CHAPTER 2

RELATED WORKS AND LITERATURE SURVEY

2.1 INTRODUCTION

There are two main causes of SYN flood attacks. The first is the inherent asymmetry feature in TCP three-way handshake protocol, which enables an attacker to consume substantial resources at the server, while sparing its own resources. The other is that the server cannot control the packets it receives; especially the SYN packets can easily reach the server without its approval. A consequence of the SYN flooding attack is that a service can be brought down by flooding the server with a few tens of SYN requests per second. It is therefore important to detect these attacks as early as possible so as to bring the halted services back to normal. As most e-commerce depends heavily on TCP based applications, the severity of the problem increases. A number of literatures investigate the mitigation or detection of the SYN flood attack.

Many algorithms have been proposed in the past, to catch and stop the spread of camouflaging worms. Most research papers discuss efforts that are related to their proposed work.

This chapter discusses the literature survey and the related works done on TCP SYN Flood attack detection by IP Trace back and Camouflaging worm detection. The aim of this chapter is to provide a deep
insight into TCP SYN Flood attack and Camouflaging worm detection and the state of the art.

2.2 TCP SYN FLOOD ATTACK DETECTION TECHNIQUES

2.2.1 General Detection Techniques

1. Anomaly Detection

Haris (2010) had proposed a mechanism for detecting threats in IP headers by taking into consideration the IP and TCP header and checked the packet using anomaly detection. TCP SYN Flood attacks and other attacks could be detected through the traffic monitoring tools and anomalies are alerted to the administrator. In this paper, rate-based detection is used for anomaly detection. Anomaly detection has three types of detection in network analysis behavior. It uses protocol to detect packets that are too short which violate specific application layers protocol, rate-based detection which detects floods in traffic using a time-based model of normal traffic volumes especially Denial of Service (DoS) attacks. Lastly, it detects through the behavioral or relational changes in how individual or groups of hosts interact with one another on a network. If the packet is infected, the system will distinguish the packet and go for analysis again to confirm whether the packet truly comes from attackers. Otherwise, the normal packet will go through the network sending the data to the destination. Threats are detected due to the IP Header (payload and unusable area).

Disadvantages: Even though this approach detects the anomalies in the IP header, the source of the attack could not be traced back. In the experiment, the main threats like SYN Flood attack had been traced in a small amount. It is because the Linux operating system is stable and it is very hard to attack by the hackers.
2. **Payload Content based Network Anomaly Detection**

Sandeep A. Thorat (2008) proposed Payload Content based Network Anomaly Detection (PCNAD) which takes into consideration the entire payload for profile calculation and effectively for anomaly detection. This detection mechanism analyzes normal payload for a particular service on a host and makes a set of payload profiles. In PCNAD, the system has two components Packet Profiler and PCNAD Anomaly Detector. The Packet Profiler deals with normal packet profiling and creates the reduced profile model. In PCNAD Anomaly Detector, incoming packet is compared with stored profiles and alerts are generated if incoming packet is significantly different from stored profiles.

**Disadvantages:** The drawback of this approach is that the system is showing poor results at ports like 22, 23, 25 and the complexity is very high.

3. **Flood Detection System and Cumulative Sum**

Wang et al (2002) used the Flood Detection System (FDS) (2002), which in turn used Cumulative Sum (CUSUM) that detect the SYN flooding attacks at leaf routers which connect end hosts to the Internet, instead of monitoring the ongoing traffic at the front end (like firewall or proxy) or a victim server itself. The detection utilizes the SYN-FIN pairs’ behavior and distinguish features make it immune to SYN flooding attacks. CUSUM method that make the detection robust and it does not undermine the end-to-end TCP performance. This mechanism not only sets alarms upon detection of ongoing SYN flooding attacks, but also reveals the location of the flooding sources.
Disadvantages: If the packets of the same TCP session go through different leaf routers, we need a loose synchronization mechanism between the FDSs in these leaf routers; this detection scheme does not work. If the spoofed source address is in the same stub network as a flooding source, it cannot detect the ongoing flooding attack.

2.2.2 Techniques on IP Trace Back

IP Trace back is a name given to any method for reliably determining the origin of a packet on the Internet. Due to the trusting nature of the IP protocol, the source IP address of a packet is not authenticated. As a result, the source address in an IP packet can be falsified (IP address spoofing) allowing for Denial of Service attacks (DoS) or one-way attack. The problem of finding the source of a packet is called the IP Trace back problem. IP Trace back is a critical ability for identifying sources of attacks and instituting protection measures for the Internet.

