Article

Unexpected-Behavior Detection Using TopK Rankings for Cybersecurity

Alvaro Parres-Peredo , Ivan Piza-Davila and Francisco Cervantes

Electronic, Systems and Informatics Department, ITESO, The Jesuit University of Guadalajara, Tlaquepaque 45604, Mexico; hpiza@iteso.mx (I.P.-D.); fcervantes@iteso.mx (F.C.)

* Correspondence: parres@iteso.mx; Tel.: +52-33-3669-3517

Received: 16 August 2019; Accepted: 8 October 2019; Published: 17 October 2019

Abstract: Anomaly-based intrusion detection systems use profiles to characterize expected behavior of network users. Most of these systems characterize the entire network traffic within a single profile. This work proposes a user-level anomaly-based intrusion detection methodology using only the user's network traffic. The proposed profile is a collection of TopK rankings of reached services by the user. To detect unexpected behaviors, the real-time traffic is organized into TopK rankings and compared to the profile using similarity measures. The experiments demonstrated that the proposed methodology was capable of detecting a particular kind of malware attack in all the users tested.

Keywords: cybersecurity; intrusion detection system; profiling; network security; TopK ranking

1. Introduction

The Internet has been growing at a very high rate, becoming the primary global media. Due to the development of novel computing technologies and the “As Service” model, the Internet has also become the operations center of many organizations. At present, a wide variety of data is traveling on the Internet: from simple email to the entire operations data of a company. This makes computer network security more critical than ever.

Day after day, information systems suffer from new kinds of attacks. As these attacks become increasingly complex, the technical skill required to create them is decreasing [1].

The term “computer security” is defined by the National Institute of Standards and Technology (NIST) [2] as follows: the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (including hardware, software, firmware, information/data, and telecommunications).

Migga [3] defines computer security as a branch of computer science that focuses on creating secure environments for the use of computers. It focuses on the behavior of the users of computers and related technologies, as well as on the protocols required to create a secure environment for everyone. When we talk about computer network security, the secure environment involves all network resources: computer, data, devices, and users.

At present, firewalls and access control systems are no longer enough to protect computer systems. Intruders find new ways to attack computers and systems. This motivated the rise of a new layer of security called the intrusion detection system (IDS). The first approach of an IDS was proposed by Anderson [4] in 1980. An IDS intends to identify intruders (or attackers) by monitoring and analyzing the events on systems, computers and/or networks. Figure 1 shows the security methods on a simple computer network diagram.

Current IDSs are classified according to the approach employed to detect intrusions. The most popular approaches are: signature based and anomaly based. The former is very efficient in detecting

well-known attacks, but it is quite inefficient in detecting new forms of attacks. The latter is more efficient in detecting new forms of attacks, but it has high rates of false positives.

Figure 1. Diagram of a simple network connected to a Wide Area Network (WAN) with a firewall and an intrusion detection system (IDS).

On the other hand, traditional authors like Stalling [5] define three types of network based on the geographical scope: (1) local area networks (LANs), (2) metropolitan area networks, and (3) wide area networks. Modern authors like Edwards Wade [6] incorporate new types of network like the campus area network (CAN), which is defined as a group of LAN segments interconnected within a building or group of buildings that form one network. Typically, the company owns the entire network, including the wiring between buildings, in contrast to metropolitan area networks.

In large organizations such as universities, many users (students, employees, visitors) are connected to the campus area network (CAN) to either access intranet services or obtain internet access, from different kinds of devices. The probability for a network attack to be originated from inside the CAN is high for two main reasons: (a) malicious behaviors of inexperienced users practicing some hacking technologies (script kiddies), and (b) privileged users are the victim of social engineering attacks when clicking links on e-mails or web pages from untrusted sources.

We believe that a viable way to prevent these security problems is by detecting when a user is having abnormal network behavior. This involves building individual network profiles representing the normal behaviors of every user in an organization. To do this, real-time traffic has to be captured from the nearest point to each user's access device or even from the same device.

