Unified Profiling of Attackers via Domain Modeling

Nesrin Hussein*, Wentao Wang*, Joseph L. Nedelec†, Xuetao Wei‡, and Nan Niu*
* Department of Electrical Engineering and Computing Systems, University of Cincinnati, USA
† School of Criminal Justice, University of Cincinnati, USA
‡ School of Information Technology, University of Cincinnati, USA
{husseinm, wang2wt}@mail.uc.edu, {nedelejh, weix2}@ucmail.uc.edu, nan.niu@uc.edu

***Abstract*—In this position paper, we present our interdisciplinary research into a unified account of profiling attackers for software-intensive systems. Our work draws on the principles from requirements engineering and criminology. Specifically, we show how a unified crime theory can be adapted to model the attackers and their degree of knowledge about the environment in which the software operates. We illustrate our approach through an example based on i^* and data flow modeling, and discuss future research directions indicated by our preliminary results.**

***Index Terms*—Requirements engineering, criminal profiling, stakeholder and domain modeling, i^*, data flow diagrams.**

I. Introduction

Engineering secure software, especially the software-intensive system that has the capability to defend itself against criminal attacks, requires a thorough understanding of the attackers and their behaviors. We use the term *attacker* to refer to a single offender or a group committing the crime. Modeling stakeholders such as the attackers is a core activity of requirements engineering (RE).

Traditional security RE approaches like fault tree, threat tree, or obstacle analyses lack direct modeling of attackers and their capabilities in terms of operations they can perform and objects they can monitor and/or control. As a result, one cannot adequately reason about the attackers.

As goal-oriented RE focuses explicitly on the objectives the system under consideration should achieve, researchers began to model attackers and their goals. For example, van Lamsweerde [1] extended the KAOS framework to address malicious obstacles (called anti-goals) set up by attackers to threaten security goals. Refining the anti-goals led to the software vulnerabilities observable by the attacker, which further led new security requirements to be obtained as countermeasures.

Another thread of research builds on the i^* modeling framework. Liu and her colleagues, for example, introduced Secure-i^* for dealing with security and privacy requirements based on the concept of strategic social actors [2]. Secure-i^* used role-based modeling to study dependency patterns, such as trust and attacker-defender relations, thereby supporting the decision making about the appropriate policies, procedures, and mechanisms to achieve the desired levels of security within the organizational context.

The research endeavors so far have modeled attackers in a fragmented fashion, meaning each method has its unique way of identifying the attackers and their attacks. Without a theoretical basis connecting the principles behind the various approaches, it is difficult to translate the success of a method in one context to the design of other or new methods for different contexts. The lack of a foundational theory therefore presents a barrier to generalizable progress in the modeling of attackers to better support security RE.

In this position paper, we present our multidisciplinary research into a unified account of profiling attackers for software-intensive systems. Our work combines foundational knowledge from RE [3] with emerging theory from crime science [4]. In particular, we adapt an explanatory perspective for unifying the origins of criminal behavior among human beings [4] to qualify the degree of knowledge that an attacker has about the environment in which the software operates [3]. We illustrate our approach through an example based on i^* and data flow modeling.

II. Background and Related Work

RE shifts decision making on security from an afterthought to a strategic consideration. Security requirements are those related to the protection of the system's assets against malicious behaviors. Prior research has focused on eliciting, prioritizing, and reasoning about these requirements [5, 6, 7, 8, 9]. The desiderata of security requirements include being explicit, precise, adequate, measurable, complete, and non-conflicting with other requirements [5, 7]. As can be seen from the above list, security is treated more as a desired system quality than a concern from the standpoint of attacking.

A couple of approaches extend use cases to model attacks and adversaries. Misuse cases invert the normal functionalities of the system to express the malicious behavior [10], whereas abuse cases analyze a type of interaction between a system and actors where the results of the interaction are harmful to the system [11]. As with the general use case modeling, these approaches emphasize system functionalities (the "what") rather than stakeholder goals (the "why").

