AFRL-AFOSR-VA-TR-2019-0225

Automated, Binary Evidence-based Attribution of Software Attacks

Kevin Hamlen
UNIVERSITY OF TEXAS AT DALLAS

07/31/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
30-07-2019

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
07/01/2014 - 06/30/2018

4. TITLE AND SUBTITLE
Automated, Binary Evidence-based Attribution of Software Attacks

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-14-1-0173

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kevin W. Hamlen, Zhiqiang Lin, Latifur Khan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Public

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This academic research project investigated and developed software attack attribution technologies and algorithms with the goal of facilitating automatic inference of possible origins and authorships of cyberattacks (e.g., malware attacks).

15. SUBJECT TERMS
cybersecurity, software, data mining

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Kevin W. Hamlen
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	SAR	11	9728834724

Final Report
AFOSR Grant No. FA9550-14-1-0173

Automated, Binary Evidence-based Attribution of Software Attacks

Period of Performance: 07/01/2014 – 06/30/2018

July 30, 2019

1 Principal Investigators

PI: Dr. Kevin W. Hamlen
Computer Science Department, EC31
The University of Texas at Dallas
Richardson, TX 75080-3021
Voice: (972) 883-4724
Email: hamlen@utdallas.edu
Web: https://www.utdallas.edu/~hamlen

Co-PI: Dr. Zhiqiang Lin
Computer Science and Engineering Department
The Ohio State University
Columbus, OH 43210-1277
Voice: (614) 292-0055
Email: zlin@cse.ohio-state.edu
Web: https://web.cse.ohio-state.edu/~lin.3021

Co-PI: Dr. Latifur Khan
Computer Science Department, EC31
The University of Texas at Dallas
Richardson, TX 75080-3021
Voice: (972) 883-4137
Email: lkhan@utdallas.edu
Web: https://www.utdallas.edu/~lkhan

2 Project Motivation and Summary

This project investigated and developed software attack attribution technologies and algorithms with the goal of facilitating automatic inference of possible origins and authorships of cyberattacks (e.g., malware attacks). Our focus is on semantic and syntactic analysis of binary attack payloads (e.g., executable malware binaries and malicious data submitted to vulnerable programs), without significant reliance on network traffic analysis or tracking. This addresses the considerable class of cyberattacks for which meaningful network routing information is unavailable or obfuscated (e.g., due to adversarial proxying, anonymous routing, or botnetting), but the received attack payloads are logged and available for analysis.

Our attribution strategies are based on scientific approaches to binary software reverse engineering and analysis, and machine learning-based approaches to attribution data analysis. This allows our results to be fully automated, for rapid response to emerging digital threats; it helps to remove human bias from the analysis, yielding objective attribution hypotheses and confidence scores; and it broadens applicability of our discoveries to a variety of software and data analysis problems.

3 Achievements of the Project

Our research throughout this project primarily focused on three major thrusts:

- Design, implementation, and analysis of a new attribution information gathering technology called *honey-patching*, which employs deception to solicit and gather attribution-relevant data on present and future cyberattacks.
- Design, implementation, and analysis of new binary code decomposition techniques, which aim to automatically decompose binary code into components such that the syntactic and semantic features of each component can be utilized in signatures for authorship attribution.
- Design, implementation, and analysis of new algorithms to: (1) identify authors over textual stream data where new authors may emerge at any time, and (2) identify websites over anonymity networks using machine learning techniques.

Summaries of final outcomes for these respective thrusts are detailed below.

3.1 Attribution Data Collection Through Cyber Deception

3.1.1 Honey-patching

One of our most significant achievements from this project is the discovery of a new approach to software security patching that substitutes vulnerabilities with traps that ensnare attackers. We call these deceptive patches *honey-patches*. Honey-patches make attacks against patched software systems look as though they succeeded even when the attacks are actually blocked. Instead of doing real damage or stealing real secrets, the attacker sees fake damage and fake secrets that can be laced with disinformation to lead attackers astray. Meanwhile, all attacker actions are meticulously monitored, revealing attacker methodologies, objectives, and gambits to defenders.

Our breakthrough invention drew international press coverage and local television news appearances in 2014 when we used it to honey-patch the now-famous Heartbleed vulnerability within hours of its disclosure:

- The Times of India. *New Technique Red Herring Fights 'Heartbleed' Virus.* April 15, 2014.
- Brian New, CBS-11 Nightly News, Dallas/Fort Worth. *North Texas Professor Setting Trap for Hackers.* April 14, 2014.
- Selena Hernandez, CW-33 Nightcap News. *UTD Professor Creates Solution to 'Heartbleed' Bug.* April 15, 2014.

Our scientific results were published in the highly competitive, top-ranked *2014 ACM Computers and Communications Security Conference (CCS)* [12]—widely regarded as one of the highest impact peer-reviewed applied security publication venues worldwide. Please see that publication for technical details and evaluation results. We subsequently won 2nd prize at the **NYU-Poly Best Applied Security Paper of the Year** competition, which awards the top applied security discoveries of all North American authors across all top conferences and journals:

- NYU Polytechnic School of Engineering. *World's Biggest Student Cyber Security Contests Reveal Best Young Hackers and Researchers.* November 17, 2014. http://engineering.nyu.edu/press-release/2014/11/17/worlds-biggest-student-cyber-security-contests-reveal-best-young-hackers-re

Our technical approach to this objective extends the state-of-the-art in process migration through fast, lightweight checkpoint-restart, to transparently redirect attacks to decoy environments live, at the moment the attack is detected. We have devised security retrofitting algorithms that imbue legacy software with such capabilities without requiring cooperation from its original developers.

3.1.2 Tool-assisted Software Honey-ification

Our second major breakthrough in this thrust is the development of compiler-side methodologies and tools that allow deceptive software to be mass-produced, for better attack attribution capabilities. To achieve this, we developed *SignaC (Secret Information Graph iNstrumentation for Annotated C)*—an extension to the LLVM C-compiler that imbues compiled programs with the ability to redact and replace secrets in its address space with honeydata during cyberattacks. SignaC-compiled programs thereby respond deceptively to attacks, allowing adversaries to penetrate a fake copy of the program whose data misdirects and misleads them.

Our results were published at the world-renowned USENIX Security Symposium [10, 13]. Please see those publications for detailed analyses and evaluation results. We published follow-up results and larger experimental outcomes as a chapter [11] in Springer's *Cyber Deception: Building the Scientific Foundation* text.

During the final year of the project lead PI Hamlen co-edited a new Springer textbook on *Autonomous Cyberdeception* [7], including a contributed chapter on integrating honey-patching with firewalls to realize deceptive firewalls for attacker attribution [9]. This text is now available in 2019:

> *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*
> E. Al-Shaer, J. Wei, K.W. Hamlen, and C. Wang, editors
> Springer, 2019
> https://www.springer.com/gp/book/9783030021092

Additional details on how to realize these deceptive firewalls within software-defined networks (SDNs) was published in NOMS [21]. This publication shows that SDNs have the advantage of being able to quickly adapt network topologies to partition and isolate attack traffic buried within a sea of non-malicious traffic. The advantage of this is that attacks are more susceptible to deobfuscation and attribution if they can be confined to decoy environments without disrupting the communication. The confined process can then be allowed to perform dangerous actions that might harm the confined environment, since that environment hosts no services or data of genuine value. Please see the NOMS publication cited above for details.

Ph.D. Graduation. Student research on honey-patching and software honey-ification culminated in the graduation of one Ph.D. student whose dissertation [8] won the 2017 **UTD EECS Best Dissertation Award** (winner among 89 competing dissertations). The student was hired by IBM T.J. Watson Research and is now active there in cyber deception research for government and industry.

3.1.3 Machine learning-based Binary Disassembly

Toward attacker attribution of malware payloads, we have innovated several new techniques for automated, binary software disassembly based on data mining. Binary disassembly is a crucial step in obtaining reliable semantic information from malware attacks and code reuse attacks. Our scientific results were published at two highly ranked, peer-reviewed conferences—one devoted to data mining (PAKDD) [27] and the other to security (RAID) [26]. Our evaluation shows that this new approach yields more accurate disassemblies of binary software than even the best commercially available reverse engineering tools, and at faster speeds. (Please see cited publications for technical details.)

Follow-on work on this problem culminated in our invention of Superset Disassembly, which can reliably disassemble any Intel x86/x64 program irrespective of obfuscations, and without relying upon heuristics. This achievement was published in the highly competitive Network and Distributed Systems Symposium (NDSS) in the final year of the project (2018) [14].

3.2 Binary Code Decomposition

Since a key goal of this project is to attribute the binary code, we must first decompose binary code into components, each of which might be the product of a distinct developer or development group. With the decoposed components, code attribution can then be performed.

3.2.1 Binary Code Decomposition through Delinking

To decompose software programs (e.g., malware) into the constituent components whence they were constructed, we have developed a novel *delinker* technology that inverts the functionality of linkers. Delinking an executable into object code is potentially useful in many applications, especially in the scenario of binary code reuse and binary code attribution. For example, it can potentially extract the unpacking component from packed malware for use in automated malware unpacking, and it can potentially separate attribution-relevant malware components from generic components that lack attribution-relevant features.

Our DELINKER system enables binary code reuse and attribution through delinking of an executable into relinkable object code. The input to DELINKER is the application binary code without any symbols, and the output is the (re-)linkable object code per function. Much like the traditional way of using library code for which source code is not available (e.g., Windows DLLs), the object code produced by DELINKER can be linked into new software.

DELINKER relocates the program code and data, rearranges the addresses of the functions and pointers, and regenerates the relocation tables so that a LINKER can later link the object file with another program. To address these technical challenges, we have designed an array of static binary code analysis techniques including (i) binary code decomposition that splits the executable code into relinkable object files, (ii) memory address recognition that identifies instruction operands as well as static values in global memory regions that are memory address (i.e., pointer) related including both data pointer and code pointers such as jump tables, (iii) relocation table reconstruction that rebuilds the relocation tables with the newly generated symbols for each object file such that they can be relinkable when reusing the delinked functions.

We have implemented our DELINKER. Our evaluation with the SPEC CPU2006 benchmark shows DELINKER can delink an executable into object code, and relink the object code using off-the-self linkers without any errors. Our case studies show that DELINKER can significantly save programmers' efforts in developing new software by reusing complicated cryptographic functions from both benign and malicious software. An MS thesis [22] has been produced to describe the tecnical details and experimental results of DELINKER.

3.2.2 Binary Code Decomposition and Clustering Based on Locality

DELINKER has made a large step towards our end goal of attributing the authors of the binary code in that it can automatically decompose binary code into pieces. However, these pieces are too coarse-grained (namely it is at function boundary level). Considering a programmer usually writes several functions together, and we have to hoist the functions into a higher level abstractions, namely, logic components. Interestingly, we notice the locality principle can be used to cluster the functions into components.

In particular, we have developed a binary decomposition technique to split binary code into a set of independent logical components based on locality programming paradigm and modularity practice used while developing the software. We found that code locality and data locality can be used to recover modularity abstractions lost in the linking process. To this end, we developed a hybrid graph based approach that combines function control flow graph and data references made by each function. From the clear boundary of the locality using graph clustering, we decompose the software into independent logical components. We have validated our approach using a dataset of 17 projects and demonstrated that our proposed method can achieve very high accuracy. The proposed techniques and the results obtained have been published in ACM ASIACCS 2018 [19]

3.3 Author Attribution Over Textual Data And Anonymity Networks

Throughout this project we have developed machine learning-based techniques to efficiently recognize authors over textual data and identify websites over anononymous networks. Significant accomplishments for this research thrust are summarized in this section.

3.3.1 Author Attribution Over Textual Data

In this work, we considered the Author Recognition problem over textual data in the streaming environment as one of the attention receiving sources for concept-evolution studies. When a new class occurs in the data stream, it can be considered as a new concept; thus the concept-evolution refers to the emergence of a new author in our study. We approached the problem by defining a novel ensemble technique *class-based* ensemble, which replaces the traditional chunk-based approach in order to detect the recurring classes. Different datasets have been evaluated including the IMDB62 dataset (gathered by Seroussi et al.[1] from the Internet movie database[2]). It consists of a collection of movie reviews with 62,000 comments from 62 authors and each author has 1,000 comments. The evaluation demonstrates that our class-based ensemble approach outperforms the traditional chunk-based one. This work has been submitted to TKDE [1].