In this work, we propose a methodology capable of detecting when a network user is having an abnormal behavior and, therefore, could be the victim of a network attack.

Our proposal uses network traffic captured at the host machine. We build a TopK ranking containing the services with the highest amount of bytes transferred from/to the host during a time-frame. Using TopK rankings for user profiling is a novel element in the design of anomaly-based IDS.

Most of the state-of-the-art anomaly-based IDSs use the traffic captured at the border of the network. Thus, their profiles represent the behavior of the entire network segment. In contrast,

our profile reproduces the behavior of a single user. Even though our proposal is clearly less scalable, our focus is on protecting privileged users in the organization, who execute critical tasks, from internal and external threats.

The present document is organized as follows: Section 2 introduces the methodologies employed by intrusion detection systems. Section 3 presents a review of works that consider user profiling to detect anomalous behaviors. Section 4 briefly describes a profiling method using TopK rankings. Section 5 introduces our unexpected-behavior identification methodology. Section 6 presents the experiments that validate the methodology. Finally, Section 7 summarizes the conclusions and future work.

2. Intrusion Detection Systems

Intrusion detection is the process of monitoring the events occurring within a computer system or network, and analyzing them to search for signs of possible violation of computer security policies. These events might have a malicious nature, such as malware or attackers. An IDS is a software system that automates the intrusion detection process [7].

NIST defines three detection methodologies: (1) signature-based detection; (2) anomaly-based detection; and (3) stateful protocol analysis [7].

2.1. Signature-Based Detection

This detection technique is the first one employed on an IDS and was introduced in Anderson reports [4]. The intrusion is detected by matching the behavior recorded in either log registries, network packets, or system status against well-known suspicious patterns. This methodology is very effective in detecting known attacks, but is useless for new forms of attacks [8].

Examples of malicious behaviors and related attacks that this methodology is able to detect are described in Table 1, where we can observe that the expected behaviors are well defined. Figure 2 depicts this detection methodology in a diagram.

Table 1. Known attacks and their expected behavior.

Attack	Expected Behavior
Unauthorized access	SSH try of a root login
Unauthorized web access	Multiple wrong password web login forms submitted from the same IP address in a short period of time
Malware execution	High CPU load into a server and multiple outgoing TCP connections
Social Engineering and malware	An email with subject “Urgent Document” and an attachment filename “authorization.exe”

Figure 2. Diagram of a generic signature-based IDS.

2.2. Anomaly-Based Detection

Anomaly-based detection is the process of comparing definitions of what is considered normal (i.e., profiles) against observed events in order to identify significant deviations. The profiles are built by monitoring the characteristics of typical activity over a period of time [7]. An advantage of this methodology is its ability to find new forms of attacks, but normally it has high rates of false positives. Figure 3 depicts a generic diagram of this detection methodology.

Figure 3. Diagram of a generic anomaly-based IDS.

Anomaly-based IDS is considered to be one of the foremost research areas in network security [9]. The detection methods and the selection of the system or network features to be monitored are two open issues [8].

Many research works have been developed around the classification of network traffic using different machine-learning techniques, with the aim of achieving 100% intrusion detection and a low rate of false positives. Two classes of machine learning techniques are used: (a) single and (b) hybrid classifiers. Examples of single machine-learning classifiers include the following: support vector machines, self-organizing maps, neural networks, and k-nearest neighbors. Hybrid machine-learning classifiers have the purpose of acquiring a superior probable accuracy for intrusion detection; these classifiers combine many machine-learning techniques to improve their performance [9].

2.3. Stateful Protocol Analysis

Stateful protocol analysis or deep packet inspection is the process of comparing predefined accepted protocol activity against the observed events to identify deviations. Unlike anomaly-based detection, which uses network profiles, stateful protocol analysis relies on vendor specifications of how the protocol should and should not be used [7]. In Figure 4 we can observe a generic diagram of this type of detection methodology.

Figure 4. Diagram of a generic stateful-protocol IDS.

