Identifying Malicious Botnet Traffic using Logistic Regression

Rohan Bapat, Abhijith Mandya, Xinyang Liu, Brendan Abraham, Donald E. Brown, Hyojung Kang, and Malathi Veeraraghavan
University of Virginia, rb2te, am6ku, xl9qw, bea3ch, deb, hk7z, mv5g@virginia.edu

Abstract - **An important source of cyber-attacks is malware, which proliferates in different forms such as botnets. The botnet malware typically looks for vulnerable devices across the Internet, rather than targeting specific individuals, companies or industries. It attempts to infect as many connected devices as possible, using their resources for automated tasks that may cause significant economic and social harm while being hidden to the user and device. Thus, it becomes very difficult to detect such activity. A considerable amount of research has been conducted to detect and prevent botnet infestation. In this paper, we attempt to create a foundation for an anomaly-based intrusion detection system using a statistical learning method to improve network security and reduce human involvement in botnet detection. We focus on identifying the best features to detect botnet activity within network traffic using a lightweight logistic regression model. The network traffic is processed by Bro, a popular network monitoring framework which provides aggregate statistics about the packets exchanged between a source and destination over a certain time interval. These statistics serve as features to a logistic regression model responsible for classifying malicious and benign traffic. Our model is easy to implement and simple to interpret. We characterized and modeled 8 different botnet families separately and as a mixed dataset. Finally, we measured the performance of our model on multiple parameters using F1 score, accuracy and Area Under Curve (AUC).**

Index Terms - Botnet Detection, Cyber Security, Logistic Regression, Machine Learning

INTRODUCTION

Cyber-attacks are causing billions of dollars in losses each year and this cost is projected to grow to USD 2.1 trillion by 2019 [1]. The growth of cyber-crimes has also resulted in cyber-intrusions, one of the forms in which cyber-crimes manifest, becoming more commonplace, more dangerous, and more sophisticated. Despite the speed, variability, and growing commercial implications of breaches and cyber-attacks, institutions have been slow in implementing robust countermeasures against them. These cyber-security vulnerabilities were exploited again recently by a ransomware, WannaCry, which affected more than 150 countries and caused global financial and economic losses of approximately $4 billion dollars, making it one of the most damaging incidents involving cyber-attacks [2]. To prevent such cyber-attacks, many institutions have deployed Intrusion Detection Systems (IDS), which are devices or software applications that monitor a network for malicious activities or policy violations.

There are two types of IDS: signature-based and anomaly-based. A signature-based IDS is like a virus scanner in that it searches for known patterns (or signatures) from previous intrusion events. While signature-based IDSs are efficient at identifying known attack patterns, they depend on receiving regular signature updates to maintain awareness of the most novel attack techniques. In other words, a signature-based IDS is only as good as the recency and range of its database of stored signatures. On the other hand, anomaly-based IDSs create a baseline of traffic statistics and detect attacks through analysis of the variations in the traffic patterns relative to the baseline. Unlike signature-based systems, anomaly-based detection systems are constantly changing and learning, and they can detect zero-day attacks.

In this paper, we create a foundation for an anomaly-based IDS that can detect botnet traffic. We implement and evaluate logistic regression on a dataset consisting of normal traffic and malicious traffic from eight different botnet families. We also study the robustness of our model by performing a Leave-One-Bot-type-Out analysis.

LITURATURE REVIEW

Botnets are groups of remotely controlled, compromised machines. These compromised machines, called bots, connect to a central server operated by a "botmaster" that gives them instructions to execute. Different protocols can be used for the periodic communications between the botmaster and the compromised machines. The authors of *Botfinder* [3] proposed a machine learning based system that uses statistical features from traces, which are sets of flows between the same source host and destination host port. In our models, we chose to focus only on HTTP-based botnets because botmasters are regularly choosing this protocol over IRC in recent years. Zeus, one of the most popular botnets and the largest contributor of malicious traces in our dataset, is HTTP based. HTTP botnets follow a pull approach, where the bot continuously polls the C&C server for updates and commands. Rossow et al. [4] analyzed about 104,000 botnet

978-1-5386-6343-1/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.

samples from VirusShare and found only 8% of the samples used IRC as a communication channel, while the other samples used HTTP and DNS.

