In Cyber-Space no one can hear you S·CREAM:

A Root Cause Analysis for Socio-Technical Security

Ana Ferreira[1], Jean-Louis Huynen[2], Vincent Koenig[2](†), and Gabriele Lenzini[2]

[1] University of Porto, CINTESIS
[2] University of Luxembourg, SnT and (†) COSA.

Abstract. Inspired by the root cause analysis techniques that in the field of safety research and practice help investigators understand the reasons of an incident, this paper investigates the use of root cause analysis in security. We aim at providing a systematic method for the security analyst to identify the socio-technical attack modes that can potentially endanger a system's security.

Keywords: root cause analysis, security analysis, socio-technical security

1 Introduction

Accounting for the impact of a user in a system's security incident is a complex matter. In safety, this impact is usually studied by applying *Human Reliability Analysis* to predict how reliable a system is or *Root Cause Analysis (RCA)* to understand the reasons of an incident where humans are involved. These techniques are not practised in security, but we argue that they should. They would help understanding why security fails in the presence of humans. However, applying RCA in security is not straightforward, it needs some adjustments.

The primary cause of any security incident is unsurprisingly the attacker. But the success of an attacker's Socio-Technical Attacks —attacks that rely at least partially on the presence of human users— also depends on the system, the user, and the context: users can err and create security failures by executing *security critical actions* (e.g., clicking on an infected attachment), whereas human factors (e.g., carelessness), usability problems, and disturbances in the human environment (e.g., noise, psychological pressure) catalyse such situations. These untangled factors are pre-conditions for the attacker's ability to trigger what is often dismissed as "human errors". But if this is the conclusion of a security analysis, no one would know how to secure the system except by extruding completely the user. This drastic solution is obviously severely limited. In the field of safety-critical systems instead, a "human error" is not a conclusion but a start, a symptom of further underlying causes that calls for investigating its "root causes". This is what an insightful security analysis should also do. To this concern we hypothesize that cascades of events that lead to the success of user-mediated attacks are comparable to the ones studied in safety and that it is possible, in a socio-technical analysis of security, to retrieve a root cause more informative than the mere "human error".

Contribution. Inspired by the RCA techniques that are used in the field of safety research and practice, we devise a method to compile a *catalog of Socio-Technical Attack Modes (AMs)*.[3] These are events injected by the attacker that may drive the user to err (e.g., trusting a malicious link) while executing a critical action (e.g., clicking on a link) and initiating a cascade of steps eventually ending with the attacker harming a system's security. AMs actually exploit Error Modes (i.e., ways to err) and reveal the complex interplay among the user, the system, and the context.

The catalog we present could serve (a) to analyze the user-system interactions in search for patterns that are known to trigger Error Modes and eventually harm the system's security, and (b) to identify realistic vulnerabilities of socio-technical nature in those interactions under an extended threat model that accounts for the effect of the intruder's action on the user. From the attacker's capabilities one can determine what effect s/he has on the system and consequently what controls can be applied in defence.

Overall, this paper answers two research questions. Does applying RCA give original insights into the cause of success of existing attacks? Can we find new attacks thanks to RCA techniques? Given the space constraints of this short paper, we can only sum up how we customized a RCA technique to build a small catalog of Attack Modes. A description of the whole methodology and the compilation of a comprehensive catalog of attacks will be developed in an extended version of this paper.

Related Works. The most relevant works related to this paper's objectives are: Cranor *et al.* work on security-related communications [1], Curzon *et al.*'s *Cognitive Framework* [2], and Carlos et al. [3] proposal of a taxonomy of human-protocol weaknesses. They all discuss the role of users in security and, from different perspectives, explain how human features may affect security. Our work can be seen as a re-elaboration of those discussions, extended and integrated in our methodology of analysis for the search of root cause analysis in security.

2 Methods

Our methodology combines socio-technical security analysis with Root Cause Analysis (RCA) as inspired from the safety field. First we select and adapt a RCA technique for security; then we use this technique to build a catalog of generic *(socio-technical) Attack Modes (AMs)* observed in actual attacks. We draw our sample of actual attacks from the CAPEC [4] attack library. We use the adapted RCA to explain the success of known attacks and to check if it covers explanations from the literature. We also augment our explanation with the causes brought up by the analysis of root causes.

