BREAKING THE PAYMENT POINTS OF INTERACTION (POI)

Nir Valtman
@ValtmaNir

&

Patrick Watson
@PatrickTheDev

THE CHALLENGE

Introduce 2 chatterboxes in 1 slide!

```
<?xml version="1.0" encoding="WTF-8"?>
<speakers>
  <identity username="Nir Valtman" twitter="@ValtmaNir">
    <org title="Head">NCR Application Security</org>
    <motto>If security is expensive, try to ignore it!</motto>
    <presented>Black Hat, DEF CON, OWASP etc.</presented>
    <opensource>SAPIA, Cloudefigo, Secure TDD, AntiDef</opensource>
  </identity>
  <identity username="Patrick Watson" twitter="@PatrickTheDev">
    <org title="Architect">NCR Application Security</org>
    <motto>If I shouldn't tell you how my system works, it isn't secure</motto>
    <presented firstTimeSpeaker="F***K YEAAAH" />
    <opensource>Who cares? I develop better than Nir!</opensource>
  </identity>
 </speakers>
```

```
    <IPAddress>1.1.1.1</IPAddress>
    <Port>8834</Port>
    <Username>aaa</Username>
    <Password>bbbbb</Password>
    <ScanPolicyId>1</ScanPolicyId>
  </Nessus>
  <Chef>
    <KeyFilePath>validation.pem</KeyFilePath>
    <ValidationClientName>cloudsecdemo-validator</ValidationClientName>
    <ServerURL>https://api.opscode.com/organizations/cloudsecdemo</ServerURL>
  </Chef>
 <Instances>
    <BucketPolicyPath>CloudServices/IaaS/RuntimeConfig/BucketPolicy.config</BucketPolicyPath>
    <IAMBasicPolicyPath>CloudServicesdfdsfdsfdsfdsfdsfds/Admin/RuntimeConfig/IAMBasicPolicy.config</IAMBasicPolicyPath>
    <IAMStrictPolicyPath>CloudServices/IaaS/sdfdsfsdRuntimeConfig/IAMStrictPolicy.config</IAMStrictPolicyPath>
    <CloudInitScriptPath>CloudServices/Asdfsdfdmin/RuntimeConfig/CloudConfig.config</CloudInitScriptPath>
    <NamingPrefix>cloudsec</NamingPrefix>
    <ProductionSecurityGroupId>sg-7befd21a</ProductionSecurityGroupId>
    <ProductionSubnetId>subnet-59e2f622</ProductionSubnetId>
    <RemediationSecurityGroupId>sg-e63235f3</RemediationSecurityGroupId>
    <EC2KeyName>mykey</EC2KeyName>
    <CanonicalUserId>e55740c643faeb405a27ab195a29e5f20f32wdsfsdfdfdsfdsfsdfsdfsdfsdf6082fc5abdfb8</CanonicalUserId>
  </Instances>
</data>
```

THE RESEARCH

Nir: ”Can’t we just change this damn flow in the POI?”
Patrick: “LOL, see what happens! We’re going to Vegas this year!”

The result – high fived our product to research anarchy!

Various Business Flows

What would you say if I told you that the Retail industry is the most secure?

Retail

Hospitality

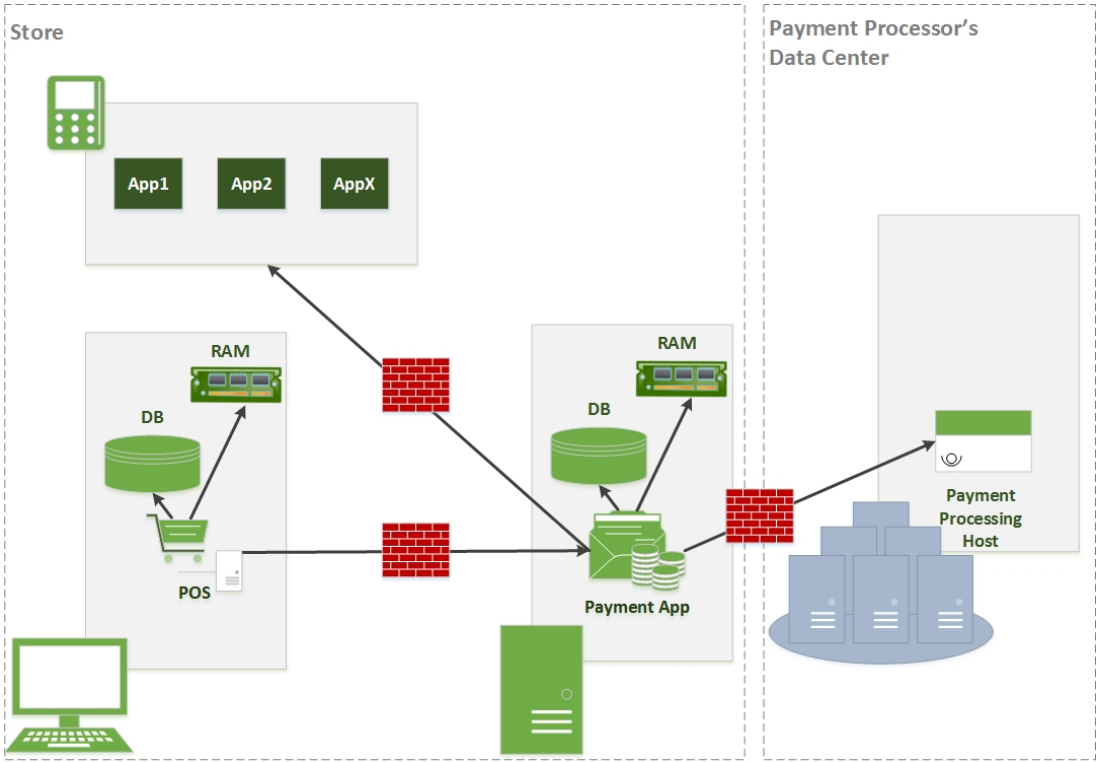
Petroleum & Convenience


LOL, good one!

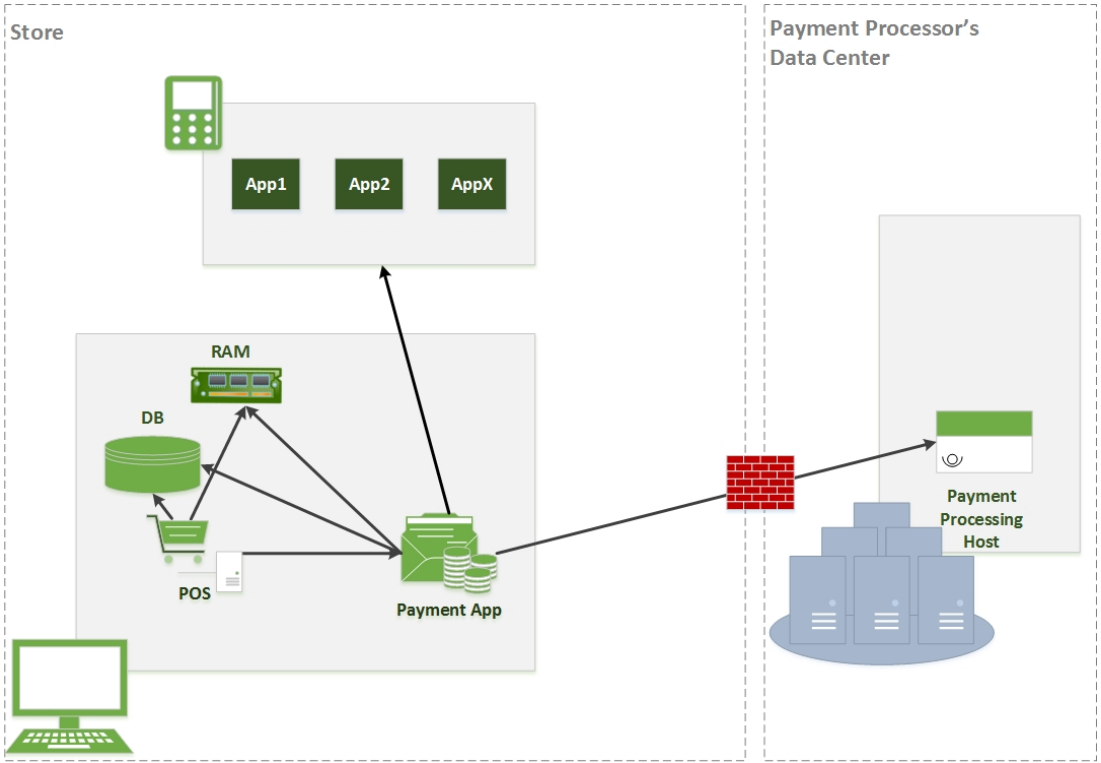
INTRO TO PAYMENTS & POI

Payments POI, meet Nir & Patrick. We hack stuff for fun!

