Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence

SAGAR SAMTANI, RYAN CHINN, HSINCHUN CHEN, AND JAY F. NUNAMAKER., JR.

SAGAR SAMTANI (sagars@email.arizona.edu; corresponding author) is a doctoral student in the Department of Management Information Systems and a research associate in the Artificial Intelligence Lab at the University of Arizona. He is also a fellow in the National Science Foundation Scholarship-for-Service program. His research focuses on the cyber security domain by examining online hacker communities and developing Internet-scale vulnerability-assessment approaches. His work has appeared in various conference and workshop proceedings, including the IEEE Conference on Intelligence and Security Informatics, the INFORMS Annual Meeting, and the Women in Cybersecurity conference.

RYAN CHINN (rmc1@eller.arizona.edu) earned his M.S. in management information systems from the University of Arizona. He was also a fellow in the National Science Foundation Scholarship-for-Service program. He works at the U.S. Department of Commerce.

HSINCHUN CHEN (hchen@eller.arizona.edu) is University of Arizona Regents Professor and Thomas R. Brown Chair in Management and Technology in the Management Information Systems Department at the Eller College of Management, University of Arizona. He received his Ph.D. in information systems from New York University. He is director of the Artificial Intelligence Lab, where he developed the COPLINK system, which has been cited as a national model for public safety information sharing and analysis, and has been adopted in more than 3,500 law enforcement and intelligence agencies. He is the author or editor of 20 books, 25 book chapters, 280 journal papers, and 150 refereed conference articles covering digital library, data/text/web mining, business analytics, security informatics, and health informatics. He is editor in chief of *Security Informatics*. He has received over 90 grants totaling more than $40 million in research funding from the National Science Foundation, National Institutes of Health, National Library of Medicine, Department of Defense, Department of Justice, Central Intelligence Agency, Department of Homeland Security, and other agencies. He is a fellow of ACM, IEEE, and AAAS.

JAY F. NUNAMAKER JR. (jnunamaker@cmi.arizona.edu) is Regents and Soldwedel Professor of MIS, Computer Science and Communication and director of the Center for the Management of Information and the National Center for Border Security and Immigration at the University of Arizona. He received his Ph.D. in operations research and systems engineering from Case Institute of Technology. He has held

Journal of Management Information Systems / 2017, Vol. 34, No. 4, pp. 1023–1053.

ISSN 0742–1222 (print) / ISSN 1557–928X (online)
DOI: https://doi.org/10.1080/07421222.2017.1394049

a professional engineer's license since 1965. He was inducted into the Design Science Hall of Fame and received the LEO Award for Lifetime Achievement from the Association for Information Systems. He was featured in the July 1997 issue of *Forbes Magazine* on technology as one of eight key innovators in information technology. His specialization is in the fields of system analysis and design, collaboration technology, and deception detection. The commercial product GroupSystems ThinkTank, based on his research, is often referred to as the gold standard for structured collaboration systems. He founded the MIS Department at the University of Arizona in 1974 and served as department head for 18 years.

ABSTRACT: Cyber attacks cost the global economy approximately $445 billion per year. To mitigate attacks, many companies rely on cyber threat intelligence (CTI), or threat intelligence related to computers, networks, and information technology (IT). However, CTI traditionally analyzes attacks after they have already happened, resulting in reactive advice. While useful, researchers and practitioners have been seeking to develop proactive CTI by better understanding the threats present in hacker communities. This study contributes a novel CTI framework by leveraging an automated and principled web, data, and text mining approach to collect and analyze vast amounts of malicious hacker tools directly from large, international underground hacker communities. By using this framework, we identified many freely available malicious assets such as crypters, keyloggers, web, and database exploits. Some of these tools may have been the cause of recent breaches against organizations such as the Office of Personnel Management (OPM). The study contributes to our understanding and practice of the timely proactive identification of cyber threats.

KEY WORDS AND PHRASES: cyber attack identification, cyber threat intelligence, hackers, hacker forums, hacker tools, proactive, social networks, topic modeling.

Computer technology allows modern organizations to conduct their operations with a level of convenience and efficiency like never before. Unfortunately, many individuals with illicit cyber intent, also known as malicious hackers or cybercriminals, often leverage dangerous cyber tools or assets to conduct destructive cyber attacks against technologically driven organizations. Cyber attacks, or the deliberate exploitation of computer systems through the use of malicious tools and techniques such as Ransomware, Zeus Trojans, and Keyloggers, cost the global economy approximately $445 billion per year and have negatively affected health-care organizations like Premera Blue Cross, government entities such as the Office of Personnel Management (OPM), and large retail and consumer companies including Target, Home Depot, Sony, and Xbox Live [16, 29, 30, 41, 49, 52, 56, 60]. The past few years have seen an unfortunate and disruptive growth in the number of cyber attacks [22].

To help mitigate cyber attacks, companies such as FireEye and Cyveillance provide cyber threat intelligence (CTI) reports designed to help organizations protect against cyber attacks. To create their reports, these companies rely on data collected from actual attacks or events through mechanisms such as network logs, antivirus logs, honeypots, database access events, system login attempts, and intrusion

defense system/intrusion protection system (IDS/IPS) events [54]. The intelligence provided is reactive rather than proactive, as it is based on data from attacks that have already happened. The reports do not provide intelligence on tools that hackers have developed but not yet used for cyber attacks that also have the potential to cause great damage (e.g., zero-day attacks). Additionally, these reports often ignore the specific actors who are responsible for such exploitations, resulting in an incomplete picture of the overall hacker ecosystem (e.g., communities, motivations, specialties, tools, etc.). These shortcomings have led industry leaders such as Ernst and Young to note that traditional CTI is not "sufficient to properly address risk in individual organizations" and that "organizations need to take a more proactive approach to cybersecurity" [17, p. 1].

To address some of the faults in traditional CTI, industry and academia alike have emphasized the need to develop more comprehensive and proactive CTI by directly collecting, identifying, and analyzing data from the online hacker community to better understand malicious tools and individuals [36, 39]. While the online hacker community contains a variety of components (e.g., underground economies, Internet Relay Chat [IRC] channels, etc.), hacker forums in particular provide the technical mechanisms for hackers to easily provide and acquire freely accessible, malicious tools such as Zeus, Ransomware, SQL injections, and DDoS, among others [8, 10, 51]. Overall, there are hundreds of hacker web forums across geopolitical regions such as the United States, Russia, and China. The forums contain tens of millions of postings and cover a wide variety of topics, such as underground economies, data breaches, and cyber warfare. Also included are tens of thousands of malicious assets created by millions of hackers, many of which pose great danger to organizations. Given the hacker forum scale and that hacker forum tools have been used to attack organizations (e.g., BlackPOS code was available in forums months before the Target attack), hacker forums can also serve as viable data sources for understanding emerging tools and assets [9, 29, 40, 51] and can contribute to more accurate, proactive, and comprehensive CTI.

Directly collecting and analyzing large amounts of hacker forum data present unique technical challenges that limit researchers' abilities to produce comprehensive CTI studies. These challenges include vast amounts of textual data, robust anticrawling measures, foreign-language barriers, little-known hacking terms, and complex forum structures. However, given the recent interest in gathering intelligence directly from hackers and the growing emphasis of security Big Data studies in information systems literature [2, 12], one purpose of this research is to contribute to information systems literature by developing a novel CTI framework leveraging principled web, text, and data mining techniques to methodically collect, identify, and analyze malicious hacker assets in underground forums. Specifically, we analyze these assets by applying state-of-the-art techniques such as support vector machine (SVM) and latent Dirichlet allocation (LDA) to identify and understand the implementations of such assets, and leverage rich forum metadata to understand the key trends of malicious cyber assets. We also employ bipartite social network analysis techniques to identify key hackers for selected malicious assets. The practical results of this research have the potential to mitigate future cyber attacks, thus reducing the

overall cost of cyber crime to the global economy. Additionally, the tools that are identified in this analysis can be presented through a cyber security education portal designed to help cyber security educators and students enhance their understanding of malicious tools and assets.

Related Work

Hacker Community Research

Hackers congregate on a variety of online platforms such as IRC channels, carding shops, and hacker forums to exchange content and knowledge [9, 10]. While all of these platforms are rich data sources for research, hacker forums are a unique and particularly fruitful platform for identifying malicious assets as they allow users to easily and systematically post, save, and retrieve certain types of content (e.g., source code, malicious files) that other online platforms do not [8, 9, 10]. Additionally, forums have seen a consistent level of usage by hackers over a long period of time and have broad geopolitical coverage and topical interests [10]. As a result, researchers have the potential to conduct large-scale, diverse, longitudinal analyses. In general, U.S. forums focus primarily on cyber crime and general hacking, Russian forums on underground economies and data breaches, and Chinese forums on cyber warfare and virtual goods [20]. Various subforums within the overall forum are dedicated to specific subareas. For example, a hacker forum specializing in malware may have subforums dedicated to Ransomware or Zeus code. Within each subforum, members can create and post in specific discussions (i.e., threads). In general, the subforum and thread titles contain specific keywords or phrases that allow users to identify the purpose of each area such that they can post and acquire content. While the specific posting privileges and mechanisms vary based on forum and forum standing, users can generally share hyperlinks, pictures, videos, source code, attachments, and other resources with each other in these threads. Figure 1 illustrates how users can navigate from the overall forum to a thread where they can acquire and post malicious source code within the popular hacker forum OpenSC.

Researchers who are interested in studying such forums still face numerous technical challenges. Many hacker forums have rigorous vetting processes for prospective members, have postings often in foreign languages, contain unfamiliar hacking related terms, and inconsistent subforum structures that make categorization of content difficult. In addition, many forums employ sophisticated anticrawling measures that make comprehensive data collection difficult. These technical challenges have generally resulted in researchers manually collecting small sets of data and applying qualitative techniques to understand the social interactions and networks of hacker community members. Though these techniques suffer from limits on their scalability, such studies provide valuable insight into the behaviors of hacker forum members, noting that regardless of geopolitical orientation (Russian, English, etc.), the majority of forum participants are unskilled and only a small percentage of hacker members possess a high level of skill in creating and disseminating malicious

Figure 1. Example of Forum Navigation. (a) From the main forum page (far left) a user can access the "C Malware Sources" subforum. (b) The malware sources link takes the user to a page listing numerous threads providing access to malicious source code. (c) The link labeled "Simple HTTP Bot" takes the user to a page with an HTTP Bot designed to flood a network.

assets [14, 25, 26, 40, 48, 62]. Additional studies using automated techniques such as deep learning-based sentiment analysis, interaction coherence analysis, and regression analysis have discovered that the quality and volume of hacker assets a member posts are key factors behind their reputation and forum standing [1, 8, 32].

While it is clear that hacker assets help drive hacker reputation and standing, only a few studies have tried to identify what these tools or assets actually are. These studies have primarily used manual explorations or interviews with a subject matter expert (SME) to identify hacker forum assets. Hacker forum assets usually come in three main forms: source code, attachments, and tutorials [51]. Source code is code written in a programming language embedded within a forum posting. Hacker forum code, much like code found in the popular forum Stack Overflow, is generally raw, incomplete, and cannot be executed without a programming environment. Hacker source code can range from benign, general purpose programming code (e.g., web development code) to more malicious, damaging code such as SQL Injections and Zeus Trojans. Attachments, on the other hand, are files attached to forum postings. These may be harmless files (such as general programming books) or malicious executables and tools (such as Ransomware). Finally, tutorials are forum posts designed to educate other hackers on specific topics (for example, creating a cyber weapon or conducting a phishing attack). Generally, the post content found with the asset is descriptive of the purpose of the asset.

Ablon et al. [3] discovered that payloads, full services, and credit card information is available in hacker black markets, while Chu et al. [14] and Samtani et al. [51] found freely accessible SQL injections, banking vulnerabilities, and Zeus Trojans in hacker forums. Ablon et al. [3] further noted that malicious code assets are the most abundant asset within hacker forums and are often used to facilitate cybercriminal activities. However, given that most of these studies leveraged manual techniques or interviews with SMEs, they cannot be scaled to larger amounts of data. Furthermore, they do not leverage the rich forum metadata (e.g., author names, post dates, etc.) available with the forums to conduct deeper analyses or systematic identification of all malicious hacker assets. Given these limitations and the inherent strengths of

hacker forum platforms, we next review the cyber threat intelligence literature to gain perspective on the data sources and the analytical procedures used to create valuable intelligence.

Cyber Threat Intelligence (CTI)

The SANS Institute defines cyber threat intelligence as "threat intelligence related to computers, networks, and information technology" [18]. CTI has traditionally been motivated by the desire for companies and individuals to better protect their cyber infrastructure from an attack [18, 54]. Today, many vendors, such as Symantec, McAfee, Trend Micro, FireEye, Cyveillance, Sophos, and Kaspersky, have established themselves as leaders in developing and providing CTI to other entities in the government and commercial spaces. Many companies also support their own internal CTI divisions to produce customized reports. Regardless of company, data are generally gathered from a variety of sources including network traffic logs, database access events, configuration modification logs, IDS/IPS events, login and access logs, external activity to commonly hacked ports (e.g., 1080, 21, 22, 23, 3306, etc.), honeypot logs, antivirus logs, and so on [54]. While these data sources are relatively comprehensive in terms of the network data gathered, their inherent weakness is that they do not integrate information from or about the actual communities from which the attacks originate (a data source that can contain a vast amount of valuable data), and instead rely on their own systems to log actual cyberattacks [54]. In addition, analyses conducted on the collected data generally opt for the calculation and visualization of basic descriptive statistics, often in the form of real-time streams, dashboards, and reports. For example, many CTI reports contain information about the malware variants that were picked most frequently by their sensors, various statistics about network traffic, suspicious URLs to monitor, and blocked attacks [54]. Some reports also leverage attacker IP addresses gathered by honeypots and other network logging software to provide geographical overviews of countries conducting specific types of attacks. Temporal analyses may also be used to illustrate the growth or decline in specific types of malware or certain cyberattacks.

Though there is great value in the prevailing CTI process and reporting style, existing CTI reports have been criticized for being too high-level and for being merely reactive to already known threats [17]. Furthermore, the reports do not detail the specifics of the exploits, such as implementation methods or programming language. Such information has proved to be valuable for companies: knowing the method of implementation for an exploit can aid in building more robust cyber defenses. Unfortunately, given that traditional malware analysis techniques generally look at one malware binary at a time, the sheer volume of hacker assets, and the fact that the majority of assets in hacker forums are source code [51], traditional malware analysis methods cannot be used.

However, the techniques detailed in source code analysis literature can be adopted for useful and advanced hacker source code analysis.

Source Code Analysis

Source code provides content and structure advantageous to the discovery of its technical implementation (i.e., programming language) as well as its purpose. Given the large amount of source code in hacker forums, we review and borrow techniques from two subareas of source code analysis literature: source code classification and source code topic extraction. Both applications have been used extensively in the context of mining software forum repositories.

Source code classification research is motivated by the desire to improve software reuse and organization and focuses primarily on classifying code in online repositories (e.g., SourceForge or Ibiblio) into predefined categories such as databases, games, e-mail, attack vectors, or other domain-specific areas [38, 59]. The general approach to classifying source code is to identify target classes for classification, develop a training set with sample source code from each class, and use textual features to train the classifier for unseen code [36, 38]. Various classifiers have been evaluated, with SVM consistently showing the highest performance [36, 38, 59]. Code can also be classified into programming languages by training a classifier on sample source code from several different languages and applying the trained classifier to unseen code files to classify them into their designated languages. Prior work has found that SVM classifiers using term frequencies as features have the most effective performance for this task [59]. Even so, these classifiers do not identify the function the code serves unless it is already in a predefined domain or manually executed. However, a second subarea of source code literature (code topic extraction) focuses on using topic modeling techniques such as latent Dirichlet allocation (LDA) to automatically identify topics and applications of large amounts of source code whose purposes are unknown [13].

LDA is a robust generative probabilistic model used to automatically discover latent topics within large text corpora [11]. Based on a set of parameters (predefined number of topics to be extracted, iterations, etc.), LDA extracts a set of topics from a collection of textual documents. Each topic is a distribution of words, and each document is a mixture of corpus-wide topics. Generally, modeling a large number of topics produces fine-grained results, while modeling fewer topics returns coarser outputs [6, 7].

Compared to classification models, LDA lacks rigorous quantitative internal and external validation metrics partially due to its unsupervised nature. LDA is often evaluated through two standard techniques: manual evaluation and a statistical metric, perplexity [11]. In the manual approach, the outputted topics are qualitatively evaluated for their coherence and clarity by several experts, with interrater reliability being a commonly used technique to ensure concordance between topic evaluators [13]. On the other hand, perplexity is a quantitative measure designed to

mathematically calculate the optimal number of topics to model for [11]. To calculate perplexity, a small set of the text corpus is held out as testing data, while the majority of the corpus is used to train the LDA model. Perplexity measures the likelihood that the held-out data are truly generated from the underlying topic distributions based on the inverse log likelihood of the held-out documents. Lower perplexity score indicates better topic model match [11].

Although LDA is traditionally applied to natural language documents, it has also been used to discern the topics in and purpose of code in code repositories such as Stack Overflow [4, 5, 7], SourceForge [6, 34, 35, 61] or large software systems such as Hadoop or Petstore [21, 37, 58]. However, source code is not natural language and any comments or post content must be preprocessed before adopting LDA. Standard preprocessing steps include special character removal (e.g., quotes, hyphens, underscores, etc.), identifier splitting (e.g., "dataAuthResponse" becomes "data Auth Response"), case-folding (e.g., "Response" to "response"), and stopword removal [13]. After preprocessing, studies typically extract between 40 and 150 topics from the code. Topics are manually labeled after extraction. However, the majority of these studies did not use perplexity to statistically evaluate the optimal topic number to model for, instead opting for manual evaluation [4, 5, 7, 33] or no evaluation [6, 35, 37, 53, 58].

Although LDA can extract topics from code, it cannot identify who disseminates the code in a forum context. However, given that postings have explicit author metadata, the posts related with certain topics from the LDA results can be paired with social network analysis (SNA) visualization and statistical measures to understand key individuals for particular subjects. Indeed, this combination of techniques has been used to great success in other literature studying the Dark Web and other forum contexts [31, 50].

Social Network Analysis (SNA)

SNA builds on graph theory to study the relationships between social entities and the implications of these relationships. Social networks are represented with two major components: nodes and edges. Nodes represent social entities such as individuals, organizations, social media members, and so on. Edges represent the communication ties, relations, or linkages between nodes, and can be directed (one-way relationship) or undirected (mutual, two-way relationship).

