IARIA

ICCGI 2024

The Nineteenth International Multi-Conference on Computing in the Global Information Technology

ISBN: 978-1-68558-130-5

March 10th –14th, 2024

Athens, Greece

ICCGI 2024 Editors

Petre Dini, IARIA, USA/EU

ICCGI 2024

Forward

The Nineteenth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2024), held between March 10th and March 14th, 2024, continued a series of international events covering a large spectrum of topics related to global knowledge concerning computation, technologies, mechanisms, cognitive patterns, thinking, communications, user-centric approaches, nanotechnologies, and advanced networking and systems.

The conference topics focused on challenging aspects in the next generation of information technology and communications related to the computing paradigms (mobile computing, database computing, GRID computing, multi-agent computing, autonomic computing, evolutionary computation) and communication and networking and telecommunications technologies (mobility, networking, bio-technologies, autonomous systems, image processing, Internet and web technologies), towards secure, self-defendable, autonomous, privacy-safe, and context-aware scalable systems.

We take here the opportunity to warmly thank all the members of the ICCGI 2024 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ICCGI 2024. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the ICCGI 2024 organizing committee for their help in handling the logistics of this event.

We hope that ICCGI 2024 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of computing in the global information technology.

ICCGI 2024 Chairs

ICCGI 2024 Steering Committee
Constantin Paleologu, Polytechnic University of Bucharest, Romania
Peter Kieseberg, St. Pölten University of Applied Sciences, Austria
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Ulrich Reimer, University of Applied Sciences St. Gallen, Switzerland
Kathryn E. Stecke, University of Texas at Dallas, USA
Juho Mäkiö, Hochschule Emden / Leer, Germany

ICCGI 2024 Publicity Chairs
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

ICCGI 2024
Committee

ICCGI 2024 Steering Committee

Constantin Paleologu, Polytechnic University of Bucharest, Romania
Peter Kieseberg, St. Pölten University of Applied Sciences, Austria
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Ulrich Reimer, University of Applied Sciences St. Gallen, Switzerland
Kathryn E. Stecke, University of Texas at Dallas, USA
Juho Mäkiö, Hochschule Emden / Leer, Germany

ICCGI 2024 Publicity Chairs

José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

ICCGI 2024 Technical Program Committee

Ahmed M. Abdelmoniem, KAUST, Saudi Arabia
António Abreu, ISEL/IPL, Portugal
Abdelouhab Aitouche, EEA - French School of High Studies in Engineering, France
Abdullah Al-Alaj, Virginia Wesleyan University, USA
Mohammad Yahya Al-Shamri, King Khalid University, Saudi Arabia
Alma Y. Alanis. University of Guadalajara, Mexico
Fernando Almeida, INESC TEC & University of Porto, Portugal
José Antonio Apolinário Junior, Military Institute of Engineering (IME), Rio de Janeiro, Brazil
Nancy Arana Daniel, University of Guadalajara, Mexico
Jocelyn Aubert, Luxembourg Institute of Science and Technology (LIST), Luxembourg
Michaela Baumann, NÜRNBERGER Versicherung, Germany
Robert Bestak, Czech Technical University in Prague, Czech Republic
Dorota Bielinska-Waz, Medical University of Gdansk, Poland
Fernando Bobillo, University of Zaragoza, Spain
Zorica M. Bogdanovic, University of Belgrade, Serbia
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Jean-Louis Boulanger, CERTIFER, France
Christos Bouras, University of Patras, Greece
Juan Carlos Burguillo-Rial, Universidade de Vigo, Spain
Pricila Castelini, Federal University of Technology, Brazil
DeJiu Chen, KTH Royal Institute of Technology, Sweden
Albert M. K. Cheng, University of Houston, USA
Rebeca Cortazar, University of Deusto, Spain
Pietro Cunha Dolci, Santa Cruz do Sul University, Brazil
Beata Czarnacka-Chrobot, Warsaw School of Economics, Poland
José Carlos da Silva Freitas Junior, UNISINOS (Vale do Rio dos Sinos University), Porto Alegre, Brazil
Maria de los Angeles Cosio León, Universidad Politécnica de Pachuca, Mexico
Laura-Maria Dogariu, University Politehnica of Bucharest, Romania

Inês Domingues, Politécnico de Coimbra | ISEC | DEIS, Portugal
Rajesh Doriya, National Institute of Technology, Raipur, India
António Dourado, University of Coimbra, Portugal
Jun Duan, IBM T. J. Watson Research Center, USA
Ahmed El Oualkadi, Abdelmalek Essaadi University, Morocco
Sylvia Encheva, Western Norway University of Applied Sciences, Norway
Ângela Ferreira, Polytechnic Institute of Bragança, Portugal
Manuel Filipe Santos, University of Minho, Portugal
Dirk Frosch-Wilke, University of Applied Sciences Kiel, Germany
Félix J. García, University of Murcia, Spain
Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Valery Y. Glizer, ORT Braude College of Engineering, Karmiel, Israel
Octavio González Aguilar, Felizmente Verde, Mexico
Shuyang Gu, Texas A & M University-Central Texas, USA
Carlos Guerrero, University of Balearic Islands, Spain
Juncal Gutiérrez-Artacho, University of Granada, Spain
Songlin He, New Jersey Institute of Technology (NJIT), USA
Gabriel Henrique de Souza, Federal University of Juiz de Fora, Brazil
Manolo Dulva Hina, ECE Paris School of Engineering, France
Wladyslaw Homenda, Warsaw University of Technology, Poland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, School of Education Intelligent Technology - Jiangsu Normal University, China
Orest Ivakhiv, Lviv Polytechnic National University, Ukraine
Jyoti S. Jagasia, S.P. Jain Institute of Management and Research, India
Maria João Ferreira, Universidade Portucalense, Portugal
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Dimitris Kanellopoulos, University of Patras, Greece
Paweł Karczmarek, Lublin University of Technology, Poland
Andrzej Kasprzak, Wroclaw University of Technology, Poland
Ahmed Khaled, Northeastern Illinois University, USA
Aminollah Khormali, University of Central Florida, USA
Peter Kieseberg, St. Poelten University of Applied Sciences, Austria
Sangpil Kim, Purdue University, USA
Emmanuel Awuni Kolog, University of Ghana, Accra, Ghana
Boris Kovalerchuk, Central Washington University, USA
Panos Kudumakis, Queen Mary University of London, UK
Ganesh Lakshmana Kumar Moganti, VIT-AP University, India
Alex Laier, Universidade Federal Fluminense, Brazil
Bobby Law, Glasgow Caledonian University, Scotland
Olga Levina, TH Brandenburg, University of Applied Science, Germany
Nan Li, The University of Newcastle, Australia
Francisco Carlos Lopes Da Silva, Universidade Federal de Pernambuco, Brazil
Carlos Lopez-Franco, University of Guadalajara, Mexico
Giuseppe Loseto, Polytechnic University of Bari, Italy
Durval Lucas Jr., EPPEN-UNIFESP, Brazil
Oswaldo Luiz Agostinho, University of Campinas / University of São Paulo, Brazil
Ivan Luković, University of Novi Sad, Serbia
Volodymyr Lytvynenko, Kherson National Technical University, Ukraine

