ufr d'**IEEA**
Formations en
Informatique de
Lille 1

Study of the security of an innovative authentication system

Etudiant
Selim Lakhdar

Encadrants
Jean-Marie Place
Gilles Grimaud

Introduction

With the growth of cyber threats and the emergence of new more sophisticated attacks, many countermeasure are employed to mitigate those risks. Therefore, malware infection and the number of web bots are increasing over the years.

Botnets are the major threat of the last decade, with the infamous Zeus trojan bank, and his multiple variants; SpyEye and Citadel. Those malwares are very modular and can be used in many way to make profit for the bot-herder. In fact, the infected computers can be used for a large DDOS attack which can be led for disturbing financial activity of companies, or mining cryptocurrencies. Therefore, Zeus and his variants are more specified in targeting online banking credentials. Its employs several techniques like the Man In The Browser Attack which consist on injecting code directly into the browser to steal user information. Over the years, new techniques are employed like stealing password from virtual keyboards by taking screenshots at every mouse click event.

In an another context, the constant increase of internet users raises new majors security concerns. The multiplication of online web bots which are used to create bulk mail account, or to fake the popularity of social profil to make profit, are increasingly used.

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) or more generally named Human Interaction Proof, are Reverse Turing Tests which are used to distinguish between humans and web bots. Indeed, those systems are used to mitigate the risk of automated attacks. They have to be easily solvable by humans while remaining hard for computer bots. It's an eternal tradeoff between security and accessibility.

2. Context

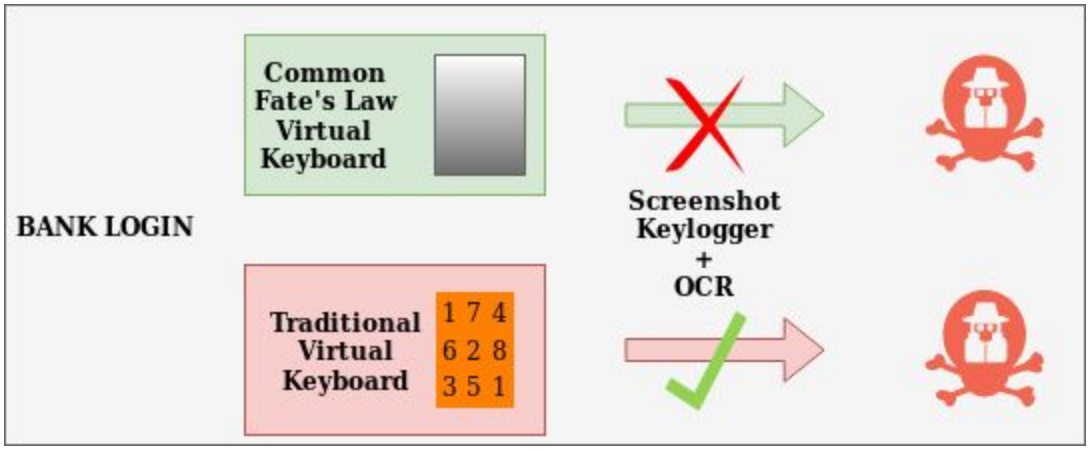


Figure 1

The 2XS team, presented a novel countermeasure to the screenshot-based keylogging attack. They described a new technique to create virtual keyboards using principles derived from the Gestalt psychology that displays informations that can be seen by humans but that cannot be recognised by modern screenshot-based keylogging methods. Indeed, Gestalt psychology or gestaltism is a psychology, philosophy and biological theory which originated in Germany in the 20's. It states that the human's process of perception and mental representation spontaneously perceives a global whole and not a juxtaposition of elements. This theory is based on 8 principles called "gestalt laws". They specifically use one of those laws: the law of common fate which states that elements moving in the same directions are perceived as belonging to the same form.

2.1. Related Work

This document follow the work of *Bacara et al* and *Echallier et al* for designing a such system. The VK is an animation of different textures randomly chosen to represent a preselected set of symbols. They concluded their work by opening question about the evaluation of the security of their VK.

2.2. Problem definition

The constant emergence of new malware with more elaborated threats push us to test the VK against basic attacks and try to break the system. The final result have to be clear and understandable enough by OCRs to automotize the attack.

Moreover, the stealthy aspect must not be neglected as we are in a restricted time/memory context. In fact, as our attacks relies on image processing, which consumes a lot of CPU time and it's easily observable with traditional monitors. The volume of data that will be processed is our Achilles heel.

3. Approach

For our analysis to reveal the numbers we had two approaches. We tried a spatial and a temporal analysis. We tried three applications over the pixels :

- Taking mean value
- Take median value
- Calculate the standard deviation

Those applications are relying on neighbours pixels. As our approach is more experimental, we tried different configuration for neighbour selection *(see Figure 1).*

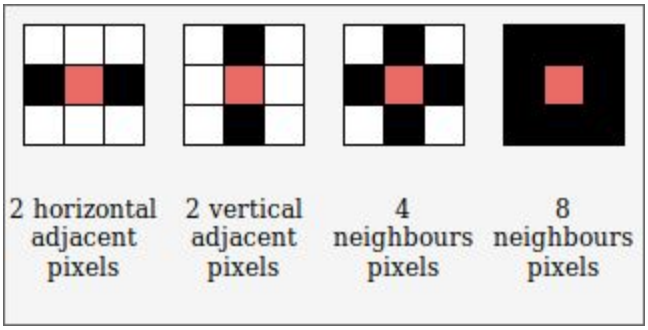


Figure 2 : Neighbors Selection

During our tests we kept in mind to improve our attack to be the more stealthiest possible. We tried to decrease the work that have to be done by :

- Skipping some pixels.
- Processing blocks of pixels.
- Checkerboard iteration.
- Skipping image(s) for the temporal analysis.

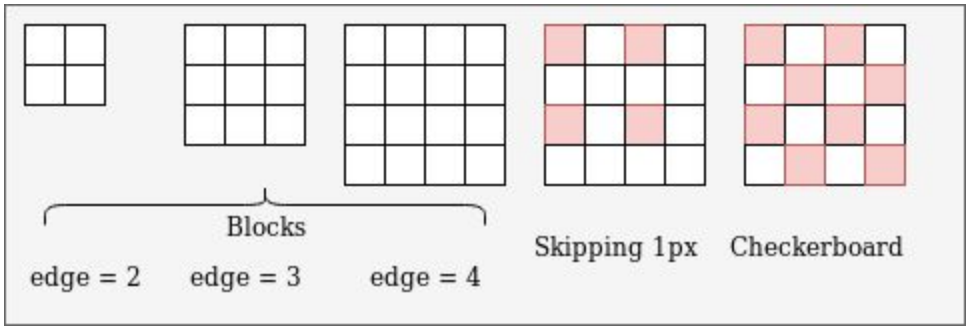


Figure 3 : Pixel/Block selection/iteration

To test the effectiveness of our different attacks, we made a numerical approach between the expected result and ours. We calculated the number of right positioned pixels, for the background and the symbols. We tried to increase that metric, while reducing the background noise.

