Using Deep Learning with Attention to Detect Data Exfiltration by POS Malware

Gabriele Martino, Federico Galatolo, Mario Cimino, Christian Callegari

This is a preprint. Please cite using:

```
@article{using2023,
    author={Martino, Gabriele and Galatolo, Federico and Cimino, Mario and Callegari, Christian},
    title={Using Deep Learning with Attention to Detect Data Exfiltration by POS Malware},
    journal={Proceedings of the 25th International Conference on Enterprise Information Systems},
    year={2023},
    volume={},
    pages={},
    publisher={SCITEPRESS - Science and Technology Publications},
    doi={10.5220/0011993900003467},
    issn={},
}
```

Gabriele Martino, Federico Galatolo, Mario Cimino, Christian Callegari. "Using Deep Learning with Attention to Detect Data Exfiltration by POS Malware" Proceedings of the 25th International Conference on Enterprise Information Systems (2023): .

Paper draft - please export an up-to-date reference from
http://www.iet.unipi.it/m.cimino/pub

Using Deep Learning with Attention to Detect Data Exfiltration by POS Malware

Gabriele Martino[1], Federico Andrea Galatolo[1][a], Mario G. C. A. Cimino[1][b] and Christian Callegari[2][c]

[1]*Dept. Information Engineering, University of Pisa, L.go Lazzarino 1, 56122, Pisa, Italy*

[2]*Quantavis s.r.l., L.go Spadoni, 56126 Pisa, Italy*

Keywords: POS Malware, RAM Scraper, Anomaly Detection, Malware Traffic Data, Self-Attention, Transformer.

Abstract: In recent years, electronic payment through Point-of-Sale (POS) systems has become popular. For this reason, POS devices are becoming more targeted by cyber attacks. In particular, RAM scraping malware is the most dangerous threat: the card data is extracted from the process memory, during the transaction and before the encryption, and sent to the attacker. This paper focuses on the possibility to detect this kind of malware through anomaly detection based on Deep Learning with attention, using the network traffic with data exfiltration occurrences. To show the effectiveness of the proposed approach, real POS transaction traffic has been used, together with real malware traffic extracted from a collection of RAM scrapers. Early results show the high potential of the proposed approach, encouraging further comparative research. To foster further development, the data and source code have been publicly released.

1 INTRODUCTION

In the last years, card and contactless payments sensibly increased, with card payments representing 51% of all payments (UK Finance, 2020). Moreover, the gap between credit and debit cards is closing, with cash usage continuing to decline (creditcards.com, 2021). It is predicted that, by 2025, as many as 75% of all transactions will be made without cash (finance-magnates.com, 2016).

RAM scraping is behind many of the major POS attacks (Caldwell, 2014). Another kind of attack is POS Skimmers (d3security.com, 2017): the cyber-criminal places an 'overlay' skimmer on top of the card reader and pin pad, to steal the data later.

RAM scraping malware is the most dangerous attack on POS systems, because a malware could be easily installed via social engineering or phishing, to exfiltrate customers' data through the Internet (Trend-Micro, 2015). When a customer's card is swiped on the POS device using the magnetic stripe, the card and owner's information present in Track 1 and Track 2 are momentarily stored in the process memory of the payment system. The POS malware steals this data before it is encrypted and deleted, and sends it to the cyber criminals.

There are many ways to exfiltrate customers' data: by sending it to a fictitious server via SMTP or FTP protocol, or to C&C servers via HTTP POST/GET/Header, RDP (Remote Desktop Protocol), or via TOR protocol in sophisticated malware (TrendMicro, 2015).

Malware types have been extensively classified in (Rodríguez, 2017) (Cimino et al., 2020), considering persistence method, protection method, functionality, how data are exfiltrated, and how data are ciphered. Since data exfiltration is characterized by well-defined behavioral patterns, the related malicious activity is easily detectable by network monitoring tools.

There are four main methods of traffic classification: port-based, deep packets inspection (DPI), statistical-based and behavioral-based (Biersack et al., 2013). The accuracy of port-based methods is very low nowadays, because of the common use of random ports and port disguises. On the other hand, DPI-based methods encounter great difficulties because they are unable to decrypt the traffic. The current research mainly focuses on statistical-based methods and behavioral-based methods. Both methods are based on machine-learning approaches (Azab

[a] https://orcid.org/0000-0001-7193-3754
[b] https://orcid.org/0000-0002-1031-1959
[c] https://orcid.org/0000-0001-7323-8069

Martino, G., Galatolo, F., Cimino, M. and Callegari, C.
Using Deep Learning with Attention to Detect Data Exfiltration by POS Malware.
DOI: 10.5220/0011993900003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 1*, pages 638-648
ISBN: 978-989-758-648-4; ISSN: 2184-4992
Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

et al., 2022). Deep Learning approaches try to solve the overly engineered features that need to be extracted from the flow of packets. Many methods have been proposed, both supervised and unsupervised: 1D-CNN (Convolutional Neural Network), 2D-CNN, RNN (Recurrent NN) + CNN, LSTM (Long Short-Term Memory) Autoencoder, Combination of LSTM + CNNs (Wang et al., 2017a), (Zhou et al., 2017), (Chen et al., 2017), (Aceto et al., 2020), (Lopez-Martin et al., 2017), (Liu et al., 2019), (Aceto et al., 2019), (Wang, 2015), (Mirsky et al., 2018), (Wang et al., 2018), (Cimino. et al., 2022).

For validation purposes, in this paper, a dataset of exfiltration occurrences has been created from real samples of POS RAM scraping malware, and from transactions of real POS systems. As anomaly detection approaches, deep learning models with attention have been developed. Early results show the high potential of the proposed approach, encouraging further comparative research. To foster further development, the data and source code have been publicly released on GitHub (Martino, 2023).

