Advance Access publication on 20 April 2011
doi:10.1093/comjnl/bxr035

Surveying Port Scans and Their Detection Methodologies

MONOWAR H. BHUYAN[1], D.K. BHATTACHARYYA[1,*] AND J.K. KALITA[2]

[1]*Department of Computer Science and Engineering, Tezpur University, Napaam, Tezpur, Assam, India*
[2]*Department of Computer Science, University of Colorado at Colorado Springs, CO 80933-7150, USA*
[*]*Corresponding author: dkb@tezu.ernet.in*

Scanning of ports on a computer occurs frequently on the Internet. An attacker performs port scans of Internet protocol addresses to find vulnerable hosts to compromise. However, it is also useful for system administrators and other network defenders to detect port scans as possible preliminaries to more serious attacks. It is a very difficult task to recognize instances of malicious port scanning. In general, a port scan may be an instance of a scan by attackers or an instance of a scan by network defenders. In this survey, we present research and development trends in this area. Our presentation includes a discussion of common port scan attacks. We provide a comparison of port scan methods based on type, mode of detection, mechanism used for detection and other characteristics. This survey also reports on the available data sets and evaluation criteria for port scan detection approaches.

Keywords: TCP/IP; UDP; OS fingerprinting; coordinated scanning

Received 21 May 2010; revised 10 February 2011
Handling editor: Erol Gelenbe

1. INTRODUCTION

The Internet is a complex entity comprising diverse networks, users and resources. Most users are oblivious to the design of the Internet and its components and only use the many available services. However, there is a small minority of advanced users who use their knowledge to explore potential system vulnerabilities [1]. Hackers can compromise vulnerable hosts as they can either partake of resources or use them as tools for attacks. The launch of an effective attack often begins with an earlier and deliberate process of analyzing potential victims' hosts or networks.

Scanning ports is an important technical information-gathering technique. On the basis of scan statistics on a real-life network, network defenders can identify malicious scans. A port scan is a method of determining whether particular services are available on a host or a network by observing responses to connection attempts [2]. A port scan can be described as being composed of '*hostile Internet searches for open "doors" or "ports", through which the intruders gain access to computers*'. These techniques consist of sending a message to a port and listening for an answer. The received response indicates port status and can be helpful in determining a host's operating system and other information relevant to launching a future attack. A vulnerability scan is similar, except that a positive response from the target results in further communication to determine whether the target is vulnerable to a particular exploit. As can be found in [3], most attacks are preceded by some form of scanning activity, particularly vulnerability scanning.

1.1. Port scan and its significance

Port scanning is designed to probe a network host for open ports and other services available. It is useful for system administrators and other network defenders to detect port scans as a useful technique for recognizing precursors to serious attacks. From the attacker's viewpoint, a port scan is useful for gathering relevant information for launching a successful attack. Thus, it is of considerable interest to attackers to determine whether or not the defenders of a network are scanning ports regularly. Defenders do not usually hide their identity during port scanning, whereas attackers do.

1.2. Port scanning and its types

Generally, machines are connected to a network and run many services that use TCP or UDP ports for communication with each other. An attacker generally follows the steps shown in Fig. 1 while launching an attack.

There are 65 536 standardly defined ports on a computer [4]. They can be categorized into three large ranges: (i) well-known ports (0–1023), (ii) registered ports (1024–49 151) and (iii) dynamic and/or private ports (49 152–65 535). Normally, a port scan does not directly damage the system, but potentially a port scan helps the attacker in finding those ports that are available to launch attacks. Essentially, a port scan consists of sending a message to each port, one at a time and listening for an answer. The kind of response received indicates whether the port is being used and can therefore be probed further for weakness to launch future attacks. Port scanning usually happens on TCP ports, i.e. ports that use a connection-oriented protocol; such ports return good feedback to the attacker. It also happens on UDP ports, but they provide connectionless services that may not readily give relevant information to attackers. Also, a UDP port may be easily blocked by network defenders. Following are the various types of port scans (shown in Fig. 2). Each of these scanning techniques is introduced in brief below.

(a) *Stealth scan:* Such a scan is designed to go undetected by auditing tools. It sends TCP packets to the destination host with stealth flags. Some of the flags are SYN, FIN and NULL.

(b) *SOCKS port probe:* A SOCKS port allows sharing of Internet connections on multiple machines. Attackers scan these ports because a large percentage of users misconfigure SOCKS ports, potentially permitting arbitrarily chosen sources and destinations to communicate. A SOCKS port on a system may allow the attacker to access other Internet hosts while hiding his or her true location.

(c) *Bounce scan:* It takes advantage of a vulnerability of the FTP protocol itself. Some applications that potentially allow bounce scans are email servers and HTTP Proxies.

(d) *TCP scanning:* A TCP connection is never fully established during this type of scanning. So, it is used by smart attackers. If the attacker can clearly know that a remote port is accepting connections, the attacker can launch an attack immediately. It is much more difficult for network defenders to detect since this kind of connection attempts are not logged by the server's logging system. Some TCP scans are TCP *Connect()*, reverse identification, Internet protocol (IP) header dump scan, SYN, FIN, ACK, XMAS, NULL and TCP fragment.

(e) *UDP scanning:* It attempts to find open ports related to the UDP protocol. However, UDP is a connectionless protocol and, thus, it is not often used by attackers since it can be easily blocked.

