|=---=|
|=----------------------=[EMV - Chip & PIN   ]=------------------=|
|=----------------------=[CVM Downgrade Attack   ]=------------------=|
|=---=|
|=---=|
|=-----------------=[Aperture Labs   ]=--------------=|
|=-----------------=[   ]=--------------=|
|=-----------------=[Adam Laurie   ]=--------------=|
|=-----------------=[<adam_at_aperturelabs_dot_com>]=--------------=|
|=-----------------=[   ]=--------------=|
|=-----------------=[Zac Franken   ]=--------------=|
|=-----------------=[<zac_at_aperturelabs_dot_com>   ]=--------------=|
|=-----------------=[   ]=--------------=|
|=-----------------=[Inverse Path   ]=--------------=|
|=-----------------=[   ]=--------------=|
|=-----------------=[Andrea "lcars" Barisani   ]=--------------=|
|=-----------------=[<lcars_at_inversepath_dot_com>]=--------------=|
|=-----------------=[   ]=--------------=|
|=-----------------=[Daniele "danbia" Bianco   ]=--------------=|
|=-----------------=[<danbia_at_inversepath_dot_com>]=--------------=|
|=---=|

--[Contents

--[1. Introduction

This whitepaper details the CVM downgrade attack presented in our "Chip & PIN is definitely broken" presentation [1].

The technique aims to expose the ineffectiveness of the EMV standard in protecting the cardholder PIN during transactions which employ the credit card smartcard and consequently the EMV protocol.

EMV stands for Europay, MasterCard and VISA, the global standard for inter-operation of integrated circuit cards (IC cards or integrated chip cards) and IC card capable point of sale (POS) terminals and automated teller machines (ATMs), for authenticating credit and debit card transactions.

IC card systems based on EMV are being phased in across the world, under names such as "IC Credit" and "Chip and PIN".

--[2. Motivation

The implementation of smartcard technology and PIN (Personal Identification Number) in credit/debit card transactions, while instrumental in the attempt [2] to phase out old and insecure magnetic stripe technology, raises important questions about potential liability shift to the cardholder when a legitimate

card and PIN are used.

The liability in case of fraud shifts away from the merchant to the bank in most cases (though if merchant does not roll EMV then liability explicitly shifts to it), however cardholders are assumed to be liable unless they can unquestionably prove they were not present for the transaction, did not authorize the transaction, and did not inadvertently assist the transaction through PIN disclosure.

PIN verification, with the help of EMV, increasingly becomes "proof" of cardholder presence [3], therefore it is absolutely important to fully understand if the PIN is sufficiently protected.

--[3. EMV Skimmer

The chip interface is inherently accessible and not protected by tamper proofing sensors, it is therefore an extremely appealing target to fraudsters.

It is nearly impossible for the cardholder (or merchant) to easily verify that the terminal has been tampered and for this reason an EMV skimmer could go undetected for a very long time.

The installation effort required is minimal as the installation can be performed by "hooking" the skimmer inside the chip interface with a credit card sized tool. Such device requires relatively short development effort and cheap components.

The skimmer can be powered by the POS itself (via the smartcard interface) and act as a man-in-the middle between the card/terminal conversation, intercepting and potentially modifying the data exchanged during the transaction. The data can be stored on a memory support and later downloaded via RF, infrared or using a special chip card recognized by the skimmer.

This threat is real and far from being theoretical as chip skimmers installations dated 2008 have been reported in the wild by law enforcement authorities after our presentation was made available.

--[4. Offline Data Authentication

The information read by the terminal from the card is stored with BER-TLV templates and contains information such as the Application PAN (the credit card number), the cardholder name, the expiration date and so on.

The data is passed in the clear between the terminal and card interface, however selected objects are signed and they take part in the Offline Data Authentication process.

Depending on the chip technology three methods are available:

```
    SDA | Static Data Authentication
    DDA | Dynamic Data Authentication
    CDA | Combined Data Authentication
```

The SDA method represents the cheapest and most widely used technology. With SDA selected records are signed with a static signature which can be verified by the terminal. SDA cards supports only offline PIN verification, where the PIN is passed in cleartext from the terminal to the card, which then verifies the correctness of the PIN.

The DDA method requires more expensive smartcard technology and is slowly

replacing SDA [4]. As the name suggests DDA employs a dynamic signature computed against static data and a random number passed by the terminal. DDA cards supports cleartext PIN verification as well as enciphered PIN verification, the latter method prevents cleartext transmission of the PIN from the terminal to the card.

The CDA method combines the dynamic signature verification with the transaction cryptogram generation, in order to ensure that the card verified by the Offline Data Authentication phase is the same used for the actual transaction. We have not been able to find any CDA cards in the wild as of 07/2011.

--[5. Cardholder Verification

The transaction flow is mainly composed by the following 5 phases:

```
1 | initiate application processing
2 | read application data
3 | offline data authentication (if indicated in the AIP)
4 | cardholder verification (if indicated in the AIP)
5 | issuer script processing
```

The Cardholder Verification phase is the target of our attack as it is responsible for the actual PIN verification.

In the EMV protocol the card itself advertises to the terminal the Cardholder Verification Method preference using the CVM List object (tag 8E). The following methods can be specified in the CVM List:

```
Plaintext PIN verification performed by ICC
Enciphered PIN verified online
Plaintext PIN verification by ICC and signature (paper)
Enciphered PIN verification by ICC
Enciphered PIN verification by ICC and signature (paper)
Signature (paper)
No CVM required
```

The CVM List is, nowadays, signed on all cards and it takes part in the Offline Data Authentication. It is therefore believed that the CVM list is tamper proof and that only when the 'Plaintext PIN verification performed by ICC' method is present and selected by the terminal the PIN can be harvested by an EMV skimmer.

--[6. Action Codes

The Action Codes are data elements used to specify policies for accepting or rejecting transactions, there are two types of Action Codes: Issuer Action Codes (published by the card) and Terminal Action Codes (set by the terminal).

The Issuer Action Codes and Terminal Action Codes are OR'ed together when applied. Additionally there are three flavours of Action Codes: Denial, Online and Default. The Online Action Codes specify which failure conditions trigger online transactions.

The following is an example of configured Action Codes on a sample credit card:

```
9f0e | Issuer Action Code - Denial  (5 bytes): 00 00 00 00 00
9f0f | Issuer Action Code - Online  (5 bytes): f0 78 fc f8 00
9f0d | Issuer Action Code - Default (5 bytes): f0 78 fc a0 00
```

In the example the translated Online Action Code reads "do not deny a

transaction without attempting to go online, if offline SDA fails transmit the transaction online".

--[7. CVM Downgrade Attack

In all tested POS terminal / card combinations we were able to manipulate the Action Codes (when necessary) so that tampering with the CVM List would not result in offline rejection.

The modified CVM List is honoured by the terminal allowing an attacker to present 'Plaintext PIN verification performed by ICC' as the preferred method.

This allows PIN harvesting with SDA and DDA cards, despite the original CVM List configuration.

--[I. FAQ

1. Where are the pretty pictures?

Check the links section down below for the full pdf presentation with all the pictures.

2. Is it possible for the backend to detect the CVM downgrade attack ?

The CVM List tampering results in flipping of the 'SDA failed' status bit presented by the terminal to the backend in the TVR (Terminal Verification Results), however we do not feel realistic for an issuer to block transactions/cards solely on this information as Offline Data Authentication can fail for several legitimate reasons.

2. Does using a DDA card prevent the attack ?

No, DDA cards are required to support SDA functionality and do not prevent the downgrade.

3. Does using a CDA card prevent the attack ?

Despite the lack of testing due to the fact that there are no CDA cards available in the wild we do not think CDA cards prevent the attack.

4. Is it possible to fix this issue?

While we would rather see EMV being replaced with a simpler, more robust protocol, it is indeed possible to patch the issue.

A patch would require disabling plaintext PIN verification on POS and ATM firmwares preventing the downgrade attack in the first place, this of course would break compatibility with the EMV specification and prevent transactions with SDA cards on terminals that do not have on-line PIN verification capabilities.

--[II. References & Notes

[1] - http://dev.inversepath.com/download/emv/emv_2011.pdf
      http://www.aperturelabs.com/download/emv/emv_2011.pdf

[2] - As of 07/2011 magstripe fallback is still accepted pretty much everywhere

[3] - The Globe and Mail, 14 Jun 2011. Canadian Imperial Bank of Commerce

(CIBC) spokesman Rob McLeod said in relation to a $81,276 fraud case: "our records show that this was a chip-and-PIN transaction. This means [the customer] personal card and personal PIN number were used in carrying out this transaction. As a result, [the customer] is liable for the transaction."

[4] - In 2011 we still witness release of SDA cards with expiration set to 2015.

--[III. Links

- Project directory
  http://dev.inversepath.com/download/emv
  http://www.aperturelabs.com/download/emv

|=[EOF]=---=|