The paper is organized as follows. Section 1 describes related works. The fundamental behavior of POS systems is shown in Section 3. Section IV explains how POS RAM scraping malware works. Section 5 shows some methodologies of application classification using network traffic with deep learning models. Datasets and Deep Learning architectures are detailed in Section 6, together with some early experimental studies. Results are discussed in Section 7. Conclusions are drawn in Section 8.

2 RELATED WORK

The following research works represent the main approaches in the literature. In (Wang et al., 2017b) a CNN has been used to classify Normal and Malware traffic. The traffic has been taken from (CTU University, 2016). However, no POS malware traffic has been used in the experimental studies. In (Bader et al., 2022) a similar approach has been proposed, building a more complex model, via different kinds of features extracted from the traffic samples. In (Shaikh and Shashikala, 2019) a pipeline made by an autoencoder and an LSTM has been used to classify Normal and DoS attack traffic. In (Anderson and McGrew, 2017) an interesting analysis of issues related to the adoption of machine learning on traffic data for detecting malware is presented: from mislabelling of data to non-stationarity of the network traffic. In (Marín et al., 2021) a combination of 1D-CNN and LSTM networks has been experimented for classifying the traffic of three types of malware, using Raw Packets and Raw Flows. In (Lichy et al., 2023) a comparison between classic ML-based and DL-based solutions is made, showing that not necessarily the DL ones outperform to classify malware traffic.

A major challenge in the literature is the huge amount of different types of malware present nowadays. For this reason, training a single DL model for classification could lead to obsolescence quite soon. Moreover, it has been shown that, although complex models could get high performance, simple models perform similarly. Last but not least, it is well known that the difficulty of labeling malicious traffic data leads to noisy datasets. The purpose of this research is to develop models that are robust to identify suspicious traffic, without recognizing the specific malware.

To the best of our knowledge, the literature lacks research works focused on traffic data extracted from RAM scraping malware, in terms of both approaches and available benchmark data. For this reason, in this research, some malware samples available in (Rodríguez, 2017) have been used to create a dataset for testing purposes. A deep learning approach based on autoencoders with an attention mechanism has been used to exploit only normal traffic for training purposes.

3 POINT-OF-SALE SYSTEMS

This section focuses on how a POS system handles transaction data flow. Fig.1 shows an overview of the transaction flow of a card payment (FirstData, 2010) (Rodríguez, 2017). Specifically, the customer inserts his card into the merchant's payment system through a POS terminal. The related data is sent to the acquirer bank, which carries out a routing to a card payment brand circuit (e.g., VISA, MasterCard, or American Express). Then, the related issuer bank verifies the card legitimacy, i.e. not reported as stolen or lost, and verifies that the customer's account has enough funds/credit available to pay. If that is the case, the issuer bank generates an authorization number and routes it back to the card payment brand, which forwards it to the acquirer bank. The acquirer bank then forwards it to the merchant, which concludes the sale with the customer, providing her with an acknowledgment (normally in terms of a receipt) (Rodríguez, 2017).

There exist different kinds of POS systems, which can be classified on the basis of the interface, i.e., an external interface with respect to the transaction processing system, or an integrated interface, such as a

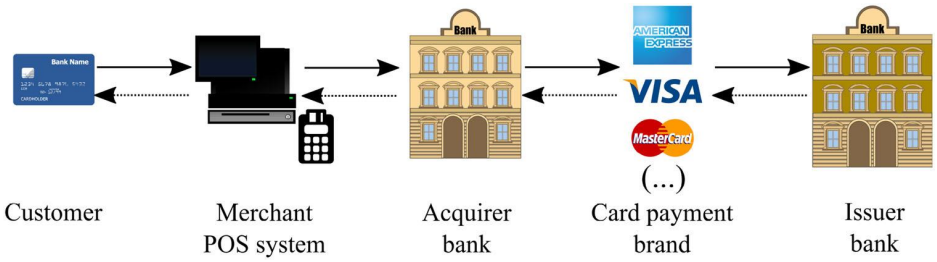


Figure 1: An abstraction of Point-of-Sale card transaction flow (extracted and adapted from (Rodríguez, 2017) (FirstData, 2010)).

mobile app on a smartphone (connectpos.com, 2020) (fitsmallbusiness.com, 2022) (intel.com, 2022). In (Gomzin, 2014) it is stated that at least three of the vulnerabilities of POS systems are located where the customer data may reside: (i) *in memory*: data manipulations are carried out by the payment application when processing an authorization or a settlement, and thus, payment card data remains in the memory of the processing machine; (ii) *at rest*, i.e., when the payment application stores data on a disk device, either temporarily or for a long term; (iii) *in transit*, when payment data are received and sent to and from other application and devices within the system.

In modern credit cards, the data stored is accessed by four different interfaces: by physical access, by magnetic stripe, by a chip reader (EMV, i.e., Europay, MasterCard and Visa), or by an NFC reader. The spread of the usage of a certain kind of interface with respect to others strongly depends on the country. EMV introduced a way to authenticate chip-card transactions and to minimize the magnetic stripe card counterfeiting fraud, but it is mainly spread in the EU, and less used in USA (Secure Technology Alliance, 2014) (Symantec, 2014).

Data provided by physical access to the card is well known: Name, Expiration Date, Credit Card Number, and Card Verification Value (CVV/CVV2). The card magnetic stripe, located on the back, is horizontally divided into three tracks. Track 1 and Track 2 contain similar data, but different formats, both standardized in ISO/IEC 7813 (ISO/IEC 7813:2006, 2006). Track 3, also called THRIFT, was originally intended for use with Automatic Teller Machines. NFC is a bidirectional short-range (less than 10 cm) contactless communication technology, operating on the 13.56 MHz spectrum, based on two Radio Frequency Identification (RFID) standards. Namely, contactless payment cards follow the ISO-14443 (ISO/IEC 7813:2006, 2013) standard. Security with NFC is debatable since a contactless card can communicate with any NFC reader, without any identification of it. Hence, a contactless card's track transmits private customer information once communication is established. In (Chabbi et al., 2022) a classification of possible attacks is reported, such as: Eavesdropping, Relay Attack, Replay attack, also Skimming, Cloning and Malware as well.

Considering the variety of card interfaces and POS systems as well as OSs on which they are based, the vulnerability of these kinds of payment methods could be everywhere, so it is crucial to develop systems that are able to detect any of these attempts of attack.

To the best of our knowledge, POS RAM scraper malware mainly searches for Track 1 and Track 2 data, given the vast diffusion of magnetic stripe interfaces for payment in the USA in the past years. However, there is the possibility that in the future this malware can steal other information deriving from different interfaces and for all Operating Systems ((Bodhani, 2013)).

4 POS RAM SCRAPING MALWARE

Point-of-Sale RAM scraping malware is a malicious software that, once installed in a POS system, usually based on Windows OS, steals credit/debit card data such as Track 1 and Track 2. The malware uses a multitude of techniques to collect data: it iterates over all running processes, uses a blacklist to avoid scanning where Track 1 and Track 2 cannot be found, uses regex matching techniques, encodes data in base64 to obfuscate their content. The malware can also differentiate over several data exfiltration methods: manually removed, HTTP POST, FTP Server, HTTP HEADER, TOR, SMTP Protocol. Some well-known names are: alina, Dexter, BlackPOS, Soraya and many others (Trend Micro, 2014). Cybercriminals register fake domains for data-exfiltration purposes with hosting providers in countries with lax Internet law enforcement, such as Russia and Romania, among others. These fake domains act like man-in-the-middle (MitM) data collectors. The Tor network conceals C&C servers' IP addresses and, by default,

encrypts all traffic. The C&C servers' addresses end with a *.onion* Top Level Domain, which cannot be resolved outside the Tor network and can only be accessed using a Tor proxy application. ChewBacca malware makes use of this functionality. Cybercriminals use compromised email accounts to exfiltrate stolen data. A command line email client invoked through a batch script may be used to exfiltrate stolen data as an attachment. BlackPOS makes use of this functionality. Cybercriminals create accounts on FTP servers that are hosted in countries with lax Internet law enforcement. Malware such as BlackPOS or BrutPOS log in to FTP servers using hardcoded credentials and copy over the stolen data (Trend Micro, 2014). Since some of these protocols can be blocked by the firewall of the POS system, malware is evolving as well. An example is the usage of the DNS protocol: it cannot be blocked for normal functioning of a device connected to the internet. Multigrane malware encrypts with a 1024-bit RSA key the stolen payment card data, then it passes data through a Base32 encoding process. The resulting encoded data is used in a DNS query for log.[encoded_data].evildomain.com, where "evildomain" is a domain name controlled by the attackers (computerworld.com, 2016) (securebox.comodo.com, 2016).

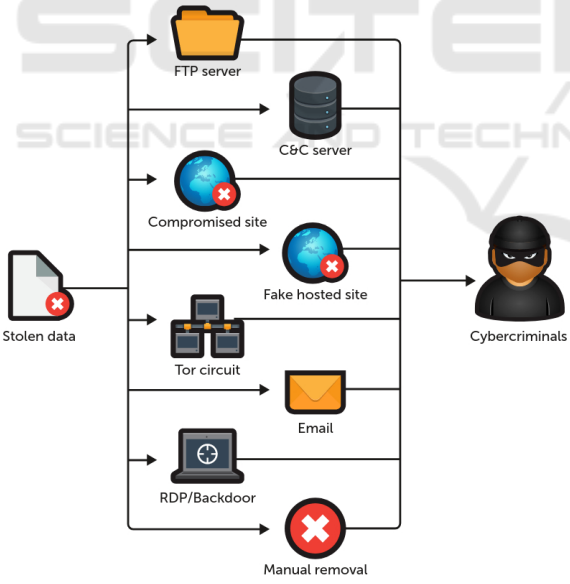


Figure 2: Data-exfiltration techniques observed among PoS RAM scrapers (Trend Micro, 2014).

5 TRAFFIC CLASSIFICATION VIA DEEP LEARNING

The most two common choices of traffic representation are session and flow (Dainotti et al., 2012). A flow is made by all the packets for which the following 5-tuple match: source IP, source port, destination IP, destination port and transport-level protocol. A session is made by combining the two flows in the opposite direction, i.e. the source and destination IP / port are swapped. In (Wang et al., 2017a) both flows and sessions are taken into account, and packets information relates to Layers 7 and 4 of the ISO/OSI model. Then, it takes the first 785 bytes of each packets and builds the temporal series. The research work compares 1D-CNN and 2D-CNN. In (Chen et al., 2017) statistical features are extracted from sessions, then a pseudo-image is created from it and a CNN classifier. In (Aceto et al., 2020) the first 784 or 576 bytes of L4 payload and a combination of CNNs are used. In (Lopez-Martin et al., 2017) flows are extracted for each packet using only six features: source port, destination port, number of bytes in payload, TCP windows size, interrarival time and direction of packet. Then the flows are divided in batch of 20 packets. At the end, they test 2D-CNN, RNN (LSTM) and a combination of the two models. In (Liu et al., 2019) an Autoencoder made of GRUs (RNN) is used to extract relevant features from raw flows, then dense layers are used for classification. In (Aceto et al., 2019) sessions are directed to two different models: a certain number of bytes of the L4 payload, to a 2D-CNN and some fields of the packets to a RNN, then combines the features in a final layer for classification. In (Yang et al., 2020) the TCP flag is reported as quite informative, especially in the intra-session correlation. In (Wang et al., 2018) a series of features are extracted from packets, and then proposed to a Stacked Autoencoder model to extract higher order features from the first 144 bytes of the packets.

6 METHOD

6.1 Traffic Datasets

The dataset is divided in Normal traffic and Malware traffic. Fig. 3 shows how the two traffic data have been sniffed. A stand-alone Android-based POS is connected to a WiFi hotspot made by a laptop connected to an access point. On the laptop, that acts like a bridge, Wireshark was activated to sniff all the traffic from and towards the mobile POS. Around a hundred of actual card transactions were made without really charge any money.

The Malware traffic is extracted from a group of POS RAM scraping sample shown in Table 1 downloaded from http://webdiis.unizar.es/. These mal-

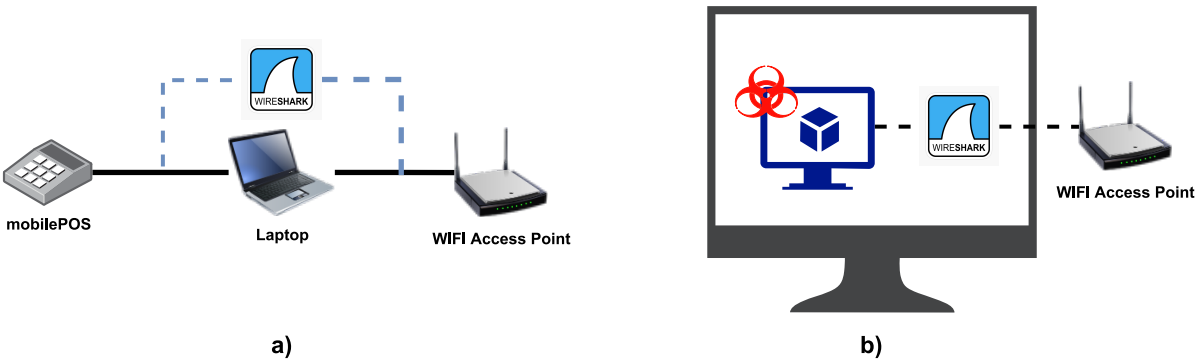


Figure 3: A) The mobilePOS is connected to a laptop as HotSpot, the laptop is connected to an Access Point. The packets are sniffed from the laptop. b) An infected virtual machine is connected to internet through the host machine. Wireshark sniffs packets from the VM.

waretypes are installed in a Virtual Machine with Windows 7 installed on it.

Since this kind of malware are able to scan in memory process to find Track 1 and Track 2 format, before installing these samples, we executed a credit card number generator (github.com/bizdak/ccgen) that was always running before the installation of the malware. This Track 1 and Track 2 generator produces a valid card number every 0.2 seconds.

To avoid that different malware traffic overlaps each other, they have been installed and then removed before the installation of the next one. We then filter out background packets from the actual exfiltration traffic generated from the malware.

Table 1: POS RAM Scraping Malware sample.

Malware	Selected Sample
alina	1efeb85c8ec2c07dc0517ccca7e8d743
backoff	05f2c7675ff5cda1bee6a168bdbecac0 6a0e49c5e332df3af78823ca4a655ae8
blackpos	0ca4f93a848cf01348336a8c6ff22daf 7f1e4548790e7d93611769439a8b39f2
decebal	46185a6ec6d527576248ef65a82b891d 91100e23e59d5744a5720a6f84b68d99
frameworkPOS	a5dc57aea5f397c2313e127a6e01aa00 b57c5b49dab6bbd9f4c464d396414685
getmypassPOS	1d8fd13c890060464019c0f07b928b1a
jackpos	00b09796519c60c7369290f19f89cd10
lusypos	bc7bf2584e3b039155265642268c94c7
soraya	1483d0682f72dfefff522ac726d22256 1661aab32a97e56bc46181009ebd80c9

Table 2: Benign and Malign Network Traffic Data Sample.

Traces	Number of Packets
Benign	62923
Malign	1132

Fig. 4 and Fig. 5 show the distribution of the protocols of the traffic of the two datasets. At first, is interesting to note that almost all the traffic outcoming from the mobilePOS is HTTPS. Instead, the larger amount of traffic from the malware dataset is made by DNS packets. Part of this DNS traffic derives from all the malwares that check for server domains, and a larger part from some malwares that uses this exfiltration method. This approach splits the card number in small chunks of bytes and send them as DNS requests.

It's worth mentioning that in some POS systems certain protocols could be blocked from the firewall, so keeping safe the data. For this reason many of these malware migrate towards DNS approach, because it is necessary for the normal function of internet and hence cannot be blocked.

POS systems based on different OSs could have different vulnerabilities and different traffic patterns. Our experiments mixed a real transaction traffic from a POS Android-based and real malware traffic installed on a Windows OS. We assume that besides the POS OS, these are the exfiltration methods that can be implemented and should be detected, regardless the OS. Moreover, our challenge is to analyse the capabilities of some models to distinguish the two kind of traffic, especially on the overlapping protocols: HTTPS and DNS.

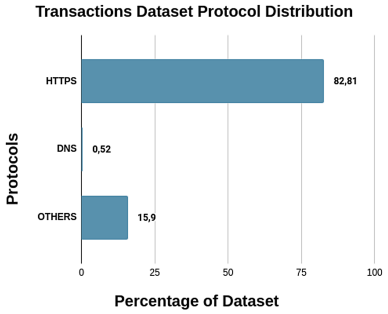


Figure 4: Protocol data distribution in Transaction Dataset.

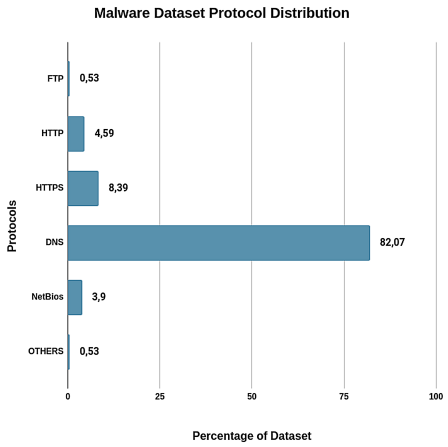


Figure 5: Protocol data distribution in Malware Dataset.

6.2 Traffic Preprocessing

The sessions (also called biflows) are extracted from the two traffic data samples. After removing the IPs, for each packet the following data was extracted: source Port, destination Port, ACK flag, PUSH flag, RESET flag, SYN flag, FYN flag, interrarival time, no. of bytes of payload, TCP window size. All flags are zero in the case of UDP packet. Then, each session is split into sub-sessions of different time windows: 5, 10, 15, 20 packets per session. Each of these features is normalized in [0, 1]. When splitting session, if the number of packets in the time series is less than the window size, the remaining size is pad with zeros. Features from (Lopez-Martin et al., 2017) and (Yang et al., 2020) have been combined.

6.3 Deep Learning Models

All the considered models follow the Autoencoder approach: the model is trained to reproduce as an output the same time series provided in input. The Normal dataset is used as training set, in our case is the one with the mobile-POS transaction.

6.3.1 LSTM Autoencoder Architecture

In (Sutskever et al., 2014) and (Cho et al., 2014) an Encoder-Decoder model with LSTMs is proposed to encode a time series (e.g. a sentence) in a single fixed-lenght vector. Then, this vector decodes it into a another time series that could be the future time-steps or a translated sentence, if purposely trained. If the training is to reproduce the same time-series, we could use this model as an Anomaly Detector. In (Wei et al., 2022) and (Srivastava et al., 2015), the authors report details of the used architecture.(Said Elsayed et al., 2020) reports an example of usage of this model for anomaly detection in network traffic. An Autoencoder has been implemented, for which Encoder and Decoder have two layers where the vector size of the latent space in the outer layers is doubled respect to the internal layers.

6.3.2 LSTM Autoencoder with Attention Mechanism

An attentional mechanism has been used to improve neural machine translation by selectively focusing on parts of the source sentence during translation (Luong et al., 2015). The idea of Global Attentional model is to consider all the hidden states of the encoder when deriving the context vector c_t (Luong et al., 2015). The context vector can be computed as weighted sum of the hidden states of the encoder at any time-steps (Bahdanau et al., 2014). The weights come in general from a *Softmax* function, in which the *Scores* of the mutual combination of the hidden states (in case of self-attention). In (Luong et al., 2015) three different alternatives for score function have been proposed:

$$score(h_x, h_y) = \begin{cases} h_x^T h_y & dot \\ h_x^T W_a h_y & general \\ v_a^T tanh(W_a[h_x; h_y]) & concat \end{cases} \quad (1)$$

The Autoencoder with attention mechanism has the same architecture of the LSTM Autoencoder, but the input of the decoder is made by context vectors, derived from self-attention mechanism of the hidden states of the encoder. Two versions for the two score functions have been developed: *dot* and *general*.

6.3.3 Transformer

(Vaswani et al., 2017) is a paper influencing the literature, which shows that by stacking multiple attention layers alone, the model is able to learn effectively high correlated temporal sequence but it is also capable of solving complex tasks in zero-shot fashion (Galatolo. et al., 2022). BERT and GPT-3 are just examples (Devlin et al., 2018) (Brown et al., 2020).

A simple transformer model in Autoencoder-like fashion has been developed: the output needs to be equal to the input and the input projection is masked, where the mask is learned. All the tested transformers have a number of heads of 4 and a depths of 6, set as hyperparameters.

6.4 Experimental Setup

The Pytorch Lightning Framework has been used which runs on Ubuntu 22.04 64bit OS. The machine is equipped with 16 CPUs and 32GB of memory.

An Nvidia Geforce RTX 3070 is used as accelerator. Mini-batch size of 128, cost function of L1 with 'sum' as reduction method. Adam as optimizer, 0.0008 as learning rate, 300 epochs and early stopping as overfitting avoiding method with 30 epochs as patience.

7 EXPERIMENTAL RESULTS

Table 3 reports the results of the experimentation. After the training, the threshold that reach the best F1 score is searched. It's possible to use several euristics to find the best threshold. The average of the distribution of the loss has been calculated. For Benign and Malign traffic, for the threshold that give the best F1 score. We used the LSTM AE model as performance reference and it's possible to see that the results are quite variable. In (Said Elsayed et al., 2020) similar results reported.

We can assert that all the attention-based models have much more stable and predictable results. Interestingly, all the attention-based models increase their F1 score with the increasing of the length of the session. Moreover, increasing the hidden size we see also a small increasing of performance.

It's important to note that the performance of these models is quite influenced from the choice method of threshold, to label a certain loss as malware or not. Furthermore, given the similarity of the dynamics of malware traffic with the normal ones, this is pretty hard problem to solve. In fact, training on certain traffic helps the model to reconstruct the traffic of malware data exfiltration as well.

The goodness of the models depends on how well the loss distribution of Normal traffic is separable from the Malware traffic one.

Fig. 6 shows an histogram of the loss distributions of the testset made from the malware traffic and a sample of the transactions traffic. In this example we notice a good separation of the two loss distribution.

In addition, it can be noticed that given the unbalancing of the protocols in the two datasets, the models, and so the results, could be biased to distinguish between the two majority protocols (HTTPS and DNS) instead of the two kind of traffic source, Transactions and Malware; see Fig. 5 and Fig. 4. To better analyse the performance of the models, we extracted from the traffic sources only the packets corresponding the predominants protocols which are in our case HTTPS and DNS. Then we analyse how much the models are able to distinguish between Normal and Malware Traffic over the same protocols.

Fig. 7a and Fig. 7b show the performance to distinguish the two majority protocols of the datasets. In both the plots we measured the F1-score for all the four window size and the four models. To make a fair comparison we fixed the hidden size to 8. We remember that the transformer has a more complex set of hyperparameters but here we just considered the size of internal features.

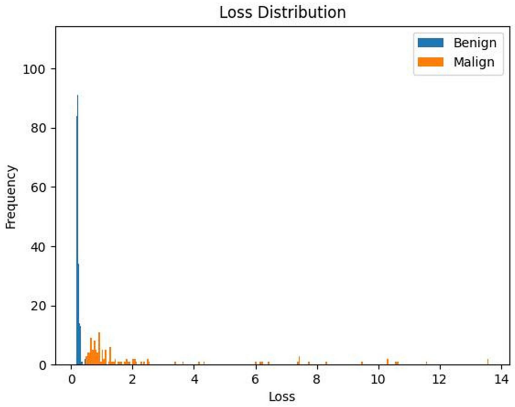


Figure 6: Loss Distribution of Attention-LSTM AE - General Score - hidden size: 128, Window Size: 20.

It's possible to notice that the actual good performance is achieved in distinguish Transaction sessions from the Malware sessions, also in overlapping protocols, that is exactly what it was supposed. The size of the session the performance increase with the window size accordingly as previously found.

The transformers seem to struggle for this kind of purpose, and needs larger window size to be more 'embedded' to the traffic type. The LSTM Autoencoder used as reference, seems to have good results as well, but it is suggested the LSTM AE models with attention-mechanism embedded for higher stability of the performance.

We can assess that with HTTPS protocol a larger window size is necessary to effectively detect the malware traffic. This is quite reasonable since this protocol needs from 6 to 10 packets to complete the handshake, that moreover doesn't bring any sensitive information. This fact strengthens the assessment of the affectiveness of the method.

Instead with the DNS protocol, even with small hidden size, we always have F1-score above 0.75, reaching values above 0.90 in average. This is interesting, because even if the DNS data-exfiltration is the most dangerous, since can bypass easily any firewall, seems that there has been no forethought in the development of this protocol to obsfuscate this usage respect to the normal one, resulting in an almost easy detectability.

Table 3: Performance of LSTM AE, LSTM AE with attention: dot, general, Transformer for the four different session size.

		Window Size							
		5		10		15		20	
	Hidden Size	F1	Recall	F1	Recall	F1	Recall	F1	Recall
LSTM AE	8	0.692	1.0	0.794	1.0	0.706	0.95	**0.917**	**1.0**
	16	0.672	0.965	0.289	0.966	0.239	0.95	0.348	0.967
	32	0.816	0.965	0.546	0.966	0.553	0.962	0.693	0.967
	64	0.681	1.0	0.694	1.0	0.377	0.962	0.674	0.967
	128	0.769	1.0	0.723	1.0	0.862	0.962	0.859	0.967
Attention-LSTM AE dot score	8	0.527	1.0	0.824	1.0	0.86	1.0	0.878	1.0
	16	0.781	1.0	0.82	1.0	0.86	1.0	0.878	1.0
	32	0.946	1.0	0.866	1.0	0.856	1.0	0.891	1.0
	64	0.787	1.0	0.751	1.0	0.741	1.0	0.783	1.0
	128	0.855	1.0	0.854	1.0	**0.968**	**1.0**	0.92	1.0
Attention-LSTM AE general score	8	0.506	1.0	0.809	1.0	0.783	1.0	0.871	1.0
	16	0.552	1.0	0.852	1.0	0.61	1.0	0.917	1.0
	32	0.75	1.0	0.522	1.0	**0.968**	**1.0**	0.862	1.0
	64	0.801	1.0	0.729	1.0	0.731	1.0	0.943	1.0
	128	0.856	1.0	0.731	1.0	0.84	1.0	0.967	1.0
TRANSFORMER	8	0.541	1.0	0.629	1.0	0.76	1.0	0.819	1.0
	16	0.575	1.0	0.634	1.0	0.777	1.0	0.825	1.0
	32	0.479	1.0	0.703	1.0	0.819	1.0	0.83	1.0
	64	0.069	0.482	0.142	0.923	0.77	0.962	**0.837**	**1.0**
	128	0.097	0.69	0.328	0.897	0.364	0.988	0.682	1.0

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

8 CONCLUSIONS

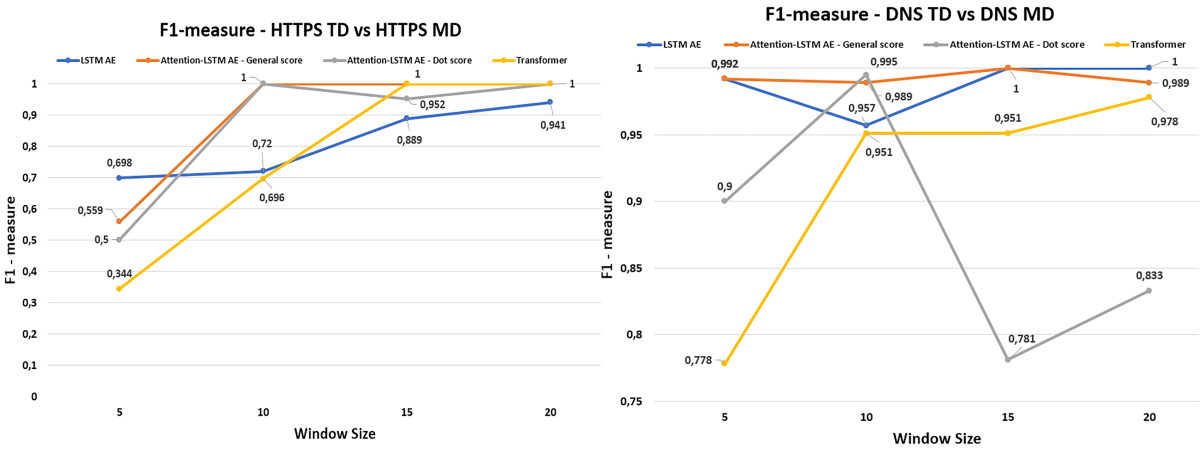
Cashless transactions are becoming always more preponderant in the market. This research brings the devices deed to these money exchanges to be always more targeted from cyberattacks. POS RAM scraping malwares seem to be nowadays still the more dangerous attack to the customers. In this paper it is analized the possibility to detect the data exfiltration methods of these malware using novel DL models. To accomplish this aspect we created two datasets from a real mobile POS device and real POS malwares that try to exfiltrate data from the Track 1 and Track 2 generator process.

As we have shown, these kind of dataset are quite difficult to retrieve and to analyse given the variables that can affect the kind of traffic: topology of network, types of applications, unbalancing of the protocols. For this reason the environment where the POS system is connected to small networks such as the stores ones can be efficiently analysed.

The LSTM Encoder-Decoder model has the known lack to encode also high informative and long time series in a single fixed-size vector that will be later decoded. Attention models try to avoid this keeping all the information of each timesteps and using them with the right re-weighting for the decoder leading to better results.

In our context, we wanted to test if the attention-mechanism brings to higher capacity to retrieve peculiar information from a time series, compact and better distinguish from similar but semantically different network traffic.

Our results suggest that attention in LSTM AE models leads to an higher regularity in the latent space, that could correspond to a better defined distribution of reconstruction loss. Future work could be creating a larger POS malware traffic dataset with all the possible exfiltration methods, to cover all the shades possibilities even for futures attacks of this kind. Morever, it could be interesting to analyse how similar time series such as network traffic are encoded. This could help to better understand which factors or hidden features lead to a differentiation of the two traffics in Explainable AI manner.



(a) (b)

Figure 7: F1 score evaluation of the DNS and HTTPS sessions of Transaction Dataset vs DNS and HTTPS sessions of Malware Dataset. All the models have an hidden size of 8.

ACKNOWLEDGEMENTS

The authors thank Leonardo Cecchelli for his work on the subject during his thesis. Work partially funded by: (i) the Tuscany Region in the framework of the SecureB2C project, POR FESR 2014-2020, Project number 7429 31.05.2017; (ii) PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", funded by the European Commission under the NextGeneration EU programme; (iii) the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence)

REFERENCES

Aceto, G., Ciuonzo, D., Montieri, A., and Pescapè, A. (2019). Mimetic: Mobile encrypted traffic classification using multimodal deep learning. *Computer networks*, 165:106944.

Aceto, G., Ciuonzo, D., Montieri, A., and Pescapé, A. (2020). Toward effective mobile encrypted traffic classification through deep learning. *Neurocomputing*, 409:306–315.

Anderson, B. and McGrew, D. (2017). Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In *Proceedings of the 23rd ACM SIGKDD International Conference on knowledge discovery and data mining*, pages 1723–1732.

Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.-K. R., and Sarsour, M. (2022). Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks*.

Bader, O., Lichy, A., Hajaj, C., Dubin, R., and Dvir, A. (2022). Maldist: From encrypted traffic classification to malware traffic detection and classification. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 527–533. IEEE.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Biersack, E., Callegari, C., Matijasevic, M., et al. (2013). Data traffic monitoring and analysis. *Lecture Notes in Computer Science*, 5(23):12561–12570.

Bodhani, A. (2013). Turn on, log in, checkout. *Engineering & Technology*, 8(3):60–63.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Caldwell, T. (2014). Securing the point of sale. *Computer Fraud & Security*, 2014(12):15–20.

Chabbi, S., El Madhoun, N., and Khamer, L. (2022). Security of nfc banking transactions: Overview on attacks and solutions. In *2022 6th Cyber Security in Networking Conference (CSNet)*, pages 1–5. IEEE.

Chen, Z., He, K., Li, J., and Geng, Y. (2017). Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In *2017 IEEE International conference on big data (big data)*, pages 1271–1276. IEEE.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014).

Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cimino, M. G., De Francesco, N., Mercaldo, F., Santone, A., and Vaglini, G. (2020). Model checking for malicious family detection and phylogenetic analysis in mobile environment. *Computers & Security*, 90:101691.

Cimino., M. G. C. A., Galatolo., F. A., Parola., M., Perilli., N., and Squeglia., N. (2022). Deep learning of structural changes in historical buildings: The case study of the pisa tower. In *Proceedings of the 14th International Joint Conference on Computational Intelligence - NCTA,*, pages 396–403. INSTICC, SciTePress.

computerworld.com (2016). New point-of-sale malware multigrain steals card data over dns. https://www.computerworld.com/article/3059317/new-point-of-sale-malware-multigrain-steals-card-data-over-dns.html.

connectpos.com (2020). Types of pos (point of sale) system for retailers. https://www.connectpos.com/types-of-pos-system-connectpos/.

creditcards.com (2021). Payment method statistics. https://www.creditcards.com/statistics/payment-method-statistics-1276/.

CTU University (2016). The stratosphere ips project dataset. https://www.stratosphereips.org/datasets-malware.

d3security.com (2017). Risks affecting point of sale terminals. https://d3security.com/blog/risks-affecting-point-of-sale-terminals/.

Dainotti, A., Pescape, A., and Claffy, K. C. (2012). Issues and future directions in traffic classification. *IEEE network*, 26(1):35–40.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

financemagnates.com (2016). The future of fintech: the death of cash and bank branches but a boost to retail. https://www.financemagnates.com/fintech/bloggers/fintech-the-death-of-cash-and-bank-branches-but-a-boost-to-retail/.

FirstData (2010). Payments 101: Credit and debit card payments- key concepts and industry issues. http://euro.ecom.cmu.edu/resources/elibrary/epay/Payments-101.pdf.

fitsmallbusiness.com (2022). Types of pos systems: A guide for small businesses. https://fitsmallbusiness.com/types-of-pos-systems/.

Galatolo., F. A., Cimino., M. G. C. A., and Vaglini., G. (2022). Zero-shot mathematical problem solving via generative pre-trained transformers. In *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 479–483. INSTICC, SciTePress.

Gomzin, S. (2014). *Hacking Point of Sale: Payment Application Secrets, Threats, and Solutions*. John Wiley & Sons.

intel.com (2022). Deliver seamless experiences with multiple types of pos. https://www.intel.com/content/www/us/en/internet-of-things/iot-solutions/pos/types-of-pos.html.

ISO/IEC 7813:2006 (2006). Iso/iec 7813:2006-information technology — identification cards — financial transaction cards. https://www.iso.org/standard/43317.html.

ISO/IEC 7813:2006 (2013). Iso/iec 18092:2013 information technology — telecommunications and information exchange between systems — near field communication — interface and protocol (nfcip-1). https://www.iso.org/standard/56692.html.

Lichy, A., Bader, O., Dubin, R., Dvir, A., and Hajaj, C. (2023). When a rf beats a cnn and gru, together—a comparison of deep learning and classical machine learning approaches for encrypted malware traffic classification. *Computers & Security*, 124:103000.

Liu, C., He, L., Xiong, G., Cao, Z., and Li, Z. (2019). Fs-net: A flow sequence network for encrypted traffic classification. In *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*, pages 1171–1179. IEEE.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J. (2017). Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE access*, 5:18042–18050.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Marín, G., Caasas, P., and Capdehourat, G. (2021). Deepmal-deep learning models for malware traffic detection and classification. In *Data Science–Analytics and Applications*, pages 105–112. Springer.

Martino, G. (2023). Ramscrapersattentiondetectors github. https://github.com/GabMartino/RAMScrapersAttentionDetectors.

Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

Rodríguez, R. J. (2017). Evolution and characterization of point-of-sale ram scraping malware. *Journal of Computer Virology and Hacking Techniques*, 13(3):179–192.

Said Elsayed, M., Le-Khac, N.-A., Dev, S., and Jurcut, A. D. (2020). Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '20, page 37–45, New York, NY, USA. Association for Computing Machinery.

Secure Technology Alliance (2014). Emv: Faq. https://www.securetechalliance.org/publications-emv-faq/#q1.

securebox.comodo.com (2016). New multigrain malware eats memory, steals pos data. https://securebox.comodo.com/blog/pos-security/new-multigrain-malware-eats-memory-steals-pos-data/.

Shaikh, R. A. and Shashikala, S. (2019). An autoencoder and lstm based intrusion detection approach against denial of service attacks. In *2019 1st International Conference on Advances in Information Technology (ICAIT)*, pages 406–410. IEEE.

Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Symantec (2014). Attacks on point-of-sales systems. https://docs.broadcom.com/doc/attacks-on-point-of-sale-systems-en.

Trend Micro (2014). Pos ram scraper malware past, present, and future. https://www.wired.com/wp-content/uploads/2014/09/wp-pos-ram-scraper-malware.pdf.

TrendMicro (2015). Defending against pos ram scrapers. https://d3security.com/blog/risks-affecting-point-of-sale-terminals/.

UK Finance (2020). Uk payment markets 2020. https://www.ukfinance.org.uk/policy-and-guidance/reports-publications/uk-payment-markets-2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, K., Chen, L., Wang, S., and Wang, Z. (2018). Network traffic feature engineering based on deep learning. In *Journal of Physics: Conference Series*, volume 1069, page 012115. IOP Publishing.

Wang, W., Zhu, M., Wang, J., Zeng, X., and Yang, Z. (2017a). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE international conference on intelligence and security informatics (ISI)*, pages 43–48. IEEE.

Wang, W., Zhu, M., Zeng, X., Ye, X., and Sheng, Y. (2017b). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)*, pages 712–717. IEEE.

Wang, Z. (2015). The applications of deep learning on traffic identification. *BlackHat USA*, 24(11):1–10.

Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., and Boulic, M. (2022). Lstm-autoencoder based anomaly detection for indoor air quality time series data. *arXiv preprint arXiv:2204.06701*.

Yang, K., Kpotufe, S., and Feamster, N. (2020). Feature extraction for novelty detection in network traffic. *arXiv preprint arXiv:2006.16993*.

Zhou, H., Wang, Y., Lei, X., and Liu, Y. (2017). A method of improved cnn traffic classification. In *2017 13th international conference on computational intelligence and security (CIS)*, pages 177–181. IEEE.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS