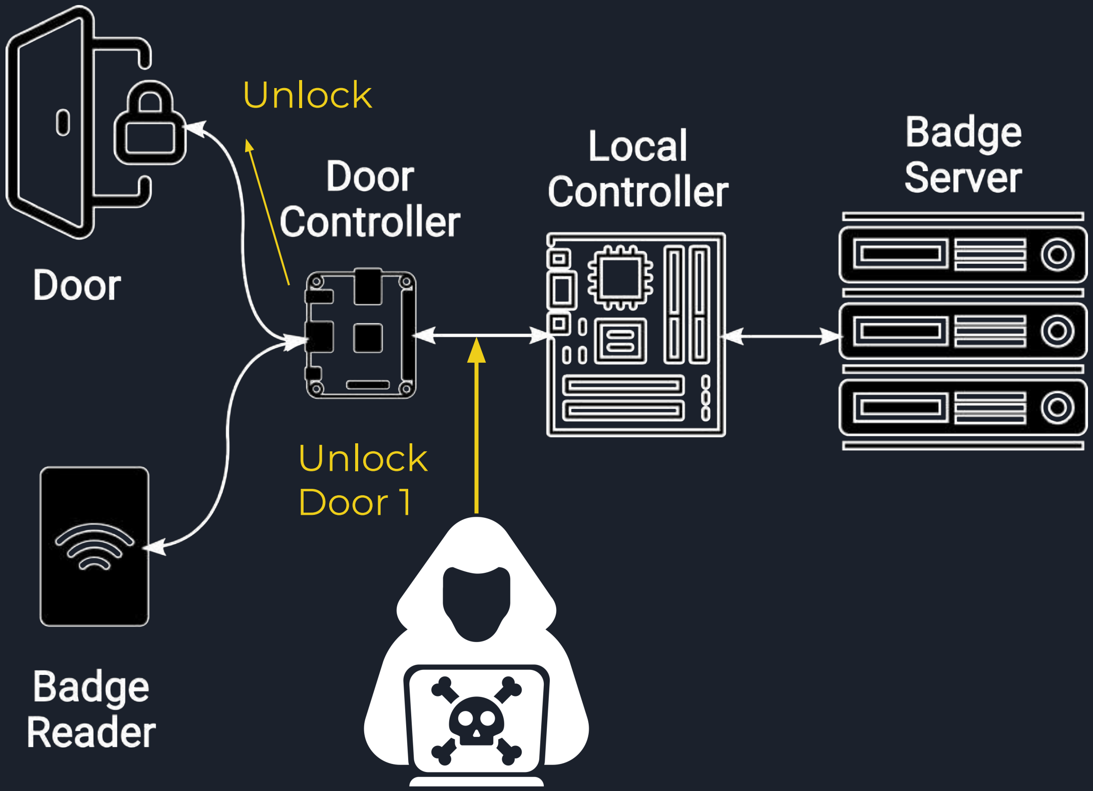
Making Working Exploit

- There’s some sequence numbers in the flow
- Door Controller connects to Local Controller
- Can’t initiate a new connection

Making Working Exploit

1. MITM Connection
2. Decrypt & get state (sequence numbers)
3. Start replying to each side
4. Send “door unlock” to door controller
5. Drop MITM
6. Profit!

Exploit in Action



So how do you fix this?

Why is this even a thing?

- This is easy to implement but still encrypted
- Doing transport security correctly is hard.

Constraints on IoT Devices

- Non-traditional interfaces
- May not have hostnames
  - How to verify certificates, even if present?
- Low power CPU/small flash footprint
- Network should not reach the Internet

Ways to Improve

- Keys should not be common across installations
- Devices must only communicate with trusted partners
- Individual messages should have confidentiality and integrity
- Do not roll your own crypto!

Hypothetical One

- Use TLS
- Vendor ships each device with a certificate
- Trusts other devices signed by vendor

Hypothetical One

- Use TLS
- Vendor ships each device with a certificate
- Trusts other devices signed by vendor
- Attacker buys their own device?
- Cert/key stolen from one device?

Hypothetical Two

- Use TLS
- Customer configures each device with a CA certificate

Hypothetical Two

- Use TLS
- Customer configures each device with a key & CA certificate
- Infeasible at scale?

Hypothetical Three

- Uses TLS
- Devices ship with hardware attestation key
- Device signs certificate request on first use, sends upstream
- Central CA signs
- CA Setup is Transparent

Hypothetical Three

- Uses TLS
- Devices ship with hardware attestation key
- Device signs certificate request on first use, sends upstream
- Central CA signs
- CA Setup is Transparent
- Requires Trustworthy Network on First Use

Conclusion

- Software Security Matters for Physical Security Systems
- Industry could be doing much more
- Customers have to ask for more

Questions?

Twitter: @Matir
Blog: https://systemoverlord.com
Slides: https://1337.fyi/doors