3.3.2 Website Fingerprinting Over Anonymity Networks

Website fingerprinting is a passive traffic analysis attack which has dangered the web navigation privacy. By using anonymous communication, Internet users (such as online activists) may wish to hide the destination web pages they access. Such anonymity can be achieved with applications like SSL/TLS or Tor (The Onion Router) where layers of encryptions are added to the exchanged packets. We introduce new statistical approach based attacks [6] that yield higher accuracies than previous studies. The problem is considered as a game between two ends, attackers and defenders. Attackers try to reveal web page destinations while defenders try to trick the attacker by changing the characteristics of the network traffic. We focused on the attacks in this work, and proposed defenses in subsequent works (see below).

To overcome the shortcomings of traditional website fingerprinting studies, we also appraise semantic and syntactic analysis for feature extraction techniques and study the effect of NLP vector space representations in website fingerprinting. We propose the packet to vector (*P2V*) approach [2] where we model the website fingerprinting attack using Global Vector space representations (GloVe). We construct a corpus from network packets and represent these packets as fixed d-dimentional real-valued vectors which are then utilized as features. Our evaluation shows the outperformance of this model over the previous website fingerprinting works.

[1] Y. Seroussi, F. Bohnert, and I. Zukerman. Personalized rating predictions for new users using latent factor models. In *Proc. ACM Conf. Hypertext and Hypermedia*, 2011.

[2] www.imdb.com

3.3.3 Encrypted Traffic Fingerprinting Attack

We present a novel method, called BIND (fingerprinting with BI-directioNal Dependence) [3], to extract features from encrypted network traffic to identify an end-node. We consider relationships among sequences of network packets occurring over sequential transmissions in opposite directions. These features are included in the conjunction with other independent features to enrich discriminating factors of end-nodes during pattern recognition. Furthermore, we explore the temporal nature of encrypted traffic and introduce an adaptive model that considers changes in data content over time. Our approach continuously monitors the classifier performance on the training data. When the accuracy drops below a predefined threshold, we replace the classifier with another trained one on the latest data. We call this ADABIND (ADAptive fingerprinting with BI-directioNal Dependence). We evaluate our analysis on two packet encrypted applications: website fingerprinting and mobile application (app) fingerprinting. Our evaluation shows how the proposed approach outperforms previous works especially in the open-world scenario and when defense mechanisms are considered.

3.3.4 Defense against Website Fingerprinting Attacks

To counteract website fingerprinting attacks, we introduced a novel defense algorithm called 'BiMorphing' [5]. The proposed defense thwarts the fingerprinting attacks by considering bi-directional dependence between consecutive sequences of packets in opposite directions. It obfuscates website patterns through the use of bi-directional statistical sampling; and through the use of mathematical optimization techniques, it achieves minimal bandwidth overhead and zero-delay transmission to actual traffic. We implement and evaluate our approach against a Tor dataset and show how the proposed methodology outperforms the state-of-the-art studies.

3.3.5 Game cheat detection over encrypted traffic network

One way to evaluate and study adversarial mining algorithms for encrypted traffic in the presence of evasive, obfuscating opponents is to apply the algorithms encrypted gaming traffic. We therefore leveraged machine learning based models to predict cheats over encrypted game traffic during game play of massive multiple online games (MMOGs). Detecting cheats is challenging mainly due to the limited client-side information, along with unknown and complex cheating techniques. Moreover, a major challenge in developing such a prediction model is the availability of sufficient training data, which is sparingly available in practice. To detect cheating in MMOGs during game play, we propose to use supervised machine learning techniques over encrypted network traffic [15]. Our framework, called GCI (Game Cheating Identification), addresses the challenge of limited labeled data for training such a model by utilizing the relative density ratio to estimate importance weights associated with training data instances. Our approach introduces an expectation-maximization technique to automatically learn model parameters for relative density ratio estimation from available data. For feature extraction from the encrypted network traffic traces, we utilize our previous method BIND [3], which analyzes consecutive packet bursts to capture any dependencies that may exist between them. We also demonstrate the scalability of the proposed model with the aid of Apache

Spark. Our empirical evaluation on a popular MMOG demonstrates significant improvement in cheat prediction compared to other competing methods.

