Detecting Computer Viruses

Manju Khari, Chetna Bajaj

Abstract— **Virus (in biology) refers to microorganism, means very small creatures. Likewise, Virus is a small sized program in computer world but can cause severe destructive actions to computer(s). Virus can result in poor performance, loss of data, loopholes in system. Because of it, everyone who uses computers have a fear of losing data. Resultant, virus developer becomes powerful by holding an exotic power to attack. High losses become an important concern for sober computer users. To restrict them, virus must be detected and observed to know how they produce severe destruction. Various virus detection techniques, which are beneficial in virus prevention, are described in paper. They are also often used in development of antivirus systems to automate virus detection, resulting in a security mechanism which has power to restrict virus i.e. does not allow virus to infect other files on system.**

Index Terms—**Malware, Virus, Virus Detection Techniques.**

I. Introduction

Malware is a program code which has an extensive capability to infect system. It can perform malicious actions on system as well as on network(s) of systems. It can propagate over the network. It can produce severe effects which lead to attack. It can also perform activities that facilitate malware for its operation i.e. disables the installed malware detector [1] [2]. Malware has grown rapidly in past few years and most of them come from well known websites. Malwares can be classified into four generations based on their payload, enabling vulnerability and propagation mechanism [2] [3]. Table 1 shows Generations of malware with, its example and propagation method.

TABLE I. Generations Of Malware, It's Example With Propagation Method

Generations	Example	Propagation Method
First (1980)	Virus	Replicate and propagate by human's intervention, emails and file sharing.
Second (1995)	Worms and Trojan horse	Hybrid in nature. Propagates itself on network, does not require human intervention.
Third (2000)	Specific to organization or geographical region	It employs multiple attack vectors and attacks on security technologies and products.
Fourth (2000 onwards)	Specific to particular application and websites that are used massively	Employs infection through already infected computers and by installing malicious programs to infect application like Facebook, Skype and Operating System like Linux.

Organization of the paper is as follows: We have described various types of malwares, their basic characteristics with detailed description and examples, in section 2. Concentrating on virus, in section 3, Virus's evolution in different phases, is illustrated. Each phase has its own characteristics and consequently, virus has evolved into more complex form to prevent its detection. Due to destructive action of virus, its detection has become an important vertical in computer world. So, various detection techniques are outlined in section 4. Section 5 through some light on analysis techniques like Static Analysis, Dynamic Analysis and Hybrid Analysis, and also describes the relationship among analysis techniques and detection techniques. Some more and efficient virus detection techniques which use machine learning technique are described in section 6. At the end, we have concluded the paper with its future scope discussion continued with references list.

II. Types of Malwares

Different types of malwares are diagrammed in Figure 1:

A. Virus

Virus is a computer program which replicates itself byte-by-byte to create a new executable file, and infect files on system. It is a self replicating program and need some host

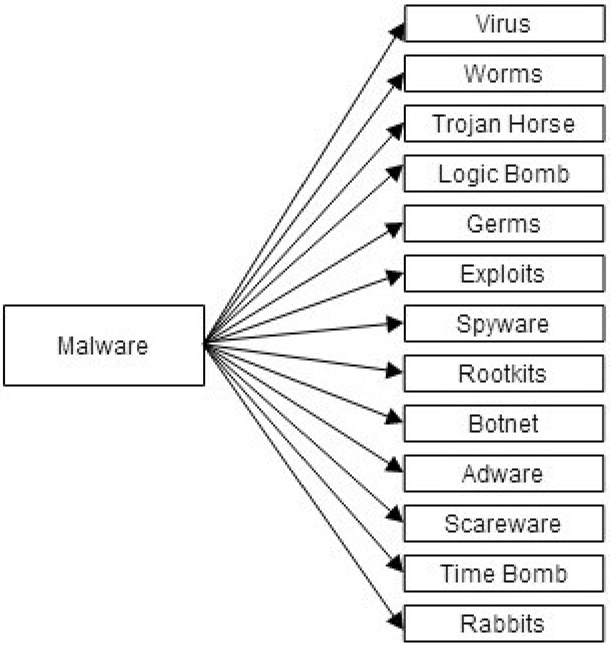

Fig. 1. Various types of Malwares

Manuscript received July, 2014.

Manju Khari, *Department of Computer Science & Engineering, Ambedkar Institute of Advanced Communication Technologies and Research, Delhi, India, Mobile No. 9899296110,*
Chetna Bajaj, *Department of Computer Science & Engineering, Ambedkar Institute of Advanced Communication Technologies and Research, Delhi, India, Mobile No. 9213730027,*

or human intervention for its propagation. Virus gets attach to host program to be able to replicate. Its main target is to infect files on system, which ensures their presence on system for long time and then these files propagates to other computer through some communication medium i.e. as an email attachment, trade program or diskette, coping files to/from server and then again replicates and propagates in same fashion on other machines as well [4].

Example: ILOVEYOU, Melissa, Christmas Tree.

B. *Worms*

Worms are self replicating and self propagating computer program. It mainly communicates to others systems on network. Its main target is to infect systems on the network unlike a virus, which wants to infect files on system [5] [6].

Example: Blaster, Morris, Code Red I and II.

C. *Trojan horse*

Trojan horse is a software program. It appears legitimate, but can easily create backdoor to application which helps to gain control of the entire system remotely without any knowledge and permission of the legitimate user [7].

Example: Netbus, Back Orifice, SubSeven and AIDS TROJAN DISK, Nuker.

D. *Logic Bomb*

Logic Bomb is developed for a specific legitimate application. It is inserted or linked with the application in such a way that user will not be able to observe or locate its presence. Means, it remains hidden but whenever application executes, user can observe its damaging effects. Logic bomb has a flaw that it doesn't replicate on other application. It only works for a application for which it has been designed [7].

Example: Jerusalem, SOBIG worm, Michelangelo.

E. *Germs*

Germs are considered as the first generation virus. It does not have any host program. It exists in its own form when it gets compiled for the first time. It has problem that it does not generate prominent effect of infection process. It infects file but does not leave any sign or mark of infection on that file resulting virus again start to infect already infected file which consumes resources that can be used to infect uninfected files. This problem gets overcome in second generation virus which leaves a flag on infected files to avoid their reinfection [3] [7].

Example: Germ was written for book sector or the diskette in the form of dropper.

F. *Exploits*

Exploits occur due to flaws or vulnerabilities. Vulnerabilities are exploited to create security hole to get into system. Attacker executes a program to locate system's problems and that can be used to make a hidden path to access important and confidential data and operations. Exploits can be executed remotely or by connecting machine to network [7].

Example: 'White hat' hacker's Exploit code is used for application's penetration testing. Exploits can also be developed to execute command prompt (cmd.exe on Windows or /bin/sh on Unix) on remote system.

G. *Spyware*

Spyware gets installed on system without the knowledge and permission of legitimate user. It usually installed as a background process. It collects user's personal and confidential information and uses it against him [2]. It can cause monetary losses as well by accessing and using user's bank details.

Example: Spyware Quake, Spyware cleaner, Security toolbar.

H. *Rootkits*

Rootkits usually get installed on already infected machines (by virus). It is used to gain system's administrator access by compromising it. It is named after term root in UNIX [2].

Example: Adore, svchost.exe, PDCOMP, Hacker Defender, Knark, LRK5.

I. *Botnet*

Botnet is an autonomous software program, remotely controlled by attacker. It is usually a zombie program which can be controlled for any network [2].

Example: Boatnet, Zer0n3t, Kelihos, LowSec, Rbot.

J. *Adware*

It is used in context of advertisements. It is advertising supported software. Its functionality is to display and download advertisements to an already infected system and through these advertisements, other malicious or controlling program can easily be executed on system to acquire its full access [2].

Example: DeskAd, Flipopia, iAdware for MAC, AdBlaster, Clickbank.

K. *Scareware*

Scareware usually comes unknowingly from unreliable internet sources like malicious or hacked websites and applications, or hacker's trap. Innocent users download some programs that appear useful as they primarily claim to provide security to system, resulting in producing vulnerable results. A free or trial version of antivirus program or any free online scam can be considered as scareware program. User gets attracted toward exciting offers and downloads false security software. When user opens attachment, malicious code gets executed and makes system prone to attacks. Some scarewares even try to collect E-commerce related information like bank details, personal information and can cause great losses [8].

Example: Smart Fortress, Android Defender. Ramsomware, System Security (blue screen of death).