As shown in Figure 2.1, there are mainly two types of trace back technique preventive and reactive which are further divided into various techniques as follows.

![Figure 2.1 IP Trace back Methods](image)
1. **Ingress Filtering**

One way to address the problem of anonymous attacks proposed by E. Ferguson and D. Senie (2000) is to eliminate the ability to forge source addresses. One such approach, frequently called ingress filtering, is to configure routers to block packets that arrive with illegitimate source addresses. This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. Consequently, ingress filtering is most feasible in customer networks or at the border of Internet Service Providers (ISPs) where address ownership is relatively unambiguous and traffic load is low.

**Disadvantages**

- The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service either because they are uninformed or have been discouraged by the administrative burden, potential router overhead, and complications with existing services that depend on source address spoofing (e.g., some versions of Mobile IP and some hybrid satellite communications architectures by C. Perkins, (1996)).

- A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network.
2. **Link Testing**

Most existing trace back techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attackers’ traffic. Ideally, this procedure is repeated recursively on the upstream router until the source is reached. This technique assumes that an attack remains active until the completion of a trace and is therefore inappropriate for attacks that are detected after the fact, attacks that occur intermittently, or attacks that modulate their behavior in response to a trace back (it is prudent to assume the attacker is fully informed). Below two varieties of link testing schemes are described, input debugging and controlled flooding. This approach is only valid if the route to and from the customer is symmetric generally at the border of single-homed stub networks.

i) **Input Debugging**: Many routers include a feature called input debugging, which allows an operator to filter particular packets on some egress port and determine which ingress port they arrived on. This capability is used to implement a trace as follows. First, the victim must recognize that it is being attacked and develop an attack signature that describes a common feature contained in all the attack packets. The victim communicates this signature to a network operator, frequently via telephone, who then installs a corresponding input debugging filter on the victim's upstream egress port. This filter reveals the associated input port, and hence which upstream router originated the traffic. The process is then repeated recursively on the upstream router, until the originating site is reached or the trace leaves the ISP's border (and hence its administrative control over the routers). In the latter case, the upstream ISP must be contacted and the procedure repeats itself. One such system, called Center Track by Stone (2000), provides an improvement over hop-by-hop backtracking by dynamically rerouting all of the victim's traffic to flow through a centralized tracking router. Once this
reroute is complete, a network operator can then use input debugging at the tracking router to investigate where the attack enters the ISP network.

**Disadvantages:** The most obvious problem with the input debugging approach, even with automated tools, is its considerable management overhead. Communicating and coordinating with network operators at multiple ISPs requires the time, attention, and commitment of both the victim and the remote personnel many of whom have no direct economic incentive to provide aid. If the appropriate network operators are not available, if they are unwilling to assist, or if they do not have the appropriate technical skills and capabilities, then a trace back may be slow or impossible to complete as by Glave (1998).

ii) **Controlled Flooding:** Burch and Cheswick (2000) have developed a link-testing trace back technique that does not require any support from network operators. This technique is controlled flooding because it tests links by flooding them with large bursts of traffic and observing how this perturbs traffic from the attacker. Using a pre generated "map" of Internet topology, the victim coerces selected hosts along the upstream route into iteratively flooding each incoming link on the router closest to the victim. Since router buffers are shared, packets traveling across the loaded link including any sent by the attacker, have an increased probability of being dropped. By observing changes in the rate of packets received from the attacker, the victim can therefore infer which link they arrived from.

**Disadvantages:** While the scheme is both ingenious and pragmatic, it has several drawbacks and limitations. Most problematic among these is that controlled flooding is itself a denial-of-service attack, exploiting vulnerabilities in unsuspecting hosts to achieve its ends. This drawback alone makes it unsuitable for routine use. Also, controlled flooding requires the
victim to have a good topological map of large sections of the Internet in addition to an associated list of "willing" flooding hosts. As Burch and Cheswick (2000) note, controlled flooding is also poorly suited for tracing distributed denial-of-service attacks because the link-testing mechanism is inherently noisy and it can be difficult to discern the set of paths being exploited when multiple upstream links are contributing to the attack. Finally, like all link-testing schemes, controlled flooding is only effective at tracing an ongoing attack and cannot be used "post mortem."

