International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 3 | ISSN : 2456-3307
DOI : 10.32628/CSEIT195378

The Legendary Era of Computer Virus and Their Detection

Soumen Chakraborty
Department of Information Technology, MCKV Institute of Engineering, MAKAUT, West Bengal, India
Email : csoumen88@gmail.com

ABSTRACT

In this paper, anti-virus professionals layout and develop new methodologies to lead them to more potent, more and more, every day. The cause of this paper is to review these methodologies and description their strengths and weaknesses to inspire those are inquisitive about more research on those regions. In this paper, first examine 4 virus advent kits to determine the degree of metamorphism provided by way of each and capable of precisely quantify the diploma of metamorphism produced by way of these virus mills. While the nice generator, the Next Generation Virus Creation Kit (NGVCK), produces virus editions that range greatly from each other, the alternative 3 turbines examined are tons much less powerful. This paper gives a trendy evaluate on laptop viruses and protective strategies. Computer virus writers commonly use metamorphic strategies to produce viruses that exchange their inner structure on every infection. On the opposite hand, anti-virus technology always follow the virus hints and methodologies to triumph over their threats.
Keywords : Antivirus Techniques, Computer Antivirus, Creation, Defensive, Metamorphism, NGVCK, Virus.

I. INTRODUCTION

In this paper, a single hidden Markov model (HMM) is used to determine whether or not a given program belongs to the virus circle of relatives that the HMM represents. This approach may be used to distinguish family member viruses from non-member packages. The challenges with the HMM. “A computer virus is a application that recursively and explicitly copies a probable evolved model of itself” [1]. A virus copies itself to a bunch document or device area. Once it receives control, it multiplies itself to form more recent generations. Over the past a long time, the range of viruses has been growing swiftly. In 1999, the infamous Melissa virus infected thousands of computers and prompted harm close to $eighty million; while the Code Red malicious program outbreak in 2001 affected structures jogging Windows NT and Windows 2000 server and brought about damage in extra of $2 billion [2]. To simplify the virus introduction process, virus writers have made virus creation kits effortlessly available on the Internet [3].

Technique include locating the proper stability among sensitivity and specificity, and conforming to the time and area constraints of the computer systems performing the detection, and evaluated the effectiveness of this technique by way of its detection price, the false advantageous and false poor costs, and the general accuracy of the classification.

II. METHODS AND MATERIAL

EVOLUTION OF VIRUS

Computer malwares may be classified according to their unique characteristics in several numerous manners, along with class by way of goal or category by way of infection mechanism. One of those type kinds is in keeping with concealment techniques employed [4].

2.1 Virus Obfuscation Techniques

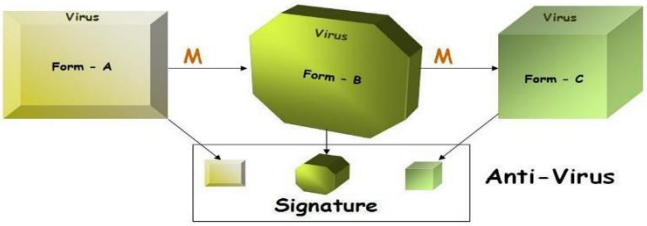
Virus-like packages first regarded on microcomputers inside the Eighties. To undertaking virus scanning products, virus writers continuously expand new obfuscation techniques to make virus code extra hard to stumble on. To escape usual scanning, a deadly disease can modify its code and alters its look on every contamination. The techniques which have been hired to attain this cease variety from encryption to polymorphic strategies, to modern metamorphic techniques [5] [6].

2.1.1 Encrypted Viruses

The simplest manner to exchange the arrival of an epidemic is to apply encryption. An encrypted virus includes a small decrypting module (a decryptor) and an encrypted virus body. If a exceptional encryption key is used for each infection, the encrypted virus frame will appearance distinctive. Typically, the encryption approach is as a substitute simple, consisting of xor of the key with each byte of the virus frame. Simple xor may be very sensible due to the fact xoring the encrypted code with the important thing again will deliver the authentic code and so a virus can use the identical routine for both encryption and decryption. With encryption, the decryptor stays constant from era to technology. As a end result, detection is feasible based on the code pattern of the decryptor. A scanner that cannot decrypt or discover the virus frame without delay can understand the decryptor in most instances.

2.1.2 Polymorphic Viruses

To overcome the hassle of encryption, particularly the fact that the decryptor code is long and precise sufficient for detection, Polymorphic viruses can alternate their decryptors in more moderen generations. They can generate a large wide variety of precise decryptors which use distinctive encryption method to encrypt the virus body. A polymorphic virus consequently has no components that live consistent on every infection. To come across polymorphic viruses, anti-virus software carries a code emulator which emulates the decryption procedure and dynamically decrypts the encrypted virus body. Because all polymorphic viruses bring a steady virus body, detection is still feasible based at the decrypted virus code.



2.1.Three Metamorphic Viruses

To make viruses greater resistant to emulation, virus writers advanced numerous advanced metamorphic strategies. According to Muttik, “Metamorphics are bodypolymorphics”. A metamorphic virus now not most effective adjustments it decryptor on every contamination however additionally its virus body. New virus generations look extraordinary from one another and that they do not decrypt to a regular virus body. A metamorphic virus adjustments its “shape” however not its behavior. This is illustrated diagrammatically by Szor in [7], and is proven in Figure 1. Because all polymorphic viruses carry a regular virus body, detection remains possible primarily based at the decrypted virus code.

2.2 Signature based totally virus detection

Signature based detection structures scan the documents for particular signatures which might be present in them. The pattern of commands found in a pandemic code is identified because the signature of the virus document. This will enhance an alarm for virus if the signature of a deadly disease is detected in any of the files scanned. This technique of intrusion detection is fast and correct for the reason that possibilities of fake alarms are very low in this system. The predominant requirement of the gadget is to have an up to date database of all of the signature files of malware. The accuracy is completely dependent on the signature database of the machine. Signature based detection systems can't detect a brand new virus since the database will now not have any statistics about the new virus. An antivirus scanner extracts the opcode pattern from an executable document and searches the signature database for the enter opcode pattern. The input opcode sample is taken into consideration as the signature of the enter report. If a healthy is determined in the signature database, the input report is classed as the corresponding virus circle of relatives matched within the signature database. For example, if the signature of the input file is 83EB 0274 EB0E 740A 81EB 0301 0000, then this could be searched within the signature database and the report might be categorised as W32/Beast virus given that 83EB 0274 EB0E 740A 81EB 0301 0000 is the signature of the W32/Beast virus [8][9]. A similar search sample used to discover Stoned virus is shown in Figure 2.

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012

2.3 Anomaly based virus detection

Anomaly based detection structures display the strategies on a bunch device for any unusual interest. If any atypical pastime is identified, the system increases an alarm signaling the feasible presence of malware [10]. In this detection technique, the gadget makes use of the collected heuristics to categorize an interest as regular or malicious. Even even though possibilities of fake alarm are rather higher in this method, it's far extra reliable due to the fact it's also capable of detecting new viruses. The essential aspect to be aware is that elevating a false alarm isn't always as capability harmful as allowing a brand new virus. However, these systems can be educated gradually by intruders to do not forget unusual behavior as ordinary. Thus, gadget will fail to locate the bizarre activity in such instances [10].

Three.Three Emulation primarily based detection

The emulation primarily based detection is an effective method where an epidemic is completed in a digital surroundings by emulating the commands inside the virus code. This form of detection is used to stumble on polymorphic, in addition to metamorphic, viruses. The virus example can be finished within the digital surroundings if you want to identify instruction collection or behavior of the virus [8][11]. In addition to the virtual environment, code optimization techniques can be implemented to the execution technique to lower the time for detection. Table 1 lists the electricity and weak point of these detection techniques.

Table 1 : Virus Detection Techniques

Detection Technique	Strength	Weakness
Signature primarily based	Efficient	New Malware
Anomaly based totally	New Malware Costly to enforce	False Positives, Unproven
Emulation based totally	Encrypted Viruses	Costly to Implement

three.4 Use of Machine Learning Techniques

Various researchers have tried to use system getting to know techniques to carry out heuristic evaluation on metamorphic viruses. This phase covers the end result and capacity of some of the strategies, which encompass[12]:

1) Data mining methods
2) Neural networks
3) Hidden Markov models.

2.3.1 Data Mining Approach

Data mining methods are often used to discover styles in a big set of facts. These patterns are then used to become aware of future instances in a similar form of records. Schultz et al. Experimented with a number of facts mining strategies to identify new malicious binaries [13]. They used three mastering algorithms to educate a fixed of classifiers on a few publicly to be had malicious and benign executables. They as compared their algorithms to a conventional signature-based totally approach and stated a better detection rate for each in their algorithms. However, their algorithms additionally led to higher false high-quality costs when as compared to signature-primarily based approach.

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.Three, May 2012

The key to any information mining framework is the extraction of features, which are residences extracted from examples inside the dataset. Schultz et al. Extracted a few static homes of the binaries as functions. These include gadget resource information (the list of DLLs, the list of DLL feature calls, and the wide variety of various function calls within each DLL) received from this system header, and consecutive printable characters located within the files. The maximum informative characteristic they used changed into byte sequences, which had been quick sequences of gadget code instructions generated by way of the hexdump device.

The capabilities were used in three exclusive schooling algorithms. There was an inductive rule-based learner that generated Boolean regulations to analyze what a malicious executable become; a probabilistic approach that carried out Bayes rule to compute the chance of a specific application being malicious, given its set of features; and a multi-classifier gadget that blended the output of other classifiers to offer the most probable prediction.

2.4. Four.2 Neural Networks

Researchers at IBM implemented a neural community for heuristic detection of boot region viruses
[13]. The functions they used were short byte strings, called trigrams, which seem frequently in viral boot sectors but no longer in easy boot sectors. They extracted about 50 functions from a corpus of schooling records, which consisted of each viral and valid boot sectors. Each pattern in the dataset become then represented by using a Boolean vector indicating the presence or absence of those features.

The community changed into single-layered and not using a hidden units. It become skilled using conventional lower back propagation approach. One common trouble with neural network is over fitting, which occurs while a network is trained to identify the education set but fails to generalize to unseen times. To get rid of this trouble, more than one networks have been educated the use of unique capabilities and a balloting scheme changed into used to decide the very last prediction. The neural network became able to discover eighty-eighty five% of viral boot sectors inside the validation set with a false high quality charge of much less than 1%. The neural network classifier has been integrated into the

IBM AntiVirus software program which has diagnosed approximately 75% of latest boot zone viruses since it became launched [13]. A comparable approach changed into later carried out by means of Arnold and Tesauro to correctly locate Win32 viruses [1]. From , we can conclude that neural networks are very effective in detecting viruses closely related to the ones in the schooling set. They also can identify new families of viruses containing similar features as the training samples.

2.5 Hidden Markov Models

Hidden Markov models (HMMs) are properly applicable for statistical sample evaluation. Since their preliminary application to speech recognition problems within the early 1970's , HMMs were implemented to many other regions along with biological collection evaluation [14] . An HMM is a country gadget where the transitions among states have fixed chances. Each nation in an HMM is related to a opportunity distribution for watching a fixed of commentary symbols. We can "train" an HMM to symbolize a hard and fast of records, which is normally in the form of remark sequences. The states within the educated HMM then represent the capabilities of the enter records, even as the transition and the observation possibilities constitute the statistical houses of these functions. Given any statement series, we can fit it in opposition to a skilled HMM to determine the chance of seeing one of these sequence. The opportunity can be high if the series is "comparable" to the schooling sequences. In protein modeling, HMMs are used to version a given own family of proteins . The states correspond to the series of positions in space at the same time as the eighty three

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.Three, May 2012

observations correspond to the possibility distribution of the 20 amino acids that could arise in every role. A version for a protein family assigns excessive probabilities to sequences belonging to that own family. A skilled HMM can then be used to discriminate family participants from non-participants. Metamorphic viruses form households of viruses. Even although participants inside the identical circle of relatives mutate and change their appearances, a few similarities must exist for the variants to keep the identical functionality. Detecting virus versions for this reason reduces to finding approaches to come across those similarities. Hidden Markov fashions provide a method to explain series variations statistically. We advocate to apply HMMs similar to those used in protein series evaluation to version virus families. In virus modeling, the states correspond to the functions of the virus code, at the same time as the observations are instructions or opcodes making up this system. A trained version should then be capable of assign excessive probabilities to and for that reason identify viruses belonging to the equal circle of relatives as the viruses in the education set.

Four. HIDDEN MARKOV MODEL

Hidden Markov Model also known as HMM is a statistical sample analysis device. HMM creates a version representing the enter records. This input records is known as training data. The education records consists of a list of specific symbols and their positional statistics in enter collection. HMM makes use of this model to decide if a given input collection follows similar pattern because the version. HMM is broadly used for speech recognition and protein modeling. Recently HM M has been effectively used to detect metamorphic viruses [15][16]. Metamorphic viruses are a circle of relatives of viruses that adjustments in appearance at the same time as keeping the identical capability. Generally a own family of viruses have similar pattern. Given a circle

of relatives of viruses HMM can provide you with the statistical model representing the circle of relatives [17][18][19]. Now any virus may be examined towards several such models to determine which family it belongs to.

III. RESULTS AND DISCUSSION

4.1 HMM as Virus Detection Tool

HMM as virus detection device calls for training information to supply a model. The schooling records consists of statement series and precise symbols. The commentary collection and specific symbols are derived from numerous viruses of a circle of relatives [20][21][22]. These viruses are packages written in meeting language. The commentary symbols are unique meeting opcodes among all viruses. The opcodes of all viruses are concatenated to supply one lengthy commentary series. HMM is trained in this commentary collection to provide the model. An example of such statement collection is shown in figure three. The model is proven in figure four, and end result proven in Figure 5.

Figure five: The Result File

In the end result report, IDAN0 to IDAN4 are the viruses from the same circle of relatives. The rating for these viruses is greater than -four.38 that are described as a threshold. A report with a rating less than the brink is not taken into consideration as part of this circle of relatives. The documents IDAR0 to IDAR4 have rankings less than the brink and consequently no longer inside the family.

IV. IMPLEMENTATION

In general metamorphic engine has to put in force a few or all code obfuscation techniques. In addition to the usage of these strategies, each implementation may have its very own heuristics. These heuristics may also include approaches that decide kind of obfuscation techniques to use, when to apply them, and the way to apply them. The implementation by following some of the prevailing metamorphic engines like Evol. Evol is a metamorphic virus that used code obfuscation techniques inclusive of dead code insertion, check in / operands utilization exchange, and equivalent eighty five

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.Three, May 2012

training substitution. In addition to the techniques utilized by Evol, we brought few extra variations of those techniques. This segment gives precise explanation of the code obfuscation strategies we used.

4.1 Goals

The implementation has following desires:

1) Generate morphed copies of a single input virus. These morphed copies need to have minimum similarity with the bottom virus and amongst themselves.
2) The morphed copies must have equal functionality as the base virus.
3) Morphed replica need to be close to everyday software.
4) The metamorphic engine should work on any meeting software.

Assumption right here is the ordinary applications are the cygwin utility documents of the same size because the base virus.

The reason in the back of the use of cygwin software documents is they probably are doing equal low level operations as a virulent disease.

4.2. Code Obfuscation Techniques Used

Dead Code Insertion

Dead code insertion is including NOP or do-noting commands. The dead code insertion to introduce opcodes which might be alien to the base virus. The alien opcodes were decided by means of studying the bottom virus and normal programs. First generate information of the bottom virus to discover all of the opcodes used. The graph in discern 6 beneath lists the opcodes used within the base virus with their frequency.

International Journal of Network Security & Its Applications (IJNSA), Vol.Four, No.Three, May 2012

The base virus has 27 unique opcodes and six of them seem extra than 10 times. Opcodes mov, push, upload, call, cmp, and jz are the maximum frequent appearing opcodes. The designed useless opcode set to encompass greater of the infrequent used opcodes. After then examine the everyday software for its opcode frequency. The graph in parent 7 shows the records of a regular report.

When the records of a ordinary file is as compared with the bottom virus, get the listing of opcodes which are precise to a everyday document. The specific opcodes are AND, INT, FNSTCW, OR, FLDCW, LEAVE, JNS, SETNZ, SETZ, JB, CLD, JNB, SHL, INC, FLD, FSTP, and REPE.

This contrast suggests that the above particular opcodes need to be included in morphed copies to make them appearance more like a everyday file. Based in this conclusion the useless code commands are modeled to consist of most of the above precise opcodes. The desk 2 shows a few examples of lifeless code instructions used. Refer to Appendix A for entire listing of lifeless code preparation.

Table 2: Arithmetic Dead Code Instructions

1. Add R,0
2. Sub R,0
3. Adc bx,zero
4. Sbb bx,0

Inc R accompanied with the aid of dec R

These useless code instructions are injected at randomly selected locations inside the base virus. For every decided on location, insert a single useless code instruction. The useless code coaching to be inserted is randomly decided on. These are classified as easy single NOP practise substitution. As the variation to easy single NOP education substitution, we delivered

International Journal of Network Security & Its Applications (IJNSA), Vol.Four, No.Three, May 2012

unconditional bounce NOP education substitution. The jump NOP works through introducing unconditional jump to subsequent immediately practise. An instance of this variation is proven underneath.

```
Mov edx, [esi + entryPoint]      p1010235:
Jmp pl1010235
Mov edx, [esi + entryPoint]
```

five.3.1.1 NOP collection insertion

Dead code insertion is used to insert a single NOP Instruction. In NOP collection insertion, a random series of NOP commands are inserted at randomly selected places. The locations to insert NOP collection have been categorized in viz. Starting of the code phase and rest of the code section. To insert or not to insert a NOP sequence inside the starting of the code section is determined randomly. While for the rest of the code phase, the insertion vicinity and a NOP series is selected randomly. The Algorithm to insert NOP collection on entry point is

1. Determine access point of an epidemic.
2. Generate random range between 0 to 3
3. If the random number is 0 then insert NOP sequence
4. To inset NOP series:

- Randomly pick out duration of a NOP series from three, five, and
- Generate random permutation of the above selected period.
- Insert this sequence into a pandemic.

Transformations of Evol

Along with a unmarried lifeless code insertion and a NOP collection insertion, we added some new dead code insertions. These insertions are stimulated from Evol virus [6], shown in desk 3. Evol virus substitutes a unmarried instruction by means of surrounding it with useless code.

Table three: Evol variations

Original			Transformed	
Mov	r/m,	reg	Push	Randomreg
Mov	reg,	r/m	MOV	Randomreg, OriginalReg
TEST	r/m,	reg	Add	Randomreg, RandomImm8
LEA	r32,	mem	OP	r/m – Randomreg, originalReg
Op	reg,	r/m	POP	Randomreg

One drawback with these transformations is an education is substituted with a block of commands starting with push observed with the aid of a few commands and finishing with pop. Therefore those ameliorations increase the quantity of push and dad opcodes. This also creates a sample of beginning with push and finishing in pop [20].

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.Three, May 2012

five.Three.2 Equivalent guidance substitution

Some opcodes appear often in the base virus like mov, push, add, call, cmp, and jz. To limit the wide variety of those opcodes, we used equal practise substitution. In an equal guidance substitution, an coaching is changed with any other instruction or a block of commands with the identical capability. For instance substitutions for add are indexed in table 4.

Table 4 : substitutions for Add

Add R, imm

1. Sub R, new_imm wherein new_imm = imm x(-1)
2. Lea R, [R+imm]

Add R, l

1. Not R
2. Neg R

Here, opcode upload is changed with opcodes like “sub”, “lea”, and “not” followed by means of “neg”. Similarly opcodes like mov, cmp, test and so forth are changed with equal instructions. The substitution for every preparation is decided based on the type of operands like

REG (8), REG (eight) REG (8), MEM REG (eight), IMM MEM, REG (8) MEM, IMM

REG (16), REG (16)
REG (sixteen), MEM
REG (sixteen), IMM
MEM, REG (sixteen)

REG (32), REG (32)
REG (32), MEM
REG (32), IMM

MEM, REG (32)

Three.3 Transpose

After a morph copy is generated the usage of useless code insertion and equivalent substitution, apply transpose to generate very last output.

V. EXPERIMENTS

Firstly generate a massive of variety of metamorphic virus editions of the bottom virus with new designed metamorphic engine. The metamorphic virus variations have been generated via applying the metamorphic engine iteratively over a single base virus. Applying metamorphic engine as soon as on an enter is 1st generation metamorphism. Applying the metamorphic engine two times on an enter is 2d technology metamorphism and so on. The metamorphic engine can take any meeting application as input. The output is a morphed reproduction of the input. These meeting resources are then complied into executables the usage of FASM [21]. These executables are then disassembled the usage of IDA Pro with default settings (686 education set) [22] [23][24]. These meeting applications have been used to perform all exams. To keep the tests more practical IDA-seasoned generated meeting files have been used instead of the unique assembly source from the engine shown in figure 8. All checks have been finished using two different tools. These encompass Commercial virus scanner, Similarity Test, and statistical sample analysis tool together with Hidden Markov Model [25][26].

5.1 Commercial virus scanner

In our testing, the bottom virus became correctly detected and quarantined by way of the commercial virus scanner installed on our device. But the identical virus scanner failed to come across morphed copies of the base virus[27].

5.2 Similarity Test

Similarity check compares and reviews the share of similarity of two meeting applications. The cause of the similarity take a look at is to measure the code variety of the morphed copies. We compared the base virus with 1st to 9th generations of metamorphic copies. These comparisons have been executed the usage of the default settings of similarity test i.E. 10 opcodes in a chain is considered a healthy. The end result of this test is shown beneath in figure nine. The similarity among the bottom virus and 1st era virus is ready 70%. The similarity decreases with higher generations. 9th era virus is set 10% just like the bottom virus. After applying the metamorphic engine to the bottom virus, the variety of opcodes in morphed copies will increase. The distinctive duration of the in comparison files might also have an effect on similarity take a look at. So we compared a pair of viruses from the equal era. The viruses from the identical technology are of comparable length. 1st technology viruses are approximately 50% similar whereas ninth generation viruses are about 2.5% comparable as proven in figure 10. Note that, the viruses generated by Next Generation Virus Creation Kit (NGVCK) had been found to be approximately 10% similar with default settings [2]. Based on those similarity exams, determine to version HMM on exceptionally assorted generation that is ninth technology.

5.3. The Base virus in opposition to the morphed virus model

After then model HMM for bizarre generations of viruses. The base virus was examined towards those modes and rankings are listed in desk five. Results indicates the statistical pattern of the bottom virus can nevertheless be detected by way of exceptional technology of viruses shown in determine eleven.

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012

Table 5: The base virus examined in opposition to N Generation Model

Model	Score
1st Generation Model	-2.26519095918038
3rd Generation Model	-2.5616088296304
fifth Generation Model	-2.7804691006756
seventh Generation Model	-6.53547571903687
9th Generation Model	-9.36420192759975

Three.4 Morphed viruses against normal document model

The accumulated forty cygwin documents as a hard and fast of normal documents. The generated HMM version on a set of everyday files. Then ninth generation viruses are examined towards this model. The threshold for everyday files is -a hundred and eighty.5254.All 9th generation viruses scored better than the threshold. The most rating of ninth era viruses is -37.2978. So the ninth technology viruses are taken into consideration as ordinary documents. This is 100% fake positives, result display in discern 12 and thirteen.

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012

HMM model of normal documents has very low threshold. The motive behind this low threshold is much less similarity inside a hard and fast of ordinary files. With much less similarity, producing maximum probable version is hard. And this is inflicting fake positives.

VI.CONCLUSION

The advanced the metamorphic engine producing morphed copies of the base virus which might be rather varied and includes a few opcodes of the everyday program. These have been the two important criteria described in which can be required in metamorphic virus to defeat HMM. In our new engine, employ code obfuscation techniques consisting of equivalent preparation substitution, lifeless code insertion, and transpose. This Paper introduce floating factor opcodes in morphed copies which are typically discovered in regular packages. The similarity showed that the morphed copies are quite metamorphic with 2.Five% similarity index. Even with the sort of excessive metamorphism, HMM changed into able to classify the morphed copies of the base virus as the own family virus. The base virus became as compared with version of morphed copies, HMM become nonetheless capable of classify the bottom virus because the same own family. This fact proves that inspite of high metamorphism, HMM is able to become aware of a not unusual statistical pattern across all morphed copies and the base virus. HMM has proved very difficult to defeat.

VII. FUTURE WORK

This skilled our fashions on disassembled virus executables. The disassembling technique can make an effort and the effects rely upon the nice of the disassembler. To speed up virus pre-processing and to cast off the reliance on a specific disassembler, should attempt to teach the HMMs immediately at the binary code of the viruses. Other system gaining knowledge of techniques, together with records mining or neural networks, may additionally paintings at once on the binaries. Training on raw executable byte sequences is extra challenging as these byte sequences are longer and contain extra inappropriate parts. To more very well evaluate the

overall performance of the HMM technique, it'd be beneficial to check on a bigger set of virus editions and also take a look at on exceptional styles of viruses. Ideally, would really like to find viruses which can be much like regular applications to a degree that the similarity index by myself can not distinguish the viruses from.

International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012

Regular code. Only with such facts can we compare the effectiveness of the HMM technique to detecting metamorphic viruses. However, it seems that no metamorphic kit available these days is capable of generating such difficult viral code.

VIII. REFERENCES

[1]. Leonard Adleman. An abstract theory of computer viruses. In Lecture Notes in Computer Science, vol 403. Springer-Verlag, 1990.

[2]. M. Stamp, "Information Security: Principles and Practice," August 2005.

[3]. Fred Cohen. Computer Viruses. PhD thesis, University of Southern California, 1985.

[4]. Peter J. Denning, editor. Computers Under Attack: Intruders, Worms and Viruses. ACMPress (Addison-Wesley), 1990.

[5]. Christopher V. Feudo. The Computer VirusDesk Reference. Business One Irwin, Homewood, IL, 1992.

[6]. Harold Joseph Highland, editor. Computer Virus Handbook. Elsevier Advanced Technology, 1990.

[7]. J. Aycock, "Computer Viruses and malware," Springer Science+Business Media, 2006.

[8]. E. Daoud and I. Jebril, "Computer Virus Strategies and Detection Methods," Int. J. Open Problems Compt. Math., Vol. 1, No. 2, September 2008. Http://www.Emis.De/journals/IJOPCM/documents/IJOPCM(vol.1.2.3.S.08).Pdf

[9]. Lance J.Hoffman, editor. Rogue Programs:Viruses,Worms, and Trojan Horses. VanNostrand Reinhold, New York, NY, 1990.

[10]. Jan Hruska. Computer Viruses and Anti-VirusWarfare. Ellis Horwood, Chichester, England, 1990.

[11]. Filiol, E., G. Jacob, M.L. Liard, 2007. Evaluation technique and theoretical version for antiviral behavioral detection strategies. J. Comput. Virol., three(1): 27-37

[12]. Ye, Y., D. Wang, T. Li and D. Ye, 2008. An sensible pe-malware detection device primarily based on association mining. In Journal in Computer Virology,

[13]. Zakorzhevsky, 2011. Monthly Malware Statistics. Available from : http://www.Securelist.Com/en/evaluation/204792182/Monthly_Malware_Statistics_June_ Accessed 2 July.

[14]. W. Wong, "Analysis and Detection of Metamorphic Computer Viruses," Master's thesis, San Jose State University, 2006. Http://www.Cs.Sjsu.Edu/college/stamp/college students/Report.Pdf

[15]. Eugene H. Spafford. Computer viruses. In John Marciniak, editor, Encyclopedia of Software Engineering. JohnWiley & Sons, 1994.

[16]. S. Attaluri, "Profile hidden Markov models for metamorphic virus evaluation," Master's thesis, San Jose State University, 2007. Http://www.Cs.Sjsu.Edu/college/stamp/college students/Srilatha_cs298Report.Pdf

[17]. P. Szor, "The Art of Computer Virus Defense and Research," Symantec Press 2005. 18] VX Heavens, http://vx.Netlux.Org/International Journal of Network Security & Its Applications (IJNSA), Vol.Four, No.Three, May 2012

[18]. Orr, “The viral Darwinism of W32.Evol: An in-depth analysis of a metamorphic engine,” 2006. Http://www.Antilife.Org/documents/Evol.Pdf

[19]. E. Konstantinou, “Metamorphic Virus: Analysis and Detection,” January 2008. 21] A. Venkatesan, “Code Obfuscation and Metamoprhic Virus Detection,” Master’s thesis, San Jose State University, 2008. Http://www.Cs.Sjsu.Edu/college/stamp/students/ashwini_venkatesan_cs298report.Doc

[20]. The Mental Driller, “Metamorphism in practice or How I made MetaPHOR and what I've learnt,” February 2002. Http://vx.Netlux.Org/lib/vmd01.Html

[21]. M. Stamp, “A Revealing Introduction to Hidden Markov Models”, January 2004. Http://www.Cs.Sjsu.Edu/school/stamp/RUA/HMM.Pdf

[22]. Walenstein, R. Mathur, M. Chouchane R. Chouchane, and A. Lakhotia, “The design area of metamorphic malware,” In Proceedings of the second International Conference on Information Warfare, March 2007.

[23]. “Benny/29A", Theme: metamorphism, http://www.Vx.Netlux.Org/lib/static/vdat/epmetam2.Htm

[24]. J. Borello and L. Me, “Code Obfuscation Techniques for Metamorphic Viruses”, Feb 2008, http://www.Springerlink.Com/content/233883w3r2652537

[25]. A. Lakhotia, “Are metamorphic viruses in reality invincible?” Virus Bulletin, December 2005.

[26]. JDB. EstiHMM: een green algoritme ter bepaling van de maximale sequenties in een imprecies hidden Markovmodel. Master Thesis at Ghent University. Supervised by GdC. June 2011.

[27]. JDB & GdC. State sequence prediction in vague hidden Markov models. ISIPTA 'eleven: pp. 159-168. July 2011.

Cite this article as :

Soumen Chakraborty, "The Legendary Era of Computer Virus and Their Detection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5, Issue 3, pp.256-267, May-June-2019.
Journal URL : http://ijsrcseit.com/CSEIT195378