

# A survey of keylogger and screenlogger attacks in the banking sector and countermeasures to them

Hugo Sbai<sup>1</sup>, Michael Goldsmith<sup>1</sup>, Samy Meftali<sup>2</sup>, and Jassim Happa<sup>1</sup>

<sup>1</sup> Oxford University, Department of Computer Science, 15 Parks Rd, Oxford OX1 3QD, UK

`{hugo.sbai,michael.goldsmith,jassim.happa}@cs.ox.ac.uk`

<sup>2</sup> Université de Lille 1, Centre de Recherche en Informarique, Signal et Automatique (Cristal), Batiment M3 extension Avenue Carl Gauss, 59655 Villeneuve d'Ascq Cedex, France

`samy.meftali@univ-lille1.fr`

**Abstract.** Keyloggers and screenloggers are one of the active growing threats to user's confidentiality as they can run in user-space, easily be distributed and upload information to remote servers. They use a wide number of different techniques and may be implemented in many ways. Keyloggers and screenloggers are very largely diverted from their primary and legitimate function to be exploited for malicious purposes compromising the privacy of users, and bank customers notably. This paper presents a survey of keylogger and screenlogger attacks to increase the understanding and awareness of their threat by covering basic concepts related to bank information systems and explaining their functioning, as it presents and discusses an extensive set of plausible countermeasures.

**Keywords:** Keyloggers, Screenloggers, Virtual keyboards, Optical Character Recognition, Neural networks, SVM, Noise.

## 1 Introduction

Currently, banking data is digital, integrated into banking information systems and accessible to employees, bank supervisors and customers. Thus, all users of such an information system connect with passwords to their accounts and get some privileges. The privileges can, for instance, be a simple consultation of an account balance, closing or creation of an account or transactions of large financial amounts from an account to another.

This simplicity of access and the large amount of money that can be manipulated or diverted by any malicious person with an adequate password make these systems privileged targets of many computer attacks using various software and malware. Among these, the use of keyloggers or screenloggers is often particularly effective and dangerous for banking information systems.

Keyloggers and screenloggers are software used to capture and save, without the user's knowledge, keystrokes or screenshots into files. Most currently

available keyloggers are considered "legitimate" applications and they are used to fulfil many legitimate and legal functions such as tracking children's use of the internet, tracking cases of inappropriate use of business computers [29]. Yet, they are very largely diverted from their primary and legitimate function to be exploited for malicious purposes, and unfortunately, the theft of various online payment systems credentials has become one of the main application of keyloggers/ screenloggers [4]. Many keyloggers/screenloggers try to conceal themselves, and unlike other types of malware, they do not affect its functioning. Despite that, they can be very dangerous for the user privacy and the organisation to which the information system belongs.

A keylogger can intercept passwords or other confidential information entered by the user with his keyboard, when a screenlogger is capable of capturing screenshots. This information is then passed to the source of the malicious program. This paper will only target the case of banking institutions, even if the theft of such data can have very serious consequences in other sectors, for example regarding economic and political intelligence operations, commercial or state secrets, compromising the security in public and private organisations. To the best of our knowledge, there is no document that provides a clear synthesis of the current knowledge about screenloggers. This is the aim of this paper. The existing works in the literature presenting an overview of this type of malware, concentrate generally on keyloggers especially on the detection phase as in [29] or on the processing and implementation details as in [30]. One of the originalities of this paper is that it presents the vulnerabilities of screenloggers at all stages of their operation, and focuses particularly on the most critical phase, which is data automatic recognition.

The rest of this document is organised as follows: in Section 2 we define the basic concepts related to keyloggers and screenloggers and their illegitimate use against bank information systems. Section 3 aims to present the general functioning of these attacks step by step and to propose countermeasures. Section 4 focuses on the data extraction process, showing the different techniques that can be used, and discussing their weaknesses and possible countermeasures. Finally, in Section 5 we conclude by summarising the work and discuss potential directions for future research.

## 2 Basic concepts

### 2.1 Keyloggers classification

**Keyloggers :** a keylogger might be either a piece of software or a hardware component that monitors key presses on a computer. These details will be saved into files and sent later to the person specified in the keylogger settings. We distinguish two types of keyloggers: software and hardware ones.

*Hardware (HW) Keyloggers:* they devices connected to the keyboard or the computer. Their detection needs a physical human verification [5]. These boxes can intercept all the data transmitted by the keyboard including the recovery of

BIOS password and bank identifiers.