3. **Logging**

An approach suggested by Sager (1998) and Stone (2000) is to log packets at key routers and then use data mining techniques to determine the path that the packets traversed. This scheme has the useful property that it can trace an attack long after the attack has completed.

**Disadvantages:** It also has obvious drawbacks, including potentially enormous resource requirements (possibly addressed by sampling) and a large scale inter provider database integration problem.

4. **ICMP Trace back**

The principle idea in this scheme is for every router to sample, with low probability (e.g., 1/20 000), one of the packets it is forwarding and copy the contents into a special ICMP Trace back message including information about the adjacent routers along the path to the destination as by Bellovin, (2000). During a flooding-style attack, the victim host can then use these messages to reconstruct a path back to the attacker. This scheme has many benefits compared to previous work and is in many ways similar to the packet marking approach we have taken.
Disadvantages: There are several disadvantages in the current design that complicate its use. Among these: ICMP traffic is increasingly differentiated and may itself be filtered in a network under attack; the ICMP Trace back message relies on an input debugging capability that is not available in some router architectures; if only some of the routers participate it seems difficult to positively "connect" trace back messages from participating routers separated by a non-participating router; and finally, it requires a key distribution infrastructure to deal with the problem of attackers sending false ICMP Trace back messages.

5. Marking Algorithms

All marking algorithms have two components: a marking procedure executed by routers in the network and a path reconstruction procedure implemented by the victim. A router "marks" one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets.

- Node Append: The simplest marking algorithm proposed by Postel (1981) conceptually similar to the IP Record Route option is to append each node's address to the end of the packet as it travels through the network from attacker to victim. Consequently, every packet received by the victim arrives with a complete ordered list of the routers it traversed, a built-in attack path. The node append algorithm is both robust and extremely quick to converge (a single packet).

Disadvantages: It has several serious limitations. Principal among these is the unfeasibly high router overhead incurred by appending data to packets in flight. Moreover, since the length of the path is not known a priori, it is impossible to ensure that there is sufficient unused space in the packet for
the complete list. This can lead to unnecessary fragmentation and bad interactions with services such as MTU discovery as stated by J. Mogul and S. Deering (1990). This problem cannot be solved by reserving "enough" space, as the attacker can completely fill any such space with false, or misleading, path information.

- **Node Sampling:** Node Sampling is a technique in which a field in packet header is reserved for marking. The router writes its address in packet with probability p. To reduce both the router overhead and the per-packet space requirement, we can sample the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header which is large enough to hold a single router address (i.e., 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path. We assume that the attacker sends enough packets and the route is stable enough that this sampling can converge.

**Disadvantages**

There are two serious limitations:

- First, inferring the total router order from the distribution of samples is a slow process. Routers far away from the victim contribute relatively few samples (especially since p must be large) and random variability can easily lead to disordering unless a very large number of samples are observed.

- Second, if there are multiple attackers, then multiple routers may exist at the same distance hence it is sampled with the
sample probability. Therefore, this technique is not robust against multiple attackers.

- **Edge Sampling:** In Edge Sampling, edges are stored which consists of start and end addresses, distance from edge to victim instead of entire router information in the path. The arriving packet contains address of last marked edge and number of hops the edge is from destination. A straightforward solution to these problems is to explicitly encode edges in the attack path rather than simply individual nodes. To do this, we would need to reserve two static address sized fields, start and end, in each packet to represent the routers at each end of a link, as well as an additional small field to represent the distance of an edge sample from the victim. When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero this indicates that the packet was marked by the previous router. In this case, the router writes its own address into the end field, thereby representing the edge between itself and the previous router and increments the distance field to one. Finally, if the router does not mark the packet then it always increments the distance field. This somewhat baroque signaling mechanism allows edge sampling to be incrementally deployed where the edges are constructed only between participating routers.

**Disadvantages:** The edge-sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP).

- **Probabilistic Packet Marking Schemes:** Probabilistic Packet Marking (PPM) developed by Computer Security Institute and Federal Bureau of Investigation (1999) is one stream of the packet marking methods. The assumption of PPM is that the attacking packets are much more frequent than
the normal packets. It marks the packets with path information in a probabilistic manner and enables the victim to reconstruct the attack path by using the marked packets. PPM encodes the information in rarely used 16-bit Fragment ID field in the IP header. To reduce the data that is to be stored in 16 bits, the compressed edge fragment sampling algorithm is used.

**Disadvantages:** Although PPM is simple and can support incremental deployment, it has many short comings that can seriously prevent it from being widely used. First, the path reconstruction process requires high computational work, especially when there are many sources. For example, a 25-source path reconstruction will take days, and thousands of false positives could happen according to D. Dean et al (2001). Second, when there are a large number of attack sources, the possible rebuilt path branches are actually useless to the victim because of the high false positives. Therefore, the routers that are far away from the victim have a very low chance of passing their identification to the victim because the information has been lost due to overwriting by the intermediate routers. Many approaches were proposed to overcome the above deficiencies.

- **Other Probabilistic Packet Marking Schemes:** For example, Song and Perrig (2001) proposed an advanced and authenticated PPM based on the assumption that the victim knows the mapping of the upstream routers. It not only reinforces the capability to trace more sources at one time but also solves the problem of spoofed marking. Another method to reduce the overhead of reconstruction was proposed by Govindan and Tangmunarunkit (2000). It uses counters to complement the loss of marking information from upstream routers, in order to save computation time and reduce false positives. Adler et al (1996) analyzed the tradeoff between mark bits required in the IP header and the number of packets required to reconstruct the paths. On the other hand, Savage et al (2000) propose a PPM scheme with edge
sampling which is called FMS. Yaar et al (2005) propose the FIT scheme. Al-Duwari and Govindarasu (2006) propose the probabilistic pipelined packet marking (PPPM) scheme. Gong and Sarac (2009) propose a practical packet marking scheme. Lee et al (2004) presented an approach to reduce the overhead of the hash-based approach. They proposed to digest packet aggregation units (packet flows or source-destination sets) instead of individual packets. Recording the digests of packet aggregation units reduces the digest table storage overhead. However, tracing an individual packet is accomplished by tracing the packet aggregation to which the packet belongs. Moreover, depending on the implementation, either the writing or reading rate of digest table should be commensurate with packet arriving rate. Thus, this approach does not alleviate the access time requirement.

Li et al (2004) proposed probabilistic packet logging where routers probabilistically select a small percentage of forwarded packets to record their digests. This method reduces both the storage overhead and access time requirement for recording packet digests at routers. But the tradeoff is the loss of the ability to trace individual packets since the probability that all routers on an attack path record a specific packet is tiny.

- **Distributed Link-List Traceback (DLLT) and Probabilistic Pipelined Packet Marking (PPPM):** Recently, Al-Duwairi and G. Manimaran (2006) proposed two hybrid IP trace back approaches, distributed link-list trace back (DLLT) and probabilistic pipelined packet marking (PPPM). The main design goal in these approaches is to reduce the number of packets required for constructing attack paths in the PPM approach through utilizing packet logging. In comparison, our work is to make use of packet marking to reduce the overhead of log-based IP trace back in tracing a single packet. The basic idea in both DLLT and PPM approaches is to preserve the marking information carried by the packet before marking a packet. When a
router probabilistically decides to mark a packet, the router records the marking value carried by the packet before writing a new value into the packet.

**Disadvantages:** In DLLT, the preserved marking information is stored at routers and victims query routers during the traceback process. In PPPM, routers transfer those marking information to the original destinations via writing them into other packets to the same destinations.

- **Deterministic Packet Marking Schemes:** Another stream of packet marking methods, which does not use the above probabilistic assumption and stores the source address in the marking field, is in the category known as the deterministic approaches, such as Deterministic Packet Marking (DPM) as proposed by Howard (1998), Kam and Simpson (1999). Recently, in "A Formal Framework and Evaluation Method for Network Denial of Service," by Meadows, (1999) the DPM scheme was modified to reduce false positive rates by adding redundant information into the marking fields. Unlike PPM, deterministic approaches only keep the first ingress edge router’s information in the marks (but not the whole path). Moreover, they record marks in a deterministic manner (but not a probabilistic manner as in PPM). This category of schemes has many advantages over others, including simple implementation, no additional bandwidth requirement, and less computation overhead. However, enough packets must be collected to reconstruct the attack path (e.g., in the best case, at least two packets are required to trace one IP source with any of the above schemes). Importantly, all previous works neither perform well in terms of, nor have addressed the problems of, the maximum number of sources that the trace back system can trace in a single trace back process, the number of packets needed to trace one source, and the overload prevention on participating routers.
Disadvantages: Though the marks in DPM cannot be spoofed, frequent spoofing/changes of the source address with a different value by an attacker may void the DPM’s effectiveness