Stephane Maag, Institut Mines Telecom / Telecom SudParis, France
Paulo Maio, School of Engineering (ISEP) of Polytechnic of Porto (IPP), Portugal
Alexander Makarenko, Institute of Applied System Analysis at National Technical University of Ukraine (Igor Sikorski Kiev Politechnic Institute), Ukraine
Juho Mäkiö, Hochschule Emden / Leer, Germany
Henrique S. Mamede, INESC-TEC | Universidade Aberta, Lisboa, Portugal
Goreti Marreiros, Institute of Engineering | Polytechnic of Porto, Portugal
Hanna Martyniuk, Mariupol State University, Ukraine
Juan Carlos Montes de Oca López, Autonomous University of the State of Mexico, Mexico
Sobhan Moosavi, Ohio State University, USA
Fernando Moreira, Universidade Portucalense Infante D. Henrique, Portugal
Elaine Mosconi, Université de Sherbrooke, Canada
Paulo Moura Oliveira, UTAD University | INESC-TEC, Portugal
Mary Luz Mouronte López, Universidad Francisco de Vitoria, Spain
Marco Mugnaini, University of Siena, Italy
Rafael Nogueras, University of Malaga, Spain
Robert Ohene-Bonsu Simmons, University of Ghana-BS-OMIS / Zenith University College, Ghana
Alexandru Onea, Technical University "Gh. Asachi' of Iasi, Romania
Constantin Paleologu, University Politehnica of Bucharest, Romania
Ronak Pansara, Tesla, USA
Thanasis G. Papaioannou, Athens University of Economics and Business (AUEB), Greece
Dhaval Patel, IBM TJ Watson Research Center, USA
Bernhard Peischl, AVL List GmbH, Austria
Mansah Preko, Ghana Institute of Management and Public Administration (GIMPA), Ghana
Kornelije Rabuzin, University of Zagreb, Croatia
M. Sohel Rahman, Bangladesh University of Engineering & Technology (BUET), Bangladesh
Ferdinand Regner, University of Vienna, USA
Fernando Reinaldo Ribeiro, Polytechnic Institute of Castelo Branco, Portugal
Éric Renault, ESIEE Paris, France
Federica Rollo, "Enzo Ferrari" Engineering Department - University of Modena and Reggio Emilia, Italy
Lyazid Sabri, El Bachir El Ibrahim University, Algeria
Carlos D. Santos Jr., Universidade de Brasilia, Brazil
Wieland Schwinger, Johannes Kepler University Linz (JKU), Austria
Floriano Scioscia, Polytechnic University of Bari, Italy
Isabel Seruca, Portucalense University, Porto, Portugal
Ashok Sharma, Keshav Memorial Institute of Technology, Hyderabad, Telangana, India
Shwadhin Sharma, California State University, Monterey Bay, USA
Alessio Signorini, Evidation Health, USA
Luis Silva Rodrigues, Politécnico do Porto / ISCAP, Portugal
Pornpat Sirithumgul, Rajamangala University of Technology Phra Nakhon, Thailand
Pedro Sousa, University of Minho, Portugal
Kathryn E. Stecke, University of Texas at Dallas, USA
Sergey Subbotin, National University "Zaporizhzhia Polytechnic", Ukraine
Vitalyi-Igorevich Talanin, Khortytsia National Academy, Zaporozhye, Ukraine
Francesco Tedesco, University of Calabria, Italy
Ion Tutanescu, University of Pitesti, Romania
Radu Vasiu, Politehnica University of Timisoara, Romania
John Violos, École de Technologie Supérieure, Montreal, Canada

Victoria Vysotska, Lviv Polytechnic National University, Ukraine
Piotr Waz, Medical University of Gdansk, Poland
Yair Wiseman, Bar-Ilan University, Israel
Ouri Wolfson, University of Illinois at Chicago / University of Illinois at Urbana Champaign, USA
Krzysztof Wolk, Polish-Japanese Academy of Information Technology, Poland
Kuan Yew Wong, Universiti Teknologi Malaysia (UTM), Malaysia
Mudasser F. Wyne, National University, USA
Muneer Masadeh Bani Yassein, Jordan University of Science and Technology, Jordan
Ali Yavari, Swinburne University of Technology, Australia
Eduard Zharikov, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Automated Social Engineering Tools - Overview and Comparison with Respect to Capabilities and Detectability

Dominik Dana
St. Pölten UAS
St. Pölten, Austria
email: is191805@fhstp.ac.at

Sebastian Schrittwieser
University of Vienna
Vienna, Austria
email: Sebastian.Schrittwieser@univie.ac.at

Peter Kieseberg
St. Pölten UAS
St. Pölten, Austria
email: Peter.Kieseberg@fhstp.ac.at

***Abstract*—The manual effort required by social engineers, to obtain information about people and organizations that are in their interest, is sometimes very high. They therefore strive to automate processes as much as possible. With a few menu entries and selections, it is already possible to export email addresses from social media profiles, as well as to send friend requests and phishing messages to a large number of people. This paper presents the extent to which processes in a Social Engineering attack can already be automated and the tools that can be used to do so. The possibilities and reliability of the freely available tools were evaluated and compared in a practical application. The clustering of the tools is based on the phases of a technical Social Engineering model, derived from the most common Social Engineering frameworks.**

***Index Terms*—Automated Social Engineering, Social Engineering Frameworks, Social Engineering Models, Technical Social Engineering.**

I. Introduction

Social Engineering (SE) is an emerging threat that has evolved along with networking and social media and has attracted increasing attention in recent years. While fraud existed long before, the widespread use of social media and cyberspace provides fertile ground for traditional fraud, as more and more personal information is shared but little awareness and measures are in place to protect it [1]. Especially the widespread and constantly available Social Networking Sites (SNS), are a playground to carry out various forms of phishing attacks [2]. There are advanced phishing attacks, that spread through sharing SNS posts that can lead to information leakage [2], but also targeted attacks, where users working for a specific company are identified and contacted through SNSs and their confidential information is stolen, e.g., via direct messages [3]. Last but not least, habituation effects also lead to various links being clicked, posts being copied, liked, shared and pasted, which ultimately promotes Social Engineering [2]. However, Social Engineering requires a great deal of time spent cultivating relationships, building trust, and then exploiting users to obtain classified information [4]. The tools used for this purpose are, in terms of basic information retrieval, mostly located in the Open Source Intelligence (OSINT) area and rely on a large collection of publicly available information on the Internet about people and organizations. From the social engineers' point of view, the attacks need to be automated, in order to reach many victims and they should behave human-like, so that more victims fall for them [5]. Automation is especially interesting in the reconnaissance phase, as e.g., in the context of an initial information gathering phase, known users would have to be searched for manually for hours on various platforms and social media channels. this task can already be performed by proprietary search engines, across hundreds of platforms, with just a few mouse clicks. It's a similar story with creating phishing messages, or phishing sites. Instead of designing websites yourself, that are used for water-holing or phishing attacks, or instead of sending out a high number of phishing messages via email yourself, a few menu selections or clicks in the respective tools are enough.

This paper describes current automation possibilities which can be used for Social Engineering. The structure of this paper, after a brief introduction and analysis of related work in Section II, it is divided into three main sections, where relevant legal and ethical aspects for the work are considered (Section III), a comparative analysis of Social Engineering phase models and frameworks (Section IV), and the application of the Social Engineering tools themselves (Section V) is conducted. Section VI provides a conclusion and suggestions for future work, including answers to these research questions:

- RQ1: To what extent are freely available Social Engineering supporting tools already automated and what does this mean in terms of Social Engineering?
- RQ2: Which phases of Social Engineering can be handled with the tools?
- RQ3: How do the different tools interact with each other, are there tool suites that start and accompany a complete Social Engineering process?
- RQ4: How reliable are the results of the tools?

II. Related Work

A. Techniques and tools

In addition to the literature by Kevin Mitnick [1] and Christopher Hadnagy [2], publications by Jeremiah Talamantes [5] and Peter Kim [6] were analyzed, in which the first tools from the OSINT domain and the first automated tools, including the Social Engineering Toolkit (SET) and Maltego, were already mentioned. Christopher Handnagy additionally describes in [7] the Social Engineering pyramid as another Social Engineering phase model. An important distinction into the attack categories ”Computer Based” and ”Human Based” within Social Engineering, is made by Wang et al. in [8], similarly in Hussain Aldawood and Geoffrey Skinner’s work [9]. In their paper, Wang et al. also state that technical attacks are becoming increasingly difficult and therefore Social Engineering attacks are on the rise. Furthermore, they assumed the most important attack media to be e-mail, websites and the telephone. Bilikis Banire et al. also describe in [10], that these also represent the most common attack methods from which phishing, vishing and smishing attacks result. In [9], it is also concluded that virtual communities, after personal data is often stored in these platforms, are the largest source of Social Engineering attacks, as little technological know-how is needed once trust has been established with the victims (see also the study from Kenya [11]). Other techniques and tools, especially from the OSINT domain and people-search engines, are described in [12]. However, their main area of application extends to the USA, as application within the EU, due to the General Data Protection Regulation, is not possible due to the personal nature of the data collection without consent.

B. Advanced attacks and automations

A definition of automation is simplistically and naively made in [13] as systems that take over the execution of tasks from humans and thereby simply reduce the amount of work, or attention, that humans need to devote to these tasks. Wang et al. state in [14], that the wide adoption and availability of SNSs, the Internet of Things (IoT), industrial Internet, and mobile devices, have created greater attack surfaces for Social Engineering. The reason behind this is that due to huge amounts of data generated by their use and that people in today’s world share more information about their own personal identities, activities, relationships, locations, and personal interests, as well as their work and work environments on social media combined with the availability of Social Engineering tools, facilitates large-scale Social Engineering attacks. Automated tools, mentioned by Wang et al. in [14], in addition to ways to bypass phishing and deep learning detection, include the automated chat bots of Markus Huber (ASE bot) [15], Tobias Lauinger et al. (Honeybot) [16], amongst others. According to their own statements, compared to the ASE bot, Honeybot moves one step further, by not having humans communicate directly with a bot, but instead initiating a conversation between two real people, with Honeybot acting as a ”Bot in the Middle”, interposed in between. The behavior of Honeybot by changing, replacing, or deleting parts of messages, is individually controllable and the chance, for example, to click on links, which are inserted, or changed by Honeybot, is greatly increased, compared to other chat bots. The project ”Social Network Automated Phishing with Reconnaissance” (SNAP_R) on the other hand, interacts with users on the Twitter platform and sends a machine-generated tweet to its targets, which mostly contains a shortlink. Broken English and shortlinks are accepted on Twitter due to the character limit, which is why the authors see SNAP_R as an extension to SET to automatically distribute phishing messages to a larger target group. The ASE bot, Honeybot and additionally the Koobface bot, spreading as malware through the Facebook social media platform, are also cited as automated Social Engineering tools in a study by Priya Kaul and Deepak Sharma [17].

C. Trust factors as the basis for automation functionality

The trust factors that enable Social Engineering to be successful, are described by Yuki Kano and Tatsuo Nakajima after an experiment [18]. The fact that people are more likely to open suspicious links in messages from Facebook friends than from, e.g., their bank is also addressed by Aron Stern at Kaspersky [19]. The latter go on to state that it is also widespread to clone unrestricted Facebook profiles and send friend requests to friends of this original profile. The goal is to use the cloned profile to send convincing phishing messages or to get the Facebook friends to click on phishing links.

D. Alternative Frameworks

In addition to the classical frameworks and Social Engineering models, presented in a subsequent section, models such as the one described by Tong Wu et al. in [3], consisting of Social Engineering sessions (SES) and Social Engineering dialogues (SED) and the models in [20], which are still in early stages of development represent alternative approaches for new Social Engineering models.

III. Legal and ethical aspects

When compiling and searching for information in the context of Social Engineering, data and information from and about specific individuals are used. This also holds true for the experiments conducted in this study. While malicious attackers will not care about legal or ethical issues regarding private data retrieval, this had, of course, been an issue during our research. Data and information that can be traced back to individuals is considered as personal data in the current version of the General Data Protection Regulation (GDPR), under Article 4 [21], the processing of which is considered to be lawful if there is consent for processing for one or more specific purposes and these are processed appropriately for the purpose and in accordance with the principle of data minimization [21] and appropriate protective measures have also been taken by the processor for the required storage period. Even if information about individuals and institutions can be found freely on the Internet, from an ethical point of view, it cannot and should not be assumed that this information is also freely available for use. However, information can also be interpreted differently in the wrong circumstances, leading to unintended and unfavorable outcomes for the individuals concerned. Another dilemma is that the OSINT sample is minimized or selected depending on the needs of the collector [12]. Thus, important sources might indeed be intentionally neglected in order to achieve a particular result. The handling of legal and ethical aspects is quite different in the related work. This ranges from permissions and questionnaires requested in advance, to simply conducting experiments. Debriefing with participants is rarely held. In order not to unknowingly turn participants into experimental subjects, which has already raised serious ethical concerns [22], own outdated and already known leaked data was searched for first tests with the tools. When processing the data and information found, an attempt was made, despite automation, to take into account the principle of data minimization and purpose limitation as far as possible. Attention was paid to emerging and possibly disadvantageous combinations of the results. The search and test results were not saved after the application of the different tools. In some cases, the tools automatically created log files that contained the results of the search queries. These log files were also deleted at the end of the tests. Screenshots, which were only taken for documentation purposes, had been strongly anonymized so that no conclusions can be drawn from them.

IV. Social Engineering models and frameworks

A standardized formulation of a Social Engineering attack, as well as the sequence and temporal events, allows researchers to compare different Social Engineering attacks with each other. Following, we will compare the following most common phase models and frameworks that divide Social Engineering attacks into phases: The *Cyber Kill Chain (M1)* [23], the *Social Engineering Cycle (M2)* [1], the *Social Engineering Lifecycle (M3)* [24], the *Social Engineering Pyramid (M4)* [7], the *Social Engineering Attack Framework (M5)* [25], the *Cycle of Deception (M6)* [26], the *Social Engineering Attack Spiral (M7)* [27], the *Session and Dialogue Based Framework (M8)* [3], and the *Phase based and Source based Model (M9)* [28]. These models differ most clearly in the area of representation. With M1, the M4, M8, and M9 represent in successive process steps, the M2, M3, M5, M6, and M7, respectively, represent in circuits. The fact that the majority of the researched frameworks use a circular structure to describe Social Engineering attacks, which mostly includes the phases of information gathering, trust exploitation, attack development, and target fulfillment, is also already described in [3]. The circular form provides the possibility of representing the repetition of previous phases when more information is needed, or the goal is not achieved in a single phase [1]. M6 does not provide the opportunity to return to a single previous phase, but provides a sequence of several cycles spherically on top of each other, which makes this framework seem to be very complex at first sight, especially in combination with the inclusion of risks as a three-dimensional component. The models and frameworks also differ in terms of the number of phases. Apart from two models, all other models were designed with fewer than eight phases. M1 is only to a limited extent suitable for Social Engineering attacks, since these types of attacks do not necessarily have to pass through all phases of the framework. Also, the complete section, in which relationships and trust are established, as well as exploited, is completely missing. M4 shows five phases and is the only model that includes reporting as the final step, for traceability and documentation of the process and results. The model M3, as well as model M2, are limited to a total of only four phases with similar names. M2 is seen as a good basis in comparison with M5, but too simplistic, according to [25], as it leaves too much room for interpretation and does not include a debriefing phase, which is intended in M5 to bring the target person back to a normal emotional state. No matter how many phases the respective models and frameworks have, a phase for thorough information gathering is required at the beginning of every successful Social Engineering attack, since the quality of the information obtained contributes significantly to the success of the subsequent phases. Based on the compared models and frameworks, the technical Social Engineering model (TSE) was designed, shown in Figure 1, which was reduced to only three common phases, within which automation with tool support is possible.

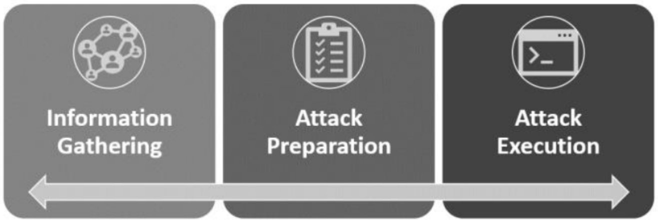

Fig. 1. The technical Social Engineering model (TSE)

A corresponding assignment of the phases of the previously described phase models and frameworks to the phases of the reduced model can be seen in Table I.

V. Tool-supported automation for Social Engineering

While we tackled a lot of different tools during our analysis, we will only be able to give a short outline on the findings in this section, grouping the tools according to the previously defined TSE model.

The tools in the information gathering phase are used to obtain all kinds of information about a (potential) target. Included in this phase are also tools used in reconnaissance and OSINT, as well as social media intelligence (SOCMINT). Still, as this is not an analysis of OSINT tools, we did not further dive into the extreme amount of apps there. We divided the tools into (i) web-based and (ii) locally installed tools.

A. Web based tools for Information Gathering

1) Searching for user data: *Google Dorks* are pre-defined searches that can be executed using the Google Programmable

TABLE I
PHASE ASSIGNMENT

Model	Information Gathering	Attack Preparation	Attack Execution
M1	Reconnaissance	Weaponization, Delivery	Exploitation, Installation, Command & Control, Action on Objectives
M2	Research	Developing Rapport and Trust, Exploiting Trust	Utilize Information
M3	Investigation	Hook	Play
M4	Information Gathering	Attack Planning	Perform Attacks
M5	Information Gathering	Preparation	Exploit Relationship
M6	Map & Bond	Execution	
M7	Recon	Relationship Building, Attack Scenario Building	Execution, Action on Objectives
M8	Attack Preparation		Attack Implementation
M9	Using suitable gates of SNSs to gather information about victim	Using suitable gates of SNSs to reach the victim	Attack

Search Engine for automation as Custom Search Engines (CSEs). Regarding Social Media plattforms, the web application *CheckUsernames* [29] allows the parallel search of over 300 platforms for user-names and linked profiles. Still, the search is very limited, only allowing for exact (partial) matches without additional intelligence. *ReconTool* [30] provides several additional features, like e.g., mindmapping information for dynamic interaction with the search engine. Even more extended functionality is provided by *HOPain Tools* [31], as it also allows searching for pics, videos, detailed content like postings (also allowing filtering like time frames, location or number of likes), as well as bitcoin addresses. Social media platforms can be searched individually or in groups, for many platforms require a respective account.

2) Technology checks: In order to expand the possibilities of pretexts and impersonations for Social Engineering in organisations, it can be helpful to examine existing websites for the technologies used and possible vulnerabilities. The following tools can be used as an alternative to considerably more expensive systems due to higher licence and operating costs. The result of a scan with *BuiltWith* [32] shows the technologies, plugins and hosting provider used for a website, but also other websites that use the same hosting provider, as well as the duration and the respective public IP address under which they were accessible. However, the results can only be viewed to a limited extent in the free version, but are sufficient for searching for *Common Vulnerabilities and Exposures* (CVE) entries and for developing pretexts. Technological information, telephone numbers, email addresses, CVE vulnerabilities with the corresponding CVE number, public IP addresses used, open ports, domain names, cybersquatting domains and much more to determine further attack surfaces and risks of a website can also be found out very conveniently with *SpiderFoot* [33]. The *SpiderFoot HX* version offers an even greater scope and an intuitive, graphical interface that can display all this information in the form of a node graph, where each node can be selected individually. The scan results were surprisingly comprehensive and consistently correct in the short time available and in view of the basic version used. Regarding the analysis of industrial (IoT) devices, Shodan [34], ZoomEye [35], Spyse [36] and Chaos [37] seem to be the most popular. Shodan provides many filter options and requires a familiarisation period in order to achieve useful results. The search results depend on the time in which Shodan has scanned the target system, but contain a high level of detail about the scanned target system. Despite language barriers, ZoomEye could be used with translation software at the time of the research and the presentation of the search results was very similar to Shodan. Surprisingly, Spyse was only able to deliver a few results during the application and using identical target systems and is therefore not very suitable for Social Engineering purposes. Chaos was still at an early stage of development at the time of the research. On the other hand, SynapsInt [38] is a freely available tool that also fits into this categorisation. It provides search results for domains, IP addresses, SSL certificates, email addresses, telephone numbers and Twitter accounts, as well as searching for ransom bitcoin addresses and CVE numbers. The results of a scan with the same inputs as before quickly delivered correct results, a current screenshot of the page, a VirusTotal analysis, the last available entry in the Internet archive Wayback Machine, open ports and information on the hosting provider used. In addition, all domains that can be reached under the same IP address, all subdomains, internal links and related social media links are listed and checked to see whether it is included in various blocklists. The blacklist check also works with entered email addresses. The leak check and the Twitter account check did not work with a private email address that has already been leaked many times.

3) Generate valid email formats: In order to generate the formats for E-Mail addresses of targets, we had a look at the search engines *Email-Format* [39] and*Hunter.io* [40]. Hunter, as well as Email-Format, derive patterns for corresponding email address formats from a large number of email addresses collected via web scans. Of the target domains entered for testing, around a third did not return any search results. The email address formats derived in both web applications appear correct, and sample data is also displayed freely in both applications, although it is not always up to date. Email address format offers, in addition to the identified conventions, a larger list of representative email addresses, as well as (depending on the payment plan) the option of downloading them. In comparison to Email Format, Hunter tends to limit the output, but in addition to more up-to-date data records, it also shows the occurrence of the representative email addresses, which are used to derive the logics for the email addresses.

4) Data breaches and data leaks: Regarding searching data breaches and data leaks, the IntelligenceX platform [41] retrieves results from Dataleaks, Wikileaks, paste sites and even the darknet for search queries, such as email, Bitcoin, MAC and IP addresses, domains, URLs, telephone numbers, credit card numbers and much more. IntelligenceX offers a so-called ”Third Party Search”, in which the search scope can be extended again to several search engines (simultaneously via pop-ups) and, for example, Vehicle Identification Numbers (VIN) can also be searched for. There are separate search functions for social media channels, links to OSINT link lists, as well as file and encoding tools. The test searches carried out delivered surprisingly accurate results. A privately used, knowingly leaked email address that was no longer in use was found, including the password used at the time of use. For another, still privately used email address, it was possible to find out in which data breach the email address appeared and which platform was affected by the breach. Valid access data was also found for other email addresses in the private sphere; Reverse image searches from the third-party search category with randomly uploaded images from private collections and quick Google searches, mostly referred to Adobe stock images, however; three out of ten uploaded images were found. The VIN search was also tested with two different VIN numbers from our own stock, but the search yielded no results.

5) Detecting online times: Online times of targets are especially interesting for targeted attacks. The tool *SleepingTime* [42] was analysed for the SNS platform Twitter and successfully used with several Twitter accounts. SleepingTime analyses the last 1000 tweets of a Twitter account and derives an estimated ”sleep schedule” from the time stamps of the respective tweets, in which the account is least active and in use. *WhatsApp Monitor* [43] is a similar tool that works with browser notifications when a specific WhatsApp contact is available online. The use of the tool sounded very interesting during the research, but could not be used at the time of the tests, as the website was not accessible at the time of the tests.

6) Searching for personal information: Regarding searching for personal information. Suche nach Personendaten, *Webmii* [44] compiles publicly available information about people on the Internet and uses it to generate an online score that is intended to show the availability of the person. Webmii usually lists the results in four sections. (i) the results list, containing the names of people who have interacted with the target person on social media channels, (ii) search results from various newspaper articles, (iii) results from various social media channels and (iv) search results obtained via a Google CSE. At first glance, *IDCrawl* [45] offers a wider range of functions, as it can be used to search not only for people’s names, but also for user names across 17 SNSs. A reverse phone search is also offered. IDCrawl offers the option of an ”opt-out”, where you can exclude yourself from search results. During the test and the search for own findable information, IDCrawl was only able to verify one search result as correct, but the topicality of the result was doubtful, as in this specific case the user profile picture did not match and had already been replaced some time ago. However, the accuracy of the data is not guaranteed in large quantities at Webmii either, as only parts of the information could be considered correct as well. The majority of the search results were not usable, and in some cases links to results could not be opened at all.

B. Locally installed tools for Information Gathering

1) Maltego and alternatives: The data mining tool Maltego [46] is one of the best-known tool suites in the OSINT environment and is almost unique in its range of functions. Depending on the licence and the added plugins, the scope anc capability of the software change. For the tests and the tool comparison with a similar tool, the registered, free Community Edition with eight free plugins was used, which provides a certain number of credits depending on the query used. With six out of one hundred available credits, it was already possible to find domain information, whois entries, company owner data, email addresses, telephone numbers, public IP addresses, all plugins used on the website, as well as archived versions of these since 2009. Audit reports from American companies in the same business sector were also found in the Maltego document cloud. However, these were not related to the exemplary target company. As part of the research, a comparable alternative, or supplement, to Maltego could be found, which, despite critical voices [47], was implemented, licensed and tested for comparison: Lampyre [48], which is only available on Windows platforms and offers a similar overview to Maltego’s Transformation Hub in the so-called ”List of requests”. The advantage of the software is that the plugins do not have to be installed individually; a selection (and like Maltego, the entry of a corresponding API key) of the modules to be used, the underlying and desired tasks, as well as the required parameters, is sufficient for the start.

In direct comparison, Maltego is clearer and more structured to use. Lampyre is simpler in terms of usability, the results are mostly displayed in tabular form and graphical dependencies are only possible in isolated cases. Furthermore, it is partially unstable, e.g., during the application tests, various result tabs suddenly stopped responding and could no longer be selected, meaning that the results could no longer be viewed.

Of the plugins already included, Lampyre offers a selection of search criteria that could not yet be found in Maltego and vice versa. These included, for example, the search for IMEI numbers, WLAN SSIDs or Vehicle Identification Numbers (VIN) in Lampyre, while Maltego offers the Wayback Machine, Movie Database, Blockchain.info or Google Maps Geocoding, which are regularly updated and expanded in both applications. Within Maltego, the origins of the search results and the use of the search providers are traceable. At first glance, it is not possible to recognise where Lampyre obtains the results of the transformations if the search provider is not described in the tasks. In the transformations to the same target organisation, more search results could be achieved with Maltego with less known data. The reliability of the data was also higher in Maltego; for example, the public company Facebook account could be found with Maltego, whereas Lampyre returned error messages for these transformations.

2) Searching for user and personal data: Regarding searching of account or personal data, *CrossLinked* [49] allows for automated searches in LinkedIn by filtering external search engine results, so-called *Search Engine Scraping*, thus not requiring account data for searching. When verifying the results, it was found that although they were plausible (by randomly comparing the results with the online employee directory), but the results also included every person who had specified St. Pölten UAS in their LinkedIn profile, not only employees. When searching for another organisation without results, it turned out that links from search engines were also counted as results. The tools UserReCon [50] and Userrecon-py [51], Nexfil [52], Sherlock [53], Us3R-F1nD3R [54] and Thorndyke [55] promise similar functionalities with search scopes spanning several hundred social media platforms. From the own descriptions and command references of these tools, it is clear that Sherlock is the only application that can process several search entries as well as prepared lists in one search run. The tools are very similar in their use and appearance, as are the results. In addition to existing social media accounts, the Instagram test account `@dominikhatkeininsta` could also be found as a registered user on several platforms according to the search results. As the test account was only created for Instagram, it can be assumed that the search results are not valid, except for the Instagram platform. This was confirmed when checking the search results for the Twitter and Reddit platforms. Buster [56] can also find users on social media platforms, but the search scope is extended to the generation of email addresses, which are provided from possible data breaches, pastes and reverse-whois queries. Buster also shows the sources of results, as the services of Hunter.io, among others, are used in the background.

3) Technology checks: Regarding checking for technology, *TheHarvester* [57] is already pre-installed under Kali Linux and offers searches for domain information and Google dorks in 38 different search engines. Corresponding API keys are required for use, and the search results can be limited in scope. In the test, the search engines did not work properly under version 4.0.3, despite reinstalling the tool; under version 3.2.2, search results could at least be obtained via Google, although most of them were not valid. Raccoon [58] is basically an extension of nmap. The tool is still in the development stage and the focus is on simplicity. The convenience of using Raccoon lies in the fact that the parameterisation of the nmap scans is already predefined by the tool. In addition to the possibilities of nmap scans and subdomain enumeration, Raccoon should also be able to search cookies, recognise web application firewalls and provide information on CMS, web servers and Whois queries. However, this did not work in the test (without nmap scan). A coherent subdomain enumeration could be carried out using three different domains, including that of the St. Pölten University of Applied Sciences, with Sublist3r [59], Sn0int [60] and Frogy [61], whereby Frogy also uses Sublister in the enumerations. Sublister also offers the option of a port scan and a brute force scan, which were not performed. Under Sn0Int, the subdomain enumeration is only a small part of the functionalities. Frogy was still under development at the time of research and testing. In addition to finding IPs, domains and subdomains, it is also designed to find live websites and login portals. What is particularly interesting about this tool is that it can access the Chaos-database. Another tool suggested in the information retrieval communities is ReconSpider [62], which is a tool for the automated scanning of IP and e-mail addresses, websites, telephone numbers, DNS and domain information, but also for searching data breaches. ReconSpider was able to consistently return correct data in the test entries, but occasionally crashed with Python errors when making entries in the menus for whois and domain queries.

4) Export data from social media: Regarding the export of data from social media profiles, ReconSpider can display information of Facebook, Twitter and Instagram accounts, but this is limited to the name, number of followers and profile description and cannot be exported. The tool OSINTGram [63] on the other hand requires a valid Instagram account to be usable. For export, optionally in *.txt and *.json file formats, all addresses that can be read from posted image material, all texts and comments that have been added to posted images, the number of followers of the target account, as well as the number of accounts that the target account follows, account information, as well as the number of all likes, hashtags, a list of all links of the target account and a list of all accounts that have commented on posts of the target account at any time are available. The ”fwersemail”, ”fwingsemail”, ”fwersnumber” and ”fwingsnumber” functions are particularly interesting features for Social Engineering purposes, each of which creates a list of telephone numbers and email addresses (if specified in the respective accounts) of the followers and followings. In the test application with the Instagram account of the St. Pölten University of Applied Sciences, several thousand pieces of data were found. With a private test account, the consistently correct information could be provided in lists within a short time. Sterra [64] also exports follower and following accounts, including their account ID, user name, specified name, biography, number of posts and links to the respective account in CSV files. Within the application, it is also possible to compare follower lists with each other and filter them for similarities or differences. As Sterra works directly with Instagram’s API, the reliability of the data is guaranteed. List comparisons can also be carried out with the Python tool Insta-Extract [65] and these are simpler in the application than within Sterra, but not as extensive. What works well on the social media platform Instagram in the test applications also works with two other applications on the Twitter platform. Twi1tter0s1nt [66], also known as

TWINT and twosint, offers pretty much the same functions on the command line that TinfoLeak [67] also offers in a GUI. These include general searches for user names, searches for geocoded tweets (if the geolocation data in the tweets can be read), tweets in a specific time window, filtering for specific terms, but also exporting the number of followers. In addition to exports in several file formats, TWINT also offers to translate tweets directly into other languages using Google Translate. A time limit between individual scrapes can also be set for scraping tweets using the ”min-wait-time” parameter. TinfoLeak is easier to use with the graphical user interface, where the desired operations are simply ticked and provided with the corresponding values or data.

C. Tools for the attack preparation phase

The attack preparation phase includes those tools that, depending on the selected attack scenario, are useful for preparing attacks, e.g., for preparing payloads or phishing messages.

1) Preparing Payloads: To prepare suitable payloads, already generated and available versions [68] can be used, or new ones can be generated. In addition to one of the best-known tools, the Social Engineering Toolkit (SET) [69], the PowerShell script [70] designed by Matt Nelson and Matt Robinson is also suitable for this, which creates an Excel document after the run that creates a Meterpreter shell when called on the target system. It also persists in the Windows registry and in the user directory so that it can be executed again when the system is restarted. A connection to the infected system can be established via Meterpreter Reverse HTTP and HTTPS. The MacroPack tool from Emeric Nasi [71] is more up-to-date and has an extended range of functions compared to the PowerShell script and requires a functioning and registered Office installation on the system on which the payload is to be integrated into an Office file. The tool also offers the service of code obfuscation so that the malicious code in the Office markers is not so easily recognisable and it supports all Microsoft Office document versions and shortcut files in the community version. The Pro version offers an even wider range of functions and can be used on existing Office files. During the tests, the generation of payloads with the PowerShell script did not work, despite changes in the execution guidelines, which originally prevented the execution of the script. For the execution and use of MacroPack, it is recommended to adjust the Windows security settings, as these prevent execution and classify the tool as a serious threat. The tool Social_X, which was supposed to be able to generate Trojans with its own reverse shell and in the form of an *.exe file, unexpectedly failed to install correctly and terminated after several start attempts. Documentation for the tool was not available at the time of testing and a linked YouTube video was no longer available. Social_X is therefore only mentioned as another possibility, as the last commit on GitHub was only a few months old and the error could possibly be fixed soon.

SET, which is included in every current installation of Kali-Linux, offers the option of automatically manipulating data carriers, so that malicious code can be automatically executed on removable media via the autorun function. This can be done via an executable file, which is executed via the autorun.inf file contained on the removable storage device, or via a file format exploit to bypass any security warnings. TrustSec also provides detailed documentation on SET. SET worked out of the box and, with the TrustSec documentation, was simple and reliable.

2) Recognising tone and emotions in texts: In order to test messages for the effect of emotions, the Tone Analyser [72] from IBM was tested during the research into automated Social Engineering tools. The Tone Analyzer can be freely tested online in a web form and recognises the emotions and tones of voice contained in an entered text via machine learning analysis. The Node.js version of the Tone Analyser [73] offers free analyses and support for several languages and files directly for the first 1000 API calls per month after registration in the IBM Developer Cloud. To quickly test the analysis, the following sample texts were entered for analysis:

- *Positive emotion: ”Dominik likes doing his master thesis all night long :-)”*
- *Negative emotion: ”Dominik does not like doing his master thesis all night long :-(”*

Tone Analyzer carried out the analyses with respect to the emotions ”Confident”, ”Joy” and ”Sadness” and classified the strength of the expressions in the messages with different colours. In further tests, with different text fragments, Tone Analyser also classified in the direction of ”Analytical” and ”Tentative”. We did not conduct any further tests, as this work is not focusing on the capabilities of emotion detection, but on the general usability of the tools.

3) Bot preparation: Parts of a Social Engineering attack can also be carried out by bots, depending on the target and attack scenario selected. Implementations of Twitter bots, modelled on Realboy [74] or SNAP_R [75], for example, can be used in the attack execution phase for the automated distribution of phishing links. In the attack preparation phase, corresponding Twitter accounts can be created, filled with content and equipped with a network of followers and followings to make them more credible. Both bots, Realboy and SNAP_R, were not tested and evaluated in this work, as there exists ample recent work analyzing bot preparation for Social Engineering.

D. Tools for the attack execution phase

The attack execution phase includes all those tools that can directly execute a Social Engineering attack. While researching the relevant tools, it emerged that the automation of attack tools is described almost exclusively in terms of phishing with website cloning, mass emails and occasionally the use of bots.

1) Phishing with website cloning: SET offers the possibility to clone any website into a website with phishing or hosting multiple attack methods. The cloned page is ready for use as soon as it is entered, and the user data entered is displayed in colour directly on the command line. Zphisher [76] works in a similar way, also with regard to website cloning. Unlike

SET, however, Zphisher only offers ready-made templates for phishing pages and does not clone individual pages. This is also the case with phishEye [77], although it is the only tool listed that also offers the option of cloning websites for mobile devices. During the application tests, it was found that although Blackeye [78] provides a number of templates for social media platforms, these could not be tested directly as an error occurred when generating the phishing links and no links were generated or output for use. SocialFish [79] could also not be fully tested and evaluated, as module error messages occurred within the main application when the application was started, despite all installed requirements and dependencies. The documentation for the app is very brief and rudimentary, so the error could not be rectified. Cloning the GitHub repository again did not help either. StormBreaker [80] extends the list of phishing tools mentioned in this subsection with a tool that cannot clone websites like the others mentioned so far, but instead generates pages and links with the help of Ngrok with a maximum of two inputs, which enable access to the camera, microphone and location data of the end devices. The location data is returned with a Google Maps link. StormBreaker also offers an ”OS Password Grabber” function, which is designed to transfer the passwords entered. During the tests, there were difficulties with this part of the function, as either the links to be sent were not generated or the application did not respond to inputs. However, the functionality of accessing the microphone, camera and location data of the potential target's device is only possible if all phishing warnings displayed by the current browser generations are ignored when the page is accessed and authorisation to access the microphone, camera or location is granted accordingly.

2) Mass mailer: In addition to individual (spear) phishing messages, the Social Engineering toolkit SET can also be used to set up the sending of mass emails. The email addresses of the recipients can be provided via a separate text file, and a separate mail server or sending via Google Mail (gmail) can be selected for sending. The message content is accepted in both HTML and plain text formatting. A test mailing with SET was carried out using our own mail server. As expected, the e-mail message was classified as SPAM and filtered accordingly. In many cases it is not clear before sending a message whether it will be blocked by a mail server or whether it will be delivered without any problems. In order to check the behaviour of mail servers when a message is received, a check can be carried out in advance using Phishious [81]. According to its own information, Phishious is the only tool to date that makes it possible to scan phishing attacks via email. Phishious analyses the header data of undeliverable messages and can therefore predict whether a message will be delivered or classified as spam or junk mail. Another mass mailer tool can be seen in Catero [82]. In addition to the option of cloning websites, Catero offers various ways of sending automated messages and can be controlled entirely via the command line interface (CLI). Catero supports sending messages via Twillo accounts for sending SMS messages, sending via LinkedIn accounts and WebMail services, Google Voice and iMessage.

3) Bot utilization: Another type of automation of Social Engineering using bots is the preparation for the use of SMSRanger [83], which is based on a Telegram bot. SMSRanger sends automated messages to people, in each case on behalf of a bank, and asks them to enter OTP codes (One Time Password) in corresponding websites or in an automated call via a voice bot using the telephone keypad. The service contains daily updates, is available in various languages and is subject to a charge. At the time of research, calls from and to various countries, including German-speaking countries, were also included for USD 425 per month. SMSRanger is controlled via a Telegram chat. This bot was also not activated for security, legal and ethical reasons. With Honeybot [16], Tobias Lauinger et al. have already shown that conversations between two people can be started and influenced and controlled by the bot-in-the-middle, which can also be used to carry out attacks. The *Honeybot* tool is only mentioned in this section and was not tested or evaluated in this paper, as this has already been done in related work.

VI. Conclusions and Future Work

A. Conclusion

In order to better understand automation in the area of Social Engineering and to be able to search for suitable tools and tool suites, but also to be able to classify automation in different phases of Social Engineering, various Social Engineering frameworks were analyzed and compared with each other. It was found, that the various models often differ in the number of phases and that classifying automated tools into individual phases in this way is not purposeful. Therefore, a compression to common phases of all models was carried out and from this, the *technical Social Engineering model* was derived. Furthermore, the individual phases of the described frameworks from other works were assigned to the phases of the technical Social Engineering model, using phase mapping. A similar and comparable abstract model could not be found by the time of writing this paper. For the listing and clustering of the automation-supported Social Engineering tools within section V, the individual phases of the technical Social Engineering model were used. The clustering of the corresponding tools shows that in the information gathering phase there exists a lot of diversity and a large number of tools allowing for the most automation possibilities, as there is a large community of interested parties and contributors from the OSINT area. This was shown not only in the short intervals, in which tools and updates to existing tools are published, but also in the linguistic diversity in which the applications are written. The short intervals make it impossible to list and test all of the available tools. A selection of over 140 tools, written in German or English language, were subjected to a practical application and comparison, where it was found that information retrieval within the European Union has become more difficult since the introduction of the General Data Protection Regulation, and that web applications for information retrieval in particular largely only provide results in the states of the USA. There are, in the applications that are available free of charge, often query limits implemented, that only allow a small number of queries within a certain period of time. Registering to receive an API key, shifts the query limits, depending on the chosen tariff and tool, but also the up-to-dateness, as well as the amount of data provided. Within this work, only freely available tools and API keys free of charge were used. Furthermore, it became apparent that results must be manually checked for plausibility and validity before further use, since the results of automated tools, with the exception of those that read information directly from social media platforms, are not necessarily correct or appropriate. When using the tools to gather information from social media platforms, most of the platforms require a registered account. When using the tools to prepare for attacks, it has been shown that automation can be summarized to the preparatory generation and creation of payloads and bots, as well as support in the formulation of texts. When using the tools in the attack execution phase, the researched and mentioned tools could be summarized into the categories ”phishing with website cloning”, ”mass mailers” and the ”use of bots”. A completely end-to-end automated software that can map a complete Social Engineering attack in all of its phases could not be found. The two tools Maltego and SET are, after completion of the tests and comparisons, the most functional and reliable tools.

B. Answering the research questions

The research questions posed at the beginning of the paper can thus be answered as follows.

a) RQ1: The freely available Social Engineering tools are automated in the sense that recurring query and search work can be performed automatically, thus significantly reducing manual effort. Searches can be performed via web applications, but also locally installed tools. Web applications shine with simpler operation and fast availability. The automation possibilities are greater when using the APIs of the search providers and platforms, since the results can be processed further in an automated manner if the appropriate output is available. A completely automated solution could not be found and is correspondingly difficult to develop, since Social Engineering can be very dynamic and the validation and decision as to, whether data and information fit a current target and scenario, must be made manually by the social engineers themselves. Automation is also already available in the execution of attacks and in the corresponding preparation, and the corresponding tools are already very easy to use. During the application and writing of the paper, it has become evident, that the selection and availability of automated tools for the purpose of information retrieval is the largest. One justification of this can be the availability of a large community from the OSINT domain. Another reason can be seen in the greater availability of these tools, among other things for awareness-raising measures. With regard to quality, it was stated in the paper that the scope of the search and the number of permitted searches are subject to certain limitations, depending on the platform and are only increased with paid subscriptions. This also affects the reliability of the search results. Regarding availability, interesting tools could be collected during the research phase, but during the testing and application phase a few weeks later, they were no longer available and applicable. The free availability of automated Social Engineering tools means, that these tools are available to any person, can be used by any person, and thus any person can easily use Social Engineering techniques, without much effort or in-depth knowledge. Due to the availability of ready-to-use system environments, pre-configured systems are provided, which, with a simplified graphical user interface, can deliver usable results within a short period of time, even for beginners.

b) RQ2: The various frameworks and phase models differ in terms of the number of phases, as well as the processes within the phases themselves. Generally speaking, the phases of reconnaissance and the phases, within which attacks take place, are best served and supported by automation. Due to the number of differences between the various Social Engineering models, it was not possible to map the automated tools to all models, which is why the abstract technical Social Engineering model was derived from the other analyzed frameworks.

c) RQ3: Records must be manually selected, validated, and formatted for the next tool. Toolsuites, that offer multiple options and whose functionalities can be extended with plugins, such as the mentioned tools Maltego, Lampyre, or also Spiderfoot HX, can transfer results into new searches most easily. These tools cannot guide a complete Social Engineering process, but they accompany a large part of it very reliably.

d) RQ4: The results of the tools depend very well on the respective mode of operation itself. While some of the tools, in order to deliver search results, make use of searching in archive databases or searching crawled and scanned websites, some tools access live data directly. In free program versions, live data was only analyzed by tools that search across social media platforms, for example Tinfoleak or OSINTGram, and required a corresponding user account. Searching crawled pages affects the reliability and the up-to-dateness of the results.

C. Future Work

As an extending future work, paid API keys of the applications, offering higher-value subscriptions, can be purchased and the results compared between the premium versions. Under appropriate legal and ethical coverage, extended use of the tools, including for awareness and training purposes, is conceivable. In the light of the increasing number of phishing messages, the comparison and use of professional Social Engineering tools, such as CanIPhish, GoPhish and SET, in the corporate context is a possibility. From this, organizational countermeasures, suitable for the respective organization, can be derived and an anti-Social Engineering framework can be designed. In the analysis of free tools, it was found, that search platforms, including Hunter.io, Shodan.io, as well as _IntelX, were used in common by some tools. In the context of a future work, the comparison of which and how many search engines and databases are used in the background, together and whether the results, despite use of same sources, differ.

Also, the development of an automated Social Engineering application, which can link the applications and results of different Social Engineering tools together, can be initiated.

REFERENCES

[1] K. D. Mitnick and W. L. Simon, *The art of deception: Controlling the human element of security*. John Wiley & Sons, 2003.

[2] C. Hadnagy, *Social engineering: The art of human hacking*. John Wiley & Sons, 2010.

[3] K. Zheng, T. Wu, X. Wang, B. Wu, and C. Wu, "A session and dialogue-based social engineering framework," *IEEE Access*, vol. 7, pp. 67781–67794, 2019.

[4] E. D. Frauenstein and S. V. Flowerday, "Social network phishing: Becoming habituated to clicks and ignorant to threats?," in *2016 Information Security for South Africa (ISSA)*, pp. 98–105, IEEE, 2016.

[5] J. Talamantes, *The Social Engineer's Playbook: A Practical Guide to Pretexting*. Hexcode Publishing, 2014.

[6] P. Kim, *The hacker playbook 2: practical guide to penetration testing*. Secure Planet, LLC, 2015.

[7] C. Hadnagy, *The Science of Human Hacking*. Wiley Publishing Inc., 2018.

[8] Z. Wang, H. Zhu, P. Liu, and L. Sun, "Social engineering in cybersecurity: a domain ontology and knowledge graph application examples," *Cybersecurity*, vol. 4, pp. 1–21, 2021.

[9] H. Aldawood and G. Skinner, "An advanced taxonomy for social engineering attacks," *International Journal of Computer Applications*, vol. 177, no. 30, pp. 1–11, 2020.

[10] B. Banire, D. Al Thani, and Y. Yang, "Investigating the experience of social engineering victims: Exploratory and user testing study," *Electronics*, vol. 10, no. 21, p. 2709, 2021.

[11] J. Obuhuma and S. Zivuku, "Social engineering based cyber-attacks in kenya," in *2020 IST-Africa Conference (IST-Africa)*, pp. 1–9, IEEE, 2020.

[12] N. A. Hassan and R. Hijazi, *Open source intelligence methods and tools*. Springer, 2018.

[13] C. P. Janssen, S. F. Donker, D. P. Brumby, and A. L. Kun, "History and future of human-automation interaction," *International journal of human-computer studies*, vol. 131, pp. 99–107, 2019.

[14] Z. Wang, L. Sun, and H. Zhu, "Defining social engineering in cybersecurity," *IEEE Access*, vol. 8, pp. 85094–85115, 2020.

[15] M. Huber, "Automated social engineering, proof of concept," *Royal Institute of Technology Stockholm*, 2009.

[16] T. Lauinger, V. Pankakoski, D. Balzarotti, and E. Kirda, "Honeybot, your man in the middle for automated social engineering.," in *LEET*, pp. 1–8, 2010.

[17] P. Kaul and D. Sharma, "Study of automated social engineering, its vulnerabilities, threats and suggested countermeasures," *International Journal of Computer Applications*, vol. 67, no. 7, pp. 13–16, 2013.

[18] Y. Kano and T. Nakajima, "Trust factors of social engineering attacks on social networking services," in *2021 IEEE 3rd global conference on life sciences and technologies (LifeTech)*, pp. 25–28, IEEE, 2021.

[19] A. Stern, "Social networkers beware: Facebook is a major phishing portal," *Kaspersky Lab*, vol. 23, 2014.

[20] K. Kikerpill and A. Siibak, "Mazephishing: The covid-19 pandemic as credible social context for social engineering attacks," *Trames: A Journal of the Humanities and Social Sciences*, vol. 25, no. 4, pp. 371–393, 2021.

[21] EUR-Lex, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," 2016.

[22] M. Huber, S. Kowalski, M. Nohlberg, and S. Tjoa, "Towards automating social engineering using social networking sites," in *2009 International Conference on Computational Science and Engineering*, vol. 3, pp. 117–124, IEEE, 2009.

[23] "The cyber kill chain." https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, Last accessed on Feb. 8, 2024.

[24] "What is social engineering." https://www.imperva.com/learn/application-security/social-engineering-attack/, Last accessed on Feb. 8, 2024.

[25] F. Mouton, M. Malan, L. Leenen, and H. S. Venter, "Social engineering attack framework," in *2014 Information Security for South Africa*, pp. 1–9, IEEE, 2014.

[26] M. Nohlberg and S. Kowalski, "The cycle of deception: a model of social engineering attacks, defenses and victims," in *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA)*, University of Plymouth, 2008.

[27] A. Cullen and L. Armitage, "The social engineering attack spiral (seas)," in *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*, pp. 1–6, IEEE, 2016.

[28] A. Algarni, Y. Xu, and T. Chan, "Social engineering in social networking sites: the art of impersonation," in *2014 IEEE International Conference on Services Computing*, pp. 797–804, IEEE, 2014.

[29] "Checkusernames." https://checkusernames.com/, Last accessed on Feb. 8, 2024.

[30] "Recontool." https://recontool.org/#mindmap, Last accessed on Feb. 8, 2024.

[31] "Hopain tools." https://osint.hopain.cyou/, Last accessed on Feb. 8, 2024.

[32] "Builtwith." https://builtwith.com/, Last accessed on Feb. 8, 2024.

[33] "Spiderfoot." https://www.spiderfoot.net, Last accessed on Feb. 8, 2024.

[34] "Shodan." https://www.shodan.io, Last accessed on Feb. 8, 2024.

[35] "Zoomeye." https://www.zoomeye.org, Last accessed on Feb. 8, 2024.

[36] "Spyse." https://spyse.com, Last accessed on Feb. 8, 2024.

[37] "Chaos." https://chaos.projectdiscovery.io, Last accessed on Feb. 8, 2024.

[38] "Synapsint." https://synapsint.com/index.php, Last accessed on Feb. 8, 2024.

[39] "Email-format." https://www.email-format.com, Last accessed on Feb. 8, 2024.

[40] "Hunter.io." https://hunter.io, Last accessed on Feb. 8, 2024.

[41] "Intelligencex platform." https://intelx.io/, Last accessed on Feb. 8, 2024.

[42] "Sleepingtime." http://sleepingtime.org/, Last accessed on Feb. 8, 2024.

[43] "Whatsapp monitor." https://github.com/rizwansoaib/whatsapp-monitor, Last accessed on Feb. 8, 2024.

[44] "Webmii." https://webmii.com/, Last accessed on Feb. 8, 2024.

[45] "Idcrawl." https://www.idcrawl.com/, Last accessed on Feb. 8, 2024.

[46] "Maltego." https://www.maltego.com, Last accessed on Feb. 8, 2024.

[47] "Be careful what you osint with." https://keyfindings.blog/2020/03/23/be-careful-what-you-osint-with/, Last accessed on Feb. 8, 2024.

[48] "Lampyre." https://lampyre.io, Last accessed on Feb. 8, 2024.

[49] "Crosslinked." https://github.com/m8r0wn/crosslinked, Last accessed on Feb. 8, 2024.

[50] "Userrecon." https://github.com/vijaysahuofficial/UserReCon?fbclid=IwAR0NAexz0KEyNDvJSOfSyOzsw9Z0Hc9j7AtB38ZK5AsI-5vupj46Dh95o-o, Last accessed on Feb. 8, 2024.

[51] "Userreconpy." https://github.com/lucmski/userrecon-py, Last accessed on Feb. 8, 2024.

[52] "Nexfil." https://github.com/thewhiteh4t/nexfil?fbclid=IwAR0NAexz0KEyNDvJSOfSyOzsw9Z0Hc9j7AtB38ZK5AsI-5vupj46Dh95o-o, Last accessed on Feb. 8, 2024.

[53] "Sherlock." https://sherlock-project.github.io, Last accessed on Feb. 8, 2024.

[54] "Us3r-f1nd3r." https://github.com/machine1337/userfinder?fbclid=IwAR3sCrgnkLvCUuLHP5VT6X8pVUvfyb8W0DZPenHVDA-VTIq3Et3zwMldWL0, Last accessed on Feb. 8, 2024.

[55] "Thorndyke." https://github.com/rly0nheart/thorndyke?fbclid=IwAR1qnLkHJOC0a-OdlRXk1svN8ypAo6BvuQTrA8L5E4VYxbgI4UzVXLUz6PE, Last accessed on Feb. 8, 2024.

[56] "Buster." https://github.com/sham00n/buster, Last accessed on Feb. 8, 2024.

[57] "The harvester." https://www.kali.org/tools/theharvester/, Last accessed on Feb. 8, 2024.

[58] "Racoon." https://github.com/evyatarmeged/Raccoon, Last accessed on Feb. 8, 2024.

[59] "Sublist3r." https://github.com/aboul3la/Sublist3r, Last accessed on Feb. 8, 2024.

[60] "Sn0int." https://github.com/kpcyrd/sn0int, Last accessed on Feb. 8, 2024.

[61] "I am the frogy." https://github.com/iamthefrogy/frogy, Last accessed on Feb. 8, 2024.

[62] "Recon spider." https://github.com/bhavsec/reconspider, Last accessed on Feb. 8, 2024.

[63] "Osintgram." https://github.com/Datalux/Osintgram, Last accessed on Feb. 8, 2024.

[64] "Sterra." https://github.com/novitae/sterraxcyl, Last accessed on Feb. 8, 2024.

[65] “Insta extract.” https://github.com/JavideSs/insta-extract, Last accessed on Feb. 8, 2024.

[66] “Tw1tteros!nt.” https://github.com/falkensmz/tw1tter0s1nt, Last accessed on Feb. 8, 2024.

[67] “Tinfoleak.” https://github.com/vaguileradiaz/tinfoleak, Last accessed on Feb. 8, 2024.

[68] “Rubber ducky.” https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads, Last accessed on Feb. 8, 2024.

[69] “The social engineering toolkit.” https://www.trustedsec.com/tools/the-social-engineer-toolkit-set, Last accessed on Feb. 8, 2024.

[70] “Powershell script.” https://github.com/enigma0x3/Generate-Macro, Last accessed on Feb. 8, 2024.

[71] “Macropac.” https://github.com/sevagas/macro_pac, Last accessed on Feb. 8, 2024.

[72] “Toneanalyzer.” https://tone-analyzer-demo.ng.bluemix.net/, Last accessed on Feb. 8, 2024.

[73] “Toneanalyzer.” https://github.com/watson-developer-cloud/tone-analyzer-nodejs, Last accessed on Feb. 8, 2024.

[74] “Realboy.” http://ca.olin.edu/2008/realboy, Last accessed on Feb. 8, 2024.

[75] “Snap_r.” https://github.com/zerofox-oss/SNAP_R, Last accessed on Feb. 8, 2024.

[76] “Zphisher.” https://github.com/htr-tech/zphisher, Last accessed on Feb. 8, 2024.

[77] “phisheye.” https://github.com/sky9262/phishEye?fbclid=IwAR1hdh_rgxK24YB4gi_2FYtY4D7Qrxt05WPwU2ZKGa1gXCh7ln7MF0RfmyI, Last accessed on Feb. 8, 2024.

[78] “Blackeyeq.” https://github.com/An0nUD4Y/blackeye, Last accessed on Feb. 8, 2024.

[79] “Socialphish.” https://github.com/UndeadSec/SocialFish, Last accessed on Feb. 8, 2024.

[80] “Stormbreaker.” https://github.com/ultrasecurity/Storm-Breaker?fbclid=IwAR2HX8B5RRQ2f-yRlWndAjxZM1PKfZxVZq-GM-9C_f317IFWGjdAVhcRHaY, Last accessed on Feb. 8, 2024.

[81] “Phishious.” https://github.com/Rices/Phishious?fbclid=IwAR2OhR2kRNkAyyGS7skSzOwlRPEWDcxzFwzohAFuj_coiQFlMdq7t9wlh_k, Last accessed on Feb. 8, 2024.

[82] “Catero.” GitHub - Section9Labs/Catero: Catero - Social Engineering Framework, Last accessed on Feb. 8, 2024.

[83] “Smsranger.” https://smsranger.io/, Last accessed on Feb. 8, 2024.