

2.1 Introduction :

Port scan detection is very important for security management. Many attackers perform port scans as a beginning to find out vulnerable hosts to compromise. Detecting such port scans indicates incoming network intrusions. Besides, recent worm epidemics, such as Code Red-II, Nimda, etc. Scan other vulnerable hosts for propagation. Network supervisors can prevent viruses from spreading by detecting those port scans and then prohibiting them.

2.2 Approaches to port scan detection:

We classify various port scan detection approaches available in the literature into two different categories: *single-source* approaches and *distributed* approaches. Single-source port scan is performed following either a *one-to-one* or a *one-to-many* model for gathering information about a target computer or network. On the other hand, distributed information gathering [10] is performed using a *many-to-one* or *many-to-many* model for gathering information about a target computer or network. A hierarchy of the scan detection approaches is reported in **Figure 2.1**. [4]

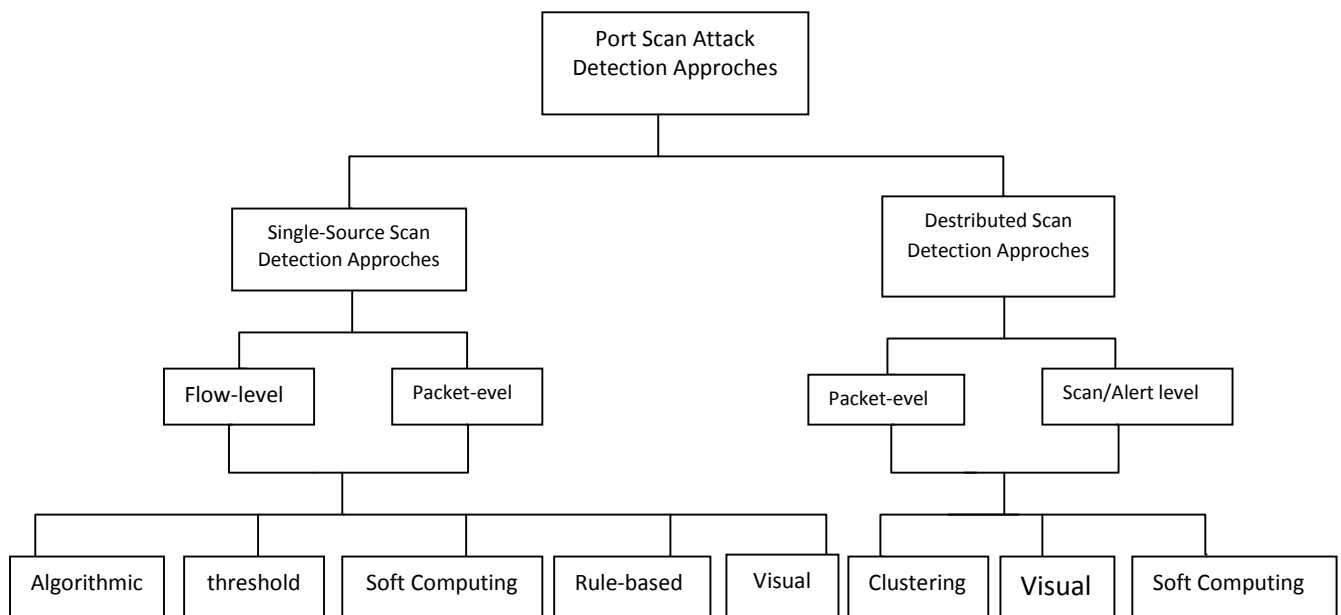


Figure 2.1 hierarchy of port scan attack detection approaches

2.2.1 Single-Source Port Scan Detection Approaches:

Detection approaches for single-source port scans have been part of intrusion detection systems since 1990, from the release of NSM [11]. We divide these detection approaches into five categories: *Algorithmic*, *Threshold-based*, *soft Computing based*, *Rule-based*, and

Visual. Each of these can be further categorized based on the type of network data processed, methodology used for detection and the evaluation criteria. For example, some approaches exploit packet level information whereas some others use flow level information. These details provide not only the connection information, but also allow one to analyze the packet payload. This allows signatures of known attacks to be used on the data to determine whether or not the packet payload contains an attack. Flow level information is provided by Cisco Net-Flow [12] and Argus [13] in the form of summarized connection information.

