Jurnal Teknik Informatika (JUTIF)
Vol. 4, No. 3, June 2023, pp. 529-553

DOI: https://doi.org/10.52436/1.jutif.2023.4.3.886
p-ISSN: 2723-3863
e-ISSN: 2723-3871

DECEPTION BASED TECHNIQUES AGAINST RANSOMWARES: A SYSTEMATIC REVIEW

Canny Siska Georgina[1], Farroh Sakinah[2], M. Ryan Fadholi[3], Setiadi Yazid[4], Wenni Syafitri[5]

[5]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia
[1,2,3,4]Faculty of Computer Science, Universitas Indonesia, Indonesia
Email: [1]canny.siska11@ui.ac.id, [2]farroh.sakinah@ui.ac.id, [3]m.ryan11@ui.ac.id, [4]setiadi@cs.ui.ac.id, [5]wenni20@gmail.com

(Article received: February 03, 2023; Revision: March 10, 2023; published: June 26, 2023)

Abstract

Ransomware is the most prevalent emerging business risk nowadays. It seriously affects business continuity and operations. According to Deloitte Cyber Security Landscape 2022, up to 4000 ransomware attacks occur daily, while the average number of days an organization takes to identify a breach is 191. Sophisticated cyber-attacks such as ransomware typically must go through multiple consecutive phases (initial foothold, network propagation, and action on objectives) before accomplishing its final objective. This study analyzed decoy-based solutions as an approach (detection, prevention, or mitigation) to overcome ransomware. A systematic literature review was conducted, in which the result has shown that deception-based techniques have given effective and significant performance against ransomware with minimal resources. It is also identified that contrary to general belief, deception techniques mainly involved in passive approaches (i.e., prevention, detection) possess other active capabilities such as ransomware traceback and obstruction (thwarting), file decryption, and decryption key recovery. Based on the literature review, several evaluation methods are also analyzed to measure the effectiveness of these deception-based techniques during the implementation process.

Keywords: *deception-based techniques, detection, mitigation, prevention, ransomware.*

1. INTRODUCTION

Ransomware has been one of the organizations' most feared cyberattacks in the last few years. It has proliferated from the frequency and severity of attacks, where the number of attacks more than doubled in 2021 to approximately 623 million. A single successful attack has cost companies $40 million in ransom payment [1], [2]. Ransomware is malware that aims to extract ransom, usually in the form of cryptocurrencies, in exchange for users' documents, pictures, and videos [3]. The attacker demands money from its victims through various means, such as locking essential or important computer functions, encrypting victims' files, and manipulating victims without restricting access [4]. After securing access to an environment, ransomware rapidly propagates through various means, such as software vulnerabilities and password cracking. Because of its propagating nature, successful ransomware attacks may cause widespread operational damage to an organization on top of financial losses [5].

Various studies on ransomware have been conducted in the last few years in response to its ever-increasing risk [6] [7], each attempting to categorize ransomware [8], [9], [10], understand the ransomware lifecycle [11], or develop possible ransomware detection and mitigation techniques [12]. Detection, prevention, and mitigation have been identified within ransomware-related research as part of overarching problems. This is because most of the proposed solutions could have been more specific and missed a lot of attack variations or more generic and ended up generating excessive false-positive alerts. Existing ransomware detection research has also been deemed insufficient to mitigate ransomware risks [5].

One of the most promising methods to overcome ransomware is to leverage deception-based techniques [13], [14] as a cyber defense. Defensive deception works on their perception by concealing the attack surface. The goal is to hide critical assets from attackers and disrupt or mislead them, thus making them waste resources, halting the attacks' impacts, and quickly disclosing the adversary tactics [15].

Defensive deception, distinguished by its ability to detect zero-day vulnerabilities and low false-positive rates, could serve as an additional layer of defense to mitigate ransomware issues.

Table 1. Research Questions And Each Corresponding Objective.

ID	*Research Question*	*Objective*
RQ1	**1.1.** What is the current approach/method using deception-based techniques for Ransomware detection/mitigation? **1.2.** What are the most recent and popular platforms used?	Discover the most recently used approach/methods to identify the trend in ransomware detection and mitigation techniques that utilize deception-based techniques. In addition, since ransomware is applied within several platforms/OS, our objective is to extract information regarding the distribution of the platform or OS used to provide a more comprehensive overview.
RQ2	**2.** What purpose (goal) does each approach/method/solution serve?	It is observed that most deception-based approaches, such as honeypots, are used for passive and preventative functions, i.e., monitoring activities/anomalies, information collection, and system vulnerability identification. The objective is to discover whether these deception-based techniques serve other purposes as well (such as reactive and even restorative functionality)
RQ3	**3.1.** What are the layers involved in each approach? **3.2.** What are the tools used/tested in the environment?	In order to extract detailed information for every deception/decoy-based approach/method, our objective is to dissect each one of them based on the following: 1. How it works, whether it tends to be implemented before or after the ransomware infection. 2. The layer in which the approach/method/technique is implemented (Data layer, system layer, network layer, etc.). 3. The tools used (e.g., module, algorithm, etc.) on the experiment setup.
RQ4	**4.** What are the proposed system's advantages, limitations, and future challenges?	Since every approach/method/solution is applied within different platforms and Operating Systems and tested against different types of ransomware, we aim to extract advantages, limitations, and potential future challenges for each corresponding approach/method /solution.
RQ5	**5.1.** How to evaluate the deception-based techniques used against ransomware?	In this question, we aim to analyze how effective deception-based strategies are when implemented and extract any measurement methods available in previous studies to evaluate their quality.

Motivation: Deception-based techniques, methods, and approaches have been a very active field of research over the years. Several previous secondary studies have discussed, reviewed, and classified decoy-based and deception-based techniques, methods, and approaches both in a general way [16], [17], [18] and in a specific way, i.e., honeypot [19], [20], [21], [22] honeynet (multiple interconnected honeypots) [23], honeytokens (for other forms of bait resources, e.g., files, password, id/accounts, database entries, vaults) [24], [25], [26], [27], and Moving Target Defense [28], [29], [30]. Despite that, the majority of the previous works were rather implemented to passively mitigate cyber threats (such as Intrusion Detection Systems, etc.) [31], limited to their static nature, and tested against malware in general [32], [33].

It is also identified that various meta-studies and sporadically thematic overviews of ransomware detection [34] [35], [36], [37], [38] [39], [40], [41], prevention [42], [43], and mitigation methods [44], [45], [46], [47]. Several relevant literature studies were even further refined into particular platforms and operating systems, such as Android [48], [49], [50], [51], Windows [52], IoT [53], [54], [55], etc., and specific type of ransomware such as crypto-ransomware [56], [57], locker-ransomware [58], [59], and scareware [60]. However, it is discovered that only a few of the recent paper surveys focused on the use of deception-based techniques specifically implemented against ransomware and how those deception-based techniques were evaluated. To the best of our knowledge, one of the pioneering research on deception-based ransomware detection was published in 2016. [61]. Since then, there has been considerable advancement and novel contributions in the common field.

Despite the importance of these studies, most of the discovered literature research is either focused on deception-based techniques for malware in general or ransomware solutions using common techniques. In other words, there is a need for another review covering the state-of-the-art deception-based solutions specifically designed to address the ransomware issue.

Contribution: Therefore, this study attempts to fill the gap by providing a comprehensive, methodological overview of state-of-the-art research done on ransomware prevention, detection, and mitigation using deception-based techniques that were published between 2016-2022. Our purpose is to collect all relevant studies on deception-based tactics, methodologies, and approaches specifically designed, used, and tested against ransomware and to provide answers to the research topics listed in Table 1. In this paper, we also present a review of any performance evaluation methods that correspond to each solution. This study aims to be a helpful reference and help fellow researchers and practitioners develop, utilize, and measure deception-based techniques on ransomware.

The rest of the paper is organized as follows. Section 2 describes the research methodology that was used. In Section 3, relevant background information is briefly described. Section 4 details the analysis and explanation for each research question. Section 5 presents the evaluation results to prove the effectiveness of deception-based solutions

2. METHODOLOGY

The purpose of the SLR is to discover any recently used deception-based approach/method; to observe each solution's served purposes and capabilities, i.e., how they are implemented and

tested across different platforms and various ransomware samples; and identify several evaluation methods to measure each solution's effectiveness. In this study, 19 shortlisted studies are taken into consideration.

This Systematic Literature Review (SLR) is carried out following Budgen and Brereton's [62], with several adjustments based on [63]. Figure 1 depicts the entire process of this SLR. Our literature review protocol is broken down into three stages: planning the review, performing the review, and reporting the review.

2.1. Search Strategy

Several search phrases were created to find papers on ransomware detection using decoy- or deception-based techniques. In order to combine search words, the fundamental strategy is to use Boolean expressions, which contain the operators "AND" and "OR." The search phrases is summarized as: (ransomware OR malware) AND (honey OR honey file OR moving target protection) and (detection OR detect). The relevant digital repositories are chosen after guaranteeing these search keywords. The following is a list of the five electronic databases used.

- Google Scholar
- ACM Digital Library
- IEEE Xplore Digital Library
- ScienceDirect
- SpringerLink

As mentioned, five electronic databases, which comprise the primary journals, conferences, and commercial items, are used for the search procedure. The period of searches is from 2016 to 2022, and all studies pertaining to search phrases are included.

2.2. Selection of studies

Inclusion criteria were also developed in the previously explained search process, based on five electronic databases, to pick relevant research. First, search phrases are included in the title, abstract, or keywords. Second, the research includes detecting strategies. Third, the study adopts decoy techniques such as honeypots, honey files, and moving target defense. In order to eliminate studies that are utterly irrelevant to the goal of this SLR, the following exclusion criteria are used:

- Studies (journals, conference papers, book chapters, etc.) that are not published in English are excluded.
- The adoption of deception-based techniques applied to malware, in general, is also filtered out. Only ones specifically for ransomware are included.

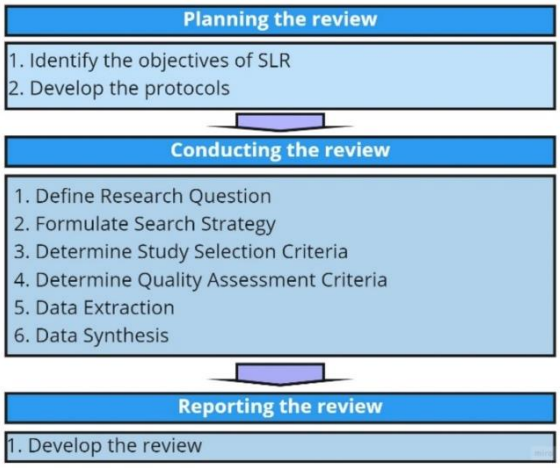

Fig.1. Systematic Literature Review Overview

- Studies describing ransomware detection, mitigation, etc. that utilize non-decoy/non-deception-based techniques and strategies are also excluded.
- In general, some studies are simultaneously published in conferences and journals. Therefore, less extensive studies with corresponding duplicate papers are excluded.

2.3. Quality Assessment Criteria

Table 2. Quality Assessment Criteria

No	*Research quality assessment criteria*
1	Are the study's objectives clearly defined?
2	Is each deception-based technique clearly stated?
3	Are ransomware detection/mitigation techniques clearly described?
4	Are the tools, platform, and OS used in each solution clearly stated and explained?
5	Does the study contribute to these SLRs?

In order to assess the quality of the chosen studies, the quality assessment criteria in Table 2 are applied. The cross-checking method is also used to determine whether the selected studies meet these criteria to ensure the reliability of the results. The actual studies are obtained after the quality assessment criteria, including 19 shortlisted studies related to decoy/deception-based techniques specifically tested against ransomware.

2.4. Data Extraction, Synthesis, and Reporting

The data extraction method involves creating forms to extract information from the primary studies. Answers to any research inquiries can be obtained based on the information in the data extraction forms. This study summarizes comparable outcomes from data extraction forms in the data synthesis process, which can give supporting evidence for decisive solutions to research questions. After the step of data synthesis, the complete results can be seen in the next section for each research question's corresponding answer.

3. LITERATURE REVIEW

3.1. Ransomware

While the first ransomware sample was found in 1989, real-world attacks involving ransomware techniques did not emerge until 2005 [5]. Ransomware is a type of malware that prevents victims from accessing their data until a ransom is paid. This malware has a direct financial impact, which has fueled an ecosystem of cybercriminals who utilize it as a business strategy [4]. Ransomware attacks adhere to a recognizable pattern that can be identified in each ransomware family and variant. Ransomware attacks are generally carried out in three stages: pre-encryption, encryption, and post-encryption. [34].

3.2. Deception/Decoy-based Techniques

Deception is an attempt to influence others' behavior by manipulating their beliefs [64]. The primary focus of cybersecurity is intrusion prevention and detection [65]. Defensive deception-based techniques are divided into honeypot, honeytokens, and moving target defense [16], [66]. Recent cyber-attacks have been demonstrated to be extremely sophisticated, highly customized, and carried out in stages. These multi-stage attacks typically involve several layers of penetration. To mitigate this, [67] introduces a multi-layer deception technique to examine the situation. As demonstrated in Figure 2, the ideas of honey persons, honey files, honey servers, and honey activities are developed as fake resources to assist the intrusion detection process.

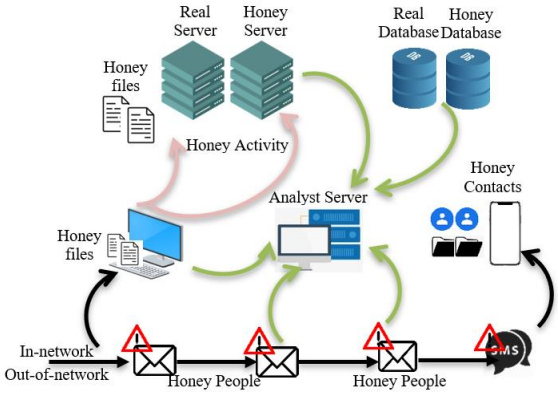

Fig. 2. Multi-Layered Deception System In [67]

3.3. Honeypot

Honeypot is an instrument to seize or grasp the attack methodology used by the attack [68]. The primary function of a honeypot is not to prevent or mitigate the attack but to disguise itself as a valuable environment or data to swindle the attacker. There are two categories of honeypots, i.e., research and production honeypots [69], [70], [71]. The research honeypot focuses on collecting attackers' information, while the production honeypot focuses on understanding system vulnerabilities and strengthening the defense system. Honeypot imitates numerous services based on their level of interaction. The honeypot system must also have files that make it appear to be a genuine server. In order to avoid these assaults, an appropriate safety mechanism is installed on the network. [72].

3.4. Honeyfiles

Honey file is trap files intentionally placed to be accessible by attackers. The honey file is designed with a persuasive title, such as "password", "username," and "userID". Once the attacker accesses a particular file, the system receives an alert [24]. This case is especially true with honey files, where the deceptions' content, look, and location are highly customizable. There are two primary metrics of interest: The first one is enticement, which measures how effective a honey file may capture an intruder's attention, and the second one is realism, which measures how convincing it is. Suppose it is assumed that an attacker will steal documents from an organizational digital repository. In that case, a honey file may be created with the repository search interface or file traversal pattern in mind. [64].

3.5. Honeytoken

The term “honeytoken” was first identified by Paes [73], which refers to any decoy user ID/accounts, documents, or information placed instead of honeypots. Simply put, a honey token is data that seems attractive to cyber criminals but is actually useless to them. Honey tokens are analogous to honeypots in this regard. While honeypots can be fictitious servers or other types of resources, honey tokens contain data taken by the attacker, inadvertently revealing information that helps IT teams prevent future attacks [74].

3.6. Cyber Kill Chain Framework

Lockheed Martin's [75] Cyber Kill Chain framework is based on a military model for identifying, preparing, attacking, and destroying the target. The cyber kill chain is used in attack modeling to identify various types of threats that organizations face, such as advanced attacks and ransomware. The phases of the cyber kill chain are as follows: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and goal-directed action [76].

4. RESULTS

This section aims to present the results obtained from primary studies. Primary studies are described first. Then, the SLR results are reported to address each research question. Overall, the amount of research on ransomware detection has expanded

considerably over the last few years. From 2016 to 2022, relevant studies on deception/decoy-based techniques/strategies have been collected and shortlisted. It can be seen from Table 3 that the number of studies from the conference paper/proceedings occupies more than 50%, followed by journals and commercial products (Canauri™).

4.1. RQ1.1: *What are the current approaches /methods that are using decoy-based or deception Based techniques for ransomware? How are they classified into the unit of deception?*

When deception-based approaches are deployed against ransomware, the primary goal is to detect, prevent, or mitigate those attacks [17]. In this context, a deception unit corresponds to the decoy asset's granularity used to implement the deception technique. The exact definition of granularity introduced in [77] was used, which includes the following deception units:

a) File (for example, decoy files, honey files [24]);
b) Service (for example, decoy service [78]);
c) Activity (for example, decoy computation activity [79]);
d) Weakness (for example, controlled vulnerability [80]);
e) User profile (for example, decoy ID, honey profiles [81]);
f) Decision (for example: allowing a connection towards a vacant IP address [82], [83]);
g) Configuration (for example, forged network topology [84]);
h) Response (for example, fake network response [82]
i) Account (for example, decoy/honey account [85], [86])

Classification is performed on each proposed approach /method/technique based on the unit of deception described above, as seen in Table 4. Several most frequently used deception units are File/Honey file (18,9%), Decision (18,9%), Configuration (17,2%), Service (13,8%), and account (10,3%).

Table 3. Shortlisted Studies

ID	Source	Title	Type	Year
[61]	IEEE	Detecting Ransomware with Honeypot Techniques	Conference Paper	2016
[96]	Science Direct	R-Locker: Thwarting Ransomware Action through a honey file-based approach	Journal	2017
[98]	DL-ACM	How to make efficient decoy files for ransomware detection?	Conference Paper	2017
[100]	DL-ACM	Paybreak: Defense against Cryptographic Ransomware	Conference Paper	2017
[89]	IEEE	Poster: A New Approach to detecting ransomware with deception	Report	2017
[90]	IEEE	UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware	Conference Paper	2017
[97]	Springer	RWGuard: A Real-Time Detection System Against Cryptographic Ransomware	Journal	2018
[101]	Google Scholar (Hindawi)	Automatically Traceback RDP-Based Targeted Ransomware Attacks	Journal	2018
[91]	IEEE	RansomWall: A layered Defense System against Cryptographic Ransomware attacks using machine learning	Conference Paper	2018
[95]	IEEE	RansomTracer: Exploiting Cyber Deception for Ransomware Tracing	Conference Paper	2018
[92]	IEEE	Ransomware Prediction using Supervised learning algorithm	Conference Paper	2019
[93]	IEEE	Malware Capturing and Analysis using Dionaea Honeypot	Conference Paper	2019
[105]	IEEE	Ransomware Honeypot: Honeypot solution designed to detect a ransomware infection and identify the ransomware family	Conference Paper	2019
[103]	IEEE	Design of intrusion Detection Honeypot Using Social Leopard Algorithm to detect IoT ransomware Attacks	Journal	2020
[104]	DL-ACM	SDN Hive: A Proof-of-Concept SDN and Honeypot System for defending against internal threats	Conference Paper	2021
[94]	DL-ACM	SODA: A System for Cyber Deception Orchestration and Automation	Conference Paper	2021
[102]	Google Scholar /Commercial	Canauri™ (Previously named CryptoStopper™)	Commercial Product	2022
[99]	Science Direct	R-Sentry: Deceptionbased ransomware detection using file access patterns	Journal	2022
[106]	IEEE	KRProtector: Detection and Files Protection for IoT Devices on Android Without ROOT Against Ransomware Based on Decoys	Journal	2022

Table 4. Ransomware Approaches/Methods/Techniques That Are Classified Into The Unit Of Deception

Ref	Approach/Method/Techniques	Unit of Deception
[61]	1. Detection: Honeyfile, Honeypot folder deployment, event log monitoring 2. Prevention & Mitigation: early warning e-mail, stop/shut down server services	▪ File ▪ Folder/Directories

		▪ Service
[96]	1. Detection: Honeyfile, Honey Archive, and Filesystem Activity Monitoring 2. Mitigation: Ransomware blocking when a honey file is accessed	▪ File
[98]	1. Detection: Decoyfile generation and placement based on source code analysis	▪ File
[100]	1. Detection: Develop RADDAR System to perform Real-time, automatic Ransomware discovery, detection, & alert 2. Mitigation: Ransomware key back-up storing, stored key usage for file decryption	▪ Service ▪ Activity ▪ Weakness ▪ Decision
[89]	1. Detection: Honeyfile generation and placement, API function altering, application behavior monitoring	▪ File
[90]	1. Detection: Artificial user environment generated, File system activity monitoring, network adjustment 2. Prevention & Mitigation: Desktop Lock Monitoring, dissimilarity testing, and text extraction from a screenshot	▪ File ▪ Service ▪ Activity ▪ Configuration
[97]	1. Detection: Decoyfile placement, File change, and Process Monitoring 2. Mitigation: decryption key recovery and the encrypted file restoration	▪ File
[101]	1. Detection: Deception environment generation (Network, hosts, Filesystem) and environment monitoring 2. Prevention & Mitigation: Ransomware trap and analysis, private information automatic & detailed analysis	▪ Service ▪ Activity ▪ Configuration ▪ Account/hosts
[91]	1. Detection: Honeyfile/Honey folders/directories placement and monitoring, executable files classification. 2. Mitigation: Static & dynamic analysis for trapped ransomware and back-up file creation for user data preservation	▪ File ▪ Folder/ Directories ▪ Decision
[95]	1. Detection: The deception environment is constructed to mimic the actual user environment (activities, functions, etc.) 2. Mitigation: Collect traceable information to be analyzed using natural language processing and machine learning technology that could trace those adversaries back,	▪ Service ▪ Activity ▪ Configuration
[92]	1. Detection: Honeypot trap files are placed to collect files from the ransomware-infected host system 2. Future Prevention: Dynamically analyze collected files and generate dataset report for ransomware further classification	▪ File ▪ Folder/ Directories ▪ Weakness
[93]	1. Detection: Low Interaction Honeypot deployed, malware and malicious activities/processes monitoring 2. Future prevention: ransomware is analyzed/studied and then classified based on its properties and activities.	▪ Service ▪ Activity ▪ Decision
[105]	1. Detection: Decoy files placement in low-cost & power embedded devices, deploy the simulated network, client operation monitoring, detection script usage in combination with Syslog	▪ File ▪ Decision ▪ Configuration
[103]	1. Detection: Honeyfolder and Honeyagents generation and placement, decoy file/folder change monitoring 2. Prevention & Mitigation: Firewall alert as early warning system triggered by honey agent, process kill if the threshold is met	▪ File ▪ Folder/ Directories ▪ Account/agent
[104]	1. Detection: Honeypot software (decoy network) and traffic analyzer placement, SDN-Hive system, application, & switch 2. Mitigation: performing a follow-up network scan for an identified suspicious IP address using Backscanner	▪ Service ▪ Decision ▪ Response ▪ Configuration
[94]	1. Detection: Detection agent generation and placement, Deception playbook creation. 2. Mitigation: Real-time deception using embedded API-hooking (deceive malware execution), Profile creation, Pre-built Profile Selection & relevant deception ploys are shown	▪ Service ▪ Decision ▪ Response ▪ Account ▪ User Profile
[102]	1. Detection: Strategic placement of decoy files and real-time detection when ransomware begins encrypting 2. Prevention & Mitigation: administrative alert & automatic shutdown for the infected workstation to prevent further network infection	▪ File ▪ Decision
[99]	1. Detection: Honeyfile generation and placement, file system activity monitoring 2. Mitigation: Ransomware block and notification for every trapped ransomware (for removal)	▪ File
[106]	1. Detection: Decoys (empty folders) are generated and deployed in a pre-determined location, then Decoys are distributed with sibling and sub-sibling decoys, then do folder monitoring. 2. Mitigation: Perform shortlisting on suspicious activities based on trusted values, then send an alert (if detected)	▪ Folder/Directories

Table 6. Preventative Approaches' Method & Techniques

No.	Method/Techniques & Purpose	Preventative method	Reference
1.	Honeyfile/Decoyfile used for ransomware. - Trapping - Tricking (obfuscation) - Analysis (common passive feature)	Honeyfiles/Decoy files are designed and placed strategically within the whole system/root system. Then the files/filesystem is monitored for any file change (entropy change, file extension changes, etc.). Some decoy files are generated automatically, and some are generated manually to serve their "lucrative" purpose of ransomware infection.	[102], [92], [89], [96], [90], [98], [61], [105], [91], [99].
2.	Honeypot, used for ransomware:	Plain vanilla honeypot is modified with an additional algorithm, Complex	[92], [103]

	- Analysis (capture logs, info, etc.)	Event Processing (for add-in services, e.g., traffic analyzer, backscanner),	
3.	Honey network used for ransomware: - Trapping - Tricking (obfuscation)	The deception environment was leveraged with decoy network services when using the decoy-based method in the network layer. This decoy network is mainly used for backtracing purposes, malicious traffic blocking/rerouting, and gaining clues regarding attackers' private information and identity.	[104], [101]
4.	Honeyfolder/directories/Decoy Folder used for ransomware. - Trapping - Tricking (obfuscation) - Analysis (common passive feature)	Fake directories/Honey folders are placed randomly and by design (based on the ransomware behavior) within the workstation and system root. Afterward, a specifically configured folder monitoring agent (manually developed or using other software: EventSentry, etc.) is assigned to monitor any file traversing/file access method used by the ransomware.	[103], [106]
5.	Decoy Hosts/system and user environment used for ransomware. - Tricking (obfuscation) - Analysis (common passive feature)	The host/System is integrated with other deception layers (ones in the network, filesystem, etc.). Thus this decoy host is equipped with a Machine Learning algorithm to extract meaningful information when ransomware infection starts (login info, clipboard content, folder path, PE file, etc.)	[93], [61], [101], [95]

4.2. RQ1.2: *What are the popular platforms in ransomware detection?*

Most of the ransomware infection occurred on Windows and Windows Server (91%), followed by other operating systems within a different type of devices, e.g., macOS (7%), Android (7%), iOS (4%), etc. [87]. This number is nothing new, considering several dominating ransomware families, such as WannaCry and Petya, for example, are infamous for their infection and propagation within Windows OS-based hosts.

In order to assess this issue, it can be observed in Table 5 that Windows dominates most of the systems/environments used for ransomware sample execution. Most of the research in ransomware-detection techniques required actual ransomware samples to be used. It is conducted to fairly evaluate the performance of their proposed solutions when faced with the current situation (actual ransomware-infected system).

Table 5. Ransomware Prevalence Across Operating Systems And Ransomware Types

No	*Operating System*	*Ransomware Type*	*%*
1.	Windows (7, 8, 8.1)	All types (Cryptographic, locker, and scareware)	68,4%
2.	Linux	Cryptographic	10,5%
3.	IoT, Cloud-based	Cryptographic	5,3%
4.	Android	Cryptographic, Locker	5,3%
5.	Not Specified	-	10,5%

Also, this has made it more practical and reasonable for professionals to understand since most ransomware incidents use Windows as their operating system.

4.3. RQ2: *What purpose does each approach /method/solution serve?*

Faced with ransomware variants that are dynamically evolving, so does the used mitigation techniques. As observed in [88], several studies have focused mainly on preventative and reactive methods. Most preventative approaches studied are detection, prevention, and defense techniques, while the identified reactive approaches focus more on ransomware obstruction, blocking, and post-ransomware recovery techniques.

- Preventative approach

The previously assessed observed studies have shown that deception-based techniques (honey files, honey directories, honeypot, decoy networks, obfuscation) worked exceptionally well in preventing the ransomware from damaging the system through real-time detection and well-integrated defense mechanisms. Several studies [89], [90], [91], [92], even imbued dynamic analysis (behavioral-based) into the proposed solutions to further analyze each ransomware's behavior and then classify them automatically using supervised Machine Learning techniques. The detailed methods, purposes and Techniques used for each purpose are detailed in Table 6.

- Reactive and Restorative approach

The old-fashioned perception about the deception-based method that can only perform passive protection, for example, honeypot only for passive monitoring, create log information based on traffic access, and so on, is proven to be quite different based on our literature studies. Several studies have shown the proposed solution's capabilities to impede, hinder, and block the ransomware's infection and further propagation using deception-based approaches. A few studies have proven specific capabilities on key restoration, thus resulting in the possibility of file decryption and recovery.

Several studies even leverage their contribution through innovative methods, e.g., plain honeypot combined with ransomware classification database [93], and integrated system orchestration for automatic and real-time reactions based on previously analyzed ransomware profiles [94]. Detailed methods, purposes, and techniques used within each purpose for reactive approaches are detailed in Table 7.

4.4. RQ3.1: *What are the layers involved in each approach?*

Deception techniques can be mapped further into the cyber kill chain model based on [16]. In order to do the mapping, the deception layer is first divided into four-layer categories, i.e., software/application, system, data, and network layer. After that, the used tools' layer operation, and

placement of honey files and honey folders, are compared and considered.

In order to assess this issue, it can be observed in our literature review that Windows dominates most of the systems/environments used for ransomware sample execution. Most of the research in ransomware-detection techniques required actual ransomware samples to be used. It is conducted to fairly evaluate the performance of their proposed solutions when faced with the current situation (actual ransomware-infected system). Also, this has made it more practical and reasonable for professionals to understand since most ransomware incidents use Windows as their operating system.

Table 7. Reactive/Restorative Approaches' Method & Techniques

No.	Method/Techniques & Purpose	Reactive/Restorative method	Reference
1.	Block Ransomware's communication to the C&C server used to: - Stop further infection.	For ransomware, communication to its control and command services (C&C server) is significant to worsen the damage within the infected system and network. It is performed by executing remote commands. Thus, blocking this communication will be a massive hindrance to the ransomware.	[103]
2.	Automatic Service/System Shut down used to: - Prevent network propagation. - Stop further infection	After giving an alert/early warning system, an infected workstation is usually shut down automatically to prevent further infection and propagation of the ransomware within the same network	[102], [61]
3.	Automatic Process Kill used to: - Prevent network propagation. - Stop further infection	Suppose the detection system identifies any suspicious or malicious-looking processes/activity found in the system. It is usually performed if certain threshold and characteristics of processes' is met.	[99], [103]
4.	File/Key Recovery used to: - Restore used keys. - Recover encrypted files.	Post ransomware infection's most significant concern (file decryption and encrypted file restoration) has made this method possible. Most of it is done by hooking Several Crypto API functions, Crypto++ library: Crypto Function Hooking to export session key & algorithm parameters, and manually created modules/functions in the proposed system.	[100], [97]
5.	Traffic monitoring analysis, rerouting, blacklisting and resource obfuscation used to: - Prevent network propagation. - Track the attack's source. - Waste ransomware's resources.	If the ransomware is trying to infect and propagate within the whole system's network, this method leverages the OpenvSwitch to support OpenFlow protocol for issuing blocking rules based on detected malicious activities, combined with the Honeypot Traffic analyzer in order to process and analyze network packet even to do backscanning (a follow-up network scan for identified suspicious IP address)	[104]

Table 8. Deception Layer [16]

No.	Layer of Deception	Reference
1.	Application/Software	[95]
2.	System	[96], [97], [98], [99], [100], [89], [101], [91], [90], [102], [92], [103], [104], [93], [94], [105], [106], [95]
3.	Data	[105], [94], [90], [91], [89], [101], [100], [95]
4.	Network	[61], [101], [90], [104], [94], [105], [95]

While the application layer encompasses deception techniques associated with specific classes of applications, i.e., web applications or databases, the system layer is more responsible for host-based deception techniques. On the other hand, the data layer addresses deception techniques that use user-specific data, such as bogus accounts or documents. The final layer, the network layer, includes decoy-based techniques available over the network and not tied to any particular user configuration.

Deception-based techniques often deal with distinguished potential threats within each layer. Hence, it is essential to consider those threats. In the application layer, they are web attacks and software vulnerability/compromise, while in the network layer: eavesdropping, scanning & fingerprinting, infiltration & attack propagation are ones to be anticipated. Table 8 shows that the system layer is the highest-used layer for deception or decoy-based techniques for ransomware mitigation/detection. Consequently, insider and external threats and attacks must be anticipated, especially in the system layer. As for the last layer (data), privacy violation, data breach/leak, identity theft & impersonation are several threats that need to be considered.

4.5. RQ3.2: *What tools are used/tested in the environment?*

In order to answer this research question, there is a need first to identify whether several proposed solutions (techniques/approaches/methods) are utilizing other modules/tools from other software (both commercial and open source, if any). It can be observed that most of the proposed solutions are generally implemented through these three categories:

- Developing novel software/systems, such as Paybreak, SODA, Canauri™, R-sentry, RWGuard, RansomTracer, and Deception Environment Prototype.
- Utilizing the existing system, software, or modules, i.e., Microsoft FSRM®, Windows Event logs & Eventsentry,
- Combining the newly developed software with existing software/modules, i.e., Wine and R-Locker, Microsoft Crypto APIs & Crypto++ library and RADDAR, Cuckoo Sandbox and UNVEIL, Windows Virtual Machine and Ransomwall, Amazon EC2 Cloud infrastructure, and Dioanea honeypot, and

lastly, SDN-Hive and Honeypot software/traffic analyzer.

4.6. RQ4: *What are the proposed system's advantages, limitations, and future challenges?*

In this section, the beneficial features of each technique have been highlighted, as seen in Table 9. There are a few substantial advantages, such as good performance under low system resource consumption (minimal usage on CPU, memory, and disk usage or load); thus, it can be used for everyday office workloads; high detection rate with very low (near zero) false negatives and false positives; lightweight, real-time detection combined with a rapid and responsive early warning system (alert) and effective countermeasures (process kill, ransomware trap, restriction, and blockage); Lastly, capabilities for encrypted file and decryption-key restoration.

Table 9. Advantages Of Each Deception Solution

No	Advantages	Reference
1	System activities/service interruption, process kill	[97], [61], [94]
2	Minimal overhead, high detection rate, and rapid countermeasures launched	[92], [96], [98]
3	Very minimal (even zero) false negatives/positives	[96], [90], [97], [105], [91]
4	Automatic, Effective, and Responsive Early Warnings/Alerts	[61], [103], [102]
5	Real-time ransomware activities blocking	[93], [96], [89], [104]
6	Novel Strategies, i.e., Honeyfiles /Honeyfolders Placement strategy, File restoration, Key recovery, new obstruction strategy: ransomware thwarting	[98], [99], [101]

On the other hand, several proposed approaches, techniques, and strategies have several limitations. Constraints such as inadequate testing environment (e.g., limited platform, homogenous Operating system, small ransomware families/sample, numerous-conditions environment setup, etc.) can still be seen in [98], [101], [89]. Other proposed methods, such as: [100], could not withstand all obfuscation attacks. At the same time, [99] has yet to be tested in a real-world setting and may be partially circumvented by acquiring access randomly.

Although commercial software [102] could help provide signatureless and real-time ransomware detection and fast post-attack action, false positive out of the detection process is still expected. The remaining limitations were mainly caused by the lack of exploration within ransomware propagation techniques, common protocol and behavior [104], the possibilities of triggering false positives, and the inability of the techniques to deceive the ransomware [94]. In addition to that, the potential existence of "decoy-aware" ransomware [107], which has the capabilities to identify static decoy files and utilizes the strategy of blacklisting and whitelisting user files before encrypting them, will also have to be taken into consideration in the future challenge.

4.7. RQ5: *How to evaluate the deception-based techniques used against ransomware?*

In order to answer this question, relevant studies on deception-based techniques' performance evaluation are explored. Several essential aspects of evaluation methods, such as the ones proposed by Genc. et al. [107]. Their approach is to measure the following things:

- "Deceitfulness" level of a decoy strategy (decoy strategy's quality measurement) against ransomware. This is done by calculating the probability of ransomware encrypting n other files (some genuine user files) before encrypting decoy files. Thus, the smaller, the better.
- Decoy strategy's usability (confoundedness). It is done by calculating the probability of a user accessing a file during a working session. A decoy strategy is considered "usable" enough when the user's probability of accessing the Decoy file is small.

Table 10. Deception-Based Techniques Evaluation Method

Ref	Evaluation Method	General/Specific	Targeted method	Purpose
[107]	Quantitative approach (probability calculation): • Decoy strategy's quality measurements • Confoundedness measurement	Specific (ransomware)	Honeyfile, Decoy file, trap files	Analyze and test the robustness of decoy strategies
[106]	Quantitative approach (evaluation matrix): • Detection accuracy against data sets • Timeliness measurement	Specific (ransomware)	Honeyfolder, Decoy/ empty folders	Evaluate effectiveness (duration & accuracy rate)
[110]	Formulate precise measurements on false negative and false positive rates when using deception to detect	General (Malware)	MTD (Moving Target Defense)	Evaluate accuracy
[111]	Quantitative approach (probability calculation): • Quantify the effectiveness of cyber deception using Dynamic Bayesian Attack Graph • Assess concealment and effectiveness indexes	General (Malware)	Network deception and general cyber deception	Evaluate cyber deception performance and effectiveness

Another evaluation method proposed by Wang., et al. [106] emphasizes more on accuracy and efficiency aspects. Detection accuracy is obtained by calculating The True Positive, False

Positive, False Negative, and True Negative, while the efficiency is measured by calculating the total duration out of 6-phases: filtering time, obtaining time, tracking time, warning time, accessing the first file time, and encrypting the first file time.

However, it can be seen from Table 10 that the literature studies on evaluating deception-based techniques, approaches, and strategies are very few, especially those focusing on combatting ransomware.

5. DISCUSSIONS

It is observed that on several methods, the multi-layer deception concept is adapted, implemented, tested, and evaluated, especially for ransomware. Several solutions are even specifically crafted based on lessons learned from previous incidents caused by multiple stages of penetration through trapping and automated analysis; mimicking user environment comprehensively (network/configuration, activity, service, hosts/account, filesystem); providing not only strategically placed (through ransomware's source code analysis), but also multi-purposes honey trap files (used for both suspicious activities/file changes monitoring, ransomware obfuscation, and file collection from ransomware infected system); imbuing novel functions, i.e., post-incident dynamic analysis (analyze and trace back the attack source), recovery of encrypted file & key, and ransomware thwarting through resource wasting, C&C communication blocking, and even triggering automatic process kill to prevent worse infection/propagation).

According to VirusTotal's Report on Ransomware in a global context [108], 130 different ransomware strains have been identified. Of those strains, the Cryptographic ransomware family (GandCrab) is the most prevalent (78,5%). In addition, most of the ransomware samples (~95%) were identified as Windows-based executable files and DLLs. Since all proposed solutions are mostly tested against Cryptographic Ransomware as the chosen strain and evaluated within the Windows Platform as the major Platform/OS, most of the reviewed approach is considered relevant, practical, and adaptable. It can be observed that not all proposed solutions are developed from scratch; several solutions use available tools, while the rest are combined from existing tools and a few adjustments and modifications. Specific details for each tool utilized can be found in Table 11.

As observed on the implementation timing, each solution can generally be divided into pre- and post-incident actions. Pre-incident ones utilize essential, conventional deception functions such as obfuscation, monitoring, and alerting. However, it has been observed that novel functions are added in the form of traffic blocking and other deception layers' (network, filesystem) involvement for automatic, dynamic, and more meaningful information extraction. On the other hand, post-incident ones are leveraged by using several Crypto API functions and libraries to export session key and algorithm parameters, thus making key and encrypted file restoration possible. They are also based on the lesson-learned activities of the ransomware propagation method, such as traffic rerouting and blocking the communication to Control and Command services to prevent further infection. Regardless of its advantages, each proposed solution has certain limitations, which are explained in detail in Table 12.

6. CONCLUSION AND FUTURE WORKS

This paper summarizes and analyzes the state-of-the-art deception-based solutions and provides a comprehensive overview of research on ransomware detection, prevention, and mitigation using any deception-based techniques, approaches, and strategies published between 2016-2022. This literature review investigates each solution further based on the categories of deception techniques, i.e., the purpose; deception unit and layer; used tools during the experiment; the environment (platform and operating system); how to evaluate the deception's performance; lastly, the advantages, limitation, and potential future challenge within the field.

According to this SLR, it is found that (1) The mostly used unit of deception asset is: File (strategic placement and distribution of decoy file), Decision (fake permissive connection), and configuration (simulated network); Windows is the most used platform for utilization and evaluation of solutions. Windows is a popular target of ransomware variants; thus, having this platform for testing and evaluation performance of the solution is reasonable; (2) For each solution, both preventative and reactive (even restorative) functions serve as the purpose. (3) All deception has its representative solution. However, the system layer was the most frequently used layer to combat ransomware; (4) By studying and comparing existing solutions, we arrive at the preliminary conclusion that the advantages and drawbacks of each strategy, approach, and technique are continuously optimized and customizable to the objective of deception.

Nevertheless, the existence of deception evasion attacks, such as traffic and system fingerprinting [109], could go as far as enabling adversaries to precisely differentiates decoys from the real servers; anti-decoy strategies using heuristics in order not to encrypt files considered as a decoy, and encrypt file perceived as user files. This study [107] precisely demonstrated that numerous previous deception-based recommended solutions might be readily countered with the help of statistics, user behavior monitoring, file access pattern, and simple ransomware redesigning. From the last

research question's answer, it is also seen that deception techniques are not prone to emerging threats, thus emphasizing the need to assess its own strategy's effectiveness from time to time. In order to address this issue, a guideline for evaluating the performance of various strategies in ransomware detection tools and procedures is required.

7. LIMITATIONS

One of the difficulties faced during this systematic review is the limited primary studies to be reviewed in the paper. Relevant research from five electronic databases' journals and conferences have been collected as much as possible using the keyword provided. However, some relevant papers may still need to be included in our collected studies. Nevertheless, this paper's research contribution on methodology, process, and step-by-step analysis is reproducible for other fellow researchers, and just in case any other contribution is made to the research field related to the usage of deception-based solutions against ransomware.

Table 11. Used Tools/Modules/Algorithm & Its Usage On Experiment Setup

Ref	Tools/Modules/Algorithm and its function (if any)
[61]	- Microsoft File Server Resource Manager (FSRM) - Windows Event Logs, EventSentry: to monitor event log changes - EventSentry: To send early warning e-mail, stop server services, and services shutdown
[96]	- R-Locker: To deploy honey files around the environment of the target (based on depth-first file traversal pattern) - R-Locker: To monitor events in the filesystem - R-Locker: To block ransomware, notify the malicious event, and automatically deploy countermeasures to solve threats. - Wine: to test Wannacry Ransomware sample (in Unix distro)
[98]	- Manually identify ransomware's file search algorithm, file/folder traversal pattern, and file encryption method - Automatically generate decoy files using the given algorithm and tweak it with the extracted ransomware behavior
[100]	- Developed RADDAR System: Perform Real-time and automatic Ransomware discovery, detection, and alert. - RADDAR System: Obtain malware/ransomware samples from VirusTotal or various locations for other samples. - RADDAR System: Identify malware sample (crypto-ransomware or not/executing malicious process or not). - Cuckoo Sandbox and Windows 7 VM: to sandbox, analyze and output a behavior report on each sample - Microsoft's Crypto APIs & Crypto++ library: Crypto Function Hooking: export session key & algorithm parameters - Paybreak Key Vault: for key material & algorithm details (recovered from hooked procedures) - Paybreak File recovery: use key material and algorithm to attempt recovery
[89]	- Manually create decoy files (common target of ransomware, e.g., txt, doc, pdf) - Manually place decoy files in every directory; Windows API: redefine FindFirstFile API & FindNextFile API function - Designed Monitor Module: observe the behavior of the suspicious application when it operates decoy files.
[90]	- UNVEIL: Generate valid content (use standard lib: python-docx, OpenSSL), file extension, File Path & Time Attribute. - UNVEIL: monitor filesystem access (I/O monitor & logs retrieval using Windows Filesystem Minifilter Driver framework). - Cuckoo Sandbox & VM: to set environment, make sample submission, manage some VMs, simulate user input: clicking / cursor movement, etc. Configure the network, and set the IP address range & MAC address to prevent VMs' fingerprints from being recognized. - UNVEIL: capture screenshot outside dynamic analysis to prevent potential tampering - UNVEIL: use a python script to implement the Structural Similarity Image Metric (SSIM) to test dissimilarity - UNVEIL: use Tesseract-OCR (open-source OCR engine) to extract text from the selected areas of the screenshots
[97]	- RWGuard Decoy File Generator & Dmon Module: To Generate and deploy decoy files to be monitored - RWGuard PMon Module: To Monitor running processes/IRP. - RWGuard FCMon Module: Check the file system for malicious activity/file change anomaly. - RWGuard CHFk Module: relevant Crypto-API function hooking for decryption key & other parameter storage.
[101]	- Manually create the 3-layer Cyberdeception environment (Windows-based deception environment prototype) - Environment monitor: detect RDP-based ransomware attacks in time and determine the attacker's behavior. - NLP and Machine Learning equipped System: Extract login information, clipboard content, folder path, PE file, etc - Deception environment Prototype: trap the ransomware attacker and collect a lot of information - Deception environment Prototype: Capture traceable clues/info & analyze both. - Deception environment Prototype: Generate a report to do traceback the attacker (from previous clues & analysis)
[91]	-RansomWall 1st layer: IDA tool: Perform Static Reverse Engineering to cryptographic ransomware samples Collected from VirusShare; - Ransomwall 2nd layer: Sets trap by tracking the occurrence of malicious activities. Honey Files and Honey Directories are deployed in critical user data folders. Modification on these (files/directories) indicate suspicious behavior. - Ransomwall 3rd layer: Monitor file system operations and entropy modifications to track massive encryption activities. - Ransomwall 4th layer: files modified by a suspicious process are backed up in a separate folder. - Ransomwall 5th layer: Use Supervised algorithm to classify executables (benign/malicious) - Cuckoo Sandbox and Windows 7 & 8.1 VM: to sandbox, execute tools and instructions per each Ransomwall layer
[95]	- RansomTracer: create an artificial, realistic, and enticing user environment for the RDP attack ransomware - RansomTracer: identify and collect traceable clues from the decoy user environment - RansomTracer: extract and analyze discovered traceable clues
[92]	- Manually obtain ransomware samples from strategic web compromise, drive-by download, email-phishing, & vulnerability exploit. - Honeypot trap file/honey file: If one of the host systems is infected with ransomware, will collect the files to be dynamically analyzed (generated data set report is in CSV format) - Utilize supervised ML algorithms e.g., Support Vector Machine, Random Forest, Decision Tree, Bayesian Network, Artificial Neural Network, & Linear Regression to do the classification (distinguishing between goodware & ransomware)
[93]	- Amazon EC2: Create the Amazon EC2 instance (instance launched and OS selected), process terminal in Amazon EC2 - Dionaea honeypot: Installed and run on Ubuntu 14.04 (logs are stored in /var/log/dionaea and var/lib/dionaea/directory."

Ref	Tools/Modules/Algorithm and its function (if any)
	- Proposed combination honeypot: Observe the mostly attacked protocol (SQL 2000XP, SMB, & FTP). - Proposed combination honeypot: Categorize sample malware based on behavior (3 high-severity malware): Wannacry ransomware: attack SMB protocol in Windows OS, Slammer: attack SQL 2000 XP version of Windows, GandCrab ransomware
[105]	- Manually Deploying honey file in a simulated network share using Docker Technology - Manually configuring Samba in combination with Syslog to monitor all client-related operations for all individual users. - Manually created detection script for detection (based on information from existing file share's files, Samba service, & logs
[103]	- Honeyfolder: randomly generated, SoLA-modeled, & installed in every host. - Honey agents: to monitor decoy folders & act as an early warning system to alert the firewall (do process kill if a certain threshold is met) - Intrusion Detection Honeypot's Audit Watch: to monitor file/folder changes - Intrusion Detection Honeypot's Complex Event Processing (CEP) taken out of: hosts/networks, honeypot agent, SDN controller, Audit Watch, logs, etc.) - Intrusion Detection Honeypot's Complex Event Processing (CEP): block ransomware communication
[104]	- SDN-Hive System components 1. SDN controller: ONOS (as production-ready OS to develop & deploy SDN app via its API. - SDN Application: to protect the host from malware. Consists of: SMB & ARP scan detection, DNS, IP & MAC address Blacklist. - SDN Switch: OpenvSwitch to support OpenFlow protocol (conduct issued blocking rules based on detected malicious activities. - Honeypot Traffic analyzer: to process and analyze network packet - Honeypot software: decoy network service to monitor malware that successfully barged its way - Backscanner: a follow-up network scan for identified suspicious IP address with three submodules: Nmap, Celery, & RabbitMQ
[94]	- SODA: create deception playbook: Malicious Subgraph (MSG) Extraction, MSG Classifier, Deception Factory Synthesis - SODA Detection agent: acts as an entry point to detect malware & trigger the orchestration. - SODA Orchestration Engine Server (OES), Orchestration Engine Client (OEC), Detection Agent, and HoneyFactory (HF). Procedures executed Real-time orchestration. - SODA Real-time deception using embedded API-hooking (deceive malware execution). - SODA Profile creation (to OES via UI), SODA: Pre-built Profile Selection: for each selected pre-built profile, relevant deception ploys are shown
[102]	- Canauri: running ransomware samples executables, automatically alert the user through a dialogue box - Canauri: system alert shows the location of several files' extensions and directories enumerated/changed (by malware /ransomware) - Canauri: Administrative alert is sent to the security team with the attack's critical detail. Folder/file enumerated differently based on different file traversal/file access patterns (the very first directory enumerated will be shown in the alert dialogue box). - Canauri: System shutdown is performed to prevent further infection and propagation of the ransomware within the same network
[99]	- Honeyfiles generated based on different categories of file attributes (manually created using common software) - R-Sentry System: to place honey files in optimal location based on file access/traversal pattern analysis - File monitor: used to monitor the sequence of every accessed file's path and actions taken in the entire user directory. - File monitor: Detect ransomware samples that access honey files placed on every root folder. - Detected Ransomware was then blocked and notified to admin for removal
[106]	- KRProtector decoy deployment module: generate & deploy decoys based on the files' distribution, including sibling directory decoys & subdirectory decoys (decoy's entity: empty folders). - KRProtector decoy deployment module: use "userfolder", "leaffolder", & "subsibling" decoys to represent folders, user folder, etc. - KRProtector decoy deployment module: If decoys are accessed, it filters all active apps based on the existing 4-rules & triggers the detection module - KRProtector ransomware detection module: use trusted value metrics (the lowest positive number is regarded as crypto-ransomware) - KRProtector ransomware detection module: Shortlist the suspicious app, discover the origin of suspicious behavior, and inform the user (alert)

Table 12. Advantages And Limitations

Ref	Advantages	Limitations
[61]	- Network activities are interrupted when a certain activity level is identified - Network services were also stopped (in 6 seconds) when a certain activity level was identified - Early warning sent to system admin via e-mail	- Only tested on a simulated script, no actual ransomware involved - No guarantee the malware would attempt to invade these areas (honeypot folders), therefore bypassing this defense
[96]	- Ransom operation is completely blocked when the trap file is accessed. - Countermeasures are automatically launched to solve the infection without affecting the system's normal operation - Complexity and overhead involved in the solution are really low - 100% detection accuracy and null damage because of the honey file effectiveness and immediate block - Simple and autonomous execution without any additional processes to complement its functionality	- Defense can be bypassed if the ransomware is randomly accessing files. - Worst case scenario, the sample/honey (files) can be blocked after encrypting other files first. - Have yet to be implemented/tested in other platforms, e.g., Windows and Android.
[98]	- Novel approach on efficient method in generating decoy files for Ransomware Detection	- Not all Ransomware behavior were analyzed. The proposed method may not efficiently detect all

Ref	Advantages	Limitations
	- New countermeasure is designed not only for existing but also for possible future ransomware	ransomware variation
[100]	- File Restoration Effectiveness evaluated against 107 ransomware samples from ~20 families (Locky, Cryptolocker, Samsam) - Protection tasks can be performed at trivial performance overhead for everyday office workload	- Does not resist all obfuscation threats - Cannot provide guarantees against malware specifically aiming to evade Paybreak - The integrity of the escrow key stored in the key vault is susceptible to a Denial of Service attack - Key stored in the vault can be corrupted or filled with nonsensical information.
[89]	- Processes have to operate decoy files first before the real ones and can only be operated after the processes pass the detection phase. - The proposed detection approach has proven to be effective in identifying the encryption process and stopping it in a timely manner. - Low on CPU, memory, and disk usage/load and works within tolerable time delays.	- Not yet tested on different kinds of ransomware - Not yet tested in a normal environment (close to unmalicious activities) to see if the approach impedes the user's normal usage.
[90]	- The evaluation is performed at a large scale (13,637 ransomware samples out of 148,223 recent general malware). - The proposed techniques work well in practice (True positive rates at 96,3% and zero false positives)	- Ransomware could remain undetected if ransomware runs at the kernel level & thwart some hooks used for filesystem monitor. - The attacker's ability to fingerprint the dynamic analysis environment is always possible. It can prevent dynamic analysis from being done.
[97]	- Effective in real-time detection of ransomware: zero false negative & negligible false positive (~0.1%) rates - Robust decoy design combined with obfuscation techniques can impede ransomware when trying to find out the decoy generator app. - Certain modules (CFHk) can retrieve parameters to specific crypto function calls & restore the encrypted files. - All modules implemented are incurring an overhead of only ~1.9%.	- Several modules are theoretically subject to missing some of the malicious activities. - Time lag between logging and parsing activities for the anomalies creates a small "vulnerable" window for ransomware
[101]	- Clue capture and analysis worked well (data on login info, clipboard content, shared folder path, uploaded PE files collected). - Traceable clues remain (as planned) after some evaluation process involving 122 volunteers provided with 12 virtual hosts. - Detailed analysis is proven helpful for automatic analysis system and display traceable information about the attacker.	- Tests against ransomware families other than the RDP-based ransomware are not presented (limited experiment samples)
[91]	- Effective approach with insignificant spatial cost and system resource consumption - Tested against 574 Crypto ransomware families samples in a real-world environment with a detection rate of 98,25% & near-0 false positive - Can detect 30 0-day intrusion samples.	- Ransomwall has not been evaluated on a large-scale, real setup
[95]	- RansomTracer is able to ensnare the attacker and collect traceable clues left by the attacker in a deception environment - Automated clue identification enables the user to converge the number of traceable clues to about 2%. - Tracing back the ransomware attackers provide a good deterrent to adversaries, thus stifling the development of ransomware.	- Only tested against RDP-based ransomware families, thus needing more various testing environments.
[92]	- Data set for classification in the data repository is heterogeneous (942 good ware & 582 ransomware samples out of 11 different families) - The detection rate within every algorithm is all right (>50%); the highest accuracy is the SVM algorithm out of the six algorithms proposed.	- Proposed solution focuses more on dynamic analysis & machine learning methods for ransomware detection (not deception). - The 11 ransomware claims are not listed/detailed in the paper. - Honeypot is used passively only for ransomware information collection medium.
[93]	- Proposed honeypot can detect three high-profile ransomware and various categories of malware.	- Only for low interaction honeypot
[105]	- Decoy files performance is evaluated against WannaCry & Stampado in a development environment composed of an "infected-VM-system." - The proposed system can also reduce the False positive rate by using an entropy calculation check. (Entropy calculation can help distinguish between the normal/slight "user-modification" and the "malicious encryption").	- Less effective when faced with other types of ransomware - For example, MBR (Master Boot Record) Ransomware can restart the infected computer.
[103]	- Effective in restricting ransomware activity (tested against 20 recent ransomware variants). - Proposed IDH improves ransomware detection time and rate.	- Optimizing loads functionality for IoT devices (e.g., auto-tabling and transfers learning) is not available
[104]	- Improvement of the previous SDN-based solution (combination of SDN and Honeypot-based protection) to defend ransomware works. - SDN-Hive has proven its capabilities in protecting devices from	- Have not explored much about various propagation techniques used by other malware - Other common protocols, e.g., Telnet and SSH, have yet to be explored.

Ref	Advantages	Limitations
	worms & worms-like ransomware via: - a. Identification of the worm component by its unique characteristics - b. Detection of malicious activities and generation of blocking rules to OpenFlow switches (in the network) - c. ARP & SMB detection function could prevent vulnerable hosts' infection in the same LAN (because of Wannacry). - d. An external module is deployed to detect various scans & alert the SDN controller	- Bruteforce-based behavior of worms has not been considered.
[94]	- For testing against ransomware, 96 samples were used, and 11 distinct malicious behaviors were observed to identify 28 valid deception ploys. Those 28 deception ploys were deployed, & SODA could deceive the ransomware using 27. - Based on the Performance measurement and comparison to Cuckoo sandbox and Any.run, SODA has better coverage. It outperforms other existing tools in identifying ransomware (Ryuk and GandCrab Family) capabilities & presenting them in the MITRE ATT&CK framework.	- Relies on existing malware detection approach (can sometimes trigger false positives). - The possibilities of API hooking detection and evasion by the malware will make SODA unable to deceive the malware
[102]	- Canauri™ does not require additional hardware to maintain. It is claimed that Canauri™ protects the system in a matter of minutes - Canauri™ also takes fast action after the attacks (administrative alert and system shutdown) - Unlike Anti-Malware, Canauri™ is signature-less. It will always detect the activity of encryption rather than detecting - The specific variant of ransomware running.	- It is a paid solution (as commercial software) - Although there is a money-back guarantee and claimed to be minimal, false positive out of the detection process is still expected
[99]	- Lightweight & real-time solution of placing honey files in multiple locations (previous anti-ransomware solution only placed on root folder) - Work properly during the execution of different ransomware samples (24~ families of crypto-ransomware used in the experiment) - Honey file placement strategy is based on the ransomware analysis (file traversal/access pattern on as-is & nextgen ransomware families)	- Can be partially bypassed by gaining access randomly - Honeyfile is still generated manually (further research can extend this to automatic generation) - Proposed work still needs to be tested in a real-world scenario.
[106]	- Relatively low storage consumption (36 user folder and 5517 sub-sibling decoys only takes around 22,14-44.29 MB) - Highest detection accuracy (96,2%) with negligible usage of computing resources (trusted value calculation's time complexity is O(n).) - Very fast detection and responsive early warning (alert), thus preventing the ransomware from encrypting the files - Have the capabilities to identify wider variations of ransomware detection (tested along ransomware with obfuscation, steganography, & dynamic malicious code)	- Difficulty in sensing the behavior of ransomware without ROOT or administrator privilege (the proposed method needs root to monitor status changes of decoys in order to activate the detection module within KRProtector) - Limited to Android, has not been tested yet on other platforms/OS such as Linux - Linux is known to be able to provide file system mechanisms (monitor file/folder without ROOT); thus, it can be explored in the future

REFERENCES

[1] SonicWall, "2022 SonicWall Cyber Threat Report," 2022. [Online]. Available: https://www.infopoint-security.de/media/2022-sonicwall-cyber-threat-report.pdf. [Accessed 12 12 2022].

K. Mehrotra and W. Turton, "CNA Financial Paid $40 Million in Ransom After March Cyberattack," Bloomberg Law, 21 Mei 2021. [Online]. Available: https://news.bloomberglaw.com/insurance/cna-financial-paid-40-million-in-ransom-after-march-cyberattack. [Accessed 10 12 2022]

[3] Australian National Cyber Security Centre, "Ransomware: Measures for preventing, limiting and recovering from a ransomware," Juni 2020. [Online]. Available: https://english.ncsc.nl/binaries/ncsc-en/documenten/factsheets/2020/june/30/factsheet-ransomware/71059_NCSC_FS+Ransomeware+EN_WEB.pdf. [Accessed 9 12 2022].

[4] S. H. Kok, A. Abdullah, N. Jhanjhi and M. Supramaniam, "Ransomware, Threat and Detection Techniques: A Review," *International Journal of Computer Science and Network Security*, vol. 19, no. 2, pp. 136-146, 2019.

[5] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Computers & Security*, 2021.

[6] L. Y. Connoly , D. S. Wall, M. Lang and B. Oddson, "An empirical study of ransomware attacks on organizations: an assessment of severity and salient factors affecting vulnerability," *Journal of Cybersecurity*, vol. 6, no. 1, pp. 1-18, 2020.

[7] Restore Record Management, "Ransomware – How safe is your organisation from this ever

increasing threat?," Restore Record Management, 14 Juni 2021. [Online]. Available: https://www.restore.co.uk/Records/Resource-Hub/News/ransomware-how-safe-is-your-organisation-from-this-ever-increasing-threat. [Accessed 15 Desember 2022].

[8] H. Madani, N. Ouerdi, A. Boumesaoud and A. Azizi, "Classification of ransomware using different types of neural networks," *Science Reports*, vol. 12, no. 4770, 2022.

[9] A. M. Maigida, S. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma and E. G. Dada, "Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms," *Journal of Reliable Intelligent Environments*, vol. 5, pp. 67-89, 2019.

[10] B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki and M. Azer, "A New Scheme for Ransomware Classification and Clustering Using Static Features," *MDPI Electronics Journal*, vol. 11, no. 20, pp. 1-26, 2022.

[11] T. Dargahi, A. Dehghantanha and P. Bahrami, "A Cyber-Kill-Chain based taxonomy of crypto-ransomware features," *Journal of Computer Virology and Hacking Techniques* , vol. 15, pp. 277-305, 2019.

[12] R. Moussaileb, N. B. Cuppens, J.-L. Lanet and H. Le Bouder, "A Survey On Windows-Based Ransomware Taxonomy And Detection Mechanisms: Case Closed?," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1-36, 2022.

[13] J. Papalitsas, S. Rauti, J. Tammi and V. Leppanen, A Honeypot Proxy Framework for Deceiving Attackers with Fabricated Content, Springer, 2018.

[14] TrapX Security, "Retail Point-of-Sale," 4 April 2017. [Online]. Available: https://img1.wsimg.com/blobby/go/8794a89a-ce4f-40d0-9882-789c102f395b/downloads/1cl9dmuf7_101089.pdf?ver=1615661407182. [Accessed 21 Desember 2022].

[15] P. V. Ross R, G. R, B. D and M. R., "Developing Cyber-Resilient Systems: A Systems Security Engineering Approach," Desember 2021. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-160/vol-2-rev-1/final. [Accessed 12 Desember 2022].

[16] L. Zhang and V. L. L. Thing, "Three Decades of Deception Techniques in Active cyber defense - Retrospect and outlook," *Computers & Security*, 2021.

[17] X. Han , N. Kheir and D. Balzarotti, "Deception Techniques In Computer Security: A Research Perspective," *ACM Computer Survey*, vol. 1, no. 1, pp. 1-36, 2019.

[18] C. Dalamagkas, P. Sarigiannidis, D. Ioannidis, E. Iturbe, O. Nikolis, F. Ramos, E. Rios, A. Sarigiannidis and D. Tzovaras, "A survey on honeypots, honeynets and their applications on smart grid," in *IEEE Conference on Network Softwarization (NetSoft)*, 2019.

[19] W. Fan, Z. Du, D. Fernandez and V. A. Villagra, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906-3919, 2018.

[20] A. Mairh, D. Barik, K. Verma and D. Jena, "Honeypot in network security: A survey," in *International Conference on Communication, Computing & Security*, New York, 2011 .

[21] R. M. Campbell, K. Padayachee and T. Masombuka, "A survey of honeypot research: Trends and opportunities," in *2015 10th international Conference for Internet Technology and Secured Transactions*, London, 2015.

[22] L. Zobal, D. Kolar and R. Fudjiak, "Current State of honeypots and deception strategies in cybersecurity," in *11th International Congress on Ultra Modern Telecommunications and Control System and Workshops (ICUMT)*, Dublin, 2019.

[23] Fan, Wenjun; Du, Zhihui; Fernandez, David;, "Taxonomy of honeynet solutions," in *SAI Intelligent Systems Conference 2015*, London, 2015.

[24] J. Yuill, M. Zappe, D. Denning and F. Feer, "Honeyfiles: Deceptive Files for Intrusion Detection," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop*, New York, 2004.

[25] H. Cheng, W. Li, P. Wang, C.-H. Chu and K. Liang, "Incrementally Updateable Honey Password Vaults," in *The Proceedings of the 30th USENIX Security Symposium*, 2021.

[26] S. Srinivasa, J. M. Pedersen and E. Vasilomanolakis, "Towards systematic honeytoken fingerprinting," in *International Conference on Security of Information and Networks (ACM SIN)*, Istanbul, 2020.

[27] M. M. Andersen, H. David and E. Vasilomanolakis, "Honeysweeper: Towards stealthy honeytoken fingerprinting techniques," in *27th Nordic Conference on Secure IT Systems*, Reykjavik, 2022.

[28] J. H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense.," *IEEE*

Communications Surveys & Tutorials, vol. 22, no. 1, pp. 709-745, 2020.

[29] G. L. Cai, B. S. H. W. Wang and T. Z. Wang, "Moving target defense: state of the art and characteristics," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 11, pp. 1122-1153, 2016.

[30] C. Lei, H. Q. Zhang, J. L. Tan, Y. C. Zhang and X. H. Liu, "Moving target defense techniques: A survey.," *Security and Communication Networks*, 2018.

[31] M. Asif and Y. Al-Harthi, "Intrusion detection system using Honey Token based Encrypted Pointers to mitigate cyber threats for critical infrastructure networks," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, San Diego, 2014.

[32] M. Almeshekah and E. Spafford, "Cyber Deception," in *Cyber Security Deception*, Springer, 2016.

[33] W. Steingartner and D. Galinec, "Cyber threats and cyber deception in hybrid warfare," *Acta Polytechnica Hungarica*, vol. 18, no. 3, pp. 25-45, 2021.

[34] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb and M. A. Rassam, "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions," *Applied Science* , vol. 12, no. 1, pp. 1-45, 2022.

[35] S. Kok, A. Abdullah, N. Jhanjhi and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *International Journal Computer Science Network Security*, vol. 19, no. 2, 2019.

[36] S. J. Lee, H. Y. Shim, Y. R. Lee, T. R. Park, S. H. Park and I. G. Lee, "Study on systematic ransomware detection techniques," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, Phoenix Pyeongchang, 2022.

[37] C. V. Bijitha, R. Sukumaran and H. V. Nath, "A survey on ransomware detection techniques," in *Secure Knowledge Management In Artificial Intelligence Era: 8th International Conference, SKM 2019*, Goa, 2020.

[38] W. Z. Zakaria, M. F. Abdollah and A. M. Ariffin, "On ransomware detection," in *Proceedings of the seventh international conference on informatics and applications (ICIA2018)*, 2018.

[39] L. B. Bhagwat and B. M. Patil, "Detection of ransomware attack: A review," in *Proceeding of International Conference on Computational Science and Applications: ICCSA*, 2019.

[40] S. Kamil, H. S. A. S. Norul, A. Firdaus and O. L. Usman, "The rise of ransomware: A review of attacks, detection techniques, and future challenges," in *International Conference on Business Analytics for Technology and Security (ICBATS)*, 2022.

[41] S. R. Davies, R. Macfarlane and W. J. Buchanan, "Review of Current Ransomware Detection Techniques," in *Proc. of the 7 th International Conference on Engineering and Emerging Technologies (ICEET)*, 2022.

[42] J. P. Tailor and A. D. Patel, "A comprehensive survey: ransomware attacks prevention, monitoring and damage control," *International Journal of Innovative Research*, vol. 4, no. 15, pp. 116-121, 2017.

[43] B. Al-Fuhaidi, W. Al-Sorori, N. Maqtary, A. Al-Hashedi and S. Al-Taweel, "Literature Review on Cyber Attacks Detection and Prevention Schemes," in *International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, 2021.

[44] Z. Manjezi and R. Botha, "Preventing and Mitigating Ransomware - A Systematic Literature Review," in *17th International Conference, ISSA 2018*, Pretoria, 2019.

[45] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma and I. E. Davidson, "Ransomware detection, avoidance, and mitigation scheme: a review and future directions," *Sustainability*, vol. 14, no. 1, 2021.

[46] H. Oz, A. Aris, A. Levi and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11, pp. 1-37, 2022.

[47] H. R. M. Husny, M. S. M. Yussof, N. Y. Abdullah and W. H. W. Ismail, "Detection and Mitigation of Virus Ransomware," *Journal of Computing Technologies and Creative Content (JTec)*, vol. 5, no. 1, pp. 8-13, 2020.

[48] Z. Abdullah, F. W. Muhadi, M. M. H. I. R. A. Saudi and C. F. M. Foozy, "Android ransomware detection based on dynamic obtained features," in *International Conference on Soft Computing and Data Mining*, Langkawi, 2020.

[49] S. Alsoghyer and I. Almomani, "Ransomware detection system for Android applications," *Electronics*, vol. 8, no. 8, 2019.

[50] S. Sharma, R. Kumar and C. R. Krishna., "A survey on analysis and detection of Android ransomware," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 16, 2021.

[51] N. Alzahrani and D. Alghazzawi, "A review on android ransomware detection using deep

learning techniques.," in *Proceedings of the 11th international conference on management of digital EcoSystems*, 2019.

[52] R. Moussaileb, N. Cuppens, J. L. Lanet and H. L. & Bouder, "A survey on windows-based ransomware taxonomy and detection mechanisms," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1-36, 2021.

[53] N. Soltani, A. M. Rahmani, M. Bohlouli and M. Hosseinzadeh, "Artificial intelligence empowered threat detection in the Internet of Things: A systematic review," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 22, 2022.

[54] A. Cimitile, F. Mercaldo, V. Nardone, A. Santone and C. A. Visaggio, "Talos: no more ransomware victims with formal methods," *International Journal of Information Security*, vol. 17, pp. 719-738, 2018.

[55] M. Abdullahi, Y. Baashar, H. Alhussian, A. Alwadain, N. Aziz, L. Capretz and S. Abdulkadir, "Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review," *Electronics*, vol. 11, no. 198, 2022.

[56] E. Berrueta, D. Morato, E. Magaña and M. Izal, "A survey on detection techniques for cryptographic ransomware," *IEEE Access*, vol. 7, pp. 144925-144944, 2019.

[57] A. Alqahtani and F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: an evolving outlook.," *Sensors*, vol. 22, no. 5, 2022.

[58] D. Su, J. Liu, X. Wang and W. Wang, "Detecting Android locker-ransomware on chinese social networks," *IEEE Access*, vol. 7, pp. 20381-20393, 2018.

[59] D. Su, J. Liu, X. Wang and W. Wang, "Detecting Android locker-ransomware on chinese social networks," *IEEE Access*, vol. 7, p. 20381–20393, 2018.

[60] H. Sultan, A. Khalique, S. I. Alam and S. Tanweer, "A SURVEY ON RANSOMEWARE: EVOLUTION, GROWTH, AND IMPACT," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, 2021.

[61] C. Moore, "Detecting Ransomware with Honeypot Techniques," in *2016 Cybersecurity and Cyberforensics Conference (CCC)*, Amman, Jordan, 2016.

[62] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner and M. Khalil, "Lessons from applying the systematic literature review processwithin the software engineering domain," *The Journal of System and Software*, vol. 80, no. 4, pp. 571-583, 2007.

[63] Y. Pan, X. Ge, C. Fang and Y. Fan, "A Systematic Literature Review of Android Malware Detection Using Static Analysis," *IEEE Access*, vol. 8, pp. 116363-116379, 2020.

[64] D. Liebowitz, S. Nepal, K. Moore, C. J. Christoper, S. S. Kanhere, D. Nguyen, R. C. Timmer, M. Longland and K. Rathakumar, "Deception for Cyber Defence: Challenges and Opportunities," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, Atlanta, Georgia, 2021.

[65] N. Cifranic, J. Romero-Mariona, B. Souza and R. Hallman, "Decepti-SCADA: A Framework for Actively Defending Networked Critical Infrastructures," in *5th International Conference on Internet of Things, Big Data and Security*, Prague, 2020.

[66] C. Wang and Z. Lu, "Cyber Deception: Overview and the Road Ahead," *IEEE Security and Privacy Magazine*, vol. 16, no. 2, pp. 80-85, 2018.

[67] W. Wang, J. Bickford, I. Murynets, R. Subbaraman, A. G. Forte and G. Singaraju, "Detecting targeted attacks by multilayer deception," *Journal of Cyber Security and Mobility*, vol. 2, pp. 175-199, 2013.

[68] T. W. Edgar and D. O. Manz, "Honeypot," in *Research Methods for Cyber Security*, Elsevier, 2017.

[69] M. H. Lopez and C. F. L. Resendez, "Honeypots: Basic Concepts, Classification and Educational Use as Resources in Information Security Education and Courses," in *Proceedings of the Informing Science & IT Education Conference* , 2008.

[70] N. Titarmare, N. Hargule and A. Gupta, "An Overview of Honeypot Systems," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 394-397, 2019.

[71] J. Franco, A. Aris, B. Canberk and A. S. Uluagac, "A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems," *IEEE Communications Surveys & Tutorials*, pp. 2351-2383, 2021.

[72] F. A. Alaba, A. Jegede and ', "Ransomware Attacks on Remote Learning Systems in 21st Century: A Survey," *Biomedical Journal of Scientific & Technical Research*, vol. 35, no. 1, pp. 27322-27330, 2021.

[73] A. P. de Barros, "RES: Protocol Anomaly

Detection IDS - Honeypots," 21 Februari 2003. [Online]. Available: https://seclists.org/focus-ids/2003/Feb/95. [Accessed 10 Desember 2022].

[74] Fortinet, "Honey Tokens," Fortinet Cyber Readiness Center and Breaking Threat Intelligence, 2023. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/honey-tokens. [Accessed 13 November 2022].

[75] L. Martin, "Cyber Kill Chain Framework," 2023, [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. [Accessed 2 Desember 2022].

[76] Q. Khan, A. Mirza, M. Brown, O. Halling, L. Shand and A. Alam, "Ransomware Analysis using Cyber Kill Chain," in *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, Rome, 2021.

[77] M. H. Almeshekah and E. H. Spafford, "Planning and integrating deception into computer security defenses," *ACM Workshop on New Security Paradigms Workshop (NSPW)*, 2014.

[78] F. Cohen, "A note on the role of deception in information protection," *Computers & Securitt*, 1998.

[79] G. Kontaxis, M. Polychronakis and A. D. Keromytis, "Computational Decoys for Cloud Security," in *Secure Cloud Computing*, 2014.

[80] F. Araujo, K. W. Hamlen, S. Biedermann and S. Katzenbeisser, "From patches to honeypatches: Lightweight attacker misdirection, deception, and disinformation," in *ACM SIGSAC conference on computer and communications security (CCS)*, 2014.

[81] S. Webb, J. Caverlee and C. Pu, "Social Honeypots: Making Friends With A Spammer Near You," 2008. [Online]. Available: https://people.engr.tamu.edu/caverlee/pubs/webb08socialhoneypots.pdf. [Accessed 21 November 2022].

[82] K. Borders, L. Falk and A. Prakash, "OpenFire: Using deception to reduce network attacks.," *Security and Privacy in Communications Networks and the Workshops*, 2007.

[83] T. Liston, " LaBrea:“Sticky” Honeypot and IDS," 2001. [Online]. Available: https://labrea.sourceforge.io/labrea-info.html. [Accessed 20 Desember 2022].

[84] S. T. Trassare, "A technique for presenting a deceptive dynamic network topology," 2013. [Online]. Available: https://core.ac.uk/download/pdf/36725616.pdf. [Accessed 12 November 2022].

[85] B. M. Bowen, V. Kemerlis, P. Prabhu, A. D. Keromytis and S. J. Stolf, "Automating the Injection of Believable Decoys to Detect Snooping," in *Proceedings of the Third ACM Conference on Wireless Network Security*, 2010.

[86] S. Chakravarty, G. Portokalidis, M. Polychronakis and A. D. Keromytis, "Detecting traffic snooping in tor using decoys," in *Workshop on Recent Advances in Intrusion Detection*, 2011.

[87] Statista Research Department, "Statista," Statista, 7 July 2022. [Online]. Available: https://www.statista.com/statistics/701020/major-operating-systems-targeted-by-ransomware/. [Accessed 19 10 2022].

[88] T. McIntosh, A. S. M. Kayes, Y.-P. P. Chen, A. Ng and P. Watters, "Ransomware Mitigation in the Modern Era: A Comprehensive Review, Research Challenges, and Future Directions," *ACM Computing Surveys*, vol. 54, no. 9, p. 1–36, 2022.

[89] Y. Feng, C. Liu and B. Liu, "Poster : A New Approach to Detecting Ransomware with Deception," in *Proceedings of the 38th IEEE Symposium on Security and Privacy Workshops*, San Jose, 2017.

[90] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware," in *IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Klagenfurt, 2017.

[91] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *10th International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, 2018.

[92] U. Adamu and I. Awan, "Ransomware Prediction Using Supervised Learning Algorithms," in *7th International Conference on Future Internet of Things and Cloud (FiCloud)*, Istanbul, 2019.

[93] V. Sethia and A. Jayasekar, "Malware Capturing and Analysis using Dionaea Honeypot," in *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, 2019.

[94] S. I. Sajid, J. Wei, B. Abdeen, E. Al-Shaer, M. Islam, W. Diong and L. Khan, "SODA: A System for Cyber Deception Orchestration and Automation," in *ACSAC '21: Annual Computer Security Applications Conference*, 2021.

[95] Z. Wang, X. Wu, C. Liu, Q. Liu and J. Zhang, "RansomTracer: Exploiting Cyber Deception for Ransomware Tracing," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, 2018.

[96] J. A. Gomez-Hernandez, L. Alvarez-Gonzalez and P. Garcia-Teodoro, "R-Locker: Thwarting Ransomware Action through a honeyfile-based approach," *Computers & Security*, vol. 73, pp. 389-398, 2018.

[97] S. Mehnaz, A. Mudgerikar and E. Bertino, "RWGuard: A Real-Time Detection System Against Cryptographic Ransomware," in *21st International Symposium, RAID* , Crete, 2018.

[98] J. Lee, J. Lee and J. Hong, "How to Make Efficient Decoy Files for Ransomware Detection?," in *RACS '17: Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, Krakow, 2017.

[99] S. Sheen, K. A. Asmitha and S. Venkatesan, "R-Sentry: Deception based ransomware detection using file access patterns," *Computers and Electrical Engineering*, vol. 103, 2022.

[100] E. Kolodenker, W. Koch, G. Stringhini and M. Egele, "PayBreak: Defense Against Cryptographic Ransomware," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, 2017.

[101] Z. Wang, C. Liu, J. Qiu, Z. Tian , X. Cui and S. Su, "Automatically Traceback RDP-Based Targeted Ransomware Attacks," *Wireless Communications and Mobile Computing*, pp. 1-13, 2018.

[102] WatchPoint Data, "Canauri™: Ransomware Protection," [Online]. Available: https://www.canauri.com/features/. [Accessed 12 Desember 2022].

[103] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi and B. Raman, "Design of Intrusion Detection Honeypot Using Social Leopard Algorithm to Detect IoT Ransomware Attacks," *IEEE Access*, vol. 8, pp. 169944-169956, 2020.

[104] M. Karakate, H. Esaki and H. Ochiai, "SDNHive: A Proof-of-Concept SDN and Honeypot System for Defending Against Internal Threats," in *ICCNS 2021: 2021 the 11th International Conference on Communication and Network Security*, Weihai, 2021.

[105] C. Pascariu and I.-D. Barbu, "Ransomware Honeypot: Honeypot solution designed to detect a ransomware infection identify the ransomware family," in *11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Pitesti, 2019.

[106] S. Wang, H. Zhang, S. Qin, W. Li, T. Tu, A. Shen and W. Liu, "KRProtector: Detection and Files Protection for IoT Devices on Android Without ROOT Against Ransomware Based on Decoys," *IEEE INTERNET OF THINGS JOURNAL*, vol. 9, no. 19, pp. 18251-18266, 2022.

[107] Z. A. Genc, G. Lenzini and D. Sgandura, "On Deception-Based Protection Against Cryptographic Ransomware," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2019.

[108] VirusTotal, "Ransomware in a global context," VirusTotal, 2022.

[109] J. Sun, K. Sun and Q. Li, "Towards a Believable Decoy System: Replaying Network Activities from Real System," in *2020 IEEE Conference on Communications and Network Security (CNS)*, Avignon, 2020.

[110] X. Han, N. Kheir and D. Balzarotti, "Evaluation of Deception-Based Web Attacks Detection," in *MTD '17: Proceedings of the 2017 Workshop on Moving Target Defense*, Dallas, 2017.

[111] H. Wu, Y. Gu, G. Cheng and Y. Zhou, "Effectiveness Evaluation Method for Cyber Deception Based on Dynamic Bayesian Attack Graph," in *CSSE 2020: 2020 3rd International Conference on Computer Science and Software*, 2020.

APPENDIX I - USED TOOLS/MODULES/ALGORITHM & ITS USAGE ON EXPERIMENT SETUP

Ref	Tools/Modules/Algorithm and its function (if any)
[61]	- Microsoft File Server Resource Manager (FSRM) - Windows Event Logs, EventSentry: to monitor event log changes - EventSentry: To send early warning e-mail, stop server services, and services shutdown
[96]	- R-Locker: To deploy honey files around the environment of the target (based on depth-first file traversal pattern) - R-Locker: To monitor events in the filesystem - R-Locker: To block ransomware, notify the malicious event, and automatically deploy countermeasures to solve threats. - Wine: to test Wannacry Ransomware sample (in Unix distro)
[98]	- Manually identify ransomware's file search algorithm, file/folder traversal pattern, and file encryption method - Automatically generate decoy files using the given algorithm and tweak it with the extracted ransomware behavior
[100]	- Developed RADDAR System: Perform Real-time and automatic Ransomware discovery, detection, and alert. - RADDAR System: Obtain malware/ransomware samples from VirusTotal or various locations for other samples. - RADDAR System: Identify malware sample (crypto-ransomware or not/executing malicious process or not). - Cuckoo Sandbox and Windows 7 VM: to sandbox, analyze and output a behavior report on each sample - Microsoft's Crypto APIs & Crypto++ library: Crypto Function Hooking: export session key & algorithm parameters - Paybreak Key Vault: for key material & algorithm details (recovered from hooked procedures) - Paybreak File recovery: use key material and algorithm to attempt recovery
[89]	- Manually create decoy files (common target of ransomware, e.g., txt, doc, pdf) - Manually place decoy files in every directory; Windows API: redefine FindFirstFile API & FindNextFile API function - Designed Monitor Module: observe the behavior of the suspicious application when it operates decoy files.
[90]	- UNVEIL: Generate valid content (use standard lib: python-docx, OpenSSL), file extension, File Path & Time Attribute. - UNVEIL: monitor filesystem access (I/O monitor & logs retrieval using Windows Filesystem Minifilter Driver framework). - Cuckoo Sandbox & VM: to set environment, make sample submission, manage some VMs, simulate user input: clicking / cursor movement, etc. Configure the network, and set the IP address range & MAC address to prevent VMs' fingerprints from being recognized. - UNVEIL: capture screenshot outside dynamic analysis to prevent potential tampering - UNVEIL: use a python script to implement the Structural Similarity Image Metric (SSIM) to test dissimilarity - UNVEIL: use Tesseract-OCR (open-source OCR engine) to extract text from the selected areas of the screenshots
[97]	- RWGuard Decoy File Generator & Dmon Module: To Generate and deploy decoy files to be monitored - RWGuard PMon Module: To Monitor running processes/IRP. - RWGuard FCMon Module: Check the file system for malicious activity/file change anomaly. - RWGuard CHFk Module: relevant Crypto-API function hooking for decryption key & other parameter storage.
[101]	- Manually create the 3-layer Cyberdeception environment (Windows-based deception environment prototype) - Environment monitor: detect RDP-based ransomware attacks in time and determine the attacker's behavior. - NLP and Machine Learning equipped System: Extract login information, clipboard content, folder path, PE file, etc - Deception environment Prototype: trap the ransomware attacker and collect a lot of information - Deception environment Prototype: Capture traceable clues/info & analyze both. - Deception environment Prototype: Generate a report to do traceback the attacker (from previous clues & analysis)

Ref	Tools/Modules/Algorithm and its function (if any)
[91]	-RansomWall 1st layer: IDA tool: Perform Static Reverse Engineering to cryptographic ransomware samples Collected from VirusShare; - Ransomwall 2nd layer: Sets trap by tracking the occurrence of malicious activities. Honey Files and Honey Directories are deployed in critical user data folders. Modification on these (files/directories) indicate suspicious behavior. - Ransomwall 3rd layer: Monitor file system operations and entropy modifications to track massive encryption activities. - Ransomwall 4th layer: files modified by a suspicious process are backed up in a separate folder. - Ransomwall 5th layer: Use Supervised algorithm to classify executables (benign/malicious) - Cuckoo Sandbox and Windows 7 & 8.1 VM: to sandbox, execute tools and instructions per each Ransomwall layer
[95]	- RansomTracer: create an artificial, realistic, and enticing user environment for the RDP attack ransomware - RansomTracer: identify and collect traceable clues from the decoy user environment - RansomTracer: extract and analyze discovered traceable clues
[92]	- Manually obtain ransomware samples from strategic web compromise, drive-by download, email-phishing, & vulnerability exploit. - Honeypot trap file/honey file: If one of the host systems is infected with ransomware, will collect the files to be dynamically analyzed (generated data set report is in CSV format) - Utilize supervised ML algorithms e.g., Support Vector Machine, Random Forest, Decision Tree, Bayesian Network, Artificial Neural Network, & Linear Regression to do the classification (distinguishing between goodware & ransomware)

APPENDIX I - USED TOOLS/MODULES/ALGORITHM & ITS USAGE ON EXPERIMENT SETUP

Ref	Tools/Modules/Algorithm and its function (if any)
[93]	- Amazon EC2: Create the Amazon EC2 instance (instance launched and OS selected), process terminal in Amazon EC2 - Dionaea honeypot: Installed and run on Ubuntu 14.04 (logs are stored in /var/log/dionaea and var/lib/dionaea/directory." - Proposed combination honeypot: Observe the mostly attacked protocol (SQL 2000XP, SMB, & FTP). - Proposed combination honeypot: Categorize sample malware based on behavior (3 high-severity malware): Wannacry ransomware: attack SMB protocol in Windows OS, Slammer: attack SQL 2000 XP version of Windows, GandCrab ransomware
[105]	- Manually Deploying honey file in a simulated network share using Docker Technology - Manually configuring Samba in combination with Syslog to monitor all client-related operations for all individual users. - Manually created detection script for detection (based on information from existing file share's files, Samba service, & logs
[103]	- Honeyfolder: randomly generated, SoLA-modeled, & installed in every host. - Honey agents: to monitor decoy folders & act as an early warning system to alert the firewall (do process kill if a certain threshold is met) - Intrusion Detection Honeypot's Audit Watch: to monitor file/folder changes - Intrusion Detection Honeypot's Complex Event Processing (CEP) taken out of: hosts/networks, honeypot agent, SDN controller, Audit Watch, logs, etc.) - Intrusion Detection Honeypot's Complex Event Processing (CEP): block ransomware communication
[104]	- SDN-Hive System components 1. SDN controller: ONOS (as production-ready OS to develop & deploy SDN app via its API. - SDN Application: to protect the host from malware. Consists of: SMB & ARP scan detection, DNS, IP & MAC address Blacklist. - SDN Switch: OpenvSwitch to support OpenFlow protocol (conduct issued blocking rules based on detected malicious activities. - Honeypot Traffic analyzer: to process and analyze network packet - Honeypot software: decoy network service to monitor malware that successfully barged its way - Backscanner: a follow-up network scan for identified suspicious IP address with three submodules: Nmap, Celery, & RabbitMQ
[94]	- SODA: create deception playbook: Malicious Subgraph (MSG) Extraction, MSG Classifier,

Ref	Tools/Modules/Algorithm and its function (if any)
	Deception Factory Synthesis - SODA Detection agent: acts as an entry point to detect malware & trigger the orchestration. - SODA Orchestration Engine Server (OES), Orchestration Engine Client (OEC), Detection Agent, and HoneyFactory (HF). Procedures executed Real-time orchestration. - SODA Real-time deception using embedded API-hooking (deceive malware execution). - SODA Profile creation (to OES via UI), SODA: Pre-built Profile Selection: for each selected pre-built profile, relevant deception ploys are shown
[102]	- Canauri: running ransomware samples executables, automatically alert the user through a dialogue box - Canauri: system alert shows the location of several files' extensions and directories enumerated/changed (by malware /ransomware) - Canauri: Administrative alert is sent to the security team with the attack's critical detail. Folder/file enumerated differently based on different file traversal/file access patterns (the very first directory enumerated will be shown in the alert dialogue box). - Canauri: System shutdown is performed to prevent further infection and propagation of the ransomware within the same network
[99]	- Honeyfiles generated based on different categories of file attributes (manually created using common software) - R-Sentry System: to place honey files in optimal location based on file access/traversal pattern analysis - File monitor: used to monitor the sequence of every accessed file's path and actions taken in the entire user directory. - File monitor: Detect ransomware samples that access honey files placed on every root folder. - Detected Ransomware was then blocked and notified to admin for removal
[106]	- KRProtector decoy deployment module: generate & deploy decoys based on the files' distribution, including sibling directory decoys & subdirectory decoys (decoy's entity: empty folders). - KRProtector decoy deployment module: use "userfolder", "leaffolder", & "subsibling" decoys to represent folders, user folder, etc. - KRProtector decoy deployment module: If decoys are accessed, it filters all active apps based on the existing 4-rules & triggers the detection module - KRProtector ransomware detection module: use trusted value metrics (the lowest positive number is regarded as crypto-ransomware) - KRProtector ransomware detection module: Shortlist the suspicious app, discover the origin of suspicious behavior, and inform the user (alert)

APPENDIX II – ADVANTAGES AND LIMITATIONS

Ref	Advantages	Limitations
[61]	- Network activities are interrupted when a certain activity level is identified - Network services were also stopped (in 6 seconds) when a certain activity level was identified - Early warning sent to system admin via e-mail	- Only tested on a simulated script, no actual ransomware involved - No guarantee the malware would attempt to invade these areas (honeypot folders), therefore bypassing this defense
[96]	- Ransom operation is completely blocked when the trap file is accessed. - Countermeasures are automatically launched to solve the infection without affecting the system's normal operation - Complexity and overhead involved in the solution are really low - 100% detection accuracy and null damage because of the honey file effectiveness and immediate block - Simple and autonomous execution without any additional processes to complement its functionality	- Defense can be bypassed if the ransomware is randomly accessing files. - Worst case scenario, the sample/honey (files) can be blocked after encrypting other files first. - Have yet to be implemented/tested in other platforms, e.g., Windows and Android.
[98]	- Novel approach on efficient method in generating decoy files for Ransomware Detection - New countermeasure is designed not only for existing but also for possible future ransomware	- Not all Ransomware behavior were analyzed. The proposed method may not efficiently detect all ransomware variation
[100]	- File Restoration Effectiveness evaluated against 107	- Does not resist all obfuscation threats

Ref	Advantages	Limitations
	ransomware samples from ~20 families (Locky, Cryptolocker, Samsam) - Protection tasks can be performed at trivial performance overhead for everyday office workload	- Cannot provide guarantees against malware specifically aiming to evade Paybreak - The integrity of the escrow key stored in the key vault is susceptible to a Denial of Service attack - Key stored in the vault can be corrupted or filled with nonsensical information.
[89]	- Processes have to operate decoy files first before the real ones and can only be operated after the processes pass the detection phase. - The proposed detection approach has proven to be effective in identifying the encryption process and stopping it in a timely manner. - Low on CPU, memory, and disk usage/load and works within tolerable time delays.	- Not yet tested on different kinds of ransomware - Not yet tested in a normal environment (close to unmalicious activities) to see if the approach impedes the user's normal usage.
[90]	- The evaluation is performed at a large scale (13,637 ransomware samples out of 148,223 recent general malware). - The proposed techniques work well in practice (True positive rates at 96,3% and zero false positives)	- Ransomware could remain undetected if ransomware runs at the kernel level & thwart some hooks used for filesystem monitor. - The attacker's ability to fingerprint the dynamic analysis environment is always possible. It can prevent dynamic analysis from being done.
[97]	- Effective in real-time detection of ransomware: zero false negative & negligible false positive (~0.1%) rates - Robust decoy design combined with obfuscation techniques can impede ransomware when trying to find out the decoy generator app. - Certain modules (CFHk) can retrieve parameters to specific crypto function calls & restore the encrypted files. - All modules implemented are incurring an overhead of only ~1.9%.	- Several modules are theoretically subject to missing some of the malicious activities. - Time lag between logging and parsing activities for the anomalies creates a small "vulnerable" window for ransomware
[101]	- Clue capture and analysis worked well (data on login info, clipboard content, shared folder path, uploaded PE files collected). - Traceable clues remain (as planned) after some evaluation process involving 122 volunteers provided with 12 virtual hosts. - Detailed analysis is proven helpful for automatic analysis system and display traceable information about the attacker.	- Tests against ransomware families other than the RDP-based ransomware are not presented (limited experiment samples)
[91]	- Effective approach with insignificant spatial cost and system resource consumption - Tested against 574 Crypto ransomware families samples in a real-world environment with a detection rate of 98,25% & near-0 false positive - Can detect 30 0-day intrusion samples.	- Ransomwall has not been evaluated on a large-scale, real setup
[95]	- RansomTracer is able to ensnare the attacker and collect traceable clues left by the attacker in a deception environment - Automated clue identification enables the user to converge the number of traceable clues to about 2%. - Tracing back the ransomware attackers provide a good deterrent to adversaries, thus stifling the development of ransomware.	- Only tested against RDP-based ransomware families, thus needing more various testing environments.

Ref	Advantages	Limitations
[92]	- Data set for classification in the data repository is heterogeneous (942 good ware & 582 ransomware samples out of 11 different families) - The detection rate within every algorithm is all right (>50%); the highest accuracy is the SVM algorithm out of the six algorithms proposed.	- Proposed solution focuses more on dynamic analysis & machine learning methods for ransomware detection (not deception). - The 11 ransomware claims are not listed/detailed in the paper. - Honeypot is used passively only for ransomware information collection medium.
[93]	- Proposed honeypot can detect three high-profile ransomware and various categories of malware.	- Only for low interaction honeypot
[105]	- Decoy files performance is evaluated against WannaCry & Stampado in a development environment composed of an "infected-VM-system." - The proposed system can also reduce the False positive rate by using an entropy calculation check. (Entropy calculation can help distinguish between the normal/slight "user-modification" and the "malicious encryption").	- Less effective when faced with other types of ransomware - For example, MBR (Master Boot Record) Ransomware can restart the infected computer.
[103]	- Effective in restricting ransomware activity (tested against 20 recent ransomware variants). - Proposed IDH improves ransomware detection time and rate.	- Optimizing loads functionality for IoT devices (e.g., auto-tabling and transfers learning) is not available
[104]	- Improvement of the previous SDN-based solution (combination of SDN and Honeypot-based protection) to defend ransomware works. - SDN-Hive has proven its capabilities in protecting devices from worms & worms-like ransomware via: - a. Identification of the worm component by its unique characteristics - b. Detection of malicious activities and generation of blocking rules to OpenFlow switches (in the network) - c. ARP & SMB detection function could prevent vulnerable hosts' infection in the same LAN (because of Wannacry). - d. An external module is deployed to detect various scans & alert the SDN controller	- Have not explored much about various propagation techniques used by other malware - Other common protocols, e.g., Telnet and SSH, have yet to be explored. - Bruteforce-based behavior of worms has not been considered.
[94]	- For testing against ransomware, 96 samples were used, and 11 distinct malicious behaviors were observed to identify 28 valid deception ploys. Those 28 deception ploys were deployed, & SODA could deceive the ransomware using 27. - Based on the Performance measurement and comparison to Cuckoo sandbox and Any.run, SODA has better coverage. It outperforms other existing tools in identifying ransomware (Ryuk and GandCrab Family) capabilities & presenting them in the MITRE ATT&CK framework.	- Relies on existing malware detection approach (can sometimes trigger false positives). - The possibilities of API hooking detection and evasion by the malware will make SODA unable to deceive the malware
[	- Canauri™ does not require additional hardware to maintain. It is claimed that Canauri™ protects the	- It is a paid solution (as commercial software) - Although there is a money-back guarantee

Ref	Advantages	Limitations
102]	system in a matter of minutes - Canauri™ also takes fast action after the attacks (administrative alert and system shutdown) - Unlike Anti-Malware, Canauri™ is signature-less. It will always detect the activity of encryption rather than detecting - The specific variant of ransomware running.	and claimed to be minimal, false positive out of the detection process is still expected
[99]	- Lightweight & real-time solution of placing honey files in multiple locations (previous anti-ransomware solution only placed on root folder) - Work properly during the execution of different ransomware samples (24~ families of crypto-ransomware used in the experiment) - Honey file placement strategy is based on the ransomware analysis (file traversal/access pattern on as-is & nextgen ransomware families)	- Can be partially bypassed by gaining access randomly - Honeyfile is still generated manually (further research can extend this to automatic generation) - Proposed work still needs to be tested in a real-world scenario.
[106]	- Relatively low storage consumption (36 user folder and 5517 sub-sibling decoys only takes around 22,14-44.29 MB) - Highest detection accuracy (96,2%) with negligible usage of computing resources (trusted value calculation's time complexity is O(n).) - Very fast detection and responsive early warning (alert), thus preventing the ransomware from encrypting the files - Have the capabilities to identify wider variations of ransomware detection (tested along ransomware with obfuscation, steganography, & dynamic malicious code)	- Difficulty in sensing the behavior of ransomware without ROOT or administrator privilege (the proposed method needs root to monitor status changes of decoys in order to activate the detection module within KRProtector) - Limited to Android, has not been tested yet on other platforms/OS such as Linux - Linux is known to be able to provide file system mechanisms (monitor file/folder without ROOT); thus, it can be explored in the future