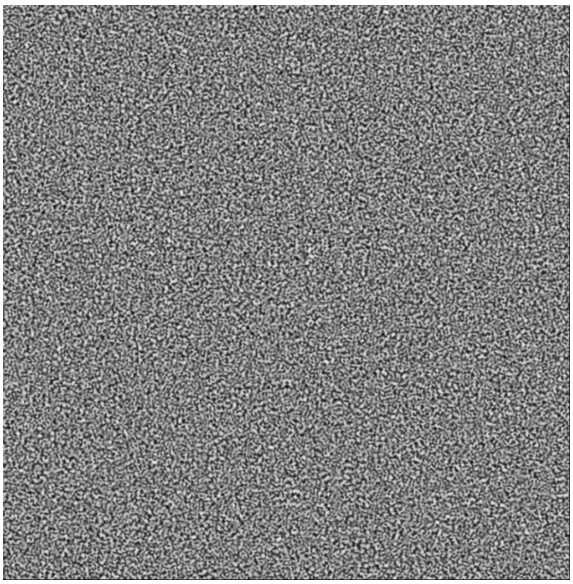
4. Steps

4.1.1 Spatial Analysis

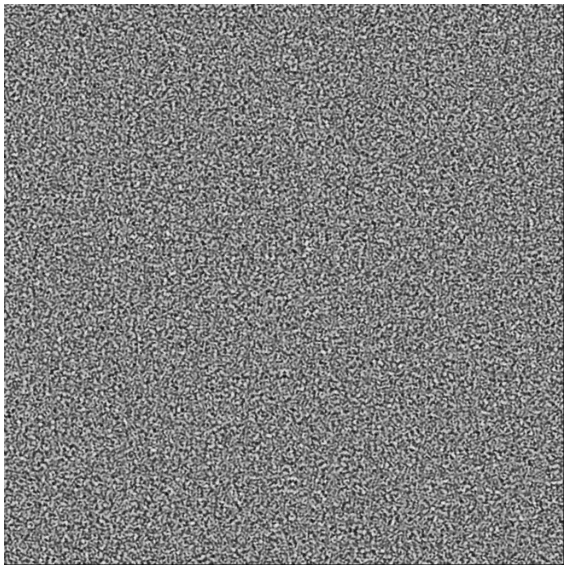
4.1.1.1 Observations

This analysis consist on working on a single screenshot. As we presented in our approach we applied different neighbour selection for our calculation. The application of the mean and median was not probant. The noise is still present and the numbers aren't visible. The calculus of standard deviation results to a dark background with an abnormal texture.

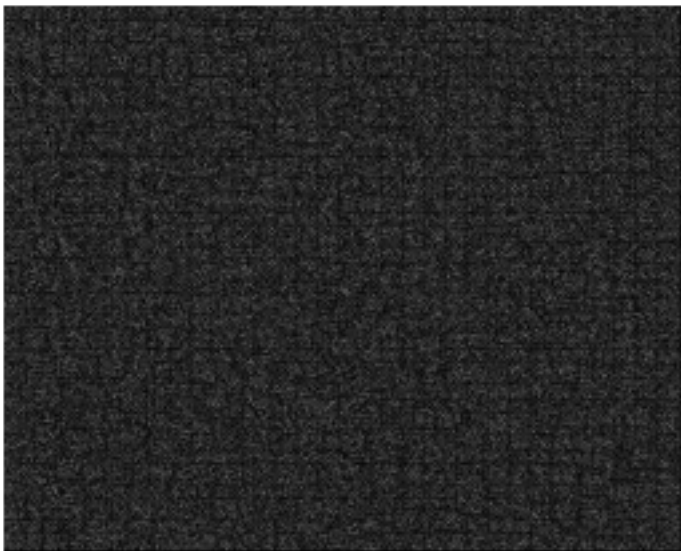
4.1.1.2 Results



screenshot_1_mean



screenshot_1_med



screenshot_1_std

4.1.2 Temporal Analysis

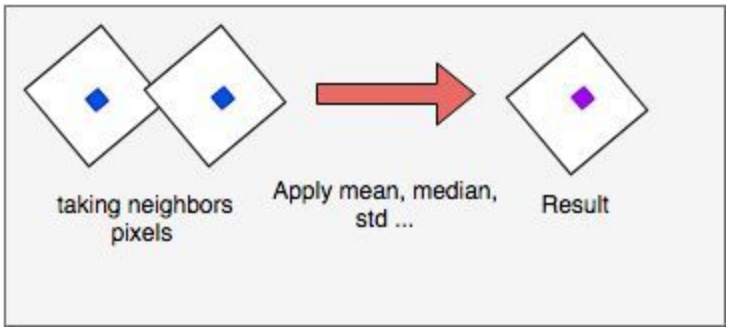


Figure 4 : Applications

4.1.2.1 Observations

Unlike the previous method which takes a single screen for input, the temporal analysis relies on successive screenshots. We noticed that the calculation of standard deviation over two successive image are enough to reveal the symbols, but the quantity of noise is still highly present. However, we found that spaced screenshots leads to image information loss. If we dismiss two successive screens the symbols are broken, that is normal because symbols are not static.

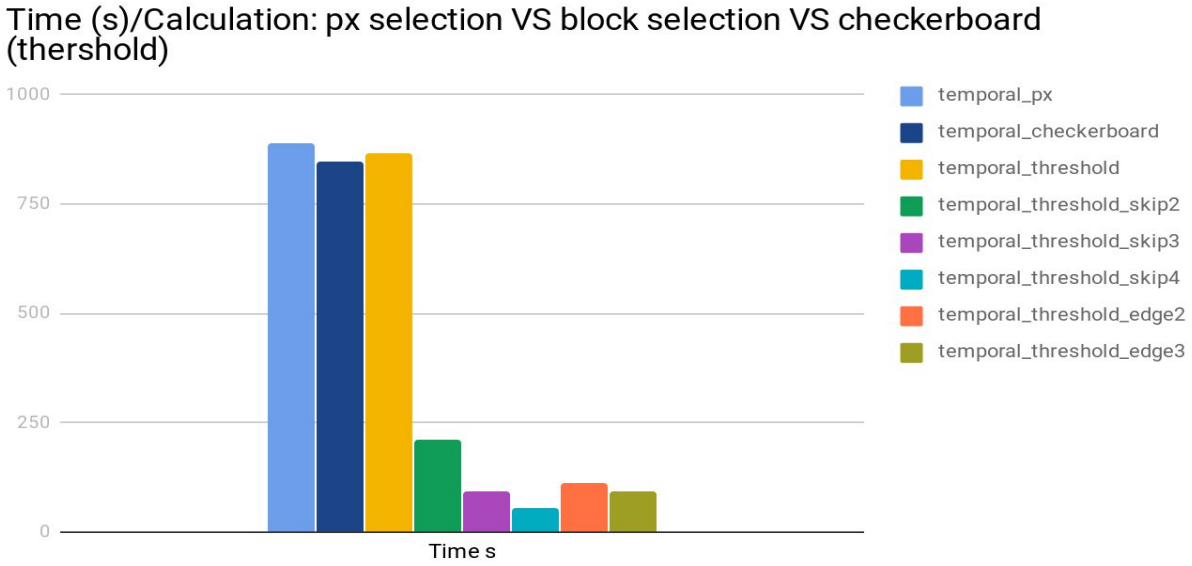
As we stated in our approach, we are seeking to make this attack the lightest as possible. We tried to work on sets of pixels regrouped as blocks. This approach take fewer time than the neighbour's selection which work on each pixel.