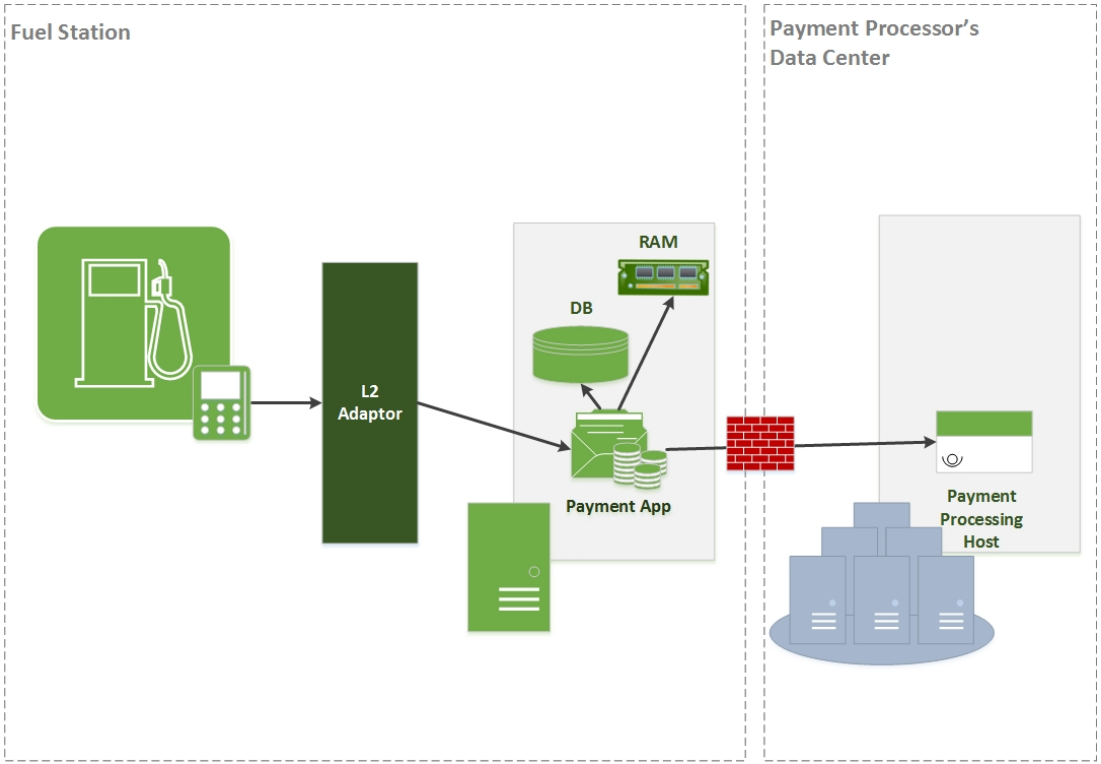
Segregated store architecture



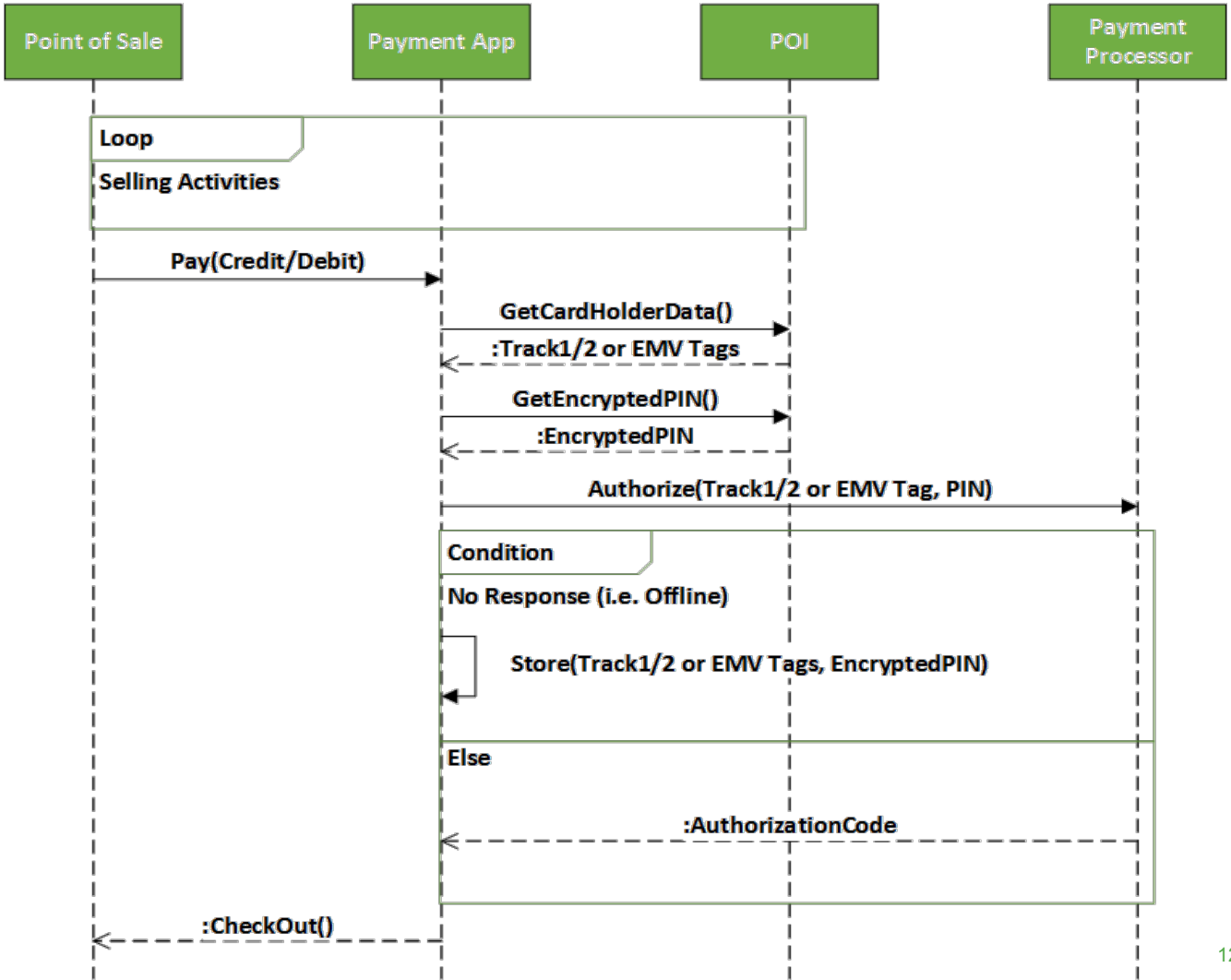
All-in-one store architecture



Fuel station architecture



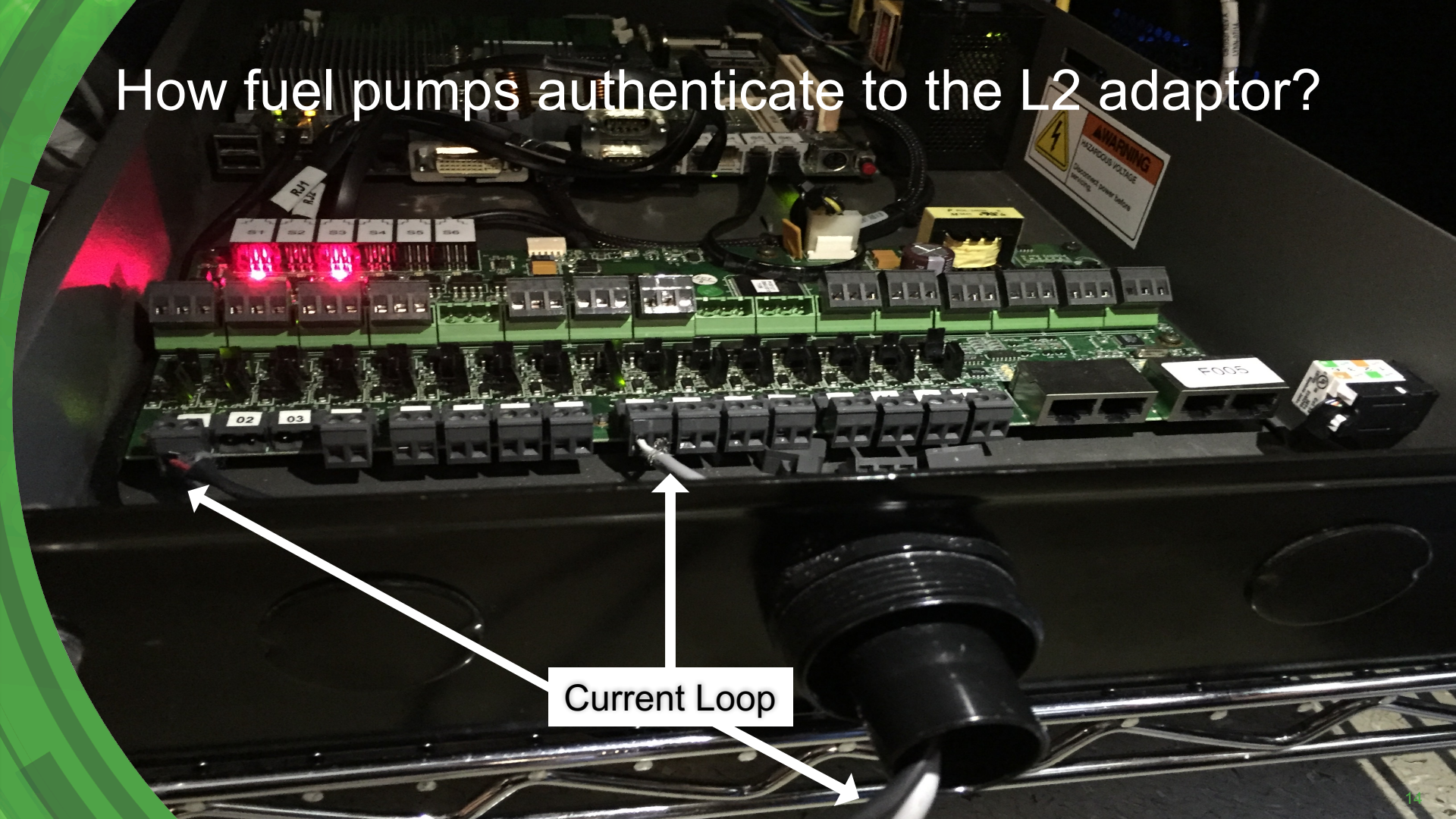