- Flexible Deterministic Packet Marking: Flexible Deterministic Packet Marking (FDPM) as by Y. Xiang et al (2004) utilizes many bits in the IP header that has a flexible length. When an IP packet enters the protected network, it will be marked by the interface close to the source of the packet on an edge ingress router. The source IP addresses are stored in the marking fields. The mark will not be changed when the packet traverses the network. At any point within the network, the source IP addresses can be assembled when necessary. Because the maximum length of mark is 25 bits, at least 2 packets are needed to carry a 32-bit source IP address. Each packet holding the mark will be used to reconstruct the source IP address at any victim end within the network. A segment number is also assigned to the mark, because when reconstructing the packet, the segment order of the source IP address bits must be known. After all the segments corresponding to the same ingress address have arrived to the destination, the source IP address of the packets can be reconstructed. In order to keep a track on a set of IP packets that are used for reconstruction, the identities shown the packets come from the same source must be given. A hash of the ingress address is kept in the mark, known as the digest. This digest will always remain the same for a FDPM interface from which the packets enter the network. It provides the victim end the ability to recognize which packets being analyzed are from a same source, although the digest itself cannot tell the real address. Even if the participating router is compromised by attackers (for example, some marks are spoofed), this scheme will not be affected because the packets with irrelevant digest will be discarded during the reconstruction process.
Disadvantages: The packet processing consumes resources such as memory and computing capacity of a participating router. Therefore, it is possible for a router to be overloaded when there are a large number of arrival packets.

2.2.3 Other Related Works

1. Authenticated Key-Exchange

Hua Jiang et al (2010) “Authenticated Key-Exchange Scheme Based on SGC-PKE for P2PSIP” analyzed threats in Peer-to-Peer SIP (P2PSIP) architecture and then proposed a new P2PSIP authenticated key exchange mechanism based on Self-Generated-Certificate Public Key Encryption scheme (SGC-PKE) which assures the Integrity, Confidentiality and Non-repudiation of messages in P2PSIP network, and overcomes some shortcomings from traditional public key cryptography (PKC) system.

2. Key Exchange Protocols

Ota et al (2010) “Security Verification for Authentication and Key Exchange Protocols” focuses on preferable authentication and key exchange protocols to be verified automatically and rapidly in accordance with security requirements. They proposed a security verification method (OKT method) for authentication and key exchange protocols based on Bellare et al.’s model (BPR model). However, there is an estrangement between the security of the OKT method and the BPR model. They reconsidered the OKT method and once again proposed a security verification method for authentication and key exchange protocols based on the BPR model. In particular, they showed the novel verification points for each security property in the authentication and key exchange protocols.
3. **Encrypted Marking based Detection and Filtering (EMDAF)**

Nagartna et al (2009) proposed EMDAF. EMDAF is encrypted based packet marking technique used as a solution for IP trace back in case of IP spoofing. But, it involves 10% of routers communication in the process and also it adds load on the server to generate the encrypted key.

4. **Hop count Based packet processing approach**

Bharathikrishnakumar et al (2010) proposed Hop count Based packet processing approach. Hop Count Based is one of the recent solutions proposed to counter DDOS attacks. In this method an assumption that systems in the current internet architecture are located max with a hop count of 255. In this approach the packets from the systems at the same hop count and traversing through the same router are marked with the same identification number. This number is derived by the concatenation of the 32 bits of the IP address of the router path and the encrypted value of the hop count. At the receiving side of the router interface the hop count value of the incoming packet is checked with the already stored value. This technique provides an advantage of immediately filtering the traffic after receiving just one attack packet and it does not require any change in the existing protocols. Thus this technique has a significant potential in reducing the threats caused by the DDoS attacks

2.2 **CAMOUFLAGING WORM DETECTION**

2.3.1 **Network based Detection Techniques**

1. **Power spectrum-based detection scheme**

the propagation traffic generated by worms. Active worms pose major security threats to the Internet. This is due to the ability of active worms to propagate in an automated fashion as they continuously compromise computers on the Internet. Active worms evolve during their propagation, and thus, pose great challenges to defend against them. In this paper, they investigated a new class of active worms, referred to as Camouflaging Worm (C-Worm). The C-Worm is different from traditional worms because of its ability to intelligently manipulate its scan traffic volume over time. Thereby, the C-Worm camouflages its propagation from existing worm detection systems based on analyzing the propagation traffic generated by worms. Their scheme uses the Power Spectral Density (PSD) distribution of the scan traffic volume and its corresponding Spectral Flatness Measure (SFM) to distinguish the C-Worm traffic from background traffic. Using a comprehensive set of detection metrics and real-world traces as background traffic, they conducted extensive performance evaluations on proposed spectrum-based detection scheme. The performance data clearly demonstrates that PSD scheme can effectively detect the C-Worm propagation. Furthermore, they have shown the generality of the spectrum-based scheme in effectively detecting not only the C-Worm, but traditional worms as well.

2. Detection based on various scan techniques

Jiang Wu et al (2004), “An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques” analyzes various scan techniques and proposed a generic worm detection architecture that monitors malicious activities. They evaluated an algorithm to detect the spread of worms using real time traces and simulations. They presented an analysis on potential scan techniques that worms can employ to scan vulnerable machines. In particular, they found that worms can choose targets more carefully than the random scan. A worm that scans only IP addresses
announced in the global routing table can spread faster than a worm that employs random scan. They analyzed a family of scan methods and compared them to the random scan. Second, they proposed worm detection architecture and algorithms for prompt detection of worm activities. Their detection architecture takes advantage of the fact that a worm typically scans some unassigned IP addresses or an inactive port of assigned IP addresses. By monitoring unassigned IP addresses or inactive ports, one can collect statistics on scan traffic. They proposed a detection algorithm called victim number based algorithm, which relies solely on the increase of source addresses of scan traffic and evaluated its effectiveness.

**Disadvantages:** Their solution can detect worm activities when only 4% of the vulnerable machines are infected. The number of false alarms increases in the case of a DDoS attack or in the case of a hot website visit.

3. **Super spreaders detection:**

Venkataraman et al (2005), “New Streaming Algorithms for Fast Detection of Super spreaders” considered the problem of detecting super spreaders, which are sources that connect to a large number of distinct destinations. They proposed new streaming algorithms for detecting super spreaders and prove guarantees on their accuracy and memory requirements. They also showed experimental results on real network traces. The algorithms are substantially more efficient (both theoretically and experimentally) than previous approaches. They also extended their algorithms to identify super spreaders in a distributed setting, with sliding windows, and when deletions are allowed in the stream (which lets us identify sources that make a large number of failed connections to distinct destinations). They proposed two efficient algorithms to find super spreaders. The one-level filtering algorithm is based on sampling from the set of distinct source destination pairs. Then presented a more complex algorithm based on a novel two-level filtering
scheme, which will be more space-efficient than the one-level filtering algorithm for the distributions that will be more common.

**Disadvantages:** One level filtering algorithm has several difficulties. It needs a certain minimum sampling rate in order to distinguish between sources.

4. **Varying Scan Rate worm detection**

Wei Yu et al (2006), “Effective detection of active worms with varying scan rate”, modeled a new form of active worms called Varying Scan Rate worm (the VSR worm). In this paper, they contributed in Worm Modeling and Effective Detection Scheme. They modeled the above new form of worms as Varying Scan Rate worm (the VSR worm). The VSR worm is generic and simple to launch. Apart from a self-propagating behavior similar to traditional worms, the VSR worm adopts the polymorphic methodology to vary the overall scan rate, aiming to avoid being detected. Therefore, it can achieve its ultimate goal to infect as many computers as possible before being detected. The scan rate variation could simply involve making a part of worm instances temporarily cease attack/scanning during the worm propagation or making worm instances deliberately change their scan rates. On further analysis the effectiveness of the VSR worm in evading detection by existing worm detection schemes is brought out. Analysis data clearly shows that existing threshold-based and trend based detection schemes are not effective in detecting this new form of worms. To detect the VSR worm, they developed a novel worm detection scheme called attack target Distribution Entropy based dynamic detection scheme (DEC detection). The DEC detection scheme is generic with several important features to make it effective in detecting both VSR worms and traditional worms. The scheme adopts the distribution of attack targets as the basic detection data to capture the key feature of worm propagation, i.e., continuously scanning different
targets, which is not expected in non-worm scan traffic. Evaluation results demonstrated that DEC detection scheme has superior performance in terms of speed and accuracy in detecting not only VSR worms but also traditional PRS worms.

**Disadvantages:** They found that packets originating from one of the host machines in their network showed behavior typical of the slapper worm (attempting to infect other hosts).