Figure 5 : Block/px Selection

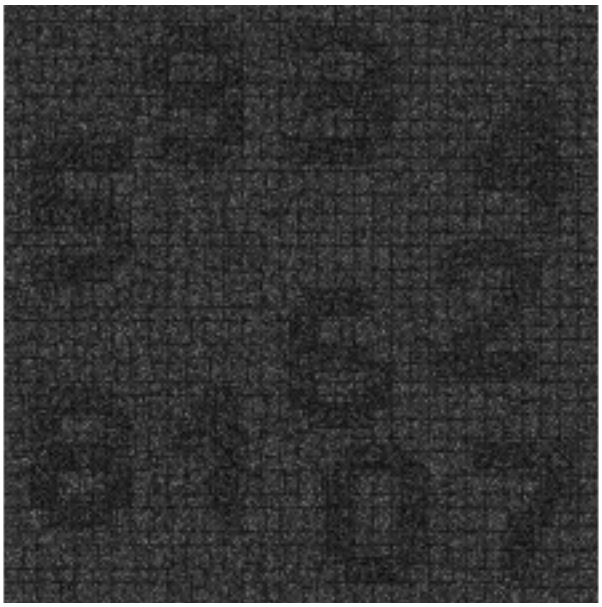
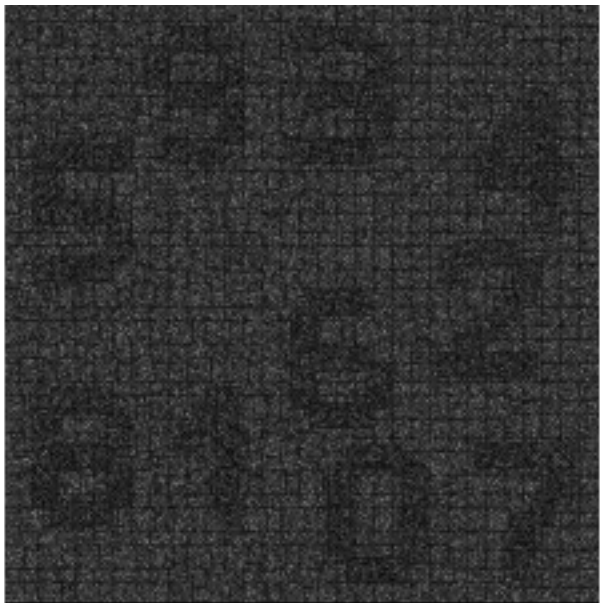
In an another perspective, we tried to skip some pixels which result in shrinking the screenshot. We also tried a checkerboard iteration.

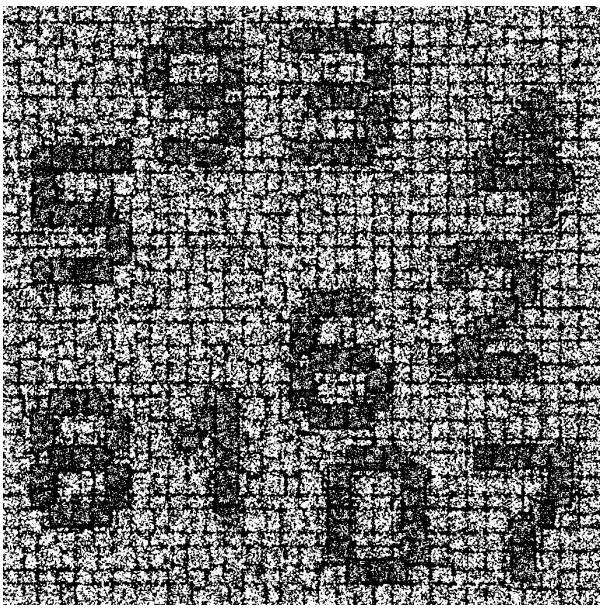
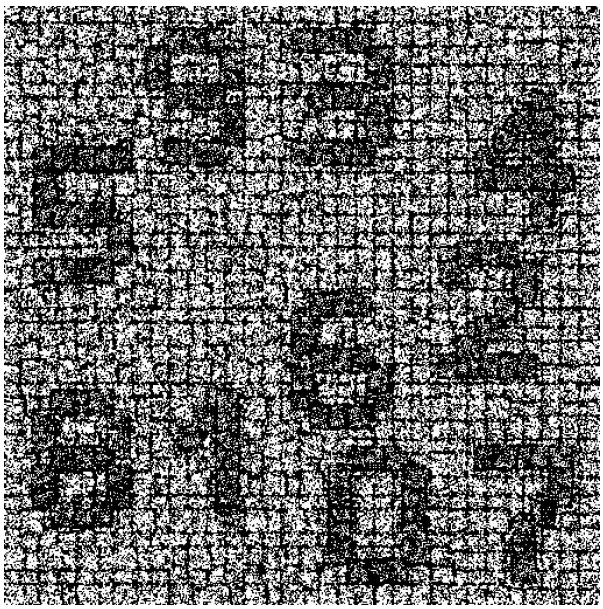
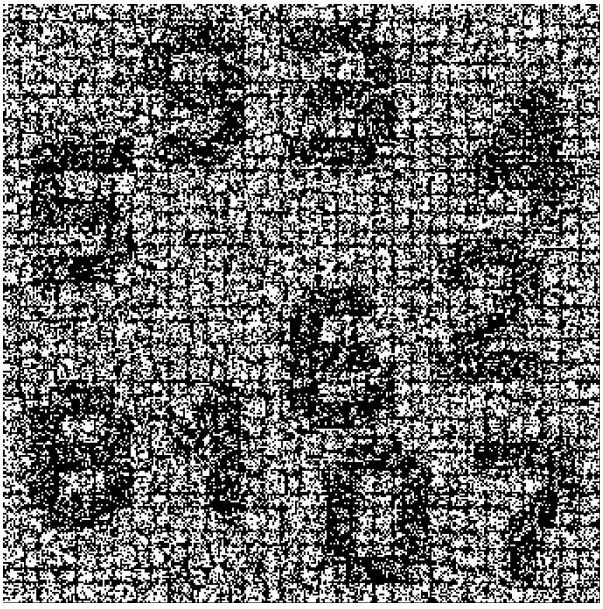
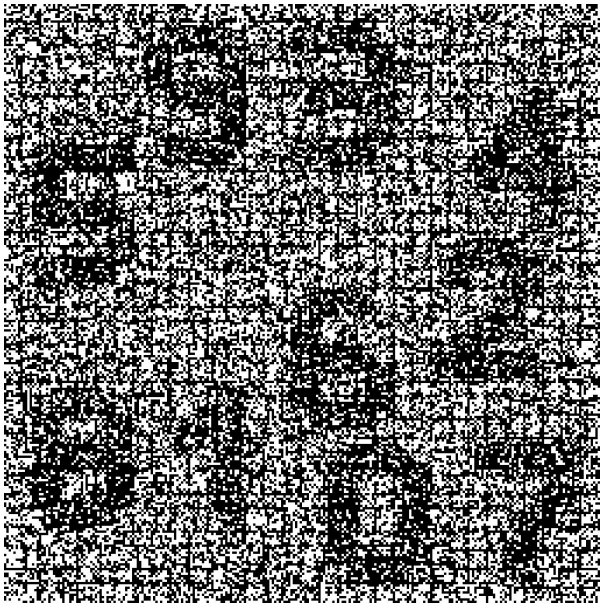
4.1.2.1 Time Consumption

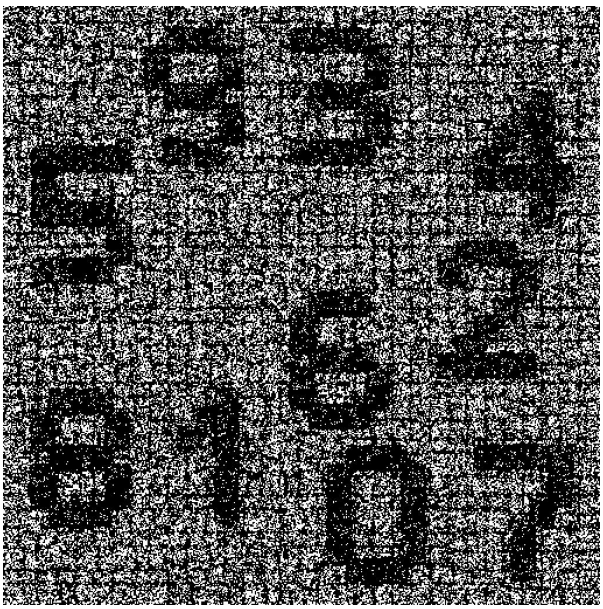
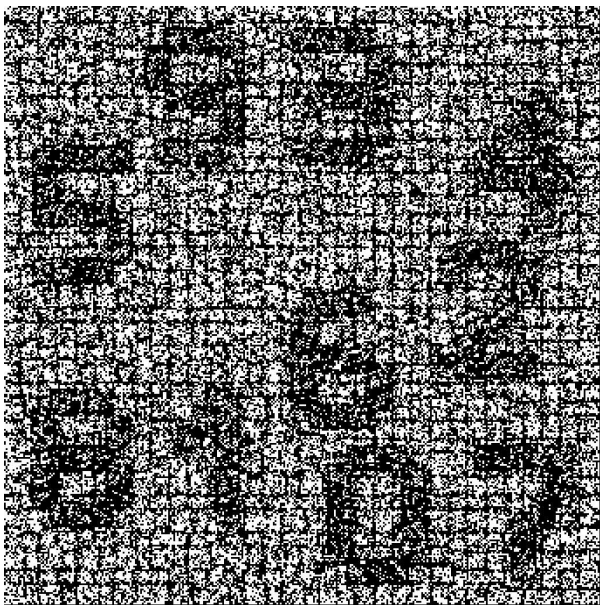
The figure below represents the time consumption for the different temporal analysis approaches.