Typical Payment Flow



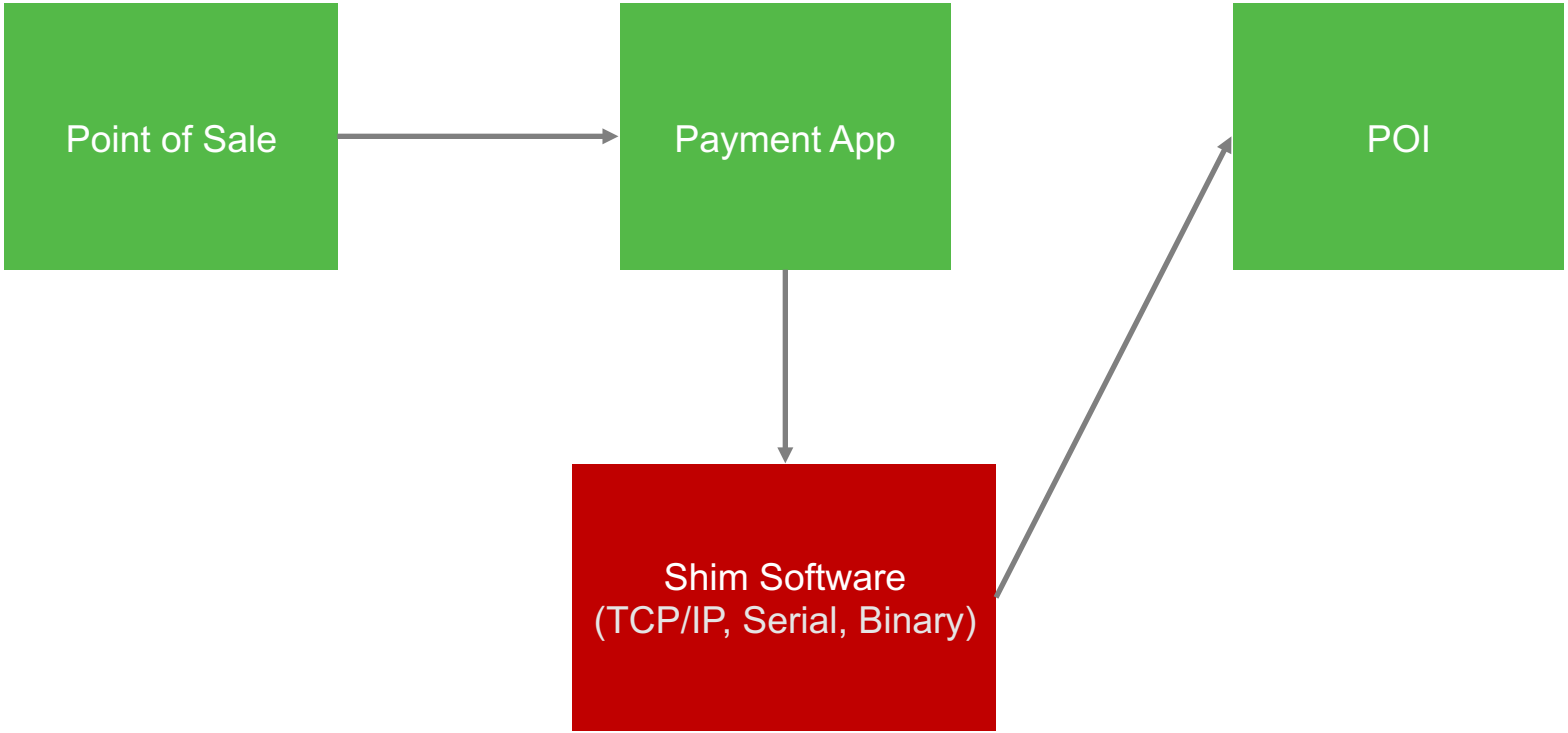
WEAK AUTHENTICATION TO POI

Knock-knock. Who’s there? It’s YOU!

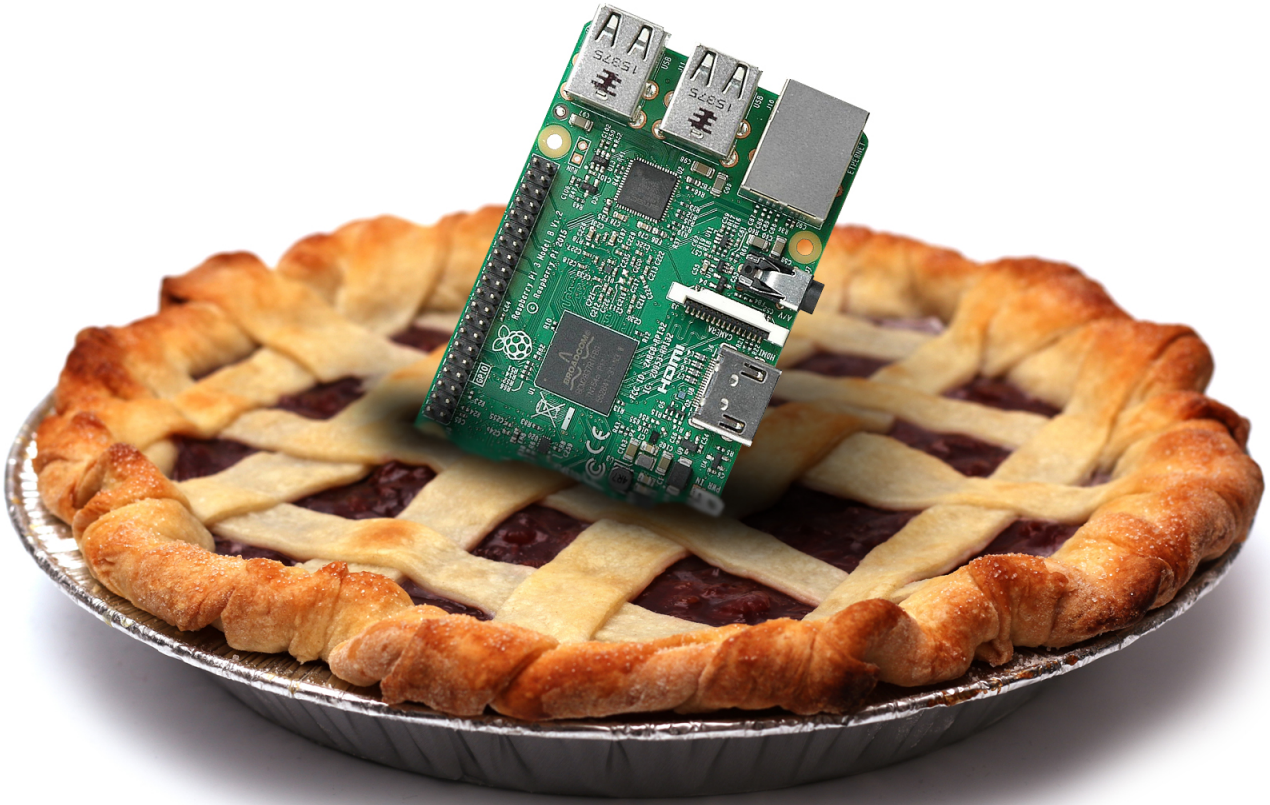
How fuel pumps authenticate to the L2 adaptor?



