PowerShell is increasingly used by cybercriminals as part of their attacks’ tool chain, mainly for downloading malicious contents and for lateral movement. Indeed, a recent comprehensive technical report by Symantec dedicated to Power-Shell’s abuse by cybercriminals reported on a sharp increase in the number of malicious PowerShell samples they received and in the number of penetration tools and frameworks that use PowerShell. This highlights the urgent need of developing effective methods for detecting malicious PowerShell commands. In the proposed paper, this challenge is addressed by implementing several novel detectors of malicious Power-Shell commands [1] and evaluating their performance. The proposal implements both “traditional” natural language processing (NLP) based detectors and detectors based on character-level convolutional neural networks (CNNs).

Private stream searching appears to be an entirely effective method for malware to surreptitiously search and exfiltrate email by resisting malware analysis techniques [2]. Malware designed to save and return messages on a specific sensitive topic will be able to do so without revealing the topic of interest upon analysis; all that will be determined is that it scans email in general. Furthermore, as the paper’s implementation demonstrates, there is nothing to prevent these techniques from being used immediately. The example of PIR-based malware illustrates the more general possibility of malware employing public key obfuscation techniques to hide its behaviour.

The traditional security systems like Intrusion Detection System/Intrusion Prevention System and Anti-Virus (AV) software are not able to detect unknown malware as they use signature-based methods. In order to solve this issue, static and dynamic malware analysis is being used along with machine learning algorithms for malware detection and classification. The main problems with these systems are that they have high false positive and false negative rate and the process of building classification model takes time (due to large feature set) which hinders the early detection of malware. Thus, the challenge is to select a relevant set of features, so that, the classification model can be built in less time with high accuracy. The proposal presents a system that addresses both the issues mentioned above. It uses an integration of both static and dynamic analysis features of malware binaries incorporated with machine learning process for detecting Zero-day malware [3]. The proposed model is tested and validated on a real-world corpus of malicious samples. The results show that the static and dynamic features considered together provide high accuracy for distinguishing malware binaries from clean ones and the relevant feature selection process can improve the model building time without compromising the accuracy of malware detection system.

**REFERENCES**

[1] Danny Hendler, Shay Kels and Amir Rubin (2018) “Detecting Malicious PowerShell Commands using Deep Neural Networks”. In: arXiv:1804.04177, Cornell University, 2018.

[2] Bethencourt, John, Dawn Xiaodong Song and Brent Waters.“Analysis-Resistant Malware”. In: NDSS (2008)

[3] Ekta Gandotra, Divya Bansal, Sanjeev Sofat (2016) “Zero-Day Malware Detection”. In: 2016 Sixth International Symposium on Embedded Computing and System Design (ISED).