Darknet Mining and Game Theory for Enhanced Cyber Threat Intelligence

John Robertson, Ahmad Diab, Ericsson Marin, Eric Nunes,
Vivin Paliath, Jana Shakarian, Paulo Shakarian [1]
Arizona State University

ABSTRACT

Due to a recent increase in popularity, Darknet hacker marketplaces and forums now provide a rich source of cyber threat intelligence for security analysts. This paper offers background information on Darknet hacker communities and their value to the cybersecurity community before detailing an operational data-collection system that is currently gathering over 300 threat warnings per week, with a precision of around 90% (Nunes 2016). Additionally, we introduce a game theoretic framework designed to leverage the exploit data mined from the Darknet to provide system-specific policy recommendations. For the framework, we provide complexity results, provably near-optimal approximation algorithms, and evaluations on a dataset of real-world exploits.

2. INTRODUCTION

The term "Darknet" refers to the anonymous communication provided by crypto-networks like "Tor". Contrast this definition with that of "Deepnet," which commonly refers to those sites hosted on the open portion of the Internet (i.e. the "Clearnet"), but are not indexed by search engines (Lacey 2015). Library catalogs and corporate websites for internal company use are good examples of deepnet presences.

Many corporations and government agencies rely on extensive penetration testing to assess the security of their computer networks. In a penetration test, a red team is hired to expose major flaws in the organization's security infrastructure. Recently, however, the market for exploit kits has continued to evolve, and what was once a rather hard to penetrate and exclusive market, whose buyers were primarily western governments (Shakarian 2013), has now become more accessible to a much wider population. Specifically, the Darknet portions of the internet is accessible through anonymization

John Robertson is a student at Arizona State University pursuing undergraduate degrees in both Computer Science and Electrical Engineering. He is the recipient of an Army Research Office Undergraduate Research Apprenticeship Program (ARO URAP) grant as well as two Fulton Undergraduate Research Initiative (FURI) grants for his work involving the application of artificial intelligence techniques to cyber-security problems in the Cyber-Socio Intelligent System (CySIS) Laboratory with Dr. Paulo Shakarian. For his work, John was nominated for the Computing Research Association's Outstanding Undergraduate Researcher award by the Computer Science faculty at ASU. John also has industry experience as a software engineering intern with Microsoft on the Windows Core Development team.

Ahmad Diab is a Computer Engineering Ph.D. student at Arizona State University. His current work in the Cyber-Socio Intelligent System (CySIS) Laboratory focuses on the application of artificial-intelligence techniques to cyber-security problems. Ahmad is a recipient of SIPGA award from ASTAR agency, Singapore. Previously, he was a Java developer at EtQ compliance Company. Ahmad holds a B.S. in computer engineering from Jordan University of Science and Technology (JUST).

Ericsson Marin is a Computer Science Ph.D. Student at Arizona State University. He works at the Cyber-Socio Intelligent System (CySIS) Laboratory under the guidance of Dr. Paulo Shakarian, with research projects in the intersection of Social Network Analysis (SNA), Artificial Intelligence (AI) and Cyber-Security. He received his MSc in Computer Science from Federal University of Goias, Brazil, and has published numerous papers in the area of social network analysis. He also holds a BSc in Computer Science and a Specialization in Software Quality and Management from Pontifical Catholic University of Goias, Brazil. He also has a real world experience as software designer managing different software factories. In 2015, Ericsson was awarded with a Brazilian Science Without Borders scholarship to pursue his Ph.D.

Vivin Paliath is a Computer Science Ph.D. student at Arizona State University. His research at ASU focuses on the application of artificial intelligence and game-theoretic techniques to cyber security problems. Vivin received both his B.S. in Computer Engineering and M.S. in Computer Science from Arizona State University. He has over a decade of industry experience and is also currently working as a Senior Software Engineer at Infusionsoft, a company that develops marketing-automation software for small businesses.

Jana Shakarian is a research scientist at Arizona State University and has been researching malicious hacking groups and their online activity since 2012. She has co-authored two books, Elsevier's *Introduction to Cyber-Warfare* and Springer's *Computational Analysis of Terrorist Groups: Lashkar-e-Tabia*. She holds an M.A. in Sociology and Cultural and Social Anthropology from the Johannes Gutenberg University, Mainz, Germany. Previously, she worked as a staff social scientist for the University of Maryland Institute for Advanced Computer Studies (UMIACS) where she worked on major projects funded by the U.S. Air Force and Lockheed Martin.

Paulo Shakarian is an Assistant Professor at Arizona State University's School of Computing, Informatics, and Decision Support Engineering where he directs the Cyber-Socio Intelligent System (CySIS) Laboratory-specializing in cyber-security, social network analysis, and artificial intelligence. He has written numerous articles in scientific journals and has authored several books, including Elsevier's *Introduction to Cyber-Warfare: A Multidisciplinary Approach*. His work has been featured in the major news media such as *The Economist, Popular Science,* and *WIRED*. He is a Cybersecurity Fellow with New America, a recipient of the Air Force Young Investigator Award, MIT Technology Review's "Best of 2013", and the DARPA Service Chiefs' Fellowship. Paulo also has won grant awards from ARO, ONR, DARPA, and others. Previously, Paulo was an Assistant Professor at West Point. Paulo holds a Ph.D. and M.S. in computer science from the University of Maryland, College Park, and a B.S. in computer science from West Point (with a Depth of Study in Information Assurance).

protocols such as Tor and i2p, which are now populated with multiple markets specializing in such products (Shakarian 2016; Ablon 2014). In particular, 2015 saw the introduction of Darknet markets specializing in zero-day exploit kits—exploits designed to leverage previously undiscovered vulnerabilities. These exploit kits are difficult and time-consuming to develop and often sold at premium prices, at times exceeding tens of thousands of dollars in cost. The widespread availability of zero-day exploits represents a potential game changer for penetration testers, specifically posing the following questions:

- *How can we automatically mine for new exploits and malware for sale in the malicious hacking community?*
- *What exploits will an attacker likely purchase if he targets a specific organization?*
- *What software used in the organization pose the biggest risk to new threats?*

However, the high cost of a variety of exploits available on the Darknet may preclude a penetration tester from simply obtaining them. In this paper, we present initial work that highlights steps toward solving these problems. To address the first question, we explore Darknet exploit markets and hacker forums through a data collection system to scrape, parse, and filter the web data. This data is then used as input to a novel, data-driven security game framework to address the second two questions. Specific contributions of this work include the following.

- A description of a system for automatically crawling and parsing Darknet malicious hacking information.
- A game-theoretic framework that, given a system configuration (or a distribution of system configurations within an organization) models an attacker as an agent who, with a finite budget, will purchase exploits to maximize his level of access to the target system. Likewise, a defender will look to adjust system configurations in an effort to minimize the effectiveness of an attacker while ensuring that necessary software dependencies are satisfied.
- A thorough formal analysis of the problems in the game-theoretic framework, including computational complexity results and approximation algorithms to identify provably near-optimal strategies for both players.
- A suite of experimental results on a prototype system that implements our game theoretic framework to demonstrate the effectiveness of this approach.

Paper organization. This paper's organization is as follows. Section 3 presents background information about the Darknet and the exploit marketplaces, and hacker forums that preside on the Darknet. Section 4 then details a data collection system for scraping and parsing these Darknet communities, including some of the technical challenges involved with utilizing such a system to provide up-to-date cyber threat intelligence. Section 5 includes a game theory framework, which mathematically formalizes problems for both the Attacker and Defender in a cyberattack scenario, along with complexity results and approximation algorithms for the framework. Finally, Section 6 presents the results of applying our framework on real-world Darknet exploits.

3. BACKGROUND

There are now a number of online communities providing users with both the ability to stay anonymous and the ability to reach geographically dispersed collaborators. As an illustration of the activity occurring on these communities, consider the exploit *MegalodonHTTP* Remote Access Trojan (RAT), which utilize the amateur black hat platform, HackForum, to facilitate its distribution. Five people accused of the malware's creation and/or distribution resided in three separate European countries, requiring law enforcement to cooperate internationally in pursuit of the malicious hackers'arrest (Wei 2013).