3.3.6 A GPU-based Game cheat detection over encrypted traffic network

We further extend our work of game cheat detection over encrypted traffic network [15] by introducing Graphics Processing Units (GPUs) in our framework. Furthermore, we aim to exploit high memory bandwidth and powerful parallel processing capability of GPU to make our framework more faster, and therefore reduce the processing burden from the CPU [18]. We offload major time and energy consuming operations in our approach, such as the learning of parameters and hyper-parameters searching for an estimator, to the GPU. Our evaluation indicates that GPU based implementation of our procedure reduces execution time, especially when the data set is substantial. This also verifies that in this particular case, GPU-based approach performs better than Spark and the baseline approach.

Publications and Submissions

[1] Tahseen Al-Khateeb, Mohammad M. Masud, Khaled Al-Naami, Sadi Evren Seker, Ahmad M. Mustafa, Latifur Khan, Charu Aggarwal, and Jiawei Han. Recurring and novel class detection using class-based ensemble for evolving data stream. Manuscript submitted for publication to *IEEE Transactions on Knowledge and Data Engineering (TKDE)* for the second round, 2015.

[2] Khaled Al-Naami, Gbadebo Ayoade, Asim Siddiqui, Nicholas Ruozzi, and Latifur Khan. P2v: Effective website fingerprinting using vector space representations. Manuscript submitted to *the 2015 IEEE Symposium on Computational Intelligence in Cyber Security (IEEE CICS'15)* for review, 2015.

[3] Khaled Al-Naami, Swarup Chandra, Ahmad Mustafa, Latifur Khan, Zhiqiang Lin, Kevin Hamlen, and Bhavani Thuraisingham. Adaptive encrypted traffic fingerprinting with bi-directional dependence. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 177–188. ACM, 2016.

[4] Khaled Al-Naami, Swarup Chandra, Ahmad M. Mustafa, Latifur Khan, Zhiqiang Lin, Kevin W. Hamlen, and Bhavani M. Thuraisingham. Adaptive encrypted traffic fingerprinting with bi-directional dependence. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC)*, pages 177–188, Los Angeles, California, December 2016.

[5] Khaled Al-Naami, Amir El Ghamry, Md Shihabul Islam, Latifur Khan, Bhavani M Thuraisingham, Kevin W Hamlen, Mohammed Alrahmawy, and Magdi Rashad. Bimorphing: A bi-directional bursting defense against website fingerprinting attacks. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[6] Khaled Al-Naami, Ahmad M. Mustafa, Murat Kantarcioglu, Zhiqiang Lin, and Latifur Khan. Efficient website fingerprinting with bi-direction bursting features. Manuscript submitted to *the 18th International Symposium on Research in Attacks, Intrusions and Defenses (RAID'15)* for review, 2015.

[7] Ehab Al-Shaer, Jinpeng Wei, Kevin W. Hamlen, and Cliff Wang, editors. *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*. Springer, 2019.

[8] Frederico Araujo. *Engineering Cyber-deceptive Software*. PhD thesis, The University of Texas at Dallas, Richardson, Texas, August 2016.

[9] Frederico Araujo, Gbadebo Ayoade, Kevin W. Hamlen, and Latifur Khan. Deception-enhanced threat sensing for resilient intrusion detection. In Ehab Al-Shaer, Jinpeng Wei, Kevin W. Hamlen, and Cliff Wang, editors, *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*, chapter 8, pages 147–165. Springer, 2019.

[10] Frederico Araujo and Kevin W. Hamlen. Compiler-instrumented, dynamic secret-redaction of legacy processes for attacker deception. In *Proceedings of the 24th USENIX Security Symposium*, Washington, D.C., August 2015.

[11] Frederico Araujo and Kevin W. Hamlen. Embedded honeypotting. In Sushil Jajodia, V.S. Subrahmanian, Vipin Swarup, and Cliff Wang, editors, *Cyber Deception: Building the Scientific Foundation*, chapter 10, pages 195–225. Springer, 2016.

[12] Frederico Araujo, Kevin W. Hamlen, Sebastian Biedermann, and Stefan Katzenbeisser. From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation. In *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)*, pages 942–953, Scottsdale, Arizona, November 2014.