### 2.3.2 Host based Detection Techniques

1. **Distributed host based worm detection system**

   Senthilkumar G. Cheetancheri et al (2006), “A distributed host based worm detection system”, presents a method for detecting large-scale worm attacks using only end-host detectors. These detectors propagate and aggregate alerts to cooperating partners to detect large scale distributed attacks in progress. The properties of the host-based detectors may in fact be relatively poor in isolation but when taken collectively result in a high-quality distributed worm detector. A cooperative alert sharing protocol coupled with distributed sequential hypothesis testing to generate global alarms about distributed attacks. They evaluated the system's response in the presence of a variety of false alarm conditions and in the presence of an Internet worm attack. Their evaluation is conducted with agents on the Emu lab and DETERS emulated test beds using real operating systems and computing platforms.

   **Disadvantages:** They have not taken into consideration the effect of the worm traffic from outside their network of interest. They have also not considered the effects of malicious nodes in the federation in their experiments.
2. **Mining Dynamic Program Execution**

Xun Wang et al (2007) “Detecting Worms via Mining Dynamic Program Execution” proposed a new worm detection approach based on mining dynamic program executions. This approach captures dynamic program behavior to provide accurate and efficient detection against both seen and unseen worms. In particular, they executed a large number of real world worms and benign programs (executables), and trace their system calls. They applied two classifier-learning algorithms (Naïve Bayes and Support Vector Machine) to obtain classifiers from a large number of features extracted from the system call traces. The learned classifiers are further used to carry out rapid worm detection with low overhead on the end-host. Their experimental results clearly demonstrate the effectiveness of their approach to detect new worms in terms of a very high detection rate and a low false positive rate.

**Disadvantages:** This method is the study of host-based detection and they did not consider information about the traffic generated by the executables during the worm detection. Since these worm behaviors are exposed from different perspectives, consideration of multiple behaviors could provide more accurate worm detection.

3. **Local worm victim detection algorithm**

GuofeiGu et al (2004), “Worm detection, early warning and response based on local victim information”, concentrates on a simple two-phase local worm victim detection algorithm, DSC (Destination-Source Correlation), based on worm behavior in terms of both infection pattern and scanning pattern. DSC can detect zero-day scanning worms with a high detection rate and very low false positive rate. DSC does not aim to detect all types of worms. DSC aims to detect scan-based, fast - spreading worms.
Further, the infection time for hosts is not very long. Compared to the fast spreading worms like SQL slammer and Code Red, slow spreading worms do less damage to networks and are easier to contain, in part because their slower spread rate allows for human intervention. The DSC algorithm includes two phases. First, they found the infection-like pattern. Second, they checked the abnormal outgoing scan rate for suspicious hosts observed in first phase. Local response can be considered as the third phase when we automatically quarantine the victim’s outgoing traffic at the right port.

**Disadvantages:** DSC may not effectively detect email worms, very slow scanning worm, or sleeper worms with very slow rates of infection.

### 2.3.3 Real-time Detection Schemes

#### 1. Early Bird System

Sumeet Singh et al (2003), “The Early Bird System for Real-time Detection of Unknown Worms” proposed an automated method for detecting new worms based on traffic characteristics common to most of them: highly repetitive packet content, an increasing population of sources generating infections and an increasing number of destinations being targeted. Network worms are a major threat to the security of today’s Internet-connected hosts and networks. The combination of unmitigated connectivity and widespread software homogeneity allows worms to exploit tremendous parallelism in propagation. Modern worms spread so quickly that no human-mediated reaction to the outbreak of a new worm can hope to prevent a widespread epidemic. This method generates content signatures for the worm without any human intervention. Preliminary results on a small network show promising results: they have identified three confirmed worms with a low percentage of false positives. Thus this method could form the core of an effective network-
level worm detection and countermeasure system capable of substantially slowing down the spread of new worms.

2. **Real-time worm detection**

Bharath Madhusudan et al (2004), “Design of a system for real-time worm detection” presents the design and implementation of a system that automatically detects new worms in real time by monitoring all traffic on a network. In this paper, they presented the design and implementation of a system that automatically detects new worms in real-time by monitoring traffic on a network. The system uses Field Programmable Gate Arrays (FPGAs) to scan packets for patterns of similar content. Given that a new worm hits the network and the rate of infection is high, the system is automatically able to detect an outbreak. Frequently occurring strings in packet payloads are instantly reported as likely worm signatures.

**Disadvantages:** The system is quite effective at detecting smaller worms at an early stage. But detecting larger worms becomes a much harder task.