4.1.2.1 Results

	
temporal_px	temporal_checkerboard

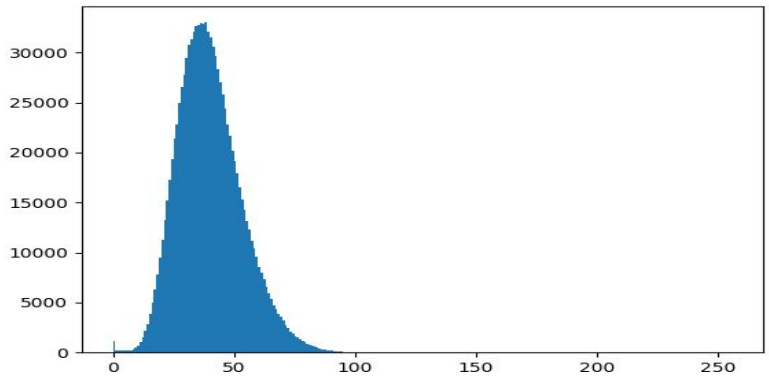
	
temporal_threshold41	temporal_threshold41_s2
	
temporal_threshold41_s3	temporal_threshold41_s4

	
temporal_threshold41_e2	temporal_threshold41_e3

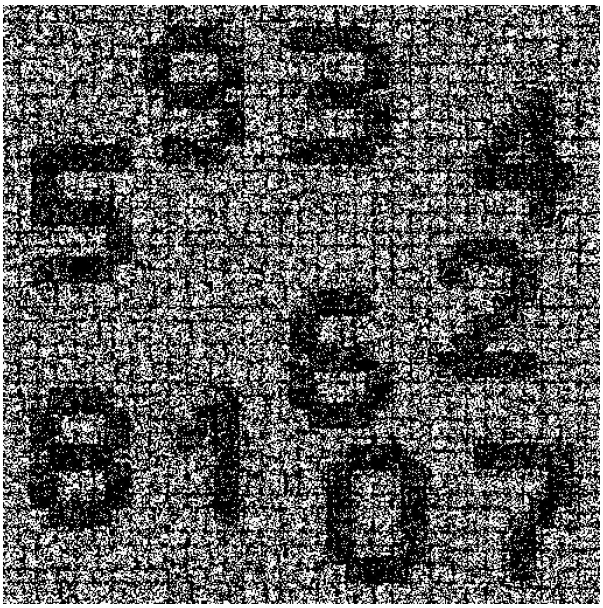
4.2. Thresholding - Binarization

4.2.1 Observation

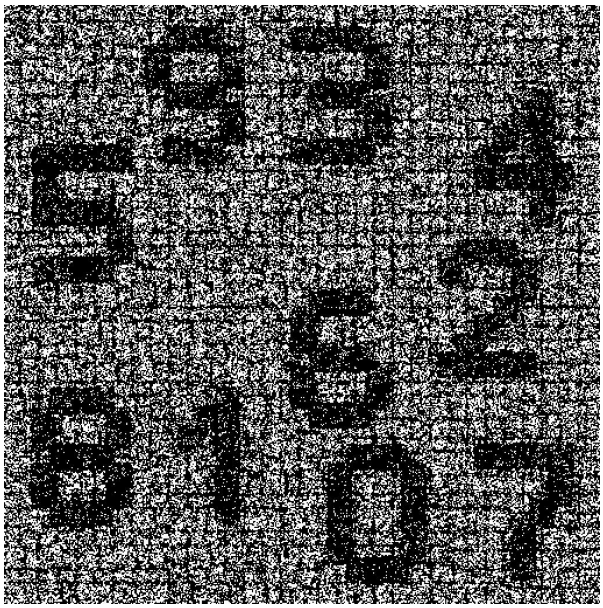
The previous step leads us to trying to separate the background from the foreground by thresholding the result of the standard deviation. We relied on a histogram of the pixels dispersion to find the right spot between the background and the symbols texture. We also applied the Otsu thresholding algorithm which confirms our guesses.



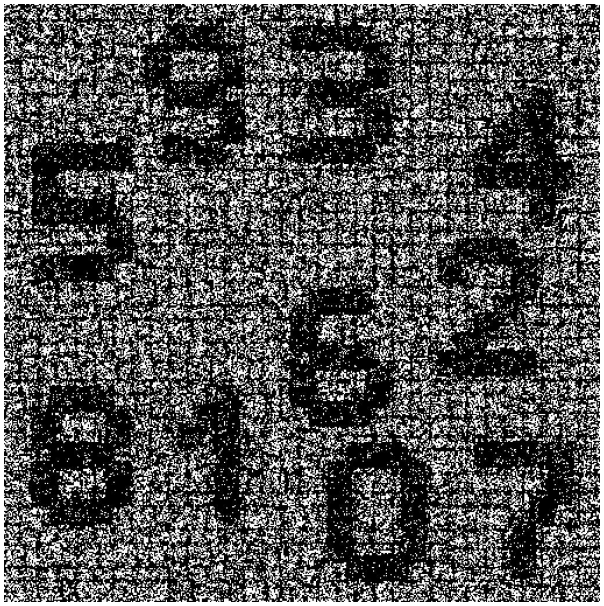
4.2.2 Results



temporal_e2_t39



temporal_e2_t40



temporal_e2_t41

4.3. Noise reduction

The thresholding wasn't enough to completely reduce the background noise. We moved to other applications.

4.3.1. Moving Average

4.3.1.1 Observation

The moving average filter (*MAF*) is used to reduce the scattered pixels while trying to reconstitute the symbols. In a general way we try to increase the number of white pixels on the background, while preserving or even reconstituting the dark pixels of the symbols.

The technique used relies on calculating the average of a preselected set of pixels. The resulted pixel is binarized through a simple thresholding of white and black pixel.

We constated, that in fact, the *MAF* is reducing the noise, but it reach invariance after some iteration.

4.3.2. Dilatation

Unfortunately, our MAF reaches invariance without providing an understandable image for OCR. We moved on to dilatation filter in search for a technique to isolate the numbers and enhance the white background.

4.3.2.1 Observations

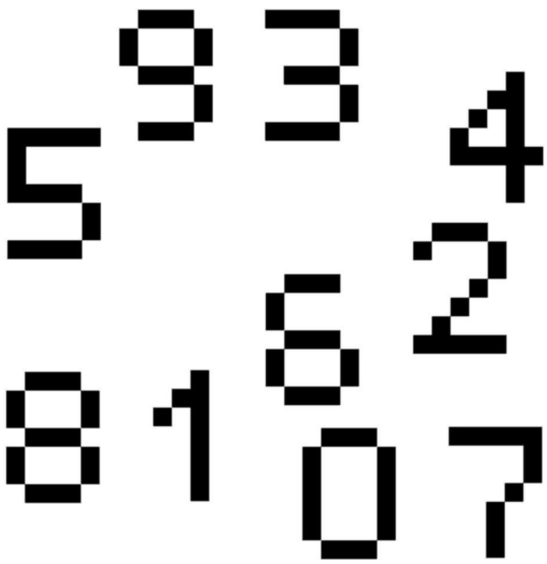
The operation used for dilatation consists on convoluting an image A with some kernel (B), which can have any shape or size, usually a square or circle. The kernel B has a defined anchor point, usually being the center of the kernel. As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to "grow".

The problem we faced is that if the symbols aren't well represented , i.e: if the pixels which represent the symbols aren't compact *(dense)*, the dilatation filter will spread them and leads to broke the boundaries.

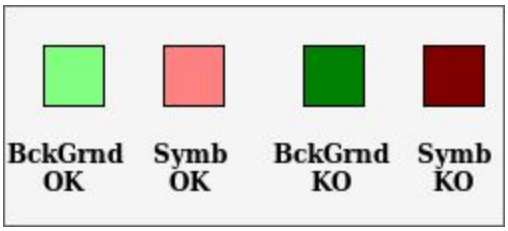
5. Evaluation

For measuring the effectiveness of our attack, we had several approaches. We firstly focused on unhiding the symbols and make them understandable by the attacker. From there we tried a statistical approach to count the well placed pixels and have a practical measure. Indeed, as our approach is more experimental, we focused on clarifying the

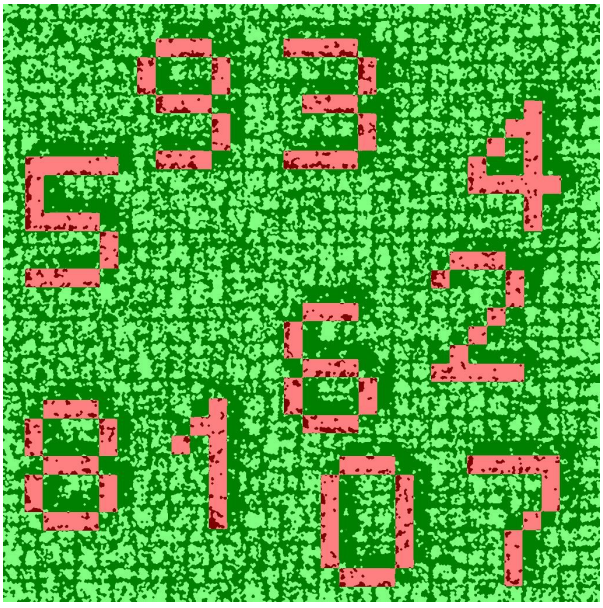
symbols as more as possible before using OCRs because they are not generic and needs heavy configuration.



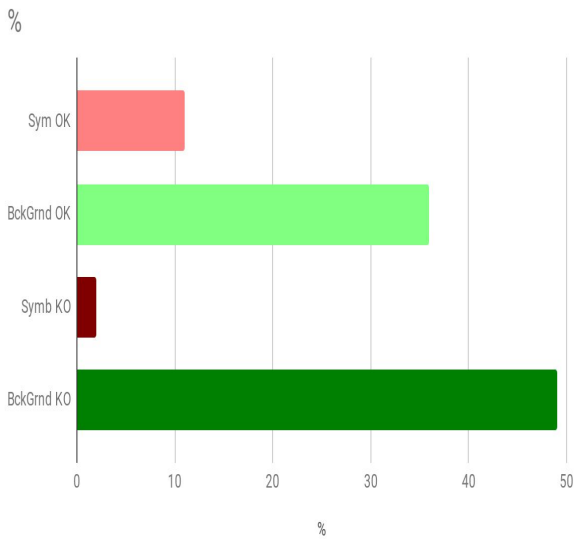
Solution



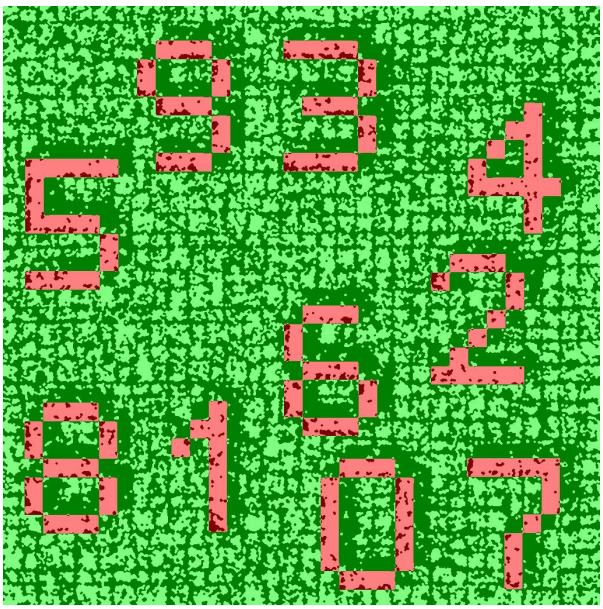
Difference Legend



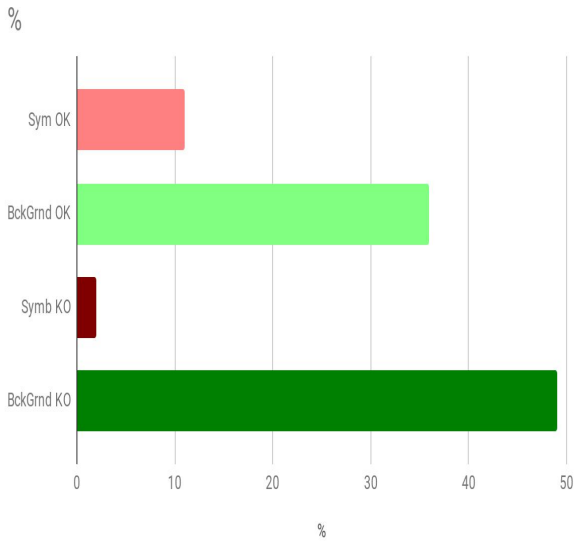
diff_sol-img_1_2_std_px_t41_tor10



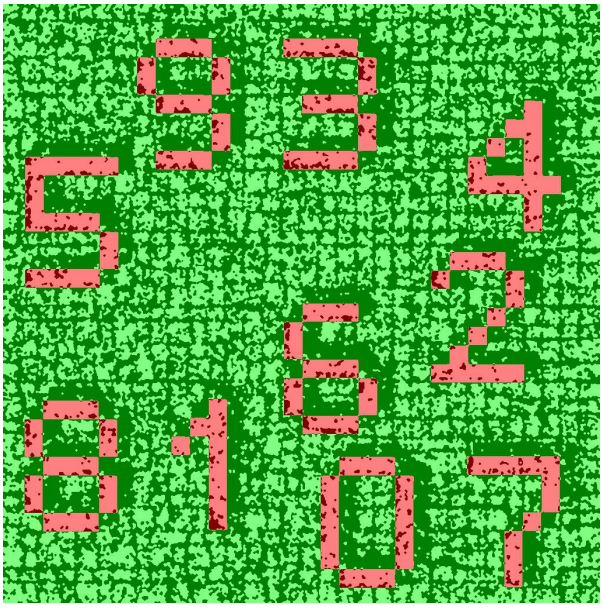
stat_sol-img_1_2_std_px_t41_tor10



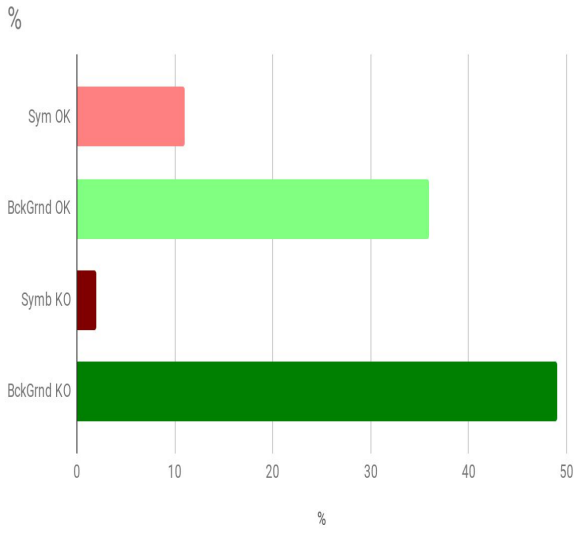
diff_sol-img_1_2_std_px_t41_tor25



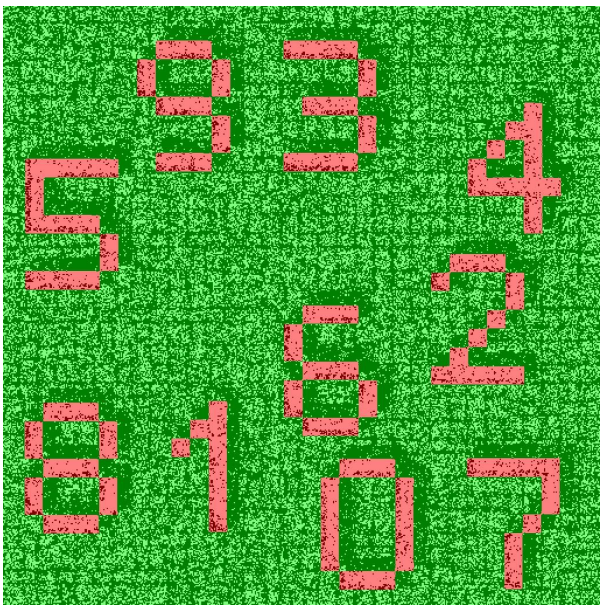
stat_sol-img_1_2_std_px_t41_tor25



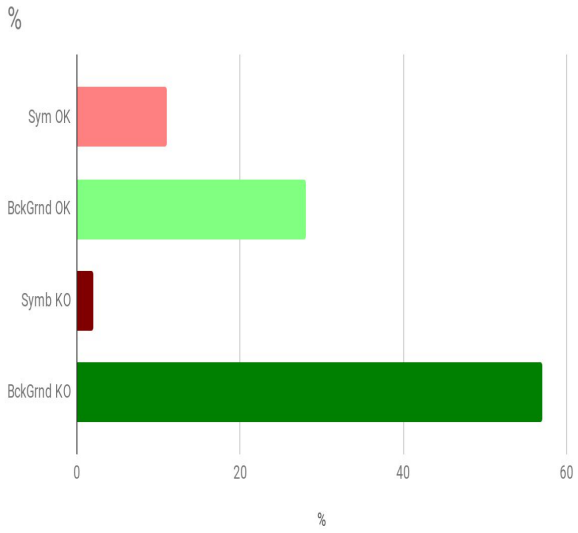
diff_sol-img_1_2_std_px_t41_s2_tor10



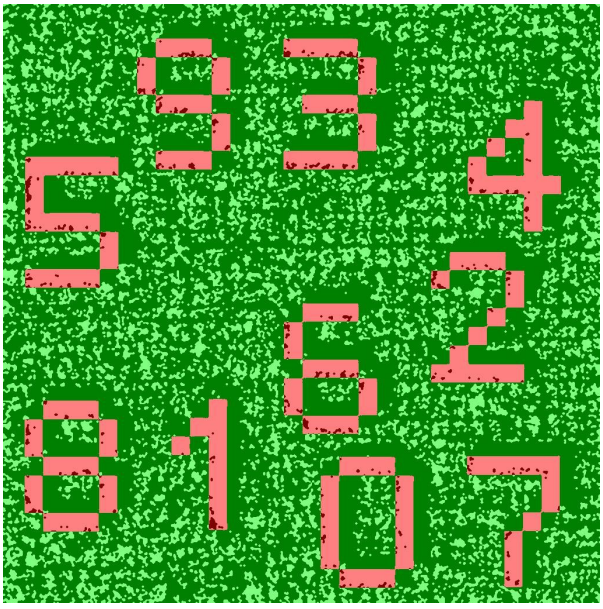
stat_sol-img_1_2_std_px_t41_s2_tor10



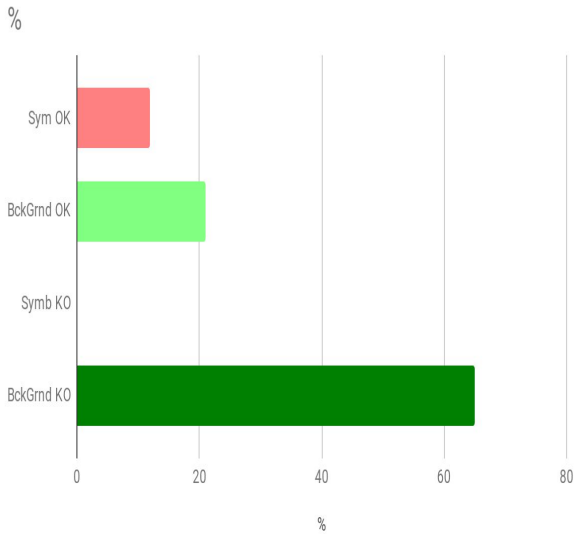
diff_sol-img_1_2_std_t41_e2



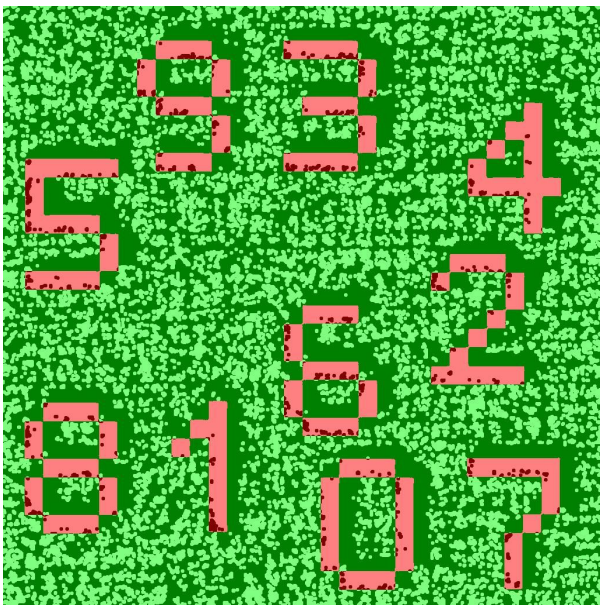
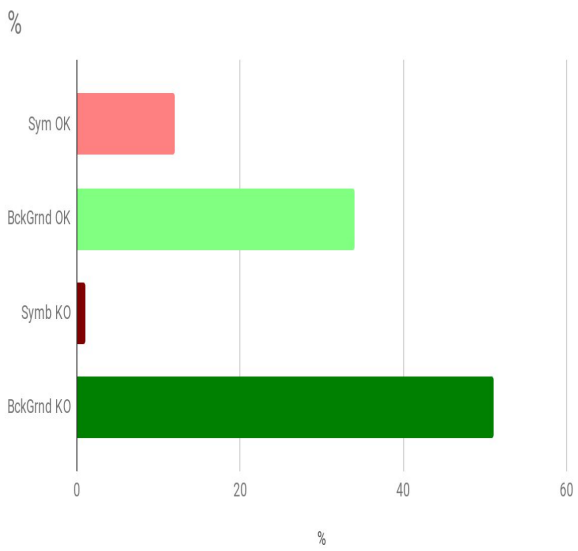
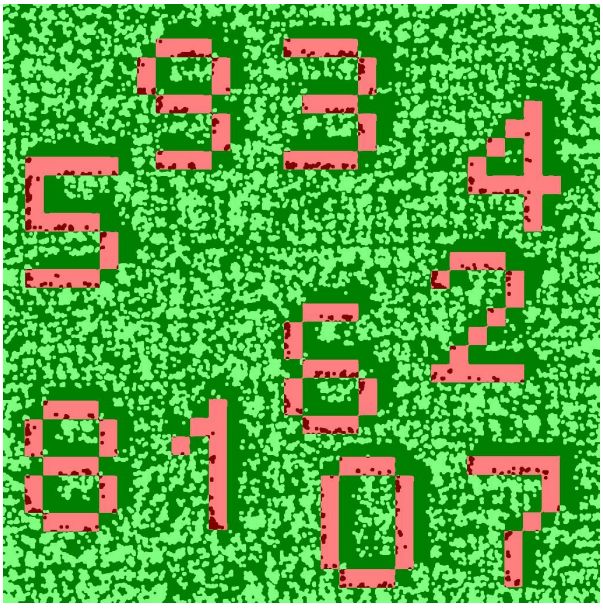
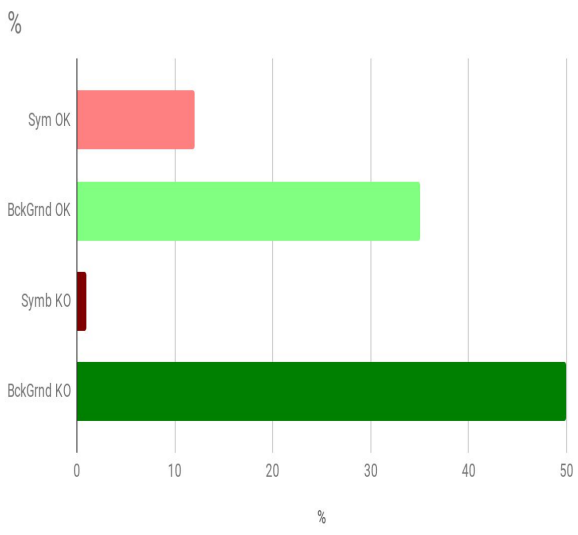
stat_sol-img_1_2_std_t41_e2

