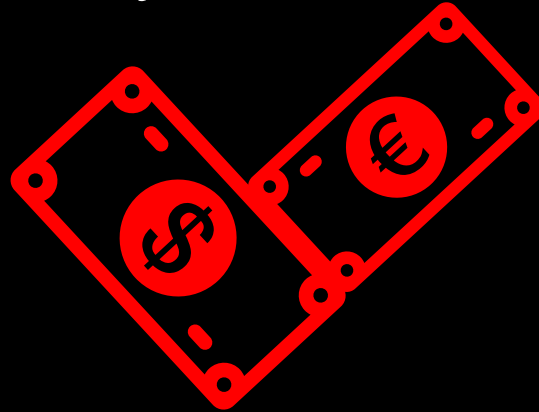


FOR THE LOVE OF MONEY

Finding and exploiting vulnerabilities in mobile point of sales
systems



LEIGH-ANNE GALLOWAY & TIM YUNUSOV

MPOS GROWTH



2010

Single vendor



2018

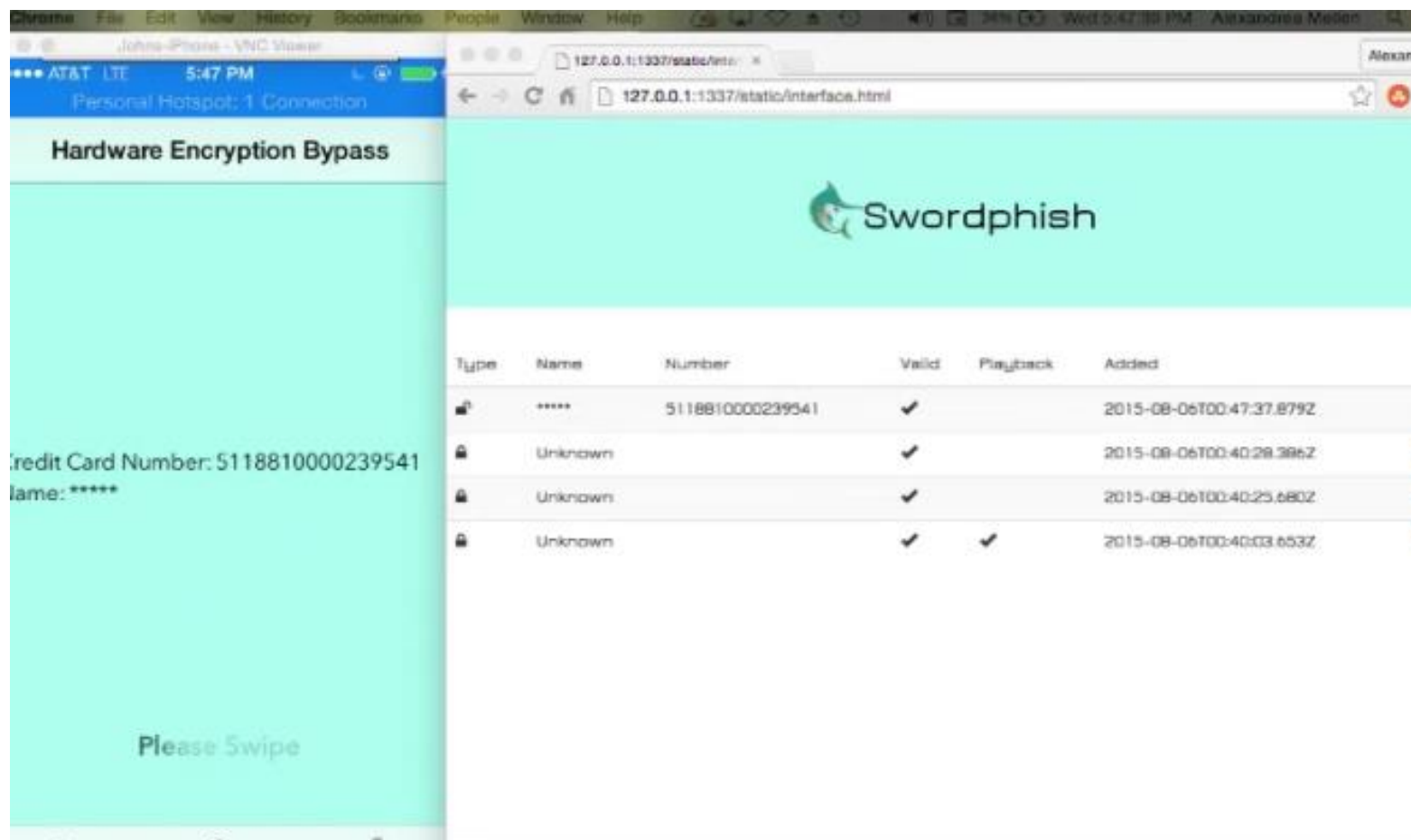
Four leading vendors
shipping thousands of units per day





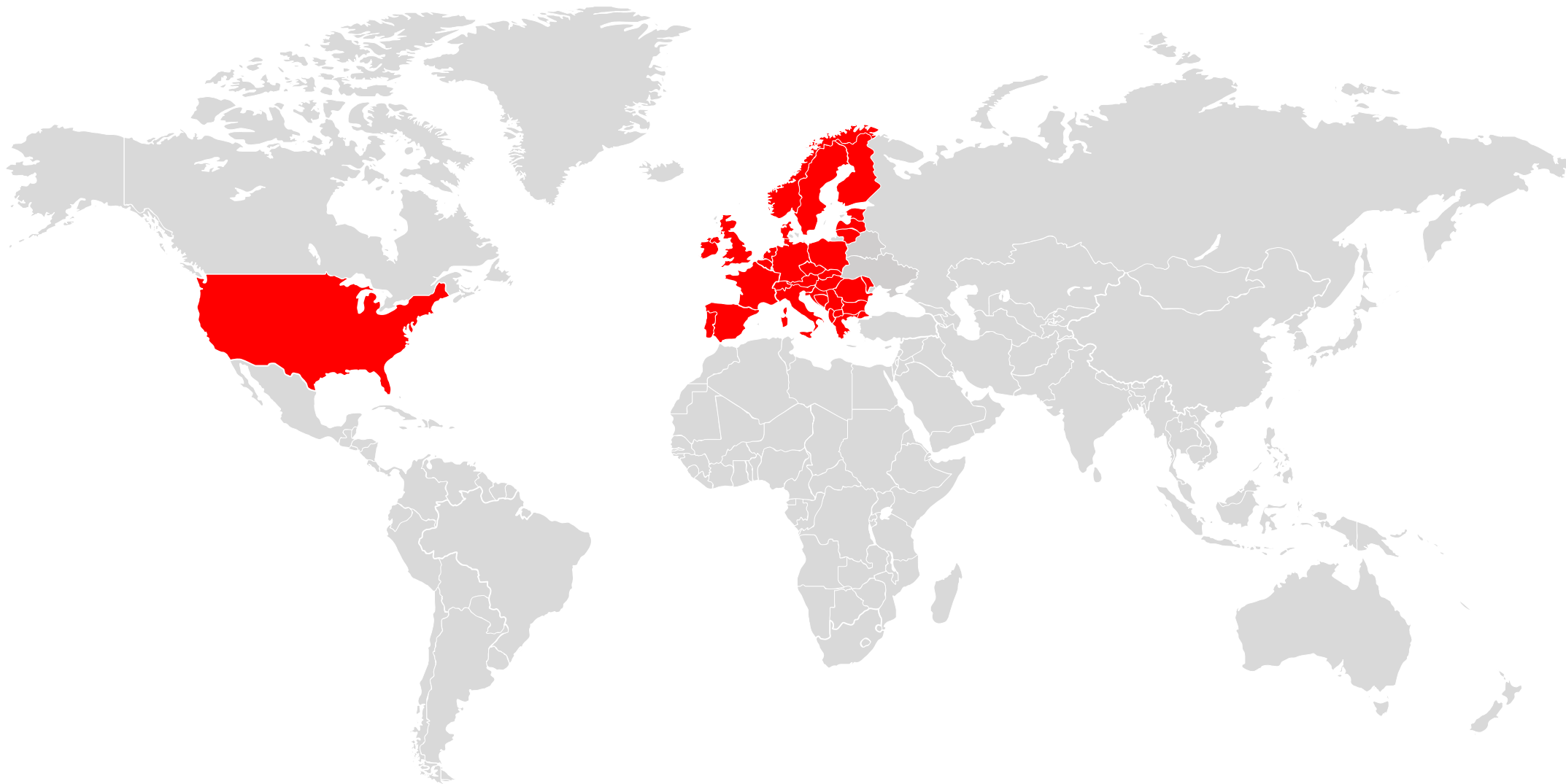
MWR Labs "Mission mPOSsible" 2014

Related Work



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





Research Scope



PAYPAL



SQUARE

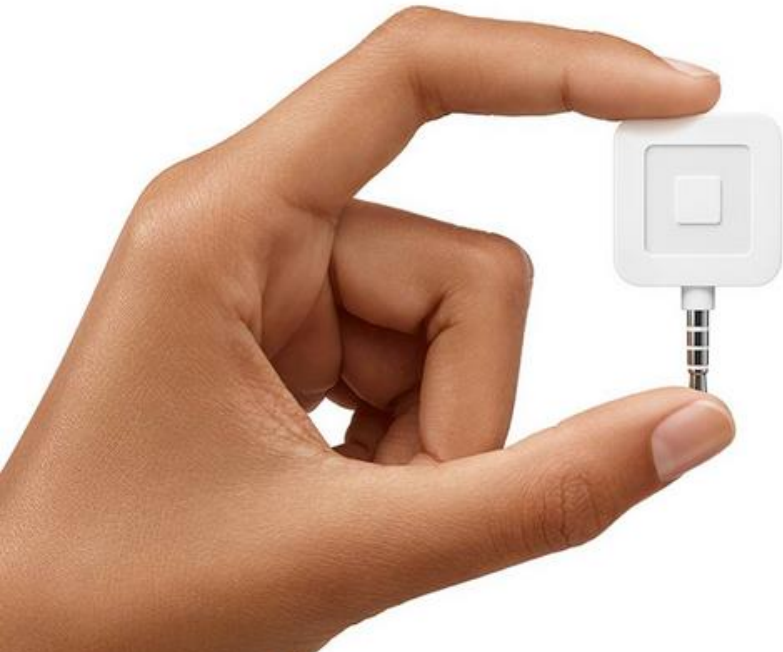


IZETTLE



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.

SECONDARY FACTORS



PHONE/SERVER



HARDWARE



DEVICE/PHONE



MOBILE APP



Background



MERCHANT



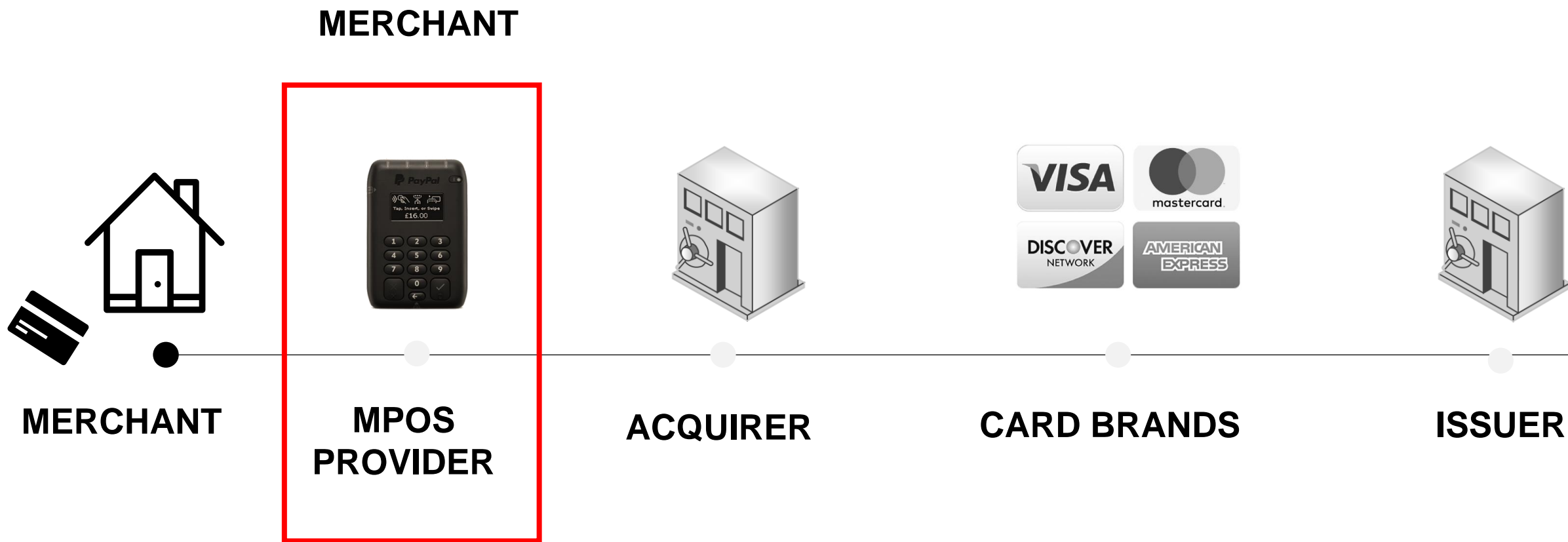
ACQUIRER



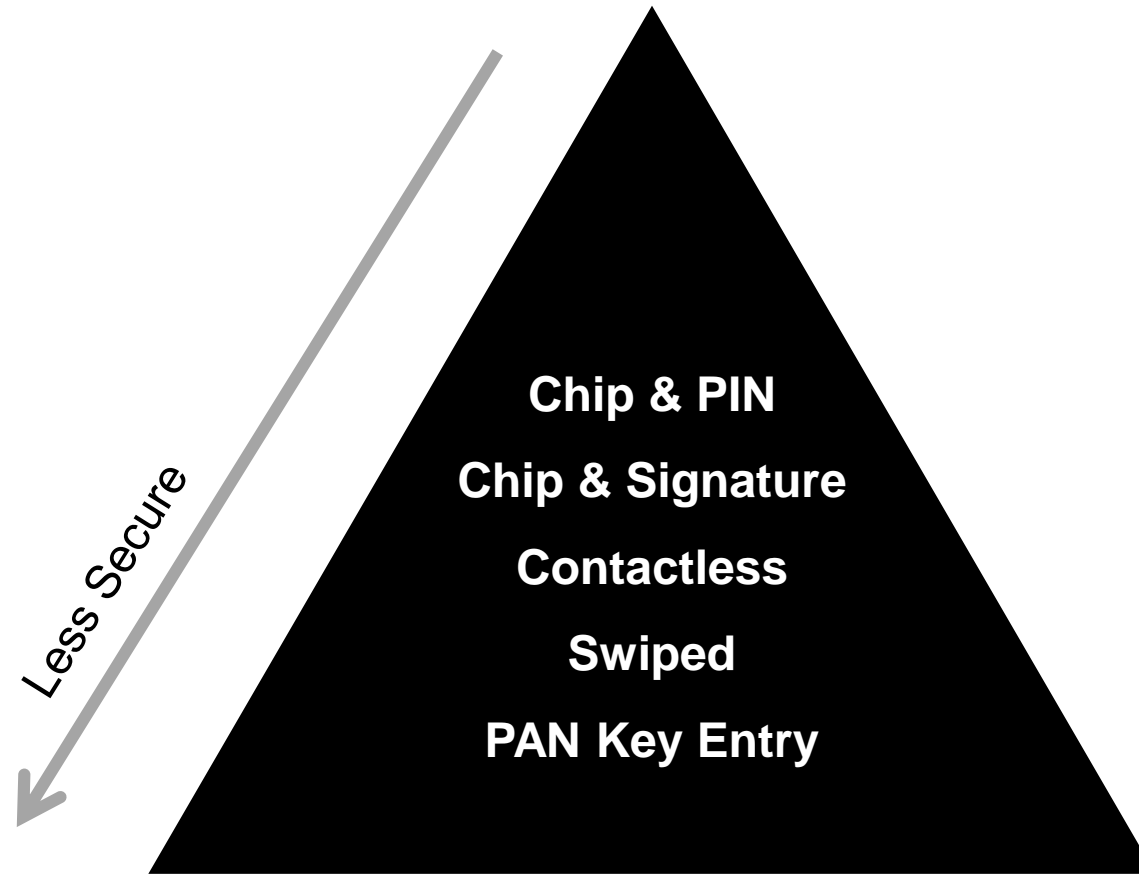
CARD BRANDS



ISSUER



CARD RISK BY OPERATION TYPE



GLOBAL ADOPTION OF EMV - POS TERMINALS

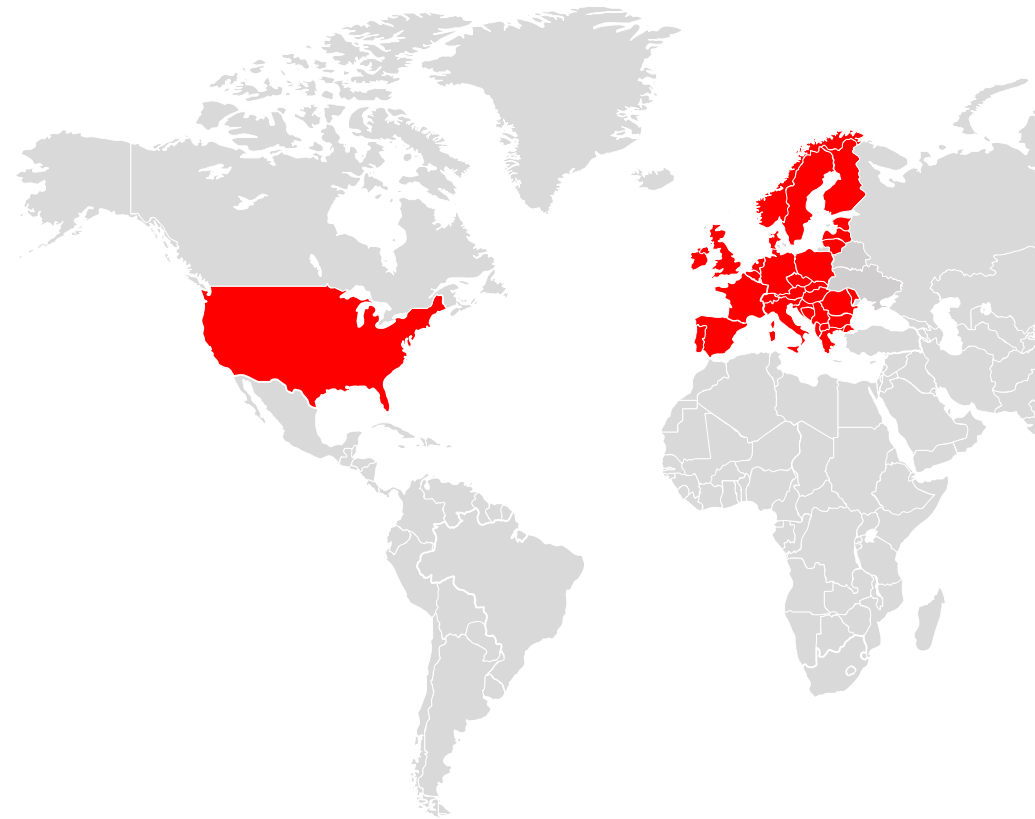
EU EMV ACCEPTANCE

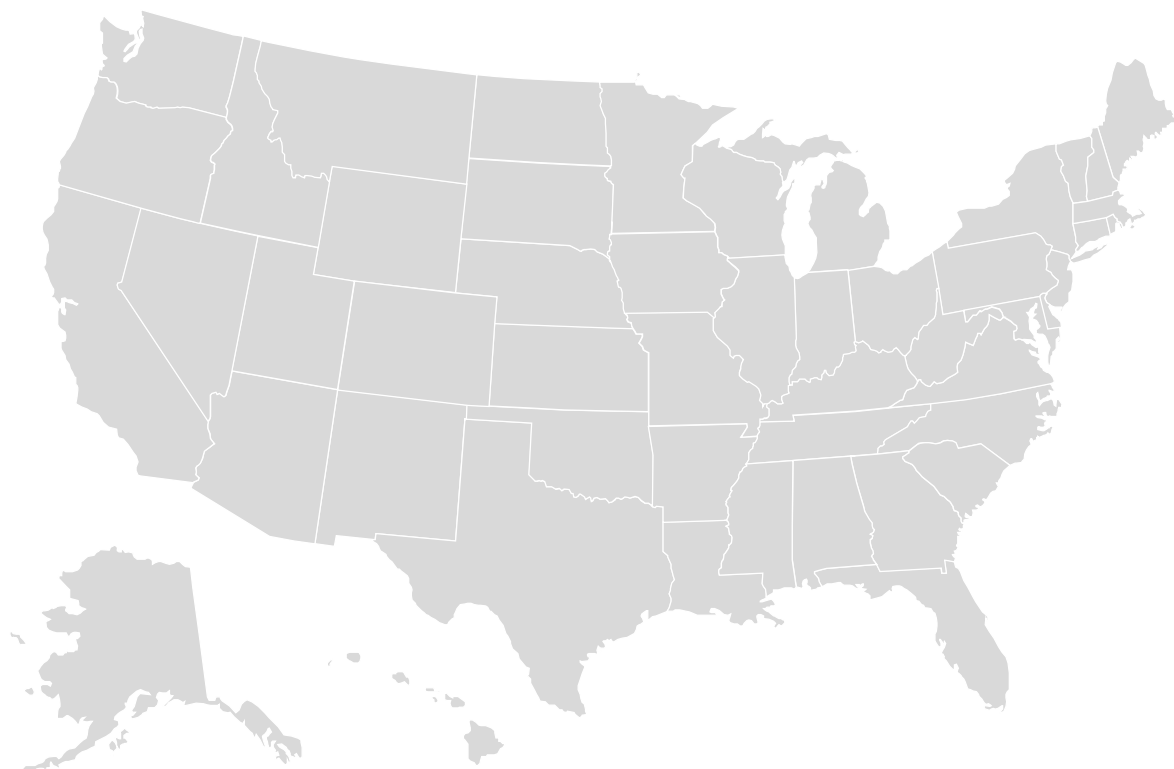
EMV enabled POS devices make up between 90-95% of POS population



US EMV ACCEPTANCE

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





EMV CREDIT CARD ADOPTION

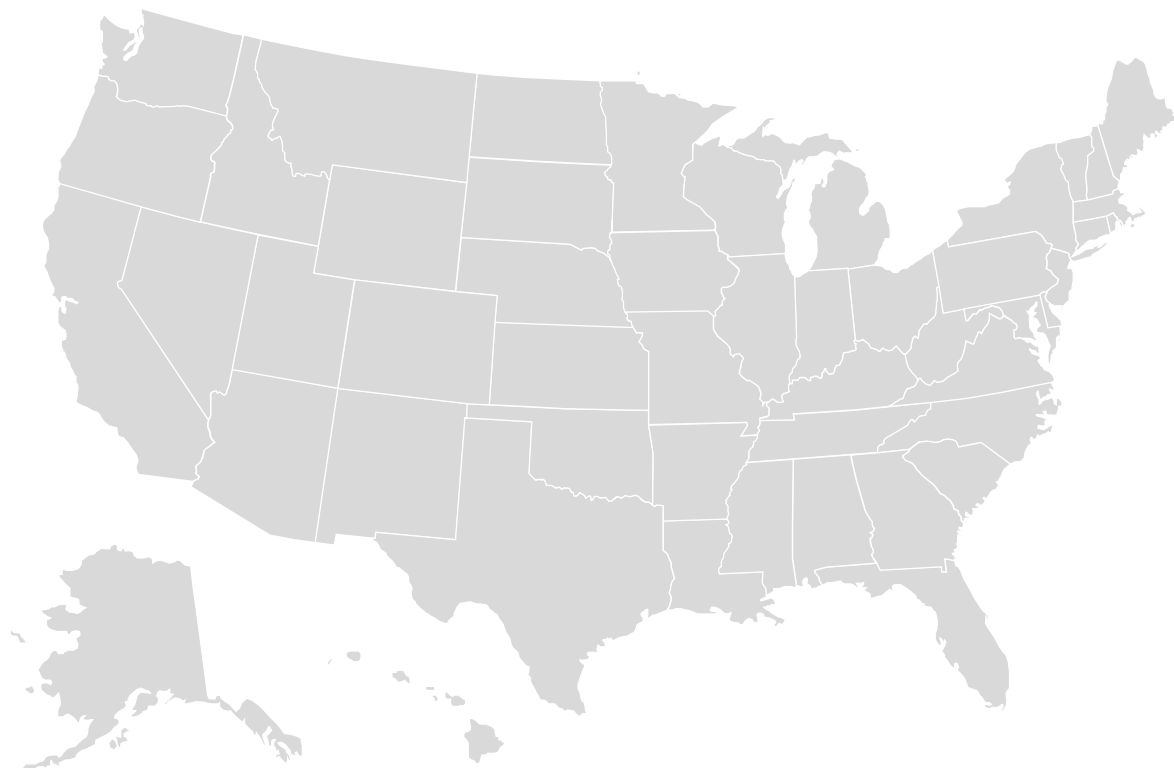
96% of credit cards in circulation support EMV as a protocol



EMV CREDIT CARD USAGE

However less than half of all transactions are made by chip





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 2019

PERCENTAGE OF TRANSACTIONS

46%

52

MILLIONS OF NUMBER OF UNITS

SCHEMATIC OVERVIEW OF COMPONENTS

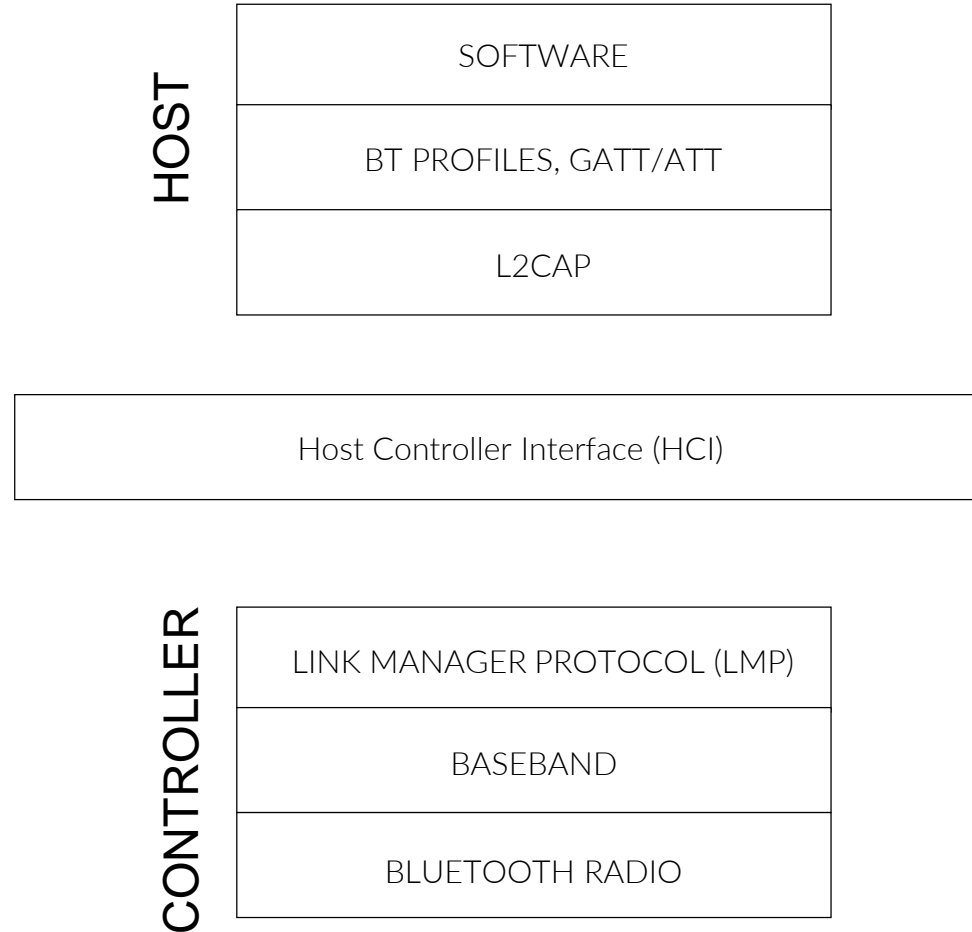


FINDINGS

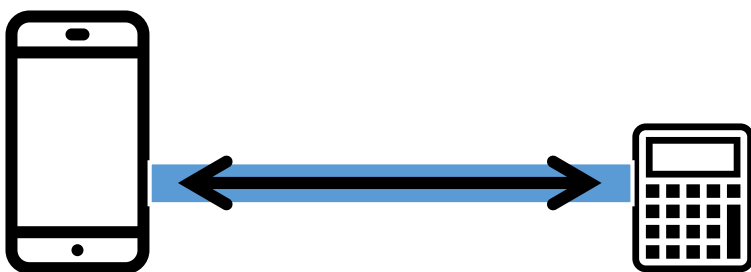
- SENDING ARBITRARY COMMANDS
- AMOUNT MODIFICATION
- REMOTE CODE EXECUTION
- HARDWARE OBSERVATIONS
- SECONDARY FACTORS

BLUETOOTH

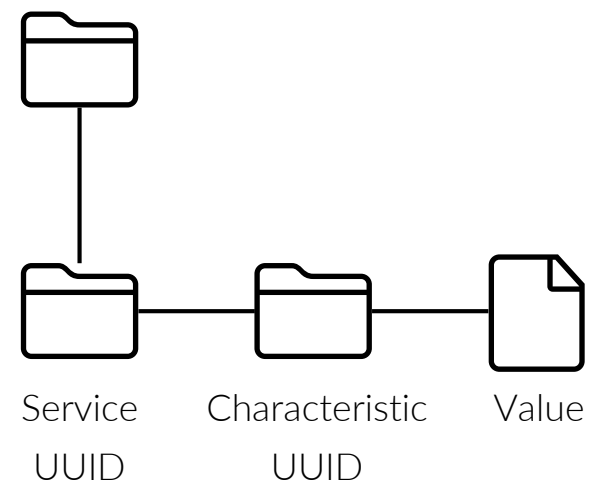
BLUETOOTH PROTOCOL



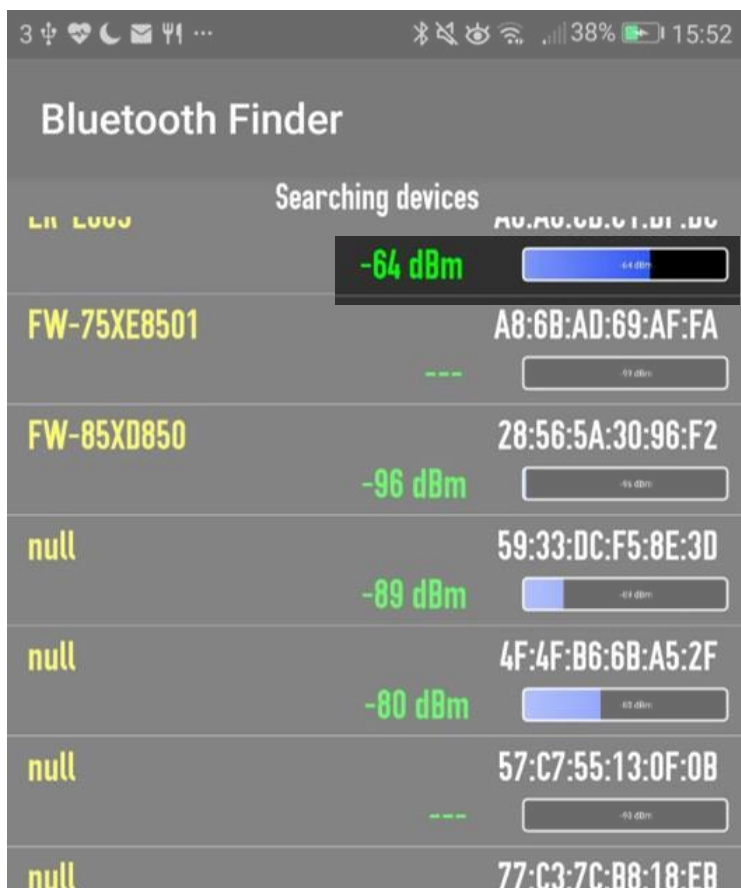
RFCOMM



GATT (Generic Attribute) /ATT(Attribute Protocol)



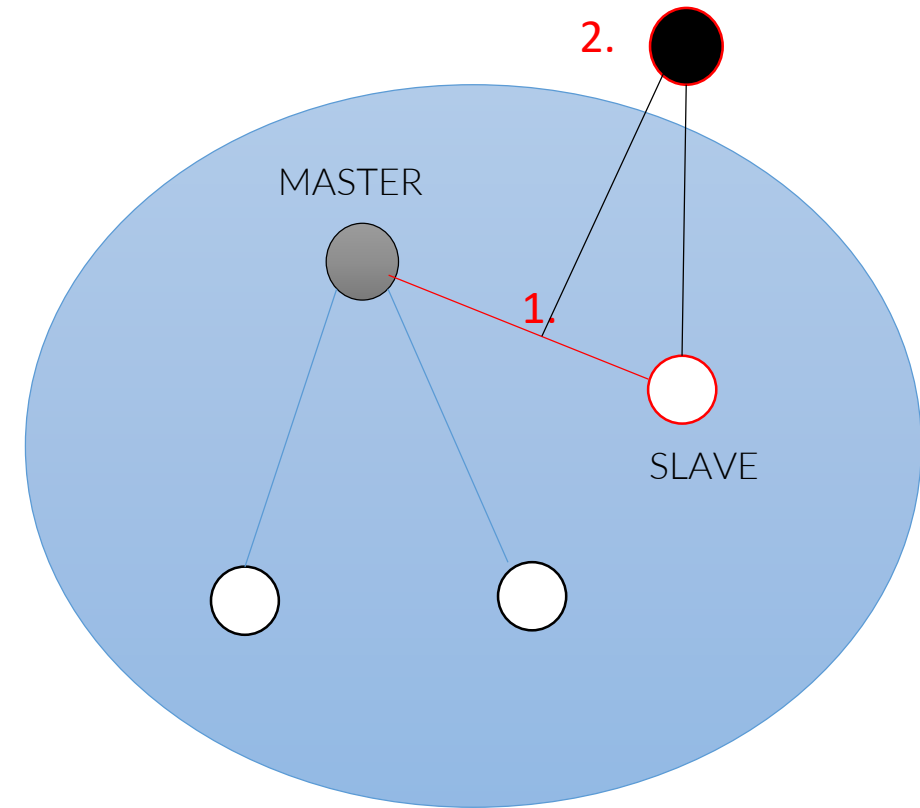
BLUETOOTH AS A COMMUNICATION CHANNEL



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

BLUETOOTH ATTACK VECTORS

- Eavesdropping/MITM
- Manipulating characteristics



Frontline BPA 600



\$20,000

Ubertooth One



\$120

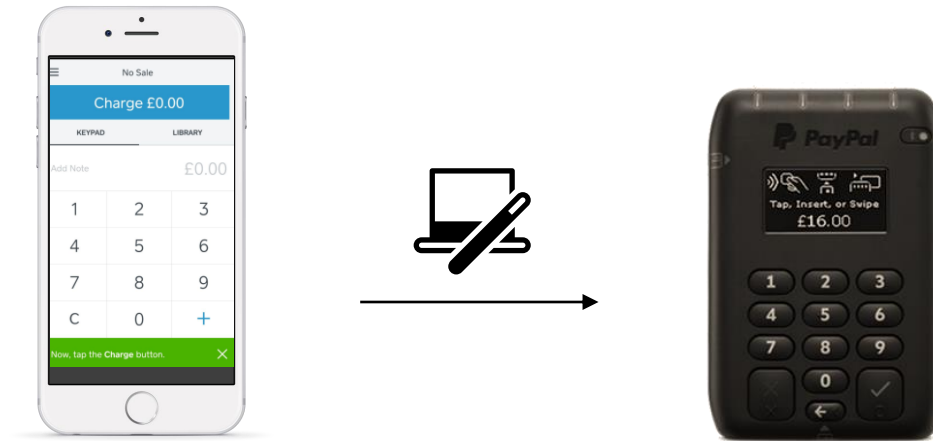

```
...1 = Packet Header and BR/EDR Payload Dewhitened: True
```

```
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



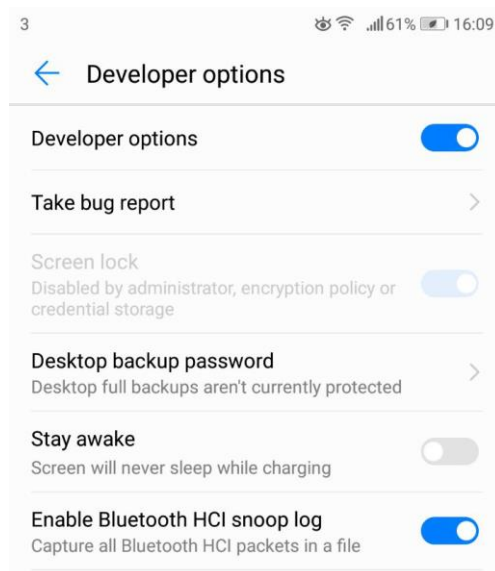
User authentication doesn't exist in the Bluetooth protocol, it must be added by the developer at the application layer

Findings

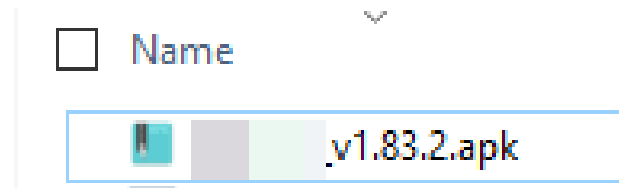
1.



2.



3.



Findings

```
localhost ()      Rcvd UIH Channel=1 UID
localhost ()      Rcvd "\030\004\001\000\000\000\035"
Datescs_0d:cc:3e Sent "\031\005\001\000\000\027\000\003\000\000\024\000Insert/swipe cardI"
host             Rcvd Number of Completed Packets
localhost ()      Rcvd UIH Channel=1 UID
localhost ()      Rcvd "\031\005\001\000\000\000\035"
controller       Sent Sniff Mode
host             Rcvd Command Status (Sniff Mode)
host             Rcvd Mode Change
```

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: Datescs_0d:cc:3e (68:aa:d2:0d:cc:3e)]

Bluetooth HCI H4

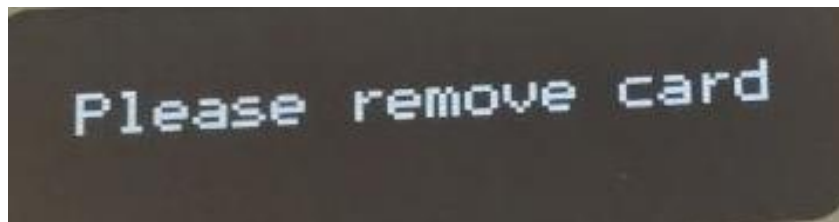
- [Direction: Sent (0x00)]
- HCI Packet Type: ACL Data (0x02)

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: Datescs_0d:cc:3e (68:aa:d2:0d:cc:3e)]

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	2fInsert/
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: 02001d06010b000000010013506c656173652072

```
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 36.187550 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

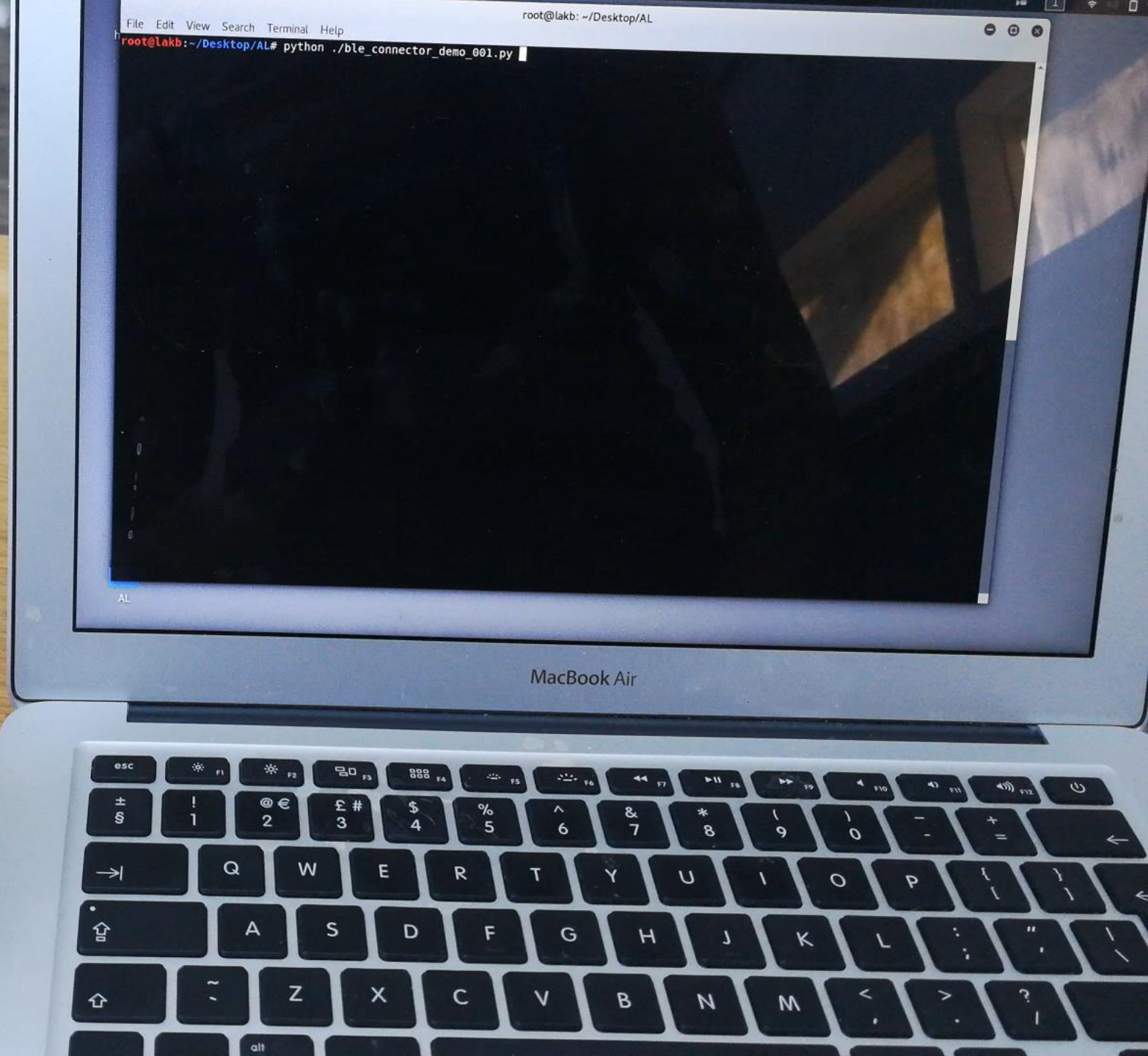
Handle: 0x001b (Unknown: Unknown)
[Service UUID: d839fc3c84dd4c369126187b07255127]
[UUID: b378db854ec34daa828e1b99607bd6a0]
Value: 656d6f7665206361726400ff083c6203

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

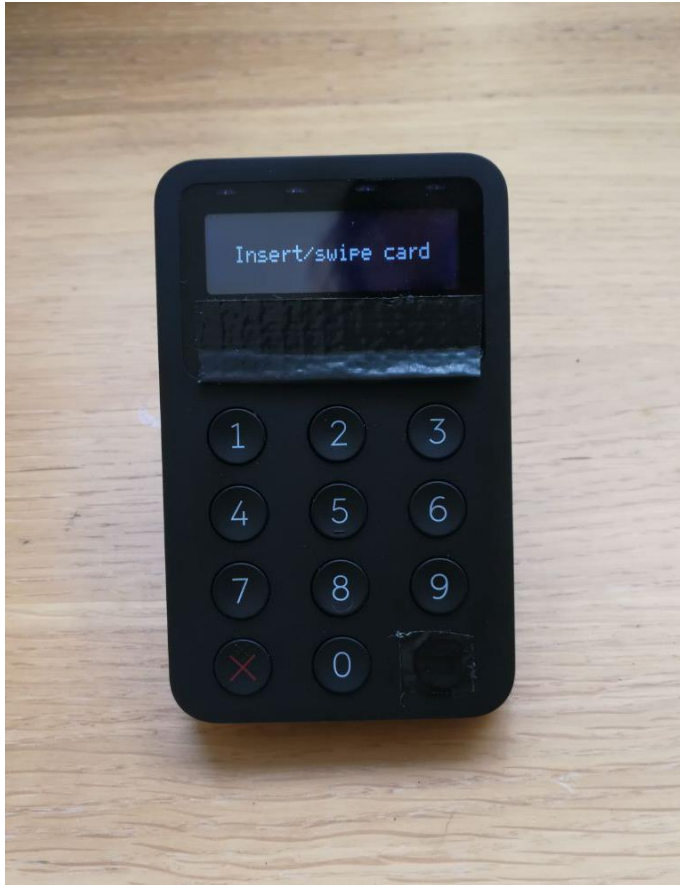
ATTACK VECTORS

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.





Findings



- ▼ Bluetooth RFCOMM Protocol
 - ▼ Address: E/A flag: 1, C/R flag: 1, Direction: 0, Channel: 1
 - ▼ 0000 10.. = DLCI: 0x02 (Direction: 0, Channel: 1)
 - 0000 1... = Channel: 1
 -0.. = Direction: 0x0
 -1. = C/R Flag: Command (0x1)
 -1 = EA Flag: Last field octet (0x1)
 - ▼ Control: Frame type: Unnumbered Information with Header check (UIH) (0xef), P/F flag: 0
 - ...0 = P/F flag: 0x0
 - 111. 1111 = Frame type: Unnumbered Information with Header check (UIH) (0xef)
 - Payload length: 32
 - Frame Check Sequence: 0x9a
- ▼ Bluetooth SPP Packet
 - Data: 0d0501000017010300000c00496e736572742f7377697065...

Data: 0d0501000017010300000c00496e736572742f73776970652063617264440d0a

LEADING PART	MESSAGE	CRC
0d0501000017	010300000c00496e736572742f737769706520636172	44
	64	
	“Insert/swipe card”	



AMOUNT TAMPERING

HOW TO GET ACCESS TO TRANSACTIONS AND COMMANDS

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

HOW TO GET ACCESS TO COMMANDS

1. 0x02ee = 7.50 USD 0x64cb = checksum

```
> Bluetooth L2CAP Protocol
v Bluetooth Attribute Protocol
```

V/ (10152): (SourceFile:31)@BtSmart-Receiver | Message length pa
D/ (10152): (SourceFile:31)@BtSmart-Receiver | Message complete,
02ee
64cb Remaining bytes:

0000	00	02R...	0.
0010	02	ee	..<...)	

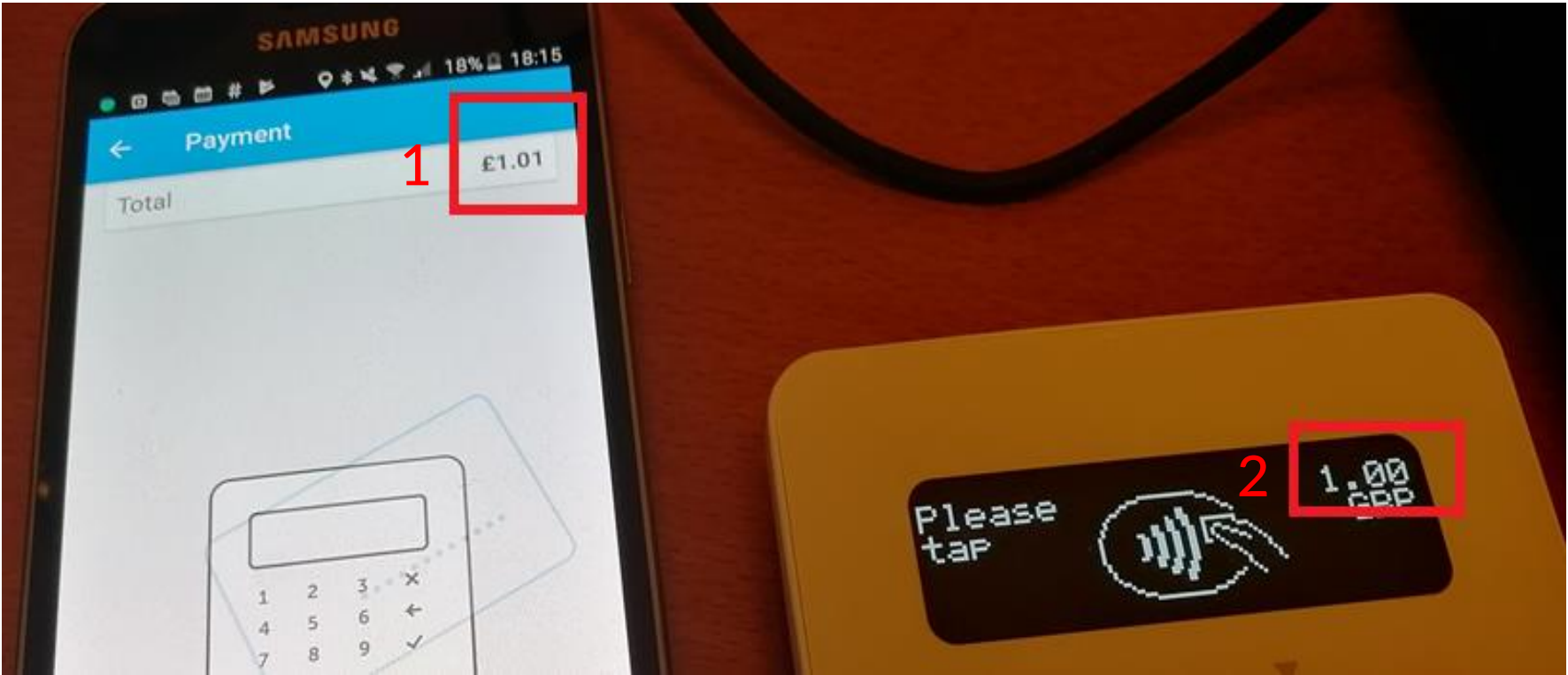
2. 0100 = 1.00 USD 0x8a = checksum

```
J config
J config

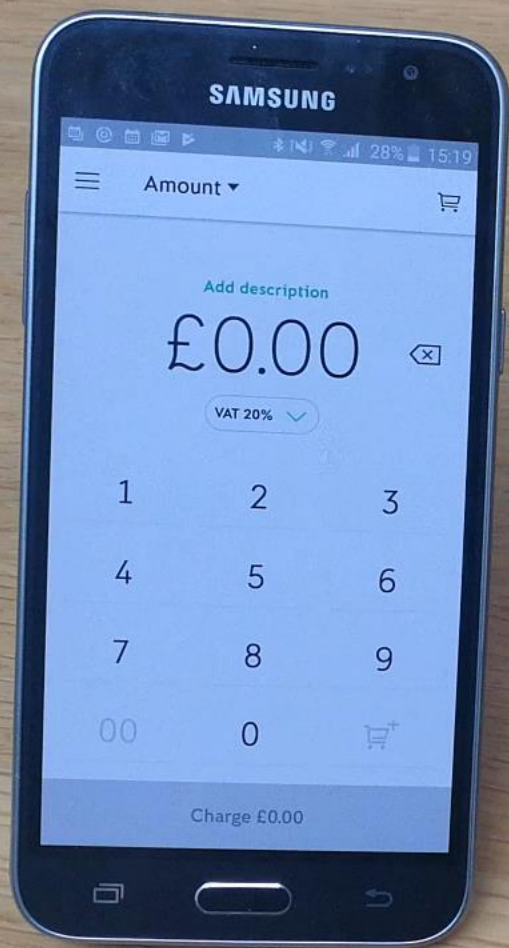
}}
),
31 D/DatecsReader(17527): Wrote to BT:
32 0206000000000100 8A with timeout 5000
33 I/DatecsReader(17527): Writing command to reader:
"INIT_TRANSACTION": {
  "COMMANDS": [{
    "HEX": "9F0206[AMOUNT]9A"
    "PARAMETERS": {
      "AMOUNT": {
        "FIXED_LENGTH": 12
      }
    }
  }
}
```


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

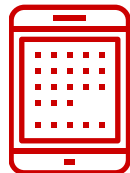
ATTACK



Customer



\$1.00
payment



\$1.00
payment



Fraudulent merchant



50,000 payment



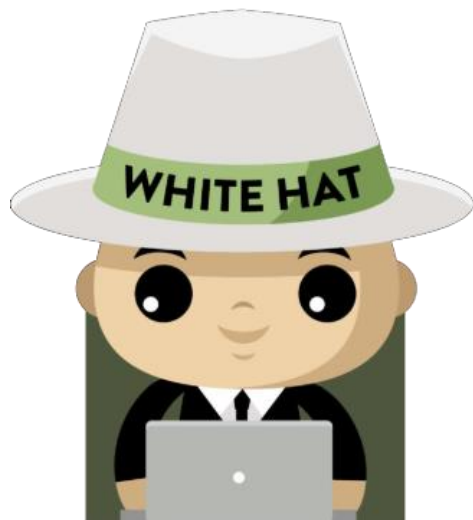
Service Provider

MITIGATION ACTIONS FOR SERVICE PROVIDERS

- REQUEST SOLUTION FROM VENDOR
- CONTROL YOUR ECOSYSTEM
- NO MAG-STRIPE

REMOTE CODE EXECUTION

RCE = 1 REVERSE ENGINEER + 1 FIRMWARE



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.sig`

`https://frw.*****.com/_prod_app_1_0_1_4.bin`

`https://frw.*****.com/_prod_app_1_0_1_4.sig`

Header - RSA-2048 signature (*0x00 - 0x100*)

Body - AES-ECB encrypted

HOW FIRMWARE ARRIVES ON THE READER

"paypalobjects" mpi tar.gz



All

Videos

News

Shopping

Images

More

Settings

About 40 results (0.33 seconds)

arun-paypal-issue/paypal log at master · arunjnair15/arun-paypal ...

<https://github.com/arunjnair15/arun-paypal-issue/blob/master/paypal%20log> ▼

11 Jul 2017 - "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-V1-39.tar.gz

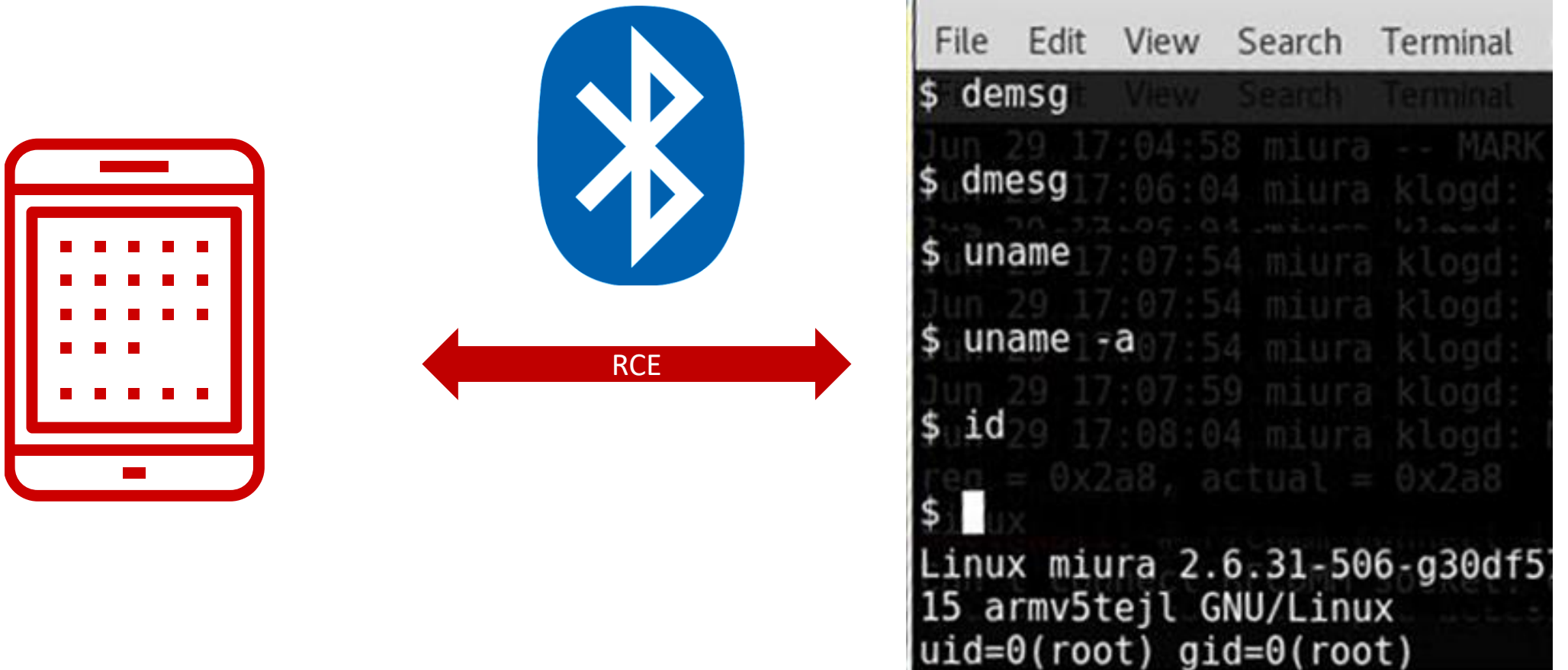
HOW FIRMWARE ARRIVES ON THE READER

```
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
```

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

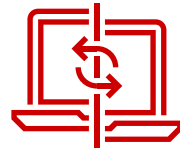
HOW FIRMWARE ARRIVES ON THE READER



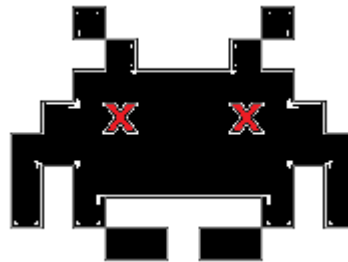
INFECTED MPOS

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

DEVICE PERSISTENCE

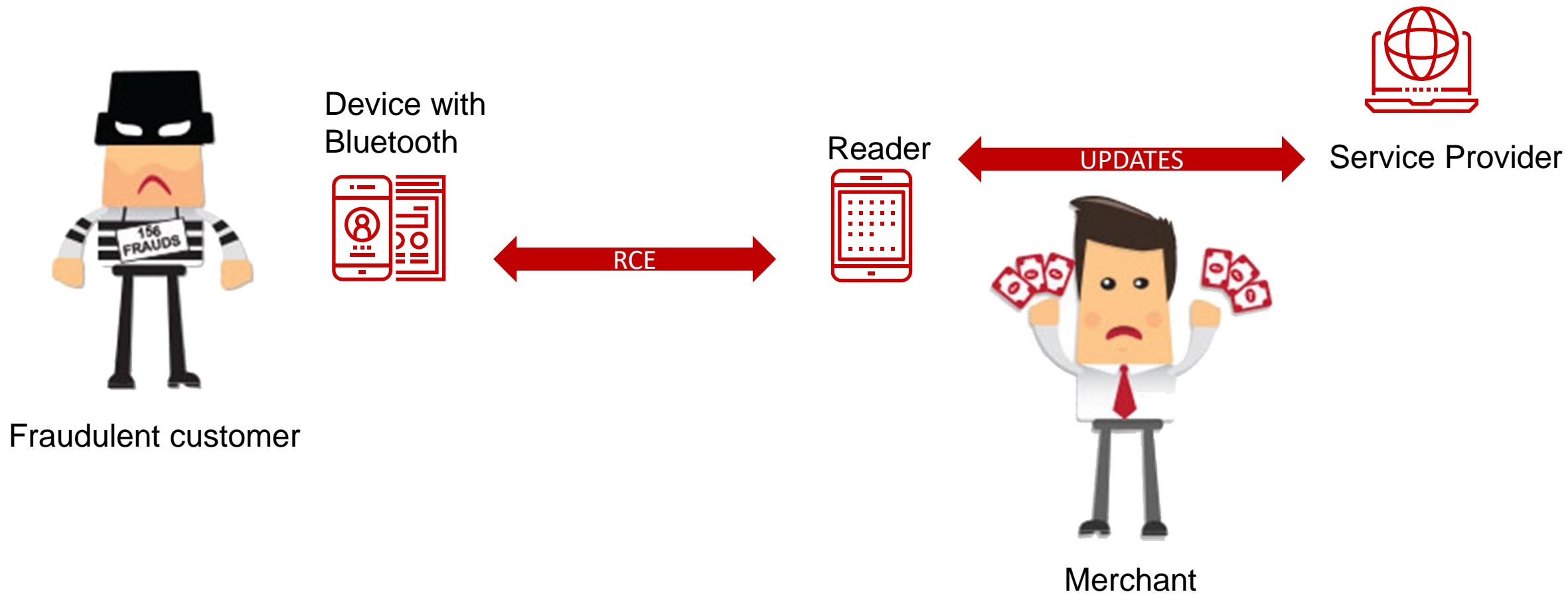


REBOOT



GAME OVER

ATTACK

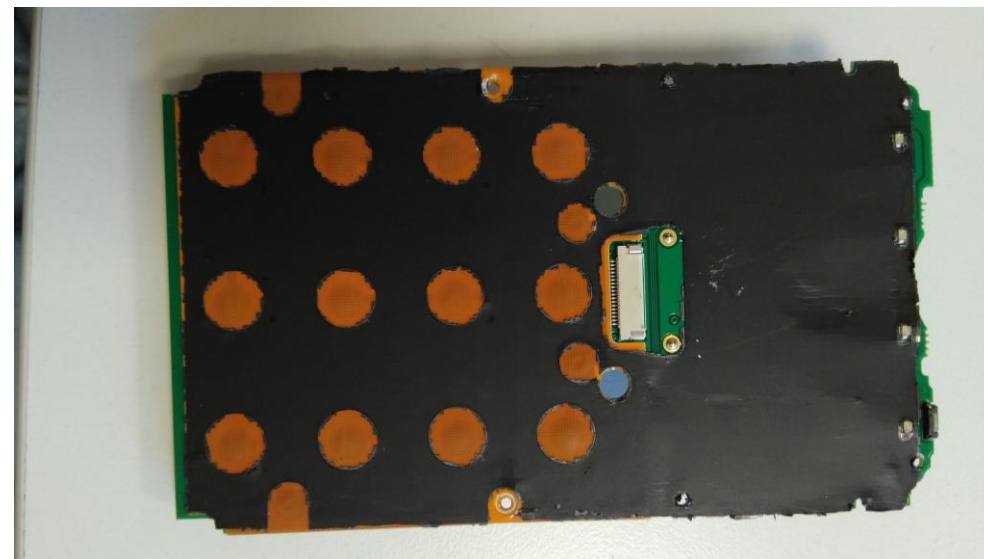


MITIGATIONS

- NO VULNERABLE OR OUT-OF-DATE
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

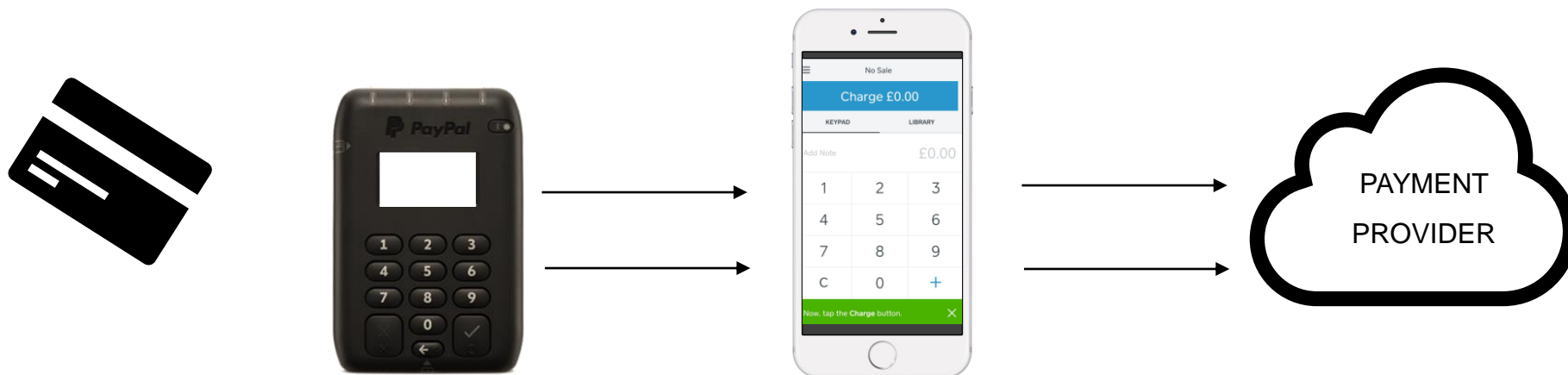


Conclusions

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%	Low - no anti money laundering checks but some ID checks	Strict – active monitoring of transactions	N/A	-	strict	-	-	-
Square [USA]	\$50 2.5-2.75%		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	\$40 1.69%			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

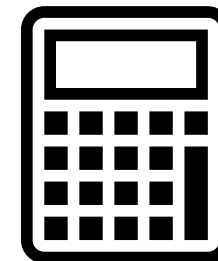
RECOMMENDATIONS FOR MPOS MANUFACTURERS



- 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 mag-stripe
- 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

Hardware and firmware:
Artem Ivachev

Tim Yunusov



@a66at

Hardware observations:
Alexey Stennikov
Maxim Goryachy
Mark Carney