Traditional SNA is useful for studying phenomena such as technology diffusion between individuals, but is less applicable in determining the relationships between two different types of entities, such as hackers and assets. However, an affiliation network (also known as a two-mode or bipartite network) is a special type of network that can model such relationships [19]. This type of network partitions nodes into two sets: individuals and events, wherein one node type (individuals) is "affiliated" with the other node type (events). Relationships in this setup are directed and are modeled between the nodes, but not within. This means that nodes

representing individuals cannot have a relationship with other individuals, only with events. Bipartite networks are useful when trying to observe relationships between different types of nodes (i.e., between individuals and events). Bipartite networks are used in a variety of contexts such as Wikipedia coauthorship [27], knowledge translation in health forums [55], and sexual relationship networks [19]. Despite their flexibility in modeling different contexts, the main disadvantage of bipartite networks is that they can generate only a subset of the well-known network metrics [19]. To calculate all network metrics (e.g., distance, length, diameter, and radius) and node-level metrics (e.g., degree, closeness, betweenness, eigenvector), bipartite networks are often projected as monopartite networks (i.e., networks with only one node type).

Research Gaps and Questions

Several research gaps were identified from our literature review of hacker communities, cyber threat intelligence, source code analysis, and social network analysis research. Hacker community research has focused on general hacker interactions and the identification of key hackers within communities. Few studies have examined hacker forum content or assets. Moreover, these studies' approaches have primarily been manual and qualitative, and not scalable or comprehensive enough for analyzing a large volume of assets. Previous studies have also not leveraged forum metadata to conduct deep, comprehensive analyses for valuable CTI. CTI reports are based primarily on data from attacks that have already occurred, and do not consider attackers' ecosystems. As a result, the reports are reactive to current threats in cyberspace rather than proactive in identifying future threats. Source code literature heavily emphasizes classification and topic modeling of large amounts of code found in online software repositories or systems, but few studies have analyzed hacker community source code. Finally, social network analysis has been used sparingly to depict hacker relationships and interactions. Furthermore, although traditional approaches are useful for identifying relationships between hackers, they do not fully represent the relationships between hackers and assets. While variations of the traditional analysis (i.e., bipartite networks) can help in this regard, we were unable to find research leveraging these techniques and their associated centrality measures to mathematically or visually represent the relationships between hackers and assets. Based on these research gaps and the desired research outcomes, the following research questions are proposed for this study:

- What types of hacker assets are available in underground communities?
- What are the characteristics and functions of these hacker assets?
- Who are the key hackers disseminating hacker assets in underground forums?

This study addresses the research gaps by creating a principled, automated, and scalable framework for proactive CTI. This framework collects a large and novel data set of hacker assets directly from the hacker community and uses state-of-the-art

topic modeling and classification techniques with forum metadata to systematically identify and gain a deep understanding of this large volume of assets. Bipartite social network analysis techniques are also used to computationally identify key hackers for selected assets. Such analysis addresses some of the current limitations present in traditional malware analysis and CTI methodologies.

Research Testbed and Design

Our hacker asset analysis framework (Figure 2) comprises three major components: a data collection and data preprocessing component, an asset analysis and evaluation section, and a social network construction component. These components are detailed in the following subsections.

Data Collection and Preprocessing

Similar to prior hacker forum literature, one English and six Russian hacker forums were identified for collection and analysis. These seven forums were selected from among hundreds of hacker forums for several reasons. First, these forums were suggested for examination by several cyber security experts. Second, English and Russian hacker communities are notorious for creating and using malicious cyber assets for cyber attacks [26]. Third, these forums can be accessed without an invitation, thus reducing the risk of researcher identification. Finally, these forums are well-known in the hacker community for containing a plethora of malicious assets.

To circumvent the forums' anticrawling mechanisms such as checking of user-agents, username and password authentication, timing out of sessions, and so on, a Tor-routed web crawler using forum credentials downloads and stores all forum

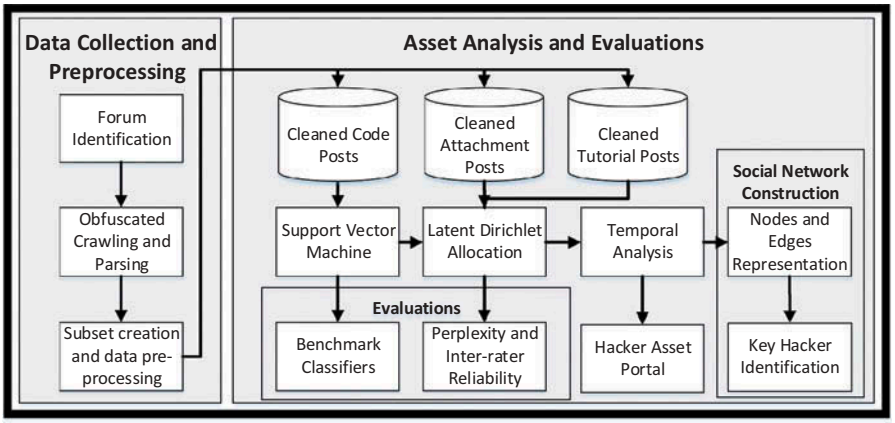

Figure 2. Methodological Research Framework

HTML web pages onto local hard disks for offline browsing and further processing. Since forums create special HTML tags and structures every time a user posts an asset (source code, attachment, or tutorial), we use a specialized parser to identify such structures and parse all of the post, asset, thread, and author data into the database. After collection and parsing, the source code, attachment, and tutorial posts are placed into separate subsets using SQL queries. If a post contains any source code, it is placed in the code subset. Regardless of subset, we use Google Translate to translate all posts to English, remove special characters, split identifiers (e.g., readFile to read File), fold case, and remove stop-words. Table 1 summarizes the final collection.

The collection contains 431,518 posts in 101,711 threads made by 40,372 authors between November 4, 2004, and September 15, 2015. The Russian forum ExploitIN and the English forum OpenSC have the most posts with 157,106 and 124,993, respectively. From the entire collection, the parsers and queries identified 11,667 source code posts with 28,489 code snippets (a source code post can have multiple code snippets), 2,412 tutorials, and 2,980 attachments. However, it should be noted that the majority of these attachments (2,776 of the total 2,980 attachments) are from OpenSC. Three forums, ExploitIN, Prologic, and Reverse4You, do not allow members to post attachments, while DamageLab, Xakepok, and Xeksec limit the privilege of posting attachments to more senior forum members. Nevertheless, these forums were still analyzed as they contained relatively large amounts of other assets: source code and tutorials.

Asset Analysis and Evaluations

The asset analysis component aims to automatically understand the nature and implementation of the preprocessed source code, attachment, and tutorial posts. First, the source code is classified by coding language to provide insight into their implementation (e.g., what language is used to program formgrabbers?). This classification also facilitates better overall code organization for educational purposes. Ten classes were selected for classification based on observations of the most used code in the forums and in past research: Java, Python, C/C++, HTML, PHP, Delphi, Visual Basic, SQL, Ruby, and Perl [51, 59]. One hundred code files for each language were used to train an SVM classifier (using term frequencies as features) with RapidMiner's LIBSVM package. These files were selected based on their uniqueness and exclusivity to each language (e.g., JSoup only appears in Java), as that helps to increase the effectiveness of the classifier [59]. After training, the SVM classifier was evaluated against other benchmark classifiers using metrics such as accuracy, precision, recall, and F-measure. Paired t-tests are also conducted. Once evaluated, the trained classifier is applied to each forums' source code.

After classification, LDA is run on each forums' code. As with previous literature, all comments and post content are left in to provide extra context during LDA analysis [13]. Including the raw source code along with the comments and post

Table 1. Summary of Research Testbed

Forum	Language	Date range	# of posts	# of threads	# of authors	# of source code snippets	# of code posts	# of tutorials	# of attachments
OpenSC	English	2/6/2005–9/15/2015	124,993	16,046	6,796	13,155	5,478	1,480	2,776
Damagelab	Russian	11/4/2004–9/15/2015	5,903	648	1,007	506	138	14	27
ExploitIN	Russian	2/26/2005–9/15/2015	157,106	16,194	7,761	6,341	3,037	336	N/A
Prologic	Russian	8/26/2006–9/15/2015	25,865	5,092	2,276	734	389	161	N/A
Reverse4you	Russian	8/3/2009–9/15/2015	4,998	758	243	4,063	370	72	N/A
Xakepok	Russian	4/15/2009–9/15/2015	50,337	14,026	3,827	2,009	1,536	202	172
Xeksec	Russian	6/6/2007–9/15/2015	62,316	48,947	18,462	1,681	719	147	5
Total:	English/Russian	11/4/2004– 9/15/2015	431,518	101,711	40,372	28,489	11,667	2,412	2,980

content has shown to significantly increase LDA's performance as compared to just the post content or comments [13]. Attachment and tutorial posts, however, do not need any sort of classification and are descriptive of the file or instructions provided in the post subject field (e.g., BlackPOS attachment and malicious document tutorial from earlier). As such, LDA is run directly on these subsets after preprocessing. For the purposes of this study, we include all assets to gain a comprehensive perspective of the entire forum. This understanding can lead to additional research inquiries that are not apparent by limiting our analysis to a subset of the data.

Regardless of asset, perplexity is calculated for each subset separately at varying epochs (e.g., 5, 10, 15, 20, etc.) to identify the appropriate number of topics to extract for each subset. The model providing the lowest perplexity, thus indicating the highest level of performance, is used. We manually label the topics based on our interpretation of the top 15 keywords that are outputted from the trained model. After extracting the appropriate number of topics from the LDA model based on the perplexity calculations, we asked a panel of six cyber security students to validate our interpretations of the outputted topics. In this task, we provided each of the students with the topic keywords for each topic, our interpretation of each topic (i.e., topic label), and some sample postings that were categorized as falling into the topic. Each panelist was asked to agree or disagree with our interpretation of the topic based on the provided keywords and postings. If the panelist disagreed with our interpretation of the topics, we asked the panelist to provide an alternate suggestion for the topic label based on the keywords and the provided postings. To limit biases, we asked each of the raters to rate the topics independent of other raters. Consistent with standard practice, we calculate the level of concordance between the raters using the Cronbach's alpha statistic. Selected topics for malicious assets are then temporally visualized.