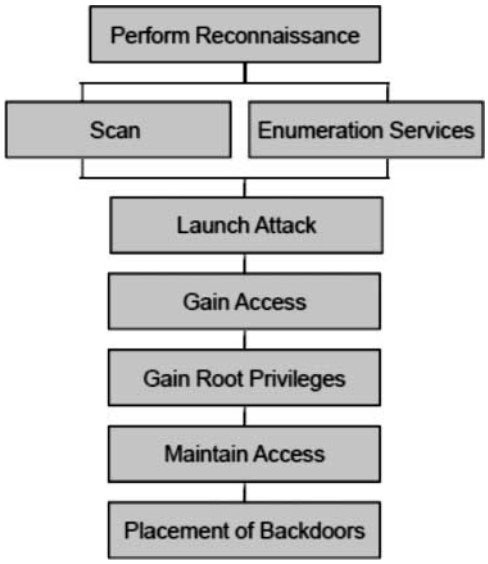

FIGURE 1. Steps in performing an attack.

We summarize the various port scan types discussed in this section along with firewall detection possibilities during the scanning process in Table 1. It can be seen from the last column of the table that, except the fist, third and fifth port scan types, the rest are not yet detectable at the firewall level.

1.3. Motivation

We are motivated to perform this survey in order to enumerate and compare the published single-source as well as distributed techniques used for port scan detection and understand their abilities as well as limitations. This survey builds on existing

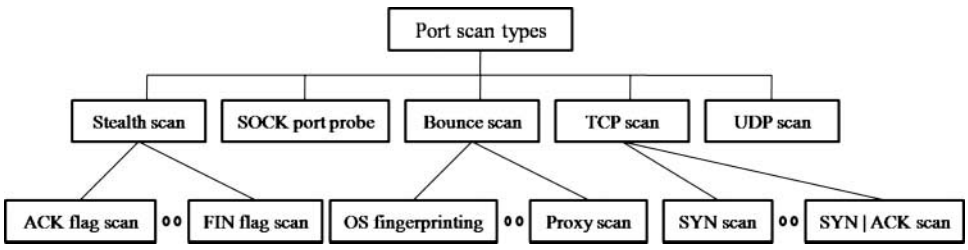

FIGURE 2. Types of port scans.

TABLE 1. Details of port scan types and its firewall level detection possibilities.

PST⋆	Protocol	TCP flag	VR⋆ (OP⋆)	VR⋆ (CP⋆)	FLDP⋆
TCP *Connect()*	TCP	SYN	ACK	RST	Yes
Reverse ident	TCP	No	No	No	No
SYN scan	TCP	SYN	ACK	RST	Yes
IP header dump scan	TCP	No	No	No	No
SYN|ACK scan	TCP	SYN|ACK	RST	RST	Yes
FIN scan	TCP	FIN	No	RST	No
ACK scan	TCP	ACK	No	RST	No
NULL scan	TCP	No	No	RST	No
XMAS scan	TCP	All flags	No	RST	No
TCP fragment	TCP	No	No	No	No
UDP scan	UDP	No	No	Port unreachable	No
FTP bounce scan	FTP	Arbitrary flag set	No	No	No

PST, port scanning technique; VR, victim's reply; OP, open port; CP, closed port; FLDP, firewall level detection possibility.

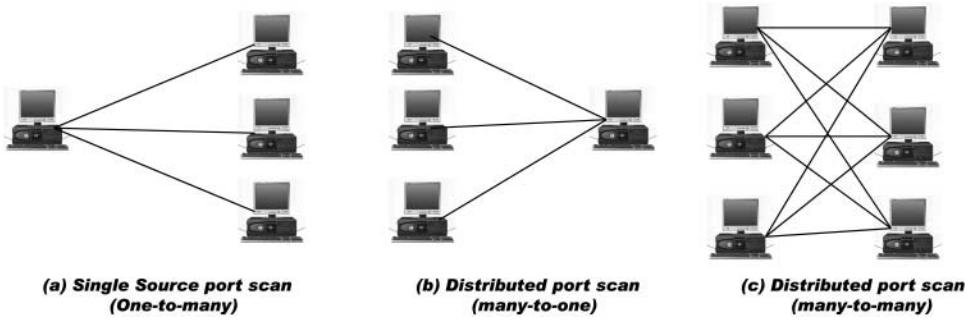

FIGURE 3. Single-source and distributed port scans.

works on port scan attack detection, significantly expanding the discussion in several directions. This survey can become the starting point for anyone trying to understand, evaluate, deploy or create port scan detection techniques.

1.4. Organization of the paper

The organization of the paper is as follows. Section 2 introduces port scan technologies, while in Section 3, we present a variety of port scan detection approaches. Section 4 describes the evaluation and performance analysis for port scan detection. In Section 5, we discuss research issues and challenges. Opportunities for future research and concluding remarks are presented in Section 6.

2. APPROACHES TO PORT SCANNING

On the basis of how scanning is performed, port scan techniques can be classified into two broad categories: *single-source* port scans and *distributed* port scans. Each of these categories is illustrated in Fig. 3 and discussed next.

