IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 31, 2021, accepted July 2, 2021, date of publication August 11, 2021, date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104260

Offensive Security: Towards Proactive Threat Hunting via Adversary Emulation

ABDUL BASIT AJMAL[1], MUNAM ALI SHAH[1], CARSTEN MAPLE[2,3], (Member, IEEE), MUHAMMAD NABEEL ASGHAR[4], AND SAIF UL ISLAM[5]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]Secure Cyber Systems Research Group, WMG, University of Warwick, Coventry CV4 7AL, U.K.
[3]The Alan Turing Institute, London NW1 2DB, U.K.
[4]Department of Computer Science, Bahauddin Zakariya University, Multan 60800, Pakistan
[5]Department of Computer Science, Institute of Space Technology, Islamabad 44000, Pakistan

Corresponding authors: Carsten Maple (cm@warwick.ac.uk) and Saif ul Islam (saiflu2004@gmail.com)

This work was supported in part by EP/R007195/1 (Academic Centre of Excellence in Cyber Security Research - University of Warwick); EP/N510129/1 (The Alan Turing Institute); EP/S035362/1 (PETRAS National Centre of Excellence for IoT Systems Cybersecurity), and in part by the Higher Education Commission, Pakistan, Technology Development Initiative from Cyber Security Lab COMSATS University Islamabad under Grant TDF-206.

ABSTRACT Attackers increasingly seek to compromise organizations and their critical data with advanced stealthy methods, often utilising legitimate tools. In the main, organisations employ reactive approaches for cyber security, focused on rectifying immediate incidents and preventing repeat attacks, through protections such as vulnerability assessment and penetration testing (VAPT) security information and event management (SIEM), firewalls, anti-spam/anti-malware solutions and system patches. Such system have weaknesses in addressing modern modern stealthy attacks. Proactive approaches, have been seen as part of the solution to this problem. However, approaches such as VAPT have limited scope and only works with threats that have already been discovered. Promising methods such as threat hunting are gaining momentum, enabling organisations to identify and rapidly respond to any potential attacks, though they have been criticised for their significant cost. In this paper, we present a novel hybrid model for uncovering tactics, techniques, and procedures (TTPs) through offensive security, specifically threat hunting via adversary emulation. The proposed technique is based on a novel approach of inducing adversary emulation (mapping each respective phase) model inside the threat hunting approach. The experimental results show that the proposed approach uses threat hunting via adversary emulation and has countervailing effects on hunting advance level threats. Moreover, the threat detection ability of the proposed approach utilizes minimum resources. The proposed approach can be used to develop the offensive security-aware environment for organizations to uncover advanced attack mechanisms and test their ability for attack detection.

INDEX TERMS Offensive security, threat hunting, proactive defense, red teaming, adversary emulation, mitre ATT&CK, threat analysis.

I. INTRODUCTION

Modern computer systems often hold information which is of significant value to competitors, foreign nation states or criminal actors. As these systems become increasingly connected, the threat of attack by adversaries increases. As a result many enterprise networks have been, or currently are, under cyber attack. The market for security tools to help protect systems and identify attacks has grown significantly over recent years, but many tools are not interactive in nature, often working on some specific logic – for example watching a specific gateway and searching for specific threats. In such cases, the security function in an organisation aims to identify active threats. This approach is based on actions that have been, or are being performed by an adversary; such an approach is known as a ''Reactive Approach''. While this approach has been widely adopted, with some success, it is not capable of foreseeing threats. Cyber criminals are well aware of reactive approaches and know how to deal with them. For example, ''polymorphous malware'' are very good at evading anti-viruses. To combat the limitations of reactive methods, new techniques such as threat hunting have been

The associate editor coordinating the review of this manuscript and approving it for publication was Ilsun You.

developed. Threat hunting involves proactive searching for cyber threats that may be lying undetected in a network. Threat hunting is used to uncover new techniques, tactics, and procedures (TTP's) to forecast new threats. Threat hunting uses information from a variety of sources such as endpoints, Indicators of Compromise (Iosco), Firewalls and intrusion detection and prevention systems (IDPS). SANS presented a formal threat hunting model in 2019 [1], which opened new doors for researchers.

Many organizations conduct offensive security exercises such as penetration testing and adversary simulation. Penetration testing determines the presence of any critical vulnerability that needs to be addressed. Such testing aims to test how well security mechanisms are working. Some organizations have dedicated Red and Blue Teams to test and evaluate the organizational security. One problem with this approach is that red team operations have been criticised for being resource intensive. In an evolving threat environment, where attackers are motivated to employ sophisticated and lingering attacks, organizations are more prone to cyber attacks. Approaches such as vulnerability scanning, management and mitigation, and Vulnerability Assessment and Penetration Testing (VAPT) rely on known threats [2], [3] [4], [5]. Such mechanisms have limited efficacy on Advanced Persistent Threat (APT) attacks, which are designed to remain stealthy for a long period before triggering a zero day attack.

A Red Team involves attack simulation which allows organizations to measure how strong security controls are against potential cyber attacks, and the resilience of systems. Most of the research available in securing systems is focused on defensive approaches that prevent the occurrence of any possible vulnerability being exploited. Many historic and recent cyber attacks have demonstrated the need to employ proactive approaches. The purpose of proactive approaches is to learn and understand TTP's to avoid *future* attacks.

Organizations are moving towards adopting proactive approaches to predict threats. Proactive methods such as threat hunting have proven effectiveness in detecting threats, although, such approach is resource intensive as it requires intense monitoring. Event logs generated at endpoints and Iosco can grow enormously over time. Such logs require significant processing and analysis which increases the usage of resources. This can also increases the false positives alerts of supposed malicious activity that ultimately prove to be non-malicious.

In this paper, we propose a threat hunting model via adversary emulation, with the aim of minimizing the resource utilization while increasing the efficiency of the approach. This can allow organizations to perform two different related tasks simultaneously. To validate our model, we have built a simulated environment and launched a real world APT attack scenario on patched systems. We have demonstrated that using an induced form of threat hunting model with adversary, an emulation is effective in hunting emerging threats.

The rest of the paper is organized as follows: the next section presents a review of the literature on threat hunting and VAPT. In Section III, we propose the formal model for threat hunting via adversary emulation. In Section IV, we describe the implementation of a the proposed model. In Section V, details are provided about the experiments that have been conducted. Furthermore, an evaluation of the proposed system using the penetration testing scoring model by Packt [6] is presented. The paper is concluded in Section VI.

II. RELATED WORK

The threat hunting model is presented in [1], Our approach uses an induced form of this model with an adversary emulation model. This research provides an efficient and repeatable method for evaluating computer and network security using threat hunting through offensive security. This approach defines offensive security as process of understanding the adversary and then building plans for launching attacks. The overall coverage and integrity of the whole process is measured.

In the proposed approach, methodologies for generating hypotheses and their validation are derived from [7]. The authors describe the two key components involved in generating hunting hypotheses. First, an analyst's ability to create hypotheses is derived from observations. Second, the hypotheses must be testable. We have used first component of the hypothesis development methodology in our model.

Table 1 presents the summary of VAPT related research work which is evaluated on the basis of three factors [6]: *realism* exhibits if emulation was close to real world attack scenario, *methodology* describes which tools are used, *limited scope* explains the techniques where only limited attack scenarios are considered. We have developed the adversary emulation process using a human-led penetration testing approach inspired by the PoinTER ''human firewall'' penetration-testing framework [8].

VAPT assists organizations in the evaluation of their cyber defense strategy. An overview of the different techniques used in VAPT is provided in [2]. The proposed approach includes a mechanism for capturing unknown threats as well as the known threats. Without consideration of unknown, future attacks, VAPT methodologies remain susceptible to APT and zero day attacks. Earlier research, for example [28] and [27] considered vulnerability and patch management as a solution for securing organizations. Recent research has been published, such as the penetration testing framework for mobile devices [29], in which authors consider testing of common security controls, but these are also limited in their consideration of, and efficacy against, unknown threats and social engineering attacks.

Adversary emulation exercises can provide cyber defenders with an opportunity to view their networks from an attacker's perspective. Recent research has discussed formalisation of the problem and has provided techniques for adversary emulation [30]. This work is built upon the ''Atomic Red Team'' mechanism [31] to create test cases for MITRE ATT&CK tactics and techniques. We have further developed

TABLE 1. Related work table.

Research Subject	Year	Limitations	Attack vector
Module Development for VAPT [9]	2020	Lack of realism	Static
VAPT Audit and implementation [10]	2020	Lack of methodology	Static
VAPT literature review[11]	2020	Limited scope	Static
Cyber attack emulation agents[12]	2020	Limited scope	Static
Learning associations of adversary techniques[13]	2020	Lack of methodology	Static
CTI for improving adversary understanding[14]	2020	Lack of methodology	Static
Zero entry hacking and mis-configurations on VAPT[15]	2019	Lack of realism	Static
VAPT as Cyber defense[16]	2019	Limited scope	Known
VAPT for web app and network[17]	2019	Limited scope	Known
Post exploitation Agentless VAPT and Automation[18]	2019	Limited scope	known
Bird eye view penetration testing knowledge[19]	2019	Lack of methodology	Known
Concept of cyber defense exercise CDX[20]	2018	Lack of methodology	Known
Enhancing automotive penetration testing[21]	2018	Limited scope	Known
Redefining VAPT[17]	2018	No methodology	Known
Cyber kill chain and cyber ops[22]	2018	No methodology	Known
Modern tendency, benefits and drawbacks of VAPT[23]	2017	No methodology	Known
Implementing VAPT tools and design[24]	2017	Lack of realism	Known
Concepts, attack methods, and defense strategies[25]	2016	Lack of methodology	Known
Improved input vectors for pen testing[26]	2016	Limited scope	Known
Analysis of VAPT[2]	2016	Lack of methodology	Known
Cyber self defense using VAPT[27]	2016	Limited scope	Static

TABLE 2. Open source adversary emulation tools.

Utility	Target Platform	Mitre ATT&CK	Emulate new-attack	Controls testing
NetworkFL Simulator[32]	Any	No	No	Network based
Atomic Red Team[31]	Any	Yes	No	Endpoint
AutoTTP[33]	Any	Yes	Partially	Endpoint
Dumpsterfire[34]	Any	No	No	Endpoint
InvokeAdversary[35]	Windows	No	No	Endpoint
InfectionMonkey[36]	Any	Yes	No	System Level
Caldera [37]	Any	Yes	No	System Level

TABLE 3. Used offensive security payloads.

Type	Description
OLE files	Embedded VBA scripts in doc/docx
PDF	Document with hidden payload at EOF
DLL	Malicious dll's
PE	Most common .exe file
Bat file	Malicious .bat file
PS1	PowerShell malicious script
Stegno Payload	Hidden payloadc

this concept to provide a more agile system that considers a diverse and increased set of TTPs. For example, to develop APT29 attack cases, careful consideration needs to be given to the sequence of attack cases.

The majority of recent research models for VAPT rely on "known threats", see for example [2]–[4] and [5]. Furthermore, a number of open source tools for adversary emulation have been developed recently; these are categorised in Table 2. many of these projects build attack cases mimicking an adversary, which has demonstrated efficacy in testing security controls for known adversary and threats.

III. NOVEL THREAT HUNTING & OFFENSIVE SECURITY APPROACH

In the proposed research, we integrate a proactive approach for hunting threats within an adversary emulation process and model threats on the basis of techniques discussed in [38]. We consider threat severity, progression and relevance for threat modeling as defacto standard [39]. Our offensive security approach specifically uses payloads described in Table 3. Similar attack vectors have been used in recent adversary emulation projects including those presented in Table 2. We have used PE files, OLE files and PS1 for experiments. The payloads described in Table 3, can be generated through the use of Algorithm 1.

The proposed offensive security model consists of eight steps which are in sequence. These stages are: purpose, scope, equip, planning, weaponizing, plan review and validation (weaponization), execute and reporting. This is achieved by mapping the adversary emulation model onto the threat hunting model. Figure 1 explains the induced model.

The proposed approach can be defined as a tuple < AE, P, S, F, E, WP, PR, EX,, HT, RP>, where AE is a set of pre-requisites, P is a purpose set, S is a scope set, F is feedback set, E is an equipping set, WP is a weaponizing process set, ER is a review set, EX is an execution information set, HT is a set of hunted threats and RP is a reporting set.

Purpose set P in Equation 1 represents the set of information about purpose of threat hunting which might be oriented to organizational goals. Executives may guide threat hunters about organizational goals and objectives. AE represents the set of prerequisites.

$$P = f_1(PxAE) \tag{1}$$

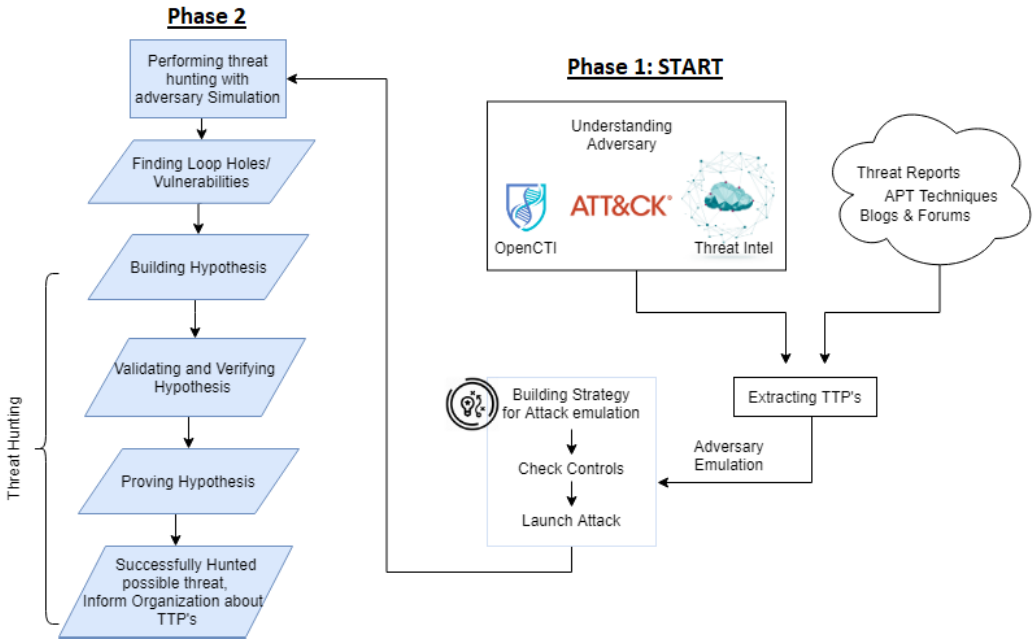

FIGURE 1. Hunt model and adversary simulation approach.

Scope set S represents all information regarding scope of our whole operation. Here, we build analytical question "Hypothesis". TTP set represents adversarial techniques, that can be obtained by threat intelligence. RVS is reconnaissance and vulnerability identification set in Equation 2.

$$S = f_2(S \times TTP \times RVS) \tag{2}$$

Equip set E in Equation 3, represents information about answering previously build questions. Set E contains two subsets E = (AC, VDS) analytical questions and verifying them. It includes organizing data which is collected from this stage. VDS is a set of information representing possible weakness or flaws in system. AC is a set of attack mappings related to expected system flaws.

$$E = f_3(E \times VDS \times AC) \tag{3}$$

Weaponization set WP in Equation 4, represents the set of exploits or actual implementation of attacks identified at previous stage. MSF set represents information about ported exploits in metasploit.

$$W = f_4(W \times MSF) \tag{4}$$

Before moving on towards emulation, we need to validate if our planned activity is according to desired plan. For this purpose, we review our plan. Plan review set PR in Equation 5 represents the information about our whole plan up-to now and compares it with our, already set objectives and Scope.

$$PR = f_5(PR \times S \times P) \tag{5}$$

Execute set EX in Equation 6 represents information about execute plan process. Which includes information about target systems and network, with ported exploits. This is the actual process where all adversary emulation takes places. Evasion and exploitation is heart of this process. EV is a subset of EX set, which includes information about newly developed evasion techniques.

$$EX = f_6(EX \times EV \times MSF) \tag{6}$$

Hunted Threats set HT in Equation 7 represents the threats that are successfully hunted. For example, while adversary emulation operation suspected a flaw in system, which may lead towards zero day attack and moreover we have successfully exploited it, so hunted threats will be moved to HT set.

$$HT = f_7(HT \times EX) \tag{7}$$

Reporting set RP in Equation 8 represents whole process carried out with analysis of each phase, feed backs from each phase from set F is used to generate report.

$$RP = f_8(RP \times F) \tag{8}$$

- **Purpose:** At this stage we define the purpose of offensive security and threat hunting exercise, which might be related to organizational goals. Top management and executes may guide offensive security team about purpose and objectives.
- **Scope:** In scope, we identify network and systems that to be a part of offensive security exercise. We define different functions at this stage that whole purpose will include or can be extended during process. Scope is further divided into two steps, first one is to define

hypothesis (a set of analytical questions) second one is developing hypothesis formally. Moreover, it will define direction to the whole process.

- **Equip:** This stage is about collecting data from different sources for analysis, this includes identification of various data sources and analytical TTP's. Threat hunters will use this analysis of data to answer analytical questions build during first stage.
  At this stage we also prove or disprove early developed hypothesis. Moreover, we build mappings of data against attacker targets and there data sources in later section (experiment) we have added example for CMF (collection management framework).
- **Weaponizing process:** At this stage we develop different type of attack vectors which can be used during adversary simulation process. These attack vectors can be build for endpoints only, that depends on scope. One of the contribution of this research is the presentation of algorithm for pre-compromise attack algorithm through mail, which is presented in Algorithm 1.

Algorithm 1 Sending and Generating Phishing Mail

Input: SE (Set of target email addresses), PT (Payload or site link), PM (payload method)
Output: Phishing mail template and mail sent to target addresses

1: **Initialize:** n = number of emails, PM = payload method, PT = payload type, num_mail = number of valid mails
2: **for** i → n
3: email_validate (SE)
4: result ← target_mails × num mail
5: **if** PM = payload **then**
6: **Inject:** payload into word docx
7: result ← payload_result
8: **else**
9: **if** PM = link **then**
9: result ← link result
10: **call** template generation eng () ← payload_result
11: **result** ← template
12: **call** mail generate () ← template × result
13: **call** mail spoof function
14: **Inject** desired header
15: result ← mail
16: **for** i → num mail
17: **call** send mail function () ← mail *x* target_mail
18: **end**

There are three main input variables; set of emails; payload or link and payload method. At very beginning algorithm validates the inputted emails by passing them one by one to `email_validate()`. Then algorithm check for payload type, if it is malicious document then it will inject malicious code inside pdf document or macro in docx. If payload type is link, then it will add website link in email. And return website in html with embedded file or link with receiver details. Result will be store in payload_result. This result will be passed to `template_generation()` function and it will return proper email template. This result will pass on to `mail_spoof()` which replace sender details and pass result to `send_mail()`.

TABLE 4. Semantic details used in Algorithm 1.

Functions	Description
email_validate()	This function inputs set of targets mails and verifies if those mails exist using MX records. Mails with valid response are considered as valid. After this process, "Valid emails" and number of mails are returned.
Inject:payload	This method runs after valid input for payload method. It might be a malicious "link" or OLE files with hidden payload, such as docx with macros or pdf file with hidden payload (using stenography). Payload is a macro "VB code" that will open bind tcp connection or use CMD to download further payload.
temp_generate _engine()	This function generates mail templates. A template might be notification from CEO or discount offers. After generating a template, it will return it.
mail_gen()	This function takes a template, valid mails and payload as input and generate a mail.
Inject:header	This is a method of the mail_generation() fucntion, which modifies mail header to bypass spam filter.
mail_spoof()	This is a subroutine of inject:header, which replaces sender details with some other mail to make it look legitimate.
mail_send()	This function inputs the mail_spoof() output, and send mail to target addresses.

- **Plan Review** At this stage, it acts like a checkpoint to make sure our plan is according to the defined goals and objectives.
- **Execute:** This process is iterative, once plan is approved, we execute plan and simultaneously launch different planned attacks and collect data for analysis. This process keeps on going for several iterations until threat hunters get enough data for the analysis.

A simplified flow chart which explains operational flow of approach is presented in Figure 2. The adversary simulation element comprises Weaponization, Shell Code development and Obfuscation, creation of the FUD (fully undetectable payload) and establishing the toolsets.

A. ADVERSARY EMULATION MODEL

We emulate adversaries through a process comprising six major sequential steps. These are described Figure 3.

1) OBJECTIVES & GOALS

Here, we formally define the purpose of adversary emulation. This process is usually aligned with organizational goals. Once the purpose is defined, objectives and goals are established.

2) GATHER THREAT INTELLIGENCE

Gathering threat intelligence is a critical task for effective threat-based adversary emulation. There are many feeds available for threat intelligence, including those from the

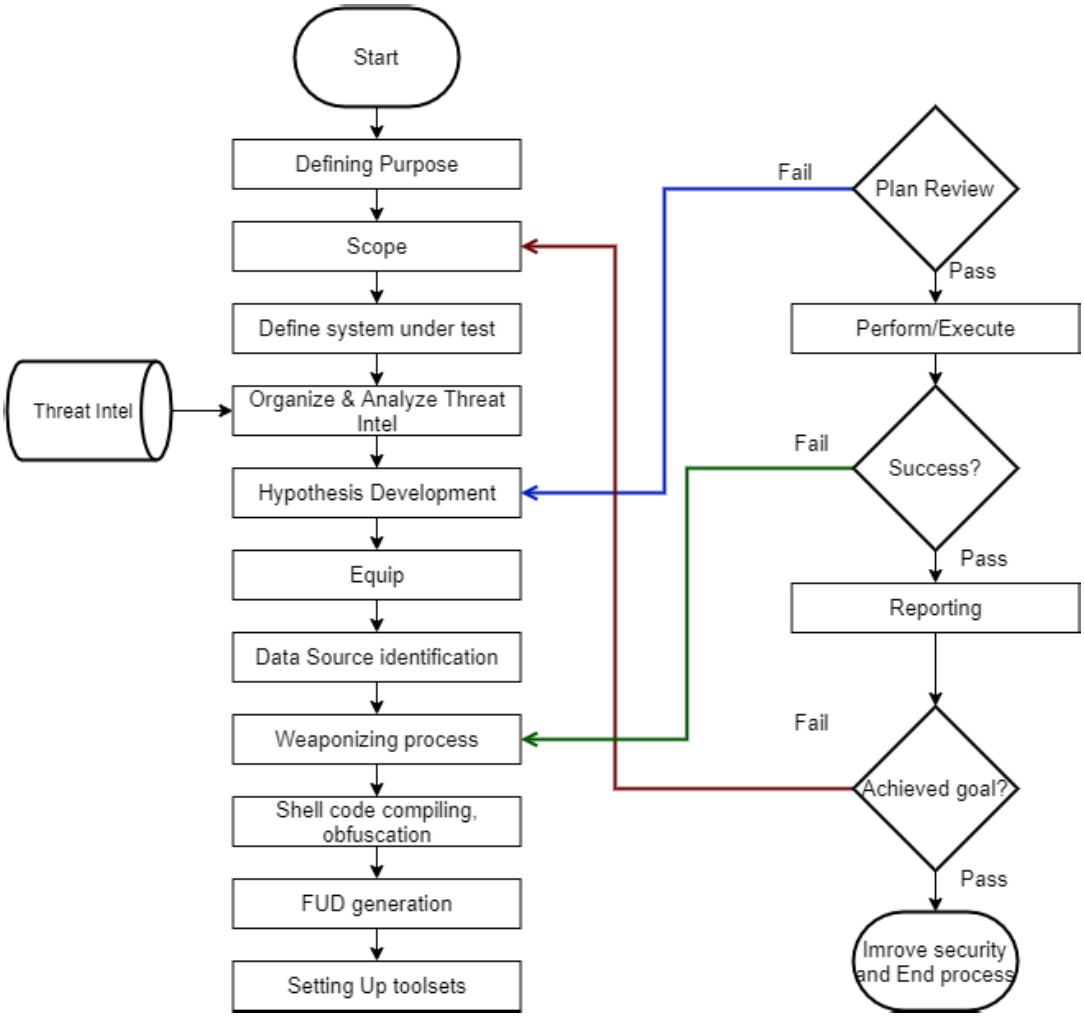

FIGURE 2. Approach flowchart.

DHS (Department for Homeland Security), FBI, SANS, and commercial and free versions of the CISCO Talos system. Further intelligence sources include threat research forums and blogs. The OpenCTI project, developed by ANSSI along with the CERT-EU, provides a system to structure, store, organize, visualize and share cyber threat intelligence.

3) EXTRACTION TECHNIQUES

According to professional bodies and industry, such as MITRE, Cisco & SANS, identifying TTPs is the toughest task in the so-called ''Pyramid of Pain'' of security. Establishing TTPs requires a structured process to ensure the effectiveness, completeness and accuracy of information. Phase 1 starts with the categorise each techniques at a tactical level. For example: malware used dll unhooking technique to evade anti-virus. If we map this technique onto MITRE ATT&CK framework, it would be categorised as defense evasion. The second phase starts involves defining the flow of methods related to a specific adversary [40]. For example, an adversary might use different techniques for stealing hashes and then used these hashes for password spraying to get access of system.

4) ANALYZE & ORGANIZE

At this stage, the understanding of adversary goals is elaborated, and mapping the method flow into an adversary plan. Figure 4, for example, shows the plan for APT28 from MITRE the APT3 plan.

5) DEVELOP TOOLS AND PROCEDURES

This stage involves the development of any new tools, or reconfiguration of existing tools, to launch the malware or attack systems. If the adversarial techniques can be conducted using existing tools, then this is most appropriate because building a new tool is a costly process. Due to the cost of developing or purchasing new tools, there is often a preference to utilise the range of open source tools available. The process involves

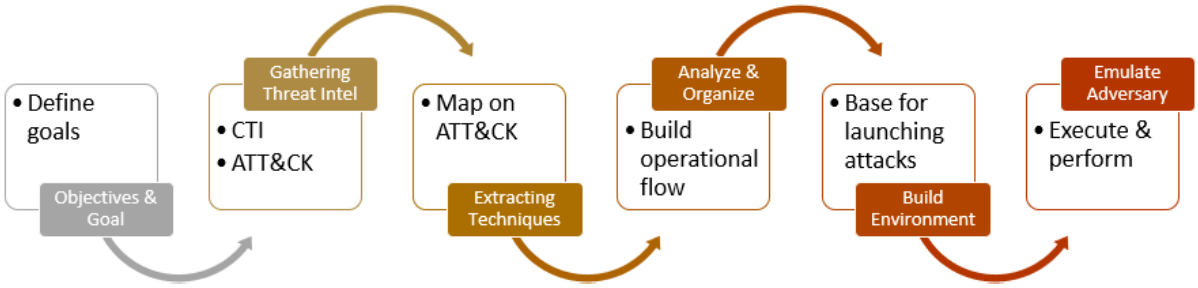

FIGURE 3. Offensive security model.

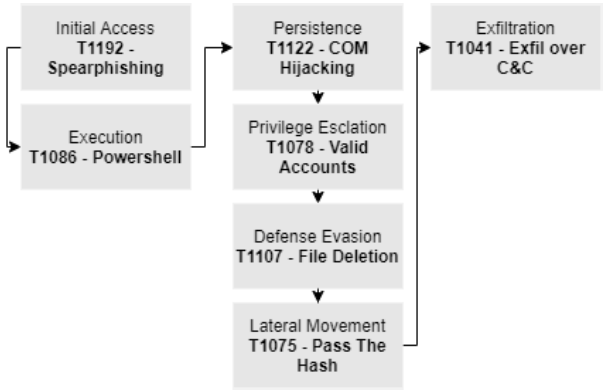

FIGURE 4. Emulation plan.

- Identifying related open source projects
- Identifying process-specific requirements
- Creating the payload

The most common existing open source tool used is Metasploit. As an example, if we are considering a dll injection being used by an adversary to evade EDR, there are a number of techniques that can be employed. Firstly, we look for similar dll injection in Metasploit. If it meets the requirements then this can be used, otherwise it is modified or rewritten.

6) EMULATE THE ADVERSARY

The killchain used in our scenario has reduced to 6 phases only, As first phase is already done in earlier phases. At this stage, we know our target system according to the defined scope we are done with planning phase. We are ready to start executing plan. Execution includes weaponization, deliver, exploit, control, install and maintain.

Threat hunting process is cyclic in nature, it consists of four processes. First one is about creation of a hypothesis, second is about verifying and validating hypothesis. This process includes investigation for any proof with the help of tools and techniques. Next process explains new TTP's and patterns. Final process includes enrichment. Informing incident response team about the new TTP's. Below is the big picture (Figure 5) of whole induced model which we were explaining is this section.

TABLE 5. Hardware and software details.

Platform	OS	Clock Speed	RAM
Virtual Box	Windows 10	2.8Ghz	4GB
Virtual Box	Windows server 2012	2.8Ghz	8GB
Virtual Box	Debian	2.8Ghz	4GB

IV. EXPERIMENT

We have devise an experiment to demonstrate the efficacy of employing the two proactive approaches. The experiments can be divided into two phases: the first phase starts with launching offensive security exercise to compromise target; the second phase involves a counter offensive exercise which aims to capture expected threats emerging from offensive exercise. Our experiment are lab-based but aim to closely mimic real world scenarios. Security mechanisms are installed at the target including the presence of host based and network level firewalls. An intrusion detection system, such as Snort, is installed at the endpoint. Hardware and software used in the experiments are detailed in Table 5.

In the experiment, we consider **T1090.004**, also known as "Domain Fronting". One mitigation technique for domain fronting is SSL/TLS Inspection, though, it is not widely deployed nor applicable in some scenarios; its effectiveness in the mitigation of this attack is limited. Figure 6 explains our lab and target environment and Figure 7 explains our strategy for domain fronting. Techniques used to evade detection at the endpoint include using a "modular design" in the payload. The execution of payload is critical process in the adversary emulation. Now, we employ T1055.012, T1055.08, T1055.04, T1055.09 and T1055.014. Some modifications are required, for example, modifying T1055.012. We modified the process using a hollowing technique with hybrid graded launch method to avoid detection. At the first trusted binary call, the payload will list itself in the PEB (Process Environment Block) and suspends itself. Once the trusted binary is executed it replaces itself with executed

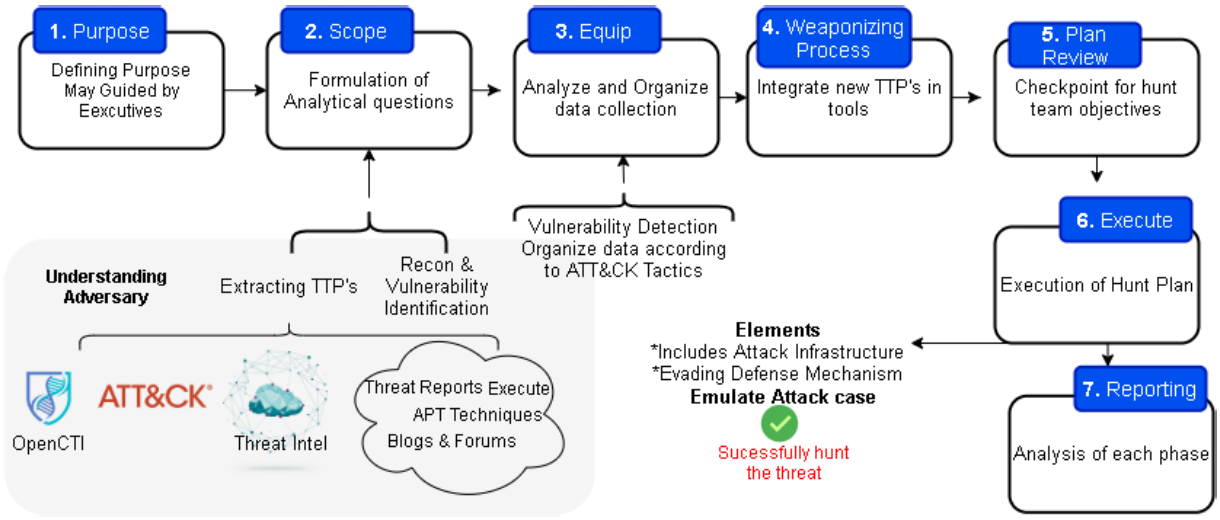

FIGURE 5. Threat hunt model simplified.

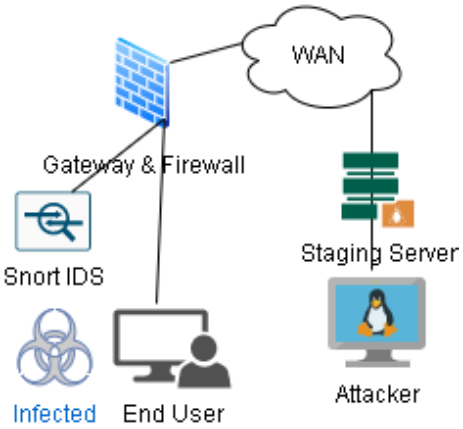

FIGURE 6. Lab and target environment.

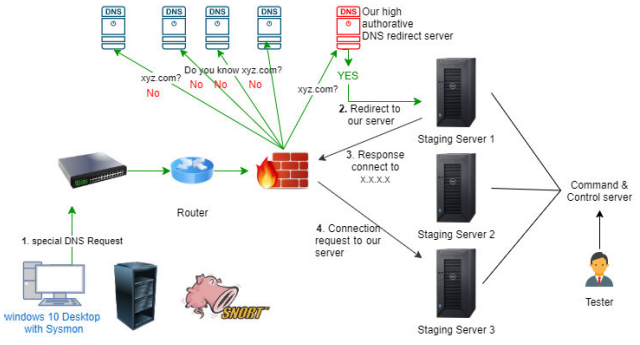

FIGURE 7. Attack strategy.

process by taking advantage of the data already stored in the PEB. Proxy techniques for C2 **T1090** with sub techniques as T1090.001, T1090.002, T1090.003, T1090.004 are some of the most reliable techniques to identify C2. Figure 7, depicts our attack strategy.

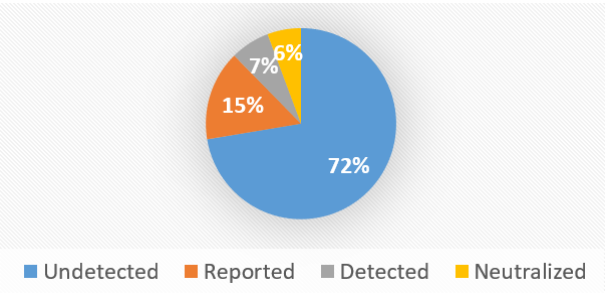

FIGURE 8. Offensive security exercise outcomes.

V. RESULTS

After successfully conducting the threat hunting and offensive security exercise it can be seen that the offensive activity was able to evade security solutions using unknown attacks. At the same time, the threat hunting team started its counter offensive activity using real time logs from the endpoint and firewalls. The attack vector (Hash: `37f56970252e51258b8583b996501d50669bf996e472bfc35a1294f09accf19e`) was fully undetected by 79 antivirus engines on *virustotal.com*. Some of the techniques detected by the hunt team are shown in Table 6.

Figure 8 shows the overall experimental results detailing the attacks that were successful, reported, neutralized and undetected. Of the total attacks, 72% of attacks were able to make it through the endpoint, 15% were reported at endpoint, 7% of attacks were detected in the initial phases and 6% of attacks were completely neutralised. The techniques that were detected, and their mapping to ATT&Ck is shown in Table 6.

A. IMPACT ANALYSIS (ATTACK VECTORS)

We have analysed different attack tactics and corresponding techniques with the methods we used in our experiments.

TABLE 6. Detected techniques mapped on ATT&CK.

ID	ATT&CK Description
T1055.002	PEInjection
T1055.003	Thread Execution Hijacking
T1055.004	AsynchronousCall
T1055.005	ThreadStorage
T1055.008	SystemCalls
T1055.009	ProcMemory
T1055.011	MemoryInjection
T1055.012	ProcessHollowing
T1562.001	Modify Tools
T1562.002	ModifyEventLogging
T1562.003	Modify System Firewall

The first attack tactic is initial access. More specifically, we utilised spear phishing.

1) PHISHING

Analyzing mail headers yields the use of blind mail with SMTP server IP address and location. The mail was encrypted with TLS during transit and the server we used was a VM on MS Azure. This email landed in the inbox. We then attached a payload (exe file) with custom UTF encoding during transit. This time mail landed in the spam folder as it was detected as suspicious due to the presence of suspicious file type.

2) ATTACK VECTOR(PAYLOAD) ANALYSIS

We tested meterpreter shell code in plain text (with slight modification) on Windows 10 2004 and virustotal.com. The shell code was able to bypass static analysis but was caught with heuristics. Moreover we tested payload with greater complexities, including heavy obfuscation, strings and variable name encryption with AES. Such payloads were completely undetectable to any AV. Analysis of the payload, using reverse engineering, is shown in Table 7.

TABLE 7. Payload analysis.

ATT&CK Tactic	ATT&CK Technique
Defense Evasion	T1027
Discovery	T1012
Execution	T1059.003, T1129
Ex-filtration	T1041
Persistence	T1543.003

3) EXECUTION

Threat hunters can detect the presence of suspicious processes through analyzing the behaviour of the system. In the experiment, we analyzed our payload execution with T1553 technique. The following are some known indicators of the hypothesis.

1) The name of PE file
2) Access rights being used to access specific process

Understanding the specific techniques implemented in known methods plays important role in detection. For example, knowledge of the use of strings inside PE files and known hashes can aid detection. Such characteristics are simple to modify with minimal effort. To evade detection we used renaming of the file with a trusted binary name such as msmpeng.exe (which is the windows defender binary).

TABLE 8. Discovered techniques.

Technique	Procedure	Sub-technique
PowerShell AMSI	Execution, download, load files reflectively	String encoding, string obfuscation, load string & script from trusted source
Reverse Shell	Initial access	PowerCat

TABLE 9. Initial access techniques.

ID	Name	Launching Method
1	Using hyper link	Mail using custom SMTP
2	Hyper link with Obfuscation	Through mail, link shorteners
3	Using scripting language file	Sending file inside mail
4	Using PE file	Sending file inside mail

TABLE 10. CMF for threat hunters.

Collection Management Framework (CMF)	Source	Source Detail
Location	Endpoint	Syetem T1
Kill chain step	Detection evasion	Process Injection
Data type	Raw log	Sysmon log
Collection method	Telemetry log sharing	Log beat
Storage duration	60 minutes	Completed 4 phase cycle

Phase 2, at this stage, threat hunters have ample data from different sources to analyse. They can approve or disapprove each hypothesis or validate it and move on to the enrichment/reporting phase. In our case we can assume that the hypothesis is validated since there are sufficient validation proofs from Table 7, 8 and 9. Findings from Tables 8 and 9 are sufficient to enable reporting. The reporting phase ends with threat and risk assessment. For this purpose, we have used a combined threat and risk assessment to show the effectiveness of merging two approaches. Table 8 describes the low level techniques that can be employed for initial access. Each of these sub-techniques are related to spear phishing. After successfully conducting threat hunting, hunters utilise a collection management framework to manage the data collected to be used in validation. Threat hunters consider the different dimensions of threats that are likely to happen or already exist.

After conducting the offensive security and threat hunting exercises, we can present a summary of the analysis in Table 11.

This table describes security posture of the target environment in the form of a threat and risk assessment. The scoring scale we use is: Low < 4; $4 <$ Medium < 7; and High > 7. Each threat value is calculated by summing

TABLE 11. Threat and risk assessment.

Threat ID	Description of Threat	Total Risk	Risk Level (H/M/L)	Damage Potential	Reproducibility	Exploitability	Affected People	Discoverability/ Detectability
T-2	Attack on AD	2.8	L	4	1	4	4	1
T-3	Attack on firewalls	3.4	L	4	4	4	1	4
T-4	Attack on server	1.6	L	1	4	1	1	1
T-5	Miss-Configurations	7	H	7	8	8	4	4
T-6	Late patch management	4.6	M	10	4	1	4	4

the damage potential, reproducibility, exploitability, affected people and detectability and the dividing this total by 5.

VI. CONCLUSION

This paper has presented a novel hybrid model for launching offensive security exercises to capture, determine and understand attack patterns by foreseen threats using threat hunting. The proposed approach has increased the efficiency of identifying and countering threats using real world attack scenarios and presents an algorithm to generate attack vectors for phishing. In contrast to traditional methods that focus on known threats, such as VAPT. The proposed scheme is designed to identify and address emerging unknown threats. In the future, we plan to focus on increasing the realism of the emulation of adversaries with advanced stealthy attacks.

REFERENCES

[1] D. Gunter and M. Seitz, "A practical model for conducting cyber threat hunting," SANS White Paper 38710, Mar. 2019. [Online]. Available: https://www.sans.org/white-papers/38710/

[2] P. S. Shinde and S. B. Ardhapurkar, "Cyber security analysis using vulnerability assessment and penetration testing," in *Proc. World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave)*, Feb. 2016, pp. 1–5.

[3] P. Dholey and A. K. Shaw, "OnlineKALI: Online vulnerability scanner," in *Proc. Int. Ethical Hacking Conf., Adv. Intell. Syst. Comput.*, vol. 811. Singapore: Springer, 2019, pp. 25–35, doi: 10.1007/978-981-13-1544-2_3.

[4] P. Russo, A. Caponi, M. Leuti, and G. Bianchi, "A web platform for integrated vulnerability assessment and cyber risk management," *Information*, vol. 10, no. 7, p. 242, Jul. 2019, doi: 10.3390/info10070242.

[5] R. Stevens, D. Votipka, E. M. Redmiles, C. Ahern, and M. L. Mazurek, "Applied digital threat modeling: It works," *IEEE Secur. Privacy*, vol. 17, no. 4, pp. 35–42, Jul. 2019.

[6] *Pentesting Scoring Model*. Accessed: Sep. 12, 2021. [Online]. Available: https://subscription.packtpub.com/book/networking_and_servers/9781788624480/1/ch01lvl1sec17/pentesting-maturity-and-scoring-model

[7] R. Lee and D. Bianco. (2016). *Generating Hypotheses for Successful Threat Hunting, Retrieved From SANS Reading Room*. [Online]. Available: https://www.sans.org/readingroom/whitepapers/threathunting/generatinghypotheses-successful-threat-hunting-37172

[8] J. M. Archibald and K. Renaud, "Refining the PoinTER 'human firewall' pentesting framework," *Inf. Comput. Secur.*, vol. 26, no. 4, pp. 575–600, 2019.

[9] N. T. Pages, "Module development in metasploit for pentesting," M.S. thesis, Univ. Politècnica de Catalunya in Faculty of Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona, 2019.

[10] S. Barad and P. Sharma, "A study of security audit and VAPT audit and implementation of cyber security controls like WSUS against cyber threats," *J. Eng. Sci. India*, vol. 11, no. 4, Apr. 2020. [Online]. Available: https://jespublication.com/upload/2020-1104141.pdf

[11] P. Vats, M. Mandot, and A. Gosain, "A comprehensive literature review of penetration testing & its applications," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, Jun. 2020, pp. 674–680.

[12] J. D. Yoo, E. Park, G. Lee, M. K. Ahn, D. Kim, S. Seo, and H. K. Kim, "Cyber attack and defense emulation agents," *Appl. Sci.*, vol. 10, no. 6, p. 2140, Mar. 2020.

[13] R. Al-Shaer, J. M. Spring, and E. Christou, "Learning the associations of MITRE ATT&CK adversarial techniques," 2020, *arXiv:2005.01654*. [Online]. Available: http://arxiv.org/abs/2005.01654

[14] M. Parmar and A. Domingo, "On the use of cyber threat intelligence (CTI) in support of developing the Commander's understanding of the adversary," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 1–6.

[15] R. Sahtyawan, "Penerapan zero entry hacking didalam security misconfiguration pada VAPT (vulnerability assessment and penetration testing)," *J. Inf. Syst. Manage.*, vol. 1, no. 1, pp. 18–22, Jul. 2019. [Online]. Available: https://jurnal.amikom.ac.id/index.php/joism/article/view/18

[16] B. A. Chandrakant and J. P. Prakash, "Vulnerability assessment and penetration testing as cyber defence," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 2, 2019. [Online]. Available: https://www.ijeast.com/papers/72-76,Tesma402,IJEAST.pdf

[17] M. Shahidullah, "Vulnerability assessment penetration testing (VAPT) for web applications," EasyChair, 2019. [Online]. Available: https://login.easychair.org/publications/preprint/3pSg

[18] S. Maddala and S. Patil, "Agentless automation model for post exploitation penetration testing," in *Proc. Int. Conf. Intell. Comput., Inf. Control Syst., Adv. Intell. Syst. Comput.*, vol. 1039. Cham, Switzerland: Springer, 2019, pp. 529–539.

[19] A. Rahman and L. Williams, "A bird's eye view of knowledge needs related to penetration testing," in *Proc. 6th Annu. Symp. Hot Topics Sci. Secur.*, 2019, pp. 1–2.

[20] E. Seker and H. H. Ozbenli, "The concept of cyber defence exercises (CDX): Planning, execution, evaluation," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Jun. 2018, pp. 1–9.

[21] J. Dürrwang, J. Braun, M. Rumez, R. Kriesten, and A. Pretschner, "Enhancement of automotive penetration testing with threat analyses results," *SAE Int. J. Transp. Cybersecur. Privacy*, vol. 1, no. 2, pp. 91–112, Nov. 2018.

[22] S. Cho, I. Han, H. Jeong, J. Kim, S. Koo, H. Oh, and M. Park, "Cyber kill chain based threat taxonomy and its application on cyber common operational picture," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (Cyber SA)*, Jun. 2018, pp. 1–8.

[23] Y. Stefinko, A. Piskozub, and R. Banakh, "Manual and automated penetration testing. Benefits and drawbacks. Modern tendency," in *Proc. 13th Int. Conf. Modern Problems Radio Eng., Telecommun. Comput. Sci. (TCSET)*, Feb. 2016, pp. 488–491.

[24] P. S. Shinde, S. B. Ardhapurkar, and P. Scholar, "Design and implementation of vapt tool for cyber security analysis using response analysis," *Int. J. Eng. Sci. Comput.*, vol. 6, no. 4, pp. 4150–4153, Apr. 2016.

[25] M. Denis, C. Zena, and T. Hayajneh, "Penetration testing: Concepts, attack methods, and defense strategies," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, Apr. 2016, pp. 1–6.

[26] W. G. J. Halfond, S. R. Choudhary, and A. Orso, "Penetration testing with improved input vector identification," in *Proc. Int. Conf. Softw. Test. Verification Validation*, Apr. 2009, pp. 346–355.

[27] S. Shah and B. Mehtre, "A reliable strategy for proactive self-defence in cyber space using VAPT tools and techniques," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, 2013, pp. 1–6.
[28] S. Shah and B. M. Mehtre, "A modern approach to cyber security analysis using vulnerability assessment and penetration testing," *Int. J. Electron. Commun. Comput. Eng.*, vol. 4, no. 6, pp. 47–52, 2013.
[29] I. L. Aller, J. M. R. Lopez, and L. A. V. Martinez, "Towards lightweight mobile pentesting tools to quickly assess machine security levels," *IEEE Latin Amer. Trans.*, vol. 17, no. 7, pp. 1116–1123, Jul. 2019.
[30] D. Miller, R. Alford, A. Applebaum, H. Foster, C. Little, and B. Strom, "Automated adversary emulation: A case for planning and acting with unknowns," MITRE, McLean, VA, USA, Tech. Rep. AD1108001, 2018. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1108001
[31] C. Smith, "Red canary introduces atomic red team, a new testing framework for defenders," Red Canary, Denver, CO, USA, 2017. Accessed: Oct. 5, 2020. [Online]. Available: https://redcanary.com/blog/atomic-redteam-testing/
[32] AlphaSOC. *AlphaSOC/Flightsim*. Accessed: Dec. 4, 2020. [Online]. Available: https://github.com/alphasoc/flightsim
[33] Jymcheong. *Jymcheong/Autottp*. [Online]. Available: https://github.com/jymcheong/AutoTTP
[34] TryCatchHCF. *Trycatchhcf/Dumpsterfire*. Accessed: Nov. 10, 2020. [Online]. Available: https://github.com/TryCatchHCF/DumpsterFire
[35] CyberMonitor. *Cybermonitor/Invoke-Adversary*. Accessed: Nov. 15, 2020. [Online]. Available: https://github.com/CyberMonitor/Invoke-Adversary
[36] Guardicore. *Guardicore/Monkey*. Accessed: Oct. 5, 2020. [Online]. Available: https://github.com/guardicore/monkey
[37] W. Mitre. *Mitre/Caldera*. Accessed: Feb. 1, 2021. [Online]. Available: https://github.com/mitre/caldera
[38] W. Xiong and R. Lagerström, "Threat modeling—A systematic literature review," *Comput. Secur.*, vol. 84, pp. 53–69, Jul. 2019.
[39] W. V. Beal. (2019). *Cyber Kill Chain*. [Online]. Available: https://www.webopedia.com/TERM/C/cyber-kill-chain.html
[40] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre ATT&CK: Design and philosophy," McLean, VA, USA, MITRE Tech. Rep. 19-01075-28, 2018.

ABDUL BASIT AJMAL is currently pursuing the master's degree in information security with COMSATS University Islamabad, Islamabad, Pakistan. Since 2019, he has been working as a Postgraduate Researcher at the Cyber Security Lab COMSATS under HEC Technology Development Initiative. His research interests include offensive security, adversary simulation, threat hunting, phishing, and designing secure systems.

MUNAM ALI SHAH received the B.Sc. and M.Sc. degrees in computer science from the University of Peshawar, Pakistan, in 2001 and 2003, respectively, the M.S. degree in security technologies and applications from the University of Surrey, U.K., in 2010, and the Ph.D. degree from the University of Bedfordshire, U.K., in 2013. Since July 2004, he has been an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Pakistan. He is the author of more than 200 research articles published in international conferences and journals. His research interests include information security design, QoS, and energy harvesting in communication systems. He received the Best Paper Award of the International Conference on Automation and Computing, in 2012.

CARSTEN MAPLE (Member, IEEE) is currently a Professor of cyber systems engineering and the Head of the Secure Cyber Systems Research Group, WMG, University of Warwick, where he is also the Principal Investigator of the NCSC-EPSRC Academic Centre of Excellence in Cyber Security Research. He is a Co-Investigator of the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, where he leads on Transport & Mobility and Warwick PI on the Autotrust Project. He is or has recently been funded by a range of sponsors, including EPSRC, EU, DSTL, the South Korean Research Agency, Innovate U.K., and private companies. He has an international research reputation and extensive experience of institutional strategy development and interacting with external agencies. He has published over 250 peer-reviewed articles and is a coauthor of the *U.K. Security Breach Investigations Report 2010*, supported by the Serious Organised Crime Agency and the Police Central e-crime Unit. Additionally, he has advised executive and non-executive directors of public sector organizations and multibillion pound private organizations. He is a fellow of the Alan Turing Institute, the past Chair of the Council of Professors and Heads of Computing in the U.K., a member of the Zenzic Strategic Advisory Board and the IoTSF Executive Steering Board, U.K. Computing Research Committee, the ENISA CarSEC Expert Group, the Interpol Car Cybercrime Expert Group, and Europol European Cybercrime Centre, and an Executive Committee Member of the EPSRC RAS Network.

MUHAMMAD NABEEL ASGHAR received the Ph.D. degree from the University of Bedfordshire, U.K., with a focus on modeling for machine vision, specifically digital imagery, and its widespread application in all vistas of life. He is currently an Assistant Professor with the Department of Computer Science, Bahauddin Zakariya University, Pakistan. He has been investigating machine learning approaches for analysing video content ranging from broadcast news, sports, surveillance, personal videos, entertainment movies, and similar domains, which is increasing exponentially in quantity and it is becoming a challenge to retrieve content of interest from the corpora. Also, on their applications such as information extraction and retrieval. His recent work is concerned with multimedia, incorporating text, and audio and visual processing into one dynamic novel framework. His research interests include information retrieval, computer graphics, computer vision, image processing and visualization, graphics modeling and simulation, CR MAC protocol design, the Internet of Things, and security issues in wireless communication systems.

SAIF UL ISLAM received the Ph.D. degree in computer science from the University of Toulouse III Paul Sabatier, France, in 2015. He was an Assistant Professor with COMSATS University Islamabad (CUI), Pakistan, for three years. He was a Focal Person of the Research Team, CUI, working in the O2 Project in collaboration with CERN, Switzerland. He has been a part of the European Union-funded research projects during his Ph.D. studies. He is currently an Assistant Professor with the Department of Computer Science, KICSIT, Institute of Space Technology, Islamabad. His research interests include resource and energy management in large-scale distributed systems, such as edge/fog, cloud, and content distribution networks (CDNs), and the Internet of Things (IoT).