Darknet hacker marketplaces and forums now provide a rich source of cyber threat intelligence for security analysts.

Darknet and Deepnet Sites. Widely used for underground communication, *The Onion Router* (Tor) is free software designed to protect the privacy of its users by obscuring traffic analysis, greatly complicating network surveillance (Dingledine 2004). The network traffic in Tor flows through a number of volunteer-operated servers (also called *nodes*). Each node of the network encrypts the information it blindly passes on, neither registering where the traffic came from nor where it is headed (Dingledine 2004). Effectively, this allows not only for anonymized browsing (the IP address revealed will only be that of the exit node), but also for circumvention of censorship. [2]

These online hacker communities may take on a number of different forms. We discuss a few below.

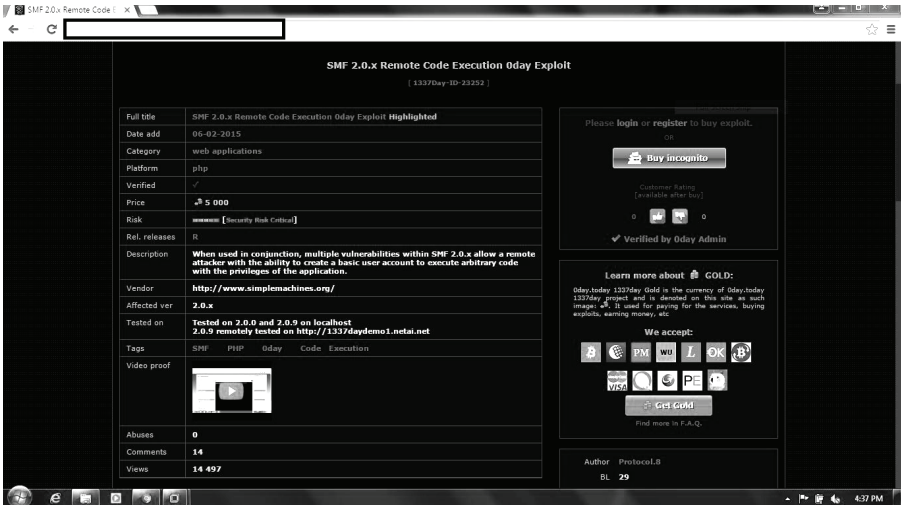

Figure 3.1: Example of Darknet Market

Markets. Darknet marketplaces provide users with a platform for buying and selling illicit merchandise. Common products include drugs, weapons, pornography, and exploits. Figure 3.1 depicts listings for zero-day exploits on one such market. These markets contain rich information about the cyber threat landscape; though commonly only a small fraction of products (12.6% in our collected data to date) are related to malicious hacking. Vendors often advertise their products on non-market communities (e.g. forums) to attract attention towards their goods and services. To facilitate transactions, marketplaces often have a wallet in which users will deposit digital currency, though sometimes administrators will serve as an escrow service. Products are most often verified before any funds are released to the seller, and if a seller is misleading or fails to deliver the appropriate item, they can be banned from the site. Similarly, buyers can be banned for not complying with site-specific transaction rules.

Forums. Forums are user-oriented platforms that have the sole purpose of enabling communication. They provide the opportunity for the emergence of a community of like-minded individuals, regardless of their geophysical location. To ensure user safety and privacy, forum administrators often incorporate different security mechanisms into the site. For example, during registration (though not necessarily with every login) every prospective member has to complete CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart), answer simple questions, solve puzzles or complete simple arithmetic operations, presumably to prevent automated access. Discussion forums on the Darknet consist of boards and sub-boards (also called *child-boards*) filled with threads concerned with different topics (for example the discussion of a platform-specific vulnerability). While the structure and organization of Darknet-hosted

forums might be very similar to the more familiar clearnet-forums, the discussion topics vary distinctly. In the English clandestine Darknet, people interested in cats, steampunk, and the latest conspiracy theories convene, but there is an abundance of arenas dedicated to child pornography (CP), drugs, and weapons. Lengthy threads seek information on the reliability of individual marketplace vendors, and the quality of specific marketplaces in general. As Darknet sites are typically not indexed by search engines (for example Google), frequently these forums will link to other Darknet sites and provide information on other potentially fraudulent websites. Forums concerning malicious hacking will feature discussions on programming and cybersecurity.

Subreddits. Reddit is a clearnet site that acts as a content aggregator where users can come together and form sub-communities focused on specific topics. These sub-communities are subreddits. Some subreddits, specifically the ones that are of interest to our research focus on the discussion of darknet exploit markets. Important information regarding the marketplace environment including reviews of marketplaces, products, and vendors are often discussed on these subreddits. These links and sentiments about markets can provide insight. For instance, we might learn to predict when popular opinion shifts with respect to a certain market. Subreddits also provide information concerning marketplaces and forums that are newly introduced or old ones that are shutting down.

The widespread availability of zero-day exploits represents a potential game changer for penetration testers.

Tor-hosted platforms are often shorter lived than their clearnet counterparts. Darknet sites migrate frequently or alternate through multiple addresses, oftentimes resulting in unreliable availability (or uptime). Through search engines and spider services, which traverse links on the Darknet and aggregate the visited links in a list (similar in nature to a *Crawler* (Section 4.1)), on the Tor-network we were able to find more than sixty forums populated by malicious hackers. Other platforms were discovered through links posted on forums, either on the Tor-network or on the clearnet. About half of these forums use English to communicate (33), but French (8), Russian (4), Swedish (2), and (5) other languages were used. On the clearnet, we found more than seventy forums for black hat hackers, the majority of which are English-speaking (52), 18 are in Russian, and one each in French and Polish.

Related Work

Exploit markets on the Darknet. While Darknet criminal activity over the past decade has been extensively studied for issues such as drug trade (Soska and Christin 2015) and terrorism (Chen 2011), the markets of exploits existing on the Darknet are much less well understood. There has been related work on malicious hacker forums (Zhao et al. 2012;

Li and Chen 2014), which did not focus on the purchase and sale of specific items. Markets of malicious products relevant to cybersecurity have been previously studied (Ablon et al. 2014; Shakarian and Shakarian 2015), but none of these works gathered data on specific exploits (or other products) from either the darkweb or open Internet, nor did they examine the markets through the lens of security games. This work extends the initial results presented in (Robertson 2016) and further describes the collection of price data on specific exploits for sale on the deep web, consequently analyzing them in a security game framework to yield policy recommendations for cyber-defenders tailored for specific system configurations.

Darknet sites migrate frequently or alternate through multiple addresses, oftentimes resulting in unreliable availability.

Security games. In recent years, *security games* where attacker-defender models are used to inform the actions of defenders in military, law-enforcement, and homeland security applications have gained much traction; see (Tambe 2011) for an overview. With regard to cybersecurity, there have been many contributions including intrusion detection (Nguyen et al. 2009), attack graph based games (Lye and Wing 2005) and honeypot placement (Kiekintveld et al. 2015). However, to the best of our knowledge, (Robertson 2016), from which this work extends, represents the first game theoretic approach to host-based defense where the activities of the attacker are informed from an *unconventional* source (information not directly related to the defender's system)—specifically information from Darknet markets in this case. Further, the very recent emergence of Darknet markets specializing in zero-day exploits allows for the integration of information that was unavailable in previous work.

4. DATA COLLECTION

Table 4.1 demonstrates how these communities leverage for valuable cyber threat intelligence, which highlights the lifecycle of a vulnerability from identification to exploitation. FireEye, a major cybersecurity firm, identified that the Dyre Banking Trojan was designed to steal credit card information exploited this particular vulnerability, illustrating how threat warnings gathered from the Darknet can provide valuable information for security professionals. Between Dyre and the similar Dridex banking trojan, nearly 6 out of every 10 global organizations were affected, a shocking statistic. [3]

In another instance, 17-year-old hacker Sergey Taraspov from St. Petersburg, Russia, along with a small team of hackers, allegedly wrote a piece of malware that targeted point-of-sale (POS) software and sold it for $2,000 on a Russian forum/marketplace. This malware was, in turn, used by around forty individuals to steal over 110 million American credit card numbers in the *Target* data breach of 2013. [3]

Timeline	Event
February 2015	Microsoft identifies Windows vulnerability MS15-010/CVE 2015-0057 for remote code execution. There was no publicly known exploit at the time the vulnerability was released.
April 2015	An exploit for MS15-010/CVE 20150057 was found on a Darknet market on sale for 48 BTC (around $10,000-15,000).
July 2015	FireEye identified that the Dyre Banking Trojan, designed to steal credit card numbers, exploited MS15010/CVE 2015-0057[2].

