IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 August 2022, accepted 29 September 2022, date of publication 10 October 2022, date of current version 18 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213644

RESEARCH ARTICLE

Offensive Security: Cyber Threat Intelligence Enrichment With Counterintelligence and Counterattack

MUHAMMAD USMAN RANA[1], OSAMA ELLAHI[1], MASOOM ALAM[1], JULIAN L. WEBBER[2], (Senior Member, IEEE), ABOLFAZL MEHBODNIYA[2], (Senior Member, IEEE), AND SHAWAL KHAN[1,3]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]Department of Electronics and Communication Engineering, Kuwait College of Science and Technology (KCST), Doha 35003, Kuwait
[3]Department of Software Engineering and IT, Ecole de technologie superieure, Universite du Quebec, Montreal, QC H3C IK3, Canada

Corresponding authors: Shawal Khan (shawal.khan.1@ens.etsmtl.ca) and Muhammad Usman Rana (usman.amir90@gmail.com)

ABSTRACT Cyber-attacks on financial institutions and corporations are on the rise, particularly during pandemics. These attacks are becoming more sophisticated. Reports of hacking activities against government and commercial sector organisations have garnered a lot of attention in the last several years. By design, the focus of Cyber Threat Intelligence (CTI) is exclusively defensive. This is because most of the CTI-derived analysis output is intended to prevent breaches or facilitate early detection. So, there is a need to have a new mechanism for unmasking the attacker. In this research, we demonstrate cyber threat intelligence enrichment with counterintelligence and counterattack combined with certain new methods to exploit the adversary's vulnerability and fully control the attacker's system. Attackers use a VPN to establish an anonymous connection. A VPN creates a secure ''tunnelling'' to the internet, with the VPN server acting as a middleman between the attacker and the web. This provides anonymity because the attacker's IP address seems to be that of the VPN rather than his own, masking the IP address. So, hackers used this application to create persistence because it is automatically launched each time a computer is restarted. As a result, we are attempting to eliminate the persistence by removing it from the startup and registry. This research will help firms detect and identify an assault in its earliest phases, allowing them to respond accordingly. This project will develop new and innovative strategies to bypass VPNs and other security measures in order to obtain correct source information. Companies will be able to identify new methods by which their systems are penetrated and rapidly harden them. Using counterattack and counterintelligence, a proposed technique can bypass a VPN and get adversarial intel. The main goal of this research is to find the attacker's footprints or tracks and find out why the attack was planned in the first place.

INDEX TERMS Cyber deception, counterintelligence, counterattack, persistence, BSSID, IP, VPN, RDP, SSH, honeypot.

I. INTRODUCTION

Security breaches and attacks are becoming more common because of the ability of attackers to exploit weaknesses in people, processes, and technology. Cyber criminals have honed their tactics and methods (TTPs) to the point that they are almost impossible for law enforcement to identify and rectify. TTPs are less predictable, more persistent, resourceful, well-funded, and driven by money as they get more well-organized and well-funded. Ransomware, which encrypts data and systems and demands money to decrypt them, is affecting many businesses. Ransomware attacks such as those that began on May 12, 2017 and spread to 150 countries and infected more than 230,000 systems within a day are only one example.

Because of the rising quantity and increasing complexity of security events, cyber threat intelligence has gained a lot of media attention in the last several years [3]. With the

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

proliferation of open-source and commercial sources of threat information, many organizations have chosen to make use of these services. The issue is that while too much data is being used, there is also a dearth of data. As a result, there will be a problem with information overflow. Cyber threat intelligence data is being managed and converted into actionable information, distributed to the various tools, and used to aid in incident response through the Threat The Intelligence Sharing Platform (TISP) Terrorism threat intelligence feeds and systems are now being offered by information security providers and the cyber security industry. Content aggregation may provide a variety of threat data feeds, and a Threat Intelligence Management System can be used to derive commercial value from the information gathered. These two types of solutions can be combined to create a comprehensive solution.

Providers like FS-ISAC, OASIS, IBM X-Force Exchange, Facebook Xchange, HP Threat Central, Checkpoint IntelliTrace, AlienVault OTX, and Crowd Strike intelligence exchange are putting more focus on material collection. More attention is being paid to threat intelligence management systems, such as Intel Works, Soltra, Threat Stream, and Vor Stack, to name a few. Many security providers have defined cyber threat intelligence in a way that is tailored to fit their marketing and commercial plans [1]. Because there isn't much academic literature about CTI, there isn't a lot of clarity in the community about how threat information is defined, what standards are used, and how they are used.

Somehow, in today's world, words like ''data breach,'' ''vulnerability,'' and ''cyberattack'' are commonplace.

It is becoming increasingly difficult to keep up with difficulties as technology advances and an increasing number of gadgets are connected to the Internet. Known colloquially as canary tokens, honeytokens have been around for a while but are a good source of information. Unique IDs that can be used in a variety of locations are what they are. If they are contacted, an alert will be sent [8]. Figure 1 demonstrates a taxonomy diagram of cyber threat intelligence enrichment.

Similarly, HoneyBadger is a honeypot designed to provide hackers access to administrative features. It has ActiveX controls and Java applets as apps. According to Strand, it geolocates the hacker to within 20 metres when they think they've succeeded in hacking into the site. Using smartphone geolocation technology, the tool triangulates a user's position with respect to other local cell sites and wireless access points. This makes it easier for law enforcement to act. HoneyBadger is a geolocation framework that focuses on a specific area. As with conventional honeypots, HoneyBadger is an Active Defense technology that identifies the malicious actor and pinpoints their location. HoneyBadger uses ''agents,'' which are built into a number of different technologies and get the information they need from the [9].

Data from these agents is sent back to the HoneyBadger API, where it is stored and made available to users through the

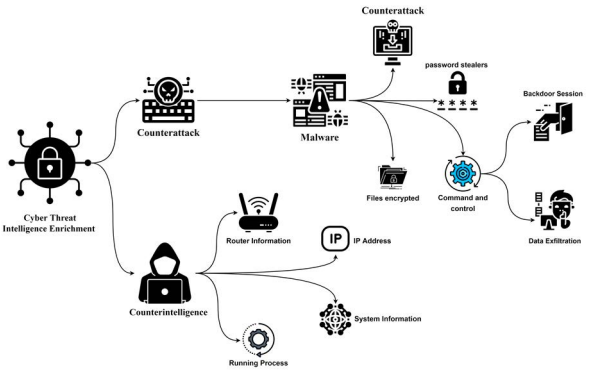

FIGURE 1. Taxonomy of Cyber threat intelligence enrichment.

HoneyBadger user interface. Using Word Web Bug Server, we can build a document that creates a callback each time it is accessed. This callback lets us know where the attacker is located based on their IP address. Linked design sheets and 1-pixel graphics are used to hide these vulnerabilities from the casual viewer. But the main problem with current strategies is that they can't unmask the real identity of an attacker because most of the attackers use VPNs to hide their IP addresses. Since most of the information that comes out of cyber threat intelligence analysis is used to stop intrusions or help find them early, CTI is only used for defensive purposes.

For addressing cyber-attacks and improving the broken processes of cyber threat intelligence, there is a need for an approach that can provide a full defence mechanism as well as concrete information about the attacker. Therefore, deception-enrichment-counterintelligence combines with some novel mechanisms that can better achieve the most information about the adversary and take full control of its system as well. It utilises honey tokens in the form of documents that are distributed over the network while masquerading as essential documents, database files, or DNS records within the system. When an attacker gains access to a system and tries to access sensitive data or files, honey tokens will be combined with these data files to identify a security breach. The Honey token will be used to conduct a counter-attack on the attacker, analyse and acquire more information about the attacker's machine while circumventing different defences such as VPN, thus disclosing the identity of the attacker.

The following components summarise the most important contributions of this work:

- To defeat the VPN and get actual intel of the adversary, we proposed a multi-stage algorithm.
- Document-based tokens: a malicious link is placed inside the word data file (once the link is clicked, the output is saved in the database).
- VBA scripts are being used in multiple documents by adding obfuscation methods.

TABLE 1. Comparative Analysis of Related Work.

Ref no, Name	Model	Methodology	Proposed Solution	Limitations
Year (2021) Best ways computation intelligent of face cyber-attacks.	Traditional activate cybersecurity model.	Defensive approach for electronic attacks (denial-of-service attacks, phishing scams, rootkits, and Day-Zero vulnerabilities)	Deploy firewall, SIEM tools filtering email and web, prevention of data Leakage and different security controls.	No mechanism for Counterintelligence and Decoys, must prioritize threat intelligence and security monitoring. Lack of revealing actual information. Lack of multistage counterattack. Lack of multiple OS compatibility.
Year (2019) Multi-Platform Honeypot for Generation of CTI	Multi-Honeypot Framework	Defensive approach to detect attacks and gather information.	Presented approach to collect cyber threat intelligence.	No mechanism to control attackers and perform real time data evaluation.
Year (2021) "Foureye: Defensive Deception based on Hypergame Theory Against Advanced Persistent Threats"	Hypergame theory	Defensive deception	To cope with Cyberattacks beyond the reconnaissance stage.	Need more realistic scenarios predict a next move of its opponent moving target defense
Year (2020) Cybersecurity Deception Experimentation System	Behavior-based model development	Evaluating dynamic deception algorithms	Cybersecurity Deception Experimentation System (CDES	Common Open Research Emulator hook functionality, Suricata (data generator) requires an interface when it is run.
Year (2021) Fake Document Generation for Cyber Deception by Manipulating Text Comprehensibility.	A novel computational method to generate fake documents	Comprehensibility manipulation of legit texts.	To counter the most sophisticated cyber-attacks, use bogus documents and deception techniques	Identify semantically related rephrases of a target concept to manipulate their occurrences.
Year (2020) CRATE Exercise Control – A cyber defense exercise management and support tool	Game Theoretic Approach	They have introduced a game theoretic approach to manipulate adversaries' belief through cyber deception	It describes CRATE Exercise Control, a tool for tracking and managing physical activity (CEC) and shown that that cyber deception is the need of time to respond sophisticated attacks	They describe the importance of cyber deception but there is lack of evidence of their research as no practical work was done
Year (2020) Cyber Deception for Computer and Network Security: Survey and Challenges	Game theoretic modeling at the strategic level	high-level deception schemes and actions. Discussed cyber defense strategies with modeling through the game theory	describe cyber deception in detail and examine elaborate deception strategies and tactics.	No framework for improving cyber deception. Computer and network security research is needed to have a solid scientific foundation.
Year (2021) Offensive Security: Towards Proactive Threat Hunting via Adversary Emulation	Novel hybrid model	Reactive Approach for hunting threats	The intended strategy is to discover and respond to previously unidentified dangers.	Increase the realism of adversary simulation with sophisticated hidden attacks in the future.

- Word document can bypass the sandbox and is not detectable. An Excel document can bypass the anonymous connection.
- To avoid an attack, the lure files will have a payload hidden in macros that accesses our server's reverse shell.

This section introduces notable academic and commercial initiatives that are related to the topic at discussion.

II. RELATED WORK

In this research, the authors take a defensive strategy for gathering and analyzing threat information and conducting security monitoring for electronic assaults and security vulnerabilities (such as a denial-of-service (DoS) attack or a phishing scam).

Installing a PDF exploit and handler, they suggested the system create an active and interactive honeypot for counterattacks via reverse TCP. The honeypot's primary goal is

to capture the identity of the attacker rather than to monitor the actions of the attacker. The proposed system may collect information such as inbound traffic, VPN existence, and the attacker's routing details using the biteback method [3]. According to their findings, the cyber threat intelligence field's analysis is frequently insufficient, and that was our perspective in this study. This is largely due to flaws in the study's methodology. It's as simple as this: because of an unreliable procedure, CTI is now providing poor service. This, on the other hand, is perfectly fair. Even though CTI has already helped reveal numerous intrusion campaigns carried out by hacker groups affiliated with countries or governments, the field is still in its infancy and requires further development to reach a more mature stage. They argue that it requires CTI to draw on techniques developed in the field of intelligence studies as a starting point. They've also shown that the area faces several obstacles that must be overcome. Qualitative and supply issues will be mitigated, as well as bias and actor naming issues, when the CTI field improves its approach. The authors present the honey trap base as a low-level interaction honeypot for effective detection and enhanced security controls [4].

A persistent threat actor in West Asia is using Microsoft OneDrive for command-and-control (C2) purposes. Trellix experts have linked the endeavour to APT28, dubbed ''Fancy Bear,'' a threat actor linked to Russia's military intelligence service. According to Trellix's campaign data analysis, Eastern European threat actors target military and government entities. Trellix's multistage APT28 campaign started with a phished Excel file. The file exploited a Microsoft browser engine remote code execution vulnerability (MSHTML or ''Trident''). Microsoft found the zero-day vulnerability in September following reports of exploits. A malicious DLL programme ran in the affected system's memory, and Trellix's ''Graphite'' virus was downloaded. A Microsoft Graph API lets Web apps use Microsoft Cloud services. Graphite is a DLL executable developed on the Empire open-source post-exploitation remote administration architecture, according to Trellix. The trojan installed an Empire agent after a multi-stage infection chain. Trellix's principal scientist dubbed the threat actor's cloud based C2 technique ''unique.'' ''Using OneDrive as a command-and-control server was shocking,'' he adds. An attacker may encrypt a victim's files. Once the attacker's OneDrive syncs with the victims' PCs, the encrypted instructions are executed, says Beek.

III. RESEARCH METHODOLOGY

This article presents research gaps that limit counterintelligence and counterattack for targeted threat intelligence to conduct proactive adversarial system intelligence and take control of attackers' machines. This research aims to provide organisations with realistic document-based tokens and a proactive defensive environment so they may capture attackers' system information, threat intelligence, attack pathways, malware, and TTPs to execute threat hunting. Our method is separated into two parts. In the first step, we develop malicious Word, Excel, and PDF documents to deceive attackers and identify APT attempts early. We used cyber deception to obtain opponents' information and discover them early.

A. METHODS USED

This study's main goals are data collection and analysis. We've set up honeypots like Cowrie and Windows, inserted lure documents, and used secure shell (SSH) and remote desktop protocol (RDP) to collect information on vulnerable computers. Honeypots and lure files are used for counterintelligence and counterattack, where threat actors may exploit their weaknesses and provide information. This data is collected after the system is installed and functioning but before it's placed into service.

The operation used a well-known company's public IP address. The attackers misunderstood this system as part of the organisation and took advantage of the situation by stealing important files. Attacks were registered, allowing us to study them in detail, uncovering important information about the attacks. This lets us obtain adversary information.

This investigation will find the attacker's traces and the exact goal of the attack.

All the testing data utilised in this project is authentic, unmodified information.

B. DATA ANALYSIS

We used a variety of tools and packages in many languages, as well as public online resources, to analyse data. The most popular coding tool is ''Visual Studio Code.'' We picked it because it has a simple user interface, requires little processing power, can be extended, and allows me to run my code in several conditions at once. Visual Studio Code's easy integration with Google Collaboration was another important factor. We mostly used Visual Studio Code, a Microsoft software; VBA; and Google Collaboration. When it comes to the data analysis aspect of the project, we've decided on Python since it offers an easy-to-use syntax, a large support community, and the availability of online packages.

C. EVALUATION METHOD(S) AND CRITERIA

As we know, in previous methods, counterintelligence is used for deception. Deception is not a new idea and has been extensively utilised by academics and practitioners since its inception, but there are numerous unknown use cases that might make it the most cost-effective and commonly used security solution on the market. The most common scenario includes the use of cyber deception, counterintelligence, and counterattack frameworks, as well as the uncertainty factor. We have utilised decoy files to enter an attacker's computer. These papers allow us to counter the attacker and decipher the hidden information. In the next few paragraphs, we will offer a comprehensive analysis of the results achieved using this research product, as well as a comparison of these results to those obtained via the use of a basic system that lacked a counterattack function.

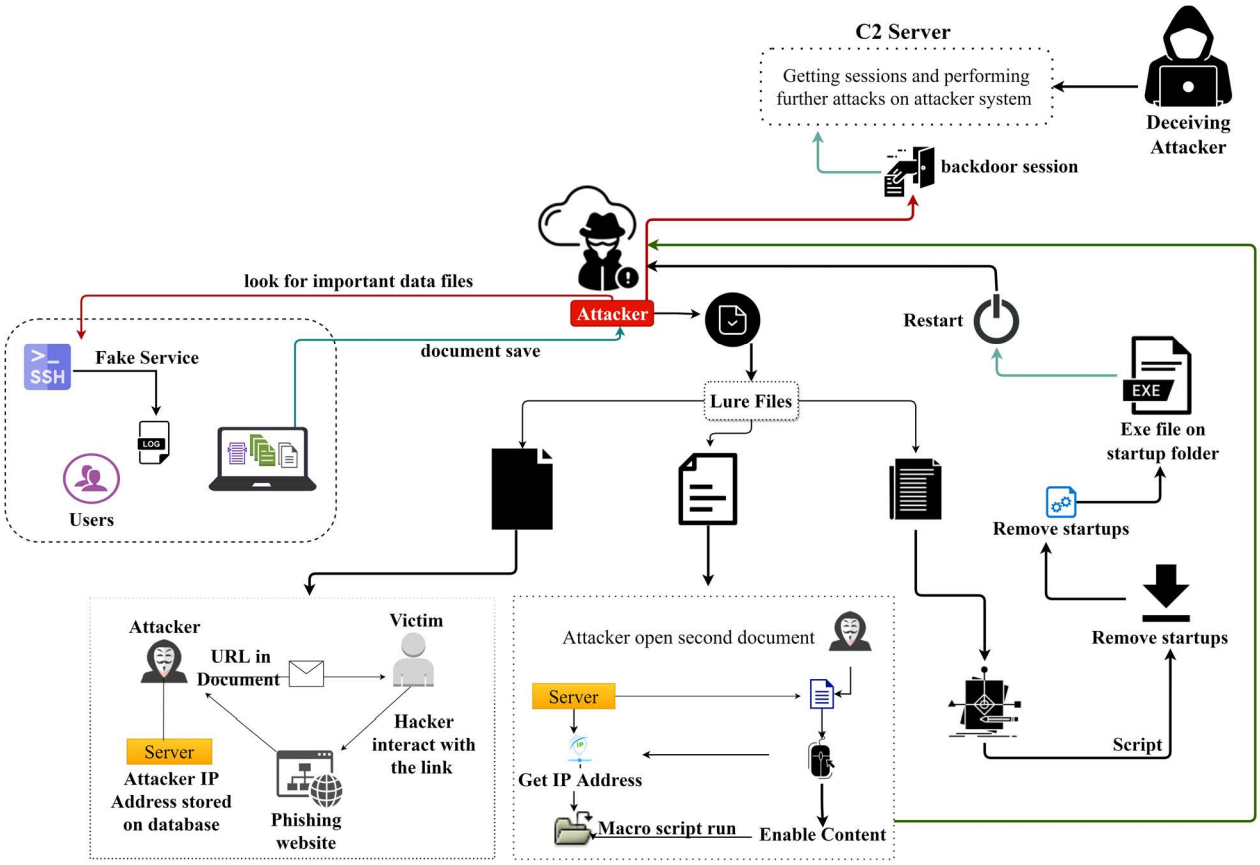

FIGURE 2. System architecture.

IV. A PROPOSED MODEL FOR COUNTERINTELLIGENCE AND COUNTERATTACK

Counter-threat intelligence is used so infrequently, and cyber-attacks have increased to such an extent that, in addition to mitigating them, we should have a look for other novel methods based on which we could not only stop the attacks but also easily track any attacker, and even hack into the system completely. This study covers the research gaps. As it stands, the main problem with the previous techniques is what if the attacker uses a VPN? Then the basic information the attacker displays is wrong, like the IP address. To solve this problem, we have created an environment for counterintelligence and counterattack. In this research, we created an environment where we created an ESXI server in a data center. On the server side, public and private paths are created where we can get information about the attacker logs. Furthermore, we use Cowrie and Windows honeypots, in which multiple ports are open like SSH. Logs are stored on a log server, and they deceive the information contained in these logs via a private path.

A. PROPOSED MODEL ARCHITECTURE

Our system setup includes ESXI, the VMware hypervisor, and an SDN, enabling numerous virtual machines to operate as docker containers (for low-interaction honeypots) or fully functioning computers (for high-interaction honeypots). My decoy arrangement uses Ubuntu's Open Virtual Switch (OVS) to spin up PCs. "Rules" refer to the mechanism that decides which honeypots a country should deploy to attract additional attackers and acquire threat information. OVS hosts a lot of fake Docker containers, which can be told apart by how they respond to an attacker.

The Payment Gateway only communicates with Cowrie, Conpot, and Snare Tanner. Because Windows is a complete OS, attackers target it more. Accordingly, honeypot restrictions fluctuate based on involvement and efficiency. Behind the firewall is a deception unit housing the complete system [34]. A single piece of equipment houses high and low engagement honeypots. We only let an attacker see services, machines, and data when they try to connect to the deception system. This is true whether they use a VPN or proxy.

On the server side, we have access to both public and private attack logs. We use SSH, Cowrie, and Windows honeypots. We use a log server in a secure manner to access logs. A well-designed RDP/SSH service should include an interactive interface for entering documents such as user passwords and transaction data. An attacker is fooled by a C2 server. Here's a system architecture description and diagram in Figure 2.

By assisting him in breaking into the system and launching attacks on his machine, the next step is for us to make real assets safer and more protected (by analysing this collected information of the attacker attempting to break into the system and engaging in malicious activities) by analysing the intel and making use of a log server that collects all this information in the form of logs. Then, logs are inspected, judgements are made, and policies are improved.

B. DATA ANALYSIS

Our Counterintelligence and Counterattack framework includes the following components:

1) HONEY TOKEN GENERATION

The Honey Token Generation Module generates tokens that are undetected by antivirus software and do not have any signatures. Marco's identity will be concealed by tokens generated, which will generate harmful tokens.

2) INFORMATION GATHERING USING COUNTERINTELLIGENCE

Code obfuscation is used to add automated scripts to the documents that are made. These scripts run when the token is accessed and gather information about the system from which the token was taken. The information gathered includes the information about the attacker's operating system, the type of network he or she is using, whether he or she is using Wi-Fi or Ethernet, and the information about the network's banner.

3) ACCESSED IP GEOGRAPHIC LOCATION

The attacker's position can be determined with the aid of this module. We may utilize the BSSID of the attacker's computer to determine the attacker's location. It will be shown a map with pins showing the location of the attacker. The location of an IP may be filtered by tokens, country, etc.

4) WINDOWS BASIC TOKENS

We deploy our embedded tokens into the network in the form of Microsoft Office documents (Word, PowerPoint, and Excel files). Tokens may only be used on PCs running Windows. The attacker can obtain the tokens from the EXSI framework. Tokens will include embedded payloads that are distinctive to a single user.

5) HONEY TOKENS FOR LINUX

Using documents that are supported by Libre Office and Open Office, we can produce lure tokens that are compatible with Linux. These tokens will be used for counterintelligence.

6) MALICIOUS DOCUMENTS

The generation of harmful documents is going to be helpful to the counterattacking framework. To stop an attack, the lure file will have a payload hidden in macros that will give the user access to the server's reverse shell.

7) COUNTER-ATTACK MANAGEMENT

After an attacker gets into our honeypot, we'll use the Counter-Attack Management module to act against him. The system will be accessed via macros. Using the tokens that the attacker got a hold of a counter-attack mechanism will be turned on. There are several alternatives available to the user when it comes to the harmful token. Once the user has the token, they can do things after the exploit has been used.

8) DROPPER FILES

The attacker group mainly targets people through fake job offers or stolen personal data from systems. Although our lure documents share certain similarities with the previous ones, we scanned the Word document for viruses and assessed it. When the malicious code that was planted is activated, a backdoor is put in place.

9) THE IP ADDRESSES INFORMATION

Using this module, we can retrieve a list of IPs that can view our lure documents using the counterintelligence function. This script will get the user's IP and user's agent. We can get the user's country name and city as well as device information like Mac addresses, etc.

10) JAVASCRIPT FILE DROPPER

This file can create a malicious JS file that can be inserted into several files, such as MS Word (.doc,.docm,.docx,.dotm), Excel (.xls, .xlsm, .xslx, .xltm), PPT (.pptm,.potm). We utilize the HXD editor to disguise our VBA scripts into base 64 format, then export them into c# files, and finally into Java Script files. This file can be used for counterintelligence and counterattack.

11) MANUAL ATTACKING USING SHELL

As the tokens are obtained by the attacker, the user is granted reverse shell access. It will be possible for the user to perform commands on the shell of the attacker. On the C2 server, the user will get the result of these operations. Post-exploit, the user might utilize the shell access to carry out further attacks. To keep the connection open, the token's macro will use shell access as a back door by taking advantage of shell access.

12) DISABLING MACRO-OPTION VIA EXE

It is very necessary to have an understanding of what the system loads at startup in order to account for the fact that the registry is loaded before the kernel. While attempting to take a command-and-control session inside the attacker's system, there are a number of issues that arise. In order to get control of the adversary's system, we are in the process of creating many lure files. Our lure document is downloaded via exe. The moment an attacker clicks on our executable, the values in the registry are modified, and malicious files are downloaded on the attacker's machine.

TABLE 2. Semantic details used in Algorithm 2.

Function	Description
OM ()	**Simple obfuscation, malicious payload, windows.**
Macro 4.0 ()	Vba script, shellcodes, TTP's and attack methods used by attackers, download exe file on system and run malignant code.
Generate ()	To create the Macro 4.0 malicious, excel donut document.
Collection ()	Malicious document malicious payload, malicious URL, malicious code, c-sharp application.
MP()	Convert C# code into xls format then put into the excel donut file.
Categorized ()	Gathering the information about the attacker, creating a backdoor and performed more realistic exploitation.

TABLE 3. Details of Security Payloads.

Type	Description
.XLS	Embedded Vba Script
.XLSM	Excel Macro Enable Document.
PDF	Document with hidden Java Script payload.
Doc/Docx	Vba Script in Word document.
PSI	PowerShell Malicious Script.
BAT	Malicious .bat file.
DOCM	Word Macro Enable Document.
Web.doc	Without VBA Script Malicious Document.
DLL	Malicious dll's.
.PPTM	Malicious JS Embedded MS PowerPoint file.
XSLX	Hidden VBA code dropper file.

C. DATA ANALYSIS

This section contains harmful document-generating algorithms. We recommend bypassing the VPN and creating a multistage algorithm. We utilise Excel to run multistage processes. Below is code for a malicious file that runs algorithms. The first three-part algorithms illustrate how to beat the VPN (Algorithm-1), the second how to build a malicious document (Algorithm-2). For this, we utilise the Excel Donut Repository's xlm macro generator. Run a C# (exe) application in memory using a Microsoft Excel 4.0 macro. Xlm (Excel 4.0) macros may be placed in xls files. A malware payload C# file (something like a Cobalt Strike Beacon EXE with a main function that runs). Microsoft Visual turns the C# code into two.NET assemblies, one for the x86 architecture and the other for the x64 architecture.

Algorithm 1 To Defeat the VPN in Counterattack

INITIALIZE: OS-info [], *Presisinfo[]*
foreach All-Software's-with-persistence[] in software []
    Presisinfo [] ← *software-Name*
    Presisinfo [] ← *software-location*
end
foreach Presisinfo [] *in soft do*
    Remove Registry Value(soft)
end
foreach startupfolder-software[] in soft do
    Delete (soft)
end
Startupfolder←Download←custom-information-gather.exe
Reboot()

D. ATTACKING PAYLOADS

In our deception environment, we have many attachments file types. Allow users to be notified when a document is opened, or macros are running.

Our counterintelligence and counterattack strategy use the payloads listed in Table 3. Experiments have utilized VBA Scripts, PE files, OLE files, and PS1. The payloads indicated, such as the Excel document payload, may be created using (Algorithm 2).

V. EXPERIMENTAL RESULTS AND EVALUATION

The report was published on Joe's sandbox, and the process tree reveals that the main executable first checks the system name, then the mac address, and finally the whole system information [49]. This assault against Aljazeera was a total recognition. Attackers may or may not prepare a broader assault based on this one. But this attack shows the intruders knew about Aljazeera's anti-virus software. If an attacker knows about the anti-virus, they may target it. Malware experts have classified this attack as both an evader and a Trojan. Spyware and ransomware may have a persistent link to the victim's PC. A categorization of the attacks is shown in the figure.

Use Case of Deployment: High interaction honeypots cowrie and window are deployed in our deception system. Our proposed deception topology changes dynamically after some sequence of time, which makes it more realistic and effective. When an attacker interacts with a deception system that is really a fake machine with lure documents, the attacker will get files that could be used to launch a counterattack against the attacker's machine.

A. WINDOWS LOG

a) Figure 4 shows the window honeypot logs and activities done by the attacker. For better evaluation of our proposed approach, we mapped collected TTP's and payload onto the MITRE ATT & CK framework. Windows honeypot logs show that the attacker executed some commands, stole some information, tried to move laterally, and escalated privileges. The wmi service is used to run code, and the svchost.exe file is used to move laterally and avoid defenses.

b) We use System Monitor (Sysmon). It is a popular Windows logging add-on. Sysmon can watch code behaviour

Algorithm 2 Create Malicious Document

Input: ES (meterpreter shell code), C2 (Lhost, Lport) OM(simple obfuscation), c-sharp (payload)
Collection: Mal-docx [], Mal-pay [], Mal-URL [], Mal-code [], C# (payload)

Initialize

1. *OM(simple obfuscation), L-Host, MP=Payload*
2. *Output.*
3. *Compiled code for Macro 4.0 Excel, Payload*
4. *Excel-donut (Malicious Payload)*
5. *If (MP =Macro 4.0)*
6. *Result ← call OM(MP)*
7. *If(MP=OM)*
8. *Result ← Macro 4.0*
9. *Compiled.(Excel 4.0 Macro)*
10. *Generate(4.0 Macro)*
11. *Embedded(Macro 4.0, Xls)*
12. *If (MP=URL)*
13. *Result ← generate(Mal-docx.docx)*
14. *If (MP=C#[payload])*
15. *Result ← generate(Excel-xls)*
16. *If (MP=Es)*
17. *Result ← call(C2)*
18. *Compile ← MP*

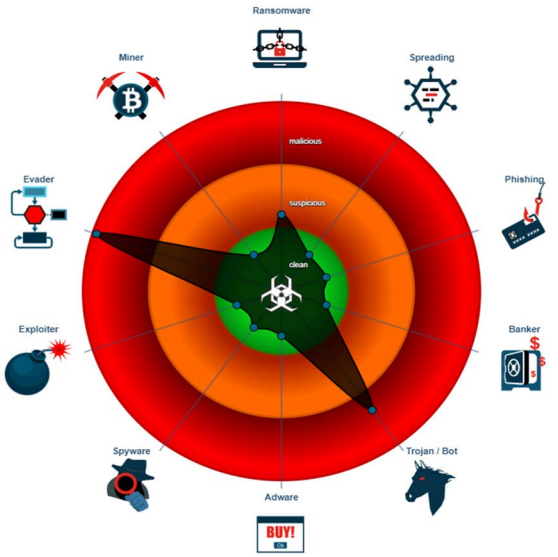

FIGURE 3. Classification of attacks.

and network traffic to detect malicious activities and produce detections based on them. The attacker used a technique called Hijack Execution Flow: DLL Side-Loading Malicious DLLs can be side-loaded by attackers. The side-loading technique is like DLL Search Order Hijacking. Alternatively, attackers can directly side-load their payloads by planting and contacting a real application that executes their payloads. Side-loading exploits the loader's DLL search order to find the victim software and malicious payload. It is probably used

FIGURE 4. Windows logs.

FIGURE 5. Cowrie logs.

by attackers to hide actions conducted by trusted, authorized, and maybe elevated system or software processes. Executables used to side-load benign payloads may not be identified during delivery or execution. Until they are loaded into the memory of a trusted process, payloads from enemies are often hidden or encrypted.

B. COWRIE

a) Figure 5 shows the logs for experimental results of the Cowrie honeypot, which describe that an attacker tried to establish SSHD connections. The log file shows the source IP address, source port number, connection type, and indication for SSHD connections. Attackers used the ssh service to get in and then tried to run commands and steal our lure files. They also tried to execute remote commands. We have collected the logs of all commands executed by attackers using a SSHD connection. Cowrie honeypots are used to deceive attackers and catch the brute force attempts used by attackers to login. A fake directory is provided to attackers after successful login. A weak password is set, so that an attacker can brute force it and log in successfully in a fake directory. Inside the fake directory, we place our malicious document type tokens that are used to do counterintelligence and counterattack features.

VI. OBFUSCATION

By default, everyone who can run the Visual Basic Editor may access Excel VBA code. Any user may read unprotected VBA code, inflicting harm. Macro viruses may introduce dangerous code that pranks or undermines PC security. Protect your VBA script. You may still run the VBA code with the password. Unfortunately, there are various free software

solutions that may quickly and simply recover a forgotten or lost VBE password. Several spreadsheet programmes can natively read Excel VBA (without the need for Excel). VBA obfuscation helps. If the original code was written to operate in both 32- and 64-bit Office, obfuscated VBA code can too. Excel VBA's runtime doesn't affect hidden VBA code. When utilised properly, obfuscation may significantly increase Excel's security while keeping it unhacked. These obfuscated files are presented in Figure 6. Screenshots show the obscured VBA script that is used in documents.

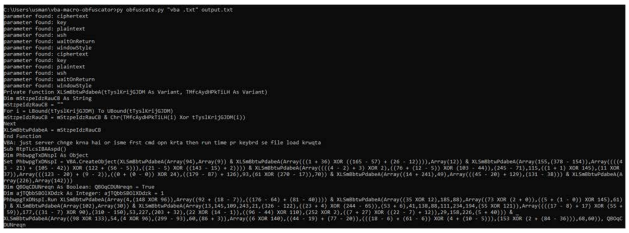

FIGURE 6. Vba code obfuscation.

A. MACRO-ENABLE OPTION PREVENTION

VBA code is used to store macros, which are small computer applications. It is possible for VBA developers to create macros that can do almost any operation on a computer, including accessing any external resources the machine may be linked to. Powerful tools like Excel's built-in macros may be misused to propagate malware, hijack machines for botnets, steal data from databases, or send e-mail spam if they get into the hands of the wrong people. You should only allow macros in workbooks that you know and trust, and be suspicious of any workbooks that have macros enabled by others.

A yellow ''SECURITY WARNING'' bar will display under the ribbon by default when you first open a worksheet with macros enabled. The ''Enable Content'' button will allow macros to be activated. When the user clicks on the macro or content, the malicious code is executed in the document. Suppose, however, that the user didn't choose the checkbox to allow macros. To address this issue, we've developed a way that allows us to perform actions on the user's operating system by altering the machine registry. When our malicious document is run on the target PC, the Enable Macro option won't show up if we change the value of Enable Content or Macro in Memory from 0 to 1.

The given command can be used to change the value in the registry. We can use this command via an a.Bat file or a VBA script and Figure 7 displays the result.

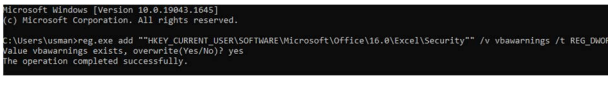

FIGURE 7. Registry value change.

B. COUNTERATTACK USING RAR FILE EXPLOIT

WinRAR, a popular file archiver, has disclosed a long-standing flaw. Check Point Research found a path traversal zero-day vulnerability in 'ACE' structured files that enables attackers to pick arbitrary destinations during file extraction. They can acquire persistence and code execution by extracting files from vulnerable locations like the Windows ''Startup'' menu. WinRAR 5.70 fixes this issue, but there is no automatic updating mechanism, so many users may continue to use obsolete versions. FireEye and 360 Threat Intelligence Center have identified similar attacks. In these operations, there are many different types of decoy documents and payloads, some of which we've never seen before and some of which use common technologies like PowerShell Empire.

In our decoy file, a password-protected RAR package named ''yearly financial information.rar'' was discovered to start an infection chain that resulted in the installation of a backdoor (''execute.exe'') on the targeted machine. Furthermore, the attacks steal malicious instructions from the target workstation, with the implant creating a separate folder on the server for each compromised host. We got the information about the attacker's machine via telegram. Basically, this executable file sniffs the victim's system information and sends it to our server. Several victims opened our exe file, and we obtained system information for a couple of them. Figure 8 shows the File Dropper Flow Diagram.

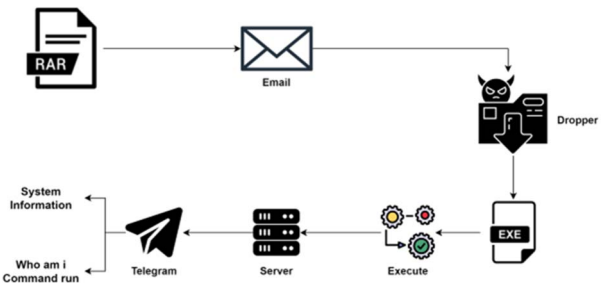

FIGURE 8. File dropper flow diagram.

C. MALICIOUS JAVASCRIPT FILE DROPPER

This file can create malicious JS code that can be inserted into multiple files, for instance, MS Word (.doc,.docm,.docx,.dotm).

Microsoft PowerPoint (.pptm,.potm) and Microsoft Excel (.xls,.xlsm,.xslx,.xltm). For implementation, we use the HXD editor for obfuscation where we can change our VBA script into base 64 format. Then we can export this obfuscated script into a c# file. After this, we can put this script into a Java Script file. Figure 9 demonstrates the different constraints of Java Script code execution. Inside the Java Script code, we have the option to change the file extension to something like Excel, PDF, or Word. The screen illustrates the information from our experiments.

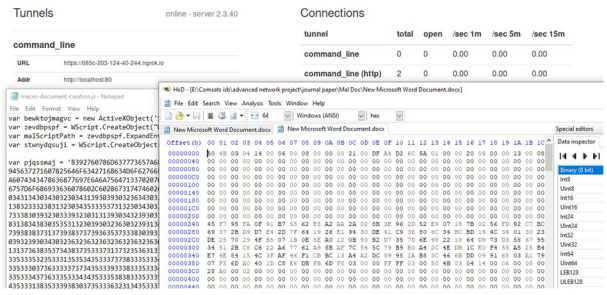

FIGURE 9. JS file execution.

FIGURE 10. Word file dropper flow diagram.

D. WORD FILE DROPPER

Individuals are usually targeted by the gang using booby-trapped job offers or organizational private detailed information that is placed on systems. Although the new campaign has some of the same characteristics as the previous one. Flow diagram shown in Figure 10 and the system information of the victim are shown in table 4.

TABLE 4. Response.

Parameters	Description
Host Name	DESKTOP-KOUVV35
OS Name	Microsoft Windows 10 Pro
OS Version	10.0.19043 N/A Build 19043
OS Manufacturer	Microsoft Corporation
OS Configuration	Standalone Workstation
Registered Owner	PsychicPowers
Original Install Date	03/02/2022, 12:20:44 PM
Windows Directory	C: WINDOWS
System Directory	C: WINDOWSsystem32
Time Zone	(UTC-08:00) Pacific Time (US and Canada)
Logon Server	DESKTOP-KOUVV35
IP Address	203.124.40.244

The Word document was analyzed by us and tested for viruses total as well. It provides information about various positions at IT, a xyz. -based consulting firm, only to start the infection chain when the embedded malicious code is activated, resulting in the deployment of a backdoor.

Aside from acquiring basic information about the attacker's system, the backdoor connects to a remote server and waits for further instructions that enable it to receive files from the server, upload arbitrary files, and run shell commands, with the results being sent back to the server. The given screenshots explain the exact information that is being collected by us on the server. The victim used a scanner to check the file, but it did not show any malicious things inside the file. So, our Malicious doc file can bypass the scanner as well.

E. FIND LOCATION VIA WORD DOCUMENT (DOCX, DOTM, DOCM)

We can use multiple word document formats for the purpose of getting system information from the attacker's machine as well as finding out its location. Using the "Counterintelligence" feature, a proposed technique may bypass defenders and get adversarial intel such as actual IP addresses, proxy servers, running processes, system information, incoming and outgoing data, and Media Access Control addresses (MAC addresses). Experimental results show multiple things in detail. For instance, system information plus BSSID Using a mac address, we can find the actual location of the attacker. We used an online free source tool named wigle.net where we put in the BSSID and the location of the cyber crooks.

F. BASH SCRIPT FOR LINUX

As we know, it is used in the Linux environment. When we open this file in an authorized user mode like sudo command, then it creates an account like test4. In the Test4 folder, passwords are disabled. In the home directory or the root directory, a file named. bashrc is created. Whenever we do the search for a user or log in the user, then that bashrc file is executed first, so we append our backend URL at the end of the bashrc file. The curl command is used to request and then output like an IP address will be shown on our backend server.

G. VICTIM REPORT

We have received several responses from people who are trapped in honeypot environments. A few of them are attackers because their intention is to merely steal information from our machines. Some cyber security organisations are also involved in this activity. They stole our lure document where we put the malicious exe for counterattacking purposes and did a scanning process on it. The results show the organization's IP addresses as well as other information. Once, we put the IP address of the system on a free tool called IP-Lookup in order to see the domain of the address. We capture several other domains as well, shown in Figure 12, that are doing scanning of our malicious exe files, and one CYREN-named cyber security organisation is involved in this process as well.

H. MULTI-STAGE STRATEGY TO DEFEAT VPN

The fundamental difficulty with the previous tactics is that they do not work if the target uses a virtual private network

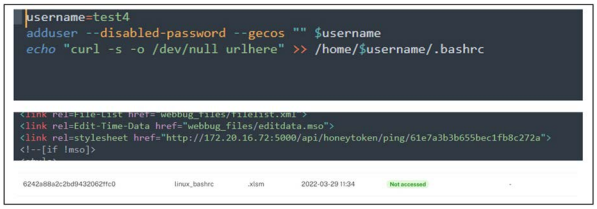

FIGURE 11. Token accessed result.

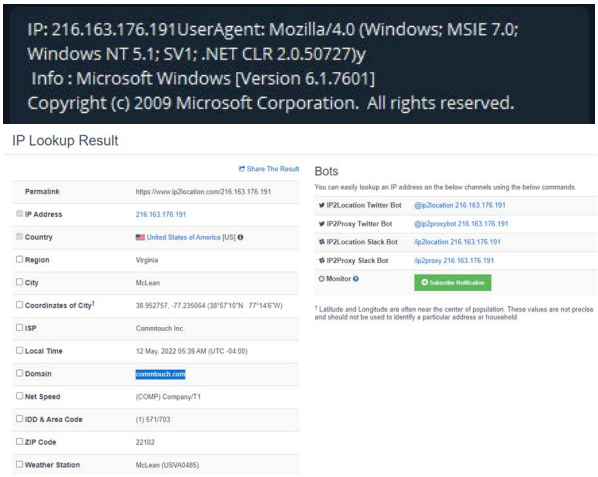

FIGURE 12. Scanning result and IP domain.

(VPN). The fundamental information about the user that is shown, such as the IP address, is incorrect.

We presented a hybrid technique in which we first get all the starting services, then delete those applications, insert our exe file in the appropriate location, and lastly reboot the system. After that, the machine uses an executable file to ping the public IP address.

The C-sharp code is put within the Excel donut document, which has an auto-enabled format; when the Excel file is clicked, the code is executed, and our algorithm is performed; we will also get system information. First, we will demonstrate the results of our use cases of defeating VPN in the form of screenshots in Figure 13, which shows the before Operation Startup Status. Figure 14 demonstrates the Code Execution Result. Similarly, after the operation startup status shown in Figure 15, we will show the persistence techniques that can be done via malicious exe.

As we can see, the experimental results illustrate how to defeat the VPN via multiple document files, for instance, excel and Word. In the next few paragraphs, we'll talk about how the word ''persistent'' is used when talking about security breaches. As we all know, attackers use paid VPNs to generate persistence, and under persistence, the VPN will automatically run every time the system reboots. To counter this persistence, we simply remove such programs from the startup and registry. Using the ''Counterattack'' features, a suggested method could get around a VPN and get information about the attacker, such as their real IP addresses,

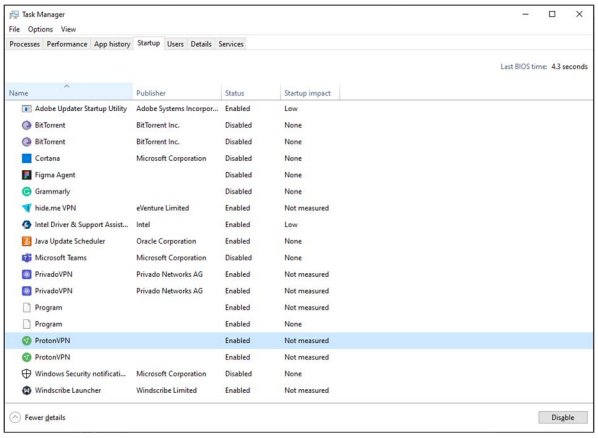

FIGURE 13. Before operation startup status.

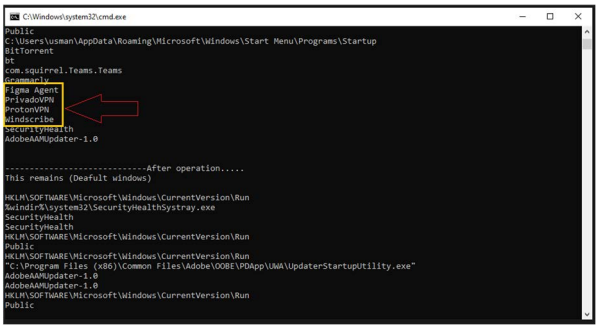

FIGURE 14. Code execution.

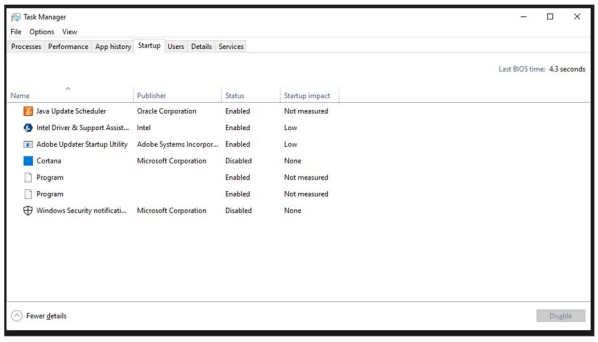

FIGURE 15. After operation startup status.

proxy servers, active processes, system data coming in and going out, and Media Access Control (MAC) addresses. Figure 16 illustrates MITRE ATT&CK mapping.

I. EVALUATION

We can evaluate our results into two parts. The first one is the evaluation of our lure files, which is illustrated in table 5. Secondly, the evaluation graph that is shown in figure 17 can be further divided into four parts. To begin with, 15 percent of the documents reported by attackers are lure files that are detected on the sand box. After this, our counterintelligence

TABLE 5. Documents evaluation.

Lure files	Counterattack									Counterintelligence				
	IP	Mac	Session	location	VPN info	VPN defeat	User agent	C & C	Version	IP	Host	Sys-info	User agent	Platform
Document 1	✓	✓	☒	☒	☒	☒	✓	☒	✓	✓	✓	☒	✓	✓
Document 2	✓	✓	☒	✓	☒	☒	✓	☒	✓	✓	☒	☒	✓	✓
Document 3	✓	☒	☒	☒	☒	☒	✓	☒	✓	✓	✓	✓	✓	✓
Document 4	✓	✓	☒	✓	☒	☒	☒	☒	☒	☒	✓	☒	☒	✓
Document 5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Document 6	✓	☒	☒	☒	☒	☒	✓	☒	☒	☒	✓	☒	✓	✓
Document 7	✓	✓	✓	☒	☒	☒	☒	✓	☒	☒	☒	☒	☒	✓
Document 8	✓	✓	☒	✓	☒	☒	✓	☒	✓	☒	✓	☒	✓	✓
Document 9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Document 10	✓	☒	☒	☒	☒	☒	✓	☒	☒	☒	☒	☒	✓	✓

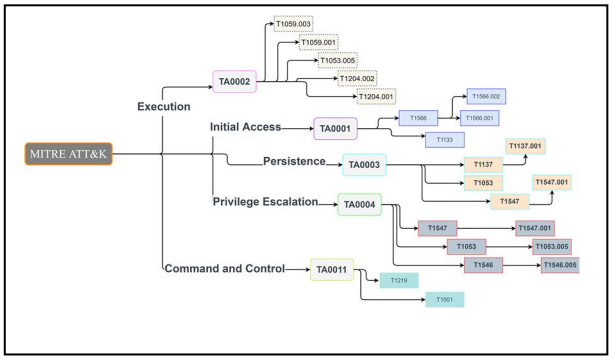

FIGURE 16. MITRE ATT&CK mapping.

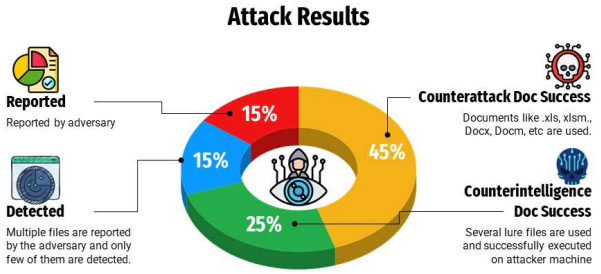

FIGURE 17. Attack results.

and counterattacking lure documents success rates are shown, which are 25 percent and 45 percent, respectively.

VII. CONCLUSION

It's difficult for companies to defend themselves against cyberattacks since the attackers are constantly improving their methods of infiltrating a network. Finding the breach after it has occurred is a difficult undertaking since the intruders are always coming up with new methods of evading detection. Hackers can access personal information even after a security breach occurs. Attackers make use of paid VPN, and paid VPN is used to generate persistence. Once the computer is in persistence, the VPN will automatically run itself anytime it reboots. Therefore, to counter this persistence, we have removed those programmes from both the registry and the start menu.

A proposed method can defeat a virtual private network (VPN) by utilising the "Counterattack" function to obtain adversarial data such as true IP addresses, proxy servers, running processes, and system information. A "counterintelligence" function can allow us to get the intel of an attacker machine via multiple techniques. We generate malicious Java Script code that may be placed into a variety of different files, such as Microsoft Word (.doc,.docm,.docx,.dotm). PowerPoint (.ppt) (.pptm,.potm) and Excel (.xls,.xlsm,.xslx, and.xltm). The proposed method improves the effectiveness of recognising and countering threats using real-world attack scenarios and includes an algorithm for generating malicious documents. In contrast to traditional methods that focus on known threats, the suggested approach is intended to reveal the attacker's identity.

VIII. FUTURE WORK

In the future, we can use more widely adopted techniques to get into the shells of attackers' computers. In the command-and-control process, there are existing frameworks that may be used. To get further information from the computers used by attackers, we may combine all existing public tools as well as develop our own custom tools. We want to develop other exploits that are compatible with a variety of systems and will do so. There are machine learning models that may be utilised in an environment that is designed to be misleading in order to lure in an attacker. Solutions like geo-location can attract

attackers. The greater the amount of work that must be done in front-end and back-end integration for a full framework, the more sophisticated the attackers these days are utilising increasingly sophisticated methods to mask their location, which means that in the future we will be able to cover every one of them.

REFERENCES

[1] B. Alhayani, S. T. Abbas, D. Z. Khutar, and H. J. Mohammed, "Best ways computation intelligent of face cyber attacks," *Mater. Today, Proc.*, Mar. 2021, doi: 10.1016/j.matpr.2021.02.557.

[2] S. Adarsh and K. Jain, "Capturing attacker identity with biteback honeypot," in *Proc. Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Jul. 2021, pp. 1–7, doi: 10.1109/icscan53069.2021.9526371.

[3] K. Oosthoek and C. Doerr, "Cyber threat intelligence: A product without a process?" *Int. J. Intell. Counterintell.*, vol. 34, no. 2, pp. 300–315, Jul. 2020, doi: 10.1080/08850607.2020.1780062.

[4] S. Kumar, B. Janet, and R. Eswari. (Dec. 1, 2019). *Multi Platform Honeypot for Generation of Cyber Threat Intelligence*. IEEE Xplore. Accessed: Mar. 7, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8971584

[5] Z. Wan, J.-H. Cho, M. Zhu, A. H. Anwar, C. Kamhoua, and M. P. Singh, "Foureye: Defensive deception based on hypergame theory against advanced persistent threats," 2021, *arXiv:2101.02863*.

[6] J. C. Acosta, A. Basak, C. Kiekintveld, N. Leslie, and C. Kamhoua. (Sep. 1, 2020). *Cybersecurity Deception Experimentation System*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9230074

[7] A. El-Kosairy and M. A. Azer. (Apr. 1, 2018). *A New Web Deception System Framework*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8442027

[8] P. Karuna, H. Purohit, S. Jajodia, R. Ganesan, and O. Uzuner, "Fake document generation for cyber deception by manipulating text comprehensibility," *IEEE Syst. J.*, vol. 15, no. 1, pp. 835–845, Mar. 2021, doi: 10.1109/JSYST.2020.2980177.

[9] A. Schlenker, O. Thakoor, H. Xu, M. Tambe, P. Vayanos, F. Fang, L. Tran-Thanh, and Y. Vorobeychik, "Deceiving cyber adversaries: A game theoretic approach," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, Jan. 2018, pp. 1–11. Accessed: May 16, 2022. [Online]. Available: https://par.nsf.gov/biblio/10050303

[10] S. Fugate and K. Ferguson-Walter, "Artificial intelligence and game theory models for defending critical networks with cyber deception," *AI Mag.*, vol. 40, no. 1, pp. 49–62, Mar. 2019, doi: 10.1609/aimag.v40i1.2849.

[11] N. C. Abay, C. G. Akcora, Y. Zhou, M. Kantarcioglu, and B. Thuraisingham, "Using deep learning to generate relational HoneyData," in *Autonomous Cyber Deception*, 2019, pp. 3–19, doi: 10.1007/978-3-030-02110-8_1.

[12] M. O. Sayin and T. Başar, "Deception-as-defense framework for cyber-physical systems," in *Safety, Security and Privacy for Cyber-Physical Systems* (Lecture Notes in Control and Information Sciences), 2021, pp. 287–317, doi: 10.1007/978-3-030-65048-3_13.

[13] J. Almroth and T. Gustafsson. (Sep. 1, 2020). *CRATE Exercise Control—A Cyber Defense Exercise Management and Support Tool*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9229649

[14] S. Huneault-Leblanc and C. Talhi. (Oct. 1, 2020). *P-Code Based Classification to Detect Malicious VBA Macro*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9297272

[15] Z. Lu, C. Wang, and S. Zhao, "Cyber deception for computer and network security: Survey and challenges," 2020, *arXiv:2007.14497*.

[16] V. Ravi, S. P. Gururaj, H. K. Vedamurthy, and M. B. Nirmala, "Analysing corpus of office documents for macro-based attacks using machine learning," *Global Transitions Proc.*, vol. 3, no. 1, pp. 20–24, Jun. 2022, doi: 10.1016/j.gltp.2022.04.004.

[17] A. B. Ajmal, M. A. Shah, C. Maple, M. N. Asghar, and S. U. Islam, "Offensive security: Towards proactive threat hunting via adversary emulation," *IEEE Access*, vol. 9, pp. 126023–126033, 2021, doi: 10.1109/ACCESS.2021.3104260.

[18] F. Fadly and D. Ulhaq, "Design of application for calculating human resources for medical record technician with ABK-Kes using the excel macro," *Jurnal Mantik*, vol. 5, no. 4, pp. 2524–2530, Feb. 2022. Accessed: May 16, 2022. [Online]. Available: http://iocscience.org/ejournal/index.php/mantik/article/view/1989

[19] M. A. R. A. Amin, S. Shetty, L. Njilla, D. K. Tosh, and C. Kamhoua, "Hidden Markov model and cyber deception for the prevention of adversarial lateral movement," *IEEE Access*, vol. 9, pp. 49662–49682, 2021, doi: 10.1109/ACCESS.2021.3069105.

[20] M. Roman, K. Alexander, S. Sergey, and S. Sergey, "Innovative development of tools and technologies to ensure the Russian information security and core protective guidelines," *Cyber Secur. Issues*, vol. 1, no. 29, pp. 10–17, 2019. Accessed: May 16, 2022. [Online]. Available: https://cyberleninka.ru/article/n/innovative-development-of-tools-and-technologies-to-ensure-the-russian-information-security-and-core-protective-guidelines

[21] R. Yamamoto and M. Mimura, "On the possibility of evasion attacks with macro malware," in *Soft Computing for Security Applications* (Advances in Intelligent Systems and Computing), Oct. 2021, pp. 43–59, doi: 10.1007/978-981-16-5301-8_4.

[22] R. Coulter, J. Zhang, L. Pan, and Y. Xiang, "Domain adaptation for Windows advanced persistent threat detection," *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102496, doi: 10.1016/j.cose.2021.102496.

[23] B. Xi and C. A. Kamhoua, "A hypergame-based defense strategy toward cyber deception in Internet of Battlefield Things (IoBT)," in *Proc. Modeling Design Secure Internet Things*, Jun. 2020, pp. 59–77, doi: 10.1002/9781119593386.ch3.

[24] M. M. Islam and E. Al-Shaer. (Sep. 1, 2020). *Active Deception Framework: An Extensible Development Environment for Adaptive Cyber Deception*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9230100/

[25] John Strand. *Getting Started With Tracking Hackers With HoneyBadger—Black Hills Information Security*. Accessed: Oct. 4, 2021. [Online]. Available: https://www.blackhillsinfosec.com/getting-started-with-tracking-hackers-with-honeybadger/

[26] *Good Attacks Make Good Detections Make Good Attacks Make*, Thinkst Canary. Accessed: Oct. 4, 2021.

[27] G. Kim, S. Kim, S. Kang, and J. Kim, "A method for decrypting data infected with hive ransomware," 2022, *arXiv:2202.08477*.

[28] T. Chakraborty, S. Jajodia, J. Katz, A. Picariello, G. Sperli, and V. S. Subrahmanian, "A fake online repository generation engine for cyber deception," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 518–533, Mar. 2021, doi: 10.1109/TDSC.2019.2898661.

[29] M. Zhu, A. H. Anwar, Z. Wan, J.-H. Cho, C. Kamhoua, and M. P. Singh, "Game-theoretic and machine learning-based approaches for defensive deception: A survey," 2021, *arXiv:2101.10121*.

[30] M. A. Kalwar, M. F. Shahzad, M. H. Wadho, M. A. Khan, and S. A. Shaikh, "Automation of order costing analysis by using visual basic for applications in Microsoft Excel," *J. Appl. Res. Technol. Eng.*, vol. 3, no. 1, pp. 29–59, Jan. 2022.

[31] V. Mavroeidis, R. Hohimer, T. Casey, and A. Jesang. (May 1, 2021). *Threat Actor Type Inference and Characterization Within Cyber Threat Intelligence*. IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/9468305

[32] V. Koutsokostas, N. Lykousas, T. Apostolopoulos, G. Orazi, A. Ghosal, F. Casino, M. Conti, and C. Patsakis, "Invoice #31415 attached: Automated analysis of malicious Microsoft office documents," *Comput. Secur.*, vol. 114, Mar. 2022, Art. no. 102582, doi: 10.1016/j.cose.2021.102582.

[33] P. S. Joshi and H. A. Dinesha. (Aug. 1, 2020). *Survey on Identification of Malicious Activities by Monitoring Darknet Access*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9214121

[34] F. Manavi and A. Hamzeh, "A novel approach for ransomware detection based on PE header using graph embedding," *J. Comput. Virol. Hacking Techn.*, vol. 2022, pp. 1–12, Jan. 2022, doi: 10.1007/s11416-021-00414-x.

[35] M. S. Rana and M. A. Shah, "Honeypots in digital economy: An analysis of intrusion detection and prevention," in *Proc. Competitive Advantage Digit. Economy (CADE)*, 2021, pp. 91–98, doi: 10.1049/icp.2021.2415.

[36] M. U. Rana, M. A. Shah, and O. Ellahi. (Sep. 1, 2021). *Malware Persistence and Obfuscation: An Analysis on Concealed Strategies*. IEEE Xplore. Accessed: May 16, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9594197

[37] I. Musa, R. Wulaningrum, O. Dhanny, F. Metandi, A. Kusumah, and C. A. Tumanggor. (Mar. 4, 2022). *Integrated Information System at Travel Samarinda Based on Macro Microsoft Office Excel*. Accessed: May 16, 2022. [Online]. Available: https://www.atlantis-press.com/proceedings/icast-ss-21/125971115

[38] B. Dasović and U. Klanšek, "A review of energy-efficient and sustainable construction scheduling supported with optimization tools," *Energies*, vol. 15, no. 7, p. 2330, Mar. 2022, doi: 10.3390/en15072330.

[39] *Ransomware—What is it & How to Remove it? | Malwarebytes*. Accessed: Jun. 14, 2021. [Online]. Available: https://www.malwarebytes.com/ransomware

[40] I. Kazoleas and P. Karampelas, "A novel malicious remote administration tool using stealth and self-defense techniques," *Int. J. Inf. Secur.*, vol. 21, no. 2, pp. 357–378, Jun. 2021, doi: 10.1007/s10207-021-00559-2.

[41] S. Lee, N.-S. Jho, D. Chung, Y. Kang, and M. Kim, "Rcryptect: Real-time detection of cryptographic function in the user-space filesystem," *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102512, doi: 10.1016/j.cose.2021.102512.

[42] R. Kumar, R. Kela, S. Singh, and R. Trujillo-Rasua, "APT attacks on industrial control systems: A tale of three incidents," *Int. J. Crit. Infrastruct. Protection*, vol. 37, Jul. 2022, Art. no. 100521, doi: 10.1016/j.ijcip.2022.100521.

[43] R. Bharadwaj, A. Bhatia, L. D. Chhibbar, K. Tiwari, and A. Agrawal. (Jan. 1, 2022). *Is This URL Safe: Detection of Malicious URLs Using Global Vector for Word Representation*. IEEE Xplore. Accessed: May 17, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9687204

[44] A. Tarazona. (Jan. 2022). *Desarrollo de Una Macro en Excel Que Sirva Como Soporte a la Validación y Determinación de Incertidumbre Mediante Cromatografía de Gases*. [Online]. Available: http://repositorio.uts.edu.co:8080/xmlui/handle/123456789/8462

[45] H. Park and A. R. Kang, "MS office malicious document detection based on CNN," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 32, no. 2, pp. 439–446, 2022, doi: 10.13089/JKIISC.2022.32.2.439.

[46] M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan. (Jan. 1, 2022). *Ransomware Classification and Detection With Machine Learning Algorithms*. IEEE Xplore. Accessed: May 17, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9720869

[47] O. Ellahi, M. A. Shah, and M. U. Rana. (Sep. 1, 2021). *The Ingenuity of Malware Substitution: Bypassing Next-Generation Antivirus*. IEEE Xplore. Accessed: Apr. 30, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9594221/

[48] E. Hoque, P. Kavehzadeh, and A. Masry, "Chart question answering: State of the art and future directions," 2022, *arXiv:2205.03966*.

[49] *Joesandbox, Automated Malware Analysis—Joe Sandbox Cloud Basic*. Accessed: May 1, 2022. [Online]. Available: https://www.joesandbox.com/analysis/487660/0/lighthtml

[50] A. Alqahtani and F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: An evolving outlook," *Sensors*, vol. 22, no. 5, p. 1837, Feb. 2022, doi: 10.3390/s22051837.

[51] X. Wei, L. Guo, Y. Wang, X. Liu, and J. Deng. (Dec. 1, 2021). *Research and Design of High Performance VPN Security System Based on VPP*. IEEE Xplore. Accessed: May 17, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9674530

[52] J. Zhu, J. Jang-Jaccard, A. Singh, I. Welch, H. Ai-Sahaf, and S. Camtepe, "A few-shot meta-learning based Siamese neural network using entropy features for ransomware classification," *Comput. Secur.*, vol. 117, Jun. 2022, Art. no. 102691, doi: 10.1016/j.cose.2022.102691.

[53] M. Mimura, "Evaluation of printable character-based malicious PE file-detection method," *Internet Things*, vol. 19, Aug. 2022, Art. no. 100521, doi: 10.1016/j.iot.2022.100521.

[54] C. D. Xuan and D. Huong, "A new approach for APT malware detection based on deep graph network for endpoint systems," *Int. J. Speech Technol.*, vol. 52, no. 12, pp. 14005–14024, Sep. 2022, doi: 10.1007/s10489-021-03138-z.

[55] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterprise Inf. Syst.*, vol. 2022, pp. 1–25, Jan. 2022, doi: 10.1080/17517575.2021.2023764.

[56] Z. Mao, Z. Fang, M. Li, and Y. Fan, "EvadeRL: Evading PDF malware classifiers with deep reinforcement learning," *Secur. Commun. Netw.*, vol. 2022, pp. 1–14, Apr. 2022, doi: 10.1155/2022/7218800.

[57] J. Hong, D. Jeong, and S.-W. Kim, "Classifying malicious documents on the basis of plain-text features: Problem, solution, and experiences," *Appl. Sci.*, vol. 12, no. 8, p. 4088, Apr. 2022, doi: 10.3390/app12084088.

[58] I. Gupta, S. Mittal, A. Tiwari, P. Agarwal, and A. K. Singh, "TIDF-DLPM: Term and inverse document frequency based data leakage prevention model," 2022, *arXiv:2203.05367*.

MUHAMMAD USMAN RANA received the master's degree in information security from COMSATS University Islamabad, Pakistan, in 2022. He is currently working as a Security Researcher and an Analyst with the Cyber Security Laboratory, COMSATS University Islamabad. His current research interests include counterintelligence, deception, penetration testing, windows exploitation, SIEM, vulnerability assessment, threat intelligence, phishing, and offensive security.

OSAMA ELLAHI received the master's degree in information security from COMSATS University Islamabad, Pakistan, in 2022. He is currently working as a Security Researcher and an Analyst with the Cyber Security Laboratory, COMSATS University Islamabad. His current research interests include cyber security, offensive security, deception, windows exploitation, vulnerability assessment, threat intelligence, and phishing.

MASOOM ALAM received the Ph.D. degree in computer sciences from the University of Innsbruck, Austria. He is currently an Associate Professor with the Department of Computer Science, COMSATS Institute of Information Technology, Islamabad, Pakistan. His research interests include access control systems, model-driven architecture, and workflow management systems.

JULIAN L. WEBBER (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees from the University of Bristol, U.K., in 1996 and 2004, respectively. He worked with Texas Instruments Europe, from September 1996 to October 1998. He was a Research Fellow with the University of Bristol, from November 2001 to August 2007, and Hokkaido University, Japan, from September 2007 to March 2012. He was a Research Scientist with the Wave Engineering Laboratories, ATR Institute International, Japan, from April 2012 to March 2018. He has been an Assistant Professor with Osaka University, since April 2018, and a Guest Research Scientist with ATR. He is currently an Associate Professor at the Kuwait College of Science and Technology (KCST). His research interests include signal processing, and wireless communication systems design and implementation. He is also a member of IEICE.

ABOLFAZL MEHBODNIYA (Senior Member, IEEE) received the Ph.D. degree from the INRS-EMT, University of Quebec, Montreal, Canada, in 2010. He is currently an Associate Professor and the Head of the ECE Department, Kuwait College of Science and Technology (KCST). Before coming to KCST, he worked as a Marie-Curie Senior Research Fellow at University College Dublin, Ireland, and prior to that, he worked as an Assistant Professor at Tohoku University, Japan, and a Research Scientist with the Advanced Telecommunication Research (ATR) International, Kyoto, Japan. His research interests include the field of communications engineering, the IoT, and artificial intelligence in wireless networks and real world applications. He was a recipient of numerous awards, including JSPS Young Faculty Startup Grant, KDDI Foundation Grant, Japan Radio Communications Society (RCS) Active Researcher Award, European Commission Marie Skodowska-Curie Fellowship, and NSERC Visiting Fellowships in Canadian Government Laboratories. He is a Senior Member of IEICE.

SHAWAL KHAN received the bachelor's degree in computer science from Shaheed Benazir Bhutto University, Upper Dir, Khyber Pakhtunkhwa, Pakistan, and the master's degree in information security from COMSATS University Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Software Engineering and IT, Ecole de technologie superieure, Universite du Quebec, Montreal, Canada. His research interests include access control, cryptography, and network security.