Madarie *et al. Crime Science* (2024) 13:27
https://doi.org/10.1186/s40163-024-00227-5

Crime Science

SYSTEMATIC REVIEW

Open Access

Introducing object-oriented modelling to cybercrime scripting: visualisation for improved analysis

Renushka Madarie[1,2*], Marleen Weulen Kranenbarg[1] and Christianne de Poot[1,2,3]

Abstract

Crime script analysis as a methodology to analyse criminal processes is underdeveloped. This is apparent from the various approaches in which scholars apply crime scripting and present their cybercrime scripts. The plethora of scripting methods raise significant concerns about the reliability and validity of these scripting studies. In this methodological paper, we demonstrate how object-oriented modelling (OOM) could address some of the currently identified methodological issues, thereby refining crime script analysis. More specifically, we suggest to visualise crime scripts using static and dynamic modelling with the Unified Modelling Language (UML) to harmonise cybercrime scripts without compromising their depth. Static models visualise objects in a system or process, their attributes and their relationships. Dynamic models visualise actions and interactions during a process. Creating these models in addition to the typical textual narrative could aid analysts to more systematically consider, organise and relate key aspects of crime scripts. In turn, this approach might, amongst others, facilitate alternative ways of identifying intervention measures, theorising about offender decision-making, and an improved shared understanding of the crime phenomenon analysed. We illustrate the application of these models with a phishing script.

Keywords Crime script analysis, Object-oriented modelling, Unified modelling language, Cybercrime, Offender decision-making, Activity diagram, Domain model, Visualisation

Background

Crime script analysis, or crime scripting, has been introduced by Cornish in 1994 as a method to examine crime commission processes. When conducting crime script analysis, researchers generally create narratives that detail, amongst others, what actions, actors, tools and locations are typically involved in a specific type of crime, and how these crime elements interact. By dissecting crimes through crime scripting, scripts could reveal, amongst others, opportunity structures and patterns in offender decision-making (Ekblom & Gill, 2016). Indeed, crime scripting is generally intended to create more insight into a crime phenomenon and inspire new and more effective intervention methods. Although crime scripting has gained traction as a method to analyse crime (Dehghanniri & Borrion, 2019), there appears to be little agreement on how crime scripting should be conducted and what crime scripts should look like. This lack of guidance raises concerns about, amongst others, the reliability, validity and usability of crime scripts.

In this paper, we present a method to visualise cybercrime scripts which could foster a more systematic

*Correspondence:
Renushka Madarie
r.madarie@vu.nl
[1] Department of Criminology, Faculty of Law, Vrije Universiteit (VU) Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
[2] Research Group Forensic Science, Faculty of Technology, Amsterdam University of Applied Sciences, Tafelbergweg 51, 1105 BD Amsterdam, The Netherlands
[3] Police Academy of the Netherlands, Postbus 834, 7301 BB Apeldoorn, The Netherlands

analysis of cybercrimes. We focused on this category of crime not only because the scripts in Dehghanniri and Borrion's (2019) systematic review most often fell into this category, but also because crimes in general are increasingly likely to include cybercrime elements as society becomes more digitised. This paper is structured as follows: first, we introduce crime script analysis as a research method. Subsequently, we explain how cybercrime scripts vary widely in terms of format and content which hinders meaningful comparisons between scripts. We also discuss several fundamental issues concerning crime scripting methodology. In part, these issues affect all sorts of crime scripts, but several issues relate to cybercrime scripts in particular. In the second part of this paper, we introduce object-oriented modelling and suggest two visualisation models to harmonise cybercrime scripts. We propose to use these models in addition to the typical textual narrative. In the third and final part, we discuss potential advantages of these models, but also address important questions about crime scripting that remain unsolved.

Crime script analysis

Cornish (1994) introduced crime script analysis to systematically organise and analyse information about the procedural aspects and requirements of crime commission processes. Essentially, crime scripts are coding schemes that dissect the actions offenders engage in before, during and after a crime event. These actions are causally related in the sense that prior actions affect or at least enable subsequent actions (Tompson & Chainey, 2011; Nisbett & Ross, 1980, in Cornish, 1994). Furthermore, these actions could comprise illicit as well as licit actions (Ekblom & Gill, 2016). Besides organising actions, crime scripts also dissect, amongst others, what actors, props (i.e., tools), and locations are involved.

Theoretically, crime scripting has been related to two paradigms in environmental criminology, namely the rational choice perspective (RCP) and situational crime prevention (SCP) (Cornish, 1994). The RCP assumes that offenders weigh the costs and benefits of criminal activities (Cornish & Clarke, 2017), albeit limited by cognitive and situational restraints (Arthur, 1994). Decision-making during the criminal process could thus be considered as instrumental behaviour to achieve desired goals. By outlining the procedural aspects and requirements of a crime commission process, crime scripts could highlight these instrumental decisions.

In addition, as offender decision-making is affected by the offender's immediate environment, crime scripts could guide the identification of environmental factors that might hinder criminal activities. This relates to SCP which encourages analysts to develop intervention measures that manipulate the environment (Clarke, 2017). SCP suggests that the offender's environment could be altered by increasing the risks and effort of offending, decreasing its rewards, reducing provocations, and removing excuses. These five SCP categories could be used as guidelines for the development of situational intervention measures based on crime scripts (Leclerc, 2017).

To create a script, analysts could adhere to the 'universal script' framework proposed by Cornish (1994). This framework divides the crime commission process into general activity categories. These categories include: preparatory activities, entry into the crime scene, waiting for or establishing the right conditions under which the crime can be executed, instrumental activities before conducting the crime, execution of the (main) crime itself, and aftermath activities relating to exiting the crime scene and the crime itself. For each of these categories, analysts could specify, amongst others, the relevant actions, actors, tools and locations. However, while Cornish clearly outlined the concept and usefulness of crime scripting, how to actually conduct crime scripting remained vague. For instance, he did not further detail how to populate the framework with data, how to distinguish between different types of scripts, and how to assess the quality of a script (Borrion, 2013; Ekblom & Gill, 2016).

Dehghanniri and Borrion (2019) conducted a systematic review on current crime scripting practices. They divided the types of crime scripted in several categories and noted that cybercrime was the crime category most commonly scripted. Their results also demonstrate that studies applying crime scripting generally do not detail all steps conducted during the data collection and analysis process. Furthermore, all scripts they examined contained a textual narrative, with half of the scripts also containing a table, and about a third containing a flowchart. While these results are indicative of the variety in crime scripting practices, it remains unclear what published crime scripts generally look like and what new insights these scripts have generated. To obtain a more elaborate understanding of current crime scripting results, we examined cybercrime scripts in greater detail. While several scripts we examined were included in Dehghanniri and Borrion's systematic review, a number of scripts were published more recently. In the following section, we compare these scripts in terms of format and content.

Current cybercrime scripts

Similar to Dehghanniri and Borrion (2019), we also noted that all cybercrime scripts we examined present the narratives through text, while several scripts also include a

table (e.g., Basamanowicz & Bouchard, 2011; Willison & Siponen, 2009) or a flowchart (e.g., Bodker et al., 2022; Matthijsse et al., 2023). Although these visualisation models neatly summarise the textual narrative, there is great variation between the formats of these models. This variation is partly due to the varying numbers of activity categories researchers use to structure their findings. While Cornish (1994) suggested nine categories for his universal framework, few researchers adhere to these categories. Most outline significantly less categories, with a few studies only reporting as little as three categories (Bodker et al., 2022; Leukfeldt, 2014).

Incongruence between script formats is further caused by varying distinctions between main categories and subcategories, and by inconsistent use of terminology. For instance, O'Hara and colleagues (2020) describe eight activity categories (termed "stages"), such as entry to the setting, various instrumental activities, execution of the main crime and post-activities. Subsequently, they group these categories into two main "phases", namely the crime set up and the crime achievement phase (originally described by Leclerc and colleagues (2011)). Conversely, Warren and colleagues (2017) describe four main "stages" which together contain six "phases", such as preconditions, instrumental initiation, crime initiation, and disengagement. To add further confusion, researchers inconsistently describe their activity categories not only as "phases" (Soudijn & Zegers, 2012) or "stages" (Van Der Bruggen & Blokland, 2021), but also as "scenes" or "scene functions" (Hutchings & Holt, 2015; Leppänen et al., 2020), or "steps" (Basamanowicz & Bouchard, 2011). To remain somewhat consistent with previous literature, we opt to refer to activity categories as "phases" in the remainder of this paper.

In contrast to the format, the content of cybercrime scripts appears to be more streamlined – at first glance. Overall, these scripts contain information about the elements suggested by Cornish (1994), namely actions, actors, tools, and locations. Actors could refer to (types of) offenders or victims, but also to third parties facilitating crime. In addition, researchers also discuss relationships or interactions between actors. The tools we distilled from the cybercrime scripts could be broadly categorised as cognitive, physical and digital tools. Cognitive tools could include motivations, knowledge and cognitive abilities such as planning (Basamanowicz & Bouchard, 2011; Van der Bruggen & Blokland, 2021; Warren et al., 2017). Physical tools could be specific to certain crimes, such as ATM cards used for withdrawing money from phishing victims (Leukfeldt, 2014), or more common, computers or mobile phones. Digital tools could comprise software, such as malware or VPN software.

Digital tools could also function as online locations, such as online forums and marketplaces, as they serve as platforms where people in the online world connect. However, these platforms are essentially composed of software as well. For instance, online marketplaces generally require at least a payment solution, databases, and a user interface. Similar yet simpler is the software application Telegram, which could be considered both a tool and a location. This application enables offenders to connect with victims or other offenders. In doing so, it functions as a platform where these actors meet, thereby also serving as a crime scene or offender convergence setting (Leukfeldt et al., 2017; Soudijn & Zegers, 2012). This brings us to the point that it might not be as easy or meaningful to distinguish between the digital versions of "props" and "locations", as it is with their counterparts in the physical realm as discussed by Cornish (1994).

In addition to software, digital tools could also comprise data. All cybercrime studies we have examined detailed data or information that is used or desired by offenders during the crime commission process.[1] This is not surprising given that cybercrime generally revolves around data (Porcedda & Wall, 2021). A typical example of data as a digital tool is victim information, such as credit card information or account information, but also child-abuse videos or pictures (Van Der Bruggen & Blokland, 2021; Van Hardeveld et al., 2016). Note that there exists a dependency between data and software. Amongst others, data requires software for its creation and usability. To this end, these digital tools could be considered as nested in the sense that digital data is generally handled by software. We return to this nesting when we explain our object-oriented models.

Actions in cybercrime scripts are sometimes nested as well. For instance, Matthijsse and colleagues (2023) scripted ransomware attacks on organisations. Two actions they describe are gaining access to a computer system and encryption of the data on these systems. Whereas the former comprises more specific actions, such as phishing or exploitation of technical vulnerabilities, the latter is a rather specific action in itself. More generally, while certain actions are quite simple or specific, more complex actions might be worthy of their own scripts. This nesting of actions was also discussed by Cornish (1994). He suggested that a complex crime could be regarded as a composite script with broad activities that each require their own script. We did not find cybercrime scripts that have applied this idea. However, several

[1] While we recognise there is a difference between data and information (Zins, 2007), we use these terms interchangeably in the remainder of this paper to avoid unnecessarily complicating matters.

non-cybercrime scripts exemplify this nesting, such as Chainey and Alonso Berbotto (2021) who examined oil theft, and Tompson and Chainey (2011) who examined illegal waste activity.

Fundamental issues of crime script analysis

The lack of a clear scripting method appears to have resulted in the use of a multitude of methods which subsequently lead to published cybercrime scripts in all sorts and sizes. While each of these scripts generated new insights into the crime phenomenon analysed, the variability in the composition of scripts also raises more fundamental concerns about current crime script analyses. In this section, we discuss three such concerns: (1) the reliability of scripts in general, (2) the reliability and validity of concepts used to structure cybercrime scripts, and (3) the value of crime scripts beyond the specific type of crime scripted.

First, the lack of a clear scripting procedure undermines the reliability of published scripts. As previously stated, analysts often provide only limited descriptions of the methods they have applied to create their scripts (Dehghanniri & Borrion, 2019). This makes it difficult to replicate these studies and to compare different (types of) scripts. In addition, as Borrion (2013) noted, without a clear and valid procedure, it is hard to verify if scripts have been correctly produced. Furthermore, scripts that were created intuitively might be influenced by cognitive biases, such as the confirmation bias and availability bias. These biases could affect, for instance, data selection and script population. Moreover, these issues also impede the identification of knowledge gaps in crime scripts. Consequently, despite the popularity of crime scripting to examine crime in general, and cybercrime specifically, it still appears that "confidence in the produced scripts is almost entirely based on the analyst's credibility" (Borrion, 2013, p.5).

Second, issues of reliability, and even validity, extend to the concepts used to structure cybercrime scripts. As discussed, crime elements in the physical world, such as tools and locations, might not be as useful for dissecting crimes committed in the digital realm. Digital 'tools' and 'locations' might not always bear meaningful differences. Moreover, the concept of 'tool' now appears to be a catch-all term for any "thing" other than what resembles a location or an actor. While the use of this concept may be technically correct, if it encompasses a wide variety of objects with widely differing utilities, it is of little help in identifying patterns in the crime commission process. To exemplify, a tool could be a Tor browser or account credentials. Both could aid offenders in committing crime, but comparing the use or attributes of these tools is futile because these are in essence very different types of objects.

Third, the lack of comparable scripts implies that the knowledge generated by these scripts likely remains limited to the specific types of crimes analysed. While the cybercrime scripts we analysed provide intriguing accounts of *those* crimes, one might wonder what they (could) teach us about cybercrimes or offender decision-making *in general*. We just addressed the futility of comparing incomparable tools. If scripts were more comparable, they could aid further theorising about, for instance, what attributes of tools offenders are likely to consider in their decision-making process. Similarly, Ekblom and Gill (2016) address the relevance of patterns in offender behaviours for theorisation. Patterns in scripted actions could, for instance, lead to hypothesising about the processes causing the observed patterns, which in turn fosters a deductive approach of theory testing. Conversely, irregularities in patterns could stem from too detailed observation methods, but could also challenge theories or signal new patterns in offender behaviour (e.g., the development of new modus operandi). However, patterns could only be identified when processes are systematically analysed and presented.

Borrion (2013) stated that everyone could intuitively create crime scripts, but there is a difference between models simply representing data and models communicating relevant information. In the next paragraphs, we introduce two visualisation methods for a more systematic analysis and presentation of key elements of cybercrime scripts. While we do not aim to completely streamline the full crime scripting *process* (Dehghanniri & Borrion, 2016), the proposed methods could better streamline *scripts*. In doing so, the methods have the potential to address several practical and fundamental issues discussed so far.

Visualising the script: a novel approach

The methods we propose to visualise and structure cybercrime scripts are based on object-oriented modelling (OOM), which was suggested by Borrion in 2013 as a promising direction to further refine crime scripting. As explained, prior cybercrime scripts, and in particular the corresponding flowcharts, have been composed in various ways and rather intuitively. Furthermore, no established methods have been applied to visualise the scripts. To facilitate the streamlining of cybercrime scripts, we suggest to visualise the typical textual narrative by creating object-oriented models with a standardised visualisation language, namely the Unified Modelling Language (UML). These models should encourage analysts to systematically consider, organise and relate several key elements of cybercrime scripts. In this section, we first

explain OOM and two models based on UML. Subsequently, we illustrate the use of these visualisation techniques with a phishing script.

Introducing object-oriented modelling

OOM is an approach to model systems and is commonly used in software engineering (Shen et al., 2002). Systems could represent a wide variety of phenomena, such as software applications, business processes like shipping a parcel, or even student administration. With OOM, analysts could model real-world entities within systems as "objects" with specific attributes and behaviours (Larman, 2005; Rumbaugh et al., 1999). As we exemplify below, objects can be things or ideas. Two kinds of object models exist: dynamic and static (Larman, 2005). Both model objects and their relationships within a system. Dynamic models focus on the interaction between objects over time to accomplish specific goals and thereby mainly represent the behaviour of a system. Static models focus on the attributes of objects and thereby mainly represent the structure of a system.

There are several ways to create dynamic and static models. We depart from UML which is a widely applied visual modelling language to specify and visualise system artifacts (Rumbaugh et al., 1999). Although other visualisation languages exist, such as Business Process Modelling, UML appears best suited to our needs as it is a highly versatile language and designed to model detailed relationships. Furthermore, UML models are suitable for conversion to computational models which has several potential advantages that we outline in our discussion section. Different types of UML dynamic and static models exist. Here, we discuss activity diagrams and domain models because we will use (elements of) these models to visualise crime scripts.

Activity diagrams are dynamic UML models and illustrate sequential actions in a system (Larman, 2005). Importantly, activity diagrams also permit actions to run parallel instead of sequential. Figure 1 illustrates an activity diagram and its notation by outlining the processing of a business order. The process starts with a start node and ends with an end node. Actions, such as receiving an order, are noted in boxes with round edges. Once an action is completed, the process moves on to the next action. While actions should be of similar specificity, less specific actions could be assigned a "rake" notation, such as for shipping the order. The rake symbol indicates that the corresponding action has its own activity (sub) diagram. Objects, such as an invoice, are noted as rectangles. Because objects are separately modelled in a static model, not all objects need to be visualised in a dynamic model.

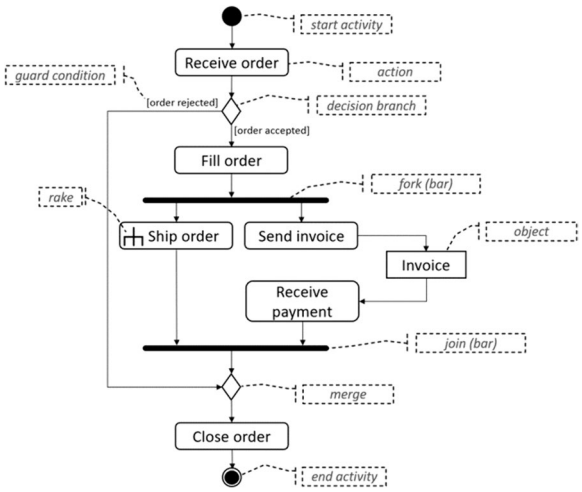

Fig. 1 Example of an activity diagram adapted from Fakhroutdinov (2024) and Larman (2005)

Two interesting features of activity diagrams relate to the transition of actions. First, activity diagrams explicitly specify decision points. These decision points (or branches) are noted with diamonds. Each decision point has two or more arrows departing from its diamonds. Each arrow has a guard condition that specifies under what condition that specific arrow is followed. In Fig. 1, after an order is received, a decision is made on how to continue: should the order be rejected or accepted? Second, activity diagrams differentiate between alternative paths and concurrent actions. A decision point branches into alternative paths. A merge diamond symbolises the point where these alternative paths merge again. As opposed to decision points, fork bars denote concurrent paths or actions. Figure 1 illustrates that shipping the order and ensuring payment happen concurrently. The join bar denotes where these concurrent paths end. As opposed to merge diamonds, the actions ending in a join bar *must all* be completed before the process can continue.

Domain models are static UML models that structure objects in a domain or system (Larman, 2005). Domain models generally visualise three static aspects of a domain, namely objects, associations between objects, and attributes of objects. Figure 2 illustrates a simplified domain model. This model exemplifies objects involved in banking fraud, such as offender and a bank account. The associations describe relationships between objects. In this case, offenders target accounts which are owned by customers. Furthermore, attributes of objects are described directly below each object name. For instance, bank accounts likely store a certain amount of money and are accessed using credentials.

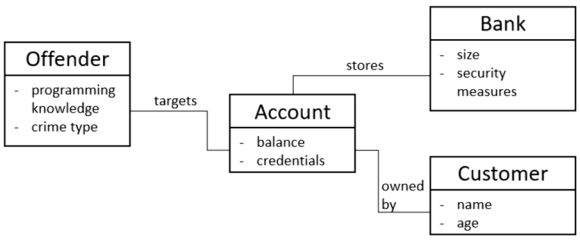

Fig. 2 Example of a simple domain model

Although domain models are intended to provide an overview of noteworthy objects within a system, these models might be difficult to understand, especially if the model contains numerous objects and the reader is unfamiliar with the process analysed. Therefore, we propose to further structure the model by incorporating the key elements we previously identified to be commonly present in cybercrime scripts as distinct layers within the model. These elements comprise actors, tools and locations. However, given the ambiguity surrounding the concepts of 'tools' and 'locations' in the digital realm, we instead differentiate between 'data' and 'software' when considering digital tools. With regard to the physical aspects of a crime, we adhere to the categories of tools and locations. Although we also distilled cognitive tools in crime scripts, we consider these as attributes of various actors and therefore these are not represented by a separate layer.

The key elements group the various objects into different hierarchical layers in the domain model similar to the layers of abstraction in computing. For instance, Kernighan (2017) explains that hardware, operating systems, and software could be considered layers that are hierarchically structured. All computers are made of hardware even though this hardware could be different for different types of computers. In a similar vein, all computers run an operating system, but different types of operating systems exist. Which operating system runs on a computer is independent of its hardware. Similarly, the same type of software could be installed on different operating systems.

In our layered domain model, we start with actors as the top layer because these are central and essential to each crime, and interact most directly and meaningfully with the second layer: digital information. For instance, actors could create, buy or encrypt information. In turn, information is handled by software which is the third layer. Software resides in physical tools which form the fourth layer. Finally, physical tools can be located in a physical setting which is the lowest and fifth layer. Figure 4 illustrates a layered domain model for our phishing script. In the next section we demonstrate how activity diagrams and layered domain models complement each other as well as the traditional textual narrative.

Applying object-oriented modelling to a cybercrime script

To illustrate the application of the models, we apply OOM to a crime script on data thieves' behaviours when trespassing organisations' computer networks that we created in a previous study (Madarie et al., 2024). To examine data thieves' behaviours, we conducted semi-structured interviews with cybersecurity experts working at a Dutch cybersecurity company. The experts had backgrounds in digital forensics, pentesting, monitoring & detection, incident response and threat intelligence. A more elaborate description of our data collection and analysis is provided in Appendix A. Our thematic analysis revealed that data thieves also conduct phishing attacks to access organisations' mailboxes. When phishing, offenders send phishing mails that appear legitimate but are only used as a tool to illicitly obtain data from the recipient.

Because our aim is to illustrate the application of the visualisation models, we only highlight a few aspects of the phishing script. The full activity diagram is attached in Appendix B, but for the reader's convenience, we illustrate a snippet of this diagram in Fig. 3. We crafted the activity diagram after creating the textual narrative. First, to create the activity diagram, we identified the main actors from the narrative as well as which actions were conducted by which actors. Subsequently, we arranged the actions in a sequential order and considered what conditions correspond to alternative courses of action. Finally, we reflected on which objects were useful to add to the diagram and which actions deserved a rake. Because we could not find clear guidelines on when to add objects to an activity diagram, we decided to add only 'target information' because this object was in the text but had no action box associated with it in the diagram.

After creating the activity diagram we crafted the layered domain model illustrated in Fig. 4. To identify which objects should be depicted in the model, we returned to both the textual narrative and the activity diagram (Larman, 2005). The objects identified in these parts of the script were arranged in their respective layer. In addition, objects that could be considered to be handled by another object were placed above each other with the handled object placed above the handling object. Furthermore, we also tried to cluster the objects topically. That is, the stronger the objects are related, the closer they are placed together. Associations were subsequently vertically drawn between objects from different layers or sublayers. We did not draw horizontal associations – those between objects in the same layer – because these were less occurring or evident. The textual narrative

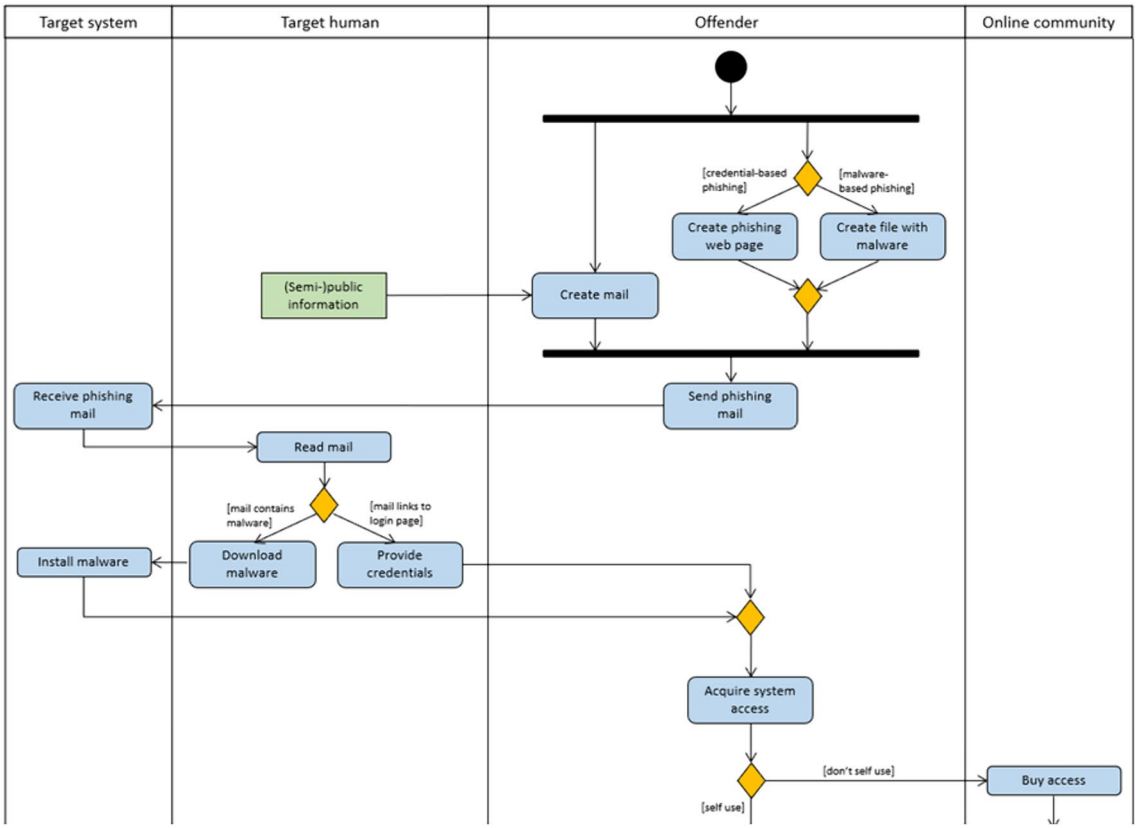

Fig. 3 Snippet of activity diagram for phishing

provided information on what attributes are relevant for which objects. While it was tempting to infer objects or attributes through deduction, such as a Web server storing a Web page, or age as an attribute of Offender, we followed Larman's advice to "not add things that are not there" (2005, p.145). This later allowed us to examine what knowledge gaps exist in our script. Finally, we used PowerPoint to create our visualisation models.

A short phishing script

In this section, we describe parts of the phishing script in text and relate these to snippet of the activity diagram (Fig. 3) and the layered domain model (Fig. 4). Note that we added another UML concept in Fig. 3 to further clarify the script, namely **columns** for actors we considered central to the script (Larman, 2005; Rumbaugh et al., 1999). These columns not only organise actions by actor, but also illustrate which interactions occur at what point during the process. To aid the reader in connecting the models with the text, we occasionally signal aspects of the models in italics and within parentheses when related to aspects of the text. To differentiate between the activity diagram and layered domain model, we refer to the former as 'F3' (i.e., 'in Fig. 3') and the latter as F4 (i.e. 'in Fig. 4').

To steal data from a mailbox, offenders could use either credential-based or malware-based phishing. In credential-based phishing, offenders send mails in which they generally lure their victims to a phishing website where the victims fill in their account credentials. In malware-based phishing, offenders send mails usually with attachments containing malware. This malware is subsequently used to access the computer system of the victim. In this script, we discuss both methods as alternative paths after a decision point (*F3: diamond*) to obtain access to a mailbox.

The fork bar (*F3*) indicates concurrent actions, namely creating a mail and adding a phishing aspect to it (i.e., link to phishing page or attachment with malware). To entice recipients of a phishing mail to click on a link or open an attachment, offenders try to make the mail look legitimate. To this end, they could mimic the appearance (*F4: attribute*) of legitimate mails, for instance, by copying the visual appearance of typical Microsoft mails. In addition to its visual appearance, offenders could also tailor the content (*F4: attribute*) to the recipient to increase the mail's credibility. To create more targeted mails,

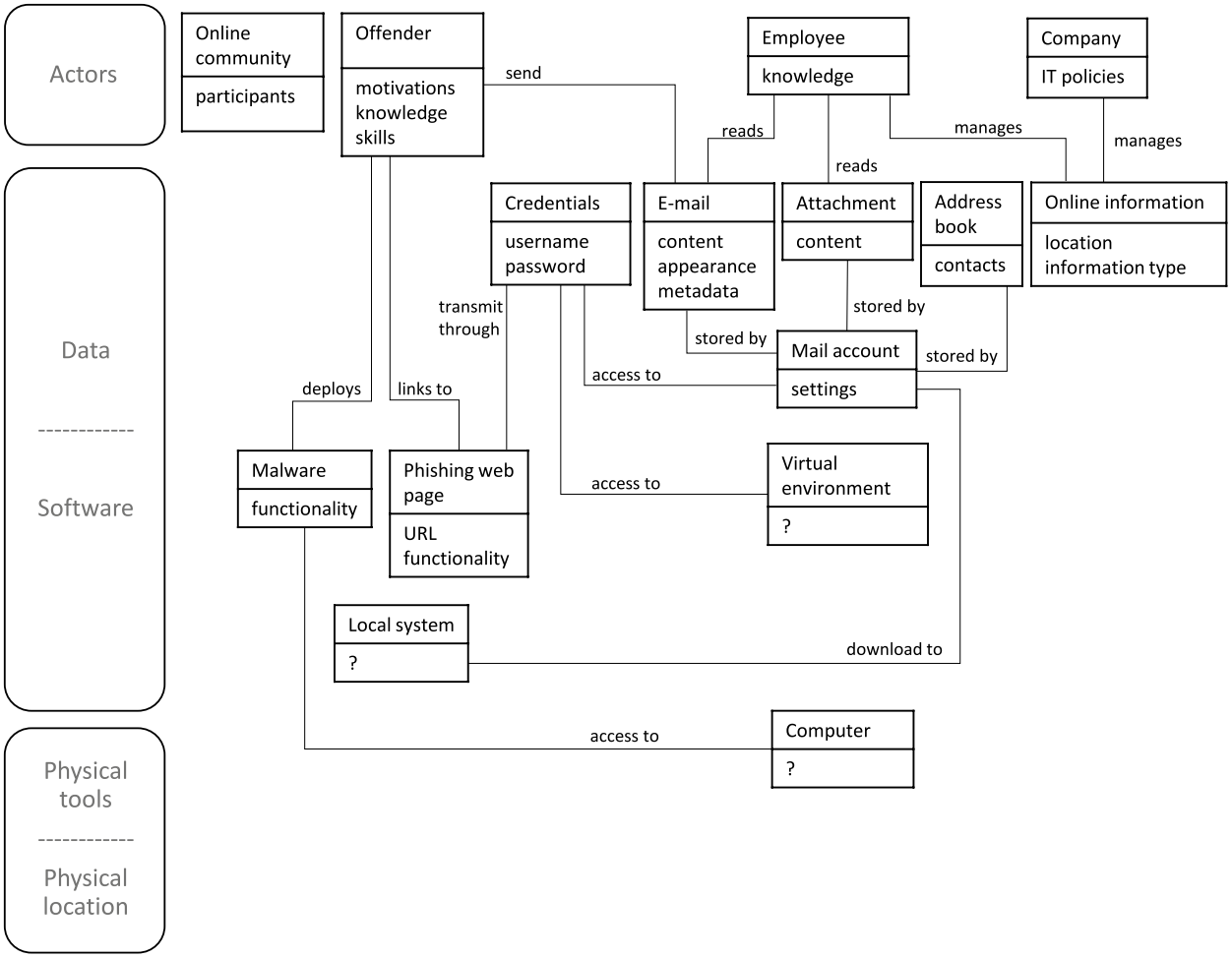

Fig. 4 Layered domain model for phishing

offenders could use (semi-)public information (*F3: object box*) about the organisation or its employees.

Offenders could also make phishing mails appear more legitimate by sending mails from previously hacked mail accounts. Because recipients recognise the sender (*F4: mail address as metadata attribute*), they are more likely to consider the mail to be legitimate. Malware such as Emotet or Qbot also enable offenders to engage in mail conversations in an automated manner. By using content of previously sent mails, it appears as if the recipient receives a response to prior messages.

Offenders sending malware-based phishing mails apply similar tactics to those sending credential-based phishing mails, but modify the message. Instead of asking for credentials, recipients are encouraged to open an attachment, for instance by stating: "*Hey, could you check the attachment?*" If recipients open the attachment (*F4: object*) and the system allows the attachment to perform certain actions, the malware (*F4: object*) could be downloaded and installed automatically. Subsequently, this malware allows offenders access to the compromised system. Respondents noted that, instead of exploiting the access themselves, offenders could also sell this access to others (*F3: interaction with online community*).

What offenders could do once inside a mailbox is further visualised in appendix B. Here, we note a few additional UML concepts that were not explained before, but that were or could be applied in our activity diagram. First, we added additional circles (Fakhroutdinov, 2024). The circles containing an "A" indicate a loop. The

circle containing a "?" implies that we do not know what actions offenders conduct after finishing the action prior to the ?-circle. Second, although all our guard conditions contain descriptions, such as "mail contains malware" versus "mail links to login page", another valid condition would be "else".

Furthermore, our activity diagram contains no distinct phases (e.g., preparation phase or instrumental initiation) because we believe the diagram is rather self-explanatory. Nevertheless, analysts could add such categories to uphold tradition and keep in line with previous crime scripts. One way to do so is by adding phases to the side of the diagram as was done similarly by Matthijsse and colleagues (2023, p.10). Moreover, this diagram, just like the narrative, describes the ideal scenario for an offender. In reality, recipients of phishing mails could realise the mail is fake after reading it and therefore delete it. Alternatively, they could change their credentials before the offender could abuse the provided access. This, in fact, points to potential intervention measures. In the next paragraph, we explain how activity diagrams like this could improve systematic development of intervention measures as well as theorising about offender decision-making.

Figure 4 illustrates the layered domain model. What stands out at first is that a relatively large portion of the objects in this model relate to data from victims, while there is very little data directly relating to the offender. This discrepancy likely stems from our method of data collection. The cybersecurity experts we interviewed focused more on investigating victims rather than offenders. Consequently, they were able to provide more information about victims and trespassed systems, including their attributes. Furthermore, the lower the layer, the less objects and attributes we could identify. This pattern likely also results from our data collection rather than there not being any relevant physical or software objects in the phishing process.

The domain model also clearly reveals a flow of objects connected through various associations. For instance, 'Credentials' and 'Mail account' could be considered nested targets that are impacted sequentially. An offender first has to obtain credentials and then could access the mail account. While this could be induced from the text and activity diagram as well, the domain model more clearly illustrates this flow. Following these flows, we might identify more targets, such as the 'Virtual environment' and 'Computer' of the victim. However, the current script contains too little information about these objects to determine their role in crime. Nevertheless, this nesting of targets might fuel theorising about offender decision-making, which we address in the discussion.

Discussion

Crime script analysis as a research method once proposed by Cornish (1994) still appears to be underdeveloped as evidenced by recently published studies on cybercrime scripts. These scripts vary greatly in terms of format and the crime elements they address. All cybercrime scripts, including their visualisations such as flowcharts, that we reviewed are intuitively structured. Furthermore, although all scripts address actors, tools and locations, especially the latter two elements are currently difficult to compare across scripts because they tend to encompass widely varying types of digital objects. As argued by Borrion (2013), the lack of a structured scripting method creates fundamental methodological issues concerning the reliability and validity of the results. In addition, while current scripts provide interesting insights into the specific phenomenon analysed, it is generally unclear how these results aid further theorising about offender behaviour (Ekblom & Gill, 2016).

While the goal of this paper is not to fully streamline the scripting *process*, we advocate the application of OOM, and in particular UML, to better streamline (cyber)crime *scripts*. The visualisation models we propose were chosen from a wide array of UML models because these models highlight and structure precisely those aspects that could be considered key elements of cybercrime scripts. In addition, using these models might facilitate a more elaborate analysis of cybercrime scripts, and perhaps even crime scripts in general. We discuss several important potential advantages of the proposed UML models next. However, we also recognise that OOM is no silver bullet and several important questions relating to the scripting process remain. We address these issues in the final part of this paper.

Potential advantages

Enhancing the reliability and validity of scripts and their key elements We argued that the use of UML could improve the reliability of scripts because UML is a standardised visualisation language and widely applied in various contexts (Larman, 2005; Rumbaugh et al., 1999). We thus rely on best practices that facilitate the development of more consistently structured scripts. Furthermore, we

slightly rephrased and ordered the key elements distilled from prior cybercrime scripts. We presume the currently used elements (e.g., 'software' and 'physical tools') to be less ambiguous, which should enhance the validity and reliability of the elements layered in the domain model.

Additionally, the formats of the models could limit the influence of biases by guiding analysts to explicitly consider key elements of the crime commission process and their relationships. For instance, analysts have to reflect on interactions between actors, decision points, and object attributes when creating their models. As we observed in Fig. 4, this reflection could reveal interesting knowledge gaps that might otherwise remain unquestioned.

Comparisons across scripts More comparable crime scripts facilitate theorising about offender decision-making that is less specific to particular crimes. Specifically, comparing decision points and alternative paths within and between activity diagrams might uncover new behavioural patterns or present additional support for assumed patterns. This notion is similar to Ekblom and Gill's (2016) argument about patterns of behaviour fuelling inductive or deductive prediction.

Similarly, layered domain models could aid theory development, testing and refinement when comparing different attributes and objects. For instance, what target attributes appear most relevant in which context? What target attributes are best conceptualised, quantified and tested? And how are different types of objects and their attributes relevant to offender decision-making or situational crime prevention? Such questions might further develop, for instance, the Routine Activity Theory that addresses various actors, their attributes and their interactions (Cohen & Felson, 1979).

Alternative ways of identifying intervention measures Activity diagrams also stimulate analysts to consider what alternative paths offenders could take at which point, and what conditions result in which route. These alternative actions relate to what Cornish (1994, p.175) termed "script permutators" and variation through "tracks". He suggested that these alternative paths stimulate thinking about, for instance, potential displacement of crime following situational intervention measures and variation due to technological innovation.

Besides alternative paths, activity diagrams also explicate concurrent paths. Theoretically, if not all actions along these paths succeed, the join bar will not be passed and the offender cannot proceed. This implies that intervention measures aimed at disrupting at least one of multiple concurrent paths might be a relatively efficient way of disturbing a criminal process.

Layered domain models could be useful for identifying useful intervention measures when comparing multiple of these models. The same or highly similar objects might recur in several scripts. Cornish (1994, p.176) mentioned "nodes" when discussing physical locations that were important in various criminal processes. Similarly, digital nodes could exist that enable various sorts of cybercrime. Indeed, such "settings" or software applications have already been identified, like the Telegram app (Hebel et al., 2021). These nodes might inspire intervention measures that could affect a range of crimes.

Improving our shared understanding Crime scripts are developed and used by both academics and practitioners. Researchers have suggested that crime scripts could serve as a common language allowing different stakeholders to efficiently share their information and knowledge (Tompson & Chainey, 2011; Warren et al., 2017). By using a standardised visualisation language to structure scripts, knowledge between different research projects, organisations and even sectors could be more easily shared and compared. Moreover, the models could serve as "visual dictionaries" about crime phenomena (Larman, 2005, p.135). As such, visualisation could aid those who are less keen on reading lengthy texts. Some people (if not all people) are simply better able to absorb information from pictures than long lines of words.

Crime scripting could also aid criminal investigations (Leclerc, 2017). The visualisation models could provide a quick overview of what is already known in an investigation and what information is still lacking. Furthermore, while not strictly UML, investigators might visually differentiate between licit and illicit actions and tools in these models, similar to Tompson and Chainey's (2011) flowchart. In doing so, these models could direct investigation efforts to those actions and tools most relevant to investigate.

Future prospects Another exciting aspect about the use of OOM and UML is their potential to implement human-drawn models in software applications and to quantify script elements. For instance, domain models could serve as blue prints for databases storing objects and their attributes and associations. Subsequently, when adding domain models of multiple crime scripts into the same database, patterns might emerge in, for instance, the co-occurrence of target attributes or the distribution of information between layers. As

explained, such patterns could aid theorising about offender decision-making or inform future data collection strategies.

Furthermore, the UML models might facilitate computational modelling. For instance, the arrows in activity diagrams could be quantified like Markov Chains by modelling the probability of transitioning from one action to the next (Bar et al., 2016; Fraunholz et al., 2018). More broadly, UML models could serve as a structured method to facilitate the development of models simulating complex and dynamic systems or scenarios. For instance, Hill and colleagues (2014) applied agent-based modelling to simulate patterns in wildlife poaching and predict where poachers would place snares. In their model, they included agents (i.e. objects), properties (i.e., attributes), interactions, and so on. Similarly to prior studies on crime scripting, they visualised their models using tables and flow charts. The application of UML could facilitate the structuring of these visualisations, thereby serving as a stepping stone to computational models. In a similar vein, Le Sage and colleagues (2013) applied OOM to structure a threat scenario. However, they only verbally described the relevant elements in the scenario, such as the objects and their attributes and associations. Again, UML could facilitate the development of such scenarios and assist in communicating about threat scenarios with stakeholders.

Remaining questions

Despite the potential advantages, the proposed visualisation models cannot address all issues concerning crime scripting methodology. We consider the models to be a starting point for more systematic crime script analysis, but important questions remain. We address several of these question here.

First, Wortley (2014; in Ekblom & Gill, 2016) describes an essential dilemma of crime scripts in general, that is how abstract or specific a script should be. While the rake symbol in activity diagrams reliefs some of the effort to describe actions of similar specificity, at the moment it remains debatable how detailed activity diagrams and their corresponding texts should be. In a similar vein, attributes in domain models could be considered objects by themselves. For instance, the 'metadata' attribute of the object E-mail could also be an object with its own attributes. More advanced OOM could make additional distinctions between classes and subclasses (concepts we did not discuss in this paper because we consider this to be an introduction of OOM in the field of criminology and crime science). Nonetheless, it remains unclear how general or specific these models should be to be useful.

Second, relatedly, how do we meaningfully differentiate between different scripts? Our script concerns data theft from mailboxes through phishing. Would it have been more useful to create two separate scripts: one for the phishing attack and one for infiltrating the mailbox? Furthermore, how do we best model crimes involving a large number of actors (or roles) or actions? Again, the rake symbol facilitates scaling of actions by allowing for higher-level descriptions of actions that, in turn, represent nested scripts describing more detailed actions. This differentiation between scripts should also limit the number of actors per script and enhance comprehensibility. For instance, a high-level action such as 'phishing for credentials' might include a rake and its subscript could include more specific actors (than 'offender') who are not included in a subscript on 'infiltrating a network'. However, the ideal focus or length of a script could be subject of future studies. Meanwhile, analysts might want to consider this issue in the very first phase of the crime scripting process, namely when formulating the problem (Dehghanniri & Borrion, 2016).

Finally, although the visualisation models could advance crime scripting, their actual benefits should be tested. How do analysts in different sectors perceive these models? Do the models actually facilitate an improved understanding of the crime phenomenon and offender decision-making? And how difficult is it for those less familiar with OOM to generate such models? Furthermore, while the activity diagram could be relatively easily applied to model non-cyber crimes, it would be intriguing to see if domain models are useful to model other types of crimes as well.

Appendix A

This appendix elaborates on our data collection and analysis strategy employed when creating the phishing crime script for this article. The following text is translated from Madarie and colleagues (2024).

Data collection

For this study, we interviewed twelve cybersecurity experts working at a cybersecurity company in the Netherlands. This company mainly catered to small and medium-sized enterprises (SMEs) and public sector organisations. We recruited participants from different departments to ensure a diverse set of knowledge and experience. All participants approached were willing to participate. We interviewed four forensic investigators, four pentesters, two security operations centre (SOC) analysts and two threat intelligence analysts. Several participants also worked on projects for other departments in addition to their primary responsibilities. In addition, most participants also fulfilled incident response duties. Because most participants have held various cybersecurity positions, they were also able to discuss cybersecurity topics beyond their primary function during the interview.

The semi-structured interviews lasted 75 min on average. The questions primarily addressed offenders' methods of infiltrating and operating within organisations' systems, and their data extraction techniques. When respondents' answers became more similar (i.e., nearing saturation), the questionnaire was gradually supplemented with more specific questions and more tailored to the participant's subdiscipline. We also sought participants' views on contradictions noted in earlier interviews. Finally, participants were given the opportunity to verify our findings for accuracy, which a few of them did.

Data analysis

All interviews were transcribed verbatim. The transcripts were analysed in two stages. The first stage involved a thematic analysis (Braun & Clarke, 2006). Thematic analysis is a method of distilling patterns from data and then analysing them. For our thematic analysis, we took inspiration from the six steps described by Braun and Clarke. These steps include familiarising yourself with the data, generating initial codes, searching for themes, reviewing and defining themes and producing the report. The second stage involved a crime script analysis.

To better familiarise ourselves with the data, we first summarised the transcripts using interview topics, or themes, based on the interview questions, which resulted in a topic list. We created new topics upon encountering information that did not fit well under the previously established topics. The resulting summary provided an initial overview of the interview data. Subsequently, we thematically coded the transcripts using our topic list, yet again remained open to new relevant information that did not fit existing codes, thereby creating new codes. Finally, we categorised the codes into larger, overarching themes and visualised their relationships in a network of codes. These relationships showed, amongst others, sequences of actions over time. These sequences provided the input for the second analysis stage, namely the crime script analysis.

During the crime script analysis, we further structured and related sequential patterns found in the data. Our script consisted of both a narrative and a flowchart. The narrative details the different actions distilled from the data in the first analysis stage. These actions are categorised according to both expert-defined phases and phases commonly used by academics. Expert-defined phases are based on the terminology used by the respondents to present their perspectives as accurately as possible. The academic phases encompass the four crime script phases suggested by Tompson and Chainey (2011). The flowchart related the actions detailed in the narrative to subgoals guiding the actions and environmental factors facilitating the attacks.

Appendix B

Full activity diagram.

See Fig. 5

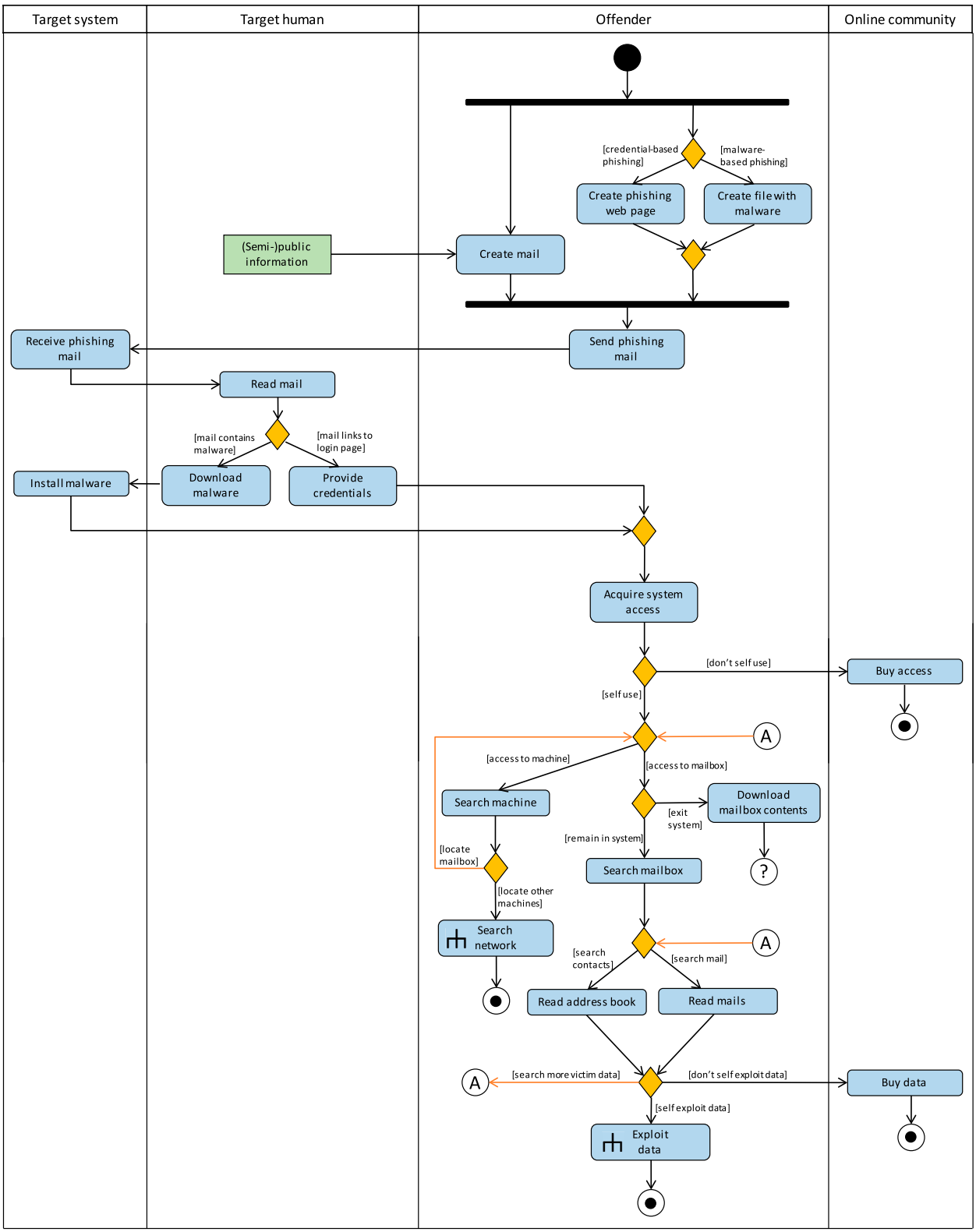

Fig. 5 Full activity diagram

Acknowledgements
We thank the anonymous reviewers for their insightful feedback and valuable suggestions. We also thank Leah and Roan for their contribution to the literature review.

Author contributions
RM: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing (original draft), Visualization, Project administration, Funding acquisition. MWK: Conceptualization, Methodology, Writing (review & editing), Supervision, Project administration, Funding acquisition. CdP: Conceptualization, Methodology, Writing (review & editing), Supervision, Project administration, Funding acquisition. All authors read and approved the final manuscript.

Funding
This study was funded by the Dutch National Police. Politie & Wetenschap PW.OV.2022.61.

Availability of data and materials
The datasets used and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Ethical approval was granted by the Ethics Committee of the Amsterdam University of Applied Sciences (Ref: HVA-239). Participants were briefed on the contents of this study and provided written consent for participation and publication before being interviewed. After the interviews, all participants were debriefed.

Competing interests
The authors declare that they have no competing interests.

Received: 15 January 2024 Accepted: 5 September 2024
Published online: 04 October 2024

References

Arthur, W. B. (1994). Inductive reasoning and bounded rationality. *The American Economic Review, 84*(2), 406.

Bar, A., Shapira, B., Rokach, L., & Unger, M. (2016). Identifying attack propagation patterns in honeypots using markov chains modeling and complex networks analysis. *2016 IEEE International Conference on Software Science Technology and Engineering (SWSTE)*. https://doi.org/10.1109/SWSTE.2016.13

Basamanowicz, J., & Bouchard, M. (2011). Overcoming the warez paradox: Online piracy groups and situational crime prevention. *Policy & Internet, 3*(2), 1–25. https://doi.org/10.2202/1944-2866.1125

Bodker, A., Connolly, P., Sing, O., Hutchins, B., Townsley, M., & Drew, J. (2022). Card-not-present fraud: Using crime scripts to inform crime prevention initiatives. *Security Journal*. https://doi.org/10.1057/s41284-022-00359-w

Borrion, H. (2013). Quality assurance in crime scripting. *Crime Science*. https://doi.org/10.1186/2193-7680-2-6

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa.

Chainey, S. P., & Alonso Berbotto, A. (2021). A structured methodical process for populating a crime script of organized crime activity using OSINT. *Trends in Organized Crime*. https://doi.org/10.1007/s12117-021-09428-9

Clarke, R. V. (2017). Situational crime prevention. In R. Wortley & M. Townsley (Eds.), *Environmental Criminology and Crime Analysis* (2nd ed., pp. 286–303). New York: Routledge.

Cohen, L. E., & Felson, M. (1979). Social change and crime rate trends: A routine activity approach (1979). *American Sociological Review, 44*, 588–608.

Cornish, D. (1994). The procedural analysis of offending and its relevance for situational prevention. *Crime Prevention Studies, 3*(1), 151.

Cornish, D. B., & Clarke, R. V. (2017). The rational choice perspective. In R. Wortley & M. Townsley (Eds.), *Environmental Criminology and Crime Analysis* (2nd ed., pp. 29–61). New York: Routledge.

Dehghanniri, H., & Borrion, H. (2016). Toward a more structured crime scripting method. *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*. https://doi.org/10.1109/REW.2016.030

Dehghanniri, H., & Borrion, H. (2019). Crime scripting: A systematic review. *European Journal of Criminology*. https://doi.org/10.1177/1477370819850943

Ekblom, P., & Gill, M. (2016). Rewriting the script: Cross-disciplinary exploration and conceptual consolidation of the procedural analysis of crime. *European Journal on Criminal Policy and Research*. https://doi.org/10.1007/s10610-015-9291-9

Fakhroutdinov, K. (2024) Activity Diagrams. https://www.uml-diagrams.org/activity-diagrams.html. Accessed 24 Sept 2024

Fraunholz, D., Schneider, D., Zemitis, J., & Schotten, H. D. (2018). Hack my company: An empirical assessment of post-exploitation behavior and lateral movement in cloud environments. *Proceedings of the Central European Cybersecurity Conference, 2018*, 1–6. https://doi.org/10.1145/3277570.3277573

Hebel, C., Hoppenstedt, M., & Rosenbach, M. (2021, June 11). The Telegram Billionaire and His Dark Empire. *Der Spiegel*. https://www.spiegel.de/international/world/the-telegram-billionaire-and-his-dark-empire-a-f27cb79f-86ae-48de-bdbd-8df604d07cc8. Accessed 24 Sept 2024

Hill, J. F., Johnson, S. D., & Borrion, H. (2014). Potential uses of computer agent-based simulation modelling in the evaluation of wildlife poaching. In A. Lemieux (Ed.), *Situational Prevention of Poaching* (pp. 120–153). London: Routledge.

Hutchings, A., & Holt, T. J. (2015). A Crime script analysis of the online stolen data market. *British Journal of Criminology*. https://doi.org/10.1093/bjc/azu106

Kernighan, B. W. (2017). *Understanding the digital world: What you need to know about computers, the internet, privacy, and security*. Princeton University Press.

Le Sage, T., Toubaline, S., & Borrion, H. (2013). An object-oriented approach for modelling security scenarios. *International Conference on Computer Modelling and Simulation*. https://doi.org/10.1109/UKSim.2013.65

Larman, C. (2005). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development* (3rd ed.). Pearson Education: New Jersey.

Leclerc, B. (2017). *Boosting crime scene investigations capabilities through crime script analysis in the routledge international handbook of forensic intelligence and criminology*. London: Routledge.

Leclerc, B., Wortley, R., & Smallbone, S. (2011). Getting into the script of adult child sex offenders and mapping out situational prevention measures. *Journal of Research in Crime and Delinquency, 48*(2), 209–237. https://doi.org/10.1177/0022427810391540

Leppänen, A., Toiviainen, T., & Kankaanranta, T. (2020). From a vulnerability search to a criminal case: Script analysis of an SQL injection attack. *International Journal of Cyber Criminology, 14*(1), 63.

Leukfeldt, E. R. (2014). Phishing for suitable targets in The Netherlands: Routine activity theory and phishing victimization. *Cyberpsychology, Behavior, and Social Networking*. https://doi.org/10.1089/cyber.2014.0008

Leukfeldt, R., Kleemans, E., & Stol, W. (2017). The use of online crime markets by cybercriminal networks: A view from within. *American Behavioral Scientist*. https://doi.org/10.1177/0002764217734267

Matthijsse, S. R., Van Thoff-de Geode, M. S., & Leukfeldt, E. R. (2023). Your files have been encrypted: A crime script analysis of ransomware attacks. *Trends in Organized Crime*. https://doi.org/10.1007/s12117-023-09496-z

Madarie, R., Weulen Kranenbarg, M. & De Poot, C. J. (2024). Online inbrekers bekeken: Een crime script analyse van datadiefstal. NFIR. https://www.nfir.nl/whitepaper-online-inbrekers-bekeken. Accessed 24 Sept 2024.

O'Hara, A. C., Ko, R. K. L., Mazerolle, L., & Rimer, J. R. (2020). Crime script analysis for adult image-based sexual abuse: A study of crime intervention points for retribution-style offenders. *Crime Science*. https://doi.org/10.1186/s40163-020-00130-9

Porcedda, M. G., & Wall, D. S. (2021). Modelling the cybercrime cascade effect in data crime. *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2021*, 161–177. https://doi.org/10.1109/EuroSPW54576.2021.00025

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The unified modeling language reference manual*. Addison-Wesley.
Shen, W., Compton, K., & Huggins, J. (2002). A toolset for supporting UML static and dynamic model checking. *Proceedings 26th Annual International Computer Software and Applications*. https://doi.org/10.1109/CMPSAC.2002.1044545
Soudijn, M. R. J., & Zegers, B. C. H. T. (2012). Cybercrime and virtual offender convergence settings. *Trends Organized Crime, 15*, 111–129. https://doi.org/10.1007/s12117-012-9159-z
Tompson, L., & Chainey, S. (2011). Profiling illegal waste activity: using crime scripts as a data collection and analytical strategy. *European Journal on Criminal Policy and Research, 17*, 179–201. https://doi.org/10.1007/s10610-011-9146-y
Van Hardeveld, G. J., Webber, C., & O'Hara, K. (2016). Discovering credit card fraud methods in online tutorials. *OnSt16*. https://doi.org/10.1145/2915368.2915369
Van Der Bruggen, M., & Blokland, A. (2021). A crime script analysis of child sexual exploitation material fora on the Darkweb. *Sexual Abuse, 33*(8), 950–974. https://doi.org/10.1177/1079063220981063
Warren, S., Oxburgh, G., Briggs, P., & Wall, D. (2017). How might crime-scripts be used to support the understanding and policing of cloud crime. In T. Tryfonas (Ed.), *Human Aspects of Information Security, Privacy and Trust* (pp. 539–556). Cham: Springer International Publishing.
Willison, R., & Siponen, M. (2009). Overcoming the insider: reducing employee computer crime through situational crime prevention. *Communication of the ACM, 52*(9), 133–137. https://doi.org/10.1145/1562164.1562198
Zins, C. (2007). Conceptual approaches for defining data, information, and knowledge. *Journal of the American Society for Information Science and Technology, 58*(4), 479–493. https://doi.org/10.1002/asi.20508

Publisher's Note