2.1. Single-source port scans

The goal of port scanning from the perspective of an attacker is to gather ideas regarding where to probe for weaknesses. One can scan the network in a *one-to-many* fashion. As discussed in [5], a scan or any network attack can be detected by using the network intrusion detection system (IDS). In a pattern recognition-based scheme, attacks are discovered by matching network traffic with some known patterns. In [5], a decision tree-based detection technique is used for detecting scanning activity from netflow data. In the literature, a port scanner is defined as consisting of '*specialized programs used to determine what TCP ports of a host have processes listening on them for possible connections*' [2]. Staniford *et al.* [6, 7] further define the scan footprint as the set of ports or IP combinations that the attacker is interested in characterizing. According to them, port scans can be of four types (shown in Fig. 4): *vertical*, *horizontal*,

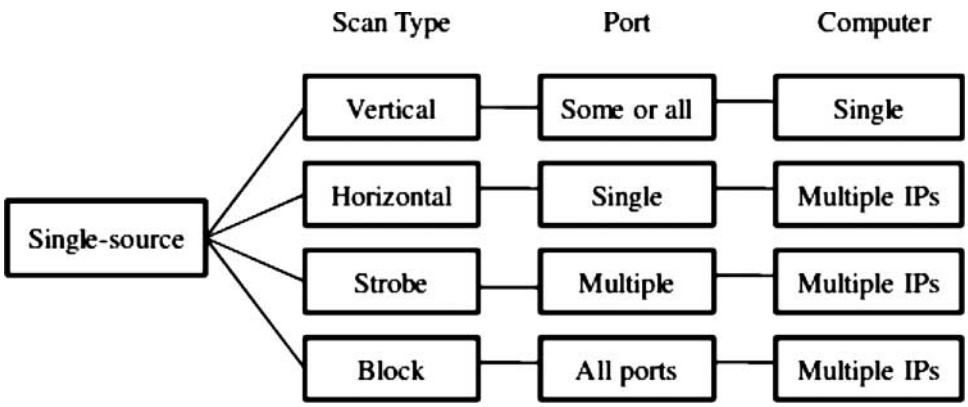

FIGURE 4. Single-source scan types with its port detail.

strobe and *block*. A *vertical* scan consists of a port scan of some or all ports on a single computer. The other three types of scans are used over multiple IP addresses. A *horizontal* scan is a scan of a single port across multiple IP addresses. If the port scan is of multiple ports across multiple IP addresses, it is called a *strobe* scan. A *block* scan is a port scan against all ports on multiple IP addresses. Yegneswaran *et al.* [8] quantified vertical and horizontal scans, defining a vertical scan as consisting of six or more ports on a single computer, and a horizontal scan as consisting of five or more IP addresses within a subnet.

2.2. Distributed port scans

Distributed information gathering is performed using either a *many-to-one* or a *many-to-many* model [9]. Here, the attacker utilizes multiple hosts to execute information-gathering techniques in different ways: *rate-limited* and *random* or *non-linear*. In a *rate-limited* information-gathering technique, the number of packets sent by a host to scan is limited [10]; this is based on the FreeBSD (BSD-Berkeley Software Distribution) implementation of Unix where separate rate limits are maintained for open ports as well as closed ports. For example: TCP RST rate limited, ICMP port unreachable rate limited and so on. On the other hand, a *random* or *non-linear* gathering technique refers to randomization of the destination IP-port pairs among the sources, as well as randomization of the time delay for each probe packet.

A *coordinated* attack has a more generic form of a distributed scan described by Staniford-Chen *et al.* [11]. It is defined as multi-step exploitation using parallel sessions with the objective of obscuring the unified nature of the attack or allowing the attackers to proceed more quickly. However, Green *et al.* [12] define a *coordinated* attack as '*multiple IP addresses working together toward a common goal*'. They also add that a coordinated attack can be viewed as multiple attackers working together to execute a distributed scan on many internal addresses or services. Staniford *et al.* [7] later define a distributed scan as one that is launched from a number of different real IP addresses, so that the scanner can investigate different parts of the footprint from different places. An attacker can scan the Internet using a few dozen to a few thousand zombies. A zombie is a compromised host, whose owner is unaware that the computer is being exploited (a remote attacker has accessed and set up to forward transmissions (spams or viruses) to other computers on the network) by the external party. Yegneswaran *et al.* [8] define *coordinated* scans as being scans from multiple sources aimed at a particular port of destinations within a 1-h window. These scans usually come from more aggressive or active sources that comprise several collaborative peers working in tandem. Finally, Robertson *et al.* [13] group source addresses together as forming a potentially distributed port scan if they are sufficiently close, where the scanner simply obtains multiple IP addresses from his Internet service provider. It should be noted that all of these definitions imply some level of coordination among the single sources used in the scan.

3. 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 [9] 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 Fig. 5.

3.1. Single-source port scan detection approaches

Detection approaches for single-source port scans have been part of IDSs since 1990, from the release of the network security monitor (NSM) [14]. 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

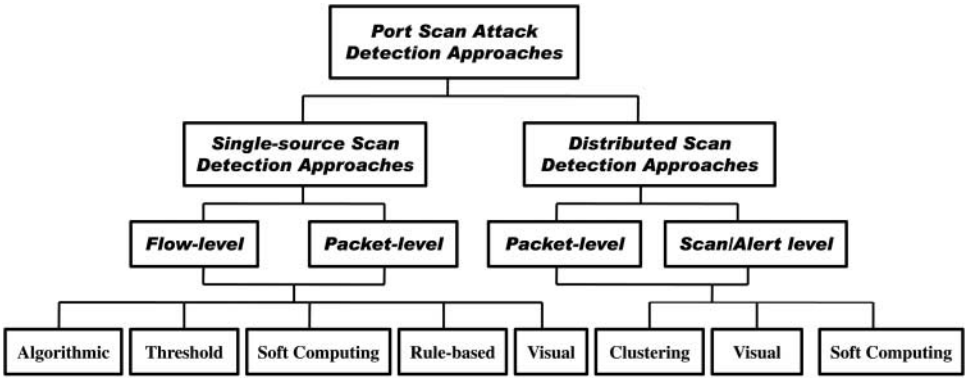

FIGURE 5. Hierarchy of port scan attack detection approaches.

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 NetFlow [15] and Argus [16] in the form of summarized connection information.

3.1.1. 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.* [11]: This graph-based scan detection technique is a part of GrIDS (Graph-based IDS), 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 denial-of-service (DoS) attacks.

(b) *Leckie and Kotagiri* [17]: 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 always be true.

(c) *Kim et al.* [18]: 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 *chi-square* 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 a true positive rate >90% and false positive rate (FPR) <15% for both the static and dynamic tests using the port scan model and statistical tests.

(d) *Kato et al.* [19]: This approach aims to detect scans over large networks and is similar to GrIDS [11]. However, it is further refined to evaluate only those connection attempts that result in an RST-ACK packet from the destination, indicating that the TCP service does not exist on the target IP address. During experiments in a 15-min window, the method is able to identify a scan (consisting four or more destinations) returning RST-ACK packets to a single source. Given that an 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 RST-ACKs. Also, it misses those scans that are not TCP based.

(e) *Robertson et al.* [13]: 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 (Peering center Surveillance Detection) 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.* [20]: The authors develop a system called Minnesota IDS that can analyze network traffic and can also detect port scan attacks. It reads NetFlow 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 [21] is estimated based on the flow data and derived data for each network traffic record. A report is generated ordered by the anomaly score. The authors also claim that it can detect both fast and slow scanning.

(g) *Gates et al.* [22]: It analyzes Cisco NetFlow 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 the destination IP and the 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* [23]: This method is based on the EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) [24] 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.* [25]: 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 [26].

(j) *Gyorgy et al.* [27]: The authors propose a model known as off-the-shelf classifier based on the data mining approach. Initially, it transforms network trace data into a feature data set with label information. Then, it selects Ripper, a fast rule-based classifier, which is capable of learning rules from multi-model data sets 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* [28]: 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 data sets and detection rate (DR).

(l) *Rong-sheng et al.* [29]: This approach uses a new mechanism termed PSD (Port Scan Detection) 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.

3.1.2. *Threshold-based approaches*

These approaches examine events of interest X across a Y-sized time window to detect port scan attacks above certain thresholds [30]. The most commonly used parameter for detecting scans is the number of unique IP addresses contacted by a host. Several IDSs 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.* [14]: NSM, which is designed based on the algorithmic approach, is considered to have pioneered the implementation of the threshold-based scan detection approach [31]. This tool has three parts: data capturing, data analysis and support. The data analysis is the core part of the NSM [14]. 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 an 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 the 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* [26]: SNORT is a signature-based IDS. 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 *port scan* 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 s or 20 different ports within 60 s, 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* [32]: 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 the increased number of false positives. Bro uses payload as well as packet-level information.

(d) *Jung et al.* [31]: The authors describe an approach called threshold random walk 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* [33]: 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 <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* [34]: In this paper, the authors propose a new port scan detection approach known as time-based flow size distribution sequential hypothesis testing (TFDS) for high-speed 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 flow size distribution entropy. This paper makes a comparison with the state-of-the-art backbone port scan detection method TAPS [35] in terms of efficiency and effectiveness using real backbone packet trace, and finds that TFDS performs much better than TAPS.

(g) *Kong et al.* [36]: 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 ⟨SIP, SP⟩ (SIP-source IP, SP-Source port) pairs which connect to more than N ⟨DIP, DP⟩ (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 ⟨SIP, SP⟩ pairs with high accuracy. Finally, those over-threshold sources are grouped as attack and reported immediately into the network defender.

(h) *Gadge and Patil* [37]: 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. On the basis of 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.

3.1.3. Soft computing approaches

Soft computing includes important methods that provide flexible information processing for handling real-life ambiguous situations [38]. Methods in soft computing exploit tolerance for imprecision and uncertainty, use approximate reasoning and partial truth in order to achieve traceability, and provide robustness and low-cost solutions to problems. Some soft computing-based approaches for scan detection are discussed next.

(a) *Basu et al. and Streilein et al.* [39, 40]: This approach includes an algorithm to detect low-profile probes and DoS attacks. A low-profile probe is defined to consist of 10 or fewer connections, or when there are more than 59 s 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 DoS attacks. It uses an artificial neural network-based classifier to classify connections.

(b) *Chen and Cheng* [41]: The authors present a novel and fast port scan detection method based on partheno-genetic algorithms (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.* [42]: 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* 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.* [43]: Here, a method known as naive Bayes kernel estimator (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.* [44]: The authors report a comparative study of three classification schemes for automated port scan detection. These include a simple fuzzy inference system (FIS) that uses classical inductive learning, a neural network that uses the back propagation algorithm and an adaptive neuro FIS (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 data set. This paper shows that ANFIS, though more complex, successfully combines the benefits of the classical FIS and neural network to achieve excellent classification accuracy.

3.1.4. Rule-based approaches

Generally, a rule-based IDS analyzes traffic data passing through it and differentiates intrusive traffic behaviors 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. A rule-based IDS must generate rules based on network activity for detecting anomaly. Some rule-based approaches are described below.

(a) *Mahoney and Chan* [45]: The PHAD (Packet Header Anomaly Detection) 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 receiver operating characteristic (ROC) curves. Another relevant work is proposed by Oke and Loukas [46]. The authors propose a DoS detection approach, which uses multiple Bayesian classifiers and random neural networks (RNNs). 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* [47]: The authors suggest an abnormal traffic control framework (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 ATCF 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.

3.1.5. *Visual approaches*

Some approaches present data to the user in a visual manner so that he or she can recognize scans by the patterns they generate. 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* [48]: 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 [49], SuperScan [50], Nessus [51], 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.* [52]: 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.* [53, 54]: 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.* [55]: 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* [56]: 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, drill-down 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.* [57]: This is an extended version of NVisionCC [58] which is a clustering tool based on an extensible visualization framework. It exploits the nature of large-scale commodity clusters to improve the 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.* [59]: ScanViewer is a visual interactive network scan detection system designed to represent traffic activities that reside in network flows and their patterns. ScanViewer 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 Localport, a tool that captures large-scale port information. It has been experimentally shown that ScanViewer not only can detect network scans, port

scans and distributed port scans, but also can detect hidden scans.

3.1.6. 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 provides a summary of the scan detection approaches that are available for detecting the single-source port scan attacks. Table 2 also shows the performances of those detection techniques wherever available and the data sets used for their evaluation.

TABLE 2. Comparing single-source port scan detection approaches.

				Performance	
Detection approach	Name of the author(s)	Nature of detection (real/non-real time)	Packet(P)/ flow (F) level	False positive (%)	Detection rate (%)
Algorithmic	Staniford-Chen *et al.* [11]	R	P		
	Porras and Valdes [23]	N	F		
	Kato *et al.* [19]	R	P		
	Leckie and Kotagiri [17]	R	P	0.03 [17]	
	Robertson *et al.* [13]	N	P	4 [13]	
	Ertoz *et al.* [20]	R	F		
	Kim *et al.* [18]	R	P		
	Rong-sheng *et al.* [29]	R	P	0.2 [29]	92 [29]
	Gyorgy *et al.* [27]	N	P		93.82 [27]
	Haan [28]	N	P		
	Gates *et al.* [22]	R	F		
	Udhayan *et al.* [25]	R	P		
Threshold	Heberlein *et al.* [14]	N	P		
	Paxson [32]	R	P		
	Roesch [26]	R	P		
	Fullmer and Romig [33]	R	F		
	Jung *et al.* [31]	N	P	0.96 [31]	
	Kong *et al.* [36]	R	P	2.0 [36]	
	Gadge and Patil [37]	N	P		
	Zhang and Fang [34]	R	F		
Soft computing	Streilein *et al.* [40]	R	P	0.1 [40]	100 [40]
	El-Hajj *et al.* [42]	R	P		
	Liu *et al.* [43]	N	P		
	Shafiq *et al.* [44]	N	P		
	Chen and Cheng [41]	N	P		
	Chen and Cheng [41]	N	P		
Rule-based	Mahoney and Chan [45]	N	P		
	Kim and Lee [47]	N	P/F		
Visual	Conti and Abdullah [48]	R	P/F		
	Lakkaraju *et al.* [52]	R	F		
	Muelder *et al.* [53]	R	F		
	Abdullah *et al.* [55]	R	F		
	Lee *et al.* [57]	N	P	0.4 [57]	95.5 [57]
	Musa and Parish [56]	R	F		
	Jiawan *et al.* [59]	N	F		

3.2. 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.

3.2.1. 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 data sets.

(a) *Gates* [30]: 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, that different coordinated scan detection approaches might identify. The author evaluates the model to analyze the detector performance over a set of different data sets. Both the detection and FPRs gathered from the experiments are modeled using regression equations.

(b) *Whyte* [60]: 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 behavior of internal hosts in a network as they respond to both legitimate connection attempts and scanning attempts. The exposure map 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 reconnaissance activity assessment 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 data sets.

3.2.2. 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.* [40]: 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 forms a distributed attack or not.

(b) *Robertson et al.* [13]: 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.* [8]: This method can detect coordinated port scans where a distributed port scan is defined as a set of scans from multiple sources (i.e. five or more) aimed at a particular port of destinations within a 1-h window. On the basis of 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.* [6, 7]: 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.

3.2.3. 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.* [61]: 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 FPRs. 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.* [62]: This distributed multi-layer 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* [63–65]: 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.* [64] present a DoS attacks detection approach using multiple Bayesian classifiers and biologically inspired recurrent neural networks (RNNs). A detailed discussion of RNNs and their extensions can be found in [66–68]. It is a probabilistic model, inspired by the spiking behavior of neurons, with an elegant mathematical treatment that describes its steady-state behavior. It facilitates an efficient platform for learning algorithms in RNNs [68]. Oke *et al.* [64] 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. In [63], an autonomic approach to DoS defence is presented. It drops the attacking packets adaptively from the node being attacked using trace-back of DoS flows. This allows paths being used by normal and attack flows to be identified, and also helps legitimate flows to find robust paths during an attack. In [65], a hybrid approach is presented by combining the approaches [63, 64] mentioned above. It is based on the observation of the incoming traffic and a combination of traditional likelihood estimation with a recurrent RNN (r-RNN) structure. They select input features that describe essential information about the incoming traffic and evaluate the likelihood ratios for each input to fuse them with an r-RNN.

3.2.4. *Visual approaches*

Theses approaches are used for visualizing network traffic to detect whether the flow of network packets is an attack or normal behavior. One such commonly found approach is proposed by Conti and Abdullah [48]. 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.

3.2.5. *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, 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 3. It can be seen from column 4 of the table that most of these approaches are non-real time and their performances are evaluated in terms of the FPR and the DR.

TABLE 3. Comparing distributed port scan detection approaches and its comparative study.

Detection approach	Ref⋆	R/N⋆	P/F⋆	Performance	
				FPR⋆(%)	DR⋆(%)
Algorithmic	[30]	N	P		
	[60]	R	P		
Clustering	[40]	R	A	0.1 [40]	100 [40]
	[6]	N	P		99.75 [6]
	[7]				[7]
	[13]	N	A	4 [13]	
	[8]	N	A		
Soft computing	[61]	N	A		80 [61]
	[62]	N	P/F		
Visual	[48]	R	P		

Ref, reference; R/N, real time/non-real time; P/F, packet level/flow level; FPR, false positive rate; DR, detection rate.

4. EVALUATION AND PERFORMANCE ANALYSIS

Evaluating an attack detection system is a difficult task due to several reasons. First, it is difficult to get high-quality data sets for performing evaluation due to privacy and competitive issues. Secondly, even if real-life data sets were available, labeling network connections as normal or attack requires an enormous amount of time for human experts. Thirdly, the constant change of network traffic cannot only introduce new types of intrusions but can also change aspects of *normal* behavior, thus making construction of useful benchmarks even more difficult. Finally, when measuring the performance of an attack detection system, there is a need to measure not only the DR (i.e. how many attacks we detect correctly), but also the false alarm rate (i.e. how many normal connections we incorrectly detect as attacks) as well as the cost of misclassification. In spite of all these issues, in the following we talk about commonly used evaluation data sets and criteria for evaluating scan detectors and analysis in terms of complexity and ROC curves.

4.1. Evaluation data sets

Different data sets have been used for experimenting with security analysis. The data sets have been created from background network traffic or real-life network traces. Some of these are discussed below.

4.1.1. Lawrence Berkeley National Laboratory data sets

Lawrence Berkeley National Laboratory background traffic. This data set can be obtained from the Lawrence Berkeley National Laboratory (LBNL) in the USA. Traffic in this data set is composed of packet-level incoming, outgoing and internally routed traffic streams at the LBNL edge routers. Traffic was anonymized using the *tcpmkpub* tool [69]. The main applications observed in internal and external traffic are Web (i.e. HTTP), email and name services. Some other applications like Windows services, network file services and backup were used by internal hosts. The details of each service, information on each service's packets and other relevant description are given in [70]. Some statistics on the background network traffic of the LBNL data set are shown in Table 4.

LBNL attack traffic. The data set identifies attack traffic by isolating scans in aggregate traffic traces. Scans are identified by flagging those hosts which unsuccessfully probe more than 20 hosts, out of which 16 hosts are probed in ascending or descending order [69]. Malicious traffic mostly consists of failed incoming TCP SYN requests, i.e. TCP port scans targeted toward LBNL hosts. However, there are also some outgoing TCP scans in the data set. Most UDP traffic observed in the data (incoming and outgoing) is composed of successful connections, i.e. host replies received for UDP flows. Clearly, the attack rate is significantly lower than the background traffic rate. The attack traffic in this data set is reported in Table 4. Complexity and privacy were two main reservations for the participants of the endpoint data collection study. To address these reservations, the authors developed a custom multi-threaded MS Windows tool using the *Winpcap* API [71] for data collection. To reduce packet logging complexity at the endpoints, they only logged very elementary session-level information (bi-directional communication between two IP addresses on different ports) of TCP and UDP packets. To ensure user privacy, an anonymization policy was used.

4.1.2. Endpoint data sets

Endpoint background traffic: In the endpoint context, we see in Table 5 that home computers generate significantly higher traffic volumes than office and university computers because: (i) they are generally shared between multiple users and (ii) they run peer-to-peer and multimedia applications. The large traffic volumes of home computers are also evident from their high mean number of sessions per second. To generate attack traffic, the developers infected virtual machines on the endpoints with different malware: Zotob.G, Forbot-FU, Sdbot-AFR, Dloader-NY, So-Big.E@mm, MyDoom.A@mm, Blaster, Rbot-AQJ and RBOT.CCC. Details of the malware can be found in [72]. Characteristics of the attack traffic in this data set are given in Table 6. These malware have diverse scanning rates and attack ports or applications.

Endpoint attack traffic. The attack traffic logged at the endpoints is mostly composed of outgoing port scans. Note that this is the opposite of the LBNL data set, in which most of the attack traffic is inbound. Moreover, the attack traffic rates at the endpoints are generally much higher than the background traffic rates of LBNL data sets. This diversity in attack direction and rates provides a sound basis for performance comparison among scan detectors. For each malware, attack traffic of a 15-min duration was inserted in the background traffic for each

TABLE 4. Background and attack traffic information for the LBNL data sets.

Date	Duration (mins)	LBNL hosts	Remote hosts	Background traffic rate (packet/s)	Attack traffic rate (packet/s)
4 October 2004	10	4767	4342	8.47	0.41
15 December 2004	60	5761	10 478	3.5	0.061
16 December 2004	60	5210	7138	243.83	72

TABLE 5. Background traffic information for four endpoints with high and low rates.

Endpoint ID	Endpoint type	Duration (months)	Total sessions	Mean session rate per second
3	Home	3	373 009	1.92
4	Home	2	444 345	5.28
6	University	9	60 979	0.19
10	University	13	152 048	0.21

TABLE 6. Endpoint attack traffic for two high and two low-rate worms.

Malware	Release date	Average scan rate per second	Port (s) used
Dloader-NY	July 2005	46.84	TCP 1,35,139
Forbot-FU	September 2005	32.53	TCP 445
Rbot-AQJ	October 2005	0.68	TCP 1,39,769
MyDoom-A	January 2006	0.14	TCP 3127–3198

endpoint at a random time instance. This operation was repeated to insert 100 non-overlapping attacks of each worm inside each endpoint's background traffic.

4.2. Evaluation criteria and analysis

In this section, we discuss the various criteria and analysis measures for evaluating the scan detection techniques. Most of these analysis criteria attempt to evaluate a scan detection technique in terms of *DR*, *FPR*, *true positive rate*, *F-measure* and *ROC*.

4.2.1. Metric

There are four metrics commonly used in the intrusion detection community. The first two are the *DR* and *FPR*. These two are also known as the true and false negative rates, respectively. These two metrics are affected by what is known as the base rate fallacy [73]. The base rate fallacy for intrusion detection is related to the volume of normal traffic compared with the volume of attack traffic. Given that attacks are fairly rare when compared with the volume of normal traffic, even when the FPR is quite low (e.g. 1% or less), the result is that an analyst might still be overwhelmed by the number of false positives. Two other metrics, *effectiveness* and *efficiency*, are defined by Staniford *et al.* [6]. *Electiveness* is defined as the ratio of detected scans (i.e. true positives) to all scans (true positives plus false negatives). *Efficiency* is defined as the ratio of the number of identified scans (i.e. true positives) to all cases flagged as a scan (true positives plus false positives), and is the same as the DR defined previously.

4.2.2. ROC analysis

ROC curves are often used to evaluate the performance of an anomaly detector. A detailed discussion on this analysis in evaluating anomaly-based detection systems can be found in [74]. An ROC curve has the FPR on its X-axis and the true positive rate on its Y-axis, thus moving from (0, 0) at the origin to (1, 1). The detection system must return a likelihood score between 0 and 1 when it detects an intrusion in order for the ROC curve to provide meaningful results.

4.2.3. Complexity and delay comparison

To evaluate the performance of an anomaly detector, one calculates the time taken by it for training and classification and also the runtime memory requirement. One such tool for this purpose is *hprof* [75]. However, contrary to common intuition, complexity does not translate directly into accuracy of an anomaly detector. A delay value of ∞ is listed if an attack is not detected altogether. It has been observed that detection delay is reasonable (<1 s) for commonly available anomaly detectors.

5. RESEARCH ISSUES AND CHALLENGES

There are three most important qualities that need to be measured while evaluating an IDS: *completeness*, *correctness* and *performance or timeliness*. Evaluation is limited by the quality of the data set that the system is measured against. Port scan attack detection methods are very limited in the degree to which they can quantify their *completeness*, *correctness* and *performance* or *timeliness*. Some important research issues in this area are listed below.

(i) Most existing attacks, especially those belonging to many-to-one or many-to-many categories, cannot be controlled at the firewall level. For example, the TCP *connect*(), SYN, SYN | ACK scan can be blocked at the firewall level, whereas controlling the other scanning techniques at the firewall level is an important issue.

(ii) The existing methods have been found to work either at the packet level or the flow level or both. However, our survey finds that most detection approaches use packet-level information for attack detection because it not only gives the connection information but can also analyze the packet payload. However, an appropriate technique for packet analysis based on both header and payload information toward the detection of known as well as unknown attacks is still called for.

(iii) On the basis of our analysis of existing methods, threshold-based methods have been found to be more effective. These threshold-based methods are highly sensitive to input parameters (thresholds) and their estimations are often found to be network scenario

dependent. Therefore, development of a generic threshold-based detection mechanism across different network scenarios is a challenging issue.

(iv) With the evolving nature of networking technology and with the constant effort of attackers toward launching newer attacks, existing IDSs are often non-adaptive and hence inadequate for handling known as well as unknown attacks.

(v) On the basis of our analysis, we find that security practitioners have both positive and negative perceptions about port scan attack detection methods. In particular, practitioners find it difficult to decide where to place the attack detection module and how to best configure them for use within an environment with multi-stage architecture.

(vi) Due to the voluminous size of network traffic data and the constant changing of traffic patterns as well as the presence of the noise in audit data, it is a challenging task to build a normal profile of network behavior. Further research toward finding appropriate machine learning or soft computing methods in this regard is necessary.

(vii) Due to lack of availability of labeled data sets for training or validation of the models, most scan detection approaches result in many false alarms that require attention. Thus, minimization of false alarm is a challenging issue.

(viii) Network traffic has a large amount of data. If the security models generate profiles for the normal as well as attack traffic, it is a challenging task to update the signatures database dynamically

6. CONCLUSIONS

In this paper, we have examined the state of modern port scan detection approaches. The discussion follows well-known criteria for categorizing scan detection approaches: detection strategy, data source and data visualization. Experiments demonstrate that for different types of port scan attacks, different anomaly detection schemes may be more successful. Research prototypes combining data mining and threshold-based analysis for scan detection have shown great promise. Such detection approaches tend to have lower FPRs, scalability and robustness.

ACKNOWLEDGEMENTS

The authors are thankful to DIT for funding the project. The authors also thank the esteemed reviewers for their extensive comments on the final draft of the article.

FUNDING

This work is one of the outcomes of a research project funded by Department of Information Technology, Government of India.

REFERENCES

[1] Lee, C.B., Roedel, C. and Elena, S. (2003) Detection and Characterization of Port Scan Attacks. Technical Report, University of California, San Diego, CA. http://cseweb.ucsd.edu/users/clbailey/PortScans.pdf.

[2] De Vivo, M., Carrasco, E., Isern, G. and de Vivo, G.O. (1999) A review of port scanning techniques. *SIGCOMM Comput. Commun. Rev.*, **29**, 41–48.

[3] Panjwani, S., Tan, S. and Jarrin, K.M. (2005) An Experimental Evaluation to Determine If Port Scans are Precursors to an Attack. *Proc. DSN'05*, Washington, DC, USA, June 28–July 1, pp. 602–611. IEEE Computer Society.

[4] Mateti, P. (2010) *Lecture Notes on Internet Security*. Wright State University.

[5] Gregr, M. (2010) Portscan Detection Using Netflow Data. *Proc. EEICT'10*, Brno, CZ, pp. 229–233. Faculty of Information Technology BUT.

[6] Staniford, S., Hoagland, J.A. and McAlerney, J.M. (2000) Practical Automated Detection of Stealthy Portscans. *Proc. CCS'00*, Athens, Greece, November 1, pp. 1–4. ACM.

[7] Staniford, S., Hoagland, J.A. and McAlerney, J.M. (2002) Practical automated detection of stealthy portscans. *J. Comput. Secur.*, **10**, 105–136.

[8] Yegneswaran, V., Barford, P. and Ullrich, J. (2003) Internet intrusions: global characteristics and prevalence. *SIGMETRICS Perform. Eval. Rev.*, **31**, 138–147.

[9] hybrid@hotmail.com (1999) Distributed information gathering. *Phrack Mag., Article 9*, **9**.

[10] Ensafi, R., Park, J.C., Kapur, D. and Crandall, J.R. (2010) Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking. *Proc. USENIX Security'10*, Washington, DC, USA, pp. 257–272. USENIX Association.

[11] Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J., Levitt, K., Wee, C., Yip, R. and Zerkle, D. (1996) Grids: A Graph Based Intrusion Detection System for Large Networks. *Proc. 19th NISS'96*, Baltimore, MD, USA, October, pp. 361–370. NIST.

[12] Green, J., Marchette, D., Northcutt, S. and Ralph, B. (1999) Analysis Techniques for Detecting Coordinated Attacks and Probes. *Proc. WIDNM'99*, Santa Clara, CA, USA, April 9–12, pp. 1–9. USENIX Association.

[13] Robertson, S., Siegel, E.V., Miller, M. and Stolfo, S.J. (2003) Surveillance Detection in High Bandwidth Environments. *Proc. DARPA DISCEX III'03*, Washington, DC, USA, April 22–24, pp. 130–139. IEEE Computer Society.

[14] Heberlein, T., Dias, G., Levitt, K., Mukherjee, B., Wood, J. and Wolber, D. (1990) A Network Security Monitor. *Proc. RISP'90*, Oakland, CA, USA, May 7–9, pp. 296–304. IEEE Computer Society.

[15] Fyodor (1997) The art of port scanning. *Phrack Mag., Article 11*, **7**.

[16] QoSient. Argus. http://www.qosient.com/argus/.

[17] Leckie, C. and Kotagiri, R. (2002) A Probabilistic Approach to Detecting Network Scans. *Proc. NOMS'02*, Florence, Italy, April 15–19, pp. 359–372. IEEE Computer Society.

[18] Kim, H., Kim, S., Kouritzin, M.A. and Sun, W. (2004) Detecting Network Portscans Through Anomaly Detection. *Proc. SPIE 5429*, Orlando, FL, USA, April 12, pp. 254–263.

[19] Kato, N., Nitou, H., Ohta, K., Mansfield, G. and Nemoto, Y. (1999) A real-time intrusion detection system (ids) for large scale networks and its evaluations. *IEICE Trans. Commun.*, **E82-B**, 1817–1825.

[20] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.-N., Dokas, P., Kumar, V. and Srivastava, J. (2003) Detection of Novel Network Attacks Using Data Mining. *Proc. ICDM WDMCS'03*, Melbourne, FL, USA, November 19, pp. 30–39.

[21] Chandola, V., Banerjee, A. and Kumar, V. (2009) Anomaly detection: A survey. *ACM Comput. Surv.*, **41**, 1–58.

[22] Gates, C., McNutt, J.J., Kadane, J.B. and Kellner, M. (2006) Scan Detection on Very Large Networks Using Logistic Regression Modeling. *Proc. ISCC'06*, Pula-Cagliari, Sardinia, Italy, June 26–29, pp. 402–408. IEEE Computer Society.

[23] Porras, P. and Valdes, A. (1998) Live Traffic Analysis of TCP/IP Gateways. *Proc. ISOC NDSS'98*, San Diego, CA, USA, March. ISOC Press.

[24] Porras, P.A. and Neumann, P.G. (1997) EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. *Proc. NCSC'97*, Menlo Park, CA 94025, USA, October 22–25, pp. 353–365. NIST.

[25] Udhayan, J., Prabu, M.M., Krishnan, V.A. and Anitha, R. (2009) Reconnaissance Scan Detection Heuristics to Disrupt the Pre-attack Information Gathering. *Proc. N2S'09*, Paris, France, June 24–26, pp. 1–5. IEEE Computer Society.

[26] Roesch, M. (1999) Snort-lightweight Intrusion Detection for Networks. *Proc. LISA'99*, Seattle, WA, USA, November 7–12, pp. 229–238. USENIX Association.

[27] Gyorgy, S.U., György, J.S. and Hui, X. (2005) Scan Detection: A Data Mining Approach. *Proc. SIAM ICDM'05*, Sutton Place Hotel, Newport Beach, CA, USA, April 21–23, pp. 118–129. SIAM.

[28] Haan, G.-H.K. (2005) Detection of Portscans Using IP Header Data. *Proc. TBRC'05*, Enschede, January 21.

[29] Rong-sheng, S., Xiao-yong, L. and Jian-hua, L. (2004) An adaptive algorithm to detect port scans. *J. Shanghai Univ. (Engl. Ed.)*, **8**, 328–332.

[30] Gates, C. (2006) Co-ordinated port scans: a model, a detector and an evaluation methodology. PhD Thesis, Dalhousie University Halifax, Nova Scotia.

[31] Jung, J., Paxson, V., Berger, A.W. and Balakrishnan, