Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

Cipher Plague: Decoding the Enigma of Viral Code and Cyber Threats

Syed Talal Musharraf
Bahria University. Email: syedtalalmusharraf10@gmail.com

Muhammad Hamza Khan
Bahria University. Email: muhammadhamzakhan2601@gmail.com

Salman Asghar
Bahria university. Email: salmanasghar370@gmail.com

Muhammad Zulkifl Hasan
Department of Computer Science, Faculty of Information Technology
University of Central Punjab Lahore Pakistan.
Email: Zulkifl.hasan@ucp.edu.pk

Muhammad Zunnurain Hussain (Corresponding Author)
Dept. of Computer Science, Bahria University Lahore Campus
Email: Zunnurain.bulc@bahria.edu.pk

Muhammad Atif Yaqub
Department of Computer Science
National College of Business Administration and Economics,
Lahore, Pakistan. Email: Atif.yaqub@ue.edu.pk

Abstract
Encrypting viruses are computer viruses that, if not identified, can cause serious problems. For routine activities, the whole world depends on computer frameworks. Encoding viruses are thought to be among the most dangerous viruses because, once infected, they may begin scrambling the basic and secret records in general and documents saved on your PC or PC, leaving the papers ineffective and unclear, or they may be erased, causing information misfortune or a programmed processing plant reset, which can involve the cancellation of all records and the entirety of the essential data. Modern encrypted viruses have emerged as strong enemies in the fast-developing environment of cybersecurity threats, creating significant challenges to computer systems and networks worldwide. This research study examines the spreading techniques of these complex viruses, the potential dangers they offer, and the countermeasures put in place to prevent their destructive influence.
Keywords: Encrypted virus, Malware, Cryptography, Encryption, Decryption, Payload, Signature-based detection, Heuristic detection, Behavioral analysis, Machine learning, Sandboxing, Rootkit, Polymorphic virus, Metamorphic virus, Code obfuscation.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

Introduction

Malware obfuscation is a significant difficulty for code analyzers and antivirus professionals. Malware employs a variety of tactics to conceal itself so that it cannot be seen and to extend the length of its lifespan. Although camouflage tactics cannot completely block the analysis and combat against malware, they do extend the process of analysis and detection, giving the virus more time to spread. Because it is challenging to break the encryption and remove the infection, cybercriminals frequently utilize encrypted ransomware, a sort of virus. Malicious code encrypts the whole contents of your computer and demands a ransom.

Nowadays, the majority of ransomware employs the exceedingly difficultto-decrypt AES-RSA encryption algorithms. In this paper, we aim to research how encrypted viruses work, analyze them, their detection, and possible ways to avoid your computer from them. As encrypted viruses grow and cause chaos, there is an urgent need for indepth study and analysis to understand their fundamental processes, transmission techniques, and the level of harm they may inflict. Furthermore, investigating and finding novel countermeasure tactics, including as behavior analysis, machine learning, and improved detection techniques, is critical to improving the resilience of computer systems and networks against these complex attacks.

Predictive AI-based technologies have been utilized to discern unknown trends in complicated datasets which could serve as a basis for determining how encrypted viruses propagate [23]. The development of advanced methods for distributing dense AI tasks in cloud ecosystems facilitates the large-scale management of virus detection and response processes [24]. The demand for researching and analyzing the activities of encrypted viruses in real time makes the utilization of scalable data lake architectures, which can ingest and store huge amounts of IoT data, essential [25].

Business strategies that incorporate machine learning are evidence of its versatility within the realm of security as it provides means of detecting and countering threats [26]. When combined with AI, Quantum computing presents the possibility of rapidly decrypting modern threats as well as providing unique ways to combat encrypting viruses [27]. The same predictive models used for weather forecasting have been applied to examine the spread of viruses and their impact thereby providing insight into predictive models [28]. The cloud-based data lake house architecture also shows that encrypted virus signatures and corresponding meta-data need to be efficiently managed and analyzed [29].

The revolutionary role AI plays in predictive analytics for healthcare is expected to be of equal importance in developing sophisticated dynamic systems for virus detection capable of “suppressing” enemy systems in advance [30]. Smart grids optimization wiring, for instance, shows an application of AI for automation on the fly

To operate on and subsequently mitigate encrypted viruses in large scale networks or operations that can be extended to detecting such viruses [31]. An operational scenario of such prediction includes neural networks trained on AI-generated data which encompasses AI-Powered business intelligence [32] frameworks. Combining different models together gives the ability to develop robust, data-driven countermeasures against evolving cyber security threats.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

Problem Statement

This research seeks to investigate several forms of encrypted viruses and examine various strategies provided in previously published publications to identify encrypted viruses. The paper revealed a gap in that while much research has been done to give approaches for encrypted virus detection, none of these studies compare different techniques to determine the best one. We'll attempt to discuss the elements of encrypted viruses such that there are clear contrasts between the key detection procedures.

III. SCOPE

Our research will cover all aspects of modern/encrypted viruses i.e., their distribution, threats, and countermeasures. In addition, it will also cover the modern approaches and methodologies followed by the anti-malware programs to identify the virus and mitigate it. Primarily, we'll try:

- To understand the types of encrypted viruses.
- To know how these viruses are transmitted.
- To know the harm and threats these modern viruses impose on the system.
- To know about the methodologies followed by the anti-virus systems for this virus mitigation.

Overview of Report

In each coming chapter, we'll cover a single point from our scope. Chapter 2 provides a description of encrypted viruses; Chapter 3 of this report introduces types of encrypted viruses. Chapter 4 is about the advancement in these viruses. Chapter 5 describes the mutation techniques used by polymorphic viruses. Chapter 6 describes the different harm and threats these modern viruses impose on the system. Chapter 7 provides motivation behind encryption and Chapter 9 comparatively analyzes the Detection Techniques and methodologies followed by the anti-virus systems for this virus mitigation.

Encrypted Virus

Encryption was one of the earliest and simplest techniques used by virus writers to conceal the viral code's operation. An encrypted virus typically has two components: a decryptor and an encrypted main body. When an infected software is run, the decryptor is activated and decrypts the virus body. The malware's main body is its encrypted source code, and until the decryption loop does its work, it is meaningless. The virus's main body must first be converted into machine executable code and useful data by the decryptor loop before it can begin to operate on the host computer. There are numerous methods for encrypting the code. To alter the code byte by byte, for instance, a straightforward encryption can employ a 1 to 1 mapping. A zero-operand instruction, such as INC or NEG, can be employed in a further straightforward encryption method. Additionally, more advanced encryption methods using reversible instructions, such as ADD or XOR with random keys, may be used. The encryption key could also remain the same.

The following figure depicts the general structure of encrypted viruses.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

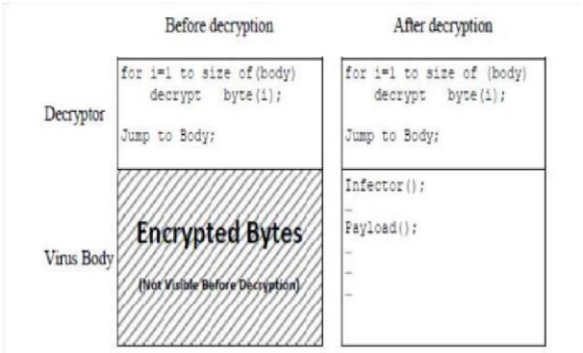

Figure 1 Encrypted Virus

A virus scanner must first decrypt the virus body in order to read the entire code; otherwise, it will not be able to quickly detect the virus using signatures. However, it is able to locate the decrypting portion, and if this portion has sufficient bytes for a string signature, it still makes it possible to detect a virus indirectly using a string signature

Types of Encrypted Virus

- A few types of encrypted viruses are:
- File Encryption Ransomware
- Full Disk Encryption Ransomware
- Cryptoviral extortion Ransomware
- Double extortion Ransomware
- Mobile Ransomware
- Pretoria Virus
- DOS virus Cascade
-

A. File Encryption Ransomware

This type of ransomware encrypts individual files on a computer, making them inaccessible to the user, and demands payment for the decryption key. *B. Full Disk*

B Encryption Ransomware

This type of ransomware encrypts the entire hard drive of a computer, making all files and programs inaccessible, and demands payment for the decryption key.

C. Cryptoviral extortion Ransomware

This type of ransomware encrypts important files and demands payment for the decryption key.

D. Double extortion Ransomware

This type of ransomware not only encrypts files but also exfiltrates sensitive data from the victim's network, and threatens to make it public if ransom is not paid.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

E. Mobile Ransomware

This kind of ransomware mainly targets the mobile devices, such as tabs and Smartphones, and can lock the device or encrypt important files.

F. Pretoria Virus

Each byte was XOR-ed by the virus with a pre-set value, which is analogous to a straightforward substitution process. Since the starting value is produced by using the same key repeatedly, the XOR command is highly useful for viruses. By doing this, virus creators can avoid using two different encryption and decryption techniques.

```
again:
lodsd                   ; get a byte to decrypt
xor al, 0a5h            ; decrypt using key
stosb                   ; and store it back
dec bx                  ; finished ?
jnz again               ; if not, continue
```

Figure 2 PRETORIA VIRUS CODE EXAMPLE

G. DOS virus Cascade

It's a simple algorithm for substitution. It's consists of XORing each byte twice with variable values, the length of the program depends one of them.

```
ea si, Start            ; position to decrypt (dynamically set)
mov sp, 0682            ; length of the encrypted body (1666 bytes)
```

Figure 3 DOS VIRUS CASCADE CODE EXAMPLE

H. Outcomes

The issue with this approach is that by detecting a virus only based on its decryptor, the algorithm is unable to distinguish the virus from variants since multiple viruses with various functionalities may use the same decryptor. By this way, antivirus software wouldn't be able to recover corrupted files ever again and instead would be producing false positives as literally anything which uses a similar decryptor wouldn't have been processed by it either (non-virus is used here: antidebug wrapper).

Advancement in Encrypted Virus

Encrypted viruses have come a long way in recent years, and many stronger variants continue to emerge. Here are some recent examples of improvements in encrypted viruses:

A. AI-powered viruses

Artificially intelligent malware that learns how to bypass security and develops countermeasures against new protection methods.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

B. Cloud-based viruses

This particular type of Malware uses the cloud infrastructure to spread and affect systems, which often makes it difficult to detect these viruses. Such viruses are very hard to contain.

C. Social engineering tactics

that use psychological manipulation to persuade victims into downloading and installing malicious files, usually via emails or deceptive software updates.

D. Encrypted communication channels

Viruses that use encryption to communicate with their command-and control servers, making it harder for security experts to track and intercept their activities.

E. Use of legitimate services

Viruses that use trusted and widely used services such as encrypted messaging apps or cloud storage providers to spread and infect systems.

The drawback of encrypted virus was that it uses the same decryptor to decrypt the encrypted body every time. So, antivirus or malware decryption system can identify the virus based on the signature of the decryptor. So, oligomorphic and polymorphic viruses were created which do not rely on a single decryptor routine.

VIII. Oligomorphism

Antivirus software typically had no trouble with such viruses that were examined and for which a signature string was obtained, despite the fact that virus creators tried to conceal the first generation of viruses with encryption techniques because the decryptor loops were constantly present in new infected files. In order to circumvent this weakness, virus writers used a variety of methods to produce a modified body for decryptors. These initiatives led to the development of oligomorphic viruses, a new class of concealing viruses. Few structures are implied by oligomorphism. It is a Greek noun made up of the term's oligo, which means "a few," and morphe, which means "shape." The drawback of basic encrypted malware is that duplicates of the malware file are created using the same decryptor, which is a disadvantage of oligomorphic malware [11]. Oligomorphism was an try to change the look of the virus decryptor loop with multiple new infections. It was an better encryption technique. It has a variety of decryptors that are each randomly selected for a new victim. Oligomorphism is a property of functions or algorithms that have a limited number of distinct outputs for a given set of inputs. It is used in computer science and mathematics to describe functions that have a limited range of outputs and are thus relatively simple and predictable in their behavior. This property can be useful in the design of efficient algorithms and data structures, as well as in the analysis of complex systems. Oligomorphic functions are typically easy to understand and implement, making them a popular choice for many applications.

As a result, the instances of the decryptor code are different from one another. The virus of Whale DOS, that is initially surfaced in 1990, was the first oligomorphic virus to be identified. In comparison to encrypted viruses, rather than focusing on single decryptors antivirus engines must scan all potential instances of decryptors, which may takes more time. Example of oligomorphic decryptor:

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

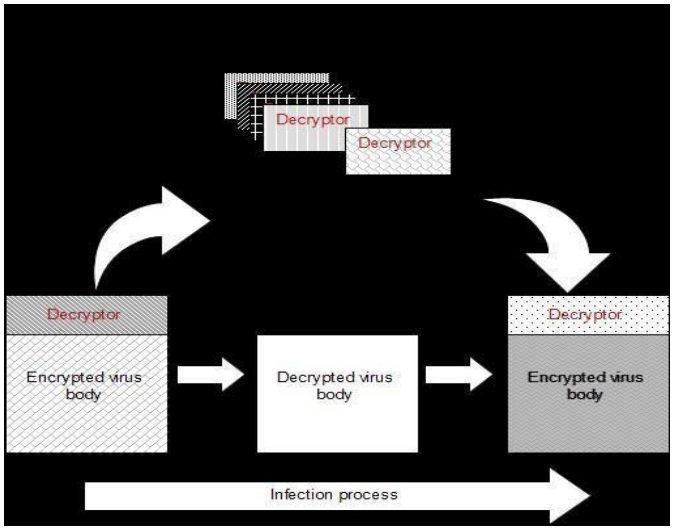

Figure 4 Oligomorphic decrytor

IX. Polymorphism

Polymorphism is an improved/better form of malware that is oligomorphic. In comparisom to Oligomorphic, To create an infinite number of malware versions an infinite number of decryptors needs to be created. As the word poly means “Many” Morphism means forms so using this method we can create different/ multiple forms or variations within a single species. In the field of biology, it refers to the occurrence of different physical traits or genetic variations within a population. In computer science, polymorphism refers to the ability of a single function or method to handle multiple data types or objects. It allows objects of different types to be treated as objects of the same type, promoting code reusability and flexibility.

In terms of code encryption, polymorphic viruses are comparable to encrypted and oligomorphic viruses, but they differ in that they can produce an infinite variety of new decryptors [1]. Mark Washburn created the first polymorphic virus, 1260, a member of the chameleon family that initially surfaced in 1990 [1]. By altering a virus's appearance, polymorphic approaches attempt to make research of it more difficult. The main rule is to constantly change the code's appearance from copy to copy. [12]. It must be done in a way that ensures there are no persistent common strings among virus versions that the antivirus scanner engine may use to identify them. Polymorphic approaches are challenging to use and maintain. [13]. The polymorphic virus uses techniques for coding obfuscation, such as inserting garbage code or swapping out instructions, to change its decryptor and create a new one for each new host it infects. [14] The area in question is known as the "mutation engine" or "obfuscation engine." [15].

Now let’s take a look at some polymorphic viruses:

A. Storm worm

Storm Worm, which was first identified in the wild on January 17th, 2007, was able to gain access to millions of devices all around the world. According to some estimates, this polymorphic worm was responsible for 10% of infections globally. Storm Worm spread through emails that included click-bait headlines like "230 people died after big storm hits Europe," "Naked teen attacked home director," "Saddam Hussein has been discovered in X area," and others. The Storm Worm carrying emails had attachments

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

that could deliver a variety of payloads, including trojans, rootkits, and botnetbuilding tools. The capacity of Storm Worm to switch between payloads and take on different themes, topics, and forms prior to the next infection is what distinguished it as polymorphic malware [22].

B. VIRLOCK

When VIRLOCK is available, why choose the meagre ransomware option? To be clear, the typical ransomware encrypts your file and demands money in exchange for the decryption key. VIRLOCK goes above and above; all encrypted files are transformed into "infector files," enabling them to infect other files wherever they may be. For instance, if you share files online with a friend and VIRLOCK infects your files, your peer's files are also infected. They resemble those virophages from StarCraft II, in a way [22].

C. BAGLE

BAGLE is a great illustration of why we shouldn't discount things that are considered to be "old school." In 2004, the BAGLE worm first surfaced. Although it has never been proven, some claim that it may have Australian roots. The original BEAGLE strain didn't do anything noteworthy; it infected roughly 120,000 people before going away. However, compared to version alpha, Bagle.B, BEAGLE's initial variant, was a little more virulent, exploiting a good number of backdoors and leaving trojans in its path. It's interesting to note that BAGLE 2.0's "half-life" was brief (i.e., a couple of weeks). Nevertheless, BAGLE would emerge and spread like a cockroach after a nuclear explosion, making it difficult for researchers and antivirus software to keep up with the most recent variations. According to some reports, the number of BAGLE versions increased significantly from 35 in 2004 to 100+ in 2005 [22].

X. Mutation Techniques

The most popular obfuscation methods used by polymorphic viruses to modify their code are

- Instruction replacement
- Instruction permutation
- Variable/Register substitution
- Junk /Dead code insertion
- Code transposition

A. Instruction Replacement

With the help of their comparable instructions, this strategy attempts to replace some instructions. A task may occasionally be carried out according to various equal sets of coding instructions. For instance, the register eax was set to zero by each of the instructions that followed [10]:

```
mov   eax, 0
xor   eax, eax
and   eax, 0
sub   eax, eax
```

Figure 5 INSTRUCTION REPLACMENT

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

This talent is used by virus programmers in their virus obfuscation engines. It is comparable to how different synonyms are used in everyday speech [16].

B. Instruction permutation

The programmer can safely reorder the order of instructions in many programmes. Binary sequences of the code are rearranged in such a way that they appear different in each generation. When certain instructions are autonomous, they can be rearranged in a different order without changing the outcome.

Given the following example: op1 R1/Mem1, R2/Mem2 op2 R3/Mem3, R4/Mem4

The above operations can be permutated If these conditions are existed [17]:

- R1/Mem1≠R2/Mem2
- R1/Mem1 ≠ R4/Mem4
- R2/Mem2 ≠ R3/Mem3

The following table displays an example, two columns contain the same result and code can be arranged in both orders, equally [18].

Code Order 1	Code Order 2
mov eax, 0F	add esi, ebx
push ecx	mov eax, 0F
add esi, ebx	push ecx

Figure 6 EXAMPLE

C. Variable/Register Substitution

Exchanges between registers or memory variables in several virus instances is another tactic utilized by mutation engines. By transforming the same bytes into multiple generations, the virus attempts to evade string signature detection. By December 1998, W95.Regswap was one of the first viruses to employ this method to create a variety of virus variations. It obviously affects the binary sequence of the code rather than the behavior of the code. Two forms of W95.Regswap are given in following figure:

```
5A              pop   edx
BF04000000      mov   edi,0004h
8BF5            mov   esi,ebp
B80C000000      mov   eax,000Ch
81C288000000    add   edx,0088h
8B1A            add   ebx,[edx]
899C8618110000  mov   [esi+eax*4+00001118],ebx
```

```
58              pop   eax
BB04000000      mov   ebx, 0004h
8BD5            mov   edx, ebp
BF0C000000      mov   edi, 000ch
81C088000000    add   eax, 0088h
8B30            mov   esi, [eax]
89B4BA18110000  mov   [edx+edi*4+00001118],esi
```

Figure 7 FROMS OF W95 REGSWAP

The code shows that some operations are replaced by their equivalents. "mov ebp, esp" is altered to a sequence of the instructions "push esp" and "pop ebp," which

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

performs a similar operation, in place of the instructions "test esi, esi," "or esi, esi," and "test edi, edi," which provide the same results. These changes obviously alter the binary sequence of the program code. Accordingly, the signatures in the given examples of Win95.Bistro are not identical.

Instruction		Operation
ADD	Reg,0	Reg←Reg+0
MOV	Reg,Reg	Reg←Reg
OR	Reg,0	Reg←Reg | 0
AND	Reg,-1	Reg←Reg&-1

Figure 8 Examples of Win95.Bistro

Version 1:

```
55          push    ebp
8BEC        mov     ebp, esp
8B7608      mov     esi, dword ptr [ebp + 08]
85F6        test    esi, esi
743B        je      401045
8B7E0C      mov     edi, dword ptr [ebp + 0c]
09FF        or      edi, edi
7434        je      401045
31D2        xor     edx, edx
```

Version 2

```
55          push    ebp
54          push    esp
5D          pop     ebp
8B7608      mov     esi, dword ptr [ebp + 08]
09F6        or      esi, esi
743B        je      401045
8B7E0C      mov     edi, dword ptr [ebp + 0c]
85FF        test    edi, edi
7434        je      401045
28D2        sub     edx, edx
```

```
mov     eax, 0
xor     eax, eax
and     eax, 0
sub     eax, eax
```

Figure 9 Assembly Details

As shown in the figure the instructions do not alter the operand register's value, although the y might alter the CPU's flag register's state. For instance, adding a register value to itself or adding a zero to a variable or register has no impact on how the execution will turn out. The second variation of this technique involves inserting an instruction into the code, which may modify the machine's state or the contents in memory or CPU registers, but before it has an impact on the program's result, another piece of code reverses it [21].

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

Instruction		Comments
PUSH	CX	It push value of AX into stack, later it must be turned back to AX before any effects on AX or stack memory
...	...	
POP	CX	
INC	AX	The value of DX increases by 14, and later before any usage of DX, its value must be changed back to its previous value
...	...	
SUB	AX, 1	

Figure 10 EXAMPLE

D. Code Transposition

In this Code transposition it works by rearranging the program's instructions or code flow while maintaining the execution flow via unconditional or conditional branching, this method modifies the program's structure. The transformation can be performed on the single instructions level or a code block [1].

The easiest way to change the binary sequence of a virus program without affecting the functionality or behavior of the code is to inject dead code or garbage code [14]. Different categories of garbage codes exist. A few examples of garbage codes are shown in below.

{

|} THREATS
XI. THREATS

These instructions are equivalent to no operation because they don't alter the contents of CPU registers or memory (NOP). Infiltrating computer networks and systems via spam emails, malicious attachments, or even network flaws. The network user (victim) will be forced to pay a "ransom" for their systems to be released from this mess by being given either a programmer capable of decrypting the encrypted files or an unlock code that undoes the modifications inflicted to the system by the payloads. In the majority of encrypted virus threat instances, paying unlocking "fees" is practically the attackers' desire. Data, files, or system settings that are changed due to installation of malicious software or virus program are examples of threats of encrypted viral. These kinds of viruses enter a system by automating themselves in the programs of desktop or rising from the surface of Operating System. Then, they will change or erase the settings of system and add fake ones that are intended to theft financial and personal information. It is difficult to delete the malicious/infected files and return
the files of system to their original state after it's been infiltrated. IT professionals typically use an encrypted viral threat event response mechanism when they learn that their systems have been compromised. It's essential to keep a live database of infected files with these issues that new infections may be dealt with right away. It may be difficult to transfer confidential data between mobile devices. This is especially valid when dealing with encrypted malware payloads. When using a smartphone or tablet, security is frequently a major concern. The majority of the confidential data packets

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

transmitted by these gadgets will undoubtedly not be encrypted. As a result, there's a chance that the information will go into the wrong hands. As a result, it's difficult to validate that all sensitive data is kept safe at all times.

To ensure that a business can deal with any potentially private data theft from their framework you need to have a safe storage and distribution mechanism. This is especially critical in the face of the threat posed.

Two approaches can do this one is with the use of ADR (Access Database) and the second one is console utility, such as the software for System Restore.

The second approach is better suited to IT specialists who want to restore a backup in real-time, the first method is simpler to use and more easy for inexperienced users. Unlike establishing an internal file server (also known as the "ADR"), installing a backup application for encrypted file stations is fully software dependent.

The ability to execute a recovery scan and restore the damage data is made possible by software solutions, which is when a backup application for an infected file system comes in handy.

With the aid of loading themselves onto computing device apps or crawling up under the running system these viruses also get access to a PC machine. They may then delete or change device settings, replacing them with phony ones designed to theft financial statistics and personal data.

XII. Motivation Of Encryption

Anyone who wants to read the virus code or modify the infected files using code viewers or hexadecimal editors must use encryption to hide the virus body [9]. Encryption is utilized by virus designers for a variety of purposes. Skulason identifies four primary motivations. [10] are:

A. To prevent static code analysis

In order to avoid static code inspection. Static analysis of code entails breaking the code down into its component parts and scanning for questionable instructions or code blocks. The suspicious instruction INT 26H, which writes data to the disc bypassing the file system, is one such instance. While preventing the use of tools for static analysis that search for previously suspicious instructions, encryption can be used to conceal faulty instructions.

B. To prolong the process of dissection

Although encryption makes it more challenging to analyze the virus code, it often only increases analysis time by a few minutes at most. The Whale virus tries to avoid disassembly by focusing the majority of its code on encryption. Armored viruses are occasionally used to describe viruses that are built to prevent disassembly.

C. To prevent tampering

It is more challenging for anyone to alter a virus and produce new variations when encryption is used. To do this, one must first decrypt the virus, make any necessary adjustments, and then re-encrypt it before reassembling it.

D. To evade detection

More advanced viruses use self-modifying encryption, which makes decryptor-based detection challenging because no two copies of the same infection share any searchable strings.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

XIII. HOW ENCRYPTED VIRUS EVADE ANTIVIRUS

The methods used by encrypted viruses to avoid antivirus detection are as follows:

- The virus creator can handle both forward and backward loops and can alter the loop's direction.
- Some viruses, like RDA. Fighters do not save the encryption key within the viral body and must conduct a thorough key search in order to decode themselves. The RDA is another name for this (random decryption algorithm). Such viruses are far more difficult to find.
- Other viruses encrypt their bodies using powerful encryption methods. The IDEA cypher is used by the IDEA family of viruses, but as the viruses also contain the decryption key, the encryption cannot be regarded as secure. However, since the antivirus must reimplement the encryption technique, the removal of such viruses is challenging.
- Some computer worms took advantage of the Microsoft crypto API, which uses a pair of secret or public key that is produced on the fly to encrypt DLLs on the system. The Win32/Crypto and the Win32/Qint@mm are two examples of malware that use the crypto API.
- Some viruses, like Win95/Resur and Win95/Silcer, don't include a decryptor in their code. When the infected application images are loaded into memory, they make the Windows Loader move them. In order to facilitate decryption, the virus injects specific relocations into the photos. By doing this, the viral body is also decrypted when the images are relocated.
- The encryption key was kept separate from the virus body by the Cheeba virus. Only after the virus viewed the filename would it be able to access the payload, which was encrypted.
- Tequila was one of several viruses that employed the decryptors' code functions as a decryption key. The prior method may result in issues if the decryptor's code is altered using a debugger. The Emulators that employ code for the optimization of techniques to run decryptors more quickly may also experience issues as a result of the strategy.
- The randomness of the encryption key is a crucial component for encrypted viruses. Some viruses only produce new keys once a day, whereas others do so each time they infect a new object.To select the seed of random.
- The virus creator has the option to decrypt the code at many places. The most popular method is to locate the encrypted viral body and decrypt the code there. The drawback of this approach is that it depends on the operating system whether the encrypted data can be written to memory. Building the decrypted viral body on the stack is another approach. This eliminates the requirement for writeable encryption data. The virus may also allot RAM for the decrypted code and data as a third approach. The technique of decrypting code has some disadvantages because non-encrypted code needs the allocation of memory before decrypting.

XIV. Virus Detection

Because viruses are a developing technology, new, more difficult-to-detect viruses are being created every day, which is unfortunate for the security sector. In particular, two

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

of the most recent viral variants that try to escape detection are polymorphic and encrypted viruses.
Polymorphic viruses frequently alter themselves (i.e., their coding), leaving little time for spotting. In order to prevent it from matching any known patterns, viruses that are encrypted encrypt their virus code. Additionally, the encryption key can be altered, creating a polymorphic encrypted virus. In addition to becoming more difficult to detect, viruses are also dispersing more quickly. Since first-generation scanners are insufficient to detect encrypted viruses the following secondgeneration scanner techniques are used to detect them.

A. Virus detection Based on Signatures

To create a virus code that operates in any setting, virus programmers invest a lot of effort and money. The infection code was made up of "Signature" programming components, which a security analyst or antivirus vendor may be able to spot. The majority of viruses are detected and contained by means of signatures.

Each class of viruses has its own distinctive signature, which they were based on. The sections that follow provide examples of how to dissect and disassemble viruses to look for patterns and mutations. This would enable the virus to be classified and the required action to be taken [34].

B. Dissection of virus

Knowing the virus code when a virus is detected it would be beneficial for other systems. A portion of the code under analysis is sufficient to identify the virus, regardless of whether the system recognizes its entire pattern. The pattern of hex that represents the detected component might reveal the existence of the virus code [33].

C. Virus Disassembly

Iterative steps are involved in the disassembling process. It entails detecting viruses, figuring out the virus's data component, figuring out the data which cannot be disassembled, and figuring out the instructions of virus. In other words, this process makes the information and instructions of the virus that was detected available.

D. 3 Pattern-checking programs and Virus Mutations

The virus was found by the pattern checker of virus. However, the virus's code may be altered so that the checker cannot find it. the virus code's malicious effects. In order to achieve high performance, any checker must maintain the new generated pattern named as Hex pattern caused by the mutations of virus [2].

E. Virus Detection Based on Heuristic

To find and recognize an encrypted virus infection, these techniques distinguish between the system's normal and abnormal behavior. This enables the prevention of virus attacks' negative effects and the resolution of their issues. Two phases make up the heuristic-based detection method. The balanced system behavior is observed at first, after which the information is recorded to compare later in the event of an attack. The observed differences during the second phase point to a problem with a specific family's encrypted virus.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

In short, heuristic-based techniques are built from three main components Data Collection, Interpretation, and Algorithm in the implementation of the Behavioral detector explanation of these components is as follows:

F. Data Collection

The part of the system that gathers system data, either statically or dynamically, is called data collection. It offers the status of the system in both normal and abnormal states and collects data for the second component to analyze later.

G. Interpretation

Data collected by the Data Collection component is analyzed by the Interpreter component. The collected data is then transformed into an intermediate format for use by the third component.

H. Algorithm

The algorithm which is matching is a component that compares the converted information from the interpretation process with the behaviours that have a signature of the encrypted virus. By referring to the three interconnected parts in the Figure below, the Behavior detector's functionality is demonstrated.

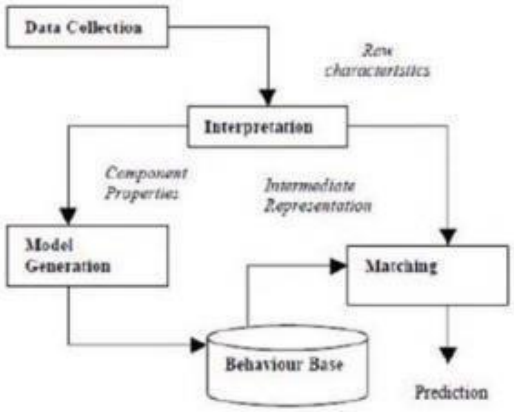

Figure 11 FUNCTIONALITY

The performance of the heuristic-based technique is effective. However, they are still constrained by the available resources. Additionally, it results in a few false positive detected states. Generally speaking, they perform well. The term "Behavior or Anomaly Detection Technique" or
"Proactive Technique" is used to describe it. Before heuristicbased detection, a number of techniques for analysis were used, including file-based, weight-based, rule-based, and generic signature analysis.

I. Virus Detection Based on Specification

A collection of techniques built on specifications is known as t. In fact, they scrutinize and test applications in accordance with the predetermined behaviors of those applications. These strategies are based on heuristic-based strategies. The analysis of behavior that the system specification provides is what the specification-based techniques are based on. In order to identify fraudulent activities of encrypted viruses,

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

machine learning methods are used in tandem with heuristic-based techniques. These methods contrast the typical behaviors of the present states along with the actions specified by the specification of the system. As a result, the resource shortage that existed was overcome. Furthermore, they increase in the level of "true positive" states while decreasing the level of "false positive" states. The below table presents a comparison of the techniques of virus detection in the previous three groups according to their advantages and disadvantages.

Technique group	Advantages	Disadvantages
Signature based	-Known malwares can be detected easily -Used less resources as compared to other techniques	-Unknown malwares cannot be detected.
Heuristic based	-Known and unknown new malware can be detected	-Data need to be updated regarding new and unknown malwares. -Need more resources in terms of time and space -level of false positive is high.
Specification based	-Known, unknown and new malware can be detected -Level of false positive is low	-level of false negative is high -not efficient in detection of new malwares. -specification development is time cousuming

Figure 12 COMPARISON XV. VIRUS ANALYSIS

The analysis of Encrypted viruses is the first step to detecting the encrypted viruses. The step includes the following sub-steps: 1947 Authorized licensed use limited to: California State University Fresno. Downloaded on July 01, 2021, at 14:28:49 UTC from IEEE Xplore. Restrictions apply. 2021 7th International Conference on Advanced Computing
& Communication Systems (ICACCS)

- Performance of Encrypted virus's Function.
- The goal of virus development

High capabilities are provided by these sub-steps in the developed encrypted virus detectors by researchers and antivirus vendors. The information gathered in this step improves the defensive capabilities. Three main groups can be found. The table shown offers a categorization of entities according to the temporal period and analytical method used in the analysis process.

A. Static Analysis

Static analysis is a method for evaluating code or software that does not need running the code or program in its entirety before reaching a conclusion. It is possible to use "passive analysis" as a synonym for "static analysis." In their field of work, computer scientists typically employ this form of analysis. Data mining of non-executable code is often known as "code analysis," and this technique is also credited with creating "Static information." The data will be used to check whether the suspect program has any malicious code that has been encrypted, such as a virus. This approach employs a wide range of various approaches in a variety of combinations to carry out the job of reverse

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

engineering the software efficiently. Deciphering the virus's encrypted code and analyzing its structure is the first step toward understanding the virus's inner workings. One method is to analyze the code's organizational structure. This is interesting because it means the virus's true mechanism of action may be determined. A broad variety of tools, including as disassemblers, decompilers, source code analyzers, and debuggers, are used throughout the process of doing a static analysis. Many supplementary instruments are required for static analysis. Static analysis makes use of methods including antivirus scanning, file format inspection, fingerprinting, string extraction, and disassembly throughout its whole."

B. Dynamic Analysis

Software or code may be analyzed in-process using a technique called dynamic analysis. A dynamic analysis tool is useful for doing this. The term "behavioral analysis" is sometimes used interchangeably with this concept. Behavioral analysis is a term that is used in various civilizations. To do this, it analyzed the program's functions, arguments, and instructions to pinpoint where the function calls and control flow were. It is necessary to utilize a virtual environment to run the fundamental activities of the codes of encrypted viruses. As a result, the habits of the viruses may be studied, and this information can be used toward developing countermeasures. By studying the viruses' patterns of action, scientists may be able to develop countermeasures to reduce the harm they do. To carry out dynamic analysis, a wide range of tools, such as "Sandbox," "Simulator," "Emulators RegShot," and "Process Explorer," are used. Some people dismiss the value of static analysis in favor of its more modern counterpart, dynamic analysis. This step was accomplished after the infected program was given some time to run under a mock OS to reveal the viruses that were encrypting data during that time. In addition, it has a rather straightforward method for distinguishing between many different encrypted viral subtypes. The time and effort required to effectively create the environment (virtual or otherwise) in which the infected software is used and analyzed, as well as the process of carrying out the dynamic analysis, might be substantial.

XVI. ALGORITHMS

It is harder to detect the encrypted virus payloads with standard algorithms used in detection: anti-malware software than other malware types

A. HMM(Hidden Markov Model)

A Hidden Markov Model (HMM) is a model of statistics that is used to explain the evolution of observable events that are depended on internal factors, and are not directly detectable. We refer to the observed event as a "symbol" and the unobservable component as the "state" that underlies the observation.

According to their families, the authors of [35] proposed and evaluated a classifier based on the Hidden Markov Model (HMM) to identify and categorize metamorphic viruses. A classifier is a device that can learn from a system that has trained numerous HHMs. Based on a log-likelihood similarity score obtained from the training, each HMM represented a virus family. A dynamic signature-based method, this one. It was used on a sample of viruses. The outcomes indicated strong performance, roughly 90.86%.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

B. Advantages

The HMM is a well-researched probabilistic graphic model, and algorithms for precise and approximate learning and inference are well known for it.

- The variance of appliance power requirements can be modelled by HMMs using probability distributions.
- HMMs capture the switch continuity principle, which Hart refers to as the dependencies between successive measurements.

C. Disadvantages

Due to the limited number of static distributions used to model appliance behavior, HMMs are unable to accurately represent appliances with continuously varying power demands.

- They don't consider the series of states leading up to any given state because of their Markovian nature.
- Once more, because of their Markovian nature, they do not explicitly record the duration of any given state. The hidden semi-Markov model, however, does indeed detect such behavior.
- Other characteristics besides the observed power demand are not recorded (e.g., time of day). However, the input-output HMM enables the modelling of such state durations.
- Appliance dependencies cannot be modelled in any way. The conditional-HMM can, however, detect these dependencies.

XVII. Counter Measures

- If an encrypted virus is detected, it can take some time for full encryption itself into the network or a system. Before the complication in entire process of detection and elimination, it can be eliminated immediately.
- Use Cyber Hygiene.
- Infected computers must also be disconnected from the network
- Pay attention to the attachments. Email attachments are a common attack method.

Conclusion

Conclusively, this paper discusses introductions to encrypt viruses and what sort of problems and damages they create. It also explores the research areas of how these viruses are transmitted over the network, and what sort of methodologies the antiviruses programs used to detect the virus. Encrypted viruses pose a significant risk to cybersecurity. They may cause severe harm to infected systems and are difficult to identify and eradicate. To identify encrypted viruses, a variety of detection approaches can be utilized, but no single technique is perfect. Using a mix of diverse security methods is the best approach to guard against encrypted malware.

In the initial part of the paper, we described the main types of encrypted virus. These were initial types of viruses that use encryption ways that's why we highlighted advancement in such viruses. In the second part, researchers discussed advancement in these types of viruses. In the last part, researchers discussed defenses from such types of viruses to avoid destruction. Machine learning and artificial intelligence (AI) can be used in a number of ways to detect and prevent encrypted viruses.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

- Signature-based detection
- Behavioral analysis
- Sandboxing
- Machine learning as a service

Here are some of the techniques used by google to prevent from these viruses:

- Google's Deep Instinct
- CrowdStrike's Falcon Prevent
- IBM's Watson for Cyber Security

References

[1] Szor, P., The Art of Computer Virus Research and Defense, AddisonWesley Professional, 2005.
[2] Perriot, F. and P. Ferrie, "Principles and practise of xraying", in 14th Virus Bulletin International Conference (VB2004), 2004, pp. 51–56.
[3] Ferrie, P., "Attacks on Virtual Machine Emulators", in AVAR Conference, 2006, pp. 128 - 143.
[4] Raffetseder, T., C. Kruegel, and E. Kirda, "Detecting system emulators", Information Security, 2007, pp. 1 -18.
[5] M. Sharif, A. Lanzi, J. Giffin et al., “Impeding Malware Analysis using Conditional Code Obfuscation,” in The 15th Annual Network and Distributed System Security Symposium (NDSS 2008), San Diego, CA, (2008).
[6] Al Daoud, Essam, Iqbal H. Jebril, and Belal Zaqaibeh. "Computer virus strategies and detection methods." Int. J. Open Problems Compt. Math 1.2 (2008): 12-20
[7] Nachenberg, Carey. "Computer virus-antivirus coevolution." Communications of the ACM 40.1 (1997): 46-51.
[8] Rad, Babak Bashari, Maslin Masrom, and Suhaimi Ibrahim. "Camouflage in malware: from encryption to metamorphism." International Journal of Computer Science and Network Security 12.8 (2012): 74-83.
[9] Skulason, F., "Virus Encryption Techniques", Virus Bulletin, November 1990, pp. 13-16.
[10] .Johansson, K., COMPUTER VIRUSES: The Technology and Evolution of an Artificial Life Form, 1994
[11] R. Hedayat, The devil’s right hand: An investigation on malwareoriented obfuscation technique. Report, pp no:31-67, 2016.
[12] S. Noreen, S. Murtaza, M. Z. Shafiq et al., “Evolvable malware,” in Proceedings of the 11th Annual conference on Genetic and evolutionary computation, Montreal, Canada, pp. 1569-1576, (2009).
[13] E. Filiol, Computer viruses: from theory to applications, Paris: Springer, (2005).
[14] L. Xufang, P. K. K. Loh, and F. Tan, "Mechanisms of Polymorphic and Metamorphic Viruses," 2011 European Intelligence and Security Informatics Conference (EISIC).
[15] J. Aycock, Computer Viruses and Malware, New York, NY, USA: Springer, (2006).
[16] A. Karnik, S. Goswami, and R. Guha, “Detecting obfuscated viruses using cosine similarity analysis,” AMS 2007: First Asia International Conference on Modelling & Simulation Asia Modelling Symposium, Proceedings, pp. 165-170, (2007).
[17] P. Desai, and M. Stamp, “A highly metamorphic virus generator,” International Journal of Multimedia Intelligence and Security vol. 1, no. 4, pp. 402 - 427, (2010).
[18] B. B. Rad, and M. Masrom, "Metamorphic Virus Variants Classification Using Opcode Frequency Histogram," LATEST TRENDS on COMPUTERS. pp. 147-155, (2010).
[19] I. You, and K. Yim, "Malware Obfuscation Techniques: A Brief

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 2 No. 3 (October) (2024)

Survey," Fifth International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA 2010). pp. 297-300, (2010).

[20] J. M. Borello, and L. Me, "Code obfuscation techniques for metamorphic viruses," Journal in Computer Virology, vol. 4, no. 3, pp. 211-220, (2008).

[21] M. W. Bailey, C. L. Coleman, and J. W. Davidson, "Defense against the dark arts," SIGCSE Bulletin, vol. 40, no. 1, pp. 315-319, (2008).

[22] Unterfingher, "Malware Polymorphism. Polymorphic vs. Oligomorphic vs. Metamorphic malware," Heimdal Security Blog, 25Jun-2021.

[23] Abdullah Al Noman, Md Tanvir Rahman Tarafder, S. M. Tamim Hossain Rimon, Asif Ahamed, Shahriar Ahmed, and Abdullah Al Sakib, "Discoverable Hidden Patterns in Water Quality through AI, LLMs, and Transparent Remote Sensing," *The 17th International Conference on Security of Information and Networks (SIN-2024)*, Sydney, Australia, 2024, pp. 259–264.

[24] S. Nuthalapati and A. Nuthalapati, "Advanced Techniques for Distributing and Timing Artificial Intelligence Based Heavy Tasks in Cloud Ecosystems," *J. Pop. Ther. Clin. Pharm.*, vol. 31, no. 1, pp. 2908–2925, Jan. 2024, doi:10.53555/jptcp.v31i1.6977.

[25] A. Nuthalapati, "Building Scalable Data Lakes For Internet Of Things (IoT) Data Management," *Educational Administration: Theory and Practice*, vol. 29, no. 1, pp. 412–424, Jan. 2023, doi:10.53555/kuey.v29i1.7323.

[26] M. A. Sufian, Z. M. Guria, N. Morshed, S. M. T. H. Rimon, A. I. Mosaddeque, and A. Ahamed, "Leveraging Machine Learning for Strategic Business Gains in the Healthcare Sector," *2024 International Conference on TVET Excellence & Development (ICTeD-2024)*, Melaka, Malaysia, 2024.

[27] A. I. Mosaddeque, Z. M. Guria, N. Morshed, M. A. Sufian, A. Ahamed, and S. M. T. H. Rimon, "Transforming AI and Quantum Computing to Streamline Business Supply Chains in Aerospace and Education," *2024 International Conference on TVET Excellence & Development (ICTeD-2024)*, Melaka, Malaysia, 2024.

[28] S. B. Nuthalapati and A. Nuthalapati, "Accurate Weather Forecasting with Dominant Gradient Boosting Using Machine Learning," *Int. J. Sci. Res. Arch.*, vol. 12, no. 2, pp. 408–422, 2024, doi:10.30574/ijsra.2024.12.2.1246.

[29] A. Nuthalapati, "Architecting Data Lake-Houses in the Cloud: Best Practices and Future Directions," *Int. J. Sci. Res. Arch.*, vol. 12, no. 2, pp. 1902–1909, 2024, doi:10.30574/ijsra.2024.12.2.1466.

[30] M. T. R. Tarafder, M. M. Rahman, N. Ahmed, T.-U. Rahman, Z. Hossain, and A. Ahamed, "Integrating Transformative AI for Next-Level Predictive Analytics in Healthcare," *2024 IEEE Conference on Engineering Informatics (ICEI-2024)*, Melbourne, Australia, 2024.

[31] A. Ahamed, M. T. R. Tarafder, S. M. T. H. Rimon, E. Hasan, and M. A. Amin, "Optimizing Load Forecasting in Smart Grids with AI-Driven Solutions," *2024 IEEE International Conference on Data & Software Engineering (ICoDSE-2024)*, Gorontalo, Indonesia, 2024.

[32] S. M. T. H. Rimon, Mohammad A. Sufian, Zenith M. Guria, Niaz Morshed, Ahmed I. Mosaddeque, and Asif Ahamed, "Impact of AI-Powered Business Intelligence on Smart City Policy-Making and Data-Driven Governance," *International Conference on Green Energy, Computing and Intelligent Technology (GEn-CITy 2024)*, Johor, Malaysia, 2024.

[33] Tahir, R. (2018). A study on malware and malware detection techniques. International Journal of Education and Management Engineering, 8(2), 20

[34] Mohanta, A., & Saldanha, A. Malware Analysis Lab Setup. In Malware Analysis and Detection Engineering ,(pp. 25-50) (2020).

[35] Thunga, S. P., & Neelisetti, R. K. (2015). Identifying metamorphic virus using n-grams and Hidden Markov Model. 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)