Usually, the vendor specifications include: rules for individual commands, sequence of commands, minimum and maximum lengths for arguments, argument data types, etc.

This methodology produces a low false-positive rate similar to signature-based detection because it is based on legitimate behaviors. In addition, it is able to detect unexpected behaviors [10]. The big

challenge of this methodology is to design secure protocol specifications strong enough to detect illegitimate behaviors.

3. Network User Profiles for Anomaly-Based Intrusion Detection Systems

The construction of profiles based on network traffic for representing the normal behavior in anomaly-based detection IDSs has been part of the research on computer network security.

Many research works on anomaly-based detection systems validate their proposals using common datasets like KDD-CUP99 [11] and NLS-KDD [12]. The former is an artificial dataset for testing intrusion detection systems, and the latter is a more realistic dataset where traffic data was generated from real profiles. The main problem with these datasets is that they contain many application-specific fields (e.g., “number of failed logins”), such that they are not available on raw real network traffic.

Kuai [13] proposes an approach for profiling traffic behavior: he identifies and analyzes clusters of hosts or applications that exhibit similar communication patterns. In this approach, he uses bipartite graphs to model network traffic at the internet-facing links of the border router; then, he constructs one-mode projections of bipartite graphs to connect source hosts that communicate with the same destination host(s) and to connect destination hosts that communicate with the same source host(s). These projections enable similarity matrices of internet end-hosts to be built, where similarity is characterized by the number of common destinations or sources between two hosts. Based on these end-hosts matrices, at the same network prefixes, a simple spectral clustering algorithm is applied to discover the inherent end-host behavior cluster.

Kuai [13] carried out an analysis over a 200-GB dataset collected from an internet backbone of 8.6 GB/s bandwidth. The data was reduced by adding packet traces into 5-tuple network flows. The dataset was built using 24-bit network prefixes with timescales of 10 s, 30 s, and 1 min; these timescales were chosen because they produced the highest percentages of hosts in the top cluster. Kuai concluded the following: (1) there was no correlation between the number of observed hosts and the number of behavior clusters, (2) the majority of end-hosts remained in the same behavior cluster over time, and (3) the profiling of network traffic in network prefixes detected anomalous traffic behaviors.

A similar approach was employed by Qin [14] using traffic at port 80 (HTTP protocol), and integrating the destination URL instead of the IP address. One of the conclusions was that 93% of the hosts remained in the same behavior cluster.

Sing et al. [15] present an intrusion detection technique using network-traffic profiling and an online sequential extreme machine-learning algorithm. The proposed methodology runs two profiling procedures: alpha and beta profiling. The former creates profiles on the basis of protocol and service features, and the latter groups the alpha profiles in order to reduce the number of profiles. The authors conducted three different experiments: (1) using all features and alpha profiling; (2) using only some features and alpha profiling; and (3) using only some features, alpha profiling, and beta profiling. The best results were obtained from the last experiment using both profiling methods. The dataset used in this work was NLS-KDD.

Jakhale [16] presents an anomaly-based IDS that builds the profile using three different data mining algorithms that identified the frequent patterns. The author evaluated the profile against real-time traffic, obtaining high detection rates and low false alarm rates.

As we can see, all these works used the traffic captured at a far point from the end-user host, even outside the local network of the user, leaving the internal network security unattended. On the other hand, the use of profiles has proved to be feasible to either identify or specify network behaviors.

4. Network User Profiling Using TopK Rankings

In accordance with the NIST, an IDS that uses anomaly-based detection has profiles that represent the normal behavior of any of the following: user, host, network connection, or application. Then, it is compared to real-time activity in order to detect a significant difference [7].

In [17], a profiling method is proposed based on building TopK rankings of accessed services from network traffic captured at the host. Each service is represented by the 3-tuple <remote IP address, transport protocol, remote port>. This profiling process is carried out within a secure environment where it can be guaranteed that the host is used only by the expected user and there is no malware, virus, trojan, or any other malicious software installed. This method produces a profile structure constituted by a list of TopKs denoting the normal behavior of a user at their computer.