2.3.4 **Other Related Works**

1. **Different host based defense systems**

Zesheng Chen et al (2003), “On Effectiveness of defense systems against active worms” focuses on the detection of active worms which use self-propagating malicious code, and have been a persistent security threat to the Internet since 1988. In this paper, they have investigated these questions through modeling and analysis. Using a discrete-time model, it is shown that three key characteristics of worm propagation are exploited by the current defense systems: number of vulnerable machines, scanning rate, and time to
complete infection. They first defined the performance and resources of defense systems. Then they have derived and analyzed the relationship between the performance and the resources for four widely-used or promising defense systems focusing on the worms that employ random scanning. As a result they found that the existing defense systems can be categorized into two groups. One exploits the number of vulnerable machines, and the other focuses on the scanning rate. Analysis shows that a significant amount of resources is required for the existing systems to fight effectively against active worms. When a single system cannot acquire enough resources to contain worms, a combined use of all defense systems provides a hope to fight against worm propagation efficiently. This work is the first attempt on understanding the essence of different host based defense systems and their combination quantitatively.

2. Taxonomy of Computer Worms

Nicholas et al (2003), “A Taxonomy of Computer Worms”, focuses on a preliminary taxonomy based on worm target discovery and selection strategies, worm carrier mechanisms, worm activation, possible payloads, and plausible attackers who would employ a worm. The carrier, activation, and payload are independent of each other, and describe the worm itself. One or more of these techniques can be used to develop more robust defenses that can focus on preventing worm. They also include a section on attackers and their motivations, because worms are ultimately written by humans, and sometimes the easiest way to defend against a worm is to remove the motivation for writing a worm in the first place.

3. WAtCoS

Madihah Mohd Saudi et al (2008), “Worm Analysis through computer simulation (WAtCoS)” aims to resolve the confusion in identifying
visualization, simulation and games in teaching malware analysis. Specifically, they describe a preliminary simulation study on the effect of the spread of the Ramen and SQL Slammer/Sapphire worms on the multicast infrastructure and their ultimate goal is to create a realistic simulation platform to evaluate and tune techniques to mitigate the effects of scanning worms on network protocols. In this research they have made a comparison on visualization, simulation and games and stated that Simulation is a useful tool in many areas of computer science education. Their examples indicate that great potential can be realized much more rapidly. The simulation is at least as effective as other methods for teaching knowledge about facts, concepts and application of knowledge.

4. Digital Signatures

Byung K. Yi et al (2006), “Digital Signatures”, describes the working and the use of digital signatures in protecting our confidential information flowing through a network. This paper briefly about the conventional and digital signature characteristics followed by explaining the procedure to create and verify the digital signature and digital envelope. Its applications in real time are also discussed over here. This work is motivated by the fact that many traditional and newer businesses and applications have recently been carrying out enormous amounts of electronic transactions, which have led to a critical need for protecting the information from being maliciously altered, for ensuring the authenticity, and for supporting non repudiation and this technology is rather new and emerging and is expected to experience growth and widespread use in the coming years.

5. Survey of internet worm detection and containment

Pele Li et al (2008), “A survey of internet worm detection and containment” presents a survey and comparison of Internet worm detection
and containment schemes. They first identified worm characteristics through their behavior, and then classified worm detection algorithms based on the parameters used in the algorithms. Furthermore, they analyzed and compared different detection algorithms with reference to the worm characteristics by identifying the type of worms that can and cannot be detected by these schemes. After detecting the existence of worms, the next step is to contain them. This proposal explores the current methods used to slow down or stop the spread of worms. The locations to implement detection and containment, as well as the scope of each of these systems or methods, are also explored in depth. Finally, this paper points out the remaining challenges of worm detection and future research directions.

2.3 SUMMARY

DDoS attacks, particularly TCP SYN Flood attack results in slow network performance, unavailability of a particular web site, inability to access any web site and dramatic increase in the amount of spam you receive in your account.

An active worm such as Camouflaging worms infects as many computers before being detected. As more and more computers get infected, they, in turn, take part in scanning other computers. Hence, it is considered the C-worm as a worst case attacking scenario that uses a closed-loop control for regulating the propagation speed based on the feedback propagation status.

In this chapter, the existing state-of-the-art to detect the TCP SYN Flood attacks and Camouflaging worms are discussed with their drawbacks. To overcome all these issues in detecting the attacks and worms, the forthcoming chapters propose advanced detection models.