2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

RThread: A thread-centric analysis of security forums

Jakapun Tachaiya, Joobin Gharibshah, Evangelos E. Papalexakis and Michalis Faloutsos
University of California - Riverside, CA
Email: {jtach001,jghar002
@ucr.edu and {epapalex,michalis}@cs.ucr.edu

***Abstract*—Online forums have been shown to contain a wealth of useful information. With a few notable exceptions, such forums have not received much attention from the research community, unlike other online social media. Our goal here is to conduct an in-depth thread-centric analysis of online forums, focusing on security forums. We propose, RThread, a comprehensive unsupervised clustering approach with a powerful visualization component, which we provide as a publicly-accessible web-based tool. Our approach leverages 92 thread features that span three groups: (a) temporal, (b) behavioral, and (c) content related. We analyze data from 8 security forums with 400k posts over a span of 8 years. First, we find that many thread-centric properties follow a log-normal distribution, which is persistent across several forums and over time. Second, we show how our approach can identify clusters of threads with similar behavior, while our visualization component provides an easy way to spot the differences between these clusters. Finally, we show how our approach can spot surprising behaviors, including a cluster, whose threads are used for Search Engine Optimization. We see our approach and our publicly available platform as a building block towards understanding forum activity and extracting interesting information in an unsupervised way.**

Keywords: Online communities mining

I. Introduction

How can we identify interesting groups of threads in computer security forums? This is the motivating question of this project. Several recent works argue that there is a plethora of useful information in these forums [15], [21], [22]. To a large extent, the information is interesting, because of the wide spectrum of users that engage in these forums. They range from benign users that mainly discuss tips and tools for how to protect themselves from cyber attack, all the way to hackers who sell hacking tools and services.

We propose, RThread, a comprehensive thread-centric analysis approach with unsupervised co-clustering and powerful visualization capabilities. Our approach is: (a) **comprehensive**: it combines 92 features that span three types of features, including temporal, behavior and text; (b) **unsupervised**: it does not rely on training data and can uncover unexpected phenomena; and (c) **interpretable:** it provides an intuitive and visual interpretation of the resulting clusters. Our results can be summarized in the following points:

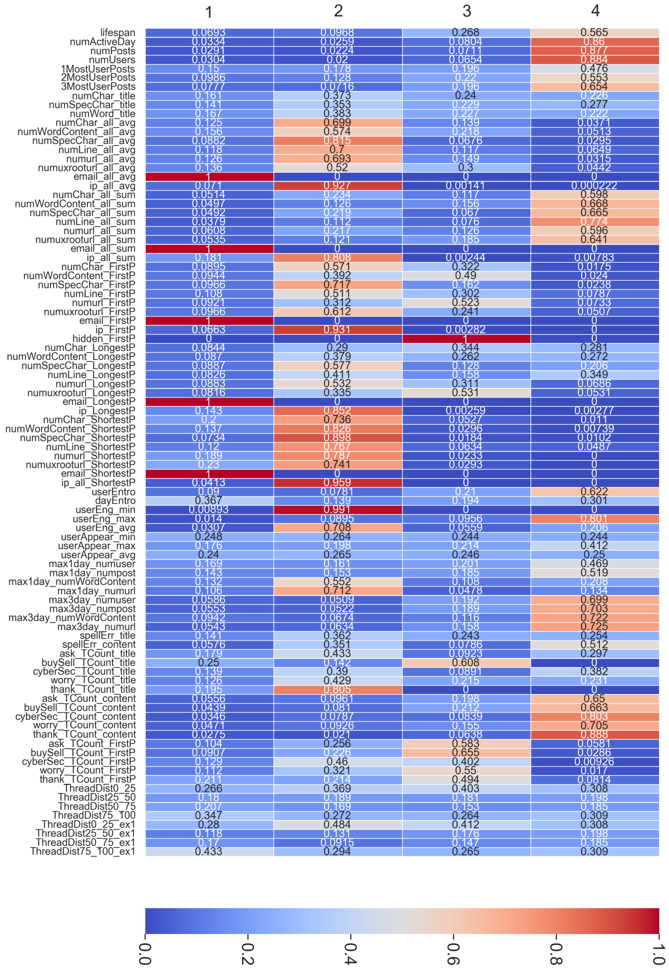


Fig. 1: Cluster Visualization for in Offensive Community: The color-coded average feature value per cluster captures the differences among the clusters in a visual and intuitive way. This clustering is derived by using SMR (20% Intensity threshold and K=4).

1. We develop a comprehensive soft co-clustering approach. We opt for soft co-clustering using the extensive set of features mentioned above. Our co-clustering does **feature selection and clustering simultaneously** by identifying the most appropriate set of features per user cluster. In addition, we develop a powerful visual way to capture the essence of each cluster, as shown in Fig. 1 for the Offensive Community forum.

2. We identify clusters with surprising behaviors. Our

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 2020 IEEE

unsupervised approach categorizes threads into clusters with different behaviors, which we outline in Table II. Among them, we identify two surprising clusters of threads. First, we find a cluster of "SEO" threads, which contain a large amount of incoherent text and many URL links to a few sites. Second, we identify "hidden" threads, which require users to register and post a reply to see the content of the post that initiated the thread.

3. We identify persistent thread properties. We find six properties of threads that follow a log-normal distribution with parameter values that are persistent over many years and comparable across many forums as we see in section II.

4. Catalyzing forum mining: platform and data. We have implemented our approach in a web-platform (http://rthread.ml). Our ambition is to make this a focal point for the research in this area by: (a) integrating more methods, and (b) providing forum data. It already provides most of the functionality here and by the time of publication it will have also all functions and data. Although we focus on security forums here, our approach and platform applies to any online forum.

II. Basic and Persistent Properties

Here, we discover fundamental and persistent thread-centric properties using capabilities from our platform.

Data. We study eight security forums which contain data ranged between 2010 and 2018 with the total of 47,000 users, 400,000 posts and 85,000 threads (details omitted due to space). The data comes from two main sources, our automated crawler and Cambridge Cybercrime Centre [1]. WilderSecurity [9] and Kernelmode [4] are considered to be *white-hacker forums* attracting IT professionals. By contrast, Offensive community [5], Garage4hackers [2], and Raidforums [6] are mainstream *dark forums*, where people often share tools and knowledge for hacking into systems. The rest of the forums, Greysec [3] Stresserforums [8] and Safeskyhacks [7] are in an in-between grey area.

We consider six thread-centric features: (i) *the number of new threads per day*, (ii) *the number of active threads per day*, (iii) *the thread lifespan*, (iv) *the number of active days in a thread*, (v) *the number of posts in a thread*, and (vi) *the number of users in a thread*.

Persistent log-normal distributions over the years and across forums. Five of the above features (except the number of active threads per day) exhibit a heavy-tail distribution especially pronounced in the large mainstream forums such as `Kernelmode` and `WilderSecurity` shown in Fig. 2. The CCDF of thread-centric features from (i) to (vi) in a log-log scale can be fitted with the log-normal distribution:

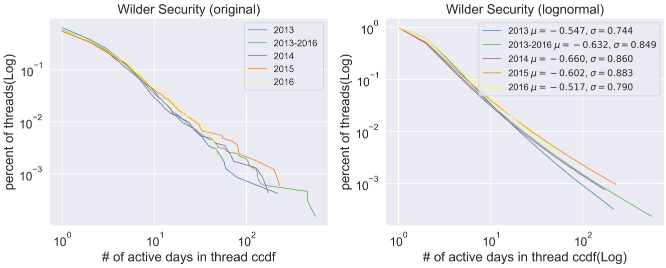
$$X = e^{\mu + \sigma Z} \tag{1}$$

where Z is standard normal variable, μ is a location parameter and σ is a scale/shape parameter.

Interestingly, the distribution parameters are fairly stable across years for each forum with a variance less than 0.04 for μ and 0.01 for σ

III. Unsupervised Thread clustering

We propose a comprehensive and systematic way to cluster threads into different categories in an unsupervised learning fashion. We consider two clustering methods here: a) the soft co-clustering, Sparse Matrix Regression or SMR method [11], b) K-Means [16] which we use mostly as reference. Our future work will consider more techniques, including a hierarchical and AutoEncoder-based [24] clustering.



(a) Original data plot (b) The fitted log-normal

Fig. 2: Many thread properties exhibit log-normal distribution which is persistence in its parameters and over time in several forums. Showing the distribution of the number of active days of a thread in CCDF (log) for WilderSecurity.

TABLE I: Brief overview of the 92 features used in clustering.

Type	Description	Num.
Temporal	Temporal feature capture thread properties in time domain, e.g. lifeSpan, #activeDays and dailyEntropy.	3
Behavioral	Behavioral feature tell how users interact within threads through posts, e.g. #posts, #users, threadDistribution, userEntropy - users' contribution in threads, and userEngagement - users' duration in threads.	24
Textual	Describing the content of the threads and their posts: #words, #characters, #lines, #URLs and #email, including intention/topic related features, e.g. asking words, thanking words etc.	65

Features. We use a total of **92 features** that can be grouped into behavioral, temporal and content related as shown in Table I. Due to space limitations and the sheer number of features, we can only provide a small subset of these features.

Clustering algorithms. We assume that K-Means is a well known algorithm, so we will only discuss the soft co-clustering approach, SMR. Given a matrix X of threads with 92 features, the soft co-clustering via SMR can be posed as the minimization of the loss function [11]:

$$||X - AB^T||_F^2 + \lambda \sum_{i,k} |A_{ik}| + \lambda \sum_{j,k} |B_{jk}| \tag{2}$$

where A and B are matrices of size I x K and J x K, respectively. K is a parameter that determines the number of clusters, and parameter λ controls how we calculate the relevance of a thread for each co-cluster. As we increase λ, we get sparser results, namely, fewer threads per cluster. We experimented with λ = 0.1, 0.2 and 0.4 and we selected λ = 0.1, which works well here.

Clustering inclusivity. In soft co-clustering, the algorithm allows overlapping members: each thread can belong to more than one cluster. In fact, the algorithm provides the **Intensity** value for each thread and cluster pair, which captures how strongly related is the thread with that cluster.

To assign threads to clusters, we use the **Intensity Threshold**: only threads with Intensity value above the Intensity Threshold will be included in that cluster. In more detail, we compare (and normalize) the Intensity of each thread with respect to the maximum observed Intensity across all threads for that cluster, thus the threshold becomes a percentage of the highest observed Intensity for that cluster. We evaluated the following values 5%, 10%, 20% and 40% of the maximum threshold. The higher number of threshold result fewer the member in each cluster. Note that the same reasoning applies for assigning features to a cluster. We use the same 20% threshold for assigning features to clusters, which gives good results here.

In this paper, we consider three algorithms: (a) K-means using the full set of 92 features, (b) SMR with a 20% Intensity Threshold, and (c) K-means-42 using the subset of 42 features that have intensity value more than 20% Intensity Threshold in SMR. The third algorithm was introduced to answer the following question: would K-means perform better, if we select the more discriminating features that we identify with our SMR algorithm? Also note, that with soft co-clustering, a thread can belong to multiple clusters. To compare SMR with K-Means, we associate each thread with the cluster for which the thread has the highest Intensity value.

A. Evaluating the clustering. To evaluate the clustering solutions, we use the average Silhouette coefficient [20]. The coefficient measures how similar is each thread to its assigned cluster compared to other clusters. Its value ranges from -1 to 1, and the higher the co-efficient value the better the clustering is. We measure the average Silhouette coefficient for each forum as a function of the number of target clusters as shown in Fig. 3, which we discuss below.

A.1. Selecting the right number of clusters. This is a key question in every clustering problems [25]. For now, this parameter is provided by the end-users, which empowers them to tailor the query to the question of interest. In Fig. 3, the knee of the curve appears between 4-6 clusters, which is the range that we used.

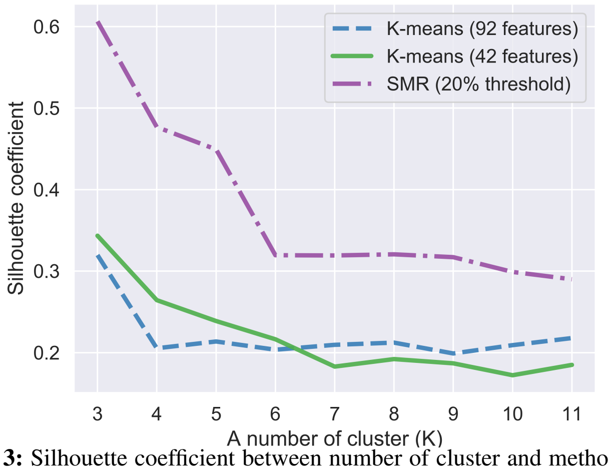


Fig. 3: Silhouette coefficient between number of cluster and methods from K=3 to K=11. It shows that soft co-clustering perform almost two times better than K-means.

A.2. Soft co-clustering outperforms the K-Means algorithm. From our experiments in Fig. 3, we find that the SRM co-clustering has almost double the Silhouette coefficient of compared to both K-mean algorithms (using 92 and 42 features). The poor performance of K-Means could be partially attributed to the large number of features. To address this, we identify a "better" set of features with higher discriminatory capability, namely, 42 features that have Intensity value more than 20% in SMR (K-Means-42). However, this did not improve the results: K-Means-42 does not exhibit consistent or statistically-significant improvement as shown in Fig. 3.

TABLE II: The different types of clusters identified by our unsupervised learning methods.

Type	Description
Ephemeral	One post and live for one day.
Long-lived	Long lifespan and high # of active days.
Hidden	Hide some part of their contents.
Long-post with URL	Threads with posts containing URLs and high # of words.
SEO	Threads with posts with repeating URLs and high # of incoherent words.

B. A Visual and Intuitive cluster analysis: To facilitate the interpretation of the clustering results, we propose the use of color-coded table as shown in Fig. 1. In this plots, we calculate the mean value of each feature over all the threads for each cluster. Dark blue indicates low values, while dark red indicates high values. We demonstrate the power of the visualization in Fig. 1, where we show the clustering of Offensive Community for four clusters. In a figure, on the top left corner, we see dark blue, which suggests that threads in cluster 1 have low number of users, posts, lifespan and active days. This cluster represents the large "low activity" threads, which is aligned with the skewed distribution of the section II. Similarly, cluster 4 consists of long-lived threads with many user contributors. This group corresponds to the "heavy hitter" threads at the tail of the skewed distribution of the previous section.

IV. Identifying interesting clusters

Here, we apply our approach on our security forums in order to provide an indication of the types of results we could derive. Specifically, we used our soft co-clustering approach with 20% Intensity Threshold, 0.1 λ and K = 4 on forums.

We identify groups of threads with distinctive behaviors, as we show and define in Table II. Note that due to space limitations, we do not present the exact threshold-based definitions of these cluster types, which involve metrics such as average thread life-span, number of active days, number of posts, number of URLs etc. We discuss each type of cluster below.

a. Identifying "Ephemeral" threads. For every forum, our clustering identifies a large cluster of primarily short lived threads, which one could have anticipated given the the skewed distribution in the size of the threads in section II. We use the term "ephemeral" to refer to a threads with only one post. We find clusters dominated by such "ephemeral" threads in all forums. These clusters can be observed by our mostly blue colors in the top part of the plot, shown in cluster 1 for Offensive Community in Fig. 1.

b. Identifying "Long-lived" threads. Some of the emerging clusters seem to be dominated by threads of long life-span (difference between first and last post). In fact, some of these threads span four years! Most of these threads are sharing information, discussion of technologies and announcements. We are able to recognize Long-lived clusters, which can be observed by our color-coded tables, shown in cluster 4 for Fig. 1. Long-lived threads appear in almost every forum, but as a small percentage of the total number of threads, which is

TABLE III: The types of "hidden" threads.

type	Note	#
Hacking tool	Rooting Android and a key-logger	3
Hacking tutorial	Range from server vulnerability to phishing for credit cards	8
Illegal dist	Games & other software	4
Selling/buying	Rooted accounts, websites and shell scripts for hacking	2
Boasting	Bragging about their hacking success	3
Benign tutorial	Web & Windows app's tutorials	3
Benign tool	Web & Windows plugins and tools	4
Sharing info	News & tips in computer security	3

aligned with the skewed distribution seen in section II.

c. Identifying "Hidden" threads. In Offensive Community, we found a cluster of threads that hide their content. These threads are always initiated by a post that requires the viewer to register as a member in the forum and post a reply to see the hidden content. It is natural to assume that this technique hides the content from an automated crawler, which will, most likely, not perform the unlocking behavior. In more detail, this cluster consists of 30 threads all of which are initiated by such a "hidden" post. Most of the replies are short “thank you" posts. The short first post, the keywords in the post, and the short "thank you" replies, are the characteristics of the threads, which our algorithm used to form the cluster.

What do these threads hide? Intrigued, we investigated 30 of these "hidden" threads. We responded with a post, and we got access to the hidden information. We found several questionable content, including hacking tutorials, hacking tools and illegal distributions of cracked software, as we list in Table III. Also, one of those posts is a boasting post about their achievement of hacking into some well-known systems, such as Google's Morocco server in 2013.

d. Identifying “Long-post with URL” threads. In some clusters, we saw threads containing a moderate number of words and URLs in their posts. We use the term "Long-post with URL" to describe such threads. Most of these threads are sharing news and some information with one or more hyperlinks. These hyperlinks point to a source of news, an image file or a file-sharing sites. In our analysis, we find "Long-post with URL" clusters in three forums, Garage4hackers, Safeskyhacks and Offensive Community (cluster 2 in Fig. 1).

e. Identifying “SEO" threads.. In Safeskyhacks, we identified a cluster of threads, which we suspect engage in Search Engine Optimization (SEO) boosting. Specifically, we find that cluster 3 of Safeskyhacks is identified as a "Long-post with URL" type. On closer inspection though, we find that it is different from the other clusters of the same type of other forums. Most of its threads have one post, which contains approximately 1k - 2k words. Upon further inspection, these posts contain *"a large amount of out-of-context text"*. The structure of these posts follows a repetitive pattern: three of four paragraphs, separated by the same image, and one or more URL links. All embedded hyperlinks in a thread typically point to the same website.

Which are the sites that benefit from "SEO" threads? We list the top five most highly-linked sites from the "SEO" threads in table IV. The top site is a gaming site, aleasegames.com, and it is pointed-to from 682 places in the cluster. Moreover, one of those highly referred site, *elitegamersclub.com*, are selling their domain name. Note that goo.gl is Google's URL shortening service. Some of those goo.gl URLs are hosting downloadable zip files, which could be malicious, and we intend to analyze this in more detail in future work.

TABLE IV: The most referred sites in the "SEO" cluster.

Sites	**IP Location - Host**	**Links**
ateasegames.com	London - Hydra Comm. Ltd.	682
elitegamersclub.com	Virginia - Amazon.com Inc.	430
goo.gl	Amsterdam - Google LLC	369
legalaidreform.org	San Jose - Websitewelcome	266
rindfleisch.reisen	Hong Kong - Host Europe Gmbh	264

V. Related Work

We briefly discuss two categories of relevant research. An extensive listing is not possible due to space limitations, but we will provide it in a subsequent full version of this work.

a. Analyzing computer security forums. Most works in this domain focus on finding function, intention, product, and services in posts. The [19], [12] make use of hand labeling data and NLP techniques in supervised classification to get function and intention of posts as well as a name and a price of product and service in a post. CrimeBB [18] is arguably the first security forum repository, which also reports on high-level trends, such as a number of threads, posts, and users. Some [21], [13], [15] uses data in forums with NLP techniques to predict a cyber attack.

b. Analyzing trends and anomalies in social media. Online social media, like Twitter and Facebook, have been studied extensively. For example, a few recent studies studies [22], [14], [17] use machine learning and data mining to detect behavioral trends and anomalies in social media platforms, such as Facebook and Reddit. There as well, several features exhibit a heavy tail distribution, similarly to our observations. Other studies focus in identifying group of users with similar behaviors. Many [23], [10] use community detection techniques to extract a group of key users with similar behavior.

VI. Conclusion

We propose, RThread, a comprehensive unsupervised co-clustering approach with visualization capabilities. Our approach provides a systematic and in-depth thread-centric analysis of online forums using We consider 92 thread features We also propose a visualization method to aid the interpretation of clusters in an intuitive way. First, we find that many properties follow a log-normal distribution, which is persistent across several forums and over time. Second, we show how our approach can identify classes of threads with similar behavior, revealing some unanticipated thread behaviors.

This preliminary work shows significant promise as a building block towards fully harnessing the wealth of information in online forums. Its unsupervised nature is a significant advantage, as it can explore and detect behaviors that we are not anticipating.

References

[1] Cambridge cybercrime centre. https://www.cambridgecybercrime.uk/.

[2] Garage4hackers. http://garage4hackers.com.

[3] Greysec. https://greysec.net/.

[4] Kernelmode. https://www.kernelmode.info/forum/.

[5] Offensive community. http://www.offensivecommunity.net.

[6] Raidforums. https://raidforums.com/.

[7] Safeskyhacks. http://www.safeskyhacks.com/.

[8] Stresserforum. https://www.stresserforums.co/.

[9] Wilders security. http://www.wilderssecurity.com.

[10] T. Anwar and M. Abulaish. Ranking radically influential web forum users. *IEEE Transactions on Information Forensics and Security*, 10(6):1289–1298, June 2015.

[11] R. Bro, E. E. Papalexakis, E. Acar, and N. D. Sidiropoulos. Coclustering—a useful tool for chemometrics. *Journal of Chemometrics*, 26(6):256–263, 2012.

[12] A. Caines, S. Pastrana, A. Hutchings, and P. J. Buttery. Automatically identifying the function and intent of posts in underground forums. *Crime Science*, 7(1):19, Nov 2018.

[13] A. Deb, K. Lerman, E. Ferrara, A. Deb, K. Lerman, and E. Ferrara. Predicting Cyber-Events by Leveraging Hacker Sentiment. *Information*, 9(11):280, nov 2018.

[14] P. Devineni, D. Koutra, M. Faloutsos, and C. Faloutsos. If walls could talk: Patterns and anomalies in facebook wallposts. In *Proceedings of the 2015 IEEE/ACM International Conference on*, ASONAM '15, pages 367–374, New York, NY, USA, 2015. ACM.

[15] J. Gharibshah, T. C. Li, M. S. Vanrell, A. Castro, K. Pelechrinis, E. E. Papalexakis, and M. Faloutsos. Inferip: Extracting actionable information from security discussion forums. In *Proceedings of the 2017 IEEE/ACM Conference on*, ASONAM '17, pages 301–304, New York, NY, USA, 2017. ACM.

[16] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[17] T. C. Li, J. Gharibshah, E. E. Papalexakis, and M. Faloutsos. Trollspot: Detecting misbehavior in commenting platforms. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, 2017.

[18] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton. Crimebb: Enabling cybercrime research on underground forums at scale.

[19] R. S. Portnoff, S. Afroz, G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson. Tools for automated analysis of cybercriminal markets.

[20] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, Nov. 1987.

[21] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara. Early warnings of cyber threats in online discussions. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 667–674. IEEE, 2017.

[22] S. Thukral, H. Meisheri, T. Kataria, A. Agarwal, I. Verma, A. Chatterjee, and L. Dey. Analyzing behavioral trends in community driven discussion platforms like reddit. *CoRR*, abs/1809.07087, 2018.

[23] N. Vo, K. Lee, C. Cao, T. Tran, and H. Choi. Revealing and detecting malicious retweeter groups. In *Proceedings of the 2017 IEEE/ACM International Conference*, ASONAM '17, pages 363–368. ACM, 2017.

[24] J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2015.

[25] M. Yan. Methods of determining the number of clusters in a data set and a new clustering criterion. 01 2005.