How shim software works?



A shim software demo using a Pi



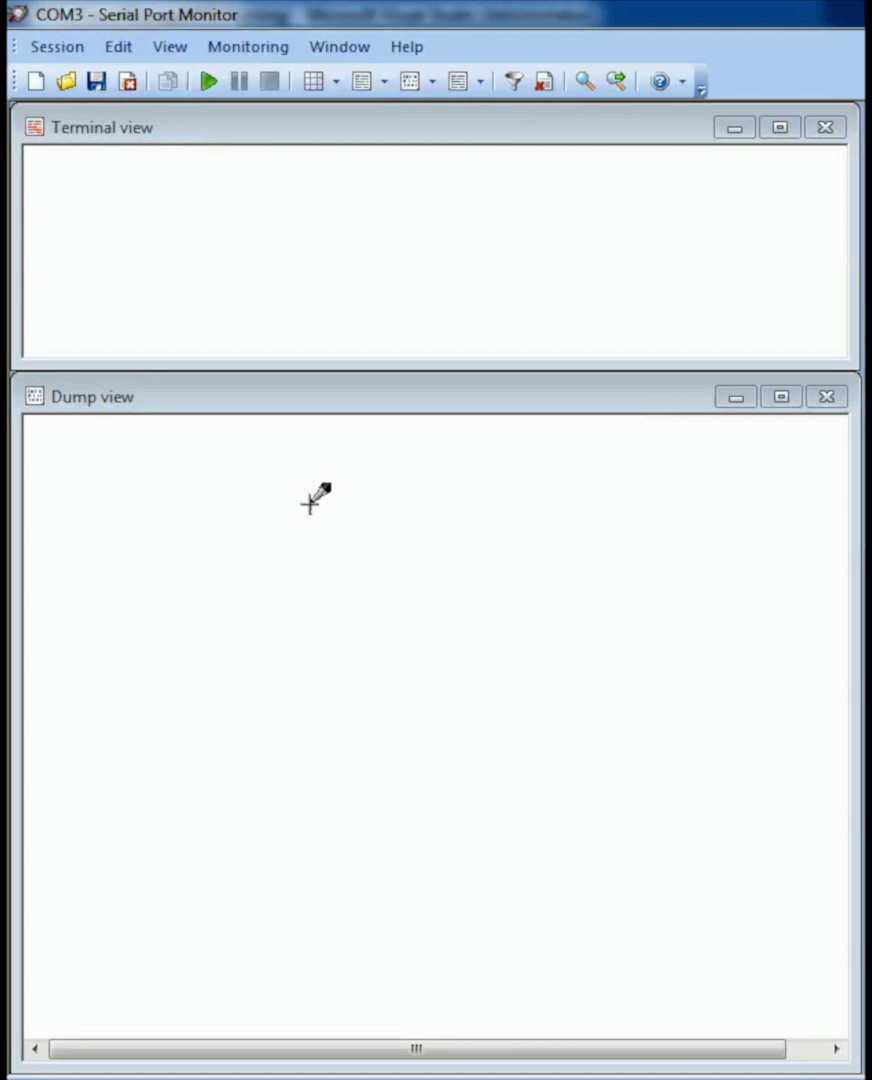
Encryption strength limitations

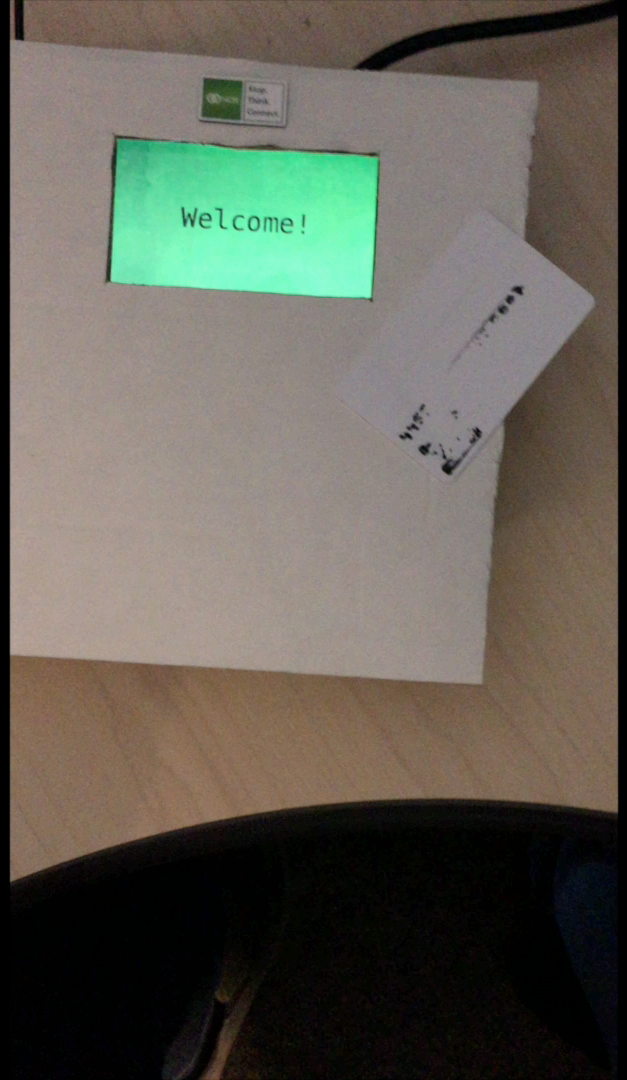
Sometimes NOT even SSLv3!

Business justification: “we have old hardware (POS or POI)”

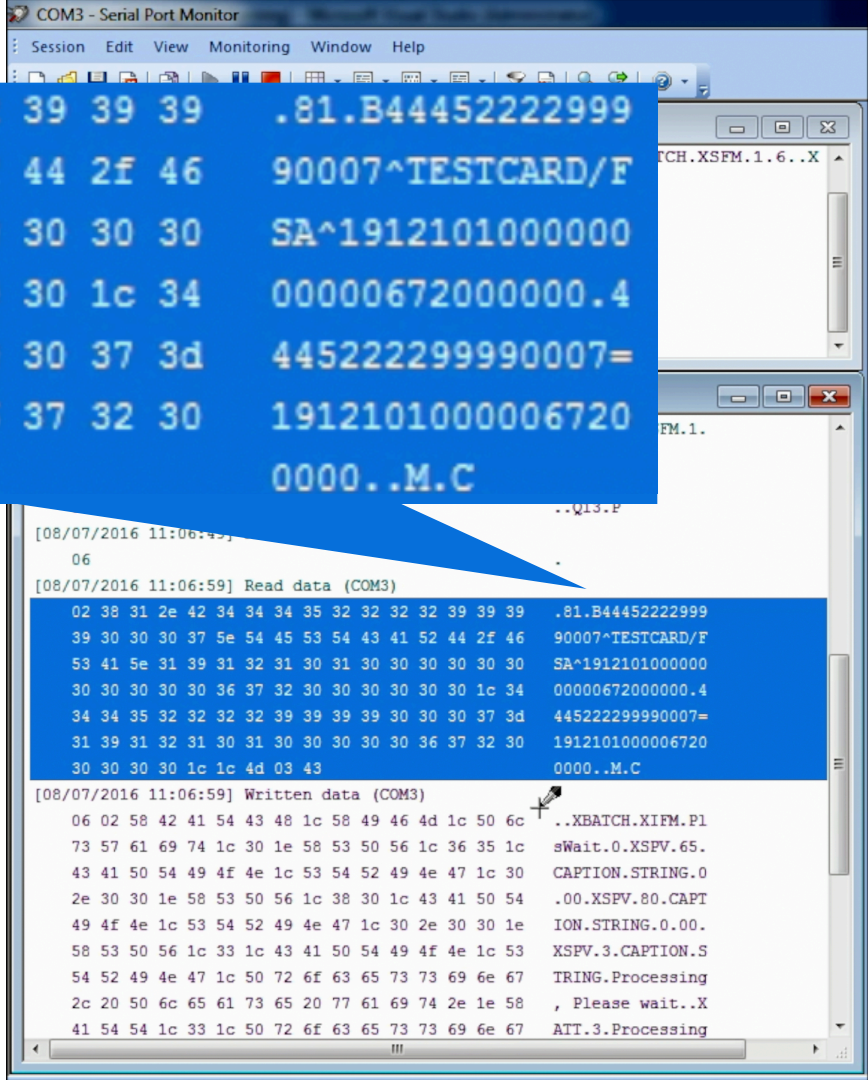
Hackers’ justification: “it wasn’t encrypted, I had to steal it”

Showing off with a demo


COM3 - Serial Port Monitor
Session Edit View Monitoring Window Help
Terminal view
Dump view


Welcome!

```
02 38 31 2e 42 34 34 34 35 32 32 32 32 39 39 39   .81.B44452222999
39 30 30 30 37 5e 54 45 53 54 43 41 52 44 2f 46   90007^TESTCARD/F
53 41 5e 31 39 31 32 31 30 31 30 30 30 30 30 30   SA^1912101000000
30 30 30 30 30 36 37 32 30 30 30 30 30 30 1c 34   00000672000000.4
34 34 35 32 32 32 32 39 39 39 39 30 30 30 37 3d   445222299990007=
31 39 31 32 31 30 31 30 30 30 30 30 36 37 32 30   1912101000006720
30 30 30 30 1c 1c 4d 03 43                        0000..M.C
```



Unclear API documentation

FORGET ABOUT SWIPING! WHAT ABOUT EMV?

It’s obviously secure as hell, isn’t it?

EMV DOES

Prevent duplication of card

Chip & PIN prevents using a stolen card

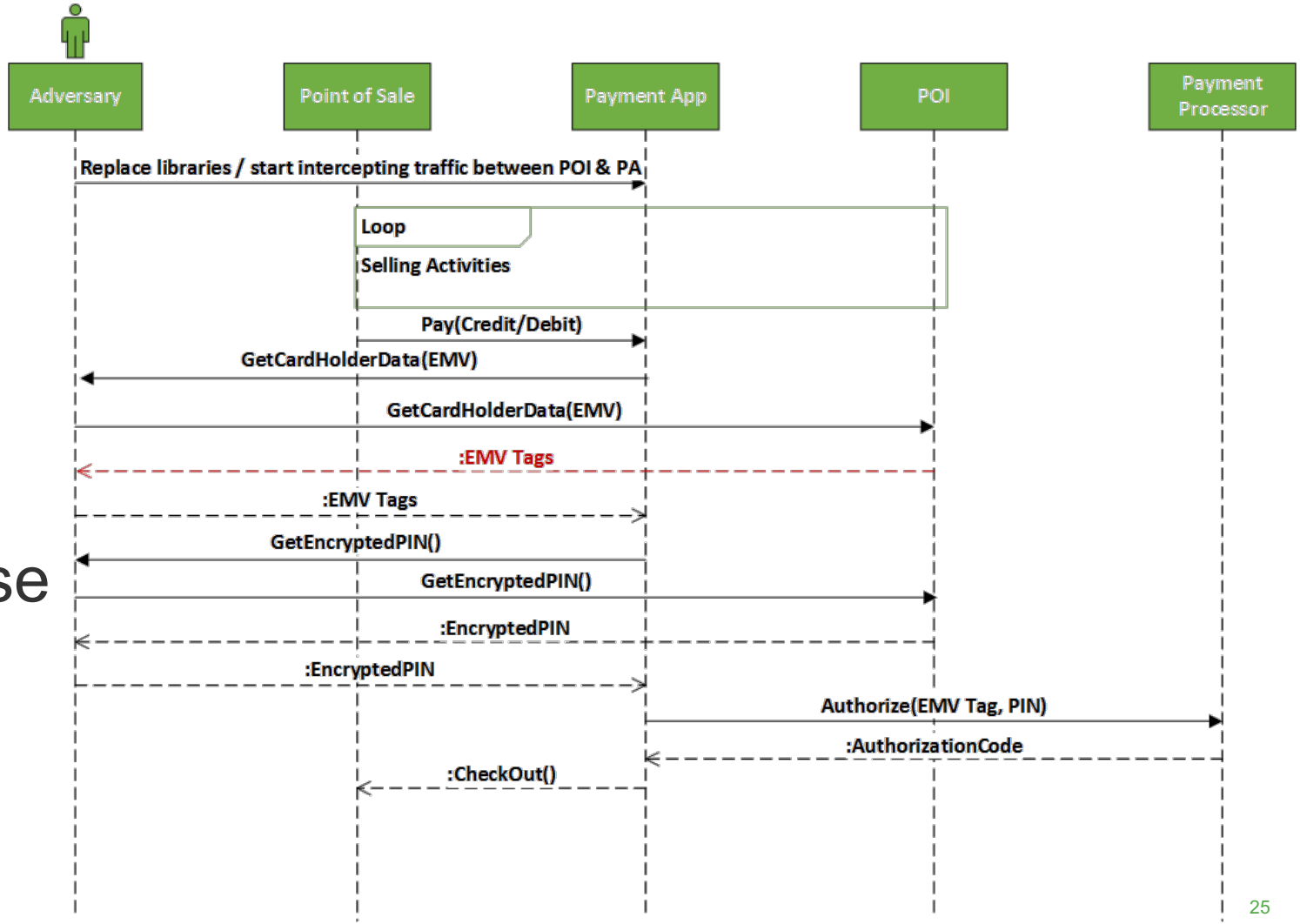
EMV DOES NOT

Prevent using card number elsewhere

Prevent using modified Track 2 in offline mode

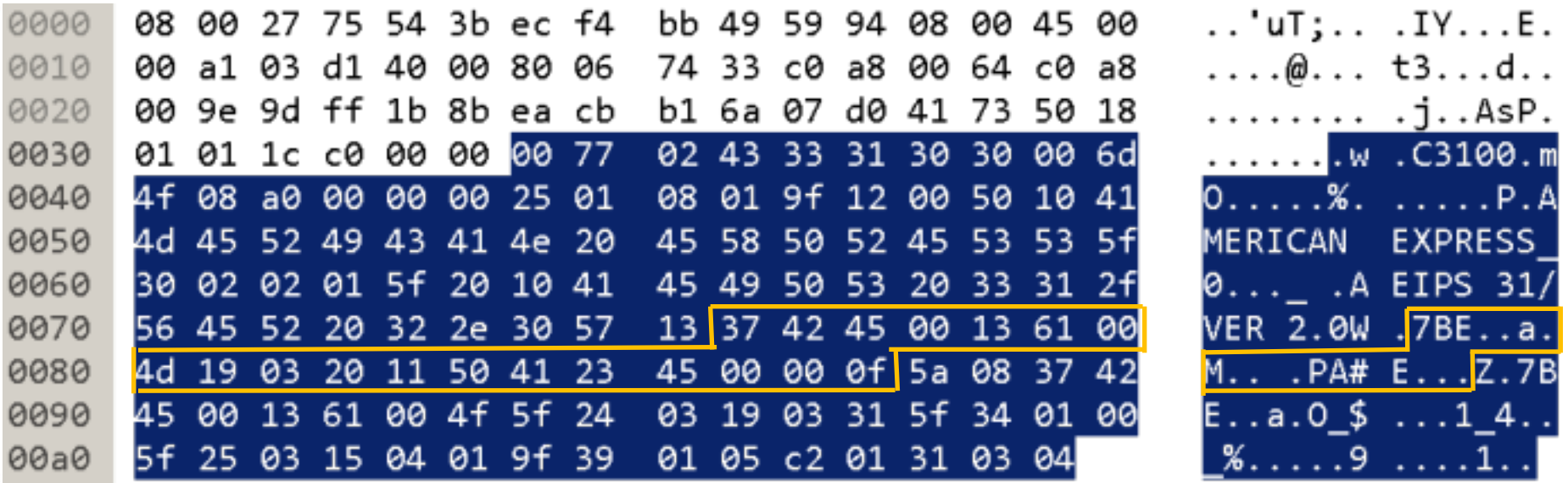
Prevent unmodified Track 2 w/ EMV after tricking into technical fallback to swipe

Passive MITM Track 2 Compromise



I’m all ears! Where’s the demo?

The Basics of EMV



;3742450013610 04=19032011504123450000 0?

On Sale Now:
Track Data!

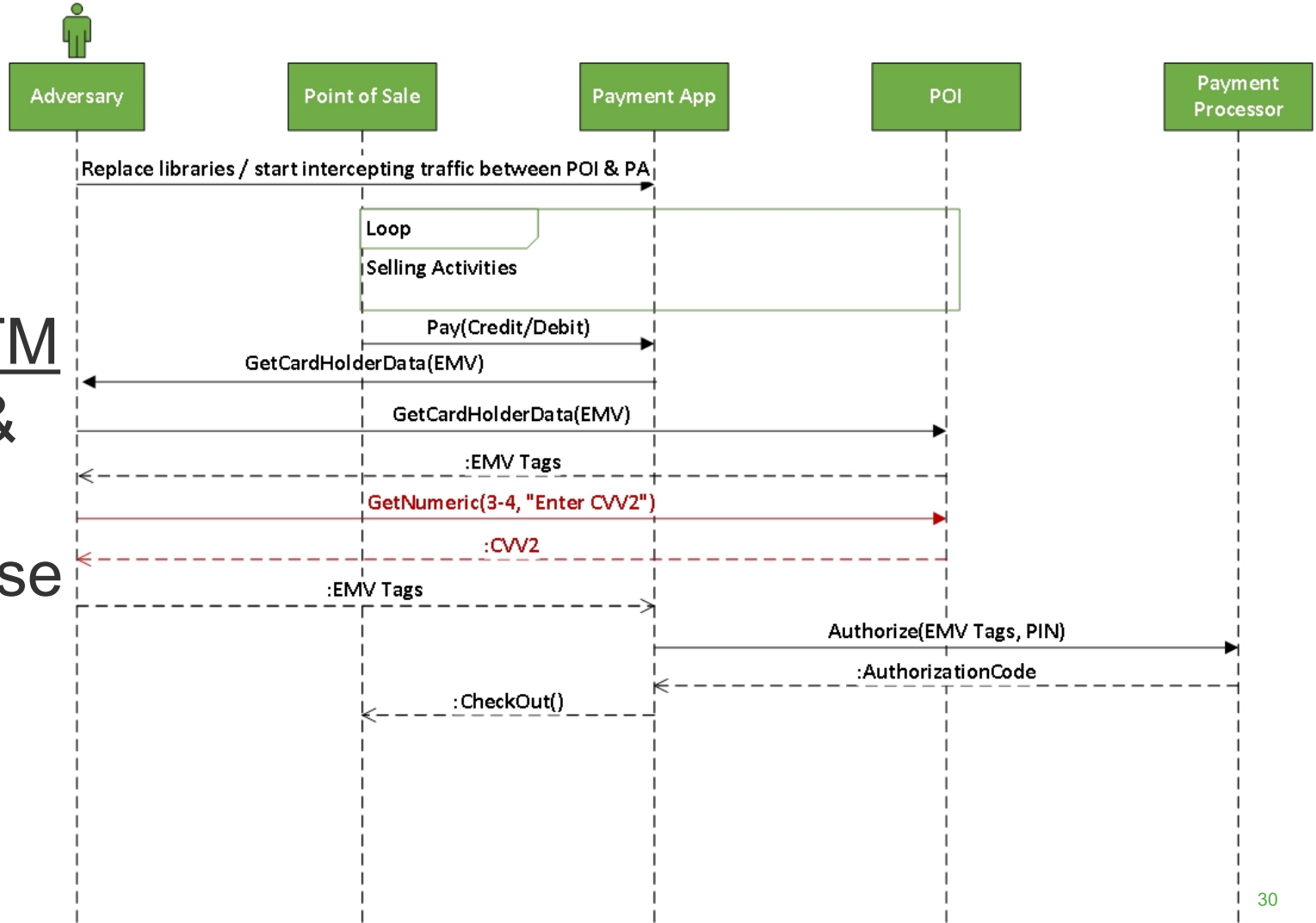
Which info you need to make a purchase online?

- ☑ Personal Account Number (aka PAN)
- ☑ Expiration Date
- ☑ Cardholder's Name
- ☐ CVV2

BYPASSING THE EMV FLOWS

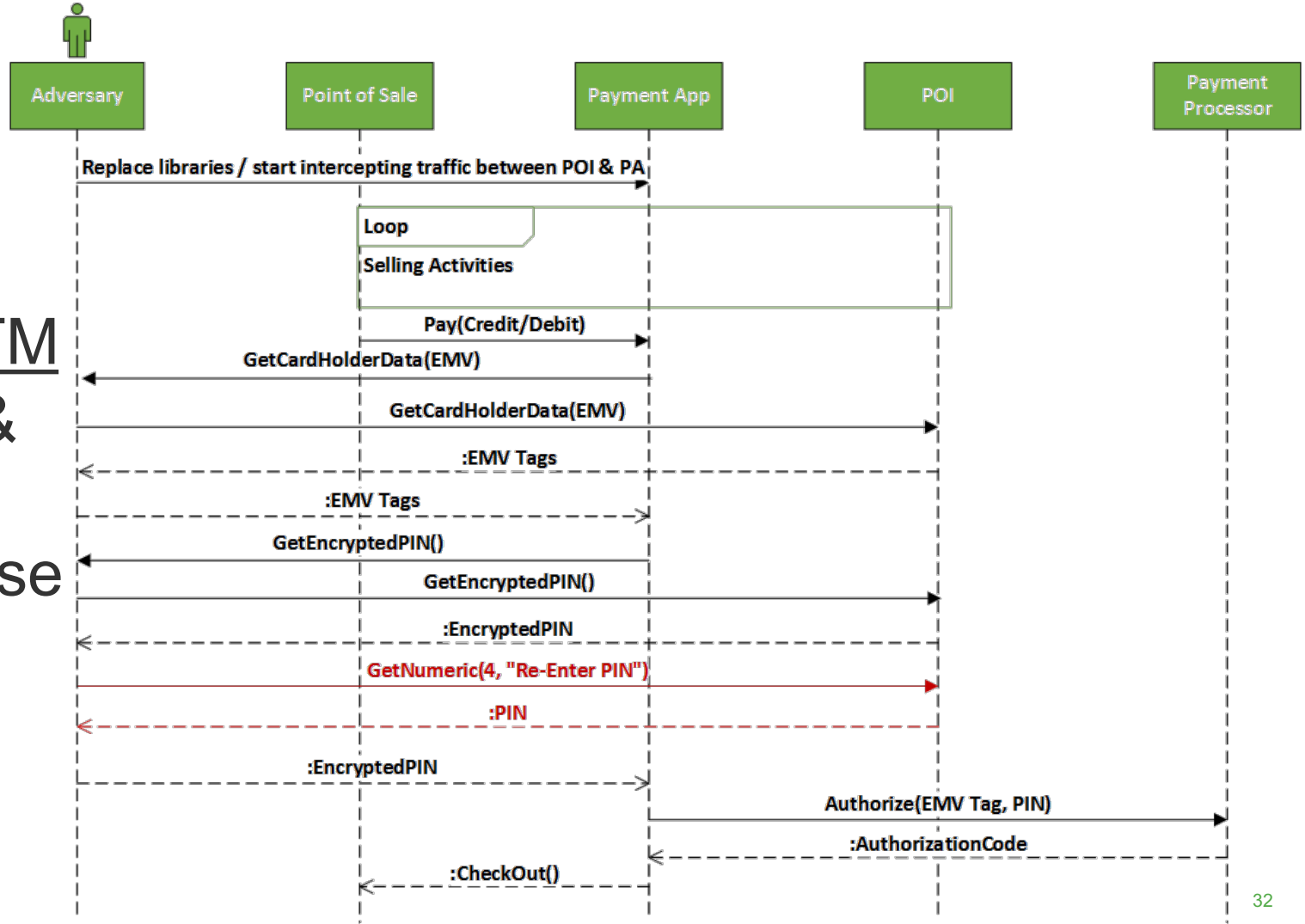
It’s obviously secure as hell, isn’t it?

Active MITM Track 2 & CVV2 Compromise




PLEASE, DO NOT USE
DEMO ONLY
THANK YOU

Active MITM Track 2 & PIN Compromise

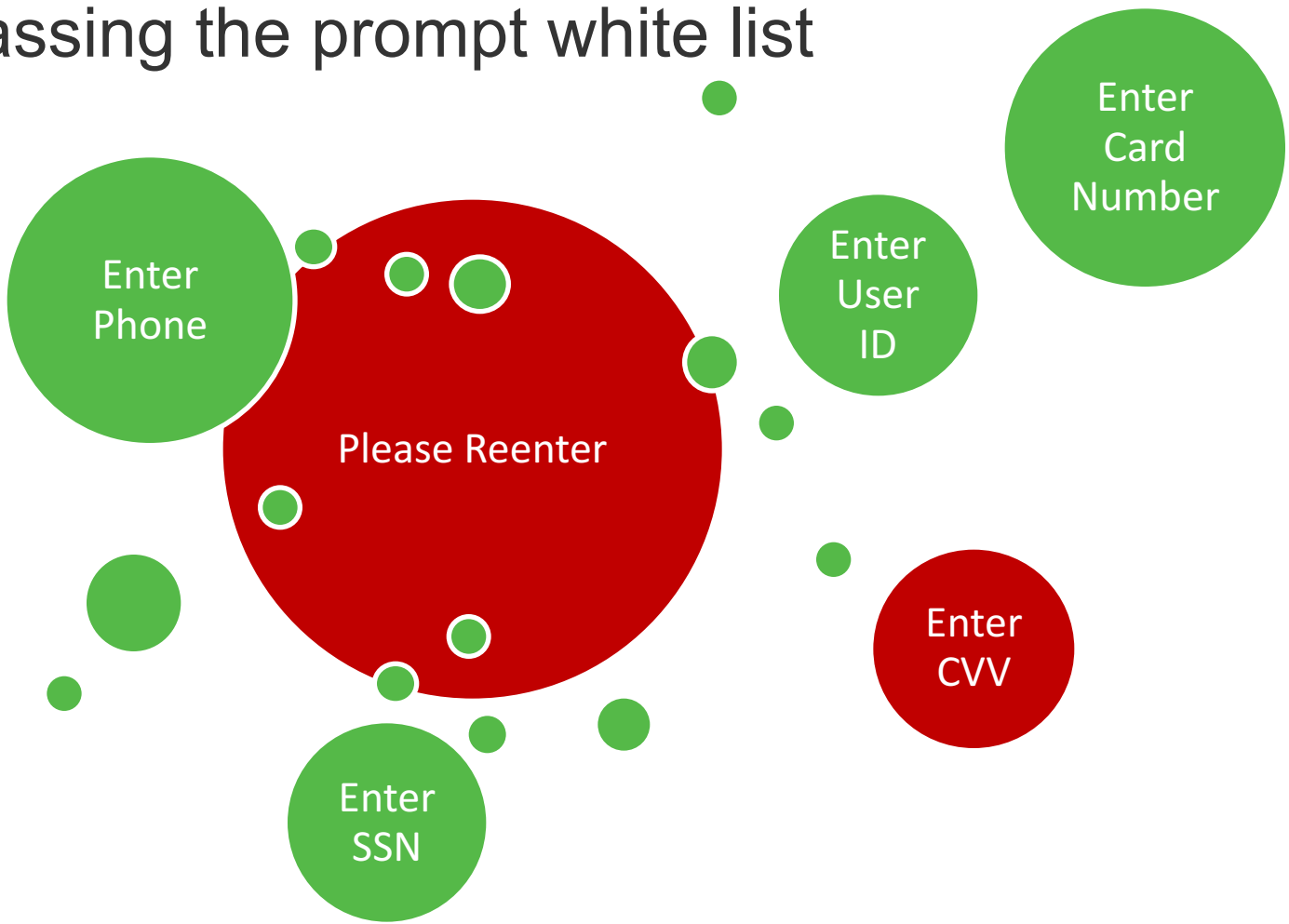


DEMO: Let’s steal few PINs

WHAT IF CVV2/PAN ARE RESTRICTED BY A WHITE LIST?

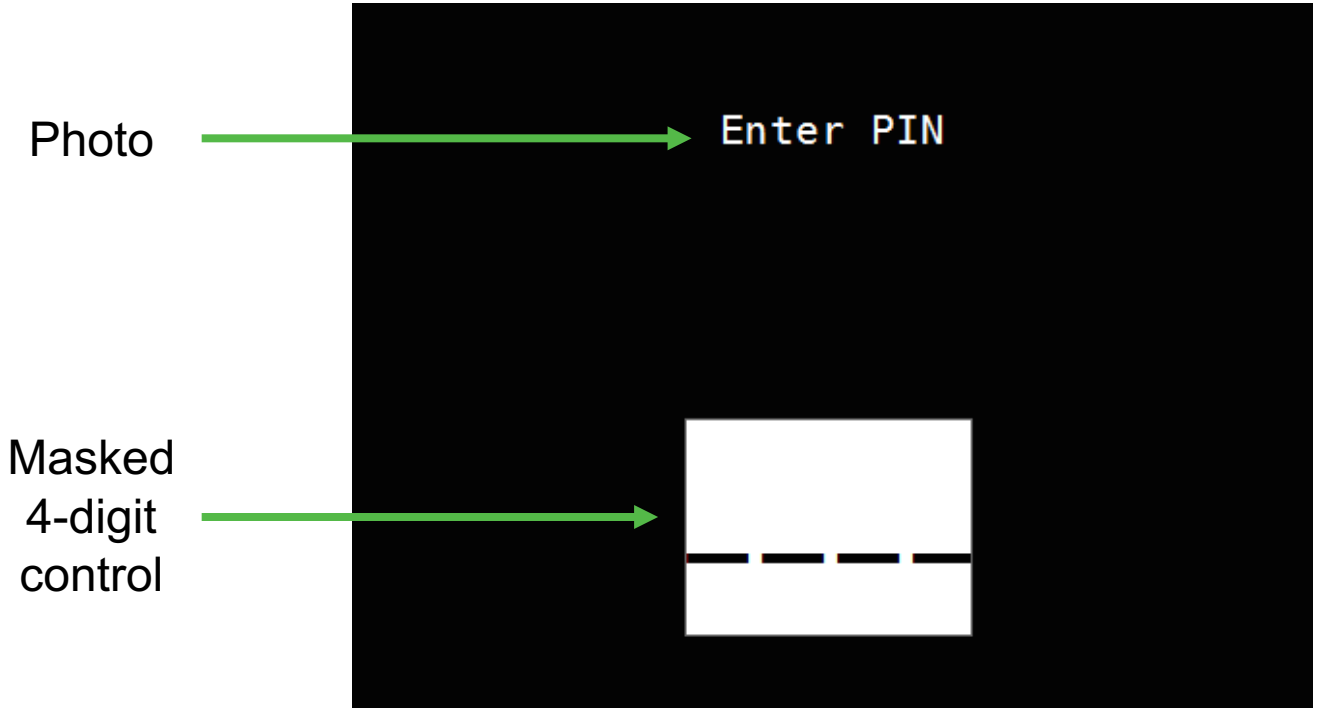
We leverage other white listed functionality!

Bypassing the prompt white list



Form/Screen Injection

Remember that there is NO authentication to POI?



Signing MAY be required by several vendors though


It’s an illusion

ANY OTHER ATTACK VECTORS?