[13] Frederico Araujo, Mohammad Shapouri, Sonakshi Pandey, and Kevin W. Hamlen. Experiences with honey-patching in active cyber security education. In *Proceedings of the 8th Workshop on Cyber Security Experimentation and Test (CSET)*, USENIX Security Symposium, Washington, D.C., August 2015.

[14] Erick Bauman, Zhiqiang Lin, and Kevin W. Hamlen. Superset disassembly: Statically rewriting x86 binaries without heuristics. In *Proceedings of the 25th Network and Distributed Systems Security (NDSS)*, San Diego, California, February 2018.

[15] Bo Dong, Md Shihabul Islam, Swarup Chandra, Latifur Khan, and Bhavani Thuraisingham. Gci: A transfer learning approach for detecting cheats of computer game. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1188–1197. IEEE, 2018.

[16] Ahsanul Haque, Swarup Chandra, Latifur Khan, Kevin Hamlen, and Charu Aggarwal. Efficient multistream classification using direct density ratio estimation. In *Proceedings of the 33rd*

IEEE International Conference on Data Engineering (ICDE), pages 155–158, San Diego, California, April 2017.

[17] Ahsanul Haque, Zhuoyi Wang, Swarup Chandra, Bo Dong, Latifur Khan, and Kevin W. Hamlen. FUSION: An online method for multistream classification. In *Proceedings of the 26th ACM Conference on Information and Knowledge Management (CIKM)*, pages 919–928, Singapore, November 2017.

[18] Md Shihabul Islam, Bo Dong, Swarup Chandra, Latifur Khan, and Bhavani Thuraisingham. Gci: A gpu based transfer learning approach for detecting cheats of computer game. Manuscript submitted to *IEEE Transactions on Dependable and Secure Computing (TDSC)* for review, 2019.

[19] Vishal Karande, Swarup Chandra, Zhiqiang Lin, Juan Caballero, Latifur Khan, and Kevin Hamlen. Bcd: Decomposing binary code into components using graph-based clustering. In *Proceedings of the 13th ACM Symposium on Information, Computer and Communications Security*, June 2018.

[20] Ahmad M. Mustafa, Gbadebo Ayoade, Khaled Al-Naami, Latifur Khan, Kevin W. Hamlen, Bhavani M. Thuraisingham, and Frederico Araujo. Unsupervised deep embedding for novel class detection over data stream. In *Proceedings of the 5th IEEE International Conference on Big Data (BigData)*, pages 1830–1839, Boston, Massachusetts, December 2017.

[21] Bahman Rashidi, Carol J. Fung, Kevin W. Hamlen, and Andrzej Kamisinski. HoneyV: A virtualized honeynet system based on network softwarization. In *Proceedings of the 14th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Taipei, Taiwan, April 2018.

[22] Brian Schieb. Delinker: Automatic delinking of elf executables for binary code reuse. MS Thesis, UT Dallas, 2016.

[23] Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. Personalised rating prediction for new users using latent factor models. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, 2011.

[24] Meera Sridhar, Mounica Chirva, Benjamin Ferrell, Dhiraj Karamchandani, and Kevin W. Hamlen. Flash in the dark: Illuminating the landscape of ActionScript web security trends and threats. *Journal of Information Systems Security (JISSec)*, 13(2):59–96, December 2017.

[25] Bhavani Thuraisingham, Murat Kantarcioglu, Kevin Hamlen, Latifur Khan, Tim Finin, Anupam Joshi, Tim Oates, and Elisa Bertino. A data driven approach for the science of cyber security: Challenges and directions. In *Proceedings of the IEEE 17th Conference on Information Reuse and Integration (IRI)*, Pittsburgh, Pennsylvania, July 2016.

[26] Richard Wartell, Yan Zhou, Kevin W. Hamlen, and Murat Kantarcioglu. Shingled graph disassembly: Finding the undecideable path. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, pages 460–462, Saint Lucia, October 2013.

[27] Richard Wartell, Yan Zhou, Kevin W. Hamlen, and Murat Kantarcioglu. Shingled graph disassembly: Finding the undecidable path. In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 273–285, Tainan, Taiwan, May 2014.

[28] Chaoshun Zuo and Zhiqiang Lin. Smartgen: Exposing server urls of mobile apps with selective symbolic execution. In *Proceedings of the 26th World Wide Web Conference (WWW'17)*, Perth, Australia, April 2017.