The oldest ones are Module type keyloggers and have a PS2 interface; they are usable on keyboards having this same interface [19], and often have a form extremely close to that of USB-PS2 adapters. There are also USB versions that look like the USB/Wifi or USB/Bluetooth peripheral. A third form is less accessible to the general public but is quite efficient. It consists of a tiny electronic card connected inside the keyboard. Lastly, probes can be used for side channel attacks. For wireless keyboards, there is no need for a specific additional box to recover the keys entered [9]. This can be done just by capturing the waves emitted by the keyboard to communicate with the receiver and then decrypt the communication, which employs weak encryption in most cases.

*Software (SW) Keyloggers:* they are much more common because they can be installed remotely, e.g. via a network, and generally, do not require physical access to a certain device for recovering collected data (the data can be transmitted periodically by email) [12]. Although these keyloggers are more easily detectable by other software tools, they still have more advantages than hardware keyloggers.

A hardware keylogger is only capable of recording keystrokes *out of context* i.e. that have no relation to the user environment. A software keylogger records not only keystrokes but also the state of the target machine. The most targeted applications are web browsers because they allow the recovery of usernames and passwords (bank accounts login for example) [14]. One of the main strengths of this type of keyloggers is that they can be deployed indifferently on computers, tablets or smartphones.

**Screenloggers (also known as touch logger, tap logger) :** they are a variant of keyloggers software [8]. Their main use is to take screenshots and even make videos retracing all computers' activity. A Screenlogger records the movements of the mouse, along with screen captures during the click event.

## 2.2 Comparative evaluation between Screenloggers, HW Keyloggers and SW Keyloggers

As shown in Table 1 below, screenloggers have some important advantages comparing to HW or SW keyloggers. Indeed, they can be used to affect a device remotely and at a very large scale in the same way as SW keyloggers, providing the hacker with a complete set of data and information. In fact, screenshots give additional details, making passwords extraction much easier.

The only way to detect hardware keyloggers is to become familiar with these devices or to check the device internally and externally [10] regularly. Even the NSA catalogue published in late 2013 reflects the difficulty of finding one's own recording devices that are barely bigger than a fingernail. This constitutes the main advantage of HW keyloggers, but still, the hacker must have physical access to the device to affect it, this represents a significant drawback. For software keyloggers, the infection tracks are the same as for other malware.

Table 1: Screenloggers vs HW Keyloggers vs SW Keyloggers: features, infection capabilities and detection.

	HW Keyloggers	SW Keyloggers	Screenloggers
Keys	Yes	Yes	Yes (but not their main use)
Use of multiple inputs (mouse, pad, ..etc)	No	No	Yes
Screenshots	No	No	Yes
System context	No	Yes	Yes
Ease of infection	****	*	*
Large scale infection	*	****	****
Ease of exploitation	*****	***	*
Ease of detection by SW	No	***	***
Ease of detection by user	***	*	*

### 2.3 Screenlogger attack against Banking Information Systems

The main objective of any hacker attacking a banking information system is to steal confidential information such as authentication information. He could try to remotely install a screenlogger program on a client device or directly on a computer inside the bank [15]. This last alternative should give more privileges to the hacker, but it is harder than attacking simple client account.

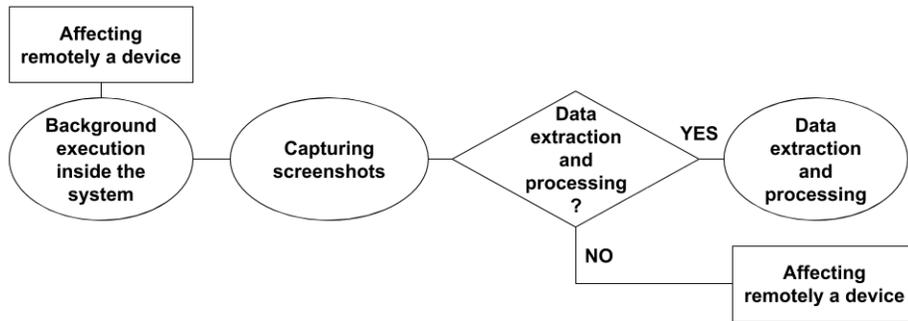


Fig. 1: Screenloggers operating process.