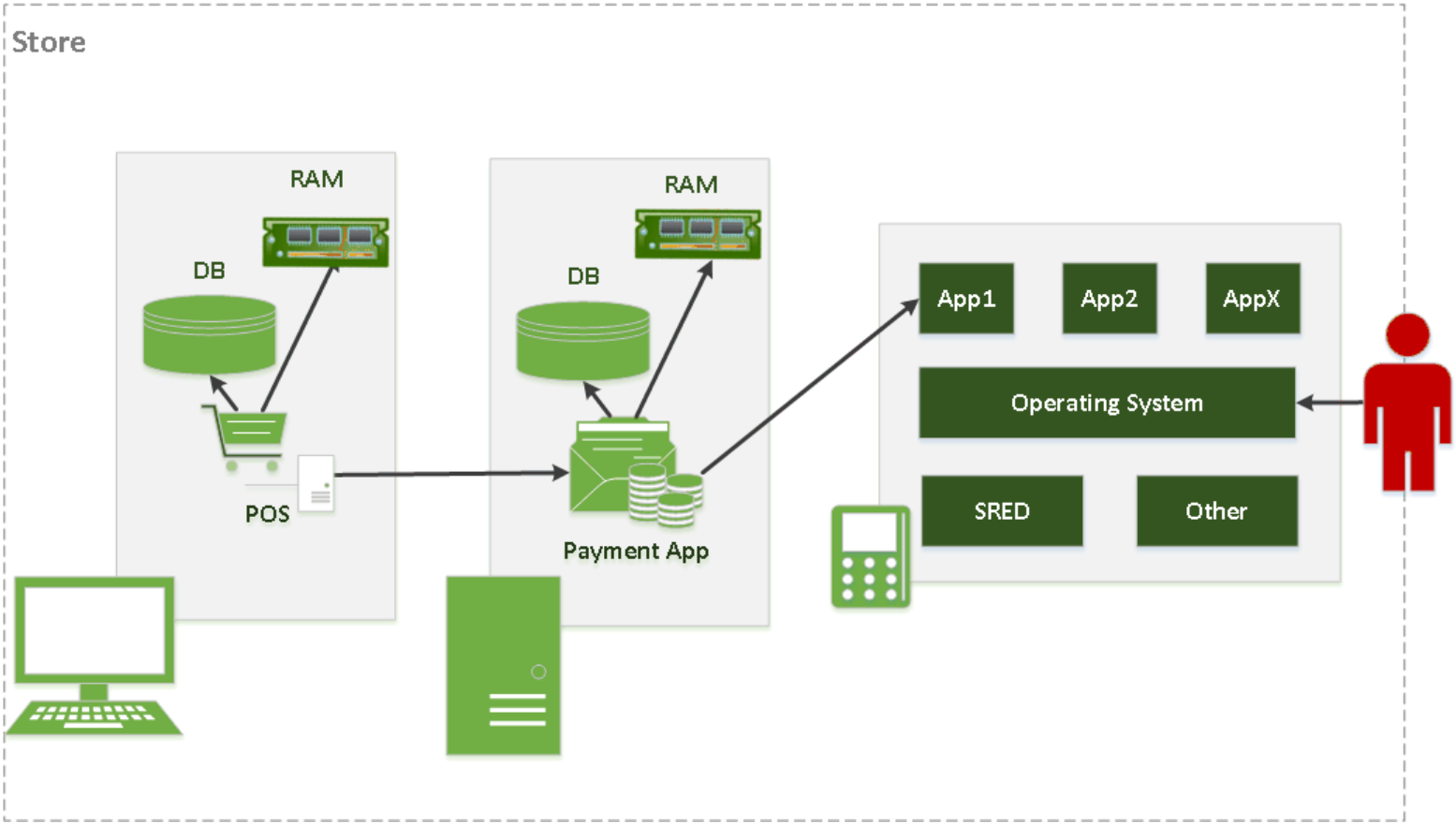
Let me think about it for a day or two … YES!

Skimmers


ENTER
9 WXY
6 MNO
3 DEF
Genuine MSR
Skimmer MSR


iSC250
1 .QZ
2 ABC
3 DEF
Cancel
4 GHI
5 JKL
6 MNO
Clear
7 PRS
8 TUV
9 WXY
Enter
+
0
-

Exploit POI's Operating System



MITIGATIONS

Enough complaining about problems!

Point-to-Point Encryption (P2PE)

Use ONLY hardware-based message encryption

Preferably SRED enabled

Point-to-Point Encryption (P2PE)

Leverage strong crypto algorithms

AES/3DES DUKPT

RSA

Point-to-Point Encryption (P2PE)

Prevent remote firmware downgrades to software-based encryption

Firmware must be signed anyway

Point-to-Point Encryption (P2PE)

Allow ONLY signed white list updates by vendor

Credit card bin ranges

Trusted Certificate Authorities

Point-to-Point Encryption (P2PE)

Encrypt offline transaction data on payment applications

What if the POI or Payment Application doesn't support P2PE?

What can merchants/vendors do?

WHAT CAN A CONSUMER DO?

Except paying with cash…

AWARENESS AWARENESS AWARENESS

Never re-enter PIN

Be cautious of unusual prompts

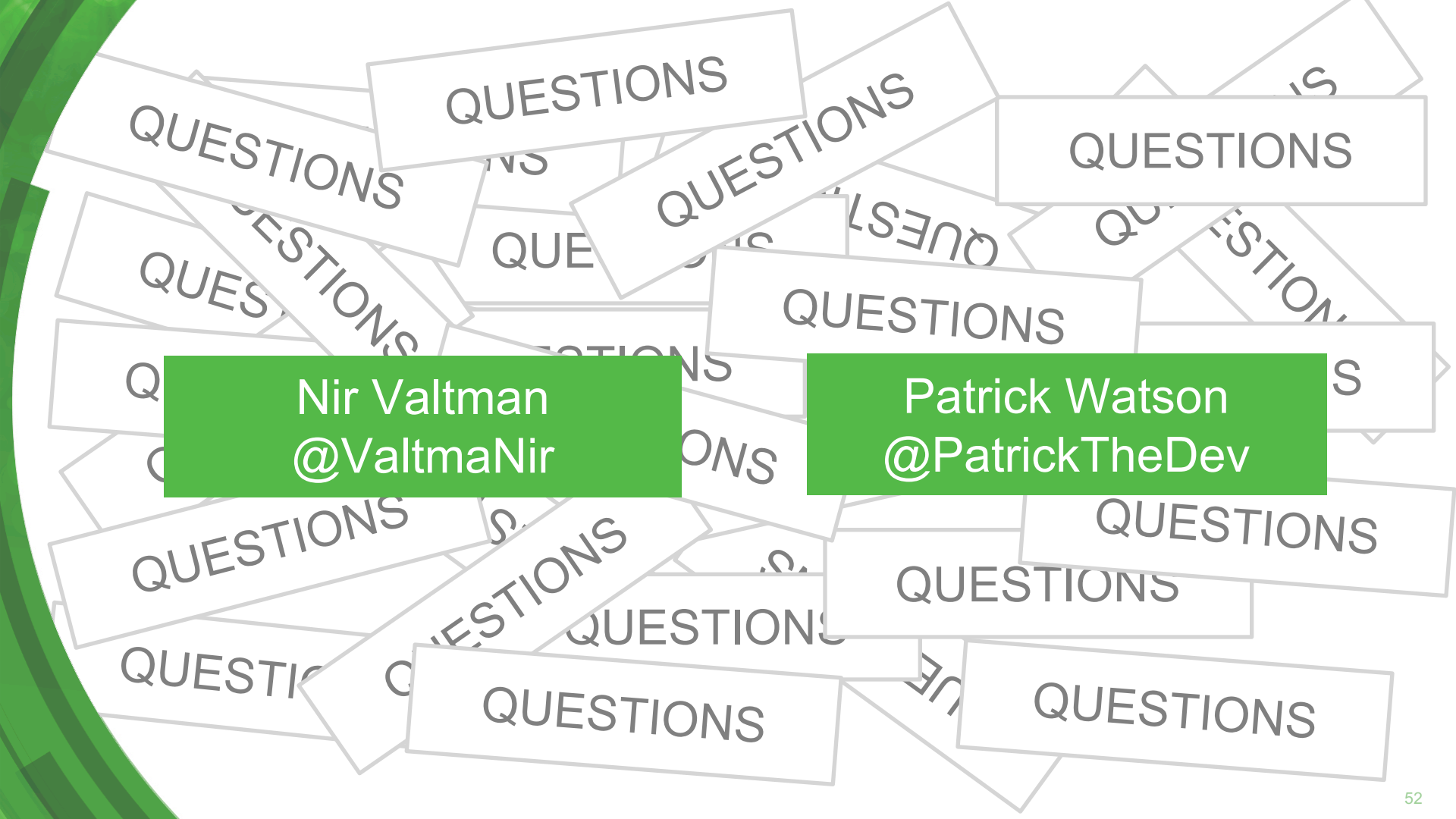
App-based payment systems MAY be preferred

No PAN transmission - tokenization

Unique Track2 per transaction

Summary

- Relatively easy to exploit common POI deployments.
  - The lack of authentication poses the POI to major risks.
  - EMV can be bypassed by easily tricking the POI.
- Points of interaction can be secure.
  - P2PE solution
  - Added authentication mechanisms


QUESTIONS
Nir Valtman
@ValtmaNir
Patrick Watson
@PatrickTheDev