Table 4.1: Exploit example.

To gather exploit information from these Darknet markets, we have assembled a sophisticated data pipeline whose system diagram is depicted in Figure 4.5. The technical challenges associated with this system will be briefly discussed in Section 4.1. This operational system currently collects over 300 cyber threats from Darknet markets each week. Figure 4.2 shows the cumulative count of detected threats for five weeks. Figure 4.3 shows a social network, which connects vendors across multiple marketplaces, built using the collected data. At the time of this writing, we are transitioning the system to a commercial partner. Table 4.4 depicts the current database statistics, including the total amount of data collected and the amount of hacking-specific data. The vendor and user statistics cited considers those individuals associated in the discussion or sale of malicious hacking-related material, as identified by our system. This data can address questions such as,

- *What vendors and users have a presence in multiple Darknet/deepnet markets/forums?*
- *What zero-day exploits are being developed by malicious hackers?*
- *What vulnerabilities do the latest zero-day exploits target?*
- *How can a system's presented attack surface be altered to reduce the potential damage of a cyberattack?*

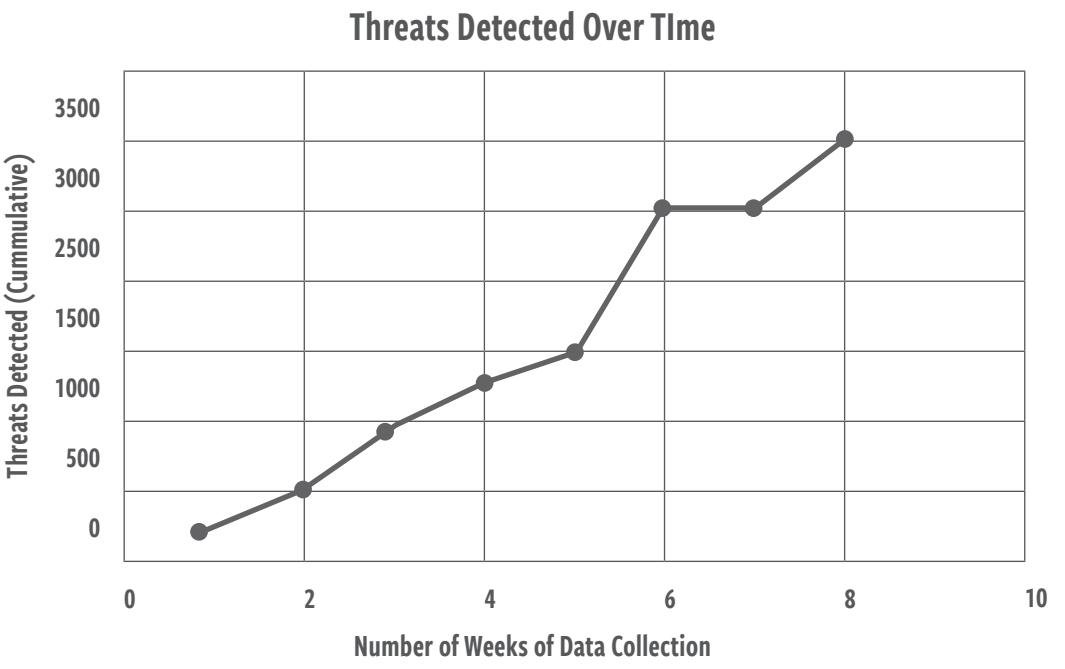

Figure 4.2: Weekly detection of cyber-threats.

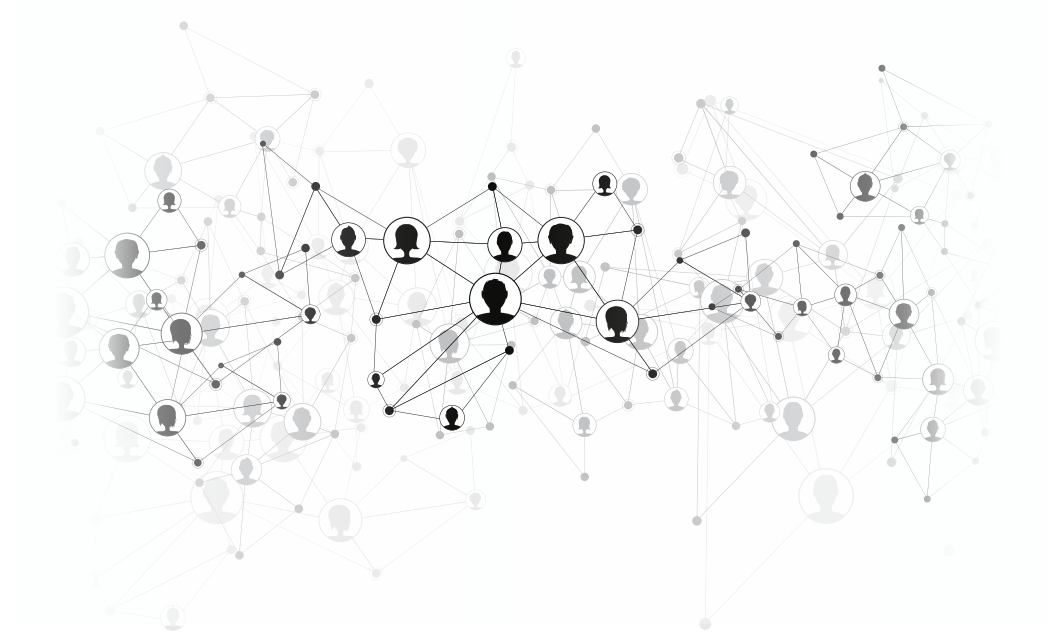

Figure 4.3: Vendor network connecting vendors across multiple marketplaces

Markets	Total Sites Total Products	32 18682
	Hacking-Related	2934
	Vendors (Hacking-Related)	508
Forums	Total Number Total Topics/Posts	23 146053/263363
	Hacking-Related	29636/18392
	Users (Hacking-Related)	11025
Subreddits	Total Number Topics/Posts	33 3940/19601
	Hacking-Related	1654/8270

Table 4.4: Current Database Status

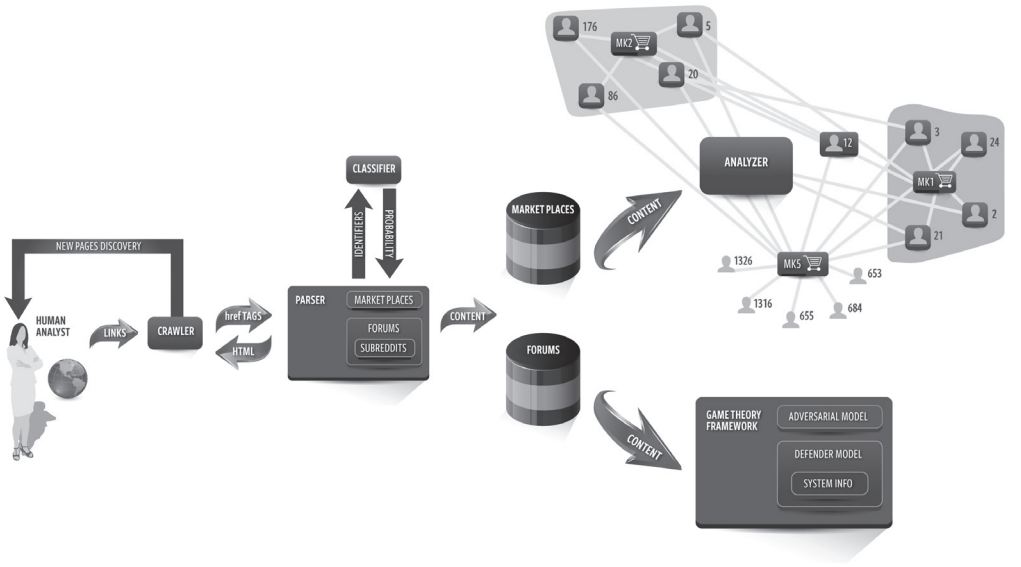

Figure 4.5: System Overview (see page 121 for an enlarged version of the diagram)

4. SYSTEM OVERVIEW

Figure 4.5 gives the overview of the system, whose components are described below.