The most common process of a screenlogger might be separated into five steps as shown in Figure 1. First, the hacker must affect a device, generally in a remote way, using emails or any other files transmission technique [16]. Second, after the malicious program has been installed, it will run as a background service. Then comes the main job of screenloggers, which consists in recording screenshots at regular periods or triggered by mouse clicks. The resulting captured screenshots

might be treated or not on the host device depending on the nature of the screenlogger. Finally, the raw or processed data is transmitted to the hacker through the network.

Each of the above steps utilises a certain number of vulnerabilities to make the whole process as efficient as possible. The objective of the next section is to give more details about operating mode of each part, vulnerabilities used, and with countermeasures that can be taken by users to protect their devices.

### 3 Screenloggers processing steps and countermeasures

At each stage of their operating mode, screenloggers exploit a number of flaws, using resources to optimise their performance and ability to collect and quarrying data [18]. However, there are measures allowing to minimise as far as possible the risk of an infection by a screenlogger before its installation on the target machine on the one hand, and a set of actions to detect the existence of malware after infection, on the other hand, trying to reduce its damages.

The purpose of the current section is to give an overview of the weaknesses used by screenloggers at each stage of their functioning (as seen in Figure 1) as well as countermeasures that could be taken by the victim to ensure his safety.

- **Device infection** : the first step of a Screenlogger process is to infect the target machine, as seen in Figure 1. The way in which such software gets on a machine is quite similar to the infection by almost all modern malware. Indeed, a Screenlogger infects a machine (a computer, a tablet, or a smartphone) through one of these main methods:
  - **Manual installation** : it can be done when the hacker has physical access to the device and the rights granting him the privileges to copy and execute programs. This is practicable to a certain level for attacks against people without specialised knowledge even about simple security basics, and those who cannot afford to protect their equipment notably against the theft.
  - **Transmission over the network** : the Screenlogger can also be transmitted on the target machine remotely, as is often the case, using a network protocol such as emails, FTP or any other file transfer protocol [12]. In all cases, the transfer is done without the knowledge of the target machine owner.
  - **Transfer from storage devices** : the malicious program can be transferred via a device such as a USB key, a memory card or an external hard disk that the user connects to the device without being aware of the malware presence.
- **Countermeasures to prevent device infection** : although zero risks can reasonably not be guaranteed on any machine connected to the internet, there are nevertheless many measures allowing to reduce the risk of a screenlogger infection substantially. As well as how a device can be infected, these measures can be divided into three parts:

- **Countermeasures to prevent manual installation :** to avoid any manual transfer of the malware, the user should take awareness in consideration as a measure, to ensure and protect the system from any possible access (especially the administrator mode).
- **Countermeasures to prevent transmission over the network :** the first recommendation is to open e-mails only coming from known senders, which can be authenticated by digital signature when possible. Second, is to not open any attachments without being sure of the non-dangerous nature of its content. It is the same for hypertext links. Indeed, the transmission of Screenloggers as attachments to emails or as files on a remote server accessible via a hypertext link is a widespread practice.
- **Countermeasures to prevent transmission via storage devices:** an effective practice, especially for professional business users, is to never use a removable storage media on their devices. Indeed, a rigorous separation of the machines for professional or personal use makes it possible to ensure their non-infection by any malware in general and Screenlogger in particular. Another measure to reduce the risk of being infected is to systematically check any external storage device using an up to date anti-virus before each use.
- **Background execution mode :** once the screenlogger is installed on the target machine, it works as an active process, continually scanning the events triggered by keyboard keys or mouse clicks. Under a windows-based system, a basic and primitive version of a keylogger/screenlogger can be based on calling the **GetAsyncKeyState** system function [11] to return the key state (pressed or released).  
This active wait is a resource consuming (CPU, storage and power). Additionally, data transmission towards the attacker may require intensive use of the network, especially if they are transmitted without prior treatment on the victim's machine. In the last few years, some keyloggers integrated dissimulation methods to prevent their files from being discovered manually or using anti-virus software. They mainly use two types of dissimulation methods: user or kernel mode [31].
- **Countermeasures for screenloggers detection :** in information systems belonging to companies, and particularly to banks, there is usually an IT department that should monitor resources used to detect any unnatural overhead. This potential overhead might be more or less important depending on the nature and performances of the screenlogger. Thus screenloggers detection is not always guaranteed. In the case of private devices belonging to individuals, monitoring device resources still possible by the mean of some integrated tools.
- **Keys captures and screenshots :**
  - **Keys recording :** the basic function of any simple keylogger is to capture pressed keys on the keyboard. An application requiring such data is generally not considered as illegitimate by operating systems or by any common anti-virus programs since reading pressed keys is a standard operation required by several legal tools. Basic Keyloggers can

capture all keystrokes and save them in files which will be sent to the hacker's server [14].