After all analyses, we developed a web portal, the Hacker Asset Portal (HAP), to host selected assets. As mentioned at the beginning of the article, providing these assets for the larger cyber security community can provide a novel approach to enhancing current cyber security education. Students and cyber security professionals alike would gain the ability to learn how the hacker community is developing their malicious tools. Such knowledge would aid in developing and implementing more robust cyber defenses to block potential attacks. Given these benefits, the HAP aims to provide selected users within the cyber security community the ability to browse, search, sort, and download assets. We also create a visualization system allowing users to identify asset trends and key hackers for assets (based on post frequency).

Social Network Construction

After identifying malicious topics, the metadata associated with selected topics' posts are used to build social networks to identify key hackers for those topics. For a specific topic, we extracted all the posts that had the highest probability of belonging to that topic, as calculated by LDA. Using the associated thread and author data for each post, we construct bipartite networks connecting hackers (node type 1) to threads with

specific types of assets (node type 2). Doing so is consistent with prior literature [31, 50]. While directly modeling hacker relationships with asset posts is preferable, such information is limited to forum administrators. Furthermore, this type of bipartite network configuration has been used in various other forum contexts [55]. Given the limited statistical analysis that can be conducted on bipartite graphs, we further project the graph into two monopartite graphs; one representing malicious hacker tools, the other representing hackers. From here, we identify topological properties of each network by running global network statistics such as network diameter, density, connected components, and average path lengths. We gain a more granular understanding of key hackers by calculating degree and betweenness centrality measures.

Results and Discussion

This section first presents SVM classification and LDA evaluation results. Subsequently, interesting results for source code, attachment, and tutorial assets are highlighted. Finally, the social network analysis results identifying key hackers for specific topics is presented.

Classification Performance

In this experiment, we aim to establish a baseline of classification performance based on prevailing classification techniques. We evaluated several state-of-the-art classifiers against SVM: k-nearest neighbor (k-NN), naive Bayes, and decision tree using tenfold cross-validation. In this approach, the training data (i.e., the 1,000 source code files from online repositories) is partitioned into 10 disjoint subsets. The classifier is trained on 9 of these subsets and tested on the remaining subset. This process is repeated until each of the 10 subsets partake in both training and testing. The accuracy, precision, recall, and *F*-measure scores are summarized in Table 2.

Overall, SVM, naive Bayes, and decision tree demonstrated high accuracy, precision, recall, and *F*-measure. Further statistical tests show that SVM significantly outperformed the other classifiers (results of paired *t*-tests are reported in Table 3), even with their strong performances in other metrics. These findings are consistent with other code classification literature [33, 38, 59]. Given SVM's strong statistical performance, it is adopted as the primary classifier for our subsequent analyses.

LDA Evaluations

In addition to evaluating the efficacy of the SVM classifier, we also calculate the perplexity at various epochs (5 to 100, incrementing by 5) to identify the optimal number of topics for which to model each subset of data. Table 4 summarizes the optimal number of topics for each set of data. After extracting the optimal number of topics, our raters independently validated our interpretation of the topics based on the provided topic keywords, our topic labels, and postings within the given topic.

Table 2. Classification Results

Classification algorithm	Overall accuracy	Precision	Recall	*F*-Measure
Support Vector Machine	98.20	96.36	98.20	98.28
k-Nearest Neighbor	64.00	83.47	64.00	72.24
Naive Bayes	86.00	88.57	86.00	87.26
Decision Tree	82.60	86.41	82.60	84.42

Table 3. *P*-values for Pair-wise *t*-Tests for SVM Against Benchmark Classifiers

Metric	SVM vs. *k*-Nearest Neighbor	SVM vs. Naive Bayes	SVM vs. Decision Tree
Precision	<.0001***	<.0001***	.000102**
Recall	<.0001***	<.0001***	<.0001***
F-measure	<.0001***	<.0001***	<.0001***

** *p*-values significant at corrected threshold alpha/n = 0.05
*** *p*-values significant at corrected threshold alpha/n = 0.0001.

Table 4. Optimal Topic Numbers

Data	Optimal topic number	Perplexity
Attachments	100	1,794.184
Tutorials	90	2,150.418
DamageLab	60	440.772
ExploitIN	65	1,424.834
OpenSC	95	4,866.838
Prologic	95	970.041
Reverse4You	80	1,576.980
Xakepok	90	390.453
Xeksec	80	1,198.133

Interrater reliability calculations demonstrated that our raters reached a 0.9393 Cronbach's alpha score, indicating a high level of concordance between all raters.

Malicious and Emerging Source Code Topics

Source code assets are particularly valuable to hackers as they can be easily adjusted or extended to incorporate new functionalities, given the hacker's technical ability to do so. Our code analysis reveals each forums' malicious code, their implementations (i.e., language), and temporal trends. For illustration purposes, we present results of

the three of the largest forums in our collection (in terms of code assets), OpenSC, ExploitIN, and Reverse4you as case examples. OpenSC and ExploitIN are also two of the longest-running forums in our collection, both forums having started in 2005.

Of the topics in OpenSC, 9.45 percent are considered malicious, while the remaining 90.55 percent were benign. While still interpretable LDA results, these benign topics were often general and did not pertain to exploitations or malware. For example, some of the benign topics include general Java or Python programming. Despite the large quantity of such assets, OpenSC members can still freely access a great variety of malicious code, including, for example, low-level system exploits such as shellcode, memory, and process injections, all designed to harm a computer's memory and system processes; application exploits such as crypters, designed to encrypt user files, and keyloggers, built to steal sensitive user data; and web attacks such as backdooring websites and SQL injections, intended to exploit webpage vulnerabilities. The malicious source codes identified by our classifier are listed in Table 5. Our classifier further revealed that these exploits are written in languages

Table 5. Malicious Source Code Topics in OpenSC Forum

Source code topic label	Primary language of implementation	% of topics in OpenSC	Keywords in topic(s)
Crypters	Java	2.10	Decrypt, encrypt, encrypted, key, generate, algorithm, keys, polymorphism
Metasploit Exploits	Ruby	1.05	Metasploit, framework, applications, install, exploit, systems, advanced, analysis, compile
Shellcode Exploits	C/C++	1.05	Shellcode, x30, x20, x65, x0a, x6f, x6e, x40 (*this is sample shellcode that appeared in topic*)
Backdooring Websites	PHP	1.05	Backdoored, http, com, html, source, org, php, index, htm, dump, log, script
Memory Injections	Delphi	1.05	Getprocaddress, process, hprocess, mem, kernel32, pmem, inject, pointer, injectstring
SQL Injections	SQL	1.05	Php, http, post, url, get, mysql, login, upload, password, select, sql
Process Injections	Delphi	1.05	Process, pid, thread, injection, processed, detach, target, library, list, ddllmain
Keylogging	C/C++	1.05	Keylogger, password, http, rar, firefox, upload, users, stealer, subject
Total	—	**9.45**	—

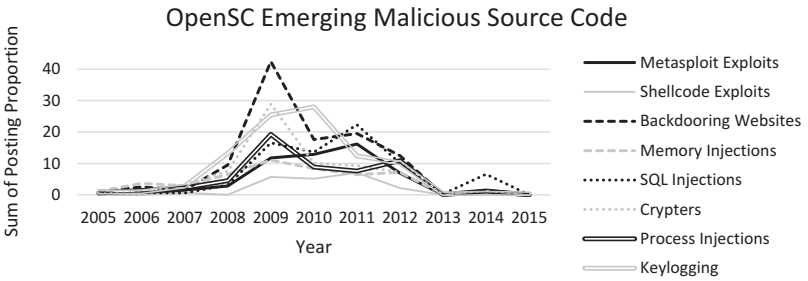

Figure 3. Emerging Malicious Source Code Topics for OpenSC

ideal for the system or target for which they are designed. For example, memory injections, shellcode exploits, and process injections were primarily written in Delphi or C/C++, Metasploit modules in Ruby, SQL injections in SQL, and for website backdooring, PHP. In addition to classifying malicious source code topics, the forum time-stamp metadata were leveraged to create a line chart plotting the proportion of posts related to a particular topic for each year in Figure 3.

Figure 3 shows that many of the malicious source code topics were popular between 2009 and 2011. Some of the peaks shown on the chart, such as backdooring websites and keylogging, are consistent with well-known events. In 2009, for example, the popular website builder Wordpress received widespread attention for being vulnerable to backdoors [40]. Additionally, there was a surge in real key-logging exploitations between 2008 and 2010 [23]. More recently in the forum, crypters have received slightly more attention, possibly because they are the foundation for the robust cyber weapon Ransomware (a tool that encrypts a user's files as ransom for payment), one of the most malicious tools known to date [57]. Such an increase in forum assets correlates and is validated with the growing amount of actual exploits conducted with crypters and Ransomware during the same time period [57]. The chart also shows a decrease in popularity for all malicious codes over time. The likely explanation for this decrease is that OpenSC has seen a decline in uptime in recent years. If the forum was not online, its users would have been unable to post forum content.

ExploitIN also contains a number of network, web, and application exploits for its members to freely access. Interestingly, SQL injections and shellcode exploits in ExploitIN were both implemented in the same languages as the SQL injections and shellcode exploits found in OpenSC. In addition to these exploits, ExploitIN also contains network binders, spam services, password-cracking tools, and banking rootkits for their members to access. Table 6 summarizes these topics.

In analyzing the temporal trends for ExploitIN assets in Figure 4, we discover several interesting findings. First, we identified more recent, emerging trends as compared to OpenSC, specifically for network binders, password crackers, spam services, crypters, and shellcode exploits. Additionally, we discovered that some of the trends present in OpenSC (e.g., SQL injection popularity in 2009–2011) were also in ExploitIN. This

Table 6. Malicious Source Code Topics in ExploitIN Forum

Source code topic label	Primary language of implementation	% of topics in ExploitIN	Keywords in topic(s)
Shellcode Exploits	C/C++	6.00	Buffer, int, shellcode, overflow, argv, stack, xff, x40, exploit, vulnerable
SQL Injections	SQL	3.00	Select, sql, table, error, database, userse, injection, query, null, union, request
Network Binders	C/C++	1.50	Bind, router, network, utility, information, interface, scanner, command, tool, cisco
Password Cracking	Python	1.50	Password, login, proxy, user, pass, username, get, type, log, auth, set
Spam Services	Java	1.50	Spam, optimization, site, traffic, engines, website, url, page, registration, visitors
Banking Rootkits	Java	1.50	Rootkit, bank, malicious, security, network, malware, banks, infected, tools
Crypters	Java	1.50	Crypt, decrypt, encrypt, encrypted, file, install, kriptor, keys
Total	—	**16.50**	—

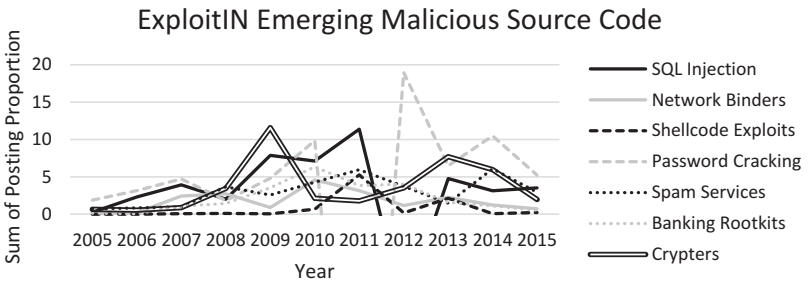

Figure 4. Emerging Malicious Source Code Topics in ExploitIN

shows that the growth in popularity for a specific asset during a given time period may not be limited to a single forum, but may be a cross-forum phenomena. This is further illustrated with crypters showing a recent increase in popularity (2014–2015), coinciding with both the growth in OpenSC and recent Ransomware events.

Another forum, Reverse4You, contains assets not seen in either OpenSC or ExploitIN. For example, many bruteforcers, or tools created to deliberately attempt to break through login systems by enumerating through sets of usernames and passwords were available for access. In addition, there were also a variety of dynamic-link library (DLL) exploits available. Such exploits are targeted directly

Table 7. Malicious Source Code Topics in Reverse4You Forum

Source code topic label	Primary language of implementation	% of topics in Reverse4You	Keywords in topic(s)
Shellcode Exploits	C/C++	10.0	Stack, pop, payload, shellcode, x66, x6a, exploit, shell, execute, system, func
Rootkits	Java	2.50	Rootkit, machine, virus, procedure, irp, tcp, stack, contents, flow
Memory Overflows	C/C++	2.50	Overflow, crash, memory buffer, import, print, violation, access, software, pydbg
DLL Exploits	C/C++	1.25	Figure, dll, module, analysis, memory, ldrloadll, calls, notice, breakpoints
Network Binding	C/C++	1.25	Socket, return, port, error, case, int, sockaddr, tcp, http, protocol, buffer, bind
Bruteforcing	PHP	1.25	Brute, force, website, password, form, access, user, php, pass
Total	**—**	**18.75**	**—**

at critical code libraries for the Windows operating system. Table 7 summarizes the malicious assets available in Reverse4You.

Similar to ExploitIN, some exploits in Reverse4You have shown recent popularity. Shellcode exploits and bruteforcers in particular have enjoyed a recent growth. While shellcode exploits, or exploits targeted at gaining root access over a target system, have been a consistently used tool over a long period of time, bruteforcing technology has garnered much recent interest. Such technology is often needed in environments in which hackers need to try a multitude of username and password combinations. A recent example in which such a tool proved to be valuable can be seen with the FBI's purchase of a bruteforcing tool from hackers to help them unlock an iPhone related to the tragic events of the San Bernardino shooting [15]. Such an example illustrates how tools acquired from hackers can prove to be valuable in a variety of contexts. Figure 5 highlights all the malicious trends of assets in Reverse4You.

Malicious and Emerging Attachment Topics

Although 80 percent of the topics modeled pertained to general topics, a variety of malevolent attachments was discovered. Table 8 presents some of the identified malicious topics.

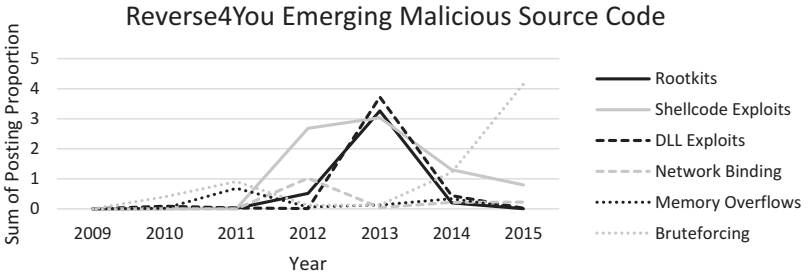

Figure 5. Emerging Malicious Source Code Topics for Reverse4You

Table 8. Malicious Attachment Topics

Attachment topics label	% of topics	Keywords in topic(s)
DarkComet Remote Administration Tool (RAT)	4.00	Remote, rat, darkcomet, website, title, users, capture, main, webcam, desktop, update, contain
Keylogging	4.00	Keylogger, lttllogger, log, dll, injection, logfile, version, bypass, inject, automatic, key, folder
Password Cracking	3.00	Password, check, rar, fake, program, download, victim, keygen, data, site, sniffer, software
Crypters	3.00	Rar, crypter, coded, crypt, cryptor, decrypter, undetector, Trojan, polymorphic
Botnet/SYN Flood	3.00	Attack, flood, icmp, udp, tcp, ddos, syn, attacker, attack, botnet, ping, control, execute
Password Stealers and Loggers	2.00	Steal, version, upload, log, pass, password, have, key, username, site, logger, remove, secure, decrypt
Rootkits	2.00	Hide, service, rootkit, windows, explorer, name, port, reg, space, start, registry, process, backdoor
Windows Process Injections	2.00	Process, into, injection, pid, kill, windows, inject, list, running, dll, loaded, computer, close, system
Visual Studio Exploits	1.00	Program, visual, Microsoft, include, studio, error, warning, ddos, buffer, flood, bind, shutdown, send
Browser Exploits	1.00	Browser, database, default, address, exploit, exec, binary, install, checks, php, internet, websites
Binders	1.00	Sub, binder, detect, build, download, pack, execution, bind, version, unpack, subs, extraction
Formgrabbing	1.00	Data, system, application, form, content, post, username, account, script
Total	**27**	—

Some of the malicious topics discovered were related to DarkComet remote administration tools (RATs), comprising 4 percent of the topics. Keylogging makes up 3 percent of the topics, and password-cracking tools about 3 percent of

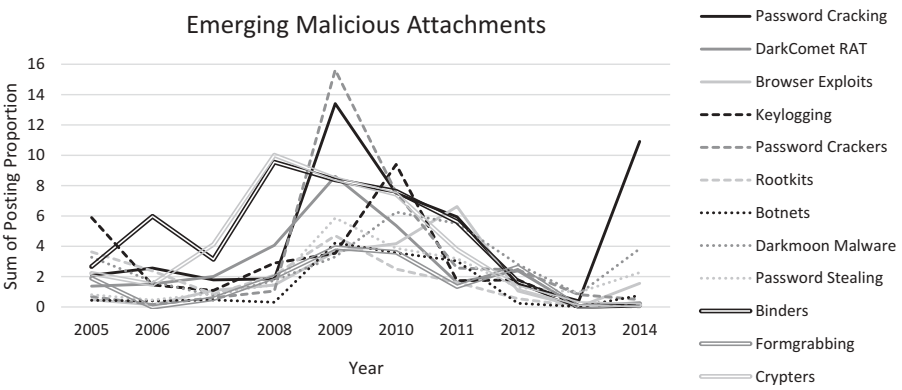

Figure 6. Emerging Malicious Attachment Topics

the topics. The DarkComet RAT is a popular remote administration tool that allows an attacker to fully and remotely control a machine (e.g., power control, locking, file system access, etc.) and spy on a victim (e.g., keylogging, webcam captures, etc.). This kind of malware has been responsible for many of the violent escalations in the Syrian conflict in 2012 [39]. Another popular topic, keylogging, while not as directly devastating as the DarkComet RAT, still has the potential to cause great damage. This software allows attackers to monitor victims' keystrokes and can potentially record sensitive information such as passwords, bank account information, and social security numbers. This information can then be used for other serious crimes such as identity theft or financial fraud.

Figure 6 illustrates that some emerging malicious topics are consistent with recent attacks. For example, Microsoft ended its support for Windows XP in early 2014. Many attacks against this platform used DarkMoon Malware [57], one of the emerging topics. Password-cracking tools show a significant spike in 2015. Weak and cracked passwords are often the easiest way to exploit a system [28]. These types of tools may have been at the core of recent password exploitations against the Office of Personnel Management (OPM). Regardless of asset, 2012 marked the beginning of a significant downward trend in popularity or access appears for most attachments. Significant downtime in the forum OpenSC, the forum where the majority of the attachments come from, could be a cause for the decrease in attachments.

Malicious and Emerging Tutorial Topics

The majority of tutorial topics (90 percent) were benign, with a purpose of educating others on how to perform basic computing tasks. However, sets of malicious tutorials are still freely available for forum members to access, summarized in Table 9.

Table 9. Malicious Tutorial Topics

Tutorial topics label	% of topics	Keywords in topic(s)
DDoS	2.22	Ddos, attacks, network, attack, security, malicious, bot, infected, large, traffic
Metasploit Tutorials	2.22	Msfconsole, metasploit, tutorial, payload, exploit, shell, reversing, exploitation, vulnerabilities, tools
Shellcode Injections	2.22	Shell, shellcode, memory, dll, exe, downloader, executable, injection, inject, remote
Carding Tutorials	1.11	Sell, money, banks, bank, card, email, transfer, number, payment, paypal, info, account
Password Stealing	1.11	Password, program, user, email, information, steal, account, victim, send, files
Crypter Tutorials	1.11	Cryter, virus, scanner, stealer, binder, crypt, zip, istealer, tool, beta, generator
SQL Injections	1.11	Page, php, sql, injection, html, action, request, cookie, error, select, web
Total	**11.10**	—

Overall, several types of malicious tutorials found are particularly interesting. First, carding tutorials educate hackers on illegal credit card manipulation and fraud. This type of crime is growing in notoriety in underground economies and is an emerging area of interest for both practitioners and researchers alike [32]. Second, crypter tutorials show how to develop tools that encrypt files, as well as how to build ransomware. Finally, SQL injection tutorials develop a hackers' expertise in taking advantage of improperly coded websites to gain access to their underlying databases. Many popular retailers today are consistently hit with SQL injection attacks [24]. Figure 7 visually illustrates the trends of these malicious topics.

Almost all the malicious tutorials have leveled out in popularity in recent years (right side of Figure 7). Among the discovered tutorials, some are consistent with major events. For example, Metasploit, a popular penetration testing framework, transitioned from Perl to Ruby from 2008 to 2011. During this time, Metasploit tutorials in forums grew, possibly due to hackers wanting knowledge about the new framework. Additionally, consistent with OpenSC and ExploitIN source code, SQL injection topics increased between 2009 and 2011, a point in time when there were many exploitations against websites' databases [45].

Hacker Asset Portal

As mentioned in the research design, the development of the HAP can offer significant value to the larger cyber security community by allowing users to search, sort, browse, and download selected assets. For the first version of our portal, we loaded 15,576 source code snippets, 2,980 attachments, and 987 tutorials. Figure 8

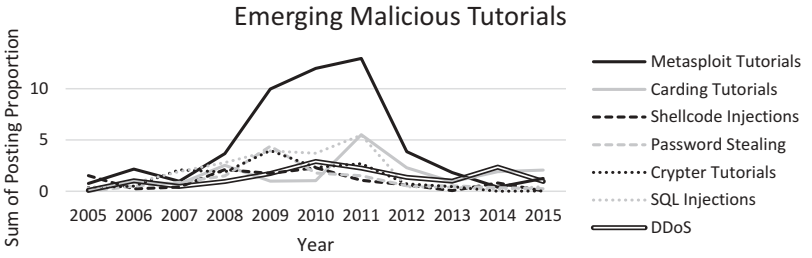

Figure 7. Emerging Malicious Tutorial Topics

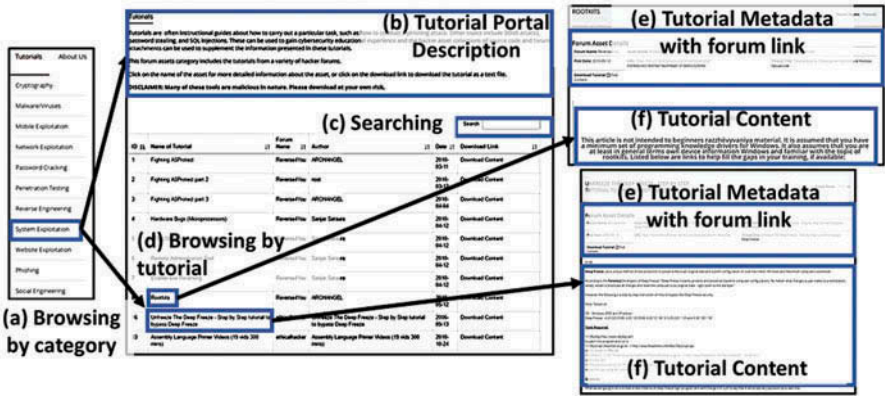

Figure 8. AZSecure Hacker Assets Portal System Interface. (a) Browsing tutorials by category, (b) description of tutorial portal, (c) searching for specific assets, (d) browsing by tutorial, (e) metadata associated with each tutorial and (f) tutorial content

illustrates the basic interface design for the HAP. For the purposes of illustration, we provide an example of how users can browse various categories of tutorials (identified from our LDA analysis), search and sort based on various post metadata (e.g., post date, author name, etc.) and access the raw tutorial. Tutorials are particularly valuable for cyber security education and CTI purposes as it allows users to understand exactly how hackers are executing their malicious tasks.

In addition to the interface, we also created CTI dashboards (Figure 9) allowing users to identify trends of malicious assets and key threat actors for those assets (based on post frequency). Built in Tableau, the dashboard dynamically updates based on users' selections of specific time points or hacker names. Assuming an organization knows its systems, the dashboard can provide a visual representation of asset trends and key hackers to inform future cyber defenses. Given that CTI becomes more valuable with newer data, organizations interested in using this framework are advised to collect forums on a monthly or bimonthly basis to ensure freshness of data.

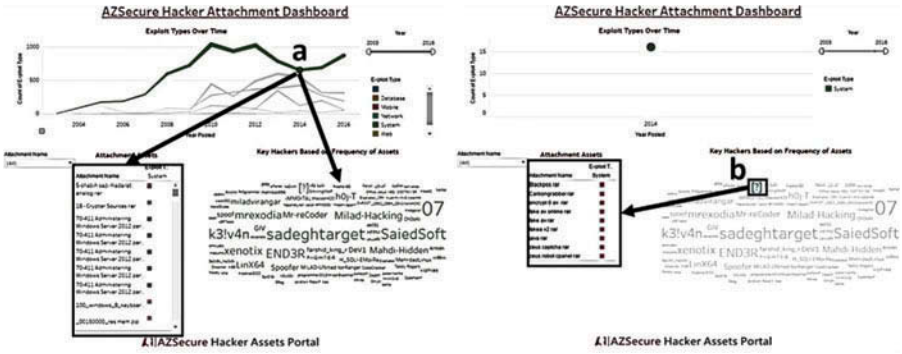

Figure 9. AZSecure Hacker Assets Portal Dashboard Interface. (a) Filtering on 2014, when BlackPOS was posted, shows assets and threat actors at that time, (b) Filtering the actor who posted BlackPOS reveals that he posts other bank exploits (e.g., Zeus).

Social Network Analysis

The final part of our framework creates a bipartite social network between hackers and threads with source code assets for specific topics extracted by LDA. For illustration purposes, we create one bipartite network for crypter source code assets found in OpenSC. We represent the social network of crypter source code assets given their popularity in OpenSC and their recent notoriety for providing key technology for ransomware. We also project the bipartite network into a monopartite network of hackers to calculate additional network measures. Figure 10 depicts both networks. A variety of topological and node-level metrics (summarized in Table 10)

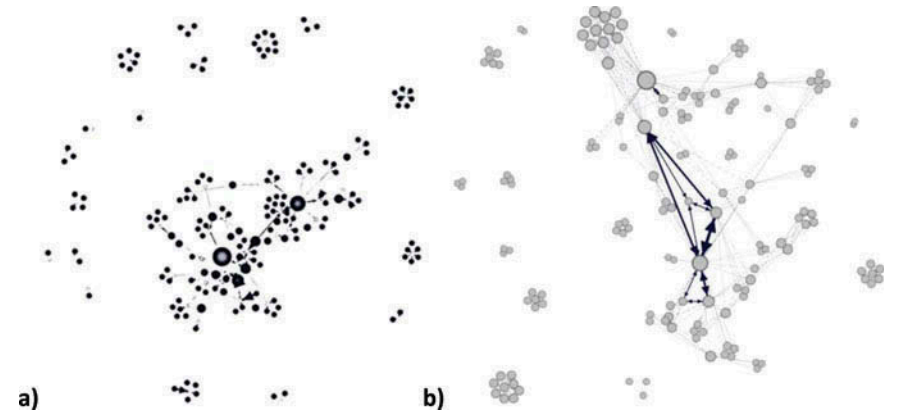

Figure 10. Bipartite and Monopartite Projections of Crypter Social Network. (a) Original bipartite network of hackers connecting to threads containing crypter source code assets. Black nodes represent the hackers, white nodes represent the thread with a crypter asset. Size of the nodes represents degree centrality. (b) Monopartite projection of the hackers in the crypter network. Size of the nodes represents degree centrality.

Table 10. Topological and Node Level Metrics for Crypter Bipartite and Monopartite Hacker Networks

Metric category	Metric	Bipartite network	Monopartite hacker network
Topological Metrics	Number of Nodes	207	156
	Number of Edges	207	938
	Network Diameter	1	6
	Graph density	0.005	0.039
	Connected components	18	14
	Size of giant component	159 (76.911%)	104 (66.67%)
	Average path length	1	3.106
Node Level Metrics	Minimum Degree	1	1
	Maximum Degree	14	68
	Average Degree	2	12.026

are calculated to better understand the specific characteristics of the networks. Given that our goal is to identify key hackers for crypter source code assets, we focus the majority of our discussion on the monopartite hacker network.

Overall, there are 207 nodes in the bipartite network and 156 nodes in the monopartite hacker network. In the hacker network, we can see that the network diameter is 6, indicating that the network is relatively compact and that each hacker has to take a minimal number of steps to reach another hacker in the network. This may be due to the fact that there are 14 connected components, with the size of the giant component comprising of 104 nodes, or 66.67 percent of the overall network. The giant component also contributes to a relatively low average path length of 3.106. However, even with the small network diameter and the low average path length, there is a small graph density (0.039), indicating that many of the hackers do not take advantage of the knowledge from all the other hackers in the community, but instead from a select few. Such a discovery is consistent with prior literature [25], and is also represented with the large disparity in the degree distribution of the network. The majority of the hackers (103) have a degree less than the average of 12.026. Only a select few hackers have a high degree within the network. To better understand these hackers, we summarize the top 10 hackers for crypters based on their degree and betweenness centrality in Table 11, both of which are strong indicators of key members within a network [19]. We also detail each hacker's role in the forum, when they joined the forum, and the total number of posts they have made overall.

Several key insights can be taken from Table 11. First, the majority of the hackers in the top ten ranking are senior members within their community, have been part of the community for a long period of time, and contribute a large number of forum posts. Such a reputation may help to drive the standing of these hackers as it pertains

Table 11. Ranking and Forum Status for Key Hackers Based on Degree and Betweenness Centrality

	Degree centrality		Betweenness centrality		Forum status		
Hacker	Rank	Value	Rank	Value	Forum role	Join date	Total # of forum posts
KriPpLer	1	68	1	4,725.667	Admin	04/16/2006	2,008
mjrod5	2	54	2	4,362.433	Senior Member	08/31/2008	3,053
counter606	3	42	3	3,734.667	Member	03/14/2008	72
ItalianFamily	4	32	10	94.167	Senior Member	12/03/2005	151
counterstrikewi	5	30	4	1,035.200	Senior Member	04/13/2009	1,913
cracksman	6	30	8	578.667	Senior Member	12/20/2006	1,684
SqUeEzEr	7	24	6	885.333	Senior Member	06/02/2008	1,535
Retired boss	8	24	7	706.267	Retired	03/07/2005	1,474
slayer616	9	20	5	942.000	Senior Member	12/05/2007	1,440
deex	10	20	9	470.000	Junior Member	01/01/2011	15

to the adoption and dissemination of crypter assets within the forum. Additionally, the total number of forum posts indicates that these hackers are active in the community as a whole, and not just within the crypter code network.

Conclusion and Future Directions

Although cybersecurity is a growing societal concern, much of the traditional cyber threat intelligence focuses on analyzing cyber assets after they have already infected or compromised a system. Despite numerous technical challenges, there has been a push in recent years from practitioners and the information systems community alike to develop more proactive CTI. One way to help accomplish this is by understanding potential threats directly from hacker communities. This study contributes a novel framework to CTI by leveraging an automated and principled web, data, and text mining approach to collect and analyze vast amounts of hacker source code, tutorials, and attachments directly from large, international underground hacker communities. The framework allows us to identify many freely available, malicious assets in underground hacker forums such as crypters, keyloggers, SQL Injections, and password crackers, some of which may have been the root cause of recent breaches against organizations like the OPM. We are also able to determine the key individuals behind these assets by using social network analysis techniques and metrics. Our approach is generalizable to any hacker forum, irrespective of subforum structure.

One of the hallmarks of design research in information systems is the development of artifacts (e.g., systems) for practical utility [42, 43, 44, 46, 47]. This research has practical implications for organizations aiming to improve their cyber security posture. Assuming an organization knows the systems it wishes to protect, it can apply this framework to forums of its choosing to identify relevant hacker assets for their systems. Future work can expand this research in several directions. First, specific types of malware can be examined to identify the manner in which they disseminate and evolve over time, both between and within forums. Second, additional analytical techniques such as sentiment analysis can be leveraged to supplement the social network analysis to better identify key hackers. Finally, novel machine learning techniques can be developed to identify key features of specific types of assets (e.g., Zeus code) to identify its key features and potentially predict how that malicious asset will evolve. All these areas have the potential to further increase proactive cyber threat intelligence capabilities and prevent future cyber attacks.

Acknowledgments: This material is based on work supported in part by the National Science Foundation (DUE-1303362 and SES-1314631).

References

1. Abbasi, A.; Li, W.; Benjamin, V.; Hu, S.; and Chen, H. Descriptive analytics: Investigating expert cybercriminals in web forums. In *Proceedings of the IEEE Joint Intelligence and Security Informatics Conference*. The Hague, The Netherlands: IEEE, 2014, pp. 55–63.
2. Abbasi, A.; Zahedi, F.; Zeng, D.; Chen, Y.; Chen, H.; and Nunamaker, J.F. Enhancing predictive analytics for anti-phishing by exploiting website genre information. *Journal of Management Information Systems*, *31*, 4 (2015), 109–157.
3. Ablon, L.; Libicki, M.C.; and Golay, A.A. Markets for cybercrime tools and stolen data: Hackers' bazaar. Rand Corporation, 2014.
4. Allamanis, M., and Sutton, C. Why, when, and what: Analyzing stack overflow questions by topic, type, and code. In *Proceedings of the 10th Working Conference on Mining Software Repositorie*s. San Francisco, CA: ACM, 2013, pp. 53–56.
5. Bajaj, K.; Pattabiraman, K.; and Mesbah, A. Mining questions asked by web developers. In *Proceedings of the 11th Working Conference on Mining Software Repositories*. 112-121. Hyderabad, India: ACM, 2014, pp. 112–121.
6. Baldi, P.F.; Lopes, C.V.; Linstead, E.J.; and Bajracharya, S.K. A theory of aspects as latent topics. *ACM Sigplan Notices*, *43*, 10 (2008), 543–562.
7. Barua, A.; Thomas, S.W.; and Hassan, A.E. What are developers talking about? An analysis of topics and trends in stack overflow. *Empirical Software Engineering*, *19*, 3 (2014), 619–654.
8. Benjamin, V., and Chen, H. Securing cyberspace: Identifying key actors in cybercriminal communities. In *Proceedings of the IEEE Joint Intelligence and Security Informatics Conference*. Washington, DC: IEEE, 2012, pp. 24–29.
9. Benjamin, V.; Zhang, B.; Nunamaker, J.F.; and Chen, H. Examining hacker participation length in cybercriminal Internet-relay-chat communities. *Journal of Management Information Systems*, *33*, 2 (2016), 482–510.
10. Benjamin, V.; Li, W.; Holt, T.; and Chen, H. Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops. In *IEEE International Conference on Intelligence and Security Informatics*. Baltimore, MD: IEEE, 2015, pp. 85–90.
11. Blei, D.M.; Ng, A.Y.; and Jordan, M.I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, (2003), 993–1022.
12. Chen, H.; Chiang, R.H.; and Storey, V.C. Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, *36*, 4 (2012), 1165–1188.
13. Chen, T.H.; Thomas, S.W.; and Hassan, A.E. A survey on the use of topic models when mining software repositories. *Empirical Software Engineering, 21,* 5 (2015), 1–77.
14. Chu, B.; Holt, T.; and Ahn, G. *Examining the Creation, Distribution, and Function of Malware On-Line*. Washington, DC: U.S. Department of Justice, National Criminal Justice Reference Service, 2010.
15. Constantin, L. FBI bought exploit from hackers to access San Bernardino iPhone. *Computerworld*, 2016. Available at www.computerworld.com/article/3055486/security/fbi-bought-exploit-from-hackers-to-access-san-bernardino-iphone.html (accessed on May 17, 2016)
16. Elkind, P. Sony Pictures: Inside the hack of the century. *Fortune*, 2015. Available at http://fortune.com/sony-hack-part-1/. (accessed on November 23, 2015)
17. EY, Global Advisory Services. Cyber threat intelligence: How to get ahead of cybercrime. From the series: *Insights on Governance, Risk, and Compliance. Ernst and Young*, 2014. Available at www.ey.com/Publication/vwLUAssets/EY-cyber-threat-intelligence-how-to-get-ahead-of-cybercrime/$FILE/EY-cyber-threat-intelligence-how-to-get-ahead-of-cybercrime.pdf (accessed on March 17, 2016)
18. Farnham, G. Tools and standards for cyber threat intelligence projects. *SANS Institute*, 2013. Available at www.sans.org/reading-room/whitepapers/warfare/tools-standards-cyber-threat-intelligence-projects-34375 (accessed on March 18, 2016)
19. Faust, K. Centrality in affiliation networks. *Social Networks*, *19*, 2 (1997), 157–191.
20. Goel, S. Cyberwarfare: Connecting the dots in cyber intelligence. *Communications of the ACM*, *54*, 8 (2011), 132–140.

21. Grant, S.; Cordy, J.R.; and Skillicorn, D.B. Reverse engineering co-maintenance relationships using conceptual analysis of source code. In *18th Working Conference on Reverse Engineering*. Limerick, Ireland: ACM, 2011, pp. 87–91.

22. Granville, K. 9 Recent cyberattacks against big businesses. *New York Times*, February 5, 2015. Available at www.nytimes.com/interactive/2015/02/05/technology/recent-cyberattacks.html (accessed on March 15, 2016)

23. HelpNetSecurity.com. *January* 2009 *Threatscape: Keylogging and spam problems, surge in exploit activity. HelpNetSecurity.com*, February 9, 2009. Available at www.helpnetsecurity.com/2009/02/09/january-2009-threatscape-keylogging-and-spam-problems-surge-in-exploit-activity/ (accessed on November 23, 2015)

24. Higgins, K. SQL injection attacks haunt retailers. *InformationWeek Dark Reading*, 2014. Available at www.darkreading.com/sql-injection-attacks-haunt-retailers/d/d-id/1269576 (accessed on November 6, 2015)

25. Holt, T.J.; Strumsky, D.; Smirnova, O.; and Kilger, M. Examining the social networks of malware writers and cybercriminals. *International Journal of Cyber Criminology*, *61*, 1 (2012), 891–903.

26. Holt, T.J. Examining the forces shaping cybercrime markets online. *Social Science Computer Review*, *31*, 2 (2013), 165–177.

27. Keegan, B.; Gergle, D.; and Contractor, N. Do editors or articles drive collaboration? Multilevel statistical network analysis of Wikipedia coauthorship. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. Seattle, WA: ACM, 2012, pp. 427–436.

28. Kennedy, D.; O'Gorman, J.; Kearns, D.; and Aharoni, M. *Metasploit: The Penetration Tester's Guide*. San Francisco, CA: No Starch Press, 2011.

29. Kitten, T. Target malware: Exploring the origins. *Bank Info Security*, 2014. Available at www.bankinfosecurity.com/interviews/intelcrawler-i-2161 (accessed on December 1, 2014)

30. Levine, M., and Date, J. 22 million affected by OPM hack, officials say. *ABC*, 2015. Available at http://abcnews.go.com/US/exclusive-25-million-affected-opm-hack-sources/story?id=32332731 (accessed on November 23, 2015)

31. L'huillier, G.; Ríos, S.A.; Alvarez, H.; and Aguilera, F. Topic-based social network analysis for virtual communities of interests in the dark web. In *ACM SIGKDD Workshop on Intelligence and Security Informatics*. Beijing, China: ACM, 2010, p. 9.

32. Li, W.; Chen, H.; and Nunamaker, J.F. Identifying and profiling key sellers in cyber carding community: AZSecure text mining system. *Journal of Management Information Systems*, *33*, 4 (2016), 1059–1086.

33. Linares-Vásquez, M.; McMillan, C.; Poshyvanyk, D.; and Grechanik, M. On using machine learning to automatically classify software applications into domain categories. *Empirical Software Engineering*, *19*, 3 (2014), 582–618.

34. Linares-Vásquez, M.; McMillan, C.; Poshyvanyk, D.; and Grechanik, M. On using machine learning to automatically classify software applications into domain categories. *Empirical Software Engineering*, *19*, 3 (2014), 582–618.

35. Linstead, E.; Lopes, C.; and Baldi, P. An application of latent Dirichlet allocation to analyzing software evolution. In *Seventh International Conference on Machine Learning and Applications*. San Diego, CA: IEEE, 2008, pp. 813–818.

36. Mahmood, A.M.; Siponen, M.; Straub, D.; Rao, H.R.; and Raghu, T.S. Moving toward black hat research in information systems security: An editorial introduction to the special issue. *MIS Quarterly*, *34* 3 (2010), 431–433.

37. Maskeri, G.; Sarkar, S.; and Heafield, K. Mining business topics in source code using latent Dirichlet allocation. In *Proceedings of the 1st India Software Engineering Conference*. Hyderabad, India: ACM, 2008, pp. 113–120.

38. McMillan, C.; Linares-Vasquez, M.; Poshyvanyk, D.; and Grechanik, M. Categorizing software applications for maintenance. In *27th IEEE International Conference on Software Maintenance*. Williamsburg, VA: IEEE, 2011, pp. 343–352.

39. McMillan, R. How the boy next door accidentally built a Syrian spy tool. *Wired*, February 2012. Available at www.wired.com/2012/07/dark-comet-syrian-spy-tool/ (accessed on November 23, 2015)

40. Motoyama, M.; McCoy, D.; Levchenko, K.; Savage, S.; and Voelker, G. M. An analysis of underground forums. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference*. Berlin, Germany: ACM,, 2011, pp. 71–80.

41. National Science and Technology Council (NSTC). Trustworthy cyberspace: Strategic plan for the Federal Cybersecurity Research and Development Program. Report of the National Science and Tehcnology Council, Executive Office of the President, 2011, pp. 1–19.

42. Nunamaker Jr., J.F.; Chen, M.; and Purdin, T.D.M. Systems development in information systems research. *Journal of Management Information Systems*, *7*, 3 (1990), 89–106.

43. Nunamaker Jr., J.F.; Briggs, R.; Derrick, D.; and Schwabe, G. The last research mile: Achieving both rigor and relevance in information systems research. *Journal of Management Information Systems*, *32*, 3 (2015), 10–47.

44. Nunamaker J.F.; Twyman, N., Giboney, J.; and Briggs, R. Creating high-value real-world impact through systematic programs of research. *MIS Quarterly*, *41*, 2 (2017), 335–351.

45. Otto on WordPress. How to find a backdoor in a hacked WordPress, 2009. Available at http://ottopress.com/2009/hacked-wordpress-backdoors/ (accessed on November 23, 2015)

46. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; and Chatterjee, S. A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*, 3 (2007), 45–77.

47. Prat, N.; Comyn-Wattiau, N.; and Akoka, J. A taxonomy of evaluation methods for information systems artifacts. *Journal of Management Information Systems*, *32*, 3 (2015), 229–267.

48. Radianti, J. A study of a social behavior inside the online black markets. In *Proceedings of the International Conference on Emerging Security Information, Systems and Technologies*. Nice, France: IEEE, 2010, pp. 88–92.

49. Riley, M.; Elgin, B.; Lawrence, D.; and Matlack, C. Missed alarms and 40 million stolen credit card numbers: How Target blew it. *Bloomberg*, March 13, 2014. Available atwww.bloomberg.com/news/articles/2014-03-13/target-missed-warnings-in-epic-hack-of-credit-card-data (accessed on November 23, 2015)

50. Rios, S.; Aguilera, F.; Bustos, F.; Omitola, T.; and Shadbolt, N. Leveraging social network analysis with topic models and the semantic web. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. 339-342. Lyon, France: IEEE, 2011, pp. 339–342.

51. Samtani, S.; Chinn, R.; and Chen, H. Exploring hacker assets in underground forums. In *IEEE International Conference on Intelligence and Security Informatics*. Baltimore, MD: IEEE, 2015, pp. 31–36.

52. Sandle, P., and Char, P. Cyber crime costs global economy $445 billion a year: Report. *Reuters*, June 9, 2014. Available at www.reuters.com/article/2014/06/09/us-cybersecurity-mcafee-csis idUSKBN0EK0SV20140609 (accessed on XXX)

53. Savage, T.; Dit, B.; Gethers, M.; and Poshyvanyk, D. Topic XP: Exploring topics in source code using latent Dirichlet allocation. In *IEEE International Conference on Software Maintenance*. Timişoara, Romania: IEEE, 2010, pp. 1–6.

54. Shackleford, D. Who's using cyberthreat intelligence and how? *SANS Institute*, 2015. Available at www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767. (accessed on March 18, 2016)

55. Stewart, S.A., and Abidi, S.S.R. Applying social network analysis to understand the knowledge sharing behaviour of practitioners in a clinical online discussion forum. *Journal of Medical Internet Research*, *14*, 6 (2012), e170.

56. Stuart, K. Lizard Squad is back: Group attacks Xbox Live and Daybreak Games. *Guardian*, February 16, 2015. Available at www.theguardian.com/technology/2015/feb/16/lizard-squad-attacks-xbox-live-daybreak-games (accessed on November 23, 2015)

57. Symantec Corporation. *Internet Security Threat Report*, 2014.

58. Tian, K.; Revelle, M.; and Poshyvanyk, D. Using latent Dirichlet allocation for automatic categorization of software. In *6th IEEE International Working Conference on Mining Software Repositories*. Vancouver, Canada: IEEE,, 2009, pp. 163–166.

59. Ugurel, S.; Krovetz, R.; and Giles, C.L. What's the code? Automatic classification of source code archives. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, July 23–25, 2002, pp. 632–638.

60. Vijayan, J. New details of Home Depot attack reminiscent of Target's breach. *InformationWeek Dark Reading*, 2014. Available atwww.darkreading.com/attacks- breaches/new-details-of-home-depot-attack-reminiscent-of-targets-breach/d/d-id/1317323 (accessed on November 23, 2015)

61. Wang, T.; Wang, H.; Yin, G.; Ling, C.X.; Li, X.; and Zou, P. Mining software profile across multiple repositories for hierarchical categorization. In *29th IEEE International Conference on Software Maintenance*. Eindhoven The Netherlands, 2013, pp. 240–249.

62. Yip, M. An investigation into Chinese cybercrime and the applicability of social network analysis. In *ACM Web Science Conference*. Koblenz, Germany: IEEE, 2011, pp. 1–4.