

# FOR THE LOVE OF MONEY

Finding and exploiting vulnerabilities in mobile point of sales  
systems



LEIGH-ANNE GALLOWAY & TIM YUNUSOV  
POSITIVE TECHNOLOGIES

# MPOS GROWTH



2010

Single vendor



Tweets    Tweets & replies    Media    Likes

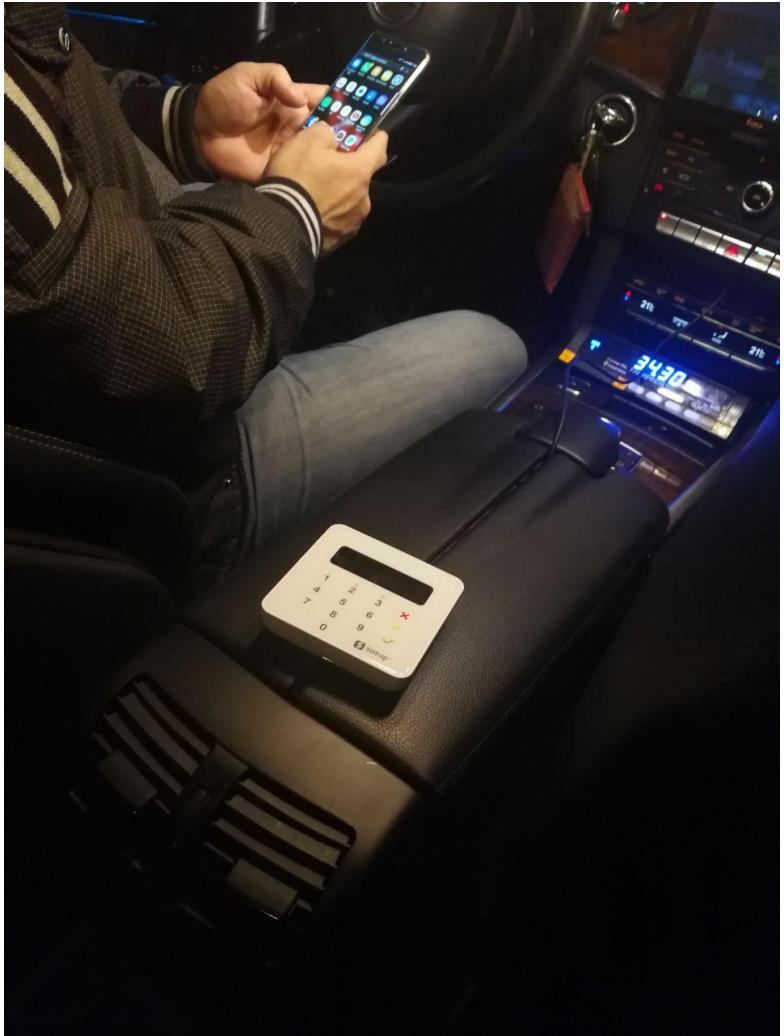
Pinned Tweet  
**SumUp Engineering** @S... · 31 Oct 17  
At [@SumUp](#) we ship more than 2,000 devices per day

A screenshot of a Twitter profile page. The 'Tweets' tab is selected. A pinned tweet from 'SumUp Engineering' (@SumUp) is displayed, stating: 'At @SumUp we ship more than 2,000 devices per day'. The tweet includes a money bag emoji.

2018

Four leading vendors  
shipping thousands of units per day

## Motivations

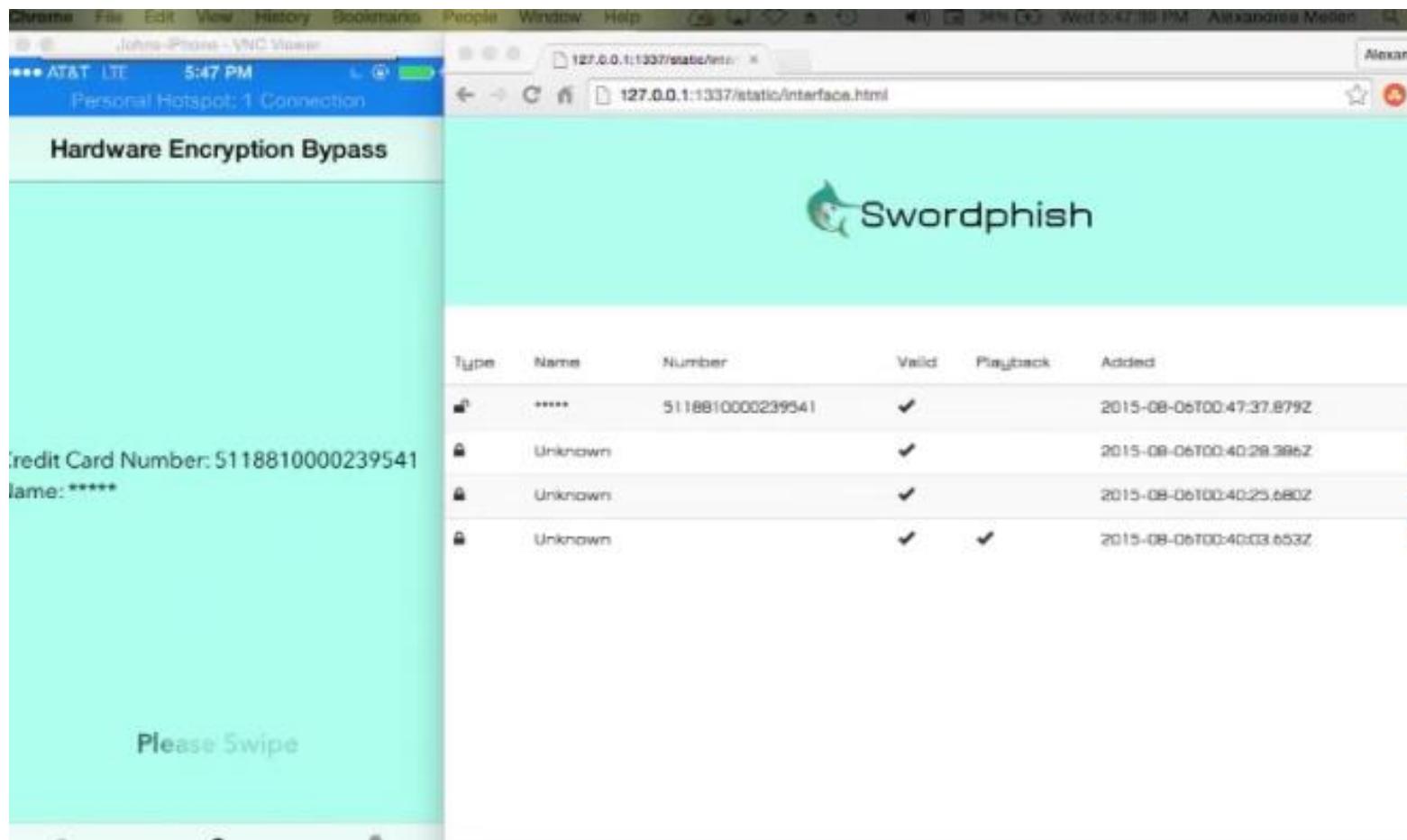


## Related Work



MWR Labs “Mission mPOSSible” 2014

## Related Work

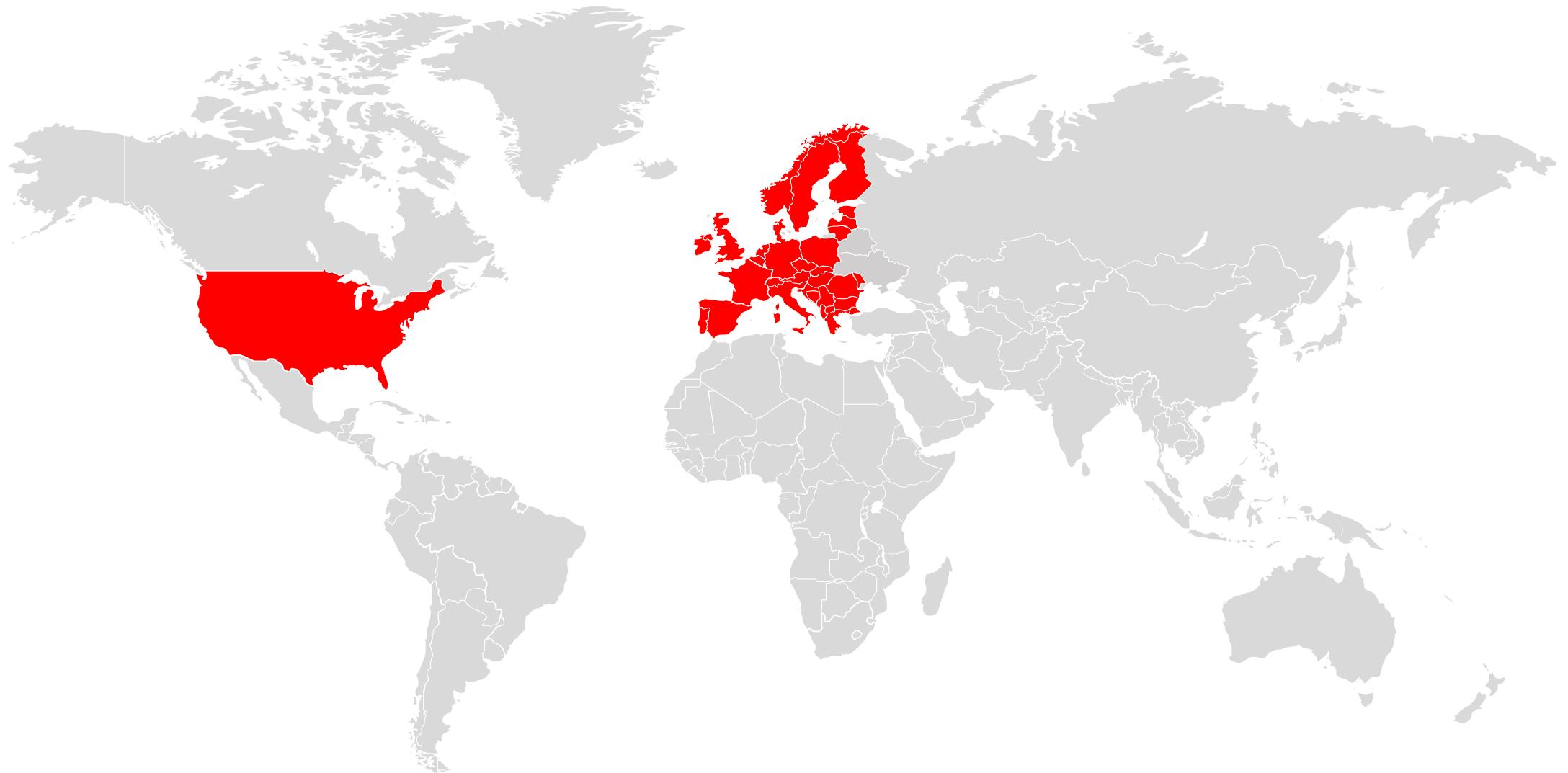


Mellen, Moore and Losev “Mobile Point of Scam: Attacking the Square Reader” (2015)

# Research Scope



## Research Scope



# Research Scope



PAYPAL



SQUARE

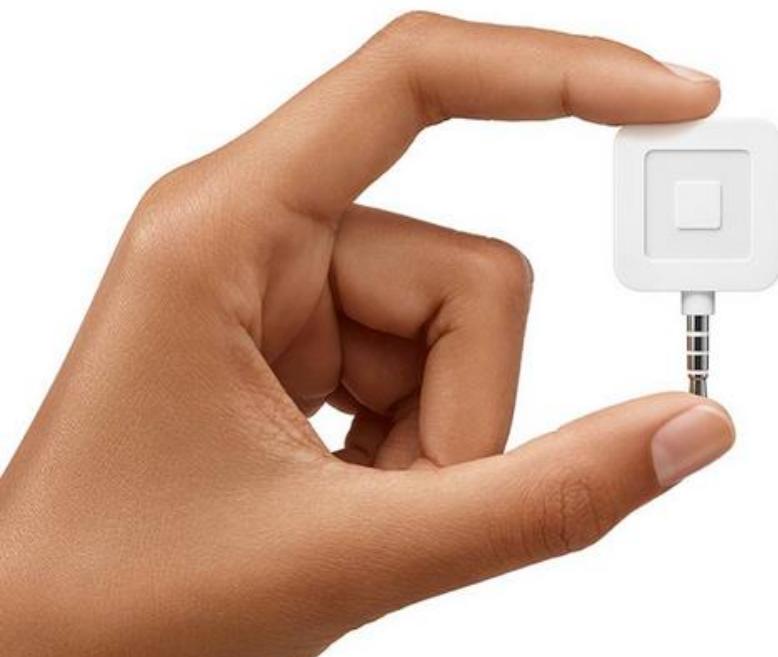


IZETTLE



SUMUP

“How much security can really be embedded  
in a device that is free?”



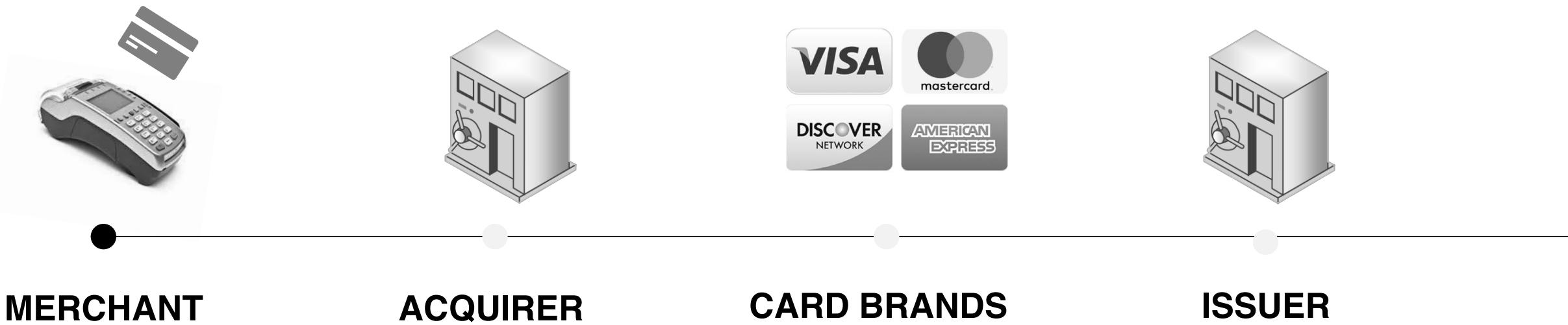
Accept credit cards anywhere. Sign up and we'll send you a free reader.

Get a free magstripe reader to swipe credit cards anywhere. Take chip cards and NFC payments with Square Reader for contactless and chip. Slip an iPad into Square Stand to make a countertop point of sale. Or sell with Square Register, the first fully integrated point-of-sale system.

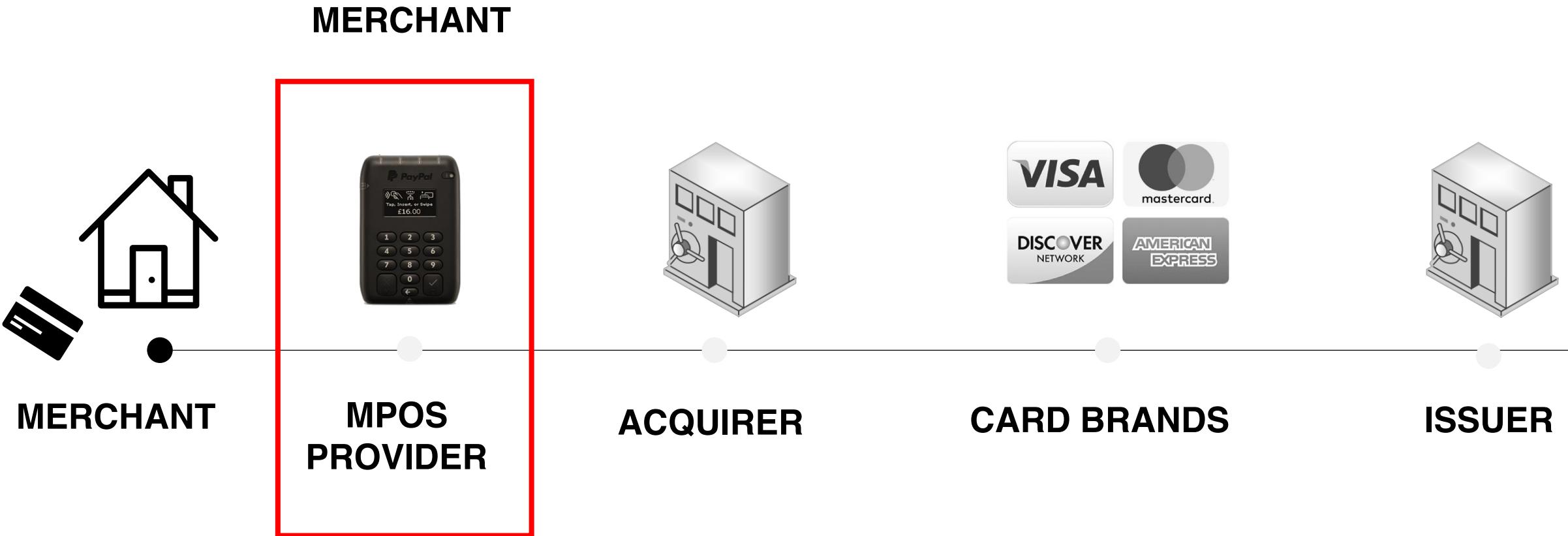
# Research Scope



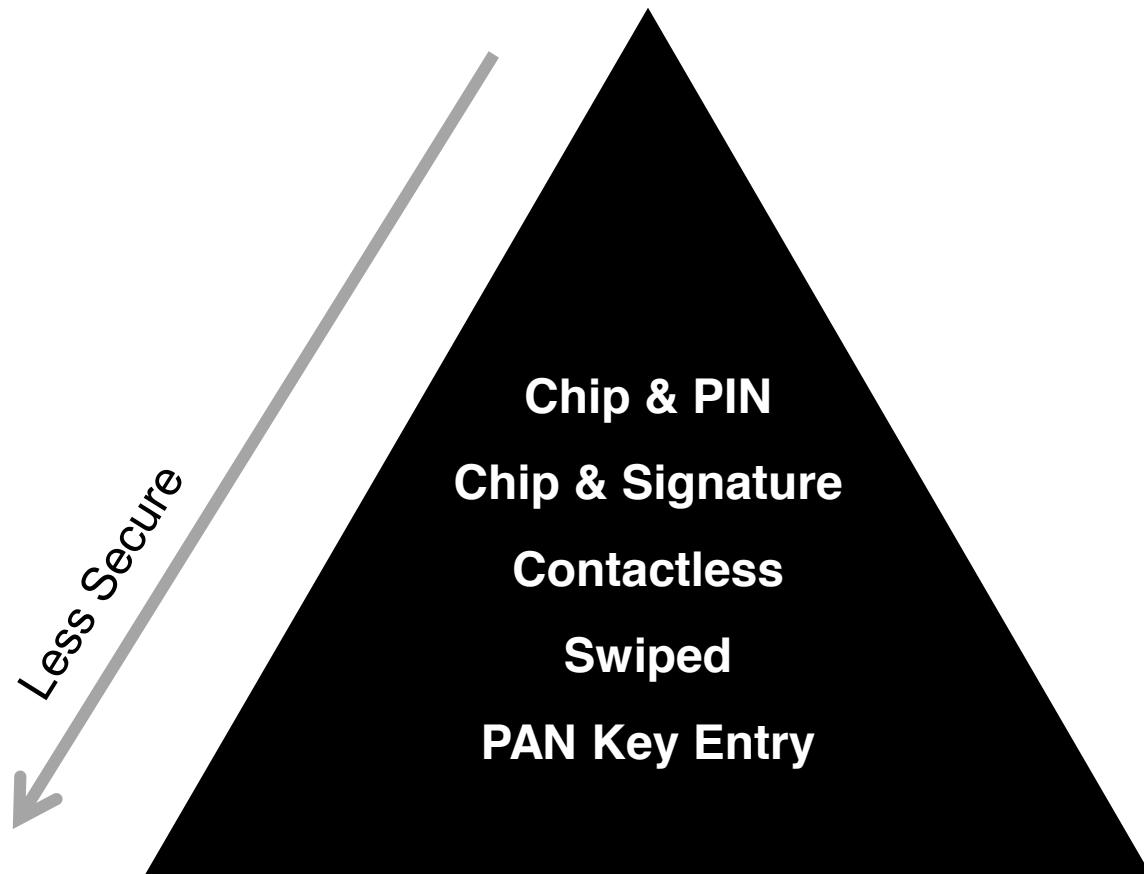
## Background



## Background



## CARD RISK BY OPERATION TYPE



## Background

# GLOBAL ADOPTION OF EMV - POS TERMINALS

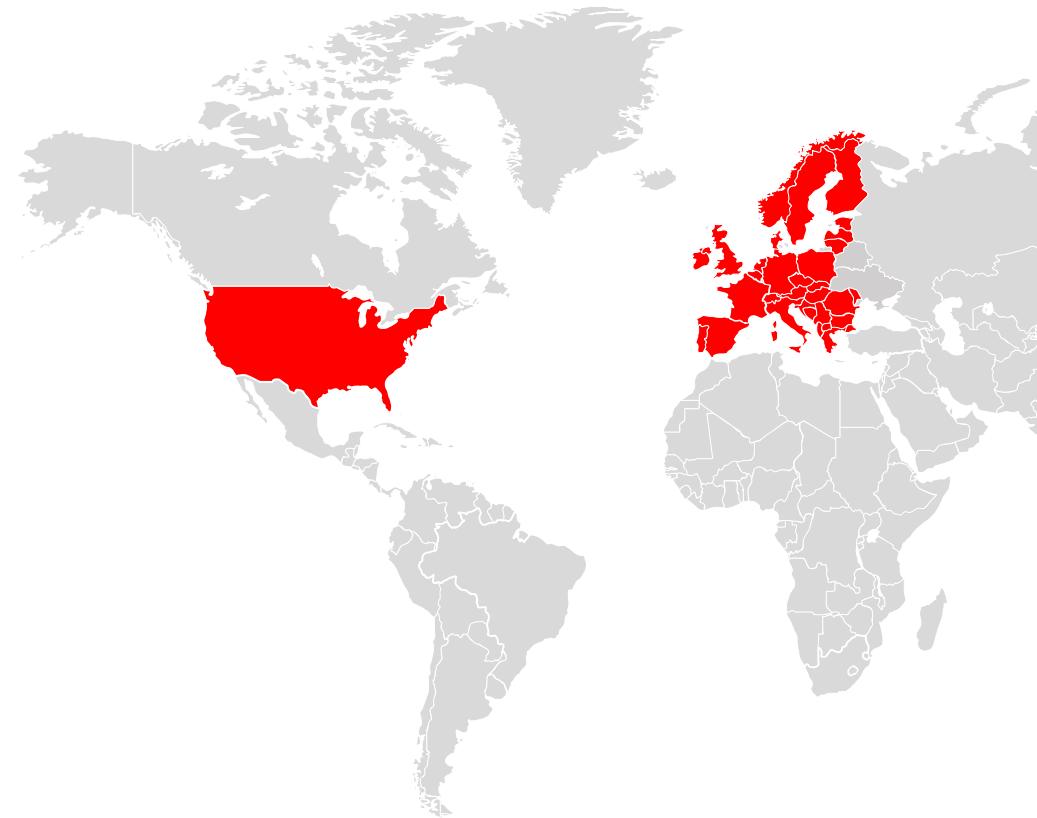
## EU EMV ACCEPTANCE

EMV enabled POS devices make up between 90-95% of POS population

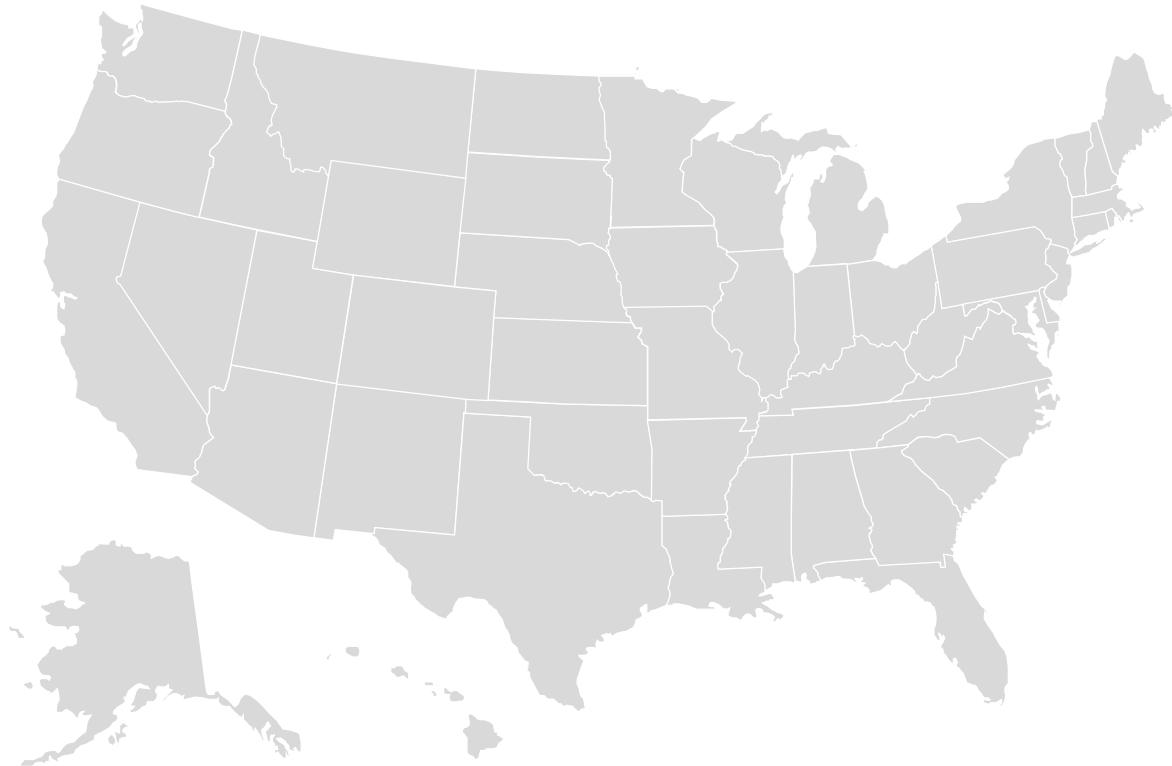


## US EMV ACCEPTANCE

EMV enabled POS devices make up 13% of POS population and 9% of the ATM population