A considerable number of studies have been conducted to create models based on a limited number of botnet families, where a different model is created for each family, or one model is created for 2-3 botnet families. For example, Hadaddi et al. [5] developed C4.5 and Naive Bayes models to detect HTTP-based botnets using two families: Zeus and Citadel. Lu et al. [6] used hidden Markov models to differentiate the behavior of Zeus C&C communication from normal traffic. Torres et al. [7] further improved upon their work using the same dataset we used but picked only two families (DonBot and Neris) to train the model.

Some botnet detection models require inspecting packet payloads to extract the features for detection [8], [9]–[12]. For example, Etemad et al. [9] proposed a method to separate IRC traffic from HTTP traffic by light payload inspection. Our approach is based on trace level information that does not require payload inspection. Bilge et al. [13] proposed a botnet detection system called DISCLOSURE that extracts three types of features from Netflow records: size, host access pattern, and temporal behavior. They hypothesized that behavioral and temporal metrics generated by C&C communication have predictability which can be used for classification. That said, their final model had an accuracy of less than 65% but achieved a false positive rate of less than 1%. Soniya et al. [14] trained a neural network classifier by running 120 botnet malware samples including Pushdo, Banbra, BlackEnergy, Sasfis, Bifrose, Dedler, and Zeus achieving a false positive rate of 2.5%. While our datasets are different, we demonstrate that it is possible to significantly improve the accuracy using similar features generated from Bro logs. Finally, Haddadi et al. [15] analyzed Netflow data from three Botnets, Zeus, Conficker and Torpig, two of which are also used in our work. Flow features such as duration, number of packets, number of bytes, flows and bits per second were used to classify malicious traffic.

METHODOLOGY

Our approach to detect malicious traffic involved multiple stages. First, we downloaded malicious and benign traffic samples from the Czech Technical University's public repository of network traffic and converted each sample to a series of traces. Then, we performed feature extraction on the traces and calculated statistics related to size, frequency, and duration of communication. Finally, we split this data into training and test sets and performed supervised learning on these new features. We performed k-fold cross validation and hyper parameter tuning to improve the performance of each model.

I. Data

The data we have used was released by the Malware Capture Facility Project under Stratosphere IPS Project [16]. The researchers used a testbed network topology consisting of a set of virtualized computers to create malicious and normal network traffic [17]. Traffic that came to or from any of the known infected IP addresses was labelled as malicious data. Traffic from the known and controlled end points in the network, such as routers, proxies, or switches, was labelled as normal data. We chose the data consisting of malicious traffic from eight different Botnet families to increase the robustness of our detection models. This data was collected over time periods ranging from a week to over a month between the years 2015-17 [18][19]. Overall, we have over 10,000 flows of malicious traffic and an equal number of flows of benign traffic. And since each sample was collected independently, there were no inter-dependencies between samples. The traffic was captured in different formats including pcap (packet capture), Netflow, and Bro logs. Bro is a network monitoring framework which generates multiple logs files from network connections like http logs, IRC logs, DNS log and conn logs. Our research is focused on detecting botnets which use TCP or UDP for C&C communication. Hence, we have chosen conn logs over the other logs as conn logs capture TCP, UDP, and ICMP connections. These records were grouped into traces containing all flows between the same four-tuple of source IP, destination IP, destination port and protocol. Overall, our dataset contained over 22k traces of 1.6 million flows of malicious and benign network traffic.

II. Data Preprocessing

For every sample, each trace was collapsed down to a single record containing statistics characterizing the typical behavior between each source and destination on the network. The features we calculated can be generalized to three groups. The first set of features relates to the volume of the communication, or the average number of packets and amount of data sent in a typical flow. The second set of features pertains to the timing of communication, or the average flow duration and average time interval between successive flows. The final set of features pertains to the state history of connections in the form of flag counts which were obtained by parsing the state history field in the conn logs. Formally, we define the features as follows:

- Mean and standard deviation of time interval between two consecutive flows in a trace
- Mean and standard deviation of umber of source packets and destination packets
- Mean and standard deviation of flow duration within a trace
- Total flag counts (a/d/f/h/r/s/t) - Records the state history of connections as a string of letters. The definitions of these flags are as follows
  - a - pure ACK, or Acknowledgement flag
  - d - packet with payload data
  - f - packet with the FIN flag set, terminating the Connection
  - h - SYN+ACK (handshake) which is standard protocol for establishing a TCP connection
  - r - packet with RST bit set which resets the Connection

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.

- s - SYN w/o the ACK bit set (or an open-ended handshake)
- t - packet with re-transmitted payload

SYN, ACK and FIN are flags utilized by TCP to control the state of a connection. SYN (Synchronize) flag initiates a connection, ACK acknowledges receipt of client's SYN packet and FIN (final) terminates a connection.

III. Botnet Families

There are a wide variety of botnet families which exploit different vulnerabilities for propagating. In this study, we have explored 8 different types of botnets, including Bunitu, Conficker, Dridex, Miuref, Necurs, Trickbot, Upatre, and Zeus.

- Bunitu - A trojan which exposes infected machines to act as a proxy for remote clients. Once installed on a machine, the malware opens ports for remote connections, registers itself in the client's database and accepts connections on the exposed ports [20].
- Conficker - Leverages an old, unpatched vulnerability to crack passwords and hijack Windows computers into a botnet [21].
- Dridex - Dridex is a banking trojan that uses an affiliate system for its botnets. The targets of this malware are Windows users who open an email attachment in Word or Excel, causing macros to activate and download Dridex, infecting the computer and opening the victim to banking theft [22].
- Miuref - a Trojan that facilitates click fraud. It installs itself as a browser plugin to hide itself from detection as well as ensure that it loads itself every time the system's Internet browser is executed [23].
- Necurs - Considered to be the largest spam botnet in the world, it has been used to distribute malware and ransomware [24].
- Trickbot - Leverages the Necurs botnet to spread via spam email and then steals banking credentials via man-in-the-browser attacks [25].
- Upatre - usually come as malicious files that appear as legitimate attachments in email messages. Upon installation, UPATRE drops copies of itself into the system and executes them. It then connects the infected system to possibly malicious URLs to download and execute malicious files [26].
- Zeus - Trojan horse malware package that runs on versions of Microsoft Windows. While it can be used to carry out many malicious and criminal tasks, it is often used to steal banking information by man-in-the-browser keystroke logging and form grabbing [27].

An exploratory analysis of network traffic data revealed new characteristics of these botnets. Table I details the flow count, mean flow duration and mean source packets at botnet family level.

We can draw inferences from these statistics - for instance, the median of mean number source packets in each flow is lowest (1.5 source packets per flow) for Necurs and Upatre. This shows that these botnets use fewer packets for C&C communication compared to other botnets.

MODELING

I. Logistic Regression

We used logistic regression to predict whether a given set of network data was benign or malicious. Logistic regression was chosen since it is light-weight, easy to implement, easy to interpret, and has low computational requirements. It is also one of the most widely used supervised learning models today. These attributes make logistic regression an excellent candidate for problems requiring quick, potentially automated decisions with very large data, such as cyber security. The logistic regression model we formulated associates the probability of the outcome (trace maliciousness) to a series of potential predictor variables as shown in the following equation:

$$log\left[\frac{p}{1-p}\right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i. \qquad (1)$$

where p is the probability of the maliciousness, β_0 is an intercept term, $\beta_1 \ldots \beta_i$ are coefficients associated with each variable $X_1 \ldots X_i$

TABLE I
STATISTICAL ANALYSIS FOR EACH BOTNET FAMILY

Family	Median (Flowcount)	Std dev (Flowcount)	Median (Mean Duration)	Std dev (Mean Duration)	Median (Mean Source Packets)	Std dev (Mean Source Packets)
Bunitu	2	165	16.1	302.3	6	258
Conflicker	3	143.6	0.7	120.7	4	80.4
Dridex	52	1478.3	1.9	10456.9	5.2	10.7
Miuref	2	440.2	61	635.9	7	281.2
Necurs	2	43	1.5	22.6	1.5	133.2
Trickbot	21	3079.8	79.6	47.3	7.3	23.6
Upatre	148	2476.1	1.5	12.2	1.5	21.5
Zeus	21	11353.3	5.7	5.7	5.2	3.4

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.

This model assumes that all predictors are related in a linear manner to the log odds of the outcome, in our case, maliciousness of the trace. The presence of multicollinearity in the predictor variables caused convergence issues for logistic regression and we used lasso regularization to overcome this problem. Lasso regression selects only one feature among the highly correlated ones and reduces the coefficient of others to zero.

Multiple metrics were used for evaluation – Accuracy, Precision, Recall, F1 Score, and AUC. This evaluation was carried out in two stages. First, the dataset was split into a training set which contained two-thirds of the traces, while the rest was used as the test set. We found that the predictor variables had highly skewed distributions, hence we performed logarithmic transformation on all skewed variables. The supervised learning model was then trained on the training data to predict whether a given set of traffic was benign or malicious, and cross-validated for model selection and model performance assessment. Next, the models were tested using the test dataset, and the predictions on test data were used to arrive at the performance metrics for model comparison.

II. Leave One Bot-Type Out

We assessed the robustness of the logistic regression model to new bot types using Leave One Bot-Type Out approach. The objective was to understand the model's transferability to novel bot families which invariably spawn in the botnet landscape. In Leave One Bot-Type Out, we trained the model on all bot types except one and used the excluded bot type data as test for the model. This was repeated 8 times, each time excluding one bot type in training. The Leave One Bot-Type Out approach is more robust than standard cross-validation since in the standard approach the model is exposed to every botnet family in training due to which this approach does not allow us to effectively measure the performance of the model when it encounters a new botnet family. Leave One Bot-Type Out approach allowed us to ascertain the performance of the logistic regression model when it encounters new botnet families on parameters like balanced accuracy, precision, recall, F1 score, and AUC.

RESULTS

I. Logistic Regression

The Logistic Regression model we trained on the entire train dataset with labelled malicious and benign traffic achieved an AUC of 0.985 and an accuracy of 95%, with the overall recall of the model being 96.7%. The high recall signifies that the model can accurately classify nearly all malicious traffic. A high recall is a critical requirement for Intrusion Detection Systems as botnet traffic can be sporadic and only a system with high recall can ensure that this malicious traffic is detected accurately.

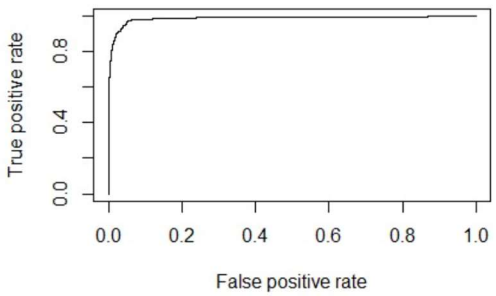

FIGURE I
ROC CURVE FOR LOGISTIC REGRESSION (AUC = 0.985)

II. Leave One Bot-Type Out

Since the predictor variables had a highly skewed distribution, we log-transformed these variables before implementing Leave One Bot-Type Out validation. Table II details the performance scores of the logistic regression model for different botnet families using Leave One Bot-Type Out validation. We observe a high AUC value (>0.97) for all botnet families except Bunitu and Zeus. The recall is equal to or greater than 95% for all botnet families. Fig II shows the ROC curves of the most highly represented botnet families in the dataset, Miuref and Bunitu. These two botnet families constitute more than 50% of all malicious traces in our dataset. These findings show that the logistic regression model can be generalized to most new botnet families. Its noteworthy that unlike other botnet families, the performance of Zeus botnet family is significantly lower. Interestingly, Zeus had a better performance when the predictors were not log-transformed (See AUC* column in Table 2). We can infer that the features of Zeus do not have a wide distribution, which makes it difficult to detect Zeus traces after log-transforming the predictor variables. Based on this result, in future we can also explore an ensemble model with transformed and untransformed predictors, and get relatively better performance on Leave One Bot-Type Out validation.