L. *Time Bomb*

Time Bomb is a malware, similar to logic bomb. The time in its name specifies the property of occurrence of malware i.e. the particular event or time at which malware will execute to reflect its destructive effects on system and network. Time bombs are programmed to get executed at scheduled time or when some specific condition is met, till the time it remains dormant or inactive [9]. Time bomb like Jerusalem virus i.e. Friday the 13th or April fool's Day, deletes files on infected computer, on every Friday the 13th. Time bomb can be used by an insider, who wishes to destroy the data of organization upon his termination. The employee will set the time of time bomb and it will burst out after his termination. For example, employee may insert malicious code in payroll system and program it to delete all files if in case his name is not present in list. It means, all important files will get deleted if employee gets removed or terminated from organization's payroll system. An organization named Omega Engineering has lost millions of dollars due to a time bomb installed by a former employee.

Examples: The Christmas or Valentine's Day, Conficker.

M. *Rabbits*

Rabbits appeared around 1974. It remains on network but caters itself from one point to other. Rabbits replicate and propagate itself, but deletes its original copy keeping only one copy on network. The feature of replication refers to fork (creating its own copy). So, it is also called as Fork Bombs. Rabbits, being a malicious program, do not cause significant destruction but execute itself recursively and infinitely on system without any restriction. It gradually occupies system's whole memory with multiple copies of recursively executed same program code and slow down the processing thus increase CPU computing time [10]. In short, they monopolize CPU, memory or disk space and can even result in DOS attack [4]. Resultant, it restricts execution of programs which require large amount of resources like processing time and memory. Rabbit worm is effective in evading its detection through antivirus resulting in difficulty in its removal from system.

Example: Fork Bombs, cmd.exe, rabbits written in Perl, JavaScript.

III. Evolution of Virus

Virus has various variant according to their structure and impact on system [1] [3]. Origin of Virus is categorized into:

A. *Initial Stage*

In initial stage, virus was developed to infect system's files and data. It was developed for corrupting disk, email accounts, private networks, etc. It infects systems, and spread across others. Initial stage viruses can be easily detected by antivirus i.e. by using their signatures or pattern.

B. *Stealth Virus*

As earlier viruses are easy to detect, virus developers coded stealth virus to elude virus detection system. Stealth virus takes the charge of file management system and conceals the changes it has made to infected files. It can corrupt and encrypt data files. Stealth virus resides in memory and remains hidden from virus detection system. Whenever antivirus scans system for malicious programs, it automatically renders the search for non-malicious data and antivirus fails to detect it. To detect stealth virus, detection system should scan active memory, compact disc or floppy disk.

Example: Brain, Frodo and Whale.

C. *Polymorphic Virus*

Polymorphic virus challenges virus detection system. It uses a polymorphic generator to mutate code without changing original program or algorithm. The simplest method for mutation is self-encryption. It encrypts virus's body and hides its signature, and makes it difficult for detection. Each time, a different key is used in self-encryption process to produce different viruses. After propagation, polymorphic virus decrypts itself by its own decryption algorithm attach to it and virus get executed on host. To further restrict its detection, polymorphic virus goes through multiple encryption-decryption process. Polymorphic virus can be detected using Emulation based detection system [2].

Example: 1260 was the first polymorphic virus, used different encryption and decryption process to evade detection, HPS, Marburg.

D. *Metamorphic virus*

Metamorphic virus modifies or manipulates itself during propagation across network. It changes its code or internal structure and generates a new virus which functions as original, but does not appear same i.e. signatures do not match [3] [11]. After propagation, metamorphic virus resumes its original structure and again modifies itself to a new virus to evade its detection [2]. Modification to the virus's program code can be done by Code obfuscation techniques like swapping interchangeable instructions, inserting garbage instruction and introducing conditional jumps to generate new virus. Code obfuscation may also include change in instruction order without altering control flow. Code obfuscation technique breaks virus's signature and make it difficult for signature based detectors to detect metamorphic virus. Metamorphic virus translates its new code according to corresponding host's operating system. Other techniques which can be used to create metamorphic virus are Expansion, Permutation, Assembling, Transformation, Disassembly-Depermutation/Shrinking [2].

Along with code obfuscation, packing also restrict malware detection. It encrypts and compresses program code. Packed code or executable is difficult to analyze for detection. So, packed code must be unpacked to get malware's actual code,

By breaking virus's signature by ordering virus's code instructions, signature based detector would not be able to detect metamorphic viruses. So, these viruses must be detected by a system which can extract essential instructions that resemble virus's presence. Therefore, machine learning technique can be used to detect them.

IV. Virus Detection Techniques

Virus detection is important to rescue system(s) or network(s) from their adverse effect. Detection comes in mind when a normal user looks for virus's erroneous effect, which could be very destructive ranging from corruption of files to destruction of whole network. To protect system from virus, it is important to detect them and void their effect. For detection, users may look for virus's execution behavior. User may also observe program's correctness to identify it as benign or malicious activity. Detection method must be able to understand obfuscation technique applied to code and identify malicious code from obfuscated code.

Different detection techniques are used to detect different viruses (with varied characteristics) as a single detection technique would not be able to detect all types of viruses [3]. Antivirus (software program used to detect malwares/virus) incorporates a combination of different detection techniques so that it wouldn't miss any virus being detected. To elaborate further about detection techniques, some of them are explained:

A. Signature based detection technique

Signature based detection technique is a traditional, simple and efficient method for detecting viruses. This technique scans files for virus's signatures. Signature resembles the instruction pattern or a specific string of byte from virus's program code [3]. Signatures can be extracted by examining disassembled code of malware. Antivirus developer must look virus program code carefully to extract virus signature pattern. If there's new virus comes, whose signature pattern is not available then developer must analyze infected file to find its signature. After collecting signatures of viruses, a signature database is created which facilities in virus detection. During detection process, whole computer system including executable files, boot records, document files etc. are checked for viruses. Matching of database's signatures with any file's signature resembles virus's presence [3]. After identifying virus, signature based detection procedure raises alarm for files immediate remedy. Figure 2 describes diagrammatic review of Signature based detection technique.

Signature based detection technique is used for the detection of known viruses, which has been identified earlier i.e. their pattern or signatures are already captured in database. The technique requires signature database to be regularly updated for better results. Higher the signatures present in database, more will be the virus detection rate. This technique produces fast and accurate results. That's why, this technique is used in almost every antivirus system [12].

But problem come when it is required to detect a new virus, which become next to impossible for signature based detection technique as its database does not have any signature for a new virus. As a solution, other methods can be used, tools like disassemblers and debuggers are available in market which can help in extracting signatures from program code. The code gets disassembled, analyzed and then signature extraction is performed [2].

B. Anomaly based detection

Anomaly based detection technique covers problems related to signature based detection technique. It can detect unknown viruses which make it more reliable than signature based detectors [12]. It monitors processes on host machine and looks for abnormal activities. It considers rules or heuristics, and most importantly, learning in training period to differentiate between normal or abnormal activities. Upon finding any abnormal activity on machine, it blocks the activity and raises alarm indicating virus's presence. With advantages, it has disadvantage of high rate of raising false alarms and of being costly.

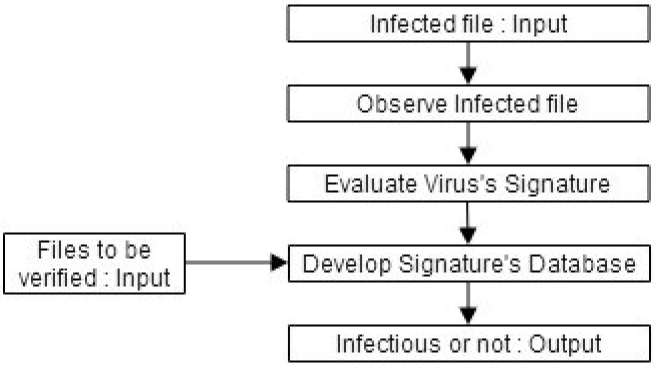
Infected file : Input
Observe Infected file
Evaluate Virus's Signature
Files to be verified : Input
Develop Signature's Database
Infectious or not : Output

Fig. 2. Diagrammatic review of Signature based detection technique.

C. Emulation based detection technique

Emulation based detection technique an effective way of virus detection. In this technique, a virtual environment is set, and virus is executed by emulating virus's instructions. During virus execution, instruction sequence and its behavior are identified. Along with virtual environment, code optimization technique can be used to optimize detection time [11].

D. Generic Signature scanning

Generic Signature scanning method is used to detect viruses belonging to same family. It is also known as heuristic signature scanning. It is possible to have a little difference in signatures of one virus family. Signature based detection does not produce good results in detecting similar signature viruses, but Generic Signature scanning provides. It uses patterns, wildcards to detect different viruses or variants from one family. Variants are formed with some minor changes performed in actual source code of virus program. Moreover, It is very fast method and also capable of detecting new and upcoming future viruses belonging to same family [3] [13].

E. Integrity checking

Integrity checking is a simple, efficient and time saving detection technique. Basic approach of the technique is hash value comparison i.e. comparison of hash value of infected file to that of an uninfected file. If there is not any difference in both hash values then there's no virus in system files [13].

Integrity checker can also use checksum/snapshot/ fingerprints of files as well apart from hash value. File's hash is computed when it is uninfected and stored in database or file to keep it safe. Integrity checker computes hash value of all important files like executable, boot records, system files, etc.

While scanning for virus in system, Integrity checker again calculate hash. The new calculated hash must match earlier calculated hash, stored in database to assure that the file does not get infected. It facilitates detector by not investigating those files whose hash matches and to put its efforts to detect virus in rest of the files. This way, Integrity checking restricts the virus detection space, thus saves measurable amount of time.

F. Heuristic scanning

Heuristic scanning is used to detect new/unknown viruses. It examines application program code and looks for particular commands to relate it with malicious programs. The process scans various programs i.e. boot record, macro files, executable files for virus-like instructions. Virus-like instructions could be malicious program's payload, worm propagation routine, virus replication routine, etc. If scanning detects virus-like instructions, then heuristic scanner informs users about it.

Heuristic scanning can be of two forms: Static scanning and Dynamic scanning. The former includes the scanning of program code and later includes the execution of program code on emulators to locate virus's presence. Heuristic scanning can be done by three techniques which are illustrated in Table 2:

- Code Anomaly Detection
- Protocol Anomaly Detection
- Mixed Heuristic Detection

G. Behavior based detection technique

Behavior based virus detection technique observes behavior of known and unknown malwares. It checks source and destination from/to malware is propagating, attachment's type and size, etc. On observing, it can also predict that a particular source will send only malicious codes to attack the system.

V. Analysis Techniques

Paper [2] and [8] have categorized detection technique in three categories and figure 3 describes relationship between detection and analysis techniques. Categories are:

- Static Analysis
- Dynamic Analysis
- Hybrid Analysis

A. Static Analysis

Static analysis is a process of analyzing program's source code. It does not execute program code. It reveals program structure and can locate executable and non-executable parts of program. It uses disassembler tool for acquiring disassembled and decompiled code to extract low level information from program code. It can also use program analyzer and debugger. Static analyzers must possess good knowledge of assembly language and Operating system's working. It can also use some other tools like program analyzer, disassemble, debugger. It is advantageous as it is very easy to perform and secondly, it does not require executing program code, so as respective execution environment.

TABLE II. Heuristic Detection Techniques

Heuristic Detection	Process
Code Anomaly	Looks program code for detection of malware.
Protocol Anomaly	Looks for protocol like TCP/IP and any deviation from its standard for detection of malware.
Mixed Heuristic	It includes the combination of above two and can also combine other detection techniques like generic detection methods to validate results.

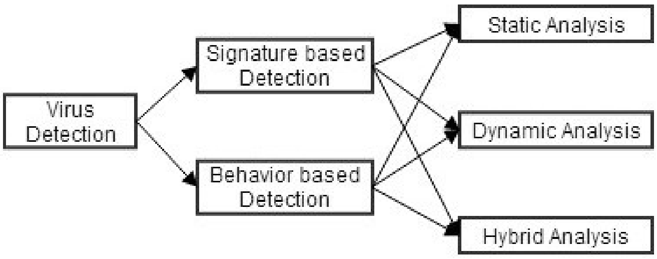

Fig. 3. Relationship between the detection and analysis techniques

B. Dynamic Analysis

Dynamic analysis is a good approach. It facilitates actual behavior by executing virus's program code and monitoring its behavior, system interaction and its vulnerable effect on system. Malware's behavior is analyzed through simulating infected files in special environment like a virtual machine, simulator, emulator, sandbox etc. As dynamic analyzers look for dynamic or behavior component, this analysis is referred as behavioral analysis. It shows problems in executing malware by not detecting an event which shows behavioral changes in code by different trigger conditions.

Static and dynamic analysis techniques are compared in table 3 with respect to their advantages and disadvantages [2] [8].

C. Hybrid Analysis

Hybrid analysis considers benefits of both two i.e. static and dynamic analysis and overcome their limitations. It provides enhanced protection as compared to earlier two. It first looks for malware's signatures and then applies behavioral constructs on it for analysis.

VI. Virus Detection Using Machine Learning

Machine learning can be used to detect viruses. It detects the similarities in virus instances. It is beneficial for detecting metamorphic viruses. Machine learning techniques used for virus detection are [11]:

- Neural Networks
- Data Mining techniques
- Hidden Markov Models

A. Neural network

Neural network technique is best for detecting viruses with similar set of features. In process, virus's features are analyzed first. Analyzed features are modeled in neural network model and trained periodically. Network model detects viruses whose features are modeled in it. It detects viruses with similar features, but they may or may not belong

TABLE III. VARIOUS VIRUS DETECTION TECHNIQUES ILLUSTRATING -- FEATURES, ADVANTAGES, DISADVANTAGE AND APPLICATIONS

Detection Technique	Features	Advantages	Disadvantages	Applications
Signature based	1. Scans files for virus's signatures. 2. Used for detecting known viruses. 3. Doesn't require knowledge of normal traffic.	1. Simple and efficient method. 2. Does not consume much time.	1. Cannot detect new viruses. 2. Large signature database is required to store virus's signatures, which is very costly to maintain. 3. Resource-consuming as every signature requires an entry in database. 4. Not reliable as a little change in signature will not detect virus.	1. Used in various basic anti-viruses to facilitate virus detection.
Anomaly based	1. Used for detecting unknown viruses. 2. Check for headers and payload of packets on network. 3. Observes network traffic and monitor host's behavior. 4. Detects abnormal behavior. 5. Notifies users about malicious content by generating alarms.	1. Keeps knowledge of normal traffic to track abnormal activities. 2. Checks packet's headers, connection type, packet payload, etc. to detect virus. 3. Suitable for unknown attacks. 4. Reliable as compared to Signature based detection.	1. Confusing to define what a normal behaviour (threshold) is or what is to be considered as abnormal behaviour. 2. A significant training period is required to define abnormal activity. So, it requires resources. 3. Time consuming, 4. May provide high false positives. 5. Costly to implement	1. Used for creating and defining signatures for new and unknown viruses.
Emulation based	1. Virtual environment is set to emulate instructions. 2. Program code's Instruction sequence and its behaviour are identified. 3. Code optimization technique can be used to optimize time required for virus detection. 4. Trail-and-error detection method.	1. Best for detecting polymorphic and metamorphic viruses, also for encrypted viruses. 2. Fast detection technique. 3. Simulate virus's behaviour to understand its remedy	1. Costly to implement, Time consuming. 2. May require extensive emulation i.e. need to execute virus program of thousands of line of code, and results in slow down the antivirus operation. 3. Does not detect virus that does not show their behavior while emulation. 4. May not provide good result for detecting memory resident programs.	1. Used to detect metamorphic viruses. 2. Used in Skulason's F-PROT antivirus program.
Generic Signature Scanning	1. Uses patterns and wildcards to detect different viruses. 2. Also known as heuristic signature scanning	1. Fast detection method. 2. Capable of detecting new and upcoming future viruses.	1. Cannot detect new viruses. 2. Not suitable for detecting complex virus like polymorphic and metamorphic viruses. 3. Require a highly skilled researcher for extracting virus's patterns.	1. Used to detect viruses belonging to same family.
Integrity Checking	1. Based on hash value of files on the system. 2. Compares hash function of files. 3. Maintains a hash (i.e. Checksum, snapshot, fingerprint) in database. 4. Restricts the virus detection space.	1. Simple and efficient method. 2. Does not consume much time. 3. Fast result producing method as we just need to check the integrity of the file. 4. Can be used to detect any viruses as it doesn't use the concept of virus's signatures.	1. Costly and resource consuming. 2. Tedious for user to update and maintain a huge hash database regularly. 3. Wouldn't have any advantage if system is already virus infected. 4. Does not provide benefit if program is from floppy disk or CD. 5. May raise alarm even on a legitimate change in program. 6. Not suitable for files that rapidly changes.	1. Used for restricting virus detection space. 2. Adopted in virus scanners and many security systems. 3. Implemented as System File Checker (SFC) in Window 2000 and XP systems to check files for change during booting.
Heuristic scanning	1. Examines program code for viruses and worm. 2. Looks for specific commands which signify the presence of malicious code. 3. User observes various programs like boot record, macro files, executable files for virus-like commands like malicious program's payload, worm propagation routine, virus replication routine, etc.	1. Used to detect new and unknown viruses. 2. Not based on virus's signatures, eliminating their drawbacks. 3. Look for protocol used in communication for detection.	1. If the program code to be detected is not updated, then new viruses may go undetected. 2. The potential for false alarms and not detecting a known virus is greater with heuristic scanners as compared to others. 3. May generate a lot of false alarms and may either scare novice users or give them a false sense of security. 4. Time consuming.	1. Used in Kaspersky Anti-Virus to facilitate virus detection.
Behavior based	1. Observes behavior and characteristics of known and unknown malwares. 2. Considers source and destination of the communication, type and size of attachments.	1. Facilitates successful detection of known and unknown viruses from several classes and under several execution conditions. 2. Provides minimal false positives and false negatives.	1. Problem is to define virus behaviour that assures detection of known and unknown viruses while not incorrectly detect benign process as virus. 2. Must be performed by knowledge experts whom are hard to find. 3. Produces false positives and false negatives.	1. Used in detection of virus with self reference replication behaviour. 2. Used for detecting virus in Smart terminal like mobile phones and PCs.
Neural Networks	1. Model is developed and trained over features. 2. Model must be trained with more features for better detection.	1. Best method for detection of viruses with similar features. 2. An efficient technique. 3. Provide low false positive rate if multiple features are used for training.	1. Efficiency depends on threshold values of the features, so maximum number of features must be considered for model. 2. Less number of features in model may result in high value of false positive.	1. Implemented for detection of boot sector viruses in IBM Antivirus program.
Data Mining	1. Rule based method. 2. Develop a model, based on virus's functionality.	1. Provides best results if multiple data mining model get combined.	1. Provides high false positive results.	1. Used for pattern detection in large set of data.

Hidden Markov Model	1. State machine based model, depends on current state and input. 2. It is based on Markov model.	1. Based on statistical model, so results can be analyzed easily. 2. Hides complexity and provides output directly.	1. Modeling process is time consuming and complex.	1. Used to detect metamorphic viruses.

to same family. A network model is efficient too. Its efficiency depends on threshold values for a minimum number of features, presented or trained in network model. Higher threshold value allows it to detect specific virus's families, but in contrast, a lower threshold value could create high false positive.

Neural Networks was implemented for detecting boot sector viruses in IBM Antivirus program. It works very efficiently and has low false positive rate as network model cover almost all features of boot sector virus [14].

A. Data mining techniques

Data mining techniques are rule based methods which create a model and train it with the functionality of the viruses. This created model detects the virus based on their functionality. This technique provides best result if multiple data mining model get combined. It provides high false positive results. It is widely used for pattern detection in a large set of data [15] [16].

B. Hidden markov models

Hidden markov model is mainly used to detect metamorphic viruses. The model is a statistical model, used to analyze markov process and provide results on the related observations. These results are used to make proper decisions. It is a state machine based model, which completely dependent on the current state and input's observation [17].

As far, we have studies various virus detection methods which are illustrated in Figure 4 and are summarized in Table 4. Among all these techniques, every detection technique does not provide full proof detection of all types of viruses, but yes they have their special field to serve. For the purpose of better detection, a user can even combine two to three techniques to protect his system.

VII. CONCLUSION

In this paper, Malwares are described which are involved in destruction of computer data and programs. There are various types of malwares including virus, worms, etc. Each malware has its own characteristic and process of infection. We have described virus, its definition, its evolution in different phases. As virus detection is very important for keeping data and programs safe, various detection techniques have been described in detail. Along with it some analysis techniques for virus detection, are also described. Some machine learning virus detection techniques are also explained to enhance the detection capability of detectors, which is evaded by the invention of various complex viruses like polymorphic viruses.

Moving forward, the paper would be helpful for new learners, who can learn and understand virus and its evolution, and their detection techniques can be easily understood. Moreover, these techniques can be used to develop antivirus software to detect viruses for protecting data on system, network, etc.

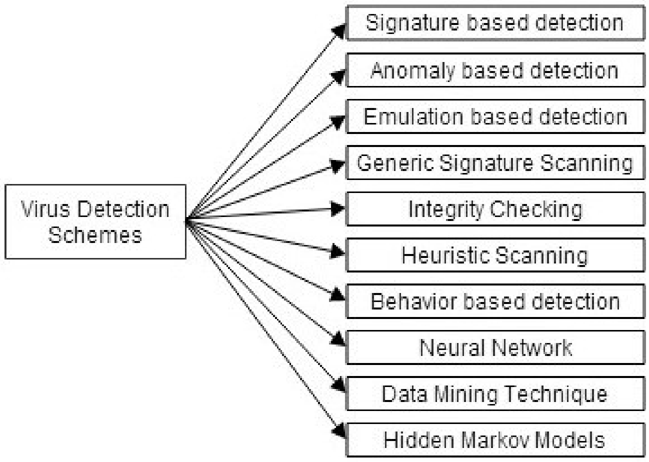

Fig. 4. Various virus detection techniques

REFERENCES

[1] S. Venkatachalam, "Detecting Undetectable Computer Viruses". Master's Projects. Paper 156. 2010. Available: http://scholarworks.sjsu.edu/etd_projects/156.

[2] K. Mathur, S. Hiranwal, "A Survey on Techniques in Detection and Analyzing Malware Executables", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Vol. 3, Issue 4, April 2013.

[3] Hossein Bidgoli, Handbook of Information Security, Volume 3,1st Edition, John Wiley & Sons, ISBN-10: 0471648337, ISBN-13: 978-0471648338, December, 2005.

[4] Peter Szor, The Art of Computer Virus and Defense, Harlow, England: Addison Wesely Professional, February, 2005, ISBN-10: 0321304543, ISBN-13: 978-0321304544.

[5] V. Sharma, "An Analytical Survey of Recent Worm Attacks", International Journal of Computer Science and Network Security, Vol.11, No.11, pp 99-103, November 2011.

[6] Craig Fosnock, "Computer Worms: Past, Present, and Future", East Carolina University, 2005

[7] H. Shravan Kumar, "Seminar Report on Study of Viruses and Worms", Indian Institute of Technology Bombay, 2005.

[8] J. Landage, M.P. Wankhade, "Malware and Malware Detection Techniques : A survey", International Journal of Engineering Research & Technology, ISSN : 2278-0181, Vol. 2, Issue 12, December 2013.

[9] Online documentation: http://antivirus.about.com/od/combinations/a/What-Is-A-Logic-Bomb.htm, Available on March, 2014.

[10] Online documentation: http://cecs.wright.edu/~pmateti/InternetSecurity/Lectures/Viruses/, Available on March, 2014

[11] S. Venkatachalam, M. Stamp, "Detecting Undetectable Metamorphic Viruses", Proceedings of the 2011 International Conference on Security & Management (SAM 2011), pp. 340-345, 2011, ISBN-10: 1-60132-196-1.

[12] Pele Li, Mehdi Salour, And Xiao Su, "A Survey of Internet worm detection And containment", IEEE Communications Surveys & Tutorials, 1st Quarter 2008, Vol. 10, NO. 1.

[13] Mishra, Umakant, "Methods of Virus Detection and Their Limitations", Available at SSRN: http://ssrn.com/abstract=1916708 or http://dx.doi.org/10.2139/ssrn.1916708, August 25, 2010.

[14] Tesauro G.J., Kephart J.O., Sorkin G.B., "Neural networks for computer virus recognition", IEEE Expert, Vol. 11, No. 4, pp 5-6, Aug 1996.

[15] J. Dai, R. Guha, J. Lee, "Feature set selection in data mining techniques for unknown virus detection - A comparison study", ACM International Conference Proceeding Series, 2009.

[16] O. Henchiri, N. Japkowicz, "A feature selection and evaluation scheme for computer virus detection", Proceedings – Sixth IEEE International Conference on Data Mining, ICDM, pp 891-895, December, 2006.

[17] S. Attaluri, S. McGhee, M. Stamp, "Profile hidden Markov models and metamorphic virus detection", Journal in Computer Virology, Vol. 5, No. 2, pp 151-169, May 2009.

Manju Khari (manjukhari@yahoo.co.in) is a research scholar with Delhi Technological University (formerly Delhi College of Engineering), Delhi, India. She is an Assistant Professor in Ambedkar Institute of Advanced Communication Technoloies and Research, Guru Gobind Singh Indraprastha University, Delhi, India. She received her Masters degree in Information Security from Ambedkar Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India. Her research interests are software testing, software quality, software metrics and artificial intelligence. She has published several papers in international journals and conference.

Chetna Bajaj (chetnabajaj7@gmail.com) is a research scholar, pursuing Masters of Technology in Information Security from Ambedkar Institute of Advanced Communication Technologies and Research, Guru Gobind Singh Indraprastha University, Delhi, India. She has completed her B.Tech in Computer Science & Engineering from Guru Premsukh Memorial college of Engineering, Guru Gobind Singh Indraprastha University, Delhi, India. Her research interests are Near Field Communcation, software security testing and their tools, Mobile Ad-hoc Network, and Virus Detection.