Practical exploitation of rounding vulnerabilities in internet banking applications

Adrian Furtună, PhD, OSCP, CEH
adif2k8@gmail.com
Agenda

- Who am I
- Rounding vulnerabilities
- How to fix
- Exploitation techniques
- Digipass automation
- Demo
Who am I

- PhD in Information Security, OSCP, CEH
- Penetration tester at KPMG Romania
  - Web applications, internet banking
  - Network infrastructures
  - Mobile applications
  - Source code reviews
  - + some annoying stuff
- Teaching assistant at Information Security Master programs from Bucharest universities
  - Teaching penetration testing classes
  - Organizing Capture the Flag contests
- Always like to prove my point…
Rounding vulnerabilities
Rounding vulnerabilities

Real life example

- How much do you *really* pay?
Real life example

- How much do you *really* pay?
- What about: $2.85 + 3.20 = 6.05$?
Rounding vulnerabilities

Real life example

- How much do you *really* pay?
- What about: 
  \[ 2.85\$ + 3.20\$ = 6.05\$ \]
- How much does the seller win from rounding?
Real life example

- How much do you *really* pay?
- What about:
  \[ 2.85\$ + 3.20\$ = 6.05\$ \]?
- How much does the seller win from rounding?
- We are a bit vulnerable…
Rounding vulnerabilities

In Internet Banking apps

- Banks are vulnerable also
In Internet Banking apps

- Banks are vulnerable also
- Amounts are specified with two decimals:

<table>
<thead>
<tr>
<th>IBAN</th>
<th>Currency</th>
<th>Current Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Account RO60...</td>
<td>EUR</td>
<td><strong>0.67</strong></td>
</tr>
<tr>
<td>Current Account RO66...</td>
<td>RON</td>
<td><strong>49.00</strong></td>
</tr>
</tbody>
</table>
In Internet Banking apps

- Banks are vulnerable also
- Amounts are specified with two decimals:

<table>
<thead>
<tr>
<th>IBAN</th>
<th>Currency</th>
<th>Current Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO60_0210000001360445</td>
<td>EUR</td>
<td>0.67</td>
</tr>
<tr>
<td>RO66_0210000001360434</td>
<td>RON</td>
<td>49.00</td>
</tr>
</tbody>
</table>

- What happens when you transfer 8.3436 EUR to your account?
  \[\text{Amount} += 8.34 \text{ EUR} \Rightarrow \text{Bank wins} 0.0036 \text{ EUR}\]
In Internet Banking apps

- Banks are vulnerable also
- Amounts are specified with two decimals:

<table>
<thead>
<tr>
<th>IBAN</th>
<th>Currency</th>
<th>Current Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Account</td>
<td>EUR</td>
<td>0.67</td>
</tr>
<tr>
<td>Current Account</td>
<td>RON</td>
<td>49.00</td>
</tr>
</tbody>
</table>

- What happens when you transfer 8.34\textsubscript{36} EUR to your account?
  Amount += 8.34 EUR  => Bank wins 0.0036 EUR

- What happens when you transfer 8.34\textsubscript{78} EUR to your account?
  Amount += 8.35 EUR  => Bank loses 0.0022 EUR
In Internet Banking apps

- Banks are vulnerable also
- Amounts are specified with two decimals:

<table>
<thead>
<tr>
<th>IBAN</th>
<th>Currency</th>
<th>Current Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Account</td>
<td>EUR</td>
<td>0.67</td>
</tr>
<tr>
<td>Current Account</td>
<td>RON</td>
<td>49.00</td>
</tr>
</tbody>
</table>

- What happens when you transfer **8.3436** EUR to your account?
  
  \[ \text{Amount} += 8.34 \text{ EUR} \implies \text{Bank wins 0.0036 EUR} \]

- What happens when you transfer **8.3478** EUR to your account?
  
  \[ \text{Amount} += 8.35 \text{ EUR} \implies \text{Bank loses 0.0022 EUR} \]

- Max to win/lose: **0.005 EUR** / transaction
  
  Rounding is done to the closest value (two decimals)
How to always win?

- Let’s make transactions that will be always rounded in our favor
How to always win?

- Let’s make transactions that will be always rounded in our favor
- How?
  - Foreign exchange transactions
  - Transfer between your own accounts having different currencies
Rounding vulnerabilities

Obtain a better exchange rate

- Transfer money between your own accounts (e.g. RON -> EUR)
- Specify how much RON you want to sell

<table>
<thead>
<tr>
<th>RON</th>
<th>EUR</th>
<th>EUR (rounded)</th>
<th>Actual exchange rate (RON / EUR rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.40</td>
<td>1</td>
<td>1.00</td>
<td>4.40 Official</td>
</tr>
<tr>
<td>2</td>
<td>0.4545</td>
<td>0.45</td>
<td>4.44</td>
</tr>
<tr>
<td>1</td>
<td>0.2272</td>
<td>0.23</td>
<td>4.34</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1136</td>
<td>0.11</td>
<td>4.54</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0113</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td>0.03</td>
<td>0.0068</td>
<td>0.01</td>
<td>3</td>
</tr>
<tr>
<td>0.023</td>
<td>0.0052</td>
<td>0.01</td>
<td>2.3 The best</td>
</tr>
<tr>
<td>0.02</td>
<td>0.0045</td>
<td>0.00</td>
<td>not good</td>
</tr>
</tbody>
</table>

\[100 \times (0.023 \text{ RON} \rightarrow 0.01 \text{ EUR}) \Rightarrow 2.3 \text{ RON} = 1 \text{ EUR}\]
What about Rubla?

- Transfer money between your own accounts (e.g. RUB -> USD)
- Specify how much RUB you want to sell

<table>
<thead>
<tr>
<th>RUB</th>
<th>USD</th>
<th>USD (rounded)</th>
<th>Actual exchange rate (RUB / USD rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.39</td>
<td>1</td>
<td>1</td>
<td>32.39 Official</td>
</tr>
<tr>
<td>10</td>
<td>0.3087</td>
<td>0.31</td>
<td>32.25</td>
</tr>
<tr>
<td>5</td>
<td>0.1543</td>
<td>0.15</td>
<td>33.33</td>
</tr>
<tr>
<td>1</td>
<td>0.0308</td>
<td>0.03</td>
<td>33.33</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0154</td>
<td>0.02</td>
<td>25</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0061</td>
<td>0.01</td>
<td>20</td>
</tr>
<tr>
<td>0.17</td>
<td>0.0052</td>
<td>0.01</td>
<td>17 The best</td>
</tr>
<tr>
<td>0.16</td>
<td>0.0049</td>
<td>0.00</td>
<td>not good</td>
</tr>
</tbody>
</table>

100 * (0.17 RUB -> 0.01 USD) => 17 RUB = 1 USD
Rounding vulnerabilities

Compute your own exchange rate

Rounding vulnerabilities

Example (1)

- **13/02/2012**
  - Schimb valutar
  - Referinta: 7
  - Din contul: RO19
  - Suma: 0.03 RON
  - Rata: 4.40580

- **13/02/2012**
  - Schimb valutar
  - Referinta: 8
  - Din contul: RO11
  - Suma: 0.03 RON
  - Rata: 4.40580

- **13/02/2012**
  - Schimb valutar
  - Referinta: 9
  - Din contul: RO11
  - Suma: 0.02 RON
  - Rata: 4.40580

**EUR**
ÉCHANGES VALUTAIRES

<table>
<thead>
<tr>
<th>Client</th>
<th>FURTUNA CONSTANTIN-ADRIAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUI / CNP</td>
<td></td>
</tr>
<tr>
<td><strong>Compte vente:</strong></td>
<td>RO7800000001360412 (RON)</td>
</tr>
<tr>
<td><strong>Montant vendu:</strong></td>
<td>0.025 RON</td>
</tr>
<tr>
<td><strong>Compte achat:</strong></td>
<td>RO7200000001360423 (EUR)</td>
</tr>
<tr>
<td><strong>Montant acheté:</strong></td>
<td>0.01 EUR</td>
</tr>
<tr>
<td><strong>Rata de schimb</strong></td>
<td>4.4614</td>
</tr>
<tr>
<td>Date d'ordre:</td>
<td>05/06/2013</td>
</tr>
</tbody>
</table>

La transaction a été introduite le 05/06/2013 à 13:10:44 heure par CONSTANTIN-ADRIAN FURTUNA
La transaction a été signée le 05/06/2013 à 13:10:59 heure par CONSTANTIN-ADRIAN FURTUNA (E)

autorisé. Referinta bancii: 021FT24131560016
Traité avec succès!
6/15/2013 6:15:33 PM
When is the best deal

- **Foreign exchange transactions:**
  - Specify how much you want to sell  => destination will be rounded
  - Specify how much you want to buy  => source will be rounded

- **Best deal is when you can specify how much of the weaker currency you want to sell/buy because the stronger currency will be rounded**
How much can I gain?

C1 = minimum amount of currency 1 that can be exchanged (e.g. 0.023 RON)
C2 = minimum amount of currency 2 that can be exchanged (e.g. 0.01 EUR)
Ex_b = exchange rate for buying C2 with microtransactions (e.g. 2.3)

\[
\text{Ex}_b = \frac{\text{C1}}{\text{C2}}
\]

Ex_s = exchange rate for selling C2 (e.g. 4.4) – real exchange rate – fixed by the Bank

Rounding vulnerabilities

- \[
z = y \times \text{Ex}_s = \left(\frac{x}{\text{Ex}_b}\right) \times \text{Ex}_s = x \times \left(\frac{\text{Ex}_s}{\text{Ex}_b}\right)
\]
- multiplication rate = \(\frac{\text{Ex}_s}{\text{Ex}_b}\)
- transactions required = \(\frac{x}{\text{C1}}\)

<table>
<thead>
<tr>
<th>Currency</th>
<th>Multiplication rate</th>
<th>Initial amount (x)</th>
<th>Final amount (y)</th>
<th>Gain</th>
<th>Transactions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RON</td>
<td>4.4 / 2.3 = 1.9</td>
<td>100 RON</td>
<td>190 RON</td>
<td>90 RON ~ 20 EUR</td>
<td>100 / 0.023 = 4347</td>
</tr>
<tr>
<td>RUB</td>
<td>32.39 / 17 = 1.9</td>
<td>1000 RUB</td>
<td>1900 RUB</td>
<td>900 RUB ~ 27.8 USD</td>
<td>1000 / 0.17 = 5882</td>
</tr>
</tbody>
</table>
Different exchange rates (buy / sell)

- Banks have different exchange rates for buying and for selling so they can always win
- Let’s say…
  - Official exchange rate: 32.39 RUB/USD
  - You buy USD from the Bank: 38.39
  - You sell USD to the Bank: 25.12
- But for small amounts it is not true!
  - I buy from the Bank (RUB → USD)
    - 0.32 RUB / 32.39 = 0.009 USD → 0.01 USD
    - 0.32 RUB / 38.00 = 0.008 USD → 0.01 USD
    - 0.32 RUB / 42.00 = 0.007 USD → 0.01 USD
    - 0.32 RUB / 50.00 = 0.006 USD → 0.01 USD
    - 0.32 RUB / 60.00 = 0.005 USD → 0.01 USD
How to fix
How the Banks should protect themselves

- Limit the number of transactions that can be performed in a given time by a regular person
- Introduce a small fee for currency exchange operations (e.g. 0.01 EUR)
- Limit the minimum amount that can be transferred in a foreign exchange operation
- Monitor for suspicious transactions (numerous transactions, very small amounts)
- State in the contract that such transactions are illegal
Exploitation techniques
General ideas

- Find a way to do lots of transactions in a relatively short time
- Transactions are made in two steps:
  - Initialization (can be automated)
  - Authorizing / Signing (requires human interaction)
- Automate / bypass transaction signing mechanism (digipass, SMS, token, etc)
Exploitation techniques

Technique 0: No signing required 😊

- 3000 transactions, 90 minutes, 30 RON → 73 RON, gain ~10 EUR
Technique 1: Init lots and sign once

- Initiate lots of transactions automatically and sign once
Technique 1: Init lots and sign once

- Initiate lots of transactions automatically and sign once
- Use Burp Suite to initiate transactions
### Technique 1: Init lots and sign once

- Initiate lots of transactions automatically and sign once

#### Sign operations

To sign operations created prior to current date use the search filter.

<table>
<thead>
<tr>
<th>Fill-in date</th>
<th>Exchange rate</th>
<th>Sell amount</th>
<th>Status</th>
<th>Message</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>15/06/2013 14:22:35</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:34</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:34</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:33</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:33</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:33</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:33</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15/06/2013 14:22:33</td>
<td>4.4960 Without negociation</td>
<td>0.025 RON 0.01 EUR</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TRANSACTION PROCESSING**

Please input the code generated by the token

123456

1 2

**SIGN ALL SELECTED ORDERS**  **FILTER**
Technique 1: Init lots and sign once

- Initiate lots of transactions automatically and sign once
- Signing can also be automated – stay tuned for next chapter
Technique 2: Payment files

- Upload a payment file containing lots of transactions and sign once
Exploitation techniques

Technique 2: Payment files

- Upload a payment file containing lots of transactions and sign once

- Signing can also be automated – stay tuned for next chapter
Technique 3: Real time transactions + rainbow tables

- Do real time transactions automatically and sign using pre-computed digipass responses
Technique 3: Real time transactions + rainbow tables

- Do real time transactions automatically and sign using pre-computed digipass responses
- Applicable when signing is done using challenge-response mechanism, with challenge code <= 5 digits
Technique 3: Real time transactions + rainbow tables

- A challenge-response digipass returns the same response for the same challenge code every time
  \[
  \text{Response} = f(\text{challenge}, \text{timestamp}, \text{client ID}, \text{other data}) = f(\text{challenge}, \text{static data})
  \]

- Build rainbow tables with digipass responses
  - Feasible for max 5 digit challenge codes
  - Max 99999 possibilities
  - Can be automated, stay tuned
Technique 4: Real time transactions + digipass automation

- Do real time transactions automatically and sign using digipass responses computed in real time
Technique 4: Real time transactions + digipass automation

- Do real time transactions automatically and sign using digipass responses computed in real time
- Requires automation of the signing device (digipass, phone, etc)
Digipass automation
LimID project (for VASCO GO3)

- [http://limid.sitadella.com](http://limid.sitadella.com)
- Code regenerates at 30 seconds

- Video
My machine (for VASCO 550)

Requires PIN authentication

Used for:
- 2\textsuperscript{nd} factor authentication
- Transaction signing
My machine - video
My machine - current performance

- 10 transactions / minute (1 transaction / 6 seconds)
  - max 14400 transactions / day
  - enter PIN, type challenge code, read response image, do OCR

- Our previous example:
  100 RON → 190 RON (gain ~20 EUR)
  => 4347 transactions * 6 sec/trans = **26082 sec**
  = **7h:14m:42s**

  1000 RUB → 1900 RUB (gain ~27.8 USD)
  => 5882 transactions * 6 sec/trans = **35292 sec**
  = **9h:48m:10s**

- Maximum amount to multiply per day:
  14400 * 0.023 RON = 331.2 RON => final 629.28 RON
  gain 298 RON ~ = 68 EUR/day

  14400 * 0.17 RUB = 2448 RUB => final 4651 RUB
  gain 2203 RUB ~ = 68 USD/day
My machine - current performance

- What about doing in parallel (on multiple bank accounts)?
- Money making machine? 😊
External vs Internal instrumentation

- **Internal instrumentation (direct electrical connections):**
  - **Pros:**
    - more reliable and faster
    - almost error free
  - **Cons:**
    - might not be possible – some digipasses deactivate when opened
    - must know the pinout of LCD screen (lots of pins!)
    - sensitive soldering required
    - mistakes can lead to deactivation

- **External instrumentation:**
  - **Pros:**
    - No interference with digipass’s internals
    - Can be applied to any digipass model
  - **Cons:**
    - Pretty slow (but good for the “low and slow” approach)
    - Some (mechanics) errors occur on pressing buttons (resolvable by a more professional construction)
    - OCR process needs special (lighting) conditions to produce correct results
Digipass automation

My machine – implementation details (1)
### Optical Character Recognition

<table>
<thead>
<tr>
<th>Original</th>
<th>Cleared background</th>
<th>Blurred</th>
<th>Threshold applied</th>
<th>OCR-ized gocr / ocrad</th>
</tr>
</thead>
<tbody>
<tr>
<td>7169309</td>
<td>7169309</td>
<td>7169309</td>
<td>7169309</td>
<td>7169309 - _16g309</td>
</tr>
<tr>
<td>1757450</td>
<td>1757450</td>
<td>1757450</td>
<td>1757450</td>
<td>1757450 1_5_G50</td>
</tr>
<tr>
<td>0432616</td>
<td>0432616</td>
<td>0432616</td>
<td>0432616</td>
<td>043i _i_i</td>
</tr>
<tr>
<td>9236414</td>
<td>9236414</td>
<td>9236414</td>
<td>9236414</td>
<td>9a__641 4</td>
</tr>
</tbody>
</table>
Digipass automation

My machine – development stages
- Live Demo
- Q & A
- Thank you!

Adrian Furtună, PhD, OSCP, CEH
adif2k8@gmail.com
http://pentest-tools.com