- **Screenshots capturing and storing :** the general idea behind Screenloggers is to capture all the bits displayed in a DC (Device Context). A DC is a GDI (Graphic Device Interface) window object that defines a set of objects and properties representing a graphical output. The image format used by screenloggers for recording screenshots is Bitmap. This is the simplest graphic format; where every pixel is coded in RGB. They can use another file format (JPG, GIF ...) but the size of the generated files could cause memory congestion and CPU overload of the target device [32].

- **Countermeasures against keys captures and screenshots :**

- **Countermeasures against key recording :** at present all banking applications use virtual keyboards on smartphones (natively present in the device or integrated in the application), and some even use them on laptops (Axa bank, Oney, Abanca...). This is an efficient countermeasure against basic keyloggers capturing only pressed keys [7]. However, hackers have adapted themselves to this new situation and have improved their programs to capture more than just keyboard keys but also screenshots and even videos. So virtual keyboards are no longer a sufficient countermeasure. Several approaches have been proposed to try to counter these modern keyloggers.
- **Countermeasures against screenshots recordings :** if a screenlogger is installed on a device without being detected and is capable of taking screenshots, there is no way to prevent it from working and taking pictures without the risk of altering the execution of other legitimate applications.

A suggestion mentioned in [6] consists in adding artefacts on the screen when a click occurs. For example, it can be the display of an artificial mouse pointer to prevent the malware from knowing which part of the screen has been clicked. Some researchers also suggested a dynamic virtual keyboard that mixes keys layout after each click [1].

A colour code is used to remember characters positions easily. The user can enter one character at a time. Initially, the user should note the position of the character he wants to use. They must then click on the *hide keys button* to hide all characters. Assuming the password is **xyz**, the user can then click to type **x**. After this action, the keyboard layout changes again and the process is repeated one more time. A similar approach was proposed by Ankit Parekh et al. [2].

- **Transmission to the hacker's machine :** once the screenshots (or extracted data) are stored on the victim system they can be transmitted to the hacker in two forms: raw images or data extracted after treatment using an Optical Character Recognition (OCR).

In the first alternative, transferring captured images as they were taken over the network will result in high bandwidth consumption. The data flow

exchanged between the attacker and the target must be minimised. Some solutions can be considered to reduce the importance of this data flow. The first one is to use various compression methods (LZ77, RLE ...) [33]. The second one is not to send the entire screen, but only the target window.

In the second alternative, the volume of transmitted data via the network will be less. However, the OCR processing on the target machine consumes a lot of resources (e.g. CPU, memory, disk storage), which can facilitate the detection of the screenlogger.

- **Countermeasures against data transmission :** a very close examination of the network usage might reveal the presence of a keylogger. A firewall is an efficient defence against key/screenloggers because it will monitor the computer's activity, and upon detecting that a program desires to send data to the hacker, the firewall will ask for permission or display a warning.

## 4 Focus on the data exploitation step

The data exploitation can be performed either by the human eye or automatically. In the following, we will present these two alternatives and discuss their weaknesses.

### 4.1 Manual screenshots exploitation

To avoid the use of OCRs on the victim machine, and thus reduce the use of resources to minimise the risk of being detected, some screenloggers directly transmit screenshots to the hacker's server without processing them locally. These screenshots are sometimes checked and processed manually by the hacker without using any program. This type of screenloggers is particularly effective not only because of the difficulty in detecting them but also because of the use of the human eye and brain to analyse screenshots.

### 4.2 Countermeasures against manual exploitation

Since it is harder to prevent a human from recognising a character than automatic algorithms, the simplest countermeasure would be to hide the character completely. Several ways of doing so were found in the literature. The keyboard hiding method [1] is one example. Another method presented in [17] uses retinal persistence, and its goal is to divide each character into segments and display them one after the other in quick succession. At sufficiently high speed, a human can see the whole character while the software does not. On a screenshot, the numbers are never fully visible. On the other hand, if the malicious software could take enough screenshots, it could rebuild characters.

### 4.3 Automatic extraction from screenshots

A text is a collection of characters belonging to an alphabet, united in words of a given vocabulary. OCR locates and identifies characters in an image (scanned

or captured). Since the recognition is done character by character [24] the recognition system depends on several phases, which are presented in Figure 2.

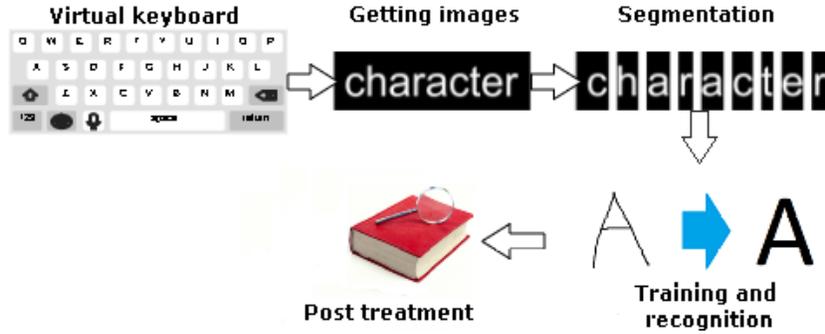


Fig. 2: Main steps of an OCR system : Once the screen is captured (*Getting images*), each character is isolated (*Segmentation*) and recognized by an OCR software (*Recognition*). Then, the result can be post-processed using a dictionary to detect possible mistakes (*Post treatment*).

**Acquisition :** it is the aspect which consists in capturing the image of a text, and converting it into appropriate digital magnitude for the processing system using screenlogger software (screenshots).

**Segmentation :** the purpose of segmentation is to find in images where the text is located and to isolate each character. The segmentation can be performed by run length smearing algorithm that consists of blackening the white pixels situated between two black pixels if their number is less than  $n$  [28]. The algorithm is performed vertically and horizontally. By varying  $n$ , we can segment characters, words, lines or paragraphs. Before this, the image must be binarised *thresholding*. All the pixels having a value lower than the threshold will be encoded 0 and become black. All the pixels having a value above the threshold will be encoded 255 and will become white. The higher the threshold chosen, the cleaner the binarized image will be. Once the image binarised and the characters are isolated, the recognition itself will be performed.

**Recognition :** before taking any decision, we need to acquire knowledge and organise it into class models. In this context, we can find two main types of algorithms: supervised and unsupervised learning [25]. Once the learning is achieved, the decision module has to make a decision on the input character by giving an answer [25]. This answer can take three forms: success if the answer is unique; confusion in case of multiple answers; or rejection of the form if no model matches its description. In the first two cases, the decision may be accompanied by a *likelihood* measure. Recognition approaches can be grouped into five main groups: template matching, statistical, structural, stochastic and hybrid.

- *Template matching* : this approach compares pixel by pixel the image taken from the character with a library consisting of a set of different types of characters called *Template* or *library models*. The system compares the grey levels of the model with those of the library different elements, and then it assigns the image the model class in which it matches the most.
- *Statistical approaches* : in the statistical approaches [25], the recognition consists in finding the class to which the character has the highest probability to belong to, and assessing the risk involved in making such a decision. There are different types of statistical approaches:
  - *Bayesian approach*: the Bayesian method allows the introduction of probability notions in solving problems about pattern recognition [20]. From the physical representation of the forms to be recognised, characteristic vectors are extracted. These vectors to classify are considered as realisations of a random vector  $x$  characterised by a probability density  $f(x)$ . The Bayesian model is easy to build and particularly useful for extensive data sets.
  - *Neural networks*: a neural network is a weighted oriented graph. The nodes of this graph are simple automaton called *formal neurons*. Neurons have an internal state, by which they influence other neurons in the network. This activity propagates in the graph along weighted arcs called *synaptic links*. In OCR, the primitives extracted on a character image are the inputs of the network. The activated output of the network corresponds to the recognised character [22].
  - *Support vector machines (SVM)*: these algorithms are supervised learning binary classifiers, designed to solve prediction problems. The algorithm search for a decision borderline between multiple classes. This approach was initially used to optimise the linear hyperplane of discrimination between classes [21]. Then, the use of the kernel functions made it possible to project the non linearly separable data in an increased space to make them linearly separable.
- *Structural approach* : the structural methods are based on the physical structure of the character; they represent the topological and geometrical properties of the form [25]. These characteristics are extracted from the form representation in the *skeleton* or *the outline*. In these approaches, the characters are generally represented by a set of primitives connected by a relation. The primitives are of a topological type such as an arc, a point, a ball, a corner, or a segment. Many structural classification methods have been developed from an application in character recognition.

*The matching of graph representations* consists in building a graph where the nodes contain the primitives and the links between these primitives [25]. Thus, the recognition consists in making a mapping between this graph and other graphs representing reference characters and constructed during the learning phase. Another method is the *metric techniques*, in which characters are represented by strings of primitives [25]. It consists in measuring the similarity between the strings of the character to be recognised and a reference model by a *distance estimation*.

The main disadvantage of structural methods is related to the extraction of primitives directly from the physical representation of the form [25]. This represents a real barrier because this description is not very resistant to geometric transformations but mostly to *noise*.

- *Stochastic approach* : in the stochastic approach, the character is modelled as a *state graph*, where each state represents an observation, we talk about a *stochastic chain*. Stochastic recognition consists of searching within a state graph the shortest path describing the observed elements. The employed technique is essentially based on the use of *Markov chains* [27].

Markov models are commonly used for character recognition. Each character is a *hidden state* or a *Markov chain of  $N$  states*. The observations extracted from the images are called *primitives* or *characteristic vectors*. An HMM is constructed for each vocabulary word, and the corresponding likelihoods are calculated. We choose the word of the vocabulary that maximises this likelihood.

- *Hybrid approach* : to improve recognition performances, the trend is to build hybrid systems [23] that use different types of features, and that combine several layered classifiers. This combination improves the quality of the classification in terms of accuracy. We can find three types of classifiers combination. The first one is the *sequential combination*, where classifiers are used one after the other, progressively to reduces the number of possible classes. Another type of combination is the *parallel combination*, where classifiers are driven independently in parallel, and final results are then merged. Parallel methods are the most widespread due to their simplicity but also their effectiveness in terms of accuracy. The third approach is the *hybrid combination*, which is a mixture of both previous approaches.

**Post-treatment** : post-treatment represents the last step in OCR operation. It aims to reduce the number of errors. In fact, depending on the quality of the original screenshot, there are often some errors because characters are broken or blurred. Thus the OCR tool must now resolve these errors so that the characters can be properly interpreted. During this step, OCRs generally use dictionary support to improve the recognition quality. Some characters like ('1', 'I') and ('C', 'G') can look very similar, and a dictionary might help to make the differences and then take decisions. Moreover, some works [3] proposed an additional step to OCR operation consisting of a manual evaluation based on error probabilities evaluation. The method employs an error estimator neural network that learns to assess the error probability of every word from *ground-truth* data. The estimated error probability is used to decide which words are inspected by humans.

#### 4.4 Countermeasures against automatic extraction

**Acquisition** : they are the same measures as the third step of a screenlogger.

**Segmentation** : the bigger the image is, the longer and difficult the segmentation will be, so it is preferable not to give indications about relevant data location

to the OCR software. The virtual keyboard area gives an indication, and if in addition there is the cursor of the mouse on the keys composing the password in the screenshots, the indication will be much too precise and would greatly facilitate the OCR's task, because it will merely segment the very specific areas corresponding to the virtual keyboard on which the cursor appears.

**Recognition :** finding countermeasures to the algorithms presented in section 4.3 requires a concrete implementation of the different methods and some testing. This will be the next focus of our work. However, a first step would be to analyse the strengths and weaknesses of each algorithm to get some insight about possible countermeasures. This is done in table 2.

Table 2: Strengths and weaknesses of the main OCR techniques

Approaches		Advantages	Disadvantages
Template Matching		Requires only little information about the forms to recognize.	<ul style="list-style-type: none"> <li>– Long processing time.</li> <li>– Sensitivity to noise</li> </ul>
Structural		Fast and does not require many learning examples.	<ul style="list-style-type: none"> <li>– The determination of the characteristic attributes can be quite difficult.</li> <li>– Difficult to extract the primitives directly from the physical representation of forms.</li> </ul>
Statistical	Neural networks	<ul style="list-style-type: none"> <li>– The ability to generalize from training examples.</li> <li>– Fast testing step.</li> <li>– Fixed problem size.</li> </ul>	<ul style="list-style-type: none"> <li>– Learning time can be very long.</li> <li>– Over-fitting problem.</li> </ul>
	SVM	<ul style="list-style-type: none"> <li>– Avoid over-fitting problem.</li> </ul>	<ul style="list-style-type: none"> <li>– The high cost of memorizing support vectors that are numerous.</li> <li>– Recognition phases is quite slow.</li> </ul>
	HMM	<ul style="list-style-type: none"> <li>– Allow segmentation and recognition at the same time.</li> <li>– Short processing time.</li> </ul>	<ul style="list-style-type: none"> <li>– Not effective for isolated characters classification.</li> <li>– This method limited by the risks of discontinuity of contours.</li> </ul>
	K-nearest neighbor	<ul style="list-style-type: none"> <li>– Effective when there are enough learning examples.</li> <li>– Easy to implement.</li> </ul>	<ul style="list-style-type: none"> <li>– Low classification speed due to the large number of distances to calculate.</li> </ul>
	Bayesian classifier	<ul style="list-style-type: none"> <li>– Fast and easy to implement</li> </ul>	<ul style="list-style-type: none"> <li>– Long learning time in order to compute class probabilities.</li> </ul>

