# Comparative Analysis of Attack Trees and Crime Scripts Modelling Credential Stuffing Attacks

LILYA SALIBA, University of Twente, The Netherlands

In crime behaviour analysis, the importance of Attack Modelling Techniques (AMT) and model-based formalisms is evident in providing a way to prepare for, or prevent, any future attacks. The attack tree and the crime script are two of these formalisms, which will be central to the topic of this paper. Both will be analysed and compared with each other to determine their similarities as well as discrepancies and to conclude the possibility of combining them. The scope of this research will be narrowed down to model credential stuffing attacks specifically, focusing on one cybercrime example. This paper will conclude that it is possible to manually translate the two formalisms into each other, denoting which aspects need to be considered to align these two formalisms using a literature review and comparative analysis. The contribution of this paper includes the prospect of using the attack tree and the crime script conjointly, proposing a concept to improve the completeness of these crime behaviour analysis formalisms.

Additional Key Words and Phrases: Crime Script, Attack Tree, Cybersecurity, Credential Stuffing.

## 1 INTRODUCTION

In the current age, with technology being a major part of society, digital crimes take up a significant portion of all criminal deeds. This is further reflected in the web publication "The Netherlands in Numbers 2022" by Statistics Netherlands (Centraal Bureau voor de Statistiek, CBS) [28], which states that 16.9% of the population fell victim to cybercrime in 2021, which was approximately equivalent to the amount of traditional crime. With digital currency, data stored online and businesses utilising digital systems, there are numerous possibilities for an attacker to target. Cyberattacks can range from simple theft to ransomware, hate crimes and destruction of data.

One common attack is credential stuffing. This is an automated digital attack that can happen on a personal as well as corporate level. During credential stuffing, the attacker will collect large samples of leaked credentials and automate bots to log into multiple systems to find any username and password combination that allows access to an account. This attack succeeds when users reuse their credentials for multiple systems and when the design of these systems lacks security measures like Multi-Factor Authentication (MFA) [20], allowing a login with just the correct username and password combination [18].

In the computer science field, learning the behaviour of cybercriminals can help prevent future attacks or incidents. To achieve this, using model-based formalisms to analyse crimes can help understand the process behind the crime or the requirements needed to commit it. As stated by Lallie et al. [12], decision makers, like developers, IT experts and CEOs, need Attack Modelling Techniques (AMT) in order to properly tackle issues related to cybersecurity. AMTs are a popular method of mathematically and visually representing the sequence of events that lead to a successful cyberattack [13]. Developers can design technical systems with the uncovered weaknesses in mind to further strengthen their security in these areas.

Being able to combine such AMTs or other formalisms can reveal more of these weaknesses since different techniques will highlight different aspects within a crime. This, in turn, would strengthen the security and, consequently, reduce the probability and frequency of such digital attacks happening, ensuring more safety in digital environments. This paper aims to analyse the possibility of such a combination using the two formalisms, the attack tree and the crime script.

### 1.1 Research Questions

Following up on the previously mentioned information, answering the following research question (RQ) will be the goal within this study.

- *How can attack trees and crime scripts, modelling credential stuffing attacks, be translated from one to another?*

The main RQ will help define to what extent it is possible to combine the two formalisms modelling credential stuffing. Furthermore, the sub-questions below will assist in answering the main RQ.

- **RQ1** What are the main similarities between attack trees and crime scripts?
- **RQ2** What are significant differences between attack trees and crime scripts?
- **RQ3** What features of the attack tree or crime script will be lost when transforming one into another?

### 1.2 Paper Structure

Following in this paper will be the acknowledgement of related work in this domain (section 2) and the research methodology (section 3). In section 4, each formalism, along with the selected crime, will be explained. Section 5 follows with the analysis of features within both formalisms, providing the foundational material to answer **RQ1** and **RQ2**. Furthermore, section 6 will aid in answering **RQ3**, noting the details on the transformation. Finally, a conclusion with a reflection and possible future work can be found at the end.

## 2 RELATED WORK

In this subsection the related studies will be acknowledged and discussed in how they are relevant to this study.

First, Cornish's paper on situational crime scripts in 1994 [7] introduces this formalism in the field of criminology. This literature provides the foundation for the crime script, explaining its features and structure in depth. Likewise, Schneier popularised attack trees in 1999 [26], providing an elaborate foundation for this formalism.

*TScIT 43, July 4, 2025, Enschede, The Netherlands*

Schneier lays out the structure and methodology and explains the utilisation of the attack tree.

Second, in 2024, a study by Madarie, Weulen Kranenbarg and De Poot [14] proposes using Object-Oriented Modelling (OOM) with the standard visualisation language Unified Modelling Language (UML) to improve crime scripts and their analysis. This study explored the enhancement of the crime script by first applying UML models to provide a reliable structure. Their study shares a similar objective, although this paper will use attack trees instead of OOM and consider mutual application.

Lastly, a study conducted by Schiele and Gadyatskaya in 2021 [25] provides a formal approach to transform attack trees to attack graphs. Schiele and Gadyatskaya introduce an algorithm for a one-way transformation along with formal definitions and a thorough analysis. Conversely, Haque and Atkison [10] performed a literature study on transforming attack graphs into attack trees and discussed different aspects regarding this transformation. Both of these studies consider a one-way transformation with attack trees and attack graphs. This study will consider both approaches and provide a comparative analysis instead.

These studies show similar research; however, they do not exactly cover the topics which this study will address. There is a lack of research regarding the combination or transformation between the crime script and the attack tree. This paper aims to cover this gap by performing a literature review and analysing these formalisms while considering credential stuffing attacks.

## 3 METHODOLOGY AND APPROACH

In order to answer the previously declared RQs, first each model is constructed, then a literature review is conducted, followed by a thorough analysis of both formalisms. This will ensure that each formalism is studied in depth by finding different perspectives, which are discussed in existing literature, so that the RQs can be answered.

### 3.1 Literature Search

For the literature search, Google Scholar[1] and dblp.org[2] were used to find the initial collection of papers. The keywords used to find relevant papers are "attack tree", "crime script", "cybercrime", "credential stuffing" and "analysis". The first two inputs are used in every query to obtain results related to either the crime script or the attack tree. However, they were used separately, as at the time this paper is written, there are merely two results when applying both of them ("attack tree" AND "crime script"), of which both results were found on Google Scholar.

*3.1.1 DBLP.* In addition to the lack of results mentioned above, dblp.org provides five results in total with the query "crime script". These were all scanned by title to determine whether they fit this study according to criteria which is mentioned later in this section. Conversely, the query "attack tree" returned 454 matches on dblp.org. The papers were filtered by the year they were published in, starting with the oldest publishings and working to more recent years. This was done to facilitate the literature search, so papers between 1999 and 2016 were considered, since these years yield under 20 results on dblp.org.

*3.1.2 Google Scholar.* All the following approaches were applied with Google Scholar. Since there are no relevant papers on comparing the attack tree and crime script, **RQ1** and **RQ2**, which focus on their comparison, must be answered through separate papers. Literature that provides an analysis of the formalisms themselves is considered for this, so papers with titles containing the name of the formalism and the keyword "analysis". This is under the presumption that such analyses will cover specific aspects within the formalisms so that they can be used to answer **RQ1** and **RQ2**. When searching for papers related to crime scripts, the keywords "cyber" or "credential stuffing" were included to find more relevant results; the full queries used are "cybercrime script" and "crime script credential stuffing".

For **RQ3** regarding transformation, additional papers are collected. Attack trees are one of many crime-modelling graph types, among fault trees, defence trees and more [11]. Literature covering topics regarding the combination, transformation or conversion between the attack tree and one of its alternatives seems to exceed that of the crime script. In order to gather sufficient literature for **RQ3** on both formalisms, papers on the enhancement of the crime script or attack tree through other frameworks were also considered.

After selecting this first set of papers with this literature search, others are considered through citation search, also referred to as backwards snowballing. Each paper was scanned to determine whether it covered relevant topics, and based on the content, any referenced papers within, which seemed valuable, were considered as well. This yielded the second set of papers. Then, the selection was refined by comparing the collected literature and repeating the process to find literature on any encountered gaps.

*3.1.3 Included papers.* In total, 11 out of 29 selected papers were included, ranging from the years 1994 up until 2025, of which 9 papers were published in 2013 or later. Papers are included when peer-reviewed, written in English and if they focus on providing an analysis or discussion on elements either within the attack tree or crime script. During this process, some papers were excluded, as they either went beyond the scope of this research project or did not cover any relevant topics. Due to citation search, some papers would overlap and cover the same material; in this case, the paper which is being referenced to is included when available, and those based on it are excluded. Papers which were unavailable or inaccessible have also been excluded. In Table 1 the list of excluded papers can be found.

### 3.2 Creating the Formalisms

Sample models of attack trees are manually created with Draw.io[3], a simple graphing tool, to recreate scenarios and compare these with the crime script, which is also manually scripted. In order to produce models on credential stuffing as complete and thorough as possible, grey literature from OWASP Foundation [18] and MITRE [20] was considered to source any information regarding the cybercrime, including the attack steps. On top of this, reports from F5

[1] https://scholar.google.com/
[2] https://dblp.org/
[3] https://www.drawio.com/

Table 1. List of excluded papers with reason of exclusion.

| Authors | Title | Source | Reason |
|---|---|---|---|
| Harjinder Singh Lallie, Kurt Debattista, Jay Bal | An Empirical Evaluation of the Effectiveness of Attack Graphs and Fault Trees in Cyber-Attack Perception. | [12] | Out of scope |
| Ana Maria Pirca, Harjinder Singh Lallie | An empirical evaluation of the effectiveness of attack graphs and MITRE ATT&CK matrices in aiding cyber attack perception amongst decision-makers | [21] | Out of scope |
| Ludovic Pièrre-Cambacédès, Marc Bouissou | Beyond Attack Trees: Dynamic Security Modeling with Boolean Logic Driven Markov Processes (BDMP) | [22] | Out of scope |
| Aliyu Tanko Ali, Damas Gruska | Dynamic Attack Trees | [2] | Duplicate |
| Maxime Audinot, Sophie Pinchinat, Barbara Kordy | Is My Attack Tree Correct? | [4] | Duplicate |
| Hamad Al-Mohannadi, Qublai Mirza, Anitta Namanya, Irfan Awan, Andrea Cullen, Jules Disso | Cyber-Attack Modeling Analysis Techniques: An Overview | [1] | Irrelevant topics / content |
| Sifra R. Matthijsse, M. Susanne van 't Hoff-de Goede, E. Rutger Leukfeldt | Your files have been encrypted: a crime script analysis of ransomware attacks | [16] | Irrelevant topics / content |
| Paul Ekblom | Design and Security | [8] | Irrelevant topics / content |
| Roger C. Schank, Robert P. Abelson | Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures (1st ed.) | [24] | Inaccessible |

Labs [17, 27] on credential stuffing were considered to supplement information on the attack steps.

## 4 BACKGROUND

In this section the two model-based formalisms will be explained in detail while illustrating a credential stuffing attack. This attack scenario is used to showcase how the formalisms are utilised in the context of a digital crime.

### 4.1 Credential Stuffing

Before proceeding on to the two formalisms, it is important to first define credential stuffing in more detail.

Credential stuffing is a type of brute-force attack [18, 20]. Brute-force attacks will generally attempt multiple inputs to guess the correct password through a trial-and-error process. There are other attacks which are similar to credential stuffing, one of which is another subset of the brute-forcing category, that is, password spraying. In a password spraying attack, the attacker uses common or frequently used passwords to attempt an account takeover [23]. Credential stuffing differs from password spraying, as it uses leaked data with the actual username and password combinations. Credential stuffing also must not be confused with credential dumping, in which the attacker gains access to credentials through the device's storage or memory [29].

In a credential stuffing attack, multiple steps can be defined which will be modelled with the two formalisms in the following subsections. First, the attacker must find credentials either through a database leak or password dump site, buy them on the dark web or deploy methods to collect credentials from the targets (e.g. phishing or malware) [18, 20]. Second, the bots must be set up for automated login attempts. This includes writing or downloading/buying a script and selecting the target systems to deploy them on [17, 27]. Then, the bots are activated, sometimes first with a test attack [27], and the attacker will collect the credentials which allow for a successful login. Now the attacker can log into and take over their target's account.

### 4.2 Crime Script

The next point of focus is one of the two formalisms, the crime script. Scripting human behaviour and cognitive processes was first introduced by Schank and Abelson in 1977 [24]. Only later, in 1994, did Cornish [7] introduce crime scripting, applying the scripting methodology to criminal situations.

During crime scripting, a scheme is created in which each step up until the end goal is written out in sequence. As Cornish stated, "Scripts are simply a way of highlighting the procedural aspects of crimes. In doing so, they emphasise the form of crime as a dynamic, sequential, contingent, improvised activity, and the content of specific crimes, considered as activities with particular requirements in terms of actions, casts, props, and spatio-temporal locations." [7, p. 175]. The script describes each step or action that is being taken and each decision that is made during the entire process of the crime as well as other details. The steps that are taken during the crime are categorised into certain stages or phases, ranging from the preparation, which generally occurs before the attacker enters the crime scene, to the preconditions for committing the crime, up until the postconditions, which might happen after the goal of the attack was achieved [7].

Furthermore, in crime scripting there are several levels of abstraction which must be mentioned [7]. The first level to consider is called the track, which is the least abstract level of generalisation. This is generally the level at which the specific crime is defined, for which the script is created, and has a high level of detail regarding the crime. These "tracks" can be part of the same script family, which generalises them to a broader concept of the attack, which is the script-level [6]. The protoscript defines the category of the script family and is of a higher level of abstraction than the script-level. After the protoscript, there is also the metascript-level, which is the highest form of generalisation in which a specific type of crime is still considered. The most abstract form is the universal script, which offers a procedural framework for crimes in general [7, p. 167]. The universal script provides a sequential order of nine stages which can be defined within a crime. These stages are preparation, entry, precondition, instrumental precondition, instrumental initiation, instrumental actualisation, doing, postcondition, and exit [6]. This framework can then be used to model scripts at the track-level [7, p. 160]. For the credential stuffing attack case, each level will be defined as follows:

- Metascript: Digital attack

Table 2. A crime script example of credential stuffing

Protoscript: Digital attack on person
Script: Account takeover
Track: Credential stuffing

| Scene/Function | Script Action |
|---|---|
| Preparation | Find leaked data from database breach<br>Format data for bots |
| Entry | Set up bots |
| Instrumental precondition | Select target systems<br>Write or buy/download script |
| Instrumental initiation | Perform a test attack<br>Run bots to test credentials automatically |
| Instrumental actualisation | Find successful logins |
| Doing | Log into target's account |
| Postcondition | Save successful credentials |
| Exit | Deactivate bots |

- Protoscript: Digital attack on person
- Script: Account takeover
- Track: Credential stuffing

In Table 2 a crime script on credential stuffing can be seen which declares these levels at the top, like Cornish does [7], and uses the stages of the universal script to categorise each attack step.

### 4.3 Attack Tree

Proceeding towards the next formalism, the attack tree will be addressed. In Schneier's article introducing the concept of attack trees in 1999 [26], a thorough explanation is provided. Attack trees are directed acyclic graphs (DAG) [11, 19] which are constructed starting at the top level with a root node, which depicts the ultimate goal of the crime. Each node can be refined by branching into child nodes on the next level [5]. In an attack tree, each child node depicts a sub-goal, being a prerequisite to reaching the next goal. A node can have several child nodes which each represent a possible way to achieve the node, allowing for multiple attack scenarios. Child nodes sharing a parent node can be connected to each other with an AND gate, meaning all are required in order to reach the parent node. Similarly, they can be depicted with an OR gate, which means at least one needs to be achieved to proceed to the parent node [26].

In Figure 1, an attack tree can be found on credential stuffing, similarly to the crime script. Aside from the graphical form, the attack tree can also be written in outline form [26], which is done in Table 3. For the outline form, every step is defined having either an AND or an OR gate, except for the final steps in sequence, which represent the leaf nodes. These two forms of the attack tree are equivalent and model the exact same attack, differing only in format.

Lastly, attack trees can also hold certain parameters within each node [26]. The parameters can be simple values like boolean values, costs or probabilities, which can be used to compute these elements for an entire attack path. A simple example of these values being used is included in section 5.4 as part of a fabricated scenario, considering the parameters are often estimated based on the real-world context of the crime case for accuracy and consistency [5].

Table 3. Attack tree on credential stuffing written in outline form

Goal: Log into account through credential stuffing (AND)
1. Enter credentials manually
2. Test credentials automatically (AND)
  2.1. Get credentials (OR)
    2.1.1. Deploy malware info-stealer
    2.1.2. Perform phishing attack
    2.1.3. Find previous database breach
    2.1.4. Acquire credentials on the dark web (password dump site)
  2.2. Perform a test attack
  2.3. Select target systems
  2.4. Set up bots (OR)
    2.4.1. Buy/Download script
    2.4.2. Write script

## 5 COMPARISON OF FEATURES

In this section, the attack tree and crime script will be compared to each other through individual aspects within their model. Four categories (structure, expansion, logical conjunction and disjunction and supplementary details) are defined in order to group these aspects of both formalisms for better coherence. The aspects are considered based on the features found in the literature from Cornish on crime scripts [7] and Schneier on attack trees [26]. After the discussion of the aspects of both formalisms, Table 4 is constructed to summarise findings and aid in answering **RQ1** and **RQ2**.

### 5.1 Structure

The discussion and comparison of the structures and foundations of both formalisms will highlight the core similarities and differences.

First of all, both attack trees and crime scripts are tools which can record and detail the steps, start and goal of a crime. They both deconstruct the crime they model into certain layers. For the attack tree these layers can be seen in the depth of the tree, where every child node exists on a lower level from its parent, depicting a precondition. For the crime script, these layers are portrayed in the different stages (e.g. preparation, entry, exit). Each step is identified with a textual label describing the attack step.

On top of this, attack trees and crime scripts can be considered linear models as the steps of a crime are denoted in sequence to form a straightforward path. However, a study by Madarie et al. [15] shows that crime scripting could also encompass crimes of cyclic nature. They state that research is often focused on successful attacks, disregarding the failed attempts. By showing the loops between failed steps and the attempts following after, the crime script becomes much more dynamic, incorporating dependencies between the steps. According to Kordy, Piètre-Cambacédès and Schweitzer, "Sequential formalisms take temporal aspects, such as dynamics time variations, and dependencies between considered actions, such as order or priority, into account; static approaches cannot model any of such relations." [11, p. 5]. Crime scripts can show such dependencies and temporal aspects with the stages/scenes, making it a sequential formalism; however, an attack tree cannot and is static instead. Although, Kordy, Piètre-Cambacédès and Schweitzer also point out several approaches in research extending attack trees to include these sequential aspects. One of these approaches includes

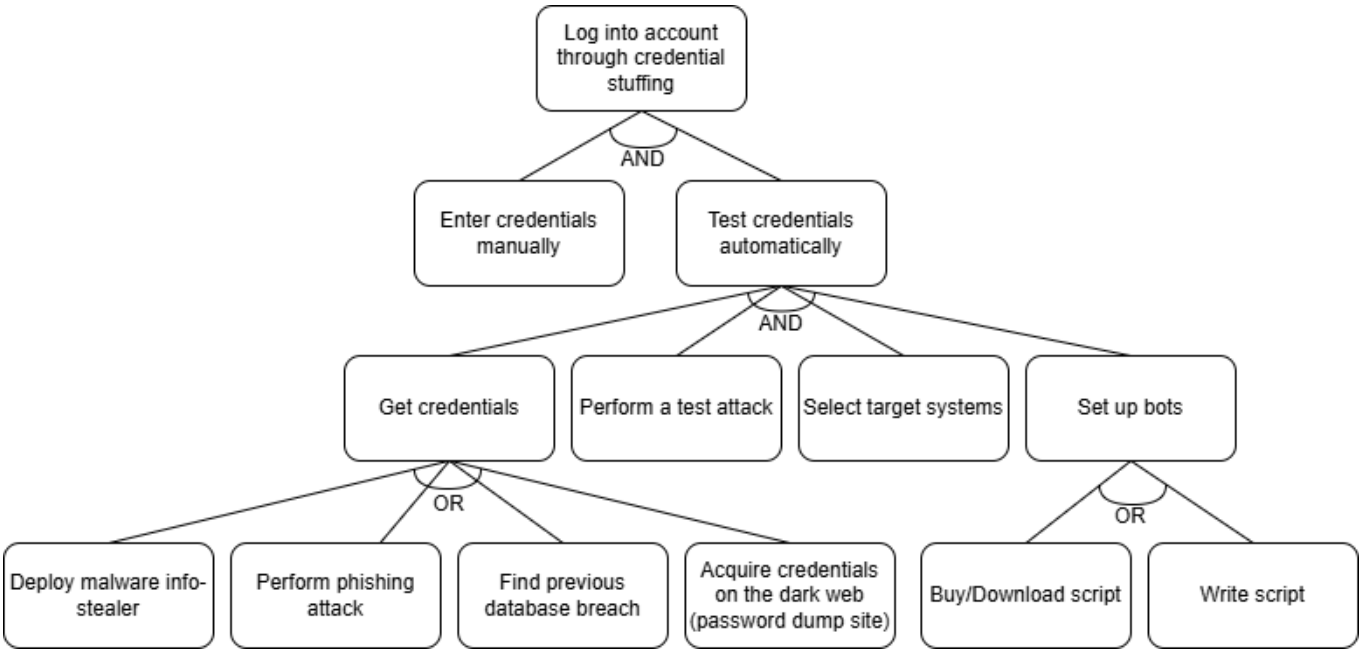


Fig. 1. Graphic attack tree on credential stuffing

sequential conjunction, which will be further discussed in section 5.3.

Lastly, there is a similarity in the way these formalisms are presented. A crime script is read top-down, starting with the beginning of the crime and ending at the exit or postcondition. On the contrary, the crime in an attack tree starts at the leaf node and ends at the goal at the root node. In the same way, the written outline form states the goal at the top and works downwards to the beginning of the attack. Reverting the order of either the crime script (Table 2) or the outline form of the attack tree (Table 3) will result in similar-looking structures.

## 5.2 Expansion

Crimes can be modelled in depth, including all details of the attack and thus constructing a very specific case. These specific cases can be categorised into more generalised crime categories. In the case of credential stuffing, it falls under the category *account takeover attacks*. Account takeover can be achieved through various attacks besides credential stuffing, acting as an encapsulating category. This subsection focuses on how these two formalisms can be expanded to a larger scale, considering these encapsulating attacks with broader scopes.

The crime script models credential stuffing at the track-level, as was mentioned before in section 4. Expanding this to the scope of *account takeover* would present a crime script at the script-level. Multiple track-level crime scripts within the same script family form this extended model at the script-level. This can be repeated for higher abstraction levels and even more generalised categories.

In attack trees, expansion can be accomplished by adjusting the root node with a new, more generalised goal and, subsequently, extending the tree with more branches depicting other ways to reach the goal. The new goal of the initial attack tree in Figure 1 would become *Log into account* as an example. This method of expansion can be repeated to increase the scope further, although it might quickly become unreadable due to its size, which is a limitation when scaling attack trees.

## 5.3 Logical Conjunction and Disjunction

In this subsection the conjunctive and disjunctive characteristics of the attack tree and crime script will be mentioned.

It is possible for an attack to have steps which all must be reached, acting as multiple preconditions. In a crime script, the steps that make up the preconditions can sometimes be grouped within the same row, appearing on multiple lines within the scene phase. In an attack tree this occurrence is depicted with the parent node having several child nodes connected through an AND gate. This indicates both formalisms are able to include a logical conjunction (AND) of attack steps. When it comes to logical disjunction (OR), however, the structure of the crime script has no specific notation for this unless it is simply formulated in writing. This is demonstrated in the crime script in Table 2 with *write or download script* in the instrumental precondition. Disjunction is clear in an attack tree through OR gates in the same way conjunction is.

Furthermore, multiple studies are incorporating dynamic attack trees, taking into account temporal aspects and dependencies between considered actions [11]. One of these studies, done by Arnold et al. [3], delves deeper into sequential and parallel modelling as an extension on dynamic attack trees. They propose a method in which the order of steps can be taken into account with both SAND (sequential AND) and SOR (sequential OR) gates. Which means

the attacker will only continue onto the next option if they fail at executing the primary action in a SOR gate, whereas for the SAND gate, all must be performed in the order they are presented. This approach provides more flexibility in modelling an attack, taking into account a dynamic rather than static nature of a crime.

### 5.4 Supplementary Details

Both the crime script and the attack tree are able to include additional information in their models aside from the attack steps and how they are connected. In this subsection, supplementary details will be pointed out.

To begin, the crime script can include additional columns besides the *Scene/Function* and *Script Action* columns. One of these is called *Failure Explanation*, which can provide the reasons as to why the corresponding step might fail at succeeding [7]. This presents additional valuable information about the structure of the crime which can be used for further analysis of criminal behaviour. Another additional column type is called *Situational Control* [7]. This column contains the ways in which the corresponding attack steps can be prevented or disrupted. The information provided by this column can present more opportunities for crime analysis. It enables a tool to determine crime prevention methods or strategies on top of crime analysis. Furthermore, additional columns are sometimes added to include other actors or roles, as was done by Garkava, Moneva and Rutger Leukfeldt [9], who present a crime script on trading stolen data with a column for the vendor, client and one for actions they both share. This provides more perspectives to the crime and can display relationships or interactions between actors [14].

As for the attack tree, most additional information is included as part of the tree graph, either inside the nodes or in their edges. As was previously shortly mentioned in section 4, attack trees can hold boolean or other simple values like costs or probabilities within each node. An example of this can be a boolean value depicting whether or not a certain step is possible, with notation *P = possible* and *I = impossible*. If the attacker is considered a person who is not well versed with programming or technical systems, the design of the attack tree can be realised as shown in Figure 2. In this design, the two possible attack paths are denoted with a dotted line. Depending on what is known about the attacker and their available resources or motive, certain attack paths can be ruled out to narrow down to the most likely scenario(s). Similarly, this method can be applied to nodes with costs or probabilities of success as parameters. A multi-parameter evaluation checks whether there exists an attack with a cost lower than the estimated budget and a probability of success greater than a certain threshold [5]. This results in the assessment of possible attack paths through estimations of data regarding the attacker, just as was done in the example with the boolean value.

## 6 TRANSFORMATION EFFECTS

In order to transform a crime script into an attack tree and vice versa, their features and capabilities must be considered. Transforming one formalism into the other can result in lost data or an incorrect application of features. This section will provide an answer to **RQ3**.

The crime script and attack tree both allow a basic representation of a crime by deconstructing it into individual attack steps linked in sequence. Both formalisms, especially when considering the dynamic attack tree, can include the sequence of actions taken and in which order they are executed.

Table 4. The similarities and differences between the attack tree and crime script

| Aspects | Attack Tree | Crime script |
|---|---|---|
| Attack steps | ✓ | ✓ |
| Logical conjunction | ✓ | ✓ |
| Logical disjunction | ✓ | ✓ |
| Dynamic extension | ✓ | ✓ |
| Expansion | ✓ | ✓ |
| Textual form | ✓ | ✓ |
| Visual (graph) form | ✓ | - |
| Multiple attack paths | ✓ | - |
| Quantitative parameters | ✓ | - |
| Attack phase/scene categorisation | - | ✓ |
| Situational control | - | ✓ |
| Failure explanation | - | ✓ |
| Roles/actors | - | ✓ |

It must also be mentioned that an attack tree can incorporate multiple attack paths, whereas a crime script can only do so for one. One solution to this dissimilarity is combining several crime scripts with the same goal to derive a complete attack tree from. Another solution is to adjust the goal of the attack tree to conform to one specific type of attack. This was also done for all the attack trees in this paper; instead of *Log into account* the goal is *Log into account through credential stuffing*, eliminating other ways of attack.

When transforming an attack tree to a crime script, multiple disjunct attack steps can be specified in a crime script with a textual description. Step 2.4.1 and 2.4.2 in the attack tree in Table 3 translate to *Write or buy/download script* in the instrumental precondition of the crime script in Table 2.

### 6.1 Difficulties

There are several features of the attack tree and crime script which are much harder to take over from one to another. This subsection will highlight the features that are likely to cause complications during transformation.

Although both formalisms include the attack steps of a crime, the crime script can hold more detail in this aspect compared to the attack tree. This can be explained by examining the process of creation for each formalism. In an attack tree it is the norm to use a decomposition technique during creation, finding all possibilities to reach the current node and repeating this for every other node [26]. The production of an attack tree is more focused on the width rather than depth of the structure. In a crime script, this is often the other way around, as only one attack is modelled in depth. The stages of the attack are considered, which can prompt the designer to approach from a different perspective. The crime script can include stages like entry and exit, involving the environment of the crime, making the location an integral element of the model [7]. Similarly, instrumental stages mark the inclusion of equipment. However, as Madarie et al. [14] pointed out, it is often difficult to discern

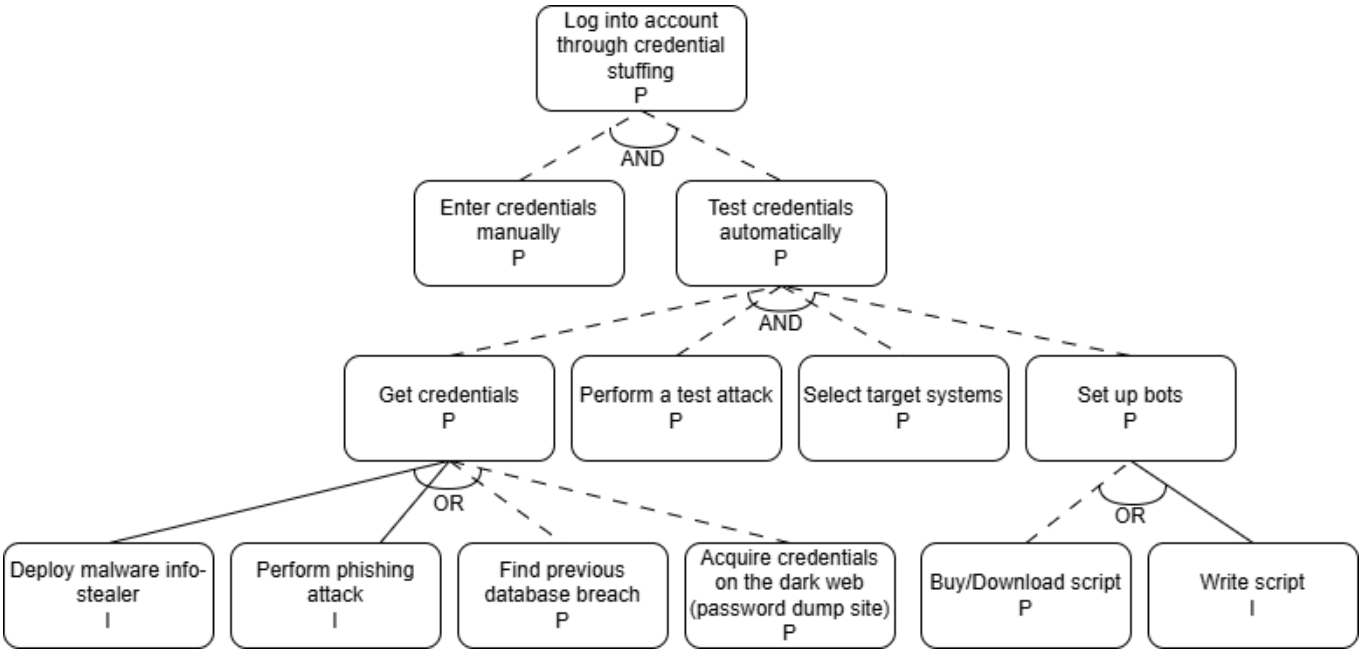


Fig. 2. Graphic attack tree on credential stuffing denoting possible attack paths with dotted lines (P = possible & I = impossible)

between the locations and tools in a digital context. In the credential stuffing crime script in Table 2, the automated system is defined as both the tool and location. An attack tree does not distinguish the equipment nor the environment when modelling a crime. These different processes of defining the attack steps result in differently structured node descriptions or attack actions. One example is step 2.1 *Get credentials* in the attack tree in Table 3, which encapsulates different ways in which the attacker can obtain credentials. This step does not occur in the crime script in Table 2, as it is already implied with the step *Find leaked data from database breach*. Another example related to the relation between the attack steps is with steps 2.3 and 2.4 in the attack tree (Table 3), sharing a parent through an AND gate, but ending up in different phases in the crime script (Table 2) because of the classification of the environment and equipment.

On top of this, the crime script does not end at the goal but can even include any actions taken after reaching it with postconditions. When considering the stages of a crime in the crime script, there is no available notation for this in the attack tree, unless the standard notation is disregarded. As stated by Lallie, Debattista and Bal [13], some attack trees can include colours or shapes for further modelling. As was mentioned in their paper, visual distance can be shown with colours or shapes, distinguishing different steps. They also mention different kinds of labels besides textual labels, like the character label, which consists of a single character and needs reference material for the reader to understand what it represents. This could be used to classify each node with the crime phase of the crime script. Unfortunately, Lallie, Debattista and Bal also discovered in their paper the ambiguity of attack tree semantics due to a lack of standardisation. This signifies that using colours, shapes or labels to incorporate phases from the crime script will demand deviation from the standard notation. On another note, when applying these phases while transforming an attack tree into a crime script, each step must be considered individually to determine to which phase they belong.

Furthermore, there is the issue of modelling a crime which has a cyclic structure. Although a crime script does show potential to incorporate such looped attack flows, the attack tree, by its definition, cannot, as it is a directed acyclic graph (DAG) [11, 19]. Instead, this must be achieved through other ways of dynamic extension on the attack tree, like sequential modelling [3].

Other features of the crime script which will prove difficult to derive are the additional columns for failure explanation, situational control or additional roles. These provide supplementary information about the crime beside the attack steps. Including defensive measures that can prevent an attack, like situation control, is possible with an attack-defence tree [11], which is an extension of the attack tree. However, for the standard attack tree, there is no notation for this.

Finally, the attack tree can include additional values like the boolean values, costs or probabilities which can be used in calculations to determine possible attack paths. This information is not available in the crime script but might be added in additional columns, although this data does not fit the script narrative. Aside from that, using incorrectly refined attack steps can lead to either over- or underestimating the probability, resulting in an incorrect attack tree [5]. This refinement of attack steps occurs when a goal is divided into sub-goals, just like how an attack tree is created. Translating a crime script to an attack tree might result in incorrect values since scripting does not define sub-goals this way.

## 7 CONCLUSION

Looking back at the research questions in section 1, these can now be answered using the discussion throughout this paper. The first two sub-questions are answered in Table 4, which provides an overview of all the differences as well as similarities between the attack tree and crime script.