Goal-oriented approaches describe systems as intentional agents that depend on or compete with each other to achieve their goals. A goal, then, is a prescriptive statement of intent about the system whose satisfaction in general requires the cooperation of some of the agents forming that system [1]. To consider attackers, van Lamsweerde [1] constructed their anti-goals. Through the refinement of the anti-goals, one could derive observable vulnerabilities and/or implementable threats. Similar analysis capabilities, though built on i^* instead of

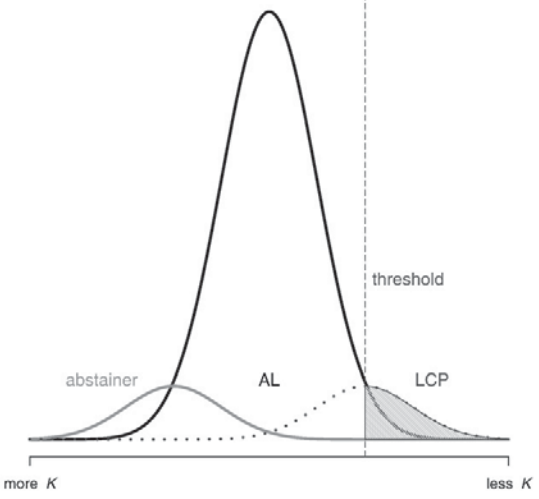

Fig. 1. A unified *Differential-K* theory where three developmental trajectories are plotted: abstainer, adolescence-limited (AL) offenders, and life-course-persistent (LCP) offenders (*adopted from [4]*).

KAOS, are provided by Secure-i^* where attacker analysis helps identify potential system abusers and their malicious intents [2].

While approaches like [1] and [2] offer insights into the security trade-offs, the modeling of the attackers extends existing goal-oriented frameworks like KAOS and i^*. In other words, what is being leveraged in the goal-elaboration methods are goals of the attackers but not the attackers themselves. To achieve an in-depth understanding, we argue that an interdisciplinary endeavor involving fields like criminology and crime science is indispensable.

Recent work by Dehghanniri *et al.* [9] used crime script to model the attack process as well as to describe the effect of the identified resolution actions as related to the attacker's activities. A crime script represents the complete sequence of actions adopted prior to, during, and following crime commission. It is the systematic and detailed nature that makes crime script a valuable tool to improve the understanding about how certain types of crime occur and hence to improve the security decision-making under uncertainty [9]. A related project allowed multiple attacking scenarios to be simulated so as to assess the attack impact without imposing extensive time or cost constraints [12]. The modeling takes into account the different types of threat (e.g., biological and chemical) that may occur within or around the target infrastructure.

In pursuit of a single overarching perspective of the origins of criminal behaviors, Boutwell *et al.* [4] presented a unified theory capable of explaining different patterns of criminal offending both at the individual level as well as the macro-level. The theory proposed in [4] argues that many types of crime can be understood in the evolutionary context of human life history. Specifically, several universal correlates of crime that are widely recognized by criminologists (e.g., race, age, sex, family size, family structure, socioeconomic status, and urban residency) could be treated as traits of a single *Differential-K* theory [13].

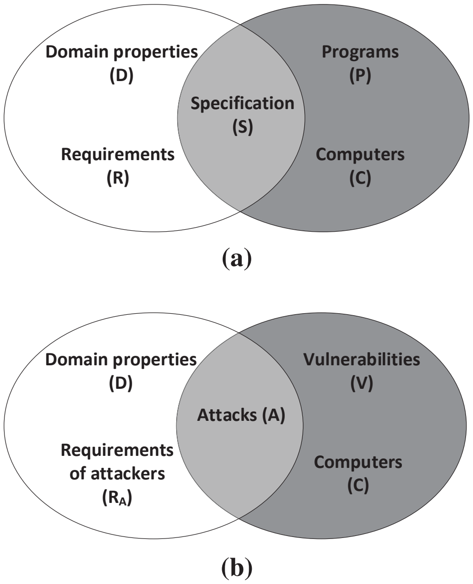

Fig. 2. Building and extending on Jackson's work **(a)** [3], we situate the attacker requirements in a similar environment-machine conceptualization **(b)**.

Fig. 1 shows Boutwell *et al.*'s theoretical conceptualization [4] where the hypothesized distributions of the density of life-history traits across three offending groups are plotted: abstainer, adolescence-limited (AL) offenders, and life-course-persistent (LCP) offenders. The x-axis of Fig. 1 offers a *unifying* account to index the life history spectrum using K. Evolutionarily speaking, K references the "carrying capacity" of an organism's environment in the sense that as individual organisms begin to rapidly deplete resources in a given area, selection pressures favor the genes corresponding to life history strategies marked by slower maturation and lower fecundity [4]. The traits that fall further from K (e.g., increased difficulty with emotional and behavioral regulation) thus appear far-right of the x-axis in Fig. 1.

The threshold shown in Fig. 1 could be considered an arrest for violence or some other severely negative life outcome [4]. Consequently, only the ALs and the LCPs have a portion of their density that crosses the threshold according to Fig. 1. It is the appeal to unification presented in [4] that motivates us to explore a way underlying the profiling of attackers in the context of security RE. Next we discuss how we build the basic analogies, and more importantly derive the differences, of the evolutionary theory [4].

III. Toward a Unified Profiling of Attackers

In his seminal work on the "meaning of requirements", Jackson [3] laid out some RE fundamentals as shown in Fig. 2a. While stakeholder requirements (R) are located in the environment with relevant domain properties expressed as D, the specification (S) serves as the bridge to connect the environment with the machine/software. RE, then, in its simplest form, shall elicit R and derive S such that "D, S |= R". Furthermore, "P, C |= S" holds in Fig. 2a, making S the most important phenomena shared between requirements engineers and software developers.

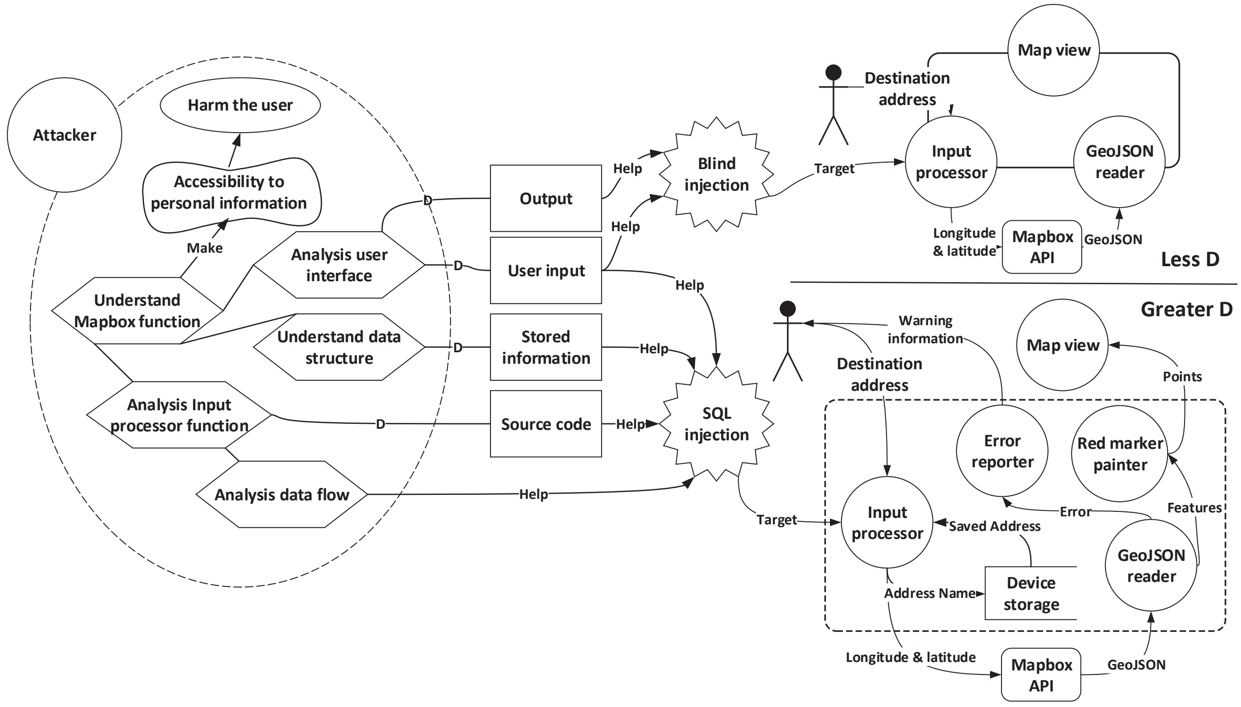

Fig. 3. The left part of the model uses i^* strategic rationale view to model the requirements of an attacker ($\mathrm{R_A}$), the middle part of the model shows the resource-based dependencies connecting $\mathrm{R_A}$ with the two attacks (blind injection and SQL injection), and the right part of the model expresses different degrees of domain knowledge (D) in terms of data flow diagrams.

We build on Jackson's work to conceptualize the attacker requirements ($\mathrm{R_A}$) in Fig. 2b. The bridge here is the set of attacks (A) employed by the attacker. Following [3], we have “D, A $|=$ $\mathrm{R_A}$”. Additionally, “V, C $|=$ A” holds in Fig. 2b, which indicates that the exploited vulnerabilities (V) combined with the computers or other computing devices (C) would entail the success of attacks (A). Note that the concepts presented in Fig. 2b are by no means complete; related are assets, risks, obstacles, countermeasures, etc. Nevertheless, these concepts allow us to adequately map Boutwell *et al.*'s theory to security RE.

Key to our mapping is D, i.e., the domain properties that the attacker uses to structure the attacks. We hypothesize that the degree of knowledge that the attacker has about the environment will be reflected in D — the more advanced understanding D is, the more likely the attacker's attack (A) is successful. The relationship between D and A can then be understood in a similar way as the unified theory shown in Fig. 1. We illustrate our position through a worked example of Mapbox[1].

Fig. 3 shows our preliminary results. In this figure, the attacker's requirements are modeled through a strategic rationale view of i^*. It is important to note that the attacker's high-level goal of accessing/stealing a benign user's personal information is fixed. However, different attackers have different levels of domain knowledge (D). Such a variation is represented by the “less D” and “greater D” divide in our example. The dependencies of the attacker and the domain knowledge are established through different resources shown in the middle of Fig. 3. The “less D” situation mimics a black-box attack (i.e., blind injection) whereas “greater D” resembles a white-box attack (i.e., SQL injection).

In addition to injection, we have developed the denial of service (DoS) attacks profiled into “less D” and “greater D” categories. Due to space constraints, the DoS attacks are not shown here. Our main objective, however, is to test the extent to which our hypothesis is sensible. To that end, we performed a manual analysis of the Common Vulnerabilities and Exposures (CVE) repository[2]. Our search covered the CVE records since Jan 1, 2015 that were of injection nature. We further extracted the attackers and the actual attacks reported to CVE. For a valid attack, a CVE ID was created; otherwise, the attack was considered unsuccessful. Table I shows our collected data.

For each of the seven attacks listed in Table I, we manually analyzed and classified the domain knowledge exploited. Assuming every knowledge source ranging from (a) to (f) receives the same weight, the column “D value” of Table I indicates the level of domain knowledge involved in the attack. Mapping the D of Table I to the K of Fig. 1 allows us to plot the seven attacks in Fig. 4. Similar to Fig. 1, a threshold is drawn in Fig. 4 to signal the divide between successful and unsuccessful attacks.

[1] The Mapbox project was created in 2010 to help offer map customization for non-profit customers. It raised over $60 million funding as of June 2015. Please see https://en.wikipedia.org/wiki/Mapbox for more information.

[2] http://cve.mitre.org

TABLE I
REPORTED INJECTION ATTACKS FROM CVE (JAN 1, 2015 – JUNE 27, 2016) WHERE THE ATTACKER INFORMATION IS EXPLICIT

ID	Attack (A)	Attacker	Domain knowledge (D)[1]	D value	Report date (CVE ID[2])
①	Blind injection in WordPress	Larry Cushdoller	(a)	1	Nov 09, 2015 (--)
②	SQL injection 1 in Cacti	Paul Gevers	(b), (c)	2	Nov 09, 2015 (--)
③	SQL injection 2 in Cacti	Paul Gevers	(b), (c)	2	Jan 4, 2016 (2016-2313)
④	SQL injection 3 in Cacti	Paul Gevers	(b), (c)	2	Mar 10, 2016 (2016-3172)
⑤	SQL injection in WordPress	Larry Cushdoller	(b), (c), (d)	3	Jun 21, 2015 (2015-4694)
⑥	Command injection in WordPress	Larry Cushdoller	(a), (c), (e)	3	Dec 2, 2015 (2015-7527)
⑦	SQL injection 4 in Cacti	Xin Wang	(a), (b), (c), (f)	4	Jun 9, 2015 (2015-4342)

[1] (a) Http header injection, (b) SQL injection, (c) PHP source code logic, (d) Project knowledge, (e) Command injection, (f) Database structure

[2] Only a validated attack receives a CVE ID. Attacks ① and ② were reported but not successful.

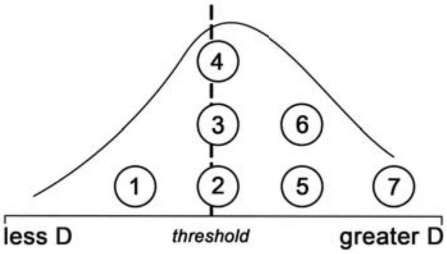

Fig. 4. Plotting the seven attacks of Table I on a unified D dimension.

IV. CONCLUDING REMARKS

The power of a theory lies in its ability to *unify* multiple situations under a common abstraction. In this paper, we have demonstrated how an evolutionary crime theory [4] can be adapted to help understand attacker profiles in security RE. While our experience suggests that resources like data flows should be treated as important assets that may need to equip themselves with defending or at least logging capabilities, we list below some questions to be researched in the future.

- How to (better) instantiate the x-axis of Fig. 4? We currently rest on the diversity of D but simply treat each kind of D equally. Adjusting D weights may offer better theoretic fit as well as insights into D's interplay.
- How to (better) instantiate the y-axis of Fig. 4? While we follow the density distributions hypothesized by Boutwell *et al.* [4], other options such as frequency and/or severity of attacks are worth exploring.
- Do different attacker profiles exist? Compared to the three profiles in Fig. 1, Fig. 4 shows only one. Mining more data (different types of attacks, unsuccessful attacks, etc.) could potentially lead to diversified attacker profiles.
- How to tackle dynamic and evolving factors? The range of applicability of the proposed theory remains to be determined. For example, can the attacker be simultaneously a benign user, evolve the domain knowledge, and change the goals?

ACKNOWLEDGEMENT

The work is funded in part by the U.S. National Science Foundation (CCF 1350487) and the National Natural Science Foundation of China (No. 61375053).

REFERENCES

[1] A. van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *International Conference on Software Engineering (ICSE)*, Edinburgh, UK, May 2004, pp. 148–157.

[2] L. Liu, E. Yu, and J. Mylopoulos, "Secure-i^*: engineering secure software systems through social analysis," *International Journal of Software and Informatics*, vol. 3, no. 1, pp. 89–120, March 2009.

[3] M. Jackson, "The meaning of requirements," *Annals of Software Engineering*, vol. 3, no. 1, pp. 5–21, January 1997.

[4] B. B. Boutwell, J. C. Barnes, K. M. Beaver, R. D. Haynes, J. L. Nedelec, and C. L. Gibson, "A unified crime theory: the evolutionary taxonomy," *Aggression and Violent Behavior*, vol. 25, pp. 343–353, November-December 2015.

[5] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Modeling security requirements through ownership, permission and delegation," in *International Requirements Engineering Conference (RE)*, Paris, France, August-September 2005, pp. 167–176.

[6] N. Niu and S. Easterbrook, "Analysis of early aspects in requirements goal models: a concept-driven approach," *Transactions on Aspect-Oriented Software Development*, vol. III, pp. 40–72, 2007.

[7] C. B. Haley, R. C. Laney, J. D. Moffett, and B. Nuseibeh, "Security requirements engineering: a framework for representation and analysis," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 133–153, January 2008.

[8] G. Cantrell, D. A. Dampier, Y. S. Dandass, N. Niu, and A. C. Bogen, "Research toward a partially-automated, and crime specific digital triage process model," *Computer and Information Science*, vol. 5, no. 2, pp. 29–38, March 2012.

[9] H. Dehghanniri, E. Letier, and H. Borrion, "Improving security decision under uncertainty: a multidisciplinary approach," in *International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, London, UK, June 2015, pp. 1–7.

[10] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, January 2005.

[11] J. P. McDermott and C. Fox, "Using abuse case models for security requirements analysis," in *Annual Computer Security Applications Conference (ACSAC)*, Scottsdale, AZ, USA, December 1999, pp. 55–64.

[12] T. L. Sage, H. Borrion, and S. Toubaline, "A user-layered approach for modelling and simulating terrorist attacks," *International Journal of Critical Infrastructures*, vol. 10, no. 3-4, pp. 398–412, 2014.

[13] J. P. Rushton, *Race, Evolution, and Behavior: A Life History Perspective*. Charles Darwin Research Institute, 2000.