Template matching relies on a distance calculation between the captured character and the models. Therefore, a countermeasure would aim to maximise this distance. We can think of two main ways of achieving this. The first one would be to introduce noise in the image (replacing some black bits by white ones and whites by blacks). Such a method was developed in [26].

The second way would be to use different fonts such that the distance between the captured character and the model would be significant even if it is the same character. In turn, OCRs can try to encompass the largest possible number of fonts for each character, but that would imply an intensive use of memory on the victim's machine. This potentially intensive use of resources can be a major disadvantage and might be exploited by the user to detect the screenlogger.

The same problem can be found with the structural approach. Indeed, there are many ways of writing the same character and therefore many representations in terms of geometric primitives. To be accurate, the algorithm will have to memorise numerous vectors of primitives for each character, which leads to the same problem of resources consumption and ease of detection.

The large number of labelled examples to memorise and distances to calculate for KNN classification implies both a high memory cost and a long running time. Similarly, SVM requires to memorise all the support vectors and to determine the position of the points to classify with respect to the boundaries.

It is also possible to avoid the computational resource consumption problem by combining these methods with lighter ones such as Bayesian classifier or neural networks.

**Post-treatment :** banks must forbid the use of simple passwords that could be found in dictionaries. Indeed, an application forcing the user to use long passwords, containing special characters and not found in the dictionaries can make the step of post-processing OCRs useless.

## 5 Conclusion

Among all vulnerabilities to which a bank or its customers are exposed, there is certainly one more prejudicial than the others; it is the theft of confidential data and money. One of the most used attacks for this is known as key/screenloggers. It aims at stealing confidential information from users by recording the keystrokes. This paper has presented the main steps of the screenloggers process, from capturing screenshots to the extraction of relevant information by insisting on the operating mode of the different types of OCRs. Some countermeasures corresponding to each stage were given and discussed, but it turns out that there is no completely effective solution against this malware. Thus, there are still many possible improvements to be made to the techniques used by these countermeasures and the safest way to defend against screenloggers seems to be using several countermeasures covering the largest number of stages of their operating mode.

## References

1. M. Agarwal, M. Mehara, R. Pawar, D. Shah. "Secure authentication using dynamic virtual keyboard layout", Proceedings of the International Conference and Workshop on Emerging Trends in Technology" ,ISSN 2349-516, Volume 2, February 2011.
2. Ankit Parekh, Ajinkya Pawar, Pratik Munot and Piyush Mantri, 2011. Secure Authentication using Anti-Screenshot Virtual Keyboard, International Journal of Computer Science Issues, 8(5): 3, September 2011.
3. A. Abdulkader and M. R. Casey, "Low Cost Correction of OCR Errors Using Learning in a Multi-Engine Environment," *2009 10th International Conference on Document Analysis and Recognition*, Barcelona, 2009, pp. 576-580.
4. Shahab Bakhtiyari; Ummair Tahir, Phishing attacks and solutions, 2010.
5. Cengage Learning, Chapter2 : Malware and Social Engineering Attacks. 2011.
6. Dadkhah, Mehdi, et Mohammad Davarpanah Jazi. 2014. Secure payment in E-commerce : Deal with Keyloggers and Phishings. International Journal of Electronics Communication and Computer Engineering 5 (3) : 656-660.
7. Dheeraj Bansal <https://www.shoutmeloud.com/online-virtual-keyboard-secure-passwords-from-keyloggers.html>, 2014.
8. N. Echallier, G. Grimaud and al. "Virtual keyboard logging counter-measures using common fate's law". International Conference on Security and Management (SAM'17). Las Vegas, USA, July 17 ? 20 2017.
9. Ryan M. Gerdes(B) and Saptarshi Mallic, Physical-Layer Detection of Hardware Keyloggers, Springer International Publishing Switzerland H. Bos et al. (Eds.): RAID 2015, LNCS 9404, pp. 26-47, 2015.
10. Ezequiel Guerra, Keyloggers: A Threat to Your Data, 2011.
11. Nikolay Grebennikov, <https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/>, Mars 2007.
12. Hemita Pathak, Apurva Pawar, Balaji Patil, "A Survey on Keyloggers: A malicious Attack", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) Volume 4 Issue 4, April 2015
13. Tom Olzak, Keystroke Logging (Keylogging), April 2008.
14. William lopez, humberto guerra, enio pena, erick barrera, juan sayol, Keyloggers ETHICAL HACKING, 2014.
15. Navjeet Kaur, A Survey on Online Banking System Attacks and its Countermeasures, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.3, March 2015.
16. Gunter Ollmann, Director of Security Strategy, The Phishing Guide Understanding and Preventing Phishing Attacks, 2014.
17. Lim, Johnny, Defeat Spyware with Anti-screen Capture Technology Using Visual Persistence. In Proceedings of the 3rd Symposium on Usable Privacy and Security, 147-148. SOUPS 07. New York, NY, USA : ACM, 2007.
18. Michael Roche, Wireless Hacking Tools, 2007.
19. magazine numerique XMCO, Cybercriminalite keylogger botnet attaques, 2011.
20. Afef Kacem Echi, Abdel Belaid, Impact of features and classifiers combinations on the performances of Arabic recognition systems. 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 85-89.
21. R. M. J. S. Bautista, V. J. L. Navata, A. H. Ng, M. T. S. Santos, J. D. Albao and E. A. Roxas, "Recognition of handwritten alphanumeric characters using Projection Histogram and Support Vector Machine," 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, 2015, pp. 1-6.

22. T. K. Das, A. K. Tripathy and A. K. Mishra, "Optical character recognition using artificial neural network," 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2017, pp. 1-4.
23. B. B. Kumar, M. Bansal and P. Verma, "Designing of Licensed Number Plate Recognition system using hybrid technique from neural network template matching," 2015 International Conference on Computing, Communication and Security (ICCCS), Pamplemousses, 2015, pp. 1-6.
24. Lu T., Palaiahnakote S., Tan C.L., Liu W. (2014) Character Segmentation and Recognition. In: Video Text Detection. Advances in Computer Vision and Pattern Recognition. Springer, London.
25. Chaudhuri A., Mandaviya K., Badelia P., Ghosh S.K. (2017) Optical Character Recognition Systems. In: Optical Character Recognition Systems for Different Languages with Soft Computing. Studies in Fuzziness and Soft Computing, vol 352. Springer, Cham.
26. Bacara, Christophe et al. "Virtual Keyboard Logging Counter-Measures Using Human Vision Properties." 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security.
27. B. S. Jeng, M. W. Chang, S. W. Sun, C. H. Shih and T. M. Wu, "Optical Chinese character recognition with a hidden Markov model classifier-a novel approach," in Electronics Letters, vol. 26, no. 18, pp. 1530-1531, 30 Aug. 1990.
28. S. Malakar, S. Halder, R. Sarkar, N. Das, S. Basu and M. Nasipuri, "Text line extraction from handwritten document pages using spiral run length smearing algorithm," 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS), Kolkata, 2012, pp. 616-619.
29. Abukar Yahye, Maarof Mohd, Hassan Fuad, Muse Abshir Mohamed. (2014). Survey of Keylogger Technologies. International Journal of Computer Science and Telecommunications. 5. 25-31.
30. D. Damopoulos, G. Kambourakis, and S. Gritzalis. 2013. From keyloggers to touchloggers: Take the rough with the smooth. Comput. Secur. 32 (February 2013), 102-114.
31. Preeti Tuli, Priyanka Sahu. System Monitoring and Security Using Keylogger. International Journal of Computer Science and Mobile Computing. IJCSMC, Vol. 2, Issue. 3, March 2013, pg.106 111.
32. Sonal Chawla, Meenakshi Beri, Ritu Mudgi. Image Compression Techniques: A Review. International Journal of Computer Science and Mobile Computing. IJCSMC, Vol. 3, Issue. 8, August 2014, pg. 291296.
33. Palvee Sharma, Rajeev Mahajan. A REVIEW ON COMPRESSION TECHNIQUES WITH RUN LENGTH ENCODING. International Journal of Application or Innovation in Engineering and Management (IJAEM). Volume 2, Issue 8, August 2013.