Each TopK in the profile represents the top K most accessed services based on total transferred bytes, during a timeframe f . A new TopK is calculated every Δt seconds. Each TopK is overlapped with the previous ones as is illustrated in Figure 5.

Figure 5. Depicting timeframe overlapping in the profile building process.

Additionally, this profiling method offers a mechanism to determine how similar a given TopK ranking is to the profile, returning a value in the range $[0.0 \cdots 1.0]$, where 0.0 and 1.0 denote, respectively, totally different and identical.

5. Unexpected Behavior Identification

This work proposes a methodology capable of detecting an unexpected network behavior—which might be an intrusion—based on computing the user’s predominant behavior. This methodology is depicted in Figure 6 and consists of the following phases:

1. Continuously capture real-time network traffic at the host;
2. Build a TopK ranking every Δt seconds from the most recently captured traffic;
3. Calculate the similarity S of each TopK to the user profile;
4. Identify the predominant behavior PB every Δw seconds;
5. Evaluate the current predominant behavior;
6. Determine whether or not to trigger an alarm.

The first two phases employ the same algorithms and parameters as those used to build the user profile. The similarity is calculated using the mechanism offered by the profiling system [17], which is based on the average overlap measure [18].

Figure 7 depicts a sequence of S calculated during six hours of capturing real-time traffic of a single user. We can observe that the points are too disperse to conclude that there is an unexpected

behavior by evaluating a single similarity value S . Therefore, a method that analyzes many successive points will be useful to conclude whether the predominant behavior is actually unexpected or not.

Figure 6. Block diagram of the proposed methodology for detecting an unexpected behavior.

Figure 7. Plot of all calculated average overlaps of real-time against the profile during 6 h.

In order to identify the predominant behavior within a sequence of S , we use a signal-processing technique called moving-average filter, formally defined as:

$$y[n] = \frac{1}{M} \sum_{l=w_1}^{w_2} x[l], \quad (1)$$

where M is the number of points at the time-frame $[w_1 \dots w_2]$, and $x[l]$ is the value of the point at time-frame l . The filter reduces these points into a single point $y[n]$ by calculating the mean value [19]. This value corresponds to the predominant behavior PB during time-frame $[w_1 \dots w_2]$.

The next time-frame starts at $w_1 + \Delta w$. In this implementation, Δw is smaller than $w_2 - w_1$ to guarantee that time-frames overlap. Figure 8 depicts the operation of the moving-average filter and how time frames overlap.

Figure 9 shows an example of applying the moving-average filter over an S succession. Blue circles denote S values and orange diamonds represent the PB values. We can observe that the orange diamonds follow the predominant behavior of blue circles.

Figure 8. Moving-average filter example. Diamond *A* represents the average of *S* from 9:00 to 9:20 and diamond *B* represents the average of *S* from 9:10 to 9:30.

Figure 9. Applying the moving-average filter to represent a predominant behavior with a threshold $T = 0.40$.

The evaluation of the current *PB* is based on comparing it continuously with a threshold value T , where T denotes the minimum value for a predominant behavior to be tagged as expected. Therefore, if *PB* is below T during N time-frames, we can conclude that the user is having an unexpected behavior and, therefore, a possible attack. In such a case, an alarm should be triggered.

Figure 9 also includes a horizontal line at $PB = 0.40$ to represent a valid threshold value T for the user being analyzed. Thus, we can appreciate an unexpected behavior starting at 11:00 h.

6. Experiments and Results

We conducted an experiment to validate the proposed methodology and its ability to detect unexpected behaviors.

The experiment was carried out in a 16-bit campus area network (CAN); it had a Windows domain controller and used an HTTP proxy. The campus applications included web-apps and remote desktop apps. The email service was provided by Microsoft Exchange Server which was hosted outside of the campus network.

Five faculty members took part in this experiment. They were endowed with brand new laptops by the IT department. Each laptop was set up with the fresh institutional image. No unauthorized software was installed. The traffic generated on each of their computers during the first month was selected as normal behavior. Each laptop had two types of network accesses: (1) a wired access with a static IP address and (2) a wireless access with a dynamic IP address. Most of the time, they used their laptops inside the campus; however, sometimes they used them outside. The traffic data was captured by means of a Java application that used the Pcap4J library (<https://www.pcap4j.org/>). This program was installed on each laptop as an auto-start service.

The traffic generated on each of their computers during the first month was selected as normal behavior and, therefore, used to build the profile of each of the professors. Then, the real-time traffic of each user was captured and processed through the steps of the methodology proposed.

In the mean time, a malware was installed on each laptop with the purpose of inducing an unexpected behavior. This malware transferred files from the laptop to an external server. After copying 1 GB of data, the malware finished its execution and removed itself. The malware was created with the Metasploit (<https://www.metasploit.com/>) framework using a reverse HTTPS Meterpreter payload that connected to the Metasploit server that was hosted outside of the university.

Figure 10a–e depicts the predominant network behavior of the five faculty members, identified as Users A to E, during the execution of the malware, an hour before it started, and a couple of hours after it finished. The first valley in each plot represents the predominant behavior during the attack.

Each plot includes a green line that represents the threshold T at $PB = 0.4$ denoting the lower bound on PB of a predominant behavior which can be labeled as expected. This bound was selected experimentally.

Figure 10. Cont.

(e)

Figure 10. Predominant behavior of five different users during a malware attack. (a) User A. (b) User B. (c) User C. (d) User D. (e) User E.

Figure 11 depicts the predominant behavior of a single user during a full work week. The periods of time during which the user seemed to exhibit an unexpected behavior are highlighted. After interviewing the user, we obtained the following explanations for these behaviors which are labeled in the plot with letters A to F: (A) the user was connected outside of the campus making personal use of the laptop, (B) the user decided to use the computer for entertainment during lunch time, (C) the user was doing some activity not registered in the profile, (D) the intentional malware was running, (E) the user was doing some activity not registered in the profile, and (F) the user had not started working yet, so it could have been system traffic such as software update.

Figure 11. Predominant behavior of user E during a full work week, highlighting unexpected behavior periods.

Similarly, Figure 12 depicts the predominant behavior of another user during a full work week. We can observe a more stable user that exhibited only two moments of unexpected network behavior: the first one (A) was very near lunch time, so maybe the user did something unusual like a video conference; the second one (B) was the intentional malware attack.

The reason behind the previously explained false-positives is that such behaviors do not match any behavior registered in the user profile.

Figure 12. Predominant behavior of user C during a full work week, highlighting unexpected behavior periods.

7. Conclusions and Future Work

Based on the experimental results, we can conclude the following:

1. A trojan malware execution affects the network behavior at a given host, causing a significant reduction of the similarity value between real-time traffic and the profile. Therefore, our methodology is capable of triggering an alarm when the predominant behavior of the user starts deviating from the expected one.
2. An anomaly-based IDS that builds a profile for each individual network user represents an additional security mechanism because it is capable of detecting unexpected network behaviors that might be originated by a malware. The antivirus was not capable of detecting the installed trojan in either of our experiments.

An anomaly-based IDS must update the profiles on a regular basis because the normal behaviors of users change periodically. The proposed profiling method relies on the creation of TopK lists instead of using a supervised classifier as other approaches do. Therefore, updating the profile is computationally viable because it does not involve a re-training process.

Future work will go in two directions: (1) a dynamic profiling method capable of removing the least common behaviors from the profile, which might include the behavior induced by an attack that occurred during a previous profiling process; and (2) a decision model that triggers an alarm when an unexpected behavior is detected.

Author Contributions: Conceptualization, A.P.-P.; methodology, A.P.-P., I.P.-D. and F.C.; software, A.P.-P.; formal analysis, A.P.-P., I.P.-D. and F.C.; investigation, A.P.-P., I.P.-D. and F.C.; writing—original draft preparation, A.P.-P.; writing—review and editing, I.P.-D.; funding acquisition, A.P.-P.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by ITESO's Program for Academic Level Enhancement (Programa de Superación de Nivel Académico, PSNA) through an assistantship granted to A. Parres-Peredo.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CAN	Campus Area Network
IDS	Intrusion Detection System
LAN	Local Area Network
NIST	National Institute of Standards and Technology
WAN	Wide Area Network
PB	Predominant Behavior

References

1. Paquet, C. *Authorized Self-Study Guide Implementing Cisco IOS Network Security (IINS)*; Cisco Press: Indianapolis, IN, USA, 2009.
2. Guttman, B.; Roback, E.A. *An Introduction to Computer Security: The NIST Handbook*; DIANE Publishing: Darby, PA, USA, 1995.
3. Kizza, J.M. *Guide to Computer Network Security*, 2nd ed.; Computer Communications and Networks; Springer: London, UK; New York, NY, USA, 2013.
4. Anderson, J.P. *Computer Security Threat Monitoring and Surveillance*; Report Contract 79F296400; NIST: Fort Washington, PA, USA, 1980.
5. Stallings, W. *Data and Computer Communications*, 10th ed.; Pearson: Boston, MA, USA, 2014.
6. Edwards, W.; Jack, T.; Lammle, T.; Skandier, T.; Padjen, R.; Pfund, A.; Timm, C. *CCNP Complete Study Guide: Exams 642-801, 642-811, 642-821, 642-831*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
7. Scarfone, K.A.; Mell, P.M. *Guide to Intrusion Detection and Prevention Systems (IDPS)*; Technical Report NIST SP 800-94 Rev1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2012.
8. Ghorbani, A.A.; Lu, W.; Tavallae, M. Detection Approaches. In *Network Intrusion Detection and Prevention: Concepts and Techniques*; Ghorbani, A.A., Lu, W., Tavallae, M., Eds.; Advances in Information Security; Springer: Boston, MA, USA, 2010; pp. 27–53. [\[CrossRef\]](#)
9. Juma, S.; Muda, Z.; Mohamed, M.A.; Yassin, W. Machine Learning Techniques for Intrusion Detection System: A Review. *J. Theor. Appl. Inf. Technol.* **2015**, *72*, 422–429.
10. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [\[CrossRef\]](#)
11. Kandall, K. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Bachelor's and Master's Thesis, Department of Electrical Engineering and Computer Science, MIT, Boston, MA, USA, 1999.
12. Shiravi, A.; Shiravi, H.; Tavallae, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [\[CrossRef\]](#)
13. Xu, K.; Wang, F.; Gu, L. Behavior analysis of internet traffic via bipartite graphs and one-mode projections. *IEEE ACM Trans. Netw.* **2014**, *22*, 931–942. [\[CrossRef\]](#)
14. Qin, T.; Guan, X.; Wang, C.; Liu, Z. MUCM: Multilevel user cluster mining based on behavior profiles for network monitoring. *IEEE Syst. J.* **2015**, *9*, 1322–1333. [\[CrossRef\]](#)
15. Singh, R.; Kumar, H.; Singla, R.K. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Syst. Appl.* **2015**, *42*, 8609–8624. [\[CrossRef\]](#)
16. Jakhale, A.R. Design of anomaly packet detection framework by data mining algorithm for network flow. In Proceedings of the 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2–3 June 2017; pp. 1–6.
17. Parres-Peredo, A.; Piza-Davila, I.; Cervantes, F. Towards a User Network Profiling for Internal Security using Top-K Rankings Similarity Measures. In Proceedings of the 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, Spain, 5–7 July 2017; pp. 16–19.
18. Webber, W.; Moffat, A.; Zobel, J. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.* **2010**, *28*, 20. [\[CrossRef\]](#)
19. Mitra, S.; Mitra, S. *Digital Signal Processing: A Computer-Based Approach*; Connect, Learn, Succeed; McGraw-Hill: New York, NY, USA, 2011.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).