Crawler. The crawler is a program designed to traverse through a website and retrieve its HTML documents. Topic based crawlers have been used for focused crawling where only webpages of interest are retrieved (Menczer 2004; Chakrabarti 1999). More recently, focused crawling was employed to collect forum discussions from the Darknet (Fu 2010). We have designed separate crawlers for different platforms (markets/forums) due to the structural difference and access control measures for each platform. Our crawler addresses technical challenges such as access control, unresponsive servers, duplicated links (which create a loop), etc., to gather information regarding products from markets and discussions on forums.

Parser. After downloading all html files from a given site, the pages are passed to a parser to extract specific information from marketplaces (e.g. price, vendor, listing date, etc.) and hacker forums (e.g. posts, participating users, etc.). This well-structured information can then be stored in a relational database. Due to idiosyncrasies with each site, typically a unique parser must be written for each site to extract the desired information. The parser also communicates with the crawler; that is, the parser communicates a list of relevant webpages to the crawler, which are then re-crawled to get time-varying data. For markets we collect the following important products fields: {item title, item description, vendor name, shipping details, item reviews, items sold, CVE, items left, transaction details, ratings}. For forums and subreddits we collect the following fields: {topic content, post content, topic author, post author, author status, reputation, topic interest}.

The very recent emergence of Darknet markets specializing in zero-day exploits allows for the integration of information that was unavailable in previous work.

Classifier. As mentioned previously, on these sites not all information is strictly related to cybersecurity and/or hacking. Because of this, it is useful to automate the process of classifying a given product or forum discussion as hacking-related or not. To that end, many data mining techniques are utilized to filter out any irrelevant (meaning not related to cybersecurity) products and discussions. In essence, we leverage a security analyst-labeled dataset with machine learning techniques to detect relevant products and topics from these sites, filtering out products and threads concerning drugs, weapons, and other material not relevant to malicious hacking. Additionally, we leverage topic modeling and other data mining techniques to expedite the process of new site discovery, see (Nunes 2016) for an overview of the machine learning techniques utilized.

Product	Price in BTC	Price in $*
GovRAT (Source Code + 1 Code Signing Certificate Included)	2.000	$456.92
0day Wordpress MU Remote Shell	1.500	$342.69
A5/1 Encryption Rainbow Tables	1.500	$342.69
Unlimited Code Signing Certificate	1.200	$274.16
Ready-made Linux botnet 600 SERVERS	1.200	$274.16
FUD version of Adobe Flash <=16.0.0.287 (CVE 2015-0311)	2.626	$600.00

*Price in U.S. Dollar on date of data collection (Sep. 1, 2015) [1 BTC = $228.46]. As of Aug. 21, 2016, the conversion rate is now [1 BTC = $580.87].

Table 4.6: Example of Products offered on Darknet Markets

5. GAME THEORETIC FRAMEWORK

Here we formalize the concept of our security game where the attacker is a malicious hacker with access to Darknet exploit markets, and the defender is tasked with host-based defense of either a single system or group of systems. We use the notation V to represent the entire set of vulnerabilities within a given computer system. Though there may be vulnerabilities not yet detected by the system administrator, we can mine for information on new vulnerabilities through an examination of Darknet hacking markets. In a real-world organization, system administrators are not able to patch all vulnerabilities for a variety of reasons. Software dependencies, use of legacy systems, and non-availability of patches are some examples. To model this, we define a *constraint set* (denoted C) as a subset of *V.* The vulnerabilities in a constraint set represent the vulnerabilities required for some system functionality. When each vulnerability in a constraint set C is in the presented attack surface (that is externally accessible), C is then said to be satisfied and the system supports the functionality modeled by C. Let $\mathbf{C}$ represent the set of all possible constraint sets. We extend this idea with an *application constraint set* which, for an arbitrary application, i, denoted $\mathcal{C}_i$, is a set of constraint sets (i.e $\mathcal{C}_i \subseteq \mathbf{C}$). Each constraint set in $\mathcal{C}_i$ represents a set of vulnerabilities that together will provide the complete functionality required of application i. $\mathcal{C}_i$ is said to be satisfied if any single constraint set in $\mathcal{C}_i$ is satisfied. If $\mathcal{C}_i$ is satisfied by a system configuration, and hence at least one constraint set in $\mathcal{C}_i$ is satisfied, application i will properly operate on the system. Then $\mathcal{C}$ is the set of all application constraint sets for a given system configuration and represents all of the applications to be

run on the system. In this framework, for a given system, a system administrator must select which vulnerabilities must be present in order to allow each application i to function. This begs the question as to how to make this selection—so we now start to define some concepts relevant to the adversary.

We will use ex to denote a particular exploit—a technique used to take advantage of a given vulnerability. Let Ex denote the set of all possible exploits and $\mathbf{Ex}$ denote the set of all possible exploit sets (i.e. $\mathbf{Ex} = 2^{Ex}$). For each $ex \in Ex$, c_{ex} is the associated cost of exploit ex—and this is specified directly on a Darknet market (normally in Bitcoin). Associated with the set of exploits is the Exploit Function, ExF, which takes a set of exploits as input and returns a set of vulnerabilities (i.e. $ExF : \mathbf{Ex} \rightarrow 2^{V}$). The set of vulnerabilities produced by $ExF(A)$, for a given set of exploits A, represents the vulnerabilities that are exploited by the exploits in A. While many possible variations of an exploit function are possible, in this paper, we will use a straightforward definition that extends the exploit function from singletons (whose associated vulnerabilities can be taken directly from the online marketplaces) to sets of exploits: $ExF(A) = \bigcup_{a \in A} ExF(\{a\})$. For use in proving complexity results, we shall denote the special case where $Ex = V$, $ExF(A) = A$, and $\forall ex \in Ex, c_{ex} = 1$ as the *Identity Exploit Model*.

5.1 PLAYER STRATEGIES AND PAYOFF

An attacker will use a set of exploits to attempt to gain access to a system, and must do so within a budget. Likewise, the defender must identify a set of vulnerabilities that he is willing to expose (often referred to as the *presented attack surface*). We define strategies for the two players formally as follows.

Definition 5.1. *(Attack Strategy). Given budget $k_{atk} \in \mathbb{R}^{+}$, an Attack Strategy, denoted A is a subset of Ex such that $\Sigma_{a \in A} c_a \leq k_{atk}$.*

Definition 5.2. *(Defense Strategy). Given a family of application constraint sets $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_n\}$, a Defense Strategy, denoted D is a subset of V such that for each $\mathcal{C}_i \in \mathcal{C}$, there exists $C \in \mathcal{C}_i$ where $C \subseteq D$ (that is each application constraint is satisfied by D).*

Note that when a defense strategy D meets the requirements of $\mathcal{C}$, as per Definition 5.2, we say D *satisfies* $\mathcal{C}$. We will use the notation $\mathbf{A}, \mathbf{D}$ to denote the set of all attack and defense strategies, respectively, and refer to an attacker-defender pair of strategies as a *strategy profile.* We will also define a *mixed strategy* for both players in the normal manner. For the attacker (respectively defender) a *mixed strategy* is a probability distribution over $\mathbf{A}$ (respectively $\mathbf{D}$). We shall normally denote mixed strategies as Pr_A, Pr_D for each player and use the notation $|Pr_A|$ (respectively $|Pr_D|$) to denote the number of strategies in $\mathbf{A}$ (respectively $\mathbf{D}$) that are assigned a nonzero probability by the mixed strategy. We now turn our attention to the payoff function, which we define formally as follows:

Definition 5.3. *(Payoff Function). A payoff function, p, is any function that takes a strategy profile as an argument and returns a positive real. Formally,*

$$p : \mathbf{A} \times \mathbf{D} \rightarrow \mathbb{R}^{+}$$

Unless noted otherwise, we will treat the payoff function as being computable in polynomial time. Also, the payoff function is underspecified—which is designed to allow flexibility in the framework. However, in the context of the results of this paper, we shall consider the following *payoff function axioms:*

$\forall D \in \mathbf{D}, \forall A \in \mathbf{A}$ such that $ExF(A) \cap D = \emptyset, p(A,D) = 0$	(1)
$\forall D \in \mathbf{D}, \forall D_0 \subseteq D, \forall A \in \mathbf{A}, p(A,D_0) \leq p(A,D)$	(2)
$\forall D \in \mathbf{D}, \forall A \in \mathbf{A}, \forall A_0 \subseteq A, p(A_0,D) \leq p(A,D)$	(3)
$\forall A \in \mathbf{A}, D, D_0 \in \mathbf{D}\ p(A,D) + p(A,D_0) \geq p(A,D \cup D_0)$	(4)
$\forall D \in \mathbf{D}, A, A_0 \in \mathbf{A}, p(A,D) + p(A_0,D) \geq p(A \cup A_0, D)$	(5)

Axiom 1 states that if the vulnerabilities generated by an attack strategy's exploits and the vulnerabilities in a defense strategy are disjoint sets, the payoff function must return 0. A consequence of axiom 1 is that if either the attack strategy or the defense strategy is the empty set, the payoff function will return 0. Axioms 2 and 3 require the payoff function to be monotonic in the size of the attack and defense strategies. Axioms 4 and 5 require the payoff function to be sub-modular with respect to the attack and defense strategies.

In this paper, we shall (in general) focus on the *overlap payoff function,* which we shall define as follows: $p(A,D) = |ExF(A) \cap D|$. Intuitively, this is simply the number of vulnerabilities exploited by the attacker. Further, when dealing with mixed strategies, we shall discuss payoff in terms of expectation. Expected payoff can be formally defined as follows:

$$Exp(Pr_A, Pr_D) = \Sigma_{D \in D} \Sigma_{A \in A} Pr_A(A) Pr_D(D) p(A,D)$$

Using the overlap function, the expected payoff can be interpreted as the *expected number of exploited vulnerabilities.*

5.2 PROBLEM FORMULATIONS

We now have the components to define a pair of decision problems dealing with the best response for the players. These problems are the deterministic host attacker problem (DHAP) and deterministic host defender problem (DHDP), respectively, and are defined as follows:

DHAP

INPUT: $k_{atk} \in \mathbb{R}^{+}$, $x \in \mathbb{R}^{+}$ mixed defense strategy Pr_D, and payoff function p.

OUTPUT: "Yes" if $\exists A \in \mathbf{A}$, such that $\Sigma_{a \in A} c_a \leq k_{atk,}$ and $\Sigma_{D \in D} Pr_D(D)\, p(A,D) \geq x$
"No" otherwise.

DHDP
INPUT: application constraints, mixed attack strategy Pr_A, and payoff function p.
OUTPUT: "Yes" if $\exists D \in \mathbf{D}$, such that $\Sigma_{A \in A} Pr_A(A)\, p(A, D) \leq x$ and D satisfies $\mathcal{C}$ and
"No" otherwise.

The natural optimization variants for these two problems will deal with maximizing the payoff in DHAP and minimizing the payoff in DHDP.

5.3 COMPLEXITY RESULTS

In this section, we analyze the complexity and limits of approximation for both DHAP and DHDP. We use the *Identity Exploit Model* for the complexity results. Unfortunately, both problems are NP-Complete in the general case.

Theorem 1. DHAP is NP-Complete, even when $|Pr_D| = 1$ and the payoff function adheres to the submodularity and monotonicity axioms.

Proof Sketch. Membership in NP is trivial if the payoff is PTIME computable. The hardness result relies on an embedding of the well-known budgeted set cover (Feige 1998). Here, the defender's strategy is treated as a set of elements to cover and the exploits are treated as subsets of D (by virtue of the exploit function). Exploit costs are set as 1 and the attacker's budget is the value budget from the embedded problem. So, the attacker must pick exploits to meet the budget and cover the determined number of the defender's vulnerabilities.

Theorem 2. When $|\mathcal{C}| > 1$ and $|Pr_A| = 1$, DHDP is NP-Complete.

Proof Sketch. Again, membership in NP is trivial if the payoff is PTIME computable. Hardness is shown by embedding the hitting set problem. In this reduction, the attacker plays all exploits and each exploit corresponds with precisely one vulnerability. This has the effect of imposing a unit cost on each vulnerability. Here, each $\mathcal{C}_i$ must be covered by a vulnerability. Hence, the defender must pick a set of all vulnerabilities to meet the cost requirement of DHDP while covering each $\mathcal{C}_i$.

We are also able to analyze the hardness of approximation for the optimization variants of DHAP and DHDP. Because the above embedding's used set cover and hitting set, we can draw upon the results of (Feige 1998) to obtain the following corollaries:

Corollary 3. DHAP cannot be approximated where the payoff is within a factor of $(1-\frac{1}{e})$ unless $P = NP$

Corollary 4. DHDP cannot be approximated where the payoff is within a factor of $(1 - o(1))ln(\mathrm{n})$ unless $P = NP$

5.4 ALGORITHMS

Technical Preliminaries

Definition 5.3 *(Marginal Gain). Given a payoff function p and a mixed defense strategy* Pr_D, $\Delta_{p, Pr_D}(a \mid A)$ *will measure the marginal gain of exploit a in the context of an attack strategy A. That is,* $\Delta_{p, Pr_D}(a \mid A) = \Sigma_{D \in Pr_D} p(A \cup \{a\}, D) - p(A, D)$

With the limits of approximation in mind, we can now introduce several algorithms to solve the optimization variants of DHAP and DHDP. The optimization variant of DHAP under the overlap payoff function is a special case of submodular maximization with the distinction that we are not simply picking k discrete objects, but instead picking items that each have a unique cost associated with them. Understanding this, we examine several different approaches to this problem based on the literature on submodular maximization. DHDP, on the other hand, can be readily approximated using the traditional set-cover algorithm (under some realistic assumptions), as cost does not affect DHDP.

Algorithm 1 Lazy Greedy Algorithm (Cost-Benefit Variant)

Input: $k_{atk} \in \mathbb{R}^+$, Pr_D, and payoff function p.

Output: A $\subseteq Ex$ such that $\Sigma_{a \in A}\, c_a \leq k_{atk}$

1. A $\leftarrow \emptyset$; cost $\leftarrow 0$; priority queue Q $\leftarrow \emptyset$; $iter \leftarrow 1$
2. **for** $e \in Ex$ **do**
3. $\quad e.key \leftarrow \frac{\Delta_{p, Pr_D}(e \mid \emptyset)}{c_e}$; $e.i \leftarrow 1$
4. $\quad$ Insert e into Q with e.key as its key
5. **end for**
6. **while** $\{a \in Ex \setminus A : c_a + cost \leq k_{atk}\} \neq \emptyset$ **do**
7. $\quad$ extract top (max) element e of Q
8. $\quad$ **if** $e.i = iter$ and $c_e + cost \leq k_{atk}$ **then**
9. $\quad\quad A \leftarrow A \cup \{e\}$; $iter \leftarrow iter + 1$
10. $\quad\quad cost \leftarrow cost + c_e$
11. $\quad$ **else if** $c_e + cost \leq k_{atk}$ **then**
12. $\quad\quad e.i \leftarrow iter$; $e.key \leftarrow \frac{\Delta_{p, Pr_D}(e \mid \emptyset)}{C_e}$
13. $\quad\quad$ re-insert e into Q
14. $\quad$ **end if**
15. **end while**
16. return A

Algorithms for DHAP

Greedy Approaches. As mentioned earlier, the non-unit cost of exploits mean that DHAP can be considered as a submodular maximization problem subject to knapsack constraints. Two versions of the traditional greedy algorithm (Nemhauser 1978) can be applied: a cost-benefit variant and uniform-cost variant, both of which will also use the lazy-greedy optimization (Minoux 1978) to further enhance performance while maintaining the approximation guarantee. We note that independently, the uniform-cost and the cost-benefit algorithms can perform arbitrarily badly. However, by extending a result from (Leskovec 2015), either the cost-benefit or the uniform-cost algorithm will provide a solution within a factor of $\frac{1}{2}(1-\frac{1}{e})$ for a given set of input parameters. By applying both algorithms to a given problem instance and returning the attack strategy which produces the larger payoff, the $\frac{1}{2}(1-\frac{1}{e})$ approximation factor is achieved for DHAP. A cost-benefit lazy approximation algorithm is shown in Algorithm 1. By removing "C_e" from the denominator in the *e.key* assignment in lines 3 and 12, the cost benefit lazy approximation algorithm is transformed into a uniform cost lazy approximation algorithm.

Multiplicative Update Approach. An improved approximation ratio, when compared with the $\frac{1}{2}(1-\frac{1}{e})$ ratio for the greedy algorithms, can be obtained by adapting Algorithm 1 from (Azar and Gamzu 2012) for DHAP. This is shown as Algorithm 2 in this paper. For some value ϵ (a parameter), this algorithm provides a $(1-\epsilon)(1-\frac{1}{e})$ approximation of the optimal solution (Theorem 1.2 in (Azar and Gamzu 2012)), which, by providing an exceedingly small ϵ value, can get arbitrarily close to the $(1-1/e)$ optimal approximation limit we discussed earlier.

Algorithm 2 Multiplicative Update

Input: $k_{atk} \in \mathbb{R}^+$ such that $0 < \epsilon < 1$, Pr_D, and payoff function p.

Output: $A \subseteq Ex$ s.t. $\Sigma_{a \in A} c_a \leq k_{atk}$

1. $Ex' \leftarrow \{ex \in Ex : c_{ex} \leq k_{atk}\}$
2. $A \leftarrow \emptyset$
3. $W \leftarrow \min_{ex'_i \in |Ex'|} k^2_{atk}/C_{ex'_i}$
4. $w \leftarrow \frac{1}{k_{atk}}$; $\lambda \leftarrow e^{\frac{\epsilon W}{4}}$
5. **while** $k_{atk} w \leq \lambda$ and $Ex' \neq \emptyset$ **do**
6. $\quad ex_j \leftarrow argmin_{exj \in Ex' \setminus A}(\frac{C_{ex_j}}{k_{atk}} w / \Delta_p, Pr_D(ex_j | A))$
7. $\quad A \leftarrow A \cup \{ex_j\}$
8. $\quad w \leftarrow w\lambda^{C_{ex_j}/k^2_{atk}}$
9. $\quad Ex' \leftarrow Ex' \setminus \{ex_j\}$

10. **end while**
11. **if** $\Sigma_{A_i \in A} c_{A_i} \leq k_{atk}$ **then**
12. return A
13. **else if** $\Sigma_{D \in Pr_D} Pr_D(D)\, p(A \backslash \{ex_j\}, D) \geq \Sigma_{D \in Pr_D} Pr_D(D)\, p(\{ex_j\}, D)$ **then**
14. return $A \backslash \{ex_j\}$
15. **else**
16. return $\{ex_j\}$
17. **end if**

Algorithms for DHDP

When using the overlap payoff function, DHDP can be modeled as a weighted set cover problem. Because the overlap payoff function is a modular function, the associated cost of a given vulnerability v, is simply the payoff produced by the singleton set $\{v\}$ with a mixed attack strategy Pr_A i.e. $c_v = \Sigma_{A \in A} Pr_A(A)\, p(A, \{v\})$. In the common case where each constraint set is a singleton set (i.e. $\forall \mathcal{C}_i \in \mathcal{C}, \forall C \in \mathcal{C}_i, |C|=1$), if the overlap payoff function is used, an adaptation on the standard greedy weighted set cover algorithm can be used for DHDP (Algorithm 3), providing a $ln(n)+1$ approximation (Feige 1998).

Algorithm 3 Weighted Greedy DHDP Algorithm for Singleton Constraint Set and Overlap Payoff Case

Input: Vulnerabilities V, Pr_A, and application constraints $\mathcal{C}$

Output: $D \subseteq V$ s.t. the application constraints $\mathcal{C}$ are satisfied

1. $D \leftarrow \emptyset$
2. $S \leftarrow$ set such that $S_i = \{ j : V_i \in \mathcal{C}_j$ where V_i is the ith vulnerability in $V\}$
3. $c_{S_i} \leftarrow \Sigma_{A \in Pr_A} Pr_A(A)\, |ExF(A) \cap \{V_i\}|$
4. $\mathcal{C}' \leftarrow [\,|\mathcal{C}|\,]$
5. **while** $\mathcal{C}' \neq \emptyset$ **do**
6. $c_{S_i} \leftarrow argmax_{s_i \in S} \frac{|s_i \cap c'|}{s_i}$
7. $\mathcal{C}' \leftarrow C' \backslash S_i$
8. $D \leftarrow D \cup \{V_i\}$
9. **end while**
10. return A

6. EVALUATION AND DISCUSSION

Darknet Market Data. We scraped and parsed eight marketplaces located on the Tor network during the month of May 2015. We use a sample of the products in our database to evaluate this game theoretic framework. This is because the exploit function, which associates Darknet exploits with their targeted vulnerabilities was manually specified by the analyst. The product list used for these experiments was comprised of 167 distinct hacking tools. We found several identical products sold on more than one market usually by the same seller (using an identical online handle). The products targeted 21 specific platforms, such as different versions of Adobe Flash, Linux, MS Windows and OS X as well as online presences such as Facebook, WordPress and others. Hardware-related software such as those associated with point-of-sale machines, routers, and servers are also reflected in this number. Figure 6.1 illustrates the variety of products in the markets and Table 6.2 illustrates exemplar exploits in this dataset.

System Configurations. Figure 6.1 illustrates a variety of platforms represented in our Darknet market data. In this paper, we describe results when using application constraints based on common configurations for Windows and Linux servers—as these were the most prominent targets of exploits found on the Darknet. In our experiments, we mapped software such as media players, databases, and FTP server software to application constraint sets to model the functional requirements of a system. We have also created (and conducted experiments with) models for Android, Point-of-Sale, and Apple systems—though qualitatively the results differed little from the Windows and Linux Server experiments.

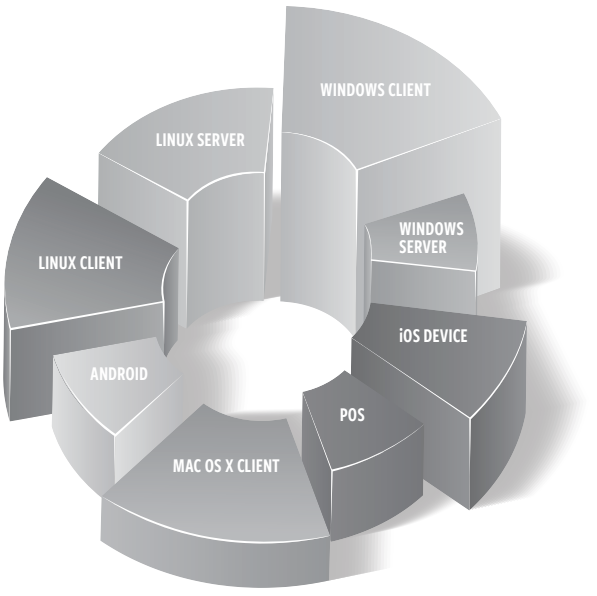

Figure 6.1: Distribution of Exploits with respect to platform.

Product	Vulnerability	Target	USD
Kernel Panic	X-display system	Linux <= 3.13.0-48	$471.56
IE <= 11	memory corr.	IE on Windows <= 7	$35.00
RemoteShell	wpconfig.php	Wordpress MU	$1,500.00
0day RCE	WebView memory corr.	Android 4.1, 4.2	$36.50
WindowsLPE	win32k elev. of priv.	Windows <= 8.1	$12.48
MS15-034 RCE	http.sys	Windows <= 8.1	$311.97
FUD Flash Exp.	unspec.	FlashPlayer <=16.0.0.287	$600.00

Table 6.2: Examples of Exploits from Darknet Markets

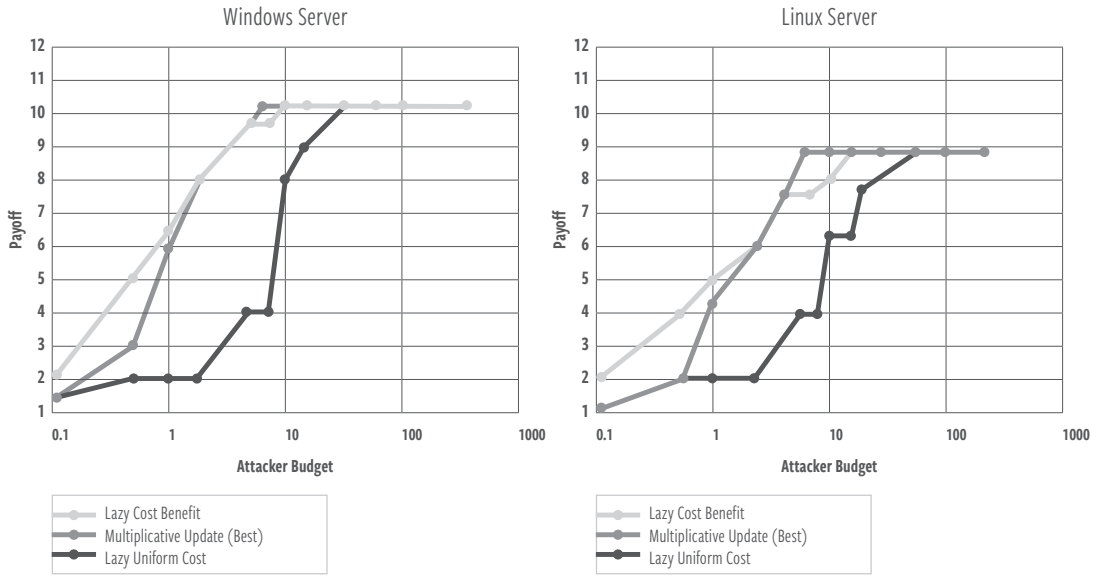

Figure 6.3: DHAP Payoff vs Budget

DHAP Results. We implemented both the greedy and multiplicative update approaches to the DHAP problem. For the greedy algorithm, we studied three variants of greedy (cost-benefit, uniform cost, and combination of the two) while we varied the parameter ϵ for the multiplicative update approach. We examined attacker payoff as a function of budget (in Bitcoin). Figure 6.3 displays this result. Though the cost-benefit greedy algorithm has the potential to perform poorly, it was, in general, the best performing approach—despite the multiplicative update approach achieving the better approximation guarantee. Further, the multiplicative update algorithm (Algorithm 2) was consistently the slowest in terms of runtime, taking much longer than the lazy greedy algorithms, particularly for high values of k_{atk}. Despite the multiplicative update algorithm having a better theoretical approximation ratio when compared to the tandem of greedy algorithms, namely $(1-\epsilon)(1-\frac{1}{e})$ compared to $\frac{1}{2}(1-\frac{1}{e})$, we see in Figure 6.3 that the greedy algorithms performed as well as or better than the multiplicative update very consistently. In all algorithms, as expected, runtime grew with budget (not pictured)—though the relationship was not strict, as an increase in budget does not necessarily mean that more exploits will be selected. In our experiments (on a commodity computer equipped with a 3.49 GHz i7 CPU and 16 GB of memory), our runtimes never exceeded ten minutes.

DHDP Results. Figure 6.4 demonstrate a defender's best response to an attack strategy (generated by DHAP) against a Windows Server and Linux Server, respectively, for varying values of k_{atk}. Though we see similar trends in Figure 6.4 as we do in Figure 6.3, we see that the payoff is generally lower, meaning that the defender can lower the expected payoff by enacting a best response strategy to an attack strategy produced by DHAP—which in our framework translates to fewer exploited vulnerabilities.

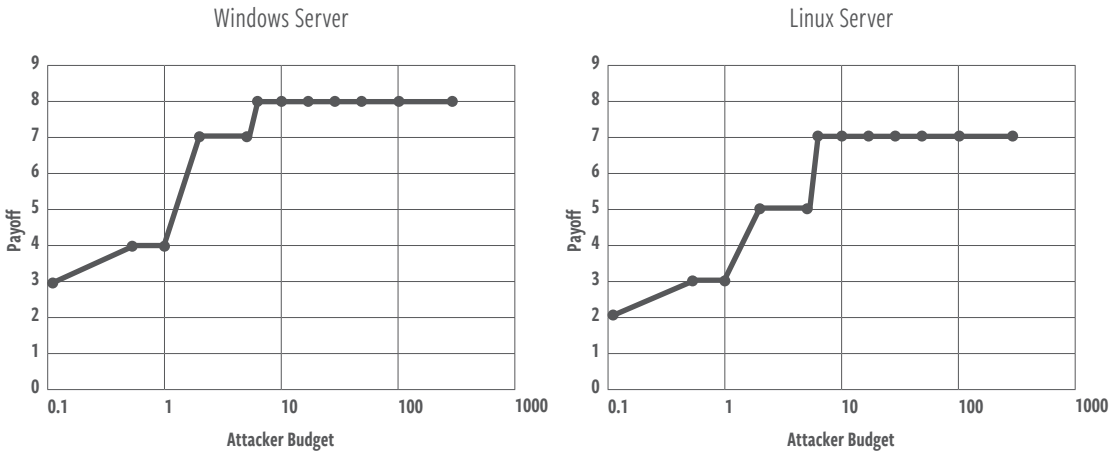

Figure 6.4: Defender Best Response, Payoff vs k_{atk}

Exploit Payoff Analysis. Instead of altering the software that appears on the host system to avoid exploits, such as in the best response approach, in exploit payoff analysis, the defender will identify which specific exploits are increasing the payoff the most. The hope being that the defender can reverse-engineer the exploit, or patch the vulnerability himself. To identify which exploits should be reverse-engineered, the defender first runs DHAP against his host system to identify what payoff an attacker could expect to produce. Then, for each exploit ex, the defender reruns DHAP against the host with the set of exploits $Ex\backslash\{ex\}$. The exploit ex that, when removed from the universe of exploits Ex, produces the largest drop in payoff for the attacker is the exploit that the defender should attempt to reverse-engineer. More formally, let A be the attack strategy produced by DHAP when using Ex as the universe of exploits and let A_{ex} be the attack strategy that is produced when DHAP is run against the host when using $Ex\backslash\{ex\}$ as the universe of exploits. The defender will attempt to reverse-engineer the exploit $ex = argmax_{ex \in Ex}\, p(A, D) - p(A_{ex}, D)$, where D is the defense strategy representing the host. To account for exploits that, though they greatly reduce payoff when removed from Ex, may be too expensive for the defender to purchase, we also consider a cost-benefit analysis, where the decrease in payoff is normalized by the cost of the exploit (i.e. $ex = argmax_{ex \in Ex} \frac{p(A,D) - p(A_{ex},D)}{C_{ex}}$).* The top exploits to reverse-engineer to defend a Windows Server host when considering an attacker budget of $k_{atk} = 5$, are shown in Table 6.5 with columns for both maximum payoff reduction and maximum cost-benefit analysis.

Exploit	Payoff Reduction	Max Cost-Benefit	Exploit Cost (BTC)
SMTP Mail Cracker	1	4.757	0.2102
SUPEE-5433	1	1.190	0.8404
Hack ICQ	1	79.089	0.01264
Plasma	0.6677	1.582	0.2563
WordPress Exploiter	0.6677	2.6467	0.2102
CVE-2014-0160	0.6677	3.178	0.2101

Table 6.5: Defender Exploit Analysis for $k_{atk} = 5$

* (i.e. $ex = argmax_{ex \in Ex} \frac{p(A,D) - p(A_{ex},D)}{C_{ex}}$)

7. CONCLUSION AND FUTURE WORK

We detailed a data collection system for gathering information from Darknet exploit markets and hacker forums. Additionally, we defined a game theoretic framework with which we can analyze the Darknet data, providing system-specific policy recommendations to system administrators. For the framework, we formalized decision problems for both the attacker and the defender, subsequently proving complexity results and providing approximation algorithms for each problem. We also evaluated the framework on a real-world dataset gathered from the previously discussed exploit markets.

In future work, we plan to extend the game-theoretic framework to include non-deterministic problem formulations, and construct algorithms to generate mixed strategies for the attacker and defender. By extending the exploit function in the framework, we plan to support blended threats, where the number of vulnerabilities affected by a cyber-attack is a superset of the union of the vulnerabilities affected by each individual exploit (i.e. $ExF(A) \supseteq \bigcup_{a \in A} ExF(\{a\})$). Additionally, we want to closely integrate the game theory framework with the crawling and parser infrastructure to provide system policy recommendations based on real-time data. We are continually adding support for additional Darknet sites in our scraping pipeline to gain a better understanding of the cyber threat landscape.

ACKNOWLEDGEMENTS

Some of the authors of this work were supported by the U.S. Navy, Office of Naval Research, NEPTUNE program as well as the Arizona State University Global Security Initiative (GSI), and the ASU Institute for Social Science Research (ISSR). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research. We thank Anum Naveed, of IntelliSpyre, Inc. for his feedback on this paper.

REFERENCES

L. Ablon, M. C. Libicki, and A. A. Golay. *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar.* Rand Corporation, 2014.

Y. Azar and I. Gamzu. Efficient submodular function maximization under linear packing constraints. *ICALP,* 1:38–50, 2012.

S. Chakrabarti, M. Van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks,* 31(11):1623–1640, 1999.

H. Chen. 2011. Dark web: Exploring and data mining the dark side of the web. Vol. 30. Springer Science & Business Media.

R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13,* SSYM'04, pages 21–21, 2004.

U. Feige. A threshold of ln n for approximating set cover. *J. ACM,* 45(4):634–652, July 1998.

T. Fu, A. Abbasi, and H. Chen. A focused crawler for dark web forums. *Journal of the American Society for Information Science and Technology,* 61(6):1213–1231, 2010.

M. Jain, Dmytro Korzhyk, Ondˇrej Vanek, Vincent Conitzer, Michal P ˇ echou ˇ cek, and Milind Tambe. 2011. A double oracle algorithm for zero-sum security games on graphs. In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 327–334.

D. Lacey and P. M. Salmon. It's dark in there: Using systems analysis to investigate trust and engagement in dark web forums. In D. Harris, editor, *Engineering Psychology and Cognitive Ergonomics,* volume 9174 of *Lecture Notes in Computer Science,* pages 117–128. Springer International Publishing, 2015.

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining,* pages 420–429. ACM, 2007.

F. Menczer, G. Pant, and P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology (TOIT),* 4(4):378–419, 2004.

M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In J. Stoer, editor, *Optimization Techniques,* volume 7 *of Lecture Notes in Control and Information Sciences,* pages 234–243. Springer Berlin Heidelberg, 1978.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming,* 14(1):265–294, 1978.

E. Nunes et al., Darknet and Deepnet Mining for Proactive Cybersecurity Threat Intelligence. IEEE Conference on Intelligence and Security Informatics (ISI-16), 2016.

J. Robertson, V. Paliath, J. Shakarian, A. Thart, and P. Shakarian. Data driven game theoretic cyber threat mitigation: Twenty-eighth aaai conference on innovative applications of artificial intelligence, 2016.

P. Shakarian and J. Shakarian. Considerations for the development of threat prediction in the cyber domain.*AAAI Workshop on Artificial Intelligence for Cyber Security (AICS),* 2016.

P. Shakarian, J. Shakarian, and A. Ruef. *Introduction to cyber-warfare: A multidisciplinary approach.* Elsevier, 2013.

K. Soska, and N. Christin. "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem." *24th USENIX Security Symposium (USENIX Security 15)*. 2015.

Milind Tambe. 2011. Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned (1st ed.). Cambridge University Press, New York, NY, USA.

Wei, Wang. "Hunting Russian Malware Author Behind Phoenix Exploit Kit". The Hacker News. 2013. http://thehackernews.com/2013/04/hunting-russian-malware-author-behind.html

Ziming Zhao, Gail-Joon Ahn, Hongxin Hu, and Deepinder Mahi. 2012. SocialImpact: Systematic Analysis of Underground Social Dynamics. In ESORICS (Lecture Notes in Computer Science), Sara Foresti, Moti Yung, and Fabio Martinelli (Eds.), Vol. 7459. Springer, 877–894. http://dblp.uni-trier.de/db/conf/esorics/esorics2012.html#ZhaoAHM12

NOTES

1. Corresponding author: shak@asu.edu.
2. See the Tor Project's official website (https://www.torproject.org/).
3. https://www.fireeye.com/blog/threat-research/2015/06/evolution of _dridex.
4. http://www.nbcnews.com/news/world/skilled-cheap-russian-hackers-power-americancybercrime-n22371.

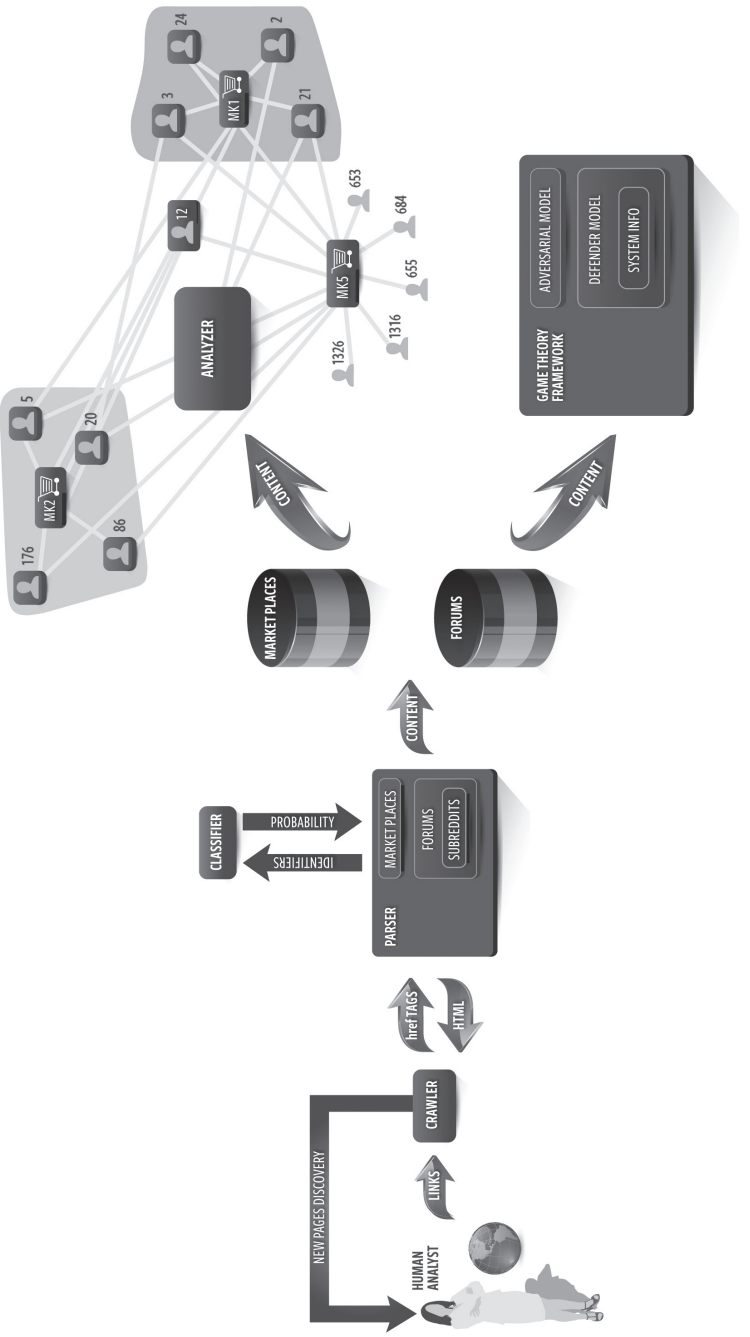

Figure 4.5: System Overview (Enlarged to show detail.)