[3] For improved readability, we do not spell out 'socio-technical' in the following while it has to be systematically assumed

Selecting an RCA technique and adapting it for security. We selected Cognitive Reliability and Error Analysis Method (CREAM) [5] as our preferred RCA. CREAM is a 2nd generation Human Reliability Analysis; it focuses on errors whereas more recent techniques consider human performance as a continuum [6]. By considering cognitive causes of errors, CREAM brings a great deal of details in the analysis of an accident and because of such richness in details it has been criticized in Human Reliability Analysis [7]. However, such richness is what makes CREAM a great candidate for computer security: a security analysis should identify all factors that an attacker can use to push a human to err. Among other criteria, the most important aspect of CREAM is that it offers retrospective and prospective analysis. Thus, it provides us with bi-directional links between causes and effects. This allows us to build a catalog of AMs that can be used in both ways: in detecting attacks (starting from observed effects) and in predicting attacks (starting from a threat model).

CREAM relies on two pillars: (1) a classification of erroneous actions (this is represented in tables linked together by causal relationships), and (2) a method that describes how to follow those links back to the human, the contextual and the technological factors at the origin of an "event". An event is caused by the manifestation of an "erroneous action", and is called the phenotype [5]. The confluence of underlying factors that made the erroneous action arise is called its genotype. CREAM's tables of causal relationships between antecedent (cause of errors) and consequent (effect of errors) link a phenotype with its genotype [5]. Following these causal relationships, it is possible to find what caused an erroneous action and the root cause(s) of an event.

CREAM is a building block of our method, but it needs to be customized for security. We call the result S·CREAM, which stands for "Security CREAM".

Applying the RCA and building a catalog of AMs. We apply our RCA to build a catalog of AMs. We take as input a library of known attack patterns which we got from Common Attack Pattern Enumeration and Classification (CAPEC) [4]. This library contains attacks "generated from in-depth analysis of specific real-world exploit examples".[4] It is maintained by MITRE Coorporation, and it is the only detailed classification scheme where attacks centered on the user are compiled and documented. We use CAPEC's repository to extract and select those Attack Patterns whose success relies on a critical action of the user. The CAPEC taxonomy contains descriptions of social-engineering Attack Patterns, together with their pre-requisites, mechanisms and possible mitigations.

3 S·CREAM: An RCA for computer security

We describe S·CREAM, the technique we devise by customizing CREAM and that we propose as the way to identify root causes of socio-technical attacks.

[4] See https://capec.mitre.org/

3.1 Adapting CREAM as an RCA technique for security

S·CREAM's retrospective analysis draws on CREAM, but it needed adaptations because of our computer security focus.

In CREAM's retrospective analysis, one first defines common performance conditions to describe the analyzed event, then the Error Modes to investigate. This investigation is a process where the analyst searches for the antecedents of each Error Mode. This process is recursive: each antecedent an analyst finds can be investigated in turn. Antecedents justified by other antecedents are called "generic"; those which are "sufficient in themselves" are called "specific". To avoid following "generic antecedents" endlessly, one must stop the investigation on the current branch when a "specific antecedent" is found to be the most likely cause of the event.

The computer security context in which we intend to use CREAM's retrospective analysis calls for a different procedure because of three main singularities that make it peculiar: (a) we already know that an attacker is the initiator of the cascade of events and what message s/he has sent the user, (b) erroneous actions are already defined, and (c) we lack contextual information as we operate from a generic description of Attack Patterns. Two adaptations to CREAM's retrospective analysis methods are therefore needed. First, we customize the phase preceding the investigation: instead of formalizing the context in common performance conditions, S·CREAM uses its own description of the event focusing on the information flowing between the attacker and the user; we describe this part in the next section. Second, S·CREAM uses a less restrictive stop rule. Doing so we avoid pointing invariably to the attacker's action, and we investigate additional contributing antecedents. So, where CREAM stops as soon as a specific antecedent is found being a likely cause of the event, S·CREAM lists all likely specific antecedents for the event plus the specific antecedents that are contained into sibling generic antecedents, it then stops the investigation of the current branch.

To choose between the different possible antecedents that CREAM's tables propose, we look at the Attack Pattern's description that we built before the analysis and we stick to the attacker's actions we described.

3.2 Using S·CREAM

As stated previously, S·CREAM needs a description of the Attack Patterns under scrutiny. The most important aspect is that this description should enable us to choose objectively among the different paths possible through the antecedent-consequent links. To describe each Attack Pattern before the analysis, we follow what has been proposed in [8]: describing Attack Patterns as a set of messages flowing between the attacker and the victim prior to the manifestation of the critical action. Thus, this first event that initiates the attack is described through common properties shared by the messages sent from the attacker to the user. In synthesis: (1) a source, that is the principal that the user believes to be interacting with, (2) an identity split into a declared identity (i.e., who the attacker says he is, like the from field of an email) and imitated identity (i.e.,

who the attacker imitates by stealing a logo for instance), (3) a command for the user to execute, (4) an action description, to state for instance if the action is booby-trapped or spoofed, (5) a sequence that describes the temporal situation of the message, and (6) a medium (web, phone, paper).

Running S·CREAM on identified Attack Patterns. Once an Attack Pattern is described, we perform the S·CREAM analyses on the critical actions carried out by the victim (those with an effect on the system's security). We have at least one Error Mode for each Attack Pattern. Additional Error Modes may have to be analysed in the course of events that lead to the critical action, for instance when the victim first encounters the attacker and misidentifies him/her as being trustworthy.

From Error Modes to AMs. Attack Modes are ways to exploit Error Modes that stem from the interaction between the user, the system and the surrounding context. For instance, an AM can state that an attacker, who can send messages to the user of a system that displays ambiguous symbols may be able to usurp somebody else's identity (this is the effect on the system's security). AMs are readily usable links between Threat Models and possible security-harming effects enabled by particular user-system interactions.

4 Building the catalog of Attack Modes

To start we need to have a database of known existing Attack Patterns, and we chose to look for them in CAPEC. Applying S·CREAM is about reconstructing the chain of events that lead to harming a system's security. The results of the S·CREAM analyses unveil contributing factors to those events that, thanks to the bi-directional nature of CREAM's causation links, we could turn into a catalog of AMs. We identified 16 Attack Patterns out of CAPEC where the user is at the source of the success of the attack. For the sake of space, we only report on one Attack Pattern, the CAPEC-195 "Principal Spoofing". This Attack Pattern is not considered an issue from a sole technical point of view: its root cause mostly depends on the user's weaknesses and technical factors only increase its likelihood. We first detail how we translate this Attack Pattern into our framework, then we perform a S·CREAM analysis of its causes of success.

Translation of CAPEC-195 "Principal Spoofing" into S·CREAM. In the CAPEC-195 "Principal Spoofing" Attack Pattern, the attacker pretends to be one more actor in the interaction. This attack relies on the content of the message to appear that it reflects an honest identity. Its translation into our framework can be summed up by the following: (1) the source is another principal that the target knows, (2) the imitated identity is used because the appearance of the message is crafted to reflect the source's identity, (3) the command is not specified, (4) the attacker is the initiator of the non-spoofed action "disclose information" or "perform action on behalf of the attacker", (5) the message is a continuation of a previous interaction as the target must know the principal,

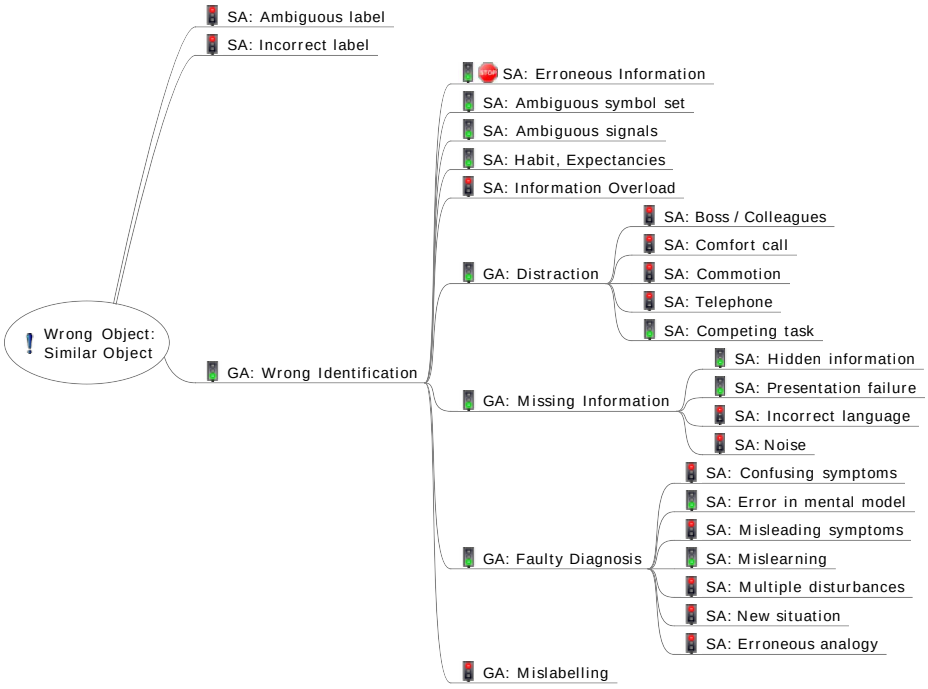

Fig. 1. Part of S·CREAM's investigation of the "Wrong Object" Error Mode (EM) observed in CAPEC-195. A green traffic light means that we consider the specific antecedent as being a contributor or that we expand the generic antecedent. A red traffic light means that we do not consider that the antecedent contributes to the EM. An additional stop sign means that we encountered a specific antecedent that is a probable cause in the current branch and that the stop rule is now engaged. Additional sibling generic antecedents of "Wrong identification" are not displayed and specific antecedents and generic antecedents are abbreviated as SA and GA.

Table 1. Justifications of the selections of contributors for the "Wrong identification" generic antecedent. Specific antecedents inside generic antecedents are not named and specific antecedents and generic antecedents are abbreviated as SA and GA.

Antecedent	Justification
SA "Ambiguous Signals and Symbols"	The usability of the interface can contribute to this Error Mode.
SA "Habit and Expectancies"	As the message sent by the attacker is a continuation from a previous interaction we can reasonably consider that this antecedent plays a role in the target's behavior.
GA "Distraction"	We don't have additional information regarding the SAs contained in this GA in our description. But it is likely that the user was performing a main task while assessing the identity of the attacker, so we consider SA "Competing task" as an additional contributor.
GA "Missing Information"	The attacker deliberately hides its real identity and the presentation fails to clearly state the sender identity. So we consider the corresponding SAs as contributors.
GA "Faulty Diagnosis"	The user may have a wrong mental model about how to assess identity, or misunderstood previous explanations.

and (6) the medium can be on screen or paper, in person or by phone ("either written, verbal, or visual").

Detailed Root Cause Analyses. In the following, the EM is analyzed using Serwy et al.'s [9] implementation of CREAM's tables. The main EM of this Attack Pattern is the misidentification of the attacker for another principal, we identify it as being a "Wrong object:Similar Object" EM. Figure 1 shows among the possible antecedents for this EM, which path we follow: as the declared identity is not used in this attack, the specific antecedents related to the labeling do not contribute to the behavior. Therefore, we continue the analysis by looking at the generic antecedents: following the generic antecedent "Wrong identification:Incorrect identification", the specific antecedent 'Erroneous information' is selected as contributor because the imitated identity is spoofed. This root cause provided by S·CREAM is the same as the explanation provided by CAPEC: the wrong information furnished by the attacker tricks the user. As shown in Table 1, we follow our custom stop rule and consider the other specific antecedents and sibling generic antecedents for this branch.

Following the analysis out of this branch, the next generic antecedent "Communication failure" leads to the specific antecedent "inattention". The last generic antecedent is "Observation missed" where the specific antecedent "multiple signal" is likely if the attack is run on a computerized medium.

Results. The analyses we performed with S·CREAM on the set of 16 Attack Patterns that we extracted from CAPEC yielded numerous antecedents; assuming that CREAM's bi-directional links of causation can be trusted, we consider that these antecedents can be exploited by an attacker to facilitate the occurrence of critical actions. For the sake of space, we only report on the AMs whose effect is to give the attacker the ability to *usurp another actor's identity*. Most of the following AMs were built by listing the specific antecedents resulting from the S·CREAM analysis of CAPEC-195: "Principal Spoofing" and CAPEC-194 "Fake source of data".

Table 2 compiles the AMs that an attacker with certain capabilities can use against a system working under specific conditions. Because those AMs are built from the data provided by CAPEC, there are no assumptions about the users.

5 Discussion

The example we develop in §4 shows that an RCA-based technique like S·CREAM improves our understanding of known attacks without the need of looking each time into the literature or performing user studies ourselves. We believe that in this regard, S·CREAM is an effective tool for security practitioners who want to investigate specific user-mediated attacks at a lesser cost. Furthermore, S·CREAM identifies contributors to the success of an attack that are not considered in taxonomies such as CAPEC.

The contribution of S·CREAM to research is also substantial. E.g., it is well known that the Wi-Fi selection process suffers from security issues bound to

Table 2. AMs an attacker can use to usurp an identity. AMs that only require the attacker to be able to send a message are not shown for the sake of space, namely: *Bad mental Models*, *Mislearning*, *Inattention*, and *Multiple Signals*.

Attack mode	Prerequisites	
	System	Attacker
Incorrect label	Amendable messages	Change sender's field in a message
Erroneous Information	Displayed information is not verified	Can send a message with falsified information or the visual identity of another actor
Ambiguous symbol	Bad usability, symbols are confusing	Can send a message that uses said symbols to convey misleading signals
Habits and Expectancies	Use is monotonous or repetitive	Can send/replay/mimic a message. Knows about previous user's interactions.
Competing task	Main task, or sub tasks	Can send a message
Hidden information	Provides amendable info	Can alter the information provided in messages
Presentation failure	Amendable interface	Can send a message that abuses presentation

the misconceptions that people have about the meaning of the different symbols used in the graphical user interface [10]. Where user-studies and surveys had been performed to investigate this problem from scratch, preliminary answers could have been readily obtained through an S·CREAM analysis to support the design of those studies. S·CREAM proposes an additional input allowing to triangulate findings and thus contributes to validating such findings. Such a combination of methods and data sources is an invaluable asset contributing to consolidating the relatively young field of socio-technical security research.

Even if the exploitation of psychological characteristics in socio-technical attacks arises (e.g., tabnabbing attack [11]), these remain mainly focused on the user's observation mistakes. S·CREAM and our catalog of AMs can help anticipating more advanced attacks because S·CREAM yields new means to trick users to err while performing security-critical actions; and our catalog lists the potential attacks that can be fomented. S·CREAM can help refining attacks, and adding information about the user and the context of the attack can open new doors. Of course, the sheer possibility of an AM in a system does not guarantee that an attack will happen, but it constitutes an additional entry point to the system's attack surface.

We think S·CREAM can be further specialized for computer security to bring more specific information about the threat model required for each AM. We can, for instance, map the concertina model of an existing framework for a socio-technical analysis (see [8]) into what CREAM categorizes as the "Man-Technology-Organization" triad, systematically linking where the attacker can strike in the concertina interaction layers.

6 Conclusion

We have illustrated how to adapt Root Cause Analysis (RCA), a technique used in safety to investigate the cause of "human errors", to security. The resulting technique, named S·CREAM, is a valuable tool to identify the factors that contribute to damage a system' security by inappropriate, security critical user

actions. We used S·CREAM to build an initial catalog of socio-technical Attack Modes (AMs). This is only a first step: there is a need to apply S·CREAM on more socio-technical attacks and attack patterns to improve the way it models them and the information it provides. We believe that the AMs catalog we started building will lead to define a more realistic threat model, that is, one that integrates user-mediated capabilities. Still, our catalog is preliminary. Expanding its scope is future work: we intend to list additional AMs by using results we could not report here. We also plan modifications to the S·CREAM method to guarantee more objectivity in the analysis process, and in support to that, the creation of a computer assisted tool to help the security analyst performing his/her tasks.

Acknowledgments

This research is supported by FNR Luxembourg, project I2R-APS-PFN-11STAS.

References

1. Cranor, L.F.: A framework for reasoning about the human in the loop. Communication (2008) 115
2. Curzon, P., Ruksenas, R., Blandford, A.: An approach to formal verification of humancomputer interaction. Formal Aspects of Computing **19**(4) (2007) 513–550
3. Carlos, M., Price, G.: Understanding the weaknesses of human-protocol interaction. In Blyth, J., Dietrich, S., Camp, L., eds.: Financial Cryptography and Data Security. Volume 7398 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 13–26
4. Corporation, M.: CAPEC - Common Attack Pattern Enumeration and Classification (2014) https://capec.mitre.org/.
5. Hollnagel, E.: Cognitive reliability and error analysis method CREAM. Elsevier, Oxford New York (1998)
6. Hollnagel, H.: FRAM:The Functional Resonance Analysis Method: Modelling Complex Socio-technical Systems. MPG Books Group (2012)
7. Cacciabue, P.C.: Guide to Applying Human Factors Methods - Human Error and Accident Management in Safety-Critical Systems. Springer (2004)
8. Ferreira, A., Huynen, J.L., Koenig, V., Lenzini, G.: A conceptual framework to study socio-technical security. In Tryfonas, T., Askoxylakis, I., eds.: Human Aspects of Information Security, Privacy, and Trust. Volume 8533 of Lecture Notes in Computer Science. Springer International Publishing (2014) 318–329
9. Serwy, R.D., Rantanen, E.M.: Evaluation of a software implementation of the cognitive reliability and error analysis method (CREAM). Proceedings of the Human Factors and Ergonomics Society Annual Meeting **51**(18) (oct 2007) 1249–1253
10. Ferreira, A., Huynen, J.L., Koenig, V., Lenzini, G., Rivas, S.: Do graphical cues effectively inform users? In Tryfonas, T., Askoxylakis, I., eds.: Human Aspects of Information Security, Privacy, and Trust. Volume 9190 of Lecture Notes in Computer Science. Springer International Publishing (2015) 323–334
11. Raskin, A.: Tabnabbing: A New Type of Phishing Attack http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/.