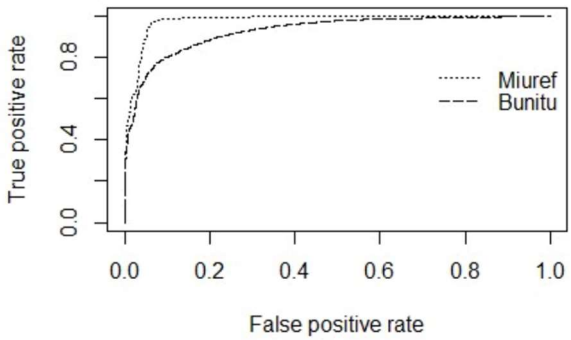

FIGURE II
LEAVE-ONE-BOT-FAMILY-OUT ROC CURVES

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.

TABLE II
EVALUATION METRICS FOR EACH BOTNET FAMILY

Family	Balanced Accuracy	Precision	Recall	F1 Score	AUC	AUC*
Bunitu	0.691	0.619	0.989	0.762	0.916	0.904
Conclicker	0.970	0.961	0.979	0.970	0.983	0.896
Dridex	0.958	0.966	0.949	0.957	0.997	0.968
Miuref	0.867	0.812	0.955	0.878	0.973	0.938
Necurs	0.983	0.999	0.966	0.982	0.999	0.922
Trickbot	0.967	0.976	0.958	0.967	0.994	0.971
Upatre	0.973	0.982	0.964	0.973	0.992	0.989
Zeus	0.491	0.495	0.962	0.654	0.681	0.985

AUC*: AUC without logarithmic transformation of predictors

CONCLUSION AND FUTURE WORKS

We achieved two main objectives through our work - first, we identified the best features to classify malicious traffic using 8 different botnet families, and second, we evaluated the performance and robustness of our logistic regression model. We experimentally demonstrated that logistic regression can perform reasonably well in detecting malicious traffic. Our model gave us an AUC of 0.985 and a recall of 96.7%. The results of Leave One Bot-Family Out validation have been encouraging and prove that the model is robust to unseen botnet families. Except for the Bunitu and Zeus families, the model gave an AUC of over 97%. These results demonstrate the suitability of logistic regression for creating an Intrusion Detection System (IDS). However, as normal network usage is highly sporadic, more robust models are required for monitoring these networks. We have tested our model on balanced distribution of malicious and benign traffic, and we need to test the model on real-world network traffic, where the distribution of malicious and benign traffic is unlikely to be balanced. This class imbalance could prove to be a challenge for traditional supervised learning models, requiring the creation of custom loss functions.

In future work, we plan to apply more supervised machine learning methods, such as SVM, Random Forest and Neural Network, to our dataset, and eventually test the models across a large network like the University of Virginia. There is evidence of temporal metrics like host access patterns and Fast Fourier Transforms (FFT) that can further improve detection [3]. Ultimately, we would like to setup an online, quick response system that can identify, trigger and quarantine botnet traffic.

ACKNOWLEDGMENT

This work was supported by a 4-VA Consortium grant and a University of Virginia School of Engineering and Applied Science Research Innovation Award. We'd like to thank Fatma Alali and Jack Morris for their support of this work. We'd also like to thank Alex Ptak and Jeff Collyer at Information Security Policy and Records Office (ISPRO) at for their continuing support helping us get the data.

REFERENCES

[1] "Cyber Crime Costs Projectd to Reach $2 Trillion by 2019." 2016. forbes.com/sites/stevemorgan/2016/01/17/cyber-crimecosts-projected-to-reach-2-trillion-by-2019/.

[2] "WannaCry Ransomware Attack Losses Could Reach $4 Billion." 2017. cbsnews.com/news/wannacry-ransomware-attackswannacry-virus-losses/.

[3] G. V. C. K. Florian Tegeler, Xiaoming Fu, "Botfinder: Finding bots in network traffic without deep packet inspection," CoNEXT '12 Proceedings of the 8th international conference on Emerging networking experiments and technologies, pp. 349 – 360, 2012.

[4] C. Rossow, C. J. Dietrich, H. Bos, L. Cavallaro, M. Van Steen, F. C. Freiling, and N. Pohlmann, "Sandnet: Network traffic analysis of malicious software," in Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, pp. 78 – 88, ACM, 2011.

[5] F. Haddadi, J. Morgan, E. Gomes Filho, and A. N. Zincir-Heywood, "Botnet behaviour analysis using ip flows: with http filters using classifiers," in Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on, pp. 7 – 12, IEEE, 2014.

[6] C. Lu and R. Brooks, "Botnet traffic detection using hidden markov models," in Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, p. 31, ACM, 2011.

[7] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of recurrent neural networks for botnet detection behavior," in Biennial Congress of Argentina (ARGENCON), 2016 IEEE, pp. 1 – 6, IEEE, 2016.

[8] T. Cai and F. Zou, "Detecting http botnet with clustering network traffic," in Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, pp. 1 – 7, IEEE, 2012.

[9] F. F. Etemad and P. Vahdani, "Real-time botnet command and control characterization at the host level," in Telecommunications (IST), 2012 Sixth International Symposium on, pp. 1005 – 1009, IEEE, 2012.

[10] M. Eslahi, M. Rohmad, H. Nilsaz, M. V. Naseri, N. Tahir, and H. Hashim, "Periodicity classification of http traffic to detect http botnets," in Computer Applications & Industrial Electronics (ISCAIE), 2015 IEEE Symposium on, pp. 119 – 123, IEEE, 2015.

[11] K. Li, C. Liu, and X. Cui, "Poster: A lightweight unknown http botnets detecting and characterizing system," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1454 – 1456, ACM, 2014.

[12] M. Grill and M. Reh´ak, "Malware detection using http user-agent discrepancy identification," in Information Forensics and Security (WIFS), 2014 IEEE International Workshop on, pp. 221 – 226, IEEE, 2014.

[13] W. R. E. K. C. K. Leyla Bilge, Davide Balzarotti, "Disclosure: Detecting botnet command and control servers through large-scale netflow analysis," ACSAC '12 Proceedings of the 28th Annual Computer Security Applications Conference, pp. 129 – 138, 2012.

[14] B. Soniya and M. Wilscy, "Using entropy of traffic features to identify bot infected hosts," in Intelligent Computational Systems (RAICS), 2013 IEEE Recent Advances in, pp. 13 – 18, IEEE, 2013.

[15] A. N. Z.-H. M. I. H. Fariba Haddadi, Dylan Runkel, "On botnet behaviour analysis using gp and c4.5," GECCO Comp '14 Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 1253 – 1260, 2014.

[16] "https://stratosphereips.org/category/dataset.html."

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.

[17] J. A. S.Garca, M.Grill, "An empirical comparison of botnet detection methods," Computers Security, pp. 100 – 123, 2014.
[18] "https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-128-1/README.md"
[19] "https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-249-1/README.md"
[20] "Bunitu Trojan botnet supports commercial VPN infrastructure." 2015. zdnet.com/article/bunitu-trojan-spreads-through-commercial-vpn-service/
[21] "Conficker." *Wikipedia*, Wikimedia Foundation, 16 Mar. 2018, en.wikipedia.org/wiki/Conficker.
[22] "It's back: Dridex strikes again." 2017. scmagazine.com/its-back-dridex-strikes-again/article/648749/
[23] "The Click Fraud Malware: How MIUREF Turns Users into Cybercriminal Accomplices." 2014. trendmicro.com.ru/vinfo/ru/ threat-encyclopedia/web-attack/133/the-click-fraud-malware-how-miuref-turns-users-into-cybercriminal-accomplices"
[24] "Necurs Botnet Returns to Top 10 Malware List." 2017. infosecurity-magazine.com/news/necurs-botnet-returns-to-top-10/"
[25] "Trickbot comes with new tricks – attacking Outlook and browsing data." 2017. malwarebytes.com/threat-analysis/2017/08/trickbot-comes-with-new-tricks-attacking-outlook-and-browsing-data/"
[26] "Keeping Up with the Damage of UPATRE." 2015. trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/3159/keeping-up-with-the-damage-of-upatre"
[27] "Zeus (Malware)." *Wikipedia*, Wikimedia Foundation, 21 Mar. 2018, en.wikipedia.org/wiki/Zeus_(malware).

Authorized licensed use limited to: University of West Florida. Downloaded on June 07,2022 at 12:59:33 UTC from IEEE Xplore. Restrictions apply.