A Metric for the Evaluation and Comparison of Keylogger Performance

Tobias Fiebig, Janis Danisevskis and Marta Piekarska

Technische Universität Berlin, FG Security in Telecommunications

CSET 2014, San Diego, 18th of August 2014
This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

Thistalkpresentsourworkon"AMetricfortheEvaluationandComparisonofKeyloggerPerformance": 
This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.
This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.
- How we did it.
This talk presents our work on "A Metric for the Evaluation and Comparison of Keylogger Performance":

- Why we did it.
- How we did it.
- And how *you* can contribute.
So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...
So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...

...and wanted to compare it's performance with other approaches.
So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...

...and wanted to compare it's performance with other approaches.

However, neither were a lot of these other keyloggers available as source, nor were the performance metrics applicable and/or portable.
So in the beginning... we had this new vector for a kind of heuristic, not 100% precise keylogger...

...and wanted to compare it's performance with other approaches.

However, neither were a lot of these other keyloggers available as source, nor were the performance metrics applicable and/or portable.

Solution: Build our own.
A method to evaluate a keylogger's performance.
- A method to evaluate a keylogger's performance.

- Thorough description of how to apply this method.
• A method to evaluate a keylogger's performance.
• Thorough description of how to apply this method.
• Demo code. :-(
Contribution

- A method to evaluate a keylogger's performance.
- Thorough description of how to apply this method.
- Demo code. :-) 
- Hopefully starting a discussion on reproduceability and comparability in scientific publications.
Design

- Found an issue in a smartphone GPU library.
- Found an issue in a smartphone GPU library.

- Can read bitmaps attached to the GPU as unprivileged process.
Found an issue in a smartphone GPU library.

Can read bitmaps attached to the GPU as unprivileged process.

Figure: Tooltips are such bitmaps.
Figure: How the keylogger works.
Before heuristic keyloggers: We got out what we put in.
Before heuristic keyloggers: We got out what we put in.

How long it takes to recover passwords [Lin et al., 2014].
Before heuristic keyloggers: We got out what we put in.

How long it takes to recover passwords [Lin et al., 2014].

Probability based metric, mostly cornered around this approach; omitting checking what the users actually typed [Simon and Anderson, 2013].
Before heuristic keyloggers: We got out what we put in.

How long it takes to recover passwords [Lin et al., 2014].

Probability based metric, mostly cornered around this approach; omitting checking what the users actually typed [Simon and Anderson, 2013].

Using metrics designed for a different purpose [Xu et al., 2013].

Counts the number of operations (insert, delete, substitute and transpose) needed to change \textit{string}_1 to \textit{string}_2.

Counts the number of operations (insert, delete, substitute and transpose) needed to change \textit{string}_1 to \textit{string}_2.

Simplified: $dl(str_1, str_2) = \sum o_d + \sum o_i + \sum o_s + \sum o_t$
Issue: Gets naturally higher, the longer the input is.
Challenges with this metric.

**Issue:** Gets naturally higher, the longer the input is.

**Solution:** Divide by the length of one (defined) input string:

\[
dl(\text{real}.\text{captured}) \over |\text{characters(captured)}| \]
Challenges with this metric.

**Issue:** Gets naturally higher, the longer the input is.

**Solution:** Divide by the length of one (defined) input string:
\[
dl \left( \frac{\text{real} \cdot \text{captured}}{|\text{characters} \cdot \text{captured}|} \right)
\]

**Issue:** Now lower means better. This is counter intuitive.
Challenges with this metric.

**Issue:** Gets naturally higher, the longer the input is.

**Solution:** Divide by the length of one (defined) input string:

\[
\frac{dl(\text{real, captured})}{|\text{characters(captured)}|}
\]

**Issue:** Now lower means better. This is counter intuitive.

**Solution:** Subtract from 1, higher becomes better:

\[
1 - \frac{dl(\text{real, captured})}{|\text{characters(captured)}|}
\]
Reproduceability. Others should be able to do the same things we did, ideally with their own keylogger.

⇒ Thoroughly document what and why we do it; share our code.
Comparability. When others do what we did, they should be able to compare their results to our's.

Test if this is actually the case by comparing two keyloggers with each other.

Figure: "Apple and Orange - they do not compare" by Michael Johnson
Experimental Setting - Requirements and Solutions.

- Reliability. We do not want external effects influencing our study, e.g. users miss-typing things.
  ⇒ We record what each user types.
Experimental Setting - Requirements and Solutions.

- Applicability. A keylogger study using a few hundred A's as input will probably have rather good results.

⇒ Create a text including passwords and "stuff people normally write".

**Figure:** Justin Baeder: https://www.flickr.com/photos/justinbaeder/5317820857/
As mentioned earlier, getting one's hands on a keylogger is kind of hard. So we simulate two distinct keyloggers:

- Subjects exhibit a different typing speed due to the keyboard size.
- Implementation slightly differs between devices.
- The tablet has more CPU power.

<table>
<thead>
<tr>
<th>Device</th>
<th>Android Version</th>
<th>Screen Size</th>
<th>PPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone</td>
<td>4.0.4</td>
<td>4.8&quot;</td>
<td>306ppi</td>
</tr>
<tr>
<td>Tablet</td>
<td>4.1.2</td>
<td>10.1&quot;</td>
<td>149ppi</td>
</tr>
</tbody>
</table>
Results - Overview

Relative Detection Rate per User
Unfiltered Data

- Phone
- Tablet

Subject Number
Results - Typing speed Phone

Relative Detection Rate vs. Keypresses/Second Phone
Unfiltered Data

\[ y = -0.0657x + 0.8898 \]
Relative Detection Rate vs. Keypresses/Second Tablet
Unfiltered Data

- \( y = -0.0685 \times + 0.6578 \)
Apparently the keylogger works better on the phone than on the tablet.
Results - What we learned.

- Apparently the keylogger works better on the phone than on the tablet.

- The curves look like there is a connection between typing speed and detection rate, but the data does not support this.
Results - What we learned.

<table>
<thead>
<tr>
<th>Recorded:</th>
<th>Thhisisademotexxt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual:</td>
<td>This is a demo text.</td>
</tr>
</tbody>
</table>
The analysis indicated some bias for the tablet implementation.
The analysis indicated some bias for the tablet implementation.

Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.
Filtering

- The analysis indicated some bias for the tablet implementation.

- Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.

- Hence we filter the data. That is, we clean it of those effects introduced by the detected bias.
Filtering

- The analysis indicated some bias for the tablet implementation.

- Closer investigation revealed: Due to the stronger CPU the keylogger oversamples. Furthermore the space key is not recorded.

- Hence we filter the data. That is, we clean it of those effects introduced by the detected bias.

- In the case at hand this means the removal of all repeated characters and spaces.
### Filtering - Applied

<table>
<thead>
<tr>
<th>Unfiltered:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thhis isademotexxt.</td>
</tr>
<tr>
<td>Actual:</td>
</tr>
<tr>
<td>This is a demo text.</td>
</tr>
<tr>
<td>Filtered:</td>
</tr>
<tr>
<td>Thsisademotext.</td>
</tr>
<tr>
<td>Actual Filtered:</td>
</tr>
<tr>
<td>Thsisademotext.</td>
</tr>
</tbody>
</table>
Results Filtered - Overview

Relative Detection Rate per User
Filtered Data

Subject Number

Phone
Tablet
Results Filtered - Typing speed Phone

Relative Detection Rate vs. Keypresses/Second Phone
Filtered Data

$y = -0.0338x + 0.9151$
Results Filtered - Typing speed Tablet

![Graph showing relative detection rate vs. keypresses per second for Tablet Filtered Data]

Relative Detection Rate vs. Keypresses/Second Tablet
Filtered Data

- $y = -0.2428 \times + 1.1080$
The keylogger on the tablet seems to be correlated to the user's typing speed.
The keylogger on the tablet seems to be correlated to the user's typing speed.

The good performance on the filtered dataset indicates, that the identified issues are actually the cause of the overall worse performance.
We could successfully compare two keylogger implementations.
We could successfully compare two keylogger implementations.

The recorded data allowed us to identify weak-spots in one implementation.
On comparing Keyloggers...

- We could successfully compare two keylogger implementations.

- The recorded data allowed us to identify weak-spots in one implementation.

- With this data we could also show that the implemented keylogger performs reasonably well.
Points for further discussion...

- There are many open challenges:
There are many open challenges:
- Semantic Properties should be incorporated into the metric.
Points for further discussion...

- There are many open challenges:
  - Semantic Properties should be incorporated into the metric.
  - In the same manner the performance footprint of a keylogger should be incorporated as well.
Points for further discussion...

There are many open challenges:

- Semantic Properties should be incorporated into the metric.

- In the same manner the performance footprint of a keylogger should be incorporated as well.

- A database with evaluations for various presented keyloggers should be created for benchmarking purposes.

- The keylogger-testing-app should be made more plug-able.
There are many open challenges:

- Semantic Properties should be incorporated into the metric.
- In the same manner the performance footprint of a keylogger should be incorporated as well.
- A database with evaluations for various presented keyloggers should be created for benchmarking purposes.
- The keylogger-testing-app should be made more plug-able.

Relevant code and data analysis tools are publicly available at https://gitlab.sec.t-labs.tu-berlin.de/mobile-keylogger/metric


This is a demo text. The demo text includes 23 passwords. Like this one 4%23$awD or this one 98%=!awdD. Just kidding. It is only two passwords. Is it not? Please add a new line with something you would consider a secure password. No need to remember it.