**RQ1:** *What are the main similarities between attack trees and crime scripts?*

The main similarities between the attack tree and crime script are regarding core elements. These elements form the foundation for aligning the two formalisms. It must be mentioned that, although the similarities are present in both formalisms, this does not mean they are identical; rather, they overlap, as they differ in how they are applied.

**RQ2:** *What are significant differences between attack trees and crime scripts?*

Presentation of additional information is the most significant discrepancy between the two formalisms. The attack tree provides quantitative data in its nodes and models a crime in a more abstract way. The crime script uses exclusively qualitative data and follows a more descriptive approach, focusing on context. Without these elements, the respective formalism will result in being incomplete or incorrect, which can make them unusable in crime analysis. Notably, these elements are absent in the alternative formalism.

**RQ3:** *What features of the attack tree and crime script will be lost when transforming them into each other?*

All features in which the two formalisms differ will be lost upon translation, except for multiple attack paths. These features cannot be included in the alternative formalism without modifying its standard syntax. Aside from that, attack trees and crime scripts have different processes of defining their attack steps which must be considered. The attack steps defined by one formalism are not adapted to the other, which can also lead to missing sub-goals or other details within the attack step.

Building upon these three RQs and their answers, the answer to the main RQ can be constructed.

**Main RQ:** *How can attack trees and crime scripts, modelling credential stuffing attacks, be translated from one to another?*

It is important to mention that the translation requires a manual process in order to prevent loss of information and inaccuracy as much as possible. Translating an attack tree to a crime script means data in the parameters of the nodes is lost and scenes/phases must be added. This entails defining any tools, locations or actors that might be involved and reordering the attack steps according to this. Translating a crime script to an attack tree means the scenes and any additional columns are lost. The resulting attack tree might only present one attack type. If this is the case, other crime scripts within the same category must be used to complement the tree, or the goal must be narrowed down to the specific attack (which is credential stuffing in this case), and the AND and OR gates must be defined as well. If needed, any data in the parameters of the nodes must be newly constructed or refined.

In both cases after establishing translation, the formalism must be re-examined to ensure correctness. Although not ideal, it is also possible to include any lost elements with non-standardised, ad-hoc syntax. This should either be for personal use or for when a common understanding is established among the users of the formalism (e.g. within a team), to avoid inconsistencies with the standard usage. In light of these findings, the attack tree and crime script show enough overlap to align and supplement each other, being used conjointly to offer a valuable combination of features and greater analytical possibilities.

### 7.1 Reflection on Literature

Although there is a substantial amount of literature on crime scripts and attack trees, research that incorporated both of them is severely lacking. This led to finding literature separately for each formalism, which showed differing points of focus depending on the formalism. Literature on crime scripts seemed to be more focused on the behavioural side within criminology and covered crimes from other domains besides cybercrime. Research on attack trees was often focused on cyberattacks and provided formal or mathematical analyses. The formal or mathematical findings often found in the research on attack trees were not integrated into this study, as they did not provide valuable insights in relation to the crime script.

### 7.2 Limitations

Some limitations to this paper are identified regarding the literature review and other factors.

Although the methodology of the literature review defined inclusion and exclusion criteria, due to the gap in the combination of attack trees and crime scripts, this criteria was not further refined in order to find a higher number of relevant papers. This lack of refinement leaves room for bias in the literature search and selection. Overall, this led to a more generalised study; rather than thoroughly considering credential stuffing, the crime is used as an example instead. Although the topic of this study is highly specific, a possible refinement that could have been used is to only consider papers in cybersecurity or on cybercrimes that fall within the same category as credential stuffing (e.g. account takeover).

Furthermore, it is not feasible to create a complete attack tree or crime script incorporating every possible attack scenario. Although the process of modelling attack trees and writing crime scripts involved finding data through published reports, the formalisms might still be incomplete.

### 7.3 Future Work

Future work on this topic can be performed around the establishment of automatic transformation between the crime script and attack tree, or defining an algorithm for this. Another point left uncovered is creating a novel attack modelling technique combining these two formalisms to include what cannot be incorporated with their standard notation.

## ACKNOWLEDGMENTS

The author of this paper would like to thank Jan-Willem Bulleé and Christina Kolb, who supervised this project, for providing guidance and feedback throughout the duration of this research project. Further gratitude is extended to Stefan Morriën and Farida Elsadany for providing additional feedback and assistance.

## REFERENCES


[1] Hamad Al-Mohannadi, Qublai Mirza, Anitta Namanya, Irfan Awan, Andrea Cullen, and Jules Disso. 2016. Cyber-Attack Modeling Analysis Techniques: An Overview. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. Institute of Electrical and Electronics Engineers (IEEE), Vienna, 69–76. https://doi.org/10.1109/W-FiCloud.2016.29

[2] Aliyu Tanko Ali and Damas P Gruska. 2021. Dynamic Attack Trees. In *OVERLAY @ GandALF*. CEUR-WS.org, Padua, 25–29. https://ceur-ws.org/Vol-2987/paper5.pdf

[3] Florian Arnold, Dennis Guck, Rajesh Kumar, and Mariële Stoelinga. 2015. Sequential and Parallel Attack Tree Modelling. In *Computer Safety, Reliability, and Security*, Floor Koornneef and Coen van Gulijk (Eds.). Springer International Publishing, Cham, 291–299. https://doi.org/10.1007/978-3-319-24249-1_25

[4] Maxime Audinot, Sophie Pinchinat, and Barbara Kordy. 2017. Is My Attack Tree Correct?. In *Computer Security – ESORICS 2017*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer International Publishing, Cham, 83–102. https://doi.org/10.1007/978-3-319-66402-6_7

[5] Maxime Audinot, Sophie Pinchinat, and Barbara Kordy. 2018. Is my attack tree correct? Extended version. arXiv:1706.08507 [cs.CR] https://arxiv.org/abs/1706.08507

[6] Hervé Borrion. 2013. Quality assurance in crime scripting. *Crime Science* 2, 1 (2013), 6. https://doi.org/10.1186/2193-7680-2-6

[7] Derek B Cornish. 1994. The procedural analysis of offending and its relevance for situational prevention. *Crime prevention studies* 3, 1 (1994), 151–196. https://popcenter.asu.edu/sites/default/files/problems/stolen_goods/PDFs/Cornish1994.pdf

[8] Paul Ekblom. 2014. *Design and Security*. Palgrave Macmillan UK, London, 133–156. https://doi.org/10.1007/978-1-349-67284-4_7

[9] Taisiia Garkava, Asier Moneva, and E. Rutger Leukfeldt. 2024. Stolen data markets on Telegram: a crime script analysis and situational crime prevention measures. *Trends in Organized Crime* (10 Apr 2024). https://doi.org/10.1007/s12117-024-09532-6

[10] Md Shariful Haque and Travis Atkison. 2017. An Evolutionary Approach of Attack Graph to Attack Tree Conversion. *International Journal of Computer Network and Information Security* 9 (11 2017), 1–16. https://doi.org/10.5815/ijcnis.2017.11.01

[11] Barbara Kordy, Ludovic Piètre-Cambacédès, and Patrick Schweitzer. 2014. DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review* 13-14 (2014), 1–38. https://doi.org/10.1016/j.cosrev.2014.07.001

[12] Harjinder Singh Lallie, Kurt Debattista, and Jay Bal. 2018. An Empirical Evaluation of the Effectiveness of Attack Graphs and Fault Trees in Cyber-Attack Perception. *IEEE Transactions on Information Forensics and Security* 13, 5 (May 2018), 1110–1122. https://doi.org/10.1109/TIFS.2017.2771238

[13] Harjinder Singh Lallie, Kurt Debattista, and Jay Bal. 2020. A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review* 35 (2020), 100219. https://doi.org/10.1016/j.cosrev.2019.100219

[14] Renushka Madarie, Marleen Weulen Kranenbarg, and Christianne de Poot. 2024. Introducing object-oriented modelling to cybercrime scripting: visualisation for improved analysis. *Crime Science* 13, 1 (04 Oct 2024), 27. https://doi.org/10.1186/s40163-024-00227-5

[15] Renushka Madarie, Marleen Weulen Kranenbarg, and Christianne de Poot. 2025. Examining the cyclical nature of crimes: A looped crime script of data theft from organizational networks. *Computers in Human Behavior Reports* 17 (2025), 100548. https://doi.org/10.1016/j.chbr.2024.100548

[16] Sifra R. Matthijsse, M. Susanne van 't Hoff-de Goede, and E. Rutger Leukfeldt. 2023. Your files have been encrypted: a crime script analysis of ransomware attacks. *Trends in Organized Crime* (27 Apr 2023). https://doi.org/10.1007/s12117-023-09496-z

[17] John Miller. 2021. Credential Stuffing Tools and Techniques, Part 1. https://www.f5.com/labs/articles/threat-intelligence/credential-stuffing-tools-and-techniques-part-1.

[18] Neal Mueller, Jmanico, Dirk Wetter, kingthorin, Nick Malcolm, and Jahanrajkar Singh. 2024. Credential stuffing. https://owasp.org/www-community/attacks/Credential_stuffing.

[19] Stefano M. Nicoletti, Marijn Peppelman, Christina Kolb, and Mariëlle Stoelinga. 2023. Model-based joint analysis of safety and security:Survey and identification of gaps. *Computer Science Review* 50 (2023), 100597. https://doi.org/10.1016/j.cosrev.2023.100597

[20] Alfredo Oliveira, David Fiser, Ed Williams, Magno Logan, Mohamed Kmal, and Yossi Weizman. 2025. Brute Force. https://attack.mitre.org/techniques/T1110/.

[21] Ana Maria Pirca and Harjinder Singh Lallie. 2023. An empirical evaluation of the effectiveness of attack graphs and MITRE ATT&CK matrices in aiding cyber attack perception amongst decision-makers. *Computers & Security* 130 (2023), 103254. https://doi.org/10.1016/j.cose.2023.103254

[22] Ludovic Piètre-Cambacédès and Marc Bouissou. 2010. Beyond Attack Trees: Dynamic Security Modeling with Boolean Logic Driven Markov Processes (BDMP). In *2010 European Dependable Computing Conference*. Institute of Electrical and Electronics Engineers (IEEE), Valencia, 199–208. https://doi.org/10.1109/EDCC.2010.32

[23] Rishu Ranjan. 2021. Password Spraying Attack. https://owasp.org/www-community/attacks/Password_Spraying_Attack.

[24] R.C. Schank and R.P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures (1st ed.)*. Psychology Press, New York. https://doi.org/10.4324/9780203781036

[25] Nathan Daniel Schiele and Olga Gadyatskaya. 2021. A Novel Approach for Attack Tree to Attack Graph Transformation: Extended Version. arXiv:2110.02553 [cs.CR] https://arxiv.org/abs/2110.02553

[26] Bruce Schneier. 1999. Attack trees. *Dr. Dobb's journal* 24, 12 (1999), 21–29. https://www.schneier.com/academic/archives/1999/12/attack_trees.html

[27] Sander Vinberg, Jarrod Overson, Dan Woods, Shuman Ghosemajumder, Sara Boddy, Raymond Pompon, and Alexander Koritz. 2021. 2021 Credential Stuffing Report. https://www.f5.com/labs/articles/threat-intelligence/2021-credential-stuffing-report.

[28] Gert Jan Wijma, Karolien van Wijk, Michel van Kooten, Paul de Winden, Ronald van der Bie, and Sidney Vergouw. 2022. The Netherlands in Numbers. https://longreads.cbs.nl/the-netherlands-in-numbers-2022/how-many-people-fall-victim-to-cybercrime/.

[29] Ed Williams, Tim (Wadhwa-)Brown, Vincent Le Toux, and Yves Yonan. 2025. OS Credential Dumping. https://attack.mitre.org/techniques/T1003/.