## Background



### EMV CREDIT CARD ADOPTION

Around 96% of credit cards in circulation support EMV as a protocol

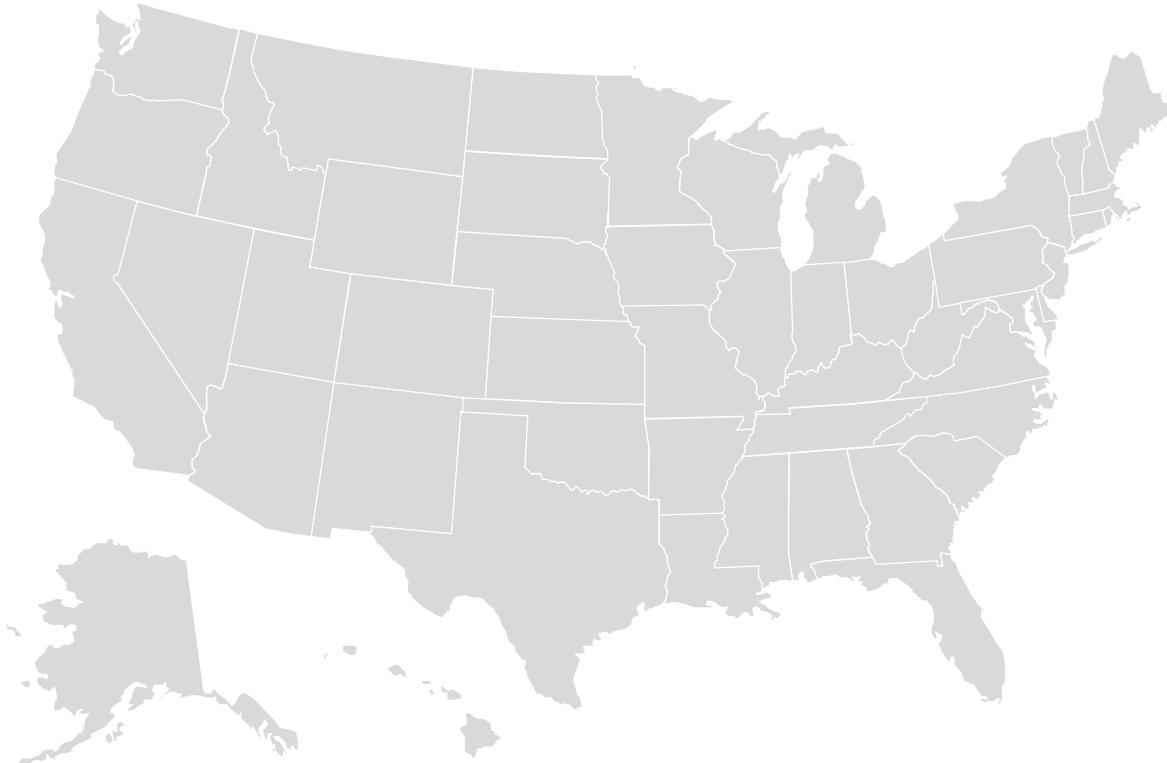


### EMV CREDIT CARD USAGE

However less than half of all transactions are made by chip



## Background



### EMV DEBIT CARD ADOPTION

79% of debit cards in circulation support EMV as a protocol

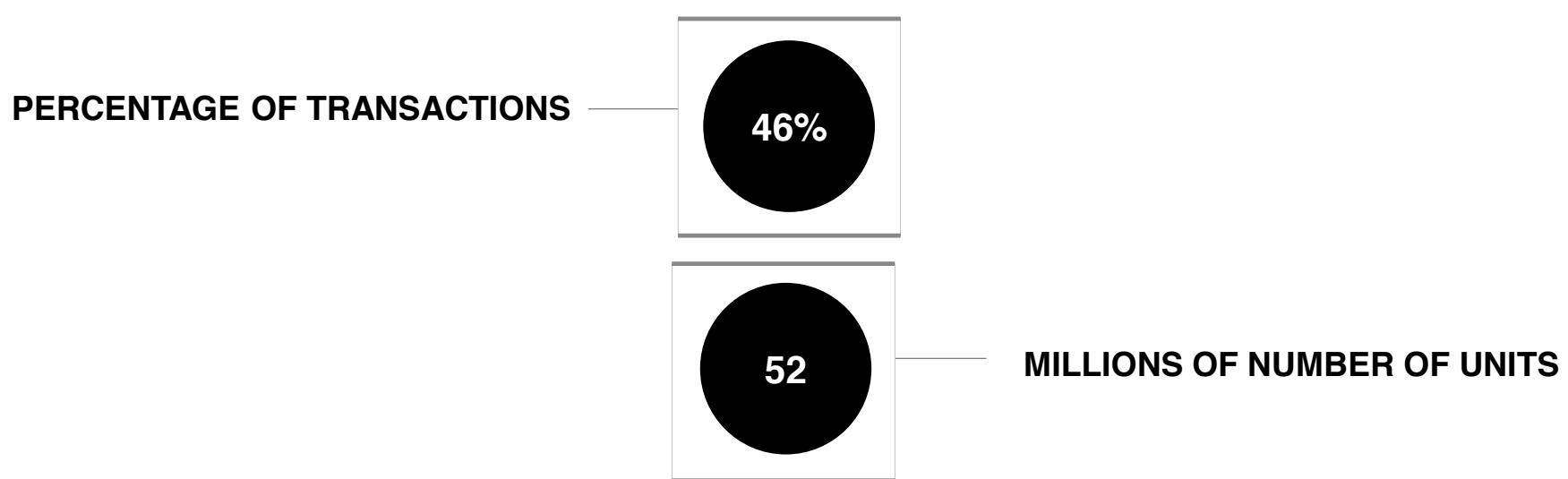


### EMV DEBIT CARD USAGE

However less than half of all transactions are made using chip

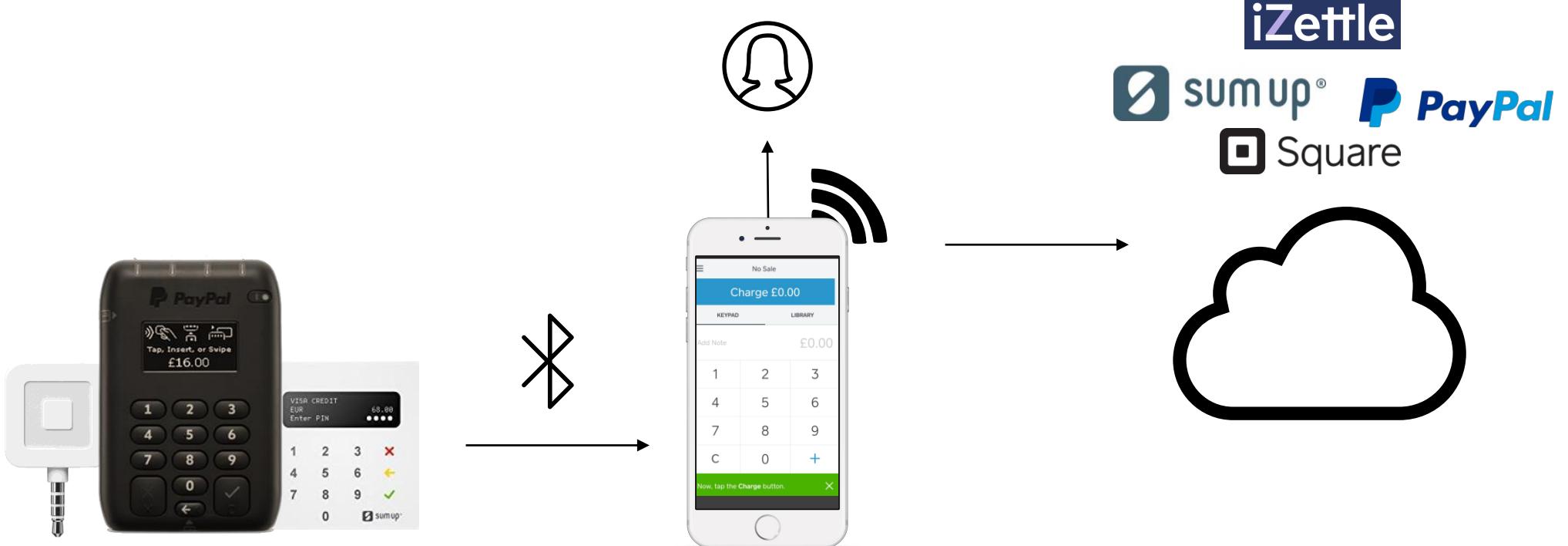


# MPOS TIMELINE



## Background

# SCHEMATIC OVERVIEW OF COMPONENTS

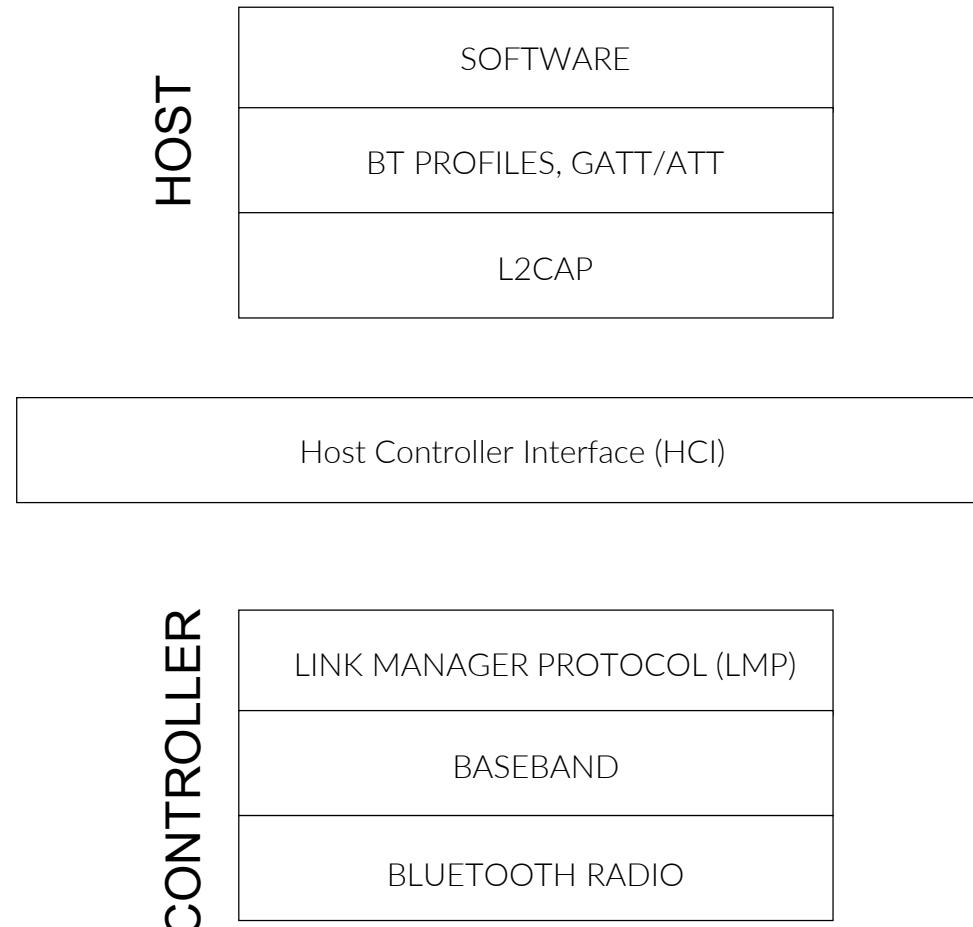


## VULNERABILITIES

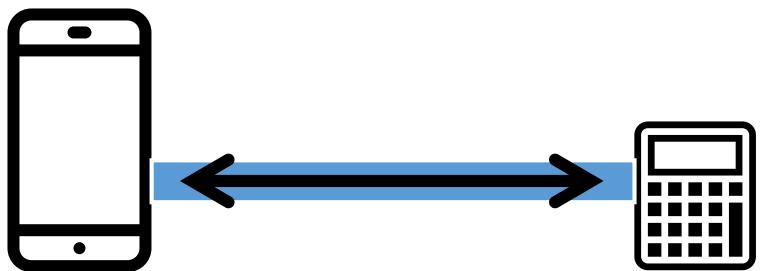
- › SENDING ARBITRARY COMMANDS
- › AMOUNT MODIFICATION
- › REMOTE CODE EXECUTION
- › HARDWARE OBSERVATIONS
- › SECONDARY FACTORS

# BLUETOOTH

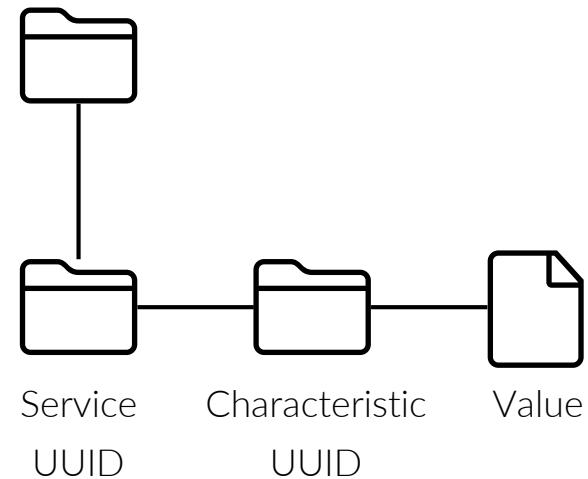
# BLUETOOTH PROTOCOL



RFCOMM



GATT (Generic Attribute)  
/ATT(Attribute Protocol)



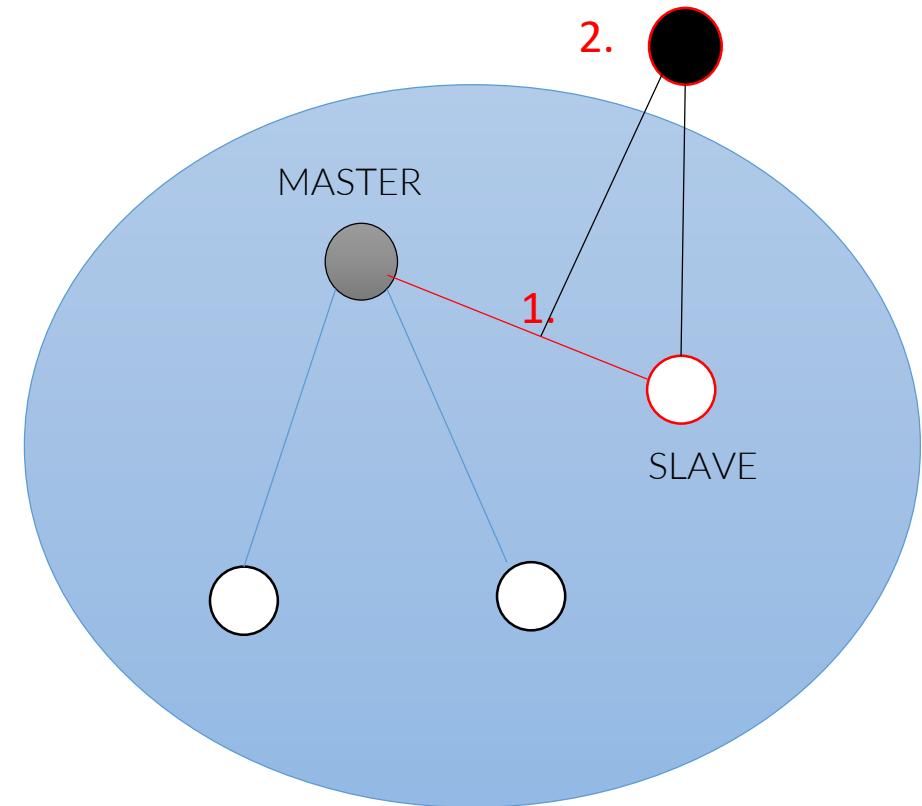
# BLUETOOTH AS A COMMUNICATION CHANNEL



NAP	UAP	LAP
68:AA	D2	0D:CC:3E
Org Unique Identifier		Unique to device

## BLUETOOTH ATTACK VECTORS

- › Eavesdropping/MITM
- › Manipulating characteristics



## Methods & Tools

Frontline BPA 600



\$20,000

Ubertooth One



\$120

## Methods & Tools

```
7 0.707450700          BT BR/EDR RF
8 0.709992400          BT BR/EDR RF
9 0.833738700          BT BR/EDR RF
10 0.846269000         BT BR/EDR RF
11 0.857516400         BT BR/EDR RF

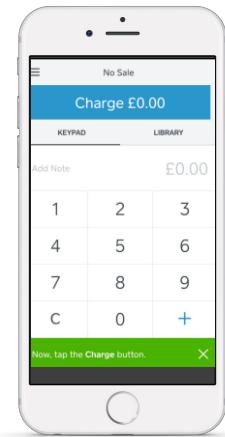
...0 ..... .... = MIC Checked: False
.... 0.... .... = CRC Pass: False
.... .0.. .... .... = CRC Checked: False
.... ..0. .... .... = HEC Pass: False
.... ...0 .... .... = HEC Checked: False
.... .... 1.... .... = Reference Upper Address Part Valid: True
.... .... .0.. .... .... = RF Channel Aliasing: False
.... .... ..0. .... .... = BR or EDR Data Present: False
.... .... ...1 .... .... = Reference Lower Address Part Valid: True
.... .... .... 0... .... = BR or EDR Payload Decrypted: False
.... .... .... .0.. .... = Noise Power Valid: False
.... .... .... .... 1 = Signal Power Valid: True
.... .... .... .... 1 = Packet Header and BR/EDR Payload Dewhitened: True

0000 0d c1 c9 01 00 00 00 00 3e cc 0d 00 3e cc 0d d2 .....>...>...
0010 00 00 00 00 93 00 .....  
.....
```

# **SENDING ARBITRARY COMMANDS**

# MANIPULATING CHARACTERISTICS

- Initiate a function
- Display text
- Turn off or on



User authentication doesn't exist in the Bluetooth protocol, it must be added by the developer at the application layer

# Findings

1.



2.



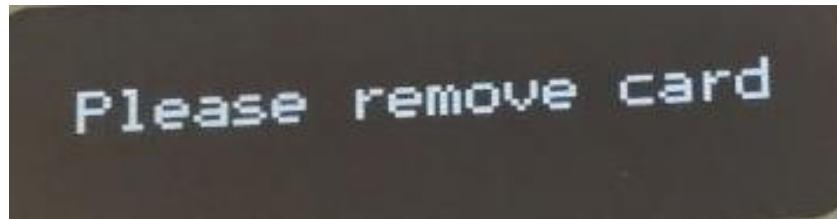
3.



## Findings

```
localhost () Rcvd UIH Channel=1 UID
localhost () Rcvd "\030\004\001\000\000\000\000\000\000"
localhost () Sent "\031\005\001\000\000\027\000\003\000\000\024\000Insert/swipe cardI"
localhost () Rcvd Number of Completed Packets
localhost () Rcvd UIH Channel=1 UID
localhost () Rcvd "\031\005\001\000\000\000\000\035"
controller Sent Sniff Mode
host Rcvd Command Status (Sniff Mode)
host Rcvd Mode Change
host
Frame 1731: 44 bytes on wire (352 bits), 44 bytes captured (352 bits)
└ Bluetooth
    [Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
    [Destination: Datecs_0d:cc:3e (68:aa:d2:0d:cc:3e)]
└ Bluetooth HCI H4
    [Direction: Sent (0x00)]
    HCI Packet Type: ACL Data (0x02)
└ Bluetooth HCI ACL Packet
    .... 0000 0011 0010 = Connection Handle: 0x032
    ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
    00.. .... .... .... = BC Flag: Point-To-Point (0)
    Data Total Length: 39
    Data
        [Connect in frame: 1579]
        [Disconnect in frame: 1771]
        [Source BD_ADDR: 00:00:00_00:00:00 (00:00:00:00:00:00)]
        [Source Device Name: ]
        [Source Role: Master (1)]
        Destination BD_ADDR: Datecs_Adress:3e_68_aa_d2_Adress:3e(1)
        0000 02 32 20 27 00 23 00 00 0e 0b ff 3d 01 19 05 01 .2 '#... ...=.....
        0010 00 00 17 00 03 00 00 14 00 49 6e 73 65 72 74 2f ..... Insert/
        0020 73 77 69 70 65 20 63 61 72 64 49 86  swipe ca rdi.
```

# Findings



```
> Frame 272: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
  Bluetooth
    [Source: SamsungE_ee:d3:be (34:2d:0d:ee:d3:be)]
    [Destination: cf:e9:ef:4f:6a:93 (cf:e9:ef:4f:6a:93)]
  Bluetooth HCI H4
    [Direction: Sent (0x00)]
    HCI Packet Type: ACL Data (0x02)
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
  Bluetooth Attribute Protocol
    > Opcode: Write Command (0x52)
    <-- Handle: 0x001b (Unknown: Unknown)
      [Service UUID: d839fc3c84dd4c369126187b07255127]
      [UUID: b378db854ec34daa828e1b99607bd6a0]
      [Value: 00000000000000000000000000000000]
```

0000	02 10 00 1b 00 17 00 04 00 52 1b 00	02 00 1d 06	.....R.....
0010	01 0b 00 00 00 01 00 13 50 6c 65 61 73 65 20 72		..... Please r

272	36.187350	SamsungE_ee:d3:be (... cf:e9:ef:4f:6a:93 () ATT	24 Sent Write Command, Handle: 0x00
274	36.177643	SamsungE_ee:d3:be (... cf:e9:ef:4f:6a:93 () ATT	28 Sent Write Command, Handle: 0x00
278	36.237365	SamsungE_ee:d3:be (... cf:e9:ef:4f:6a:93 () ATT	23 Sent Write Command, Handle: 0x00

```
> Frame 274: 28 bytes on wire (224 bits), 28 bytes captured (224 bits)
  ▼ Bluetooth
    [Source: SamsungE_ee:d3:be (34:2d:0d:ee:d3:be)]
    [Destination: cf:e9:ef:4f:6a:93 (cf:e9:ef:4f:6a:93)]
  ▼ Bluetooth HCI H4
    [Direction: Sent (0x00)]
    HCI Packet Type: ACL Data (0x02)
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
  ▼ Bluetooth Attribute Protocol
    > Opcode: Write Command (0x52)
    ▼ Handle: 0x001b (Unknown: Unknown)
      [Service UUID: d839fc3c84dd4c369126187b07255127]
      [UUID: b378db854ec34daa828e1b99607bd6a0]
    Value: 656d6f7665206361726400ff083c6203
```

0000	02	10	00	17	00	13	00	04	00	52	1b	00	65	6d	6f	76	.....	.R..	emov
0010	65	20	63	61	72	64	00	ff	08	3c	62	03	e	card..	.<b.				

## Findings

Handle: 0x001b (Unknown: Unknown)

[Service UUID: d839fc3c84dd4c369126187b07255127]

[UUID: b378db854ec34daa828e1b99607bd6a0]

Value: 02001d06010b000000010013506c656173652072

Handle: 0x001b (Unknown: Unknown)

[Service UUID: d839fc3c84dd4c369126187b07255127]

[UUID: b378db854ec34daa828e1b99607bd6a0]

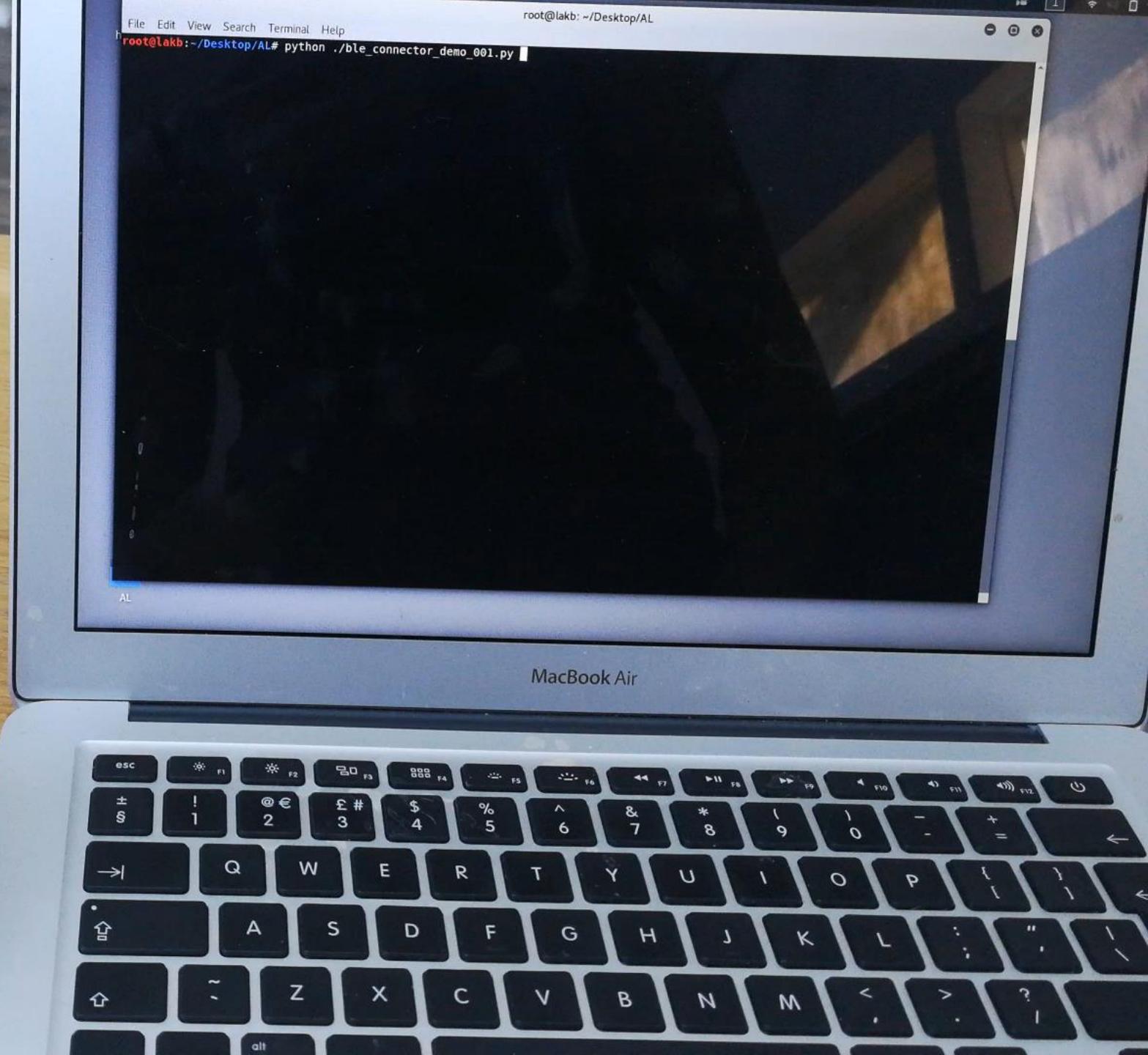
Value: 656d6f7665206361726400ff083c6203

LEADING PART	MESSAGE	TRAILING PART	CRC	END
02001d06010b000000 010013	506c656173652072656d6f76652063 617264	00ff08	3c62	03
“Please remove card”				

## ATTACK VECTORS

1. Force cardholder to use a more vulnerable payment method such as mag-stripe
2. Once the first payment is complete, display “Payment declined”, force cardholder to authorise additional transaction.





# AMOUNT TAMPERING

## **HOW TO GET ACCESS TO TRANSACTIONS AND COMMANDS**

- › HTTPS
- › DEVELOPER BLUETOOTH LOGS
- › RE OF APK ENABLE DEBUG
- › BLUETOOTH SNIFFER

## Findings

# HOW TO GET ACCESS TO COMMANDS

1.  $0x02ee = 7.50 \text{ USD}$        $0x64cb = \text{checksum}$

```
> Bluetooth L2CAP Protocol      V/ (10152) : (SourceFile:31)@BtSmart-Receiver | Message length pa
└ Bluetooth Attribute Protocol   D/ (10152) : (SourceFile:31)@BtSmart-Receiver | Message complete,
                                         02ee
                                         64cb Remaining bytes:
0000 ..... R...0.
0010 ..<...). ....
```

The screenshot shows a debugger interface with two panes. The left pane shows a tree view with 'Bluetooth L2CAP Protocol' expanded and 'Bluetooth Attribute Protocol' selected. The right pane displays a hex dump of a message. The bytes **02 ee** at offset 0010 are highlighted with a red box. Above them, the bytes **00 02** are also highlighted. To the right of the bytes, the ASCII representation shows **..... R...0.**. Below the bytes, the text **..<...). ....** is visible.

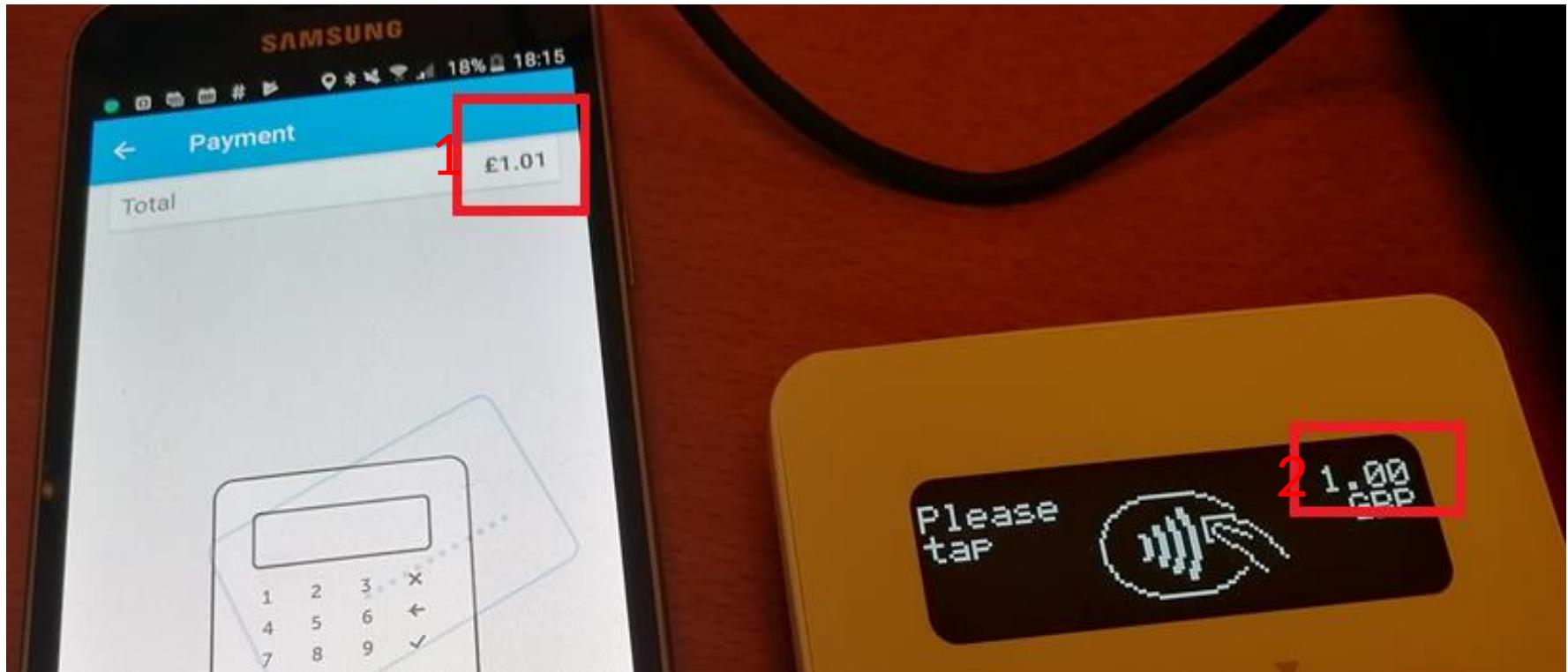
2.  $0100 = 1.00 \text{ USD}$        $0x8a = \text{checksum}$

```
J config
J config
        }
    },
    "INIT_TRANSACTION": {
        "COMMANDS": [
            {
                "HEX": "9F0206[AMOUNT]9A"
            }
        ],
        "PARAMETERS": {
            "AMOUNT": {
                "FIXED_LENGTH": 12
            }
        }
    }
}
```

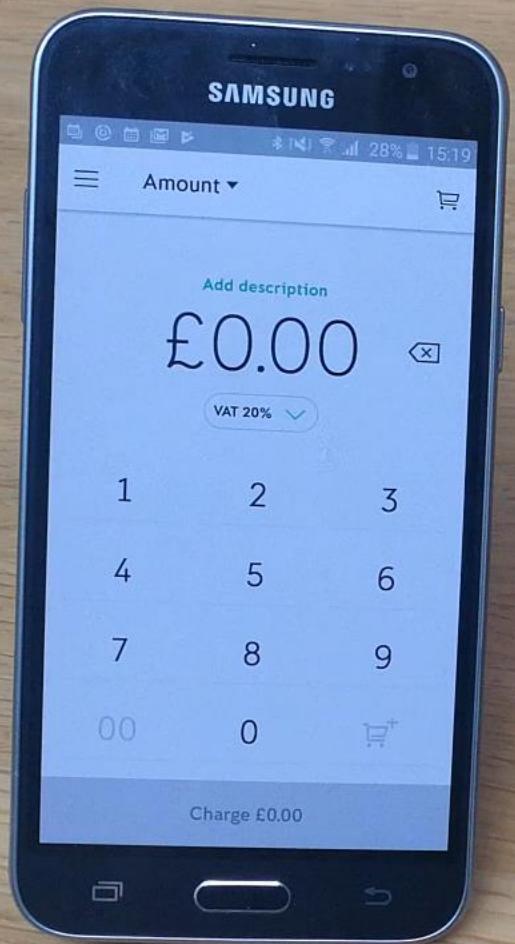
The screenshot shows a code editor with syntax highlighting. On the left, there are two tabs labeled 'config'. The main code area contains a JSON-like structure. The byte sequence **9F0206[AMOUNT]9A** is highlighted with a red box. The word **[AMOUNT]** is also highlighted in orange.

## MODIFYING PAYMENT AMOUNT

1. Modified payment value
2. Original (lower) amount displayed on card reader for the customer
3. Card statement showing higher authorised transaction amount



3 Date	Card Detail	Amount
14/03/18	3005 18031316504027569 Card purchase	-£1.01



# MODIFYING PAYMENT AMOUNT

Type of Payment	Amount Tampering	Security Mechanisms
MAG-STRIPE	TRACK2	----
CONTACTLESS	POSSIBLE	AMOUNT CAN BE STORED IN CRYPTOGRAM
CHIP AND PIN	-----	AMOUNT IS STORED IN CRYPTOGRAM

LIMIT PER TRANSACTION: 50,000 USD

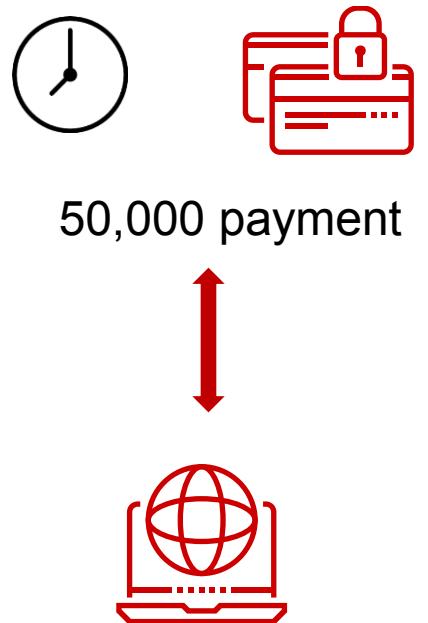
# ATTACK



Customer



Fraudulent merchant



Service Provider

## **MITIGATION ACTIONS FOR SERVICE PROVIDERS**

- › DON'T USE VULNERABLE OR OUT-OF-DATE FIRMWARE
- › NO DOWNGRADES
- › PREVENTATIVE MONITORING

# **REMOTE CODE EXECUTION**

**RCE = 1 REVERSE ENGINEER + 1 FIRMWARE**



@ivachyou

## HOW FIRMWARE ARRIVES ON THE READER

[https://frw.\\*\\*\\*\\*\\*.com/\\_prod\\_app\\_1\\_0\\_1\\_5.bin](https://frw.*****.com/_prod_app_1_0_1_5.bin)  
[https://frw.\\*\\*\\*\\*\\*.com/\\_prod\\_app\\_1\\_0\\_1\\_5.sig](https://frw.*****.com/_prod_app_1_0_1_5.sig)

[https://frw.\\*\\*\\*\\*\\*.com/\\_prod\\_app\\_1\\_0\\_1\\_4.bin](https://frw.*****.com/_prod_app_1_0_1_4.bin)  
[https://frw.\\*\\*\\*\\*\\*.com/\\_prod\\_app\\_1\\_0\\_1\\_4.sig](https://frw.*****.com/_prod_app_1_0_1_4.sig)

Header	- RSA-2048 signature ( <i>0x00 - 0x100</i> )
Body	- AES-ECB encrypted

## HOW FIRMWARE ARRIVES ON THE READER

The screenshot shows a search interface with a search bar containing the query "paypalobjects" mpi tar.gz. Below the search bar is a navigation menu with tabs: All (highlighted in blue), Videos, News, Shopping, Images, More, and Settings. A microphone icon is located in the top right corner of the search bar. The search results section indicates "About 40 results (0.33 seconds)". The first result is a link to a GitHub repository: arun-paypal-issue/paypal log at master · arunjnair15/arun-paypal ... The link is <https://github.com/arunjnair15/arun-paypal-issue/blob/master/paypal%20log>. Below the link, the timestamp is 11 Jul 2017, followed by a truncated URL: "https://www.paypalobjects.com/webstatic/mobile/pph/sw\_repo\_app/us/.../pph/sw\_repo\_app/us/miura/m010/prod/7/M000-MPI-V1-41.tar.gz".

[https://www.paypalobjects.com/webstatic/mobile/pph/sw\\_repo\\_app/us/miura/m010/prod/7/M000-MPI-V1-41.tar.gz](https://www.paypalobjects.com/webstatic/mobile/pph/sw_repo_app/us/miura/m010/prod/7/M000-MPI-V1-41.tar.gz)

[https://www.paypalobjects.com/webstatic/mobile/pph/sw\\_repo\\_app/us/miura/m010/prod/7/M000-MPI-V1-39.tar.gz](https://www.paypalobjects.com/webstatic/mobile/pph/sw_repo_app/us/miura/m010/prod/7/M000-MPI-V1-39.tar.gz)

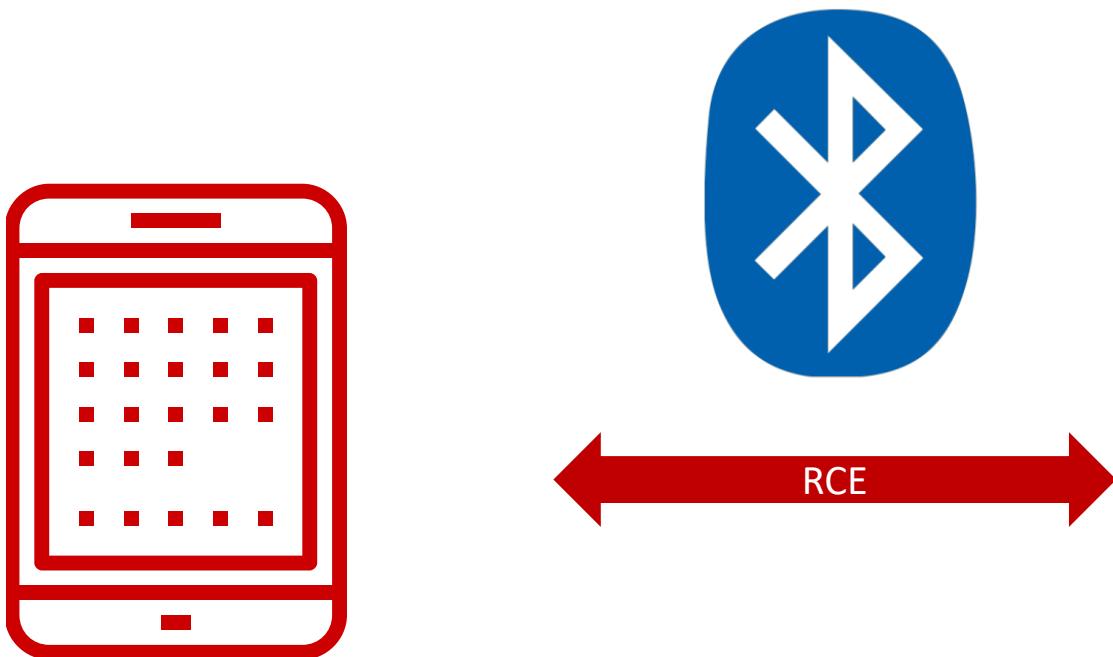
# HOW FIRMWARE ARRIVES ON THE READER

 EMV-Config	
 Images	
 SecureConfig	
 Retail-API	
 M000-EMVL2CL-V1-10.tar.gz	7 206
 M000-EMVL2K3-V1-0.tar.gz	87 452
 dbus-pinagent	350 972
 M000-EMVL2K2-V1-0.tar.gz	870 885
 libcrypto.so.1.0.0	12 805
	100 225
	116 332
	115 268
	1 457 188

```
no_prompt
TRANSACTION DECLINED
    ENTER PIN
    PROCESSING ERROR
    REMOVE CARD
no_prompt

    PROCESSING CARD
Card was read. OK to remove card.
TRY ANOTHER INTERFACE
PRESENT ONLY ONE CARD
TRANSACTION APPROVED PLEASE SIGN RECEIPT
no_prompt
no_prompt
no_prompt
clear_screen
    SEE PHONE
PRESENT CARD AGAIN
REFER TO YOUR PAYMENT DEVICE
```

## HOW FIRMWARE ARRIVES ON THE READER



```
File Edit View Search Terminal
$ dmesg
Jun 29 17:04:58 miura -- MARK
$ dmesg
Jun 29 17:06:04 miura klogd:
$ uname
Jun 29 17:07:54 miura klogd:
$ uname -a
Jun 29 17:07:54 miura klogd:
$ id
Jun 29 17:07:59 miura klogd:
$ id
Jun 29 17:08:04 miura klogd:
$ re = 0x2a8, actual = 0x2a8
$ █
$ █
Linux miura 2.6.31-506-g30df5
15 armv5tejl GNU/Linux
uid=0(root) gid=0(root)
```

## **INFECTED MPOS**

- › PAYMENT ATTACKS
- › COLLECT TRACK 2/PIN
- › PAYMENT RESEARCH

# KrebsOnSecurity

In-depth security news and investigation

## Posts Tagged: EMV replay attack

The Coming Storm / Web Fraud 2.0 — 145 Comments

# 27 ‘Replay’ Attacks Spoof Chip Card Charges

OCT 14

An odd new pattern of credit card fraud emanating from Brazil and targeting U.S. financial institutions could spell costly trouble for banks that are just beginning to issue customers more secure chip-based credit and debit cards.

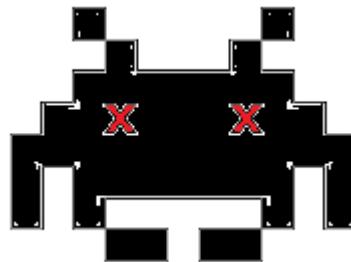
Over the past week, at least three U.S. financial institutions reported receiving tens of thousands of dollars in fraudulent credit and debit card transactions coming from Brazil and hitting card accounts stolen in recent retail heists, principally



# DEVICE PERSISTENCE



REBOOT



GAME OVER

# ATTACK

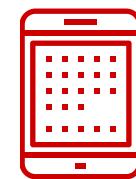


Device with  
a Bluetooth



RCE

Reader



UPDATES



Service Provider



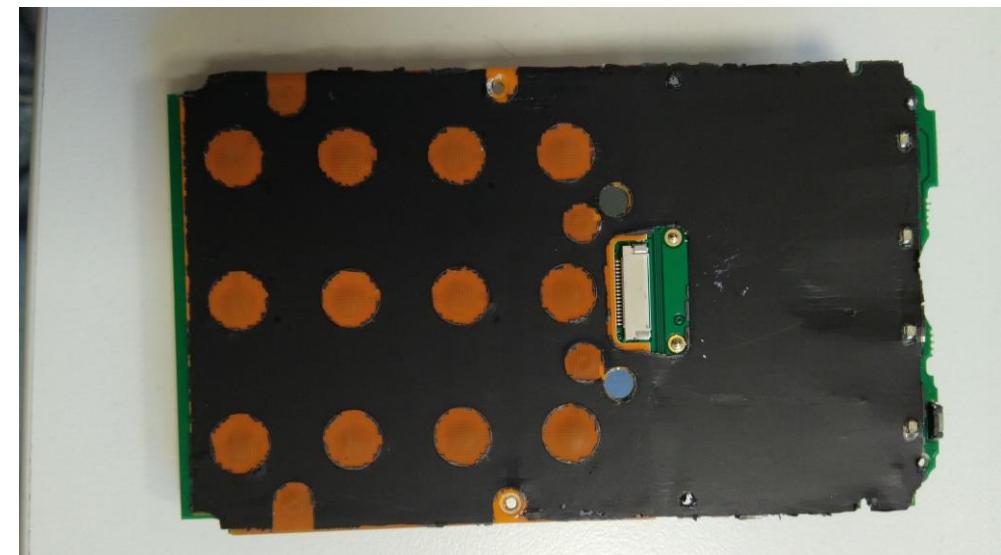
Merchant

## MITIGATIONS

- › NO VULNERABLE OR OUT-OF-DATED FIRMWARE
- › NO DOWNGRADES
- › PREVENTATIVE MONITORING



## HARDWARE OBSERVATIONS



## SECONDARY FACTORS

- ✓ ENROLMENT PROCESS
- ✓ ON BOARDING CHECKS VS TRANSACTION MONITORING
- ✓ DIFFERENCES IN GEO – MSD, OFFLINE PROCESSING
- ✓ WHAT SHOULD BE CONSIDERED AN ACCEPTED RISK?
- ✓ ACCESS TO HCI LOGS/APP, LOCATION SPOOFING

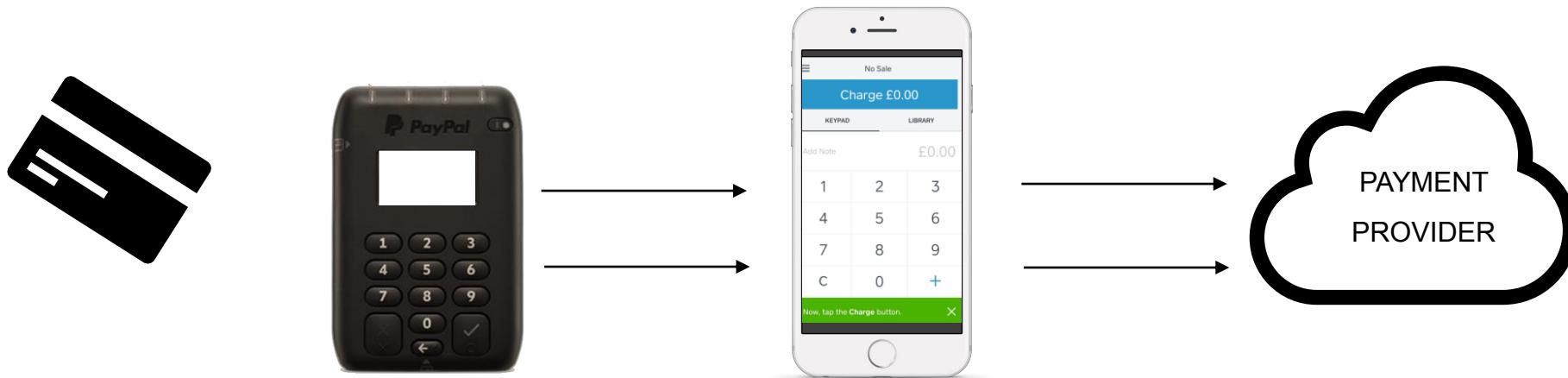


# Conclusions

Reader	Cost reader/Fee per transaction	Enrollment process	Antifraud + Security checks	Physical security	FW RE	Mobile Ecosystem	Arbitrary commands	Red teaming	Amount tampering
Square [EU]	\$51 1.75-2.5%		Strict – active monitoring of transactions	N/A	-	strict	-	-	-
Square [USA]	\$50 2.5-2.75%	Low - no anti money laundering checks but some ID checks	Strict – correlation of “bad” readers, phones and acc info	N/A	-	medium (dev)	-	+	-
Square mag-stripe [EU + USA]	Free 2.5-2.75%		Strict (see above)	Low	-	low	-	+	+ [no display]
Square miura [USA]	\$130 2.5-2.75%		Strict (see above)	N/A	+	N/A	+ [via RCE]	+	+ (via RCE)
PayPal miura	\$60 1-2.75%	High - anti-money laundering checks + credit check (to take out credit agreement)	Strict – transaction monitoring	N/A	+	low	+ [via RCE]	+	+ (via RCE)
SumUp datecs	\$40 1.69%	Low - no anti money laundering checks but some ID checks	Low – limited monitoring of accounts	Medium	-	low	+	+	+
iZettle datecs	\$40 1.75%	Medium - anti-money laundering check + ID checks	Low – limited monitoring, on finding suspect activity block withdrawal - acc otherwise active	High	-	low	+	-	+

# MPOS FOR RED TEAMING

1. Carry out an assessment of reader to gather preliminary data + info from cards.
2. Use data to carry out normal transactions to obtain baseline.
3. Use info obtained during this process to identify potential weaknesses and vulnerabilities.
4. Carry out “modified” transactions



## ASSESSING RISK - WHAT DOES THIS MEAN FOR YOUR BUSINESS?



## Conclusions



# CONCLUSIONS

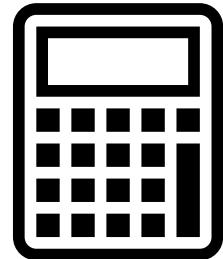
RECOMMENDATIONS FOR MPOS MANUFACTURERS



- › Control firmware versions, encrypt & sign firmware
- › Use Bluetooth pairing mode that provides visual confirmation of reader/phone pairing such as pass key entry
- › Integrate security testing into the development process
- › Implement user authentication and input sanitisation at the application level

# CONCLUSIONS

RECOMMENDATIONS FOR MPOS VENDORS



- › Protect deprecated protocols such as magstripe
- › Use preventive monitoring as a best practice
- › Don't allow use of vulnerable or out-of-date firmware, prohibit downgrades
- › Place more emphasis on enrolment checks
- › Protect the mobile ecosystem
- › Implement user authentication and input sanitization at application level

# CONCLUSIONS

RECOMMENDATIONS FOR MPOS MERCHANTS



- > Control physical access to devices
- > Do not use mag-stripe transactions
- > Assess the mPOS ecosystem
- > Choose a vendor who places emphasis on protecting whole ecosystem

# THANKS

Leigh-Anne Galloway



@L\_AGalloway

Tim Yunusov



@a66at

Hardware and firmware:  
Artem Ivachev

Hardware observations:  
Alexey Stennikov  
Maxim Goryachy  
Mark Carney