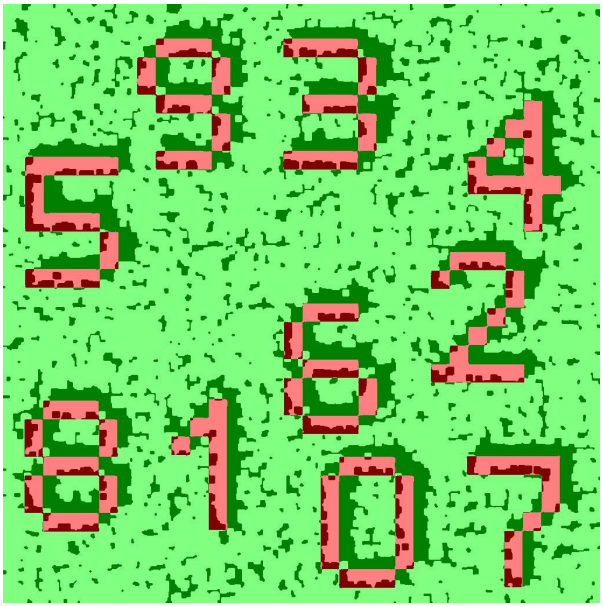
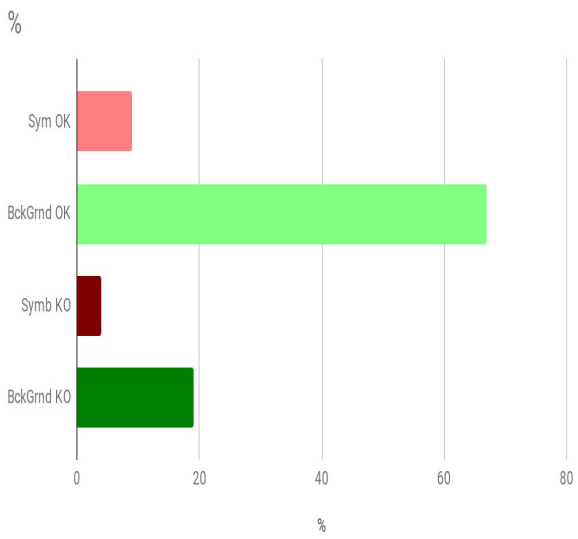


diff_sol-img_1_2_std_t41_e2_tor10



stat_sol-img_1_2_std_t41_e2_tor10

	
diff_sol-img_1_2_std_t41_e2_tor10_dilate1	stat_sol-img_1_2_std_t41_e2_tor10_dilate1
	
diff_sol-img_1_2_std_t41_e2_tor10_dilate1_tor20	stat_sol-img_1_2_std_t41_e2_tor10_dilate1_tor20

	
diff_sol-img_1_2_std_t41_e2_tor10_dilate4_tor20	stat_sol-img_1_2_std_t41_e2_tor10_dilate4_tor20

6. Ways of Improvements

Our work isn't finished yet. We lacked time to conduce completely our analysis. We didn't achieve to pass the final OCR test for automatization. We may increase the success rate with improving the MAF and dilatation filter. The final code can be refactored and optimized to reach better performances.

7. Conclusion and Perspectives

In this paper we presented a way to break the Common Fate's Law Virtual Keyboard. This study shows that visual CAPTCHA aren't strong enough. We may break the system by exploring more in deep the Dilatation + MAF filters. We could probably break the system with Deep Learning Attacks which may be hard to set up, but could reach good performance.

To sem up, the technologie used is still a good way to prevent and patch the screenshot keylogging attack for the moment.

8. Bibliographie

- Virtual Keyboard Logging Counter-measures using Common Fate's Law

  Nicolas Echallier, Gilles Grimaud, Julien Iguchi-Cartigny, Jean-Marie Place, Jean-Philippe Wary. Universite de Lille, France. Orange Labs, France

- Bacara, C., V. LeFils, J. Iguchi-Cartigny, G. Grimaud, et J. P. Wary. 2015.

  `Virtual Keyboard Logging Counter-Measures Using Human Vision Properties`. In 2015 IEEE 17th International Conference on High Performance

  Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), 1230.

  oi:10.1109/HPCC-CSS-ICESS.2015.269.

- Yu-Hsiang Wang. Tutorial: Image Segmentation. (http://disp.ee.ntu.edu.tw/meeting/%E6%98%B1%E7%BF%94/Segmentation%20tutorial.pdf) *(access time : 05/02/2019)*
- Image Segmentation. (https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic3.htm) *(access time : 05/02/2019)*
- IMAGE SEGMENTATION USING K-MEANS CLUSTERING BASED THRESHOLDING ALGORITHM. (**https://pdfs.semanticscholar.org/1b21/66d8d5fa3ff281edf52216763aad92a3628e.pdf**) *(access time : 05/02/2019)*
- Thresholding. (**https://en.wikipedia.org/wiki/Thresholding_(image_processing)**) *(access time : 05/02/2019)*
- Balanced Histogram Thresholding. (**https://en.wikipedia.org/wiki/Balanced_histogram_thresholding**) *(access time : 05/02/2019)*
- Otsu Method. (**https://en.wikipedia.org/wiki/Otsu%27s_method**) *(access time : 05/02/2019)*
- Dilatation/Erosion. (https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html) *(access time : 05/02/2019)*
- Dilatation/Erosion. (https://docs.gimp.org/2.8/en_US/plug-in-dilate.html) *(access time : 05/02/2019)*