2.2.1.1 Algorithmic Approaches:

These approaches use methods such as hypothesis testing and probabilistic models, to detect port scan attacks based on analysis of network activity. Some of the most well known approaches are discussed below.

- (a) **Staniford-Chen et al [14]:** graph-based scan detection technique is a part of GrIDS, which uses packet level information to generate graphs that represent communication patterns observed on a network. It tries to detect and analyze large scale attacks and can be found capable of detecting attacks in individual hosts. It aggregates network activity of interest into an activity graph. They used a hierarchical reduction scheme for the construction of graphs, which is helpful in detecting large scale attacks. For example, a worm is indicated by a tree like structure while a scan is identified by a fan like structure representing one IP connecting to multiple IPs. It takes much time for aggregation and also the technique is not resistant to DoS attacks.
- (b) **Leckie and Kotagiri [15]:** The authors present an algorithm based on a probabilistic model. For each IP address in the monitored network, the algorithm generates a probability $P(d|s)$ that represents how likely a source will contact that particular destination IP, where d is the destination IP and s is the source, based on how commonly that destination IP is contacted by other sources, $P(d)$. Similarly, it also computes a probability for each port that represents how likely a source will contact a particular destination port, $P(p|s)$ where p is the destination port. A limitation of this approach is that $P(d)$ is based on the prior distribution of sources that have accessed that IP address. This implies that if the probabilities for this approach are generated based on a sample of network data, and if the monitored network is scanned, the resulting distributions may include scans as well as normal traffic. Another limitation of this approach is that it assumes that an attacker accesses the destinations at random; this may not be always true.

- (c) **Kim et al [16]**: This method aims to detect network port scans using anomaly detection. First, the method performs statistical tests to analyze traffic rates. Then, it makes use of two dynamic *chisquare* tests to detect anomalous packets. It models network traffic as a marked point process and introduces a general port scan model. The authors present simulation results to detect 10 malicious vertical scans with true positive rate greater than 90% and false positive rate smaller than 15% for both the static and dynamic tests using the port scan model and statistical tests.
- (d) **Kato et al [17]**: This approach aims to detect scans over large networks and is similar to GrIDS. However, it is further refined to evaluate only those connection attempts that result in a RST-ACK packet from the destination, indicating that the TCP service does not exist on the target IP address. During experiments in a 15-minute window, the method is able to identify a scan (consisting four or more destinations) returning RST-ACK packets to a single source. Given that a RST-ACK packet is only returned if the destination IP address has an active host, it is possible that scans of sparse networks are missed, since at best ICMP responses are returned rather than RSTACKs. Also, it misses those scans that are not TCP-based.
- (e) **Robertson et al [18]**: This method based on network return traffic, reconstructs sessions, and flags any source IP that is found to contact a destination for which no response is returned. An anomaly score is estimated for each source IP based on the number of destinations contacted where no response is observed. It can view almost all traffic in both directions. However, it may not be possible to use it on large networks due to asymmetric routing policies. The authors present a second method, called PSD, which has additional heuristics for analyzing traffic where there is the possibility that traffic for one direction is available; hence, no response does not necessarily indicate a scan.
- (f) **Ertoz et al [19]**: The authors develop a system called MINDS, that can analyze network traffic and can also detect port scan attacks. It reads Net-Flow data and generates data characteristics, including flow level information; e.g., source IP, source port, number of bytes, etc. It then derives information such as the number of connections from a single source, the number of connections to a single destination, the number of connections from a single source to the same port, and the number of connections from a single destination to the same source port. These four features are counted over a time window and over a connection window. An anomaly score is estimated based on the flow data and derived data for each network traffic record. A

report is generated ordered by anomaly score. The authors also claim that it can detect both fast and slow scanning.

- (g) **Gates et al [20]**: It analyzes Cisco Net-Flow data for port scan attacks. The method extracts the events (bursts of network activities surrounded by quiescent periods) for each source and the flows in each event are then sorted according to destination IP and destination port. It attempts to calculate six characteristics for each event based on statistical analysis of port scans. It estimates a probability using logistic regression with these six characteristics as input variables to predict whether the events contain a scan or not. The main drawback of the method is that it is non-real time.
- (h) **Porras and Valdes [21]**: This method is based on the EMERALD [22] system, which is used to detect port scan attacks. EMERALD considers each source IP address communicating with the monitored network as a subject. It constructs statistical profiles for subjects, and matches a short term weighted profile of subject behavior to a long term weighted profile. When the short term profile goes far enough into the tails of the distribution for the long term profile, EMERALD views it as suspicious. One aspect of subject behavior is the volume of network traffic of a particular kind generated. This can be used to detect port scanning as a sudden increase in the volume of SYN packets, for example, from a particular source IP.
- (i) **Udhayan et al [23]**: The authors report a heuristic approach for detecting port scan attacks. One possible solution to curb a zombie army or a malicious botnet attack is by detecting and blocking or dropping reconnaissance scans, i.e., port scans. They derive a set of heuristics for their detection, some quite crafty. It is written into the firewall and is triggered immediately after a port scan is detected, to drop packets with the IP address of the source of port scan for a pre-determined period. This detection approach is more user friendly than other approaches like SNORT [24].
- (j) **Gyorgy et al [25]**: The authors propose a model known as off-the-shelf classifier based on the data mining approach. Initially, it transforms network trace data into feature dataset with label information. Then, it selects Ripper, a fast rule based classifier, which is capable of learning rules from multi-model datasets and the results provided by it are easy to interpret. The authors successfully demonstrate that data mining models can encapsulate expert knowledge to create an adaptive algorithm that can substantially outperform the state-of-the-art for heuristic based scan detection in both precision and recall. Also, this technique is capable of detecting the scanners at an early stage.

- (k) **Haan [26]**: Haan presents a conceptual model of port scan detection and uses it to analyze the possibility of scan detection based on network layer header data only. The model uses different features based on the IP header list: source and destination IP addresses, datagram size, transport layer protocol field, fragmentation information, and the checksum. This model has been shown to be effective and robust in terms of size of the datasets and detection rate.
- (l) **Rong-sheng et al [27]**: This approach uses a new mechanism termed PSD is based on TCP packet anomaly evaluation. By learning the port distribution and flags of TCP packets arriving at the protected hosts, PSD can compute the anomaly score of each packet and effectively detect port scans including slow scans and stealthy scans. It shows that PSD has high detection accuracy and low detection latency.

2.2.1.2 Threshold-based Approaches:

These approaches examine events of interest X across a Y -sized time window to detect port scan attacks above certain thresholds [28]. The most commonly used parameter for detecting scans is the number of unique IP addresses contacted by a host. Several intrusion detection systems have been developed in the past couple of years in the public domain that use the threshold-based approach to detect anomaly. The approach requires the packet level information.

- (a) **Heberlein et al [11]**: NSM, which is designed based on the algorithmic approach, is considered to have pioneered the implementation of the threshold based scan detection approach [29]. This tool has three parts: data capturing, data analysis and support. The data analysis is the core part of the NSM [11]. It collects data in different forms such as statistical, session, full content and alert data. Statistical data represent the aggregation of network traffics, protocol breakdown and distribution. Session data represent the connection pairs, and conversation between two hosts. Full content data represent the log of every single bit of network traffic. Alert data represent the data collected by IDS. It recognizes a source as anomalous and potentially malicious if it is found to contact more than 15 other IP addresses during an unspecified period of time. It also identifies a source as anomalous if it tries to contact an IP address that does not contain a responding computer on the monitored network. With this last heuristic, it assumes that an external source would contact an internal IP address only for a reason backed by knowledge of the existence of a service at an internal IP address; for

example an FTP server, a mail server, etc. NSM is neither a security event management system nor a intrusion prevention system.

- (b) **Roesch [24]**: SNORT is a signature-based intrusion detection system. It uses a pre-processor that extracts port scans, based on either invalid flag combination (for example, NULL scans, Xmas scans, SYN-FIN scans) or on exceeding a threshold. SNORT uses a pre-processor, called *portscan* that watches connections to determine whether a scan is occurring. By default, SNORT is configured to generate an alarm only if it has detected SYN packets sent to at least five different IP addresses within 60 seconds or 20 different ports within 60 seconds, although this can be adjusted manually. By having such a high threshold, the number of false positives is reduced. However, a careful scan at a rate lower than the threshold can easily go undetected.
- (c) **Paxson [30]**: This detection system, also known as *Bro*, attempts to detect scans based on a thresholding approach. Network scans are detected when a single source contacts multiple destinations ($>$ some threshold). It also detects vertical scans when a single source contacts too many different ports. It assumes that the external site has initiated the conversation in both cases. However, a major limitation of this method is increased number of false positives. Bro uses payload as well as packet level information.
- (d) **Jung et al [29]**: The authors describe an approach called TRW based on sequential hypothesis testing. It detects port scans using an Oracle database that contains the assigned IP addresses and ports inside a network after performing an analysis of return traffic. When a connection request is received, the source IP is entered into a list, along with each destination to which this source has attempted a connection. If the current connection is to a destination which is already in the list, the connection is ignored. If it is to a new destination, it is added to the list, and a measure that determines whether the connection is scanning or not is computed and updated based on the status of the connection. The entire source is flagged as either scanning or not scanning depending on whether the measure has exceeded the maximum threshold or has dropped below the minimum threshold, respectively. It has been observed that benign activity rarely results in connections to hosts or services that are not available, whereas scanning activity often makes such connections, with the probability of connecting to a legitimate service dependent on the density of the target network.
- (e) **Fullmer and Romig**: The authors develop a flow analysis tool called *flow-dscan*. This tool examines flows for floods and port scans. Floods are identified by excessive

packets per flow. Port scans are identified by a source IP address contacting more than a certain threshold number of destination IP addresses or destination ports (only ports less than 1024 are examined) on a single IP address. To minimize the false alarm rate, this approach makes use of a suppress list consisting of IP addresses.

- (f) **Zhang and Fang:** In this paper, the authors propose a new port scan detection approach known as Time based Flow size Distribution Sequential hypothesis testing (TFDS) for highspeed transit networks where only unidirectional flow information is available. TFDS uses the main ideas of sequential hypothesis testing to detect scanners that exhibit abnormal access patterns in terms of FSD entropy. This paper makes a comparison with the state-of-the-art backbone port scan detection method TAPS in terms of efficiency and effectiveness using real backbone packet trace, and finds that TFDS performs much better than TAPS.
- (g) **Kong et al:** The authors present a scalable scheme for real-time port scan detection without keeping any per-flow state. Their method uses a double filter structure to find $\langle \text{SIP}, \text{SP} \rangle$ (SIP-source IP, SP-Source port) pairs which connect to more than N $\langle \text{DIP}, \text{DP} \rangle$ (DIP-destination IP, DP-Destination port) pairs in T amount of time. The authors test their scheme over real network traces and are able to find those over-threshold $\langle \text{SIP}, \text{SP} \rangle$ pairs with high accuracy. Finally, those over threshold sources are grouped as attack and reported immediately into the network defender.
- (h) **Gadge and Patil:** In this paper, the authors propose a method to identify possible port scans and try to gather additional information about the scanner or attacker such as probable location, operating system, etc. The scan detection system collects all the information and stores it to generate the reports in terms bar graphs. Analysis of stored data can be done in terms of: time and day by which type of scan was performed, from which IP the scan was performed, different ports, etc. Based on the analysis of the various parameters used, it can recognize and report the type of attack or scan performed during a time window. This method can detect scans coming from most of common scanners such as Angry IP, Nmap and MegaPing.

2.2.1.3 Soft Computing Approaches:

Soft computing includes important methods that provide flexible information processing for handling real-life ambiguous situations. Methods in soft computing exploit tolerance for imprecision and uncertainty, use approximate reasoning and partial truth in order to achieve

traceability, provide robustness and low-cost solutions to problems. Some soft computing based approaches for scan detection are discussed next.

- (a) **Basu and Streilein et al:** This approach includes an algorithm to detect low-profile probes and DoS attacks. A low-profile probe is defined to consist of ten or fewer connections, or when there are more than 59 seconds between connection attempts. To maintain connection states in the session that it observes and to read packets in real-time, the system monitors a bi-directional network link. It estimates the anomaly score for connections based on the likelihood of finding a particular connection or with the assumption that legitimate connections are more common and, hence, more normal, than scans or denial of service attacks. It uses an ANN based classifier to classify connections.
- (b) **Chen and Cheng:** The authors present a novel and fast port scan detection method based on PGA. The method can efficiently discover ports that are open most often. During genetic evolution, ports with more open times survive to the next generation with higher probabilities. This approach demonstrates that PGA-based port scan is efficient in average as well as worst cases. Sequential port scans are better in best cases only.
- (c) **El-Hajj et al:** The authors report on a fuzzy logic based port scan attack detection approach. They design a fuzzy logic controller and integrate it with SNORT. The new method, known as *fuzzy-based SNORT (FB-SNORT)* enhances the functionality of port scan detection. The authors use fuzzy logic for detection because: (i) Clear boundaries do not exist between normal and abnormal events, and (ii) Fuzzy logic rules help in smoothing the abrupt separation of normality and abnormality (i.e., anomaly). The authors experiment with both wired and wireless networks. Their method shows that applying fuzzy logic for scan detection adds to the accuracy of determining bad traffic. Moreover, it gives a rank for each type of port scanning attack.
- (d) **Liu et al:** Here, a method known as NBKE is used to identify flooding attacks and port scans from normal traffic. The method represents all known attacks in terms of traffic features. The method takes hand-identified traffic instances as training examples for the NBKE. This method achieves high accuracy in the detection of flooding attacks and port scan attacks. The authors show that the Kernel based Estimator can provide improved accuracy of 96.8% over the simple Naive Bayes estimator.

- (e) **Shafiq et al:** The authors report a comparative study of three classification schemes for automated port scan detection. These include a simple FIS that uses classical inductive learning, a Neural Network that uses the back propagation algorithm and an ANFIS that also employs the back propagation algorithm. They use two information theoretic features, namely entropy and KL divergence of port usage, to model network traffic behavior for normal user applications. The authors carry out an unbiased evaluation of these schemes using an endpoint based traffic dataset.

2.2.1.4 Rule-based Approaches:

Generally, a rule-based IDS analyzes traffic data passing through it and differentiates intrusive traffic behaviours from the normal. A rule-based IDS uses rules stored in its knowledge base to detect and take actions when anomaly occurs in the traffic or when there are unauthorized activities. Rule-based IDS must generate rules based on network activity for detecting anomaly. Some rule-based approaches are described below.

- (a) **Mahoney and Chan:** The PHAD system learns the normal range of values for all 33 fields in the Ethernet, IP, TCP, UDP, and ICMP headers. A score is assigned to each packet header field in the testing phase and the fields' scores are summed to obtain a packet's aggregate anomaly score. The authors evaluate PHAD-C32 using the packet header fields: *source IP*, *destination IP*, *source port*, *destination port*, *protocol type*, and *TCP flags*. Normal intervals for the six fields are learned from 5 days of training data. In the test data, field values not falling in the learned intervals are flagged as suspect. The top n packet score values are labeled anomalous. The value of n is varied over a range to obtain ROC curves. Another relevant work is proposed by Oke and Loukas. The authors propose a DoS detection approach which uses multiple Bayesian classifiers and RNN. Their method is based on measuring various instantaneous and statistical variables describing the incoming network traffic, acquiring a likelihood estimation and fusing the information gathered from the individual input features using likelihood averaging and different architectures of RNNs.
- (b) **Kim and Lee:** The authors suggest an ATCF to detect slow port scan attacks using fuzzy rules. ATCF acts as an intrusion prevention system disallowing suspicious network traffic. It manages traffic with a stepwise policy: (i) decreasing network bandwidth and then (ii) discarding traffic. The authors show that the abnormal traffic control framework can effectively detect slow port scan attacks using Fuzzy rules and a stepwise policy. Apart from these two, several other rule-based IDSs have been

discussed in the literature that are not included here due to being non-relevant to port scan attack detection.

2.2.1.5 Visual Approaches:

Some approaches present data to the user in a visual manner so that he or she can recognize scans by the patterns it generates. Such approaches detect and investigate port scans using packet level information, and flow level information. Some visual approaches are presented here.

- (a) Conti and Abdullah:** The authors use visualization to detect network events, including scans, using packet level information. They show that parallel coordinate plots can be used to illustrate relationships among ports and IP numbers. They also demonstrate how different attack tools (e.g., Nmap, SuperScan, Nessus, etc.) exhibit different fingerprint patterns. While they conclude that scans demonstrate identifiable patterns in visual data, they do not examine the limitations of such detection. For example, they do not examine how much traffic can be visualized at once before any scan traffic is obscured by normal traffic, and how this in turn affects how slowly an adversary would need to scan to remain undetected.
- (b) Lakkaraju et al:** NVisionIP, a visualization system based on bi-directional flow level data, has been found capable of detecting horizontal scans. It allows a user to quickly view all connection activity on a network since they appear as horizontal stripes.
- (c) Muelder et al:** PortVis, a tool designed for scan detection, uses summarized network traffic for each protocol and port for a user-specified time period. The summaries include the number of unique source addresses, the number of unique destination addresses, and the number of unique source-destination address pairs. A series of visualization techniques and drill-downs are used to determine whether the monitored traffic contains horizontal or vertical scans. It is unclear how well this algorithm scales to larger networks. Additionally, this approach requires a manual analysis of the visualizations, rather than an automated recognition of scans.
- (d) Abdullah et al:** This visualization technique for network traffic attempts to recognize attacks in real-time. It also uses an improved representation for detecting and responding to malicious activity based on port-based overviews. It combines stacked histograms of aggregate activity to facilitate drill-down operations for visualization of finer details. When network traffic becomes large and the variety in the port numbers

and IP address ranges becomes wide, it uses an appropriate scaling technique to provide finer details.

- (e) **Musa and Parish:** The authors describe prototype software that enables visualization alerts effectively in real-time. The prototype software incorporates various projections of the alert data in 3-dimensional displays. Filtering, drilldown, and playback of alerts at variable speeds are incorporated to strengthen analysis. The developers integrate a false alert classifier using a decision tree algorithm to classify alerts into false and true alerts. The authors also work on the analysis of both *portsweep* and *ntinfoscan* attacks.
- (f) **Lee et al:** This is an extended version of NVisionCC which is a clustering tool based on an extensible visualization framework. It exploits the nature of large-scale commodity clusters to improve illegal service detection mechanism. The cluster properties are only visible when one ceases to look at the cluster as a collection of disparate nodes. The tool can help make insightful observations by correlating open network ports observed on cluster nodes with other emergent properties such as the number and nature of active processes and the contents of important system files. This approach can greatly restrict the actions that an attacker can carry out undetected.
- (g) **Jiawan et al:** Scan-Viewer is a visual interactive network scan detection system designed to represent traffic activities that reside in network flows and their patterns. Scan-Viewer combines characteristics of network scans with novel visual structures, and utilizes a set of visual concepts to map the collected datagram to the graphs that emphasize their patterns. Additionally, it provides Local-port, a tool that captures large-scale port information. It has been experimentally shown that Scan-Viewer not only can detect network scans, port scans, distributed port scans, but also can detect hidden scans.

2.2.1.6 Discussion:

A large number of techniques for detection of port scans have been reported in this section under five distinct categories of approaches. However, it is not always easy to unambiguously classify a technique into any one of these approaches since often it uses elements from multiple classes. These approaches use features such as the source IP and port, destination IP and port, protocol, start time and end time of the session, and the number of bytes, and packets transferred. **Table 2.1** provides a summary of the scan detection approaches that are available for detecting the single-source port scan attacks. **Table 2.1** also shows the performances of those detection techniques wherever available and the datasets used for their evaluation.

Detection Approach	Name of the author(s)	Nature of Detection (Real/Nonreal time)	Packet(P) / Flow(F) level	Performance	
				False Positive (%)	Detection Rate (%)
Algorithmic	Staniford-Chen et al.	R	P	0.03	92 93.82
	Porras and Valdes	N	F		
	Kato et al.	R	P		
	Leckie and Kotagiri	R	P		
	Robertson et al.	N	P	4	
	Ertoz et al.	R	F		
	Kim et al.	R	P		
	Rong-sheng et al.	R	P	0.2	
	Gyorgy et al.	N	P		
	Haan	N	P		
	Gates et al.	R	F		
	Udhayan et al.	R	P		
Threshold	Heberlein et al.	N	P	0.96	
	Paxson	R	P		
	Roesch	R	P		
	Fullmer and Romig	R	F		
	Jung et al.	N	P	2.0	
	Kong et al.	R	P		
	Gadge and Patil	N	P		
	Zhang and Fang	R	F		
Soft	Streilein et al.	R	P	0.1	100

Computing	El-Hajj et al.	R	P		
	Liu et al.	N	P		
	Shafiq et al.	N	P		
	Chen and Cheng	N	P		
	Chen and Cheng	N	P		
Rule-based	Mahoney and Chan	N	P		
	Kim and Lee	N	p/F		
Visual	Conti and Abdullah	R	p/F	0.4	95.5
	Lakkaraju et al.	R	F		
	Muelder et al.	R	F		
	Abdullah et al.	R	F		
	Lee et al.	N	P		
	Musa and Parish	R	F		
	Jiawan et al.	N	F		

Table 2.1 Comparing Single-source port scan detection approaches

2.2.2 Distributed Scan Approaches:

The main goal of these approaches is to detect coordinated attacks. These types of attacks attempt to compromise a single host from multiple systems. There are various methods for detecting these attacks. Like the single-source scan detection approaches, based on the features used by the methods, the approaches also can be categorized into four classes: *Algorithmic, Clustering, Soft Computing, and Visual*.

2.2.2.1 Algorithmic Approaches:

Only few algorithmic approaches that operate in a distributed mode can be found in the literature. We report here two popular techniques which have been established to perform satisfactorily over multiple datasets.

- (a) **Gates:** This approach describes a model of potential adversaries based on the information they wish to obtain, where each adversary is mapped to a particular scan footprint pattern. The adversary model forms the basis of an approach to detect forms of coordinated scans, employing an algorithm that is inspired by heuristics for the set covering problem. The model also provides a framework for comparing various types of adversaries, different coordinated scan detection approaches might identify. The author evaluates the model to analyse the detector performance over a set of different datasets. Both the detection and false positive rates gathered from the experiments are 38behavio using regression equations.
- (b) **Whyte:** The author describes the design, implementation and evaluation of fully functional prototypes to detect internal and external scanning activity at an enterprise network. These techniques offer the possibility of identifying local scanning systems within an enterprise network after the observation of only a few scanning attempts with a low false positive and negative rates. To detect external scanning activity directed at a network, it makes use of the concept of exposure maps that are identified by passively characterizing the connectivity 38behaviour of internal hosts in a network as they respond to both legitimate connection attempts and scanning attempts. The exposure maps technique enables: (1) active response options to be safely focused exclusively on those systems that directly threaten the network, (2) the ability to rapidly characterize and group hosts in a network into different exposure profiles based on the services they offer, and (3) the ability to perform a RAA that determines what specific information was returned to an adversary as a result of a directed scanning campaign. Finally, the author experimented with real life scan activity as well as in offline datasets.

2.2.2.2 Clustering Approaches:

Clustering is the process for partitioning data into groups of similar objects. It is an unsupervised learning process. There are many approaches available for detecting network scans based on the similarity of the data, compactness of the cluster, and so on. Some approaches are discussed below.

- (a) **Streilein et al:** The approach uses a series of tables that maintain connection information (i.e., type of probe, source IP addresses, time, duration of the probes, etc.) about current sessions as well as alerts generated by probes. The table data are

analyzed based on a clustering approach to see, whether any of them form a distributed attack or not.

- (b) **Robertson et al:** The authors define a distributed port scan as a set of port scans that originate from source IP addresses that are located close together. In other words, they assume that a scanner is likely to use several IP addresses on the same subnet. This implies that if a particular IP address scans a network, IP addresses near this IP address, rather than those far away, are more likely to have also scanned the network.
- (c) **Yegneswaran et al:** This method can detect coordinated port scans where a distributed port scan is defined as a set of scans from multiple sources (i.e., 5 or more) aimed at a particular port of destinations within a one hour window. Based on this definition, the authors find that a large proportion of daily scans are coordinated in nature, with coordinated scans being roughly as common as vertical and horizontal scans. The system looks to see if different sources start and stop scanning either at the same time, or in very similar temporal patterns. There is little locality in IP space for these coordinated scanning sources. The authors do not discuss characteristics of the target hosts.
- (d) **Staniford et al:** This approach begins with an analysis of the stealthy port scan detection problem based on an intrusion correlation engine. The authors maintain records of event likelihood to estimate the anomalousness of a given packet. For effective detection performance, they use simulated annealing to cluster anomalous packets together into port scans based on heuristics developed from real scans. Packets that score high on anomalousness are kept around longer. They claim that the system is capable of detecting all scans detected by all other current techniques, plus many stealthy scans, with a manageable proportion of false positives.

2.2.2.3 Soft Computing Based Approaches:

In addition to those approaches discussed so far, there are several distributed scan detection approaches that use soft computing techniques. Next we discuss a few of these.

- (a) **Curtis et al:** The authors describe an intrusion response architecture based on intelligent agents to detect distributed port scans. The authors use a master analysis agent to find a confidence measure based on observed false positive rates. The master analysis agent can combine various alerts using a two-level fuzzy rule set to determine whether a current attack is a continuation of a previous attack, or a new attack. The agent considers characteristics of the attack such as the time between the incident

reports, IP addresses, the user name, and the program name. The details of the fuzzy logic employed are not provided in this article, nor are the results of any experiments indicating how well this algorithm performs.

- (b) **Zhang et al:** This distributed multilayer cooperative model for scan attack detection is composed of feature-based detection, scenario-based detection and statistics-based detection. A scan attack always happens at the network layer and the transport layer. The authors categorize scan techniques into three: port scan, bug scan, and detecting scan. The authors claim that the model not only detects common scan attacks or their variants, but can also detect some slow scan attacks, camouflage attacks and DoS attacks that use the TCP/IP protocol.
- (c) **RNN and Other Approaches:** To launch a DoS attack, the attacker or intruder attempts to identify victim machines for executing malicious programs by scanning over the Internet. The recent DoS attacks are practically distributed (DDoS). The attacker attempts to take control of a large number of victim machines and use them to send a large number of packets to a specific target. **Oke et al** present a denial-of-service attacks detection approach using multiple Bayesian classifiers and biologically inspired RNN (It is a probabilistic model, inspired by the spiking behaviour of neurons, with an elegant mathematical treatment that describes both its steady-state behaviour. It facilitates an efficient platform for learning algorithms in recurrent neural networks). **Oke et al** use RNN structure to fuse real-time networking statistical data and distinguish between normal and attack traffic during a DoS attack. The approach performs satisfactorily in terms of correct detections, missed detections and false alarms.

2.2.2.4 Visual Approaches:

These approaches are used for visualizing network traffic to detect whether the flow of network packet is an attack or normal behavior. One such commonly found approach is proposed by Conti and Abdullah. The approach (discussed in the context of single source port scan earlier) attempts to detect distributed scans against a background of normal traffic based on visualization. Due to lack of details, it is difficult to understand how a distributed scan would use this tool. Also, it is not clear how much traffic can be viewed at one time without obscuring features of interest.

2.2.2.5 Discussion:

Most distributed port scan detection approaches analyze packet level information. They can detect port scan attacks based on the IP addresses (source IP, destination IP), connection information, and port (source ports, destination ports) fields in the IP header. A general comparison of the distributed scan detection approaches discussed in this section is given in *Table 2.2*. It can be seen from column 4 of the table that most of these approaches are non real time and their performance are evaluated in terms of FPR (False Positive Rate) and DR (Detection Rate).

Detection Approach	Ref *	R/N *	P/F *	Performance	
				FPR *(%)	DR*(%)
Algorithmic	Gates	N	P		
	Whyte	R	P		
Clustering	Streilein et al	R	A	0.1	100
	Staniford et al	N	P		99.75
	Staniford et al				99.75
	Robertson et al	N	A	4	
	Yegneswaran et al	N	A		
Soft Computing	Curtis et al	N	A		80
	Zhang et al	N	P/F		
Visual	Conti and Abdullah	R	P		

Table 2.2 Comparing distributed port scan detection approaches and its comparative study

Note: Ref(Reference), R/N (Real time / Non-real time), P/F(Packet level/Flow level), FPR(False Positive Rate), DR(Detection Rate)

In a distributed slow port scanning, the number of ports being scanned by a single attacker is relatively small, thus being able to fool the IDS.

Another approach used to detect slow port scanning is the connection-based detection method. This method was suggested by **Dokas et al** and it relies on labelling network connections. Connection-based features are the same as time-based features that are used in the time-based detection method, but they are computed for the last N connections, i.e. they are connection windowed rather than time windowed. This connection window is mainly

utilized to cope with the slow port scans. However, similar to the attack used to deceive the time-based features, the attacker may choose to generate legitimate connections greater than the window size N between each successive port scans. [1]

2.3 Conclusion:

In this chapter we present an overview of some methods designed to detect port scanning and we describe the techniques used by attackers to make port scanning stealthier and difficult to detect. In the next chapter we propose a simple and efficient method for detecting slow port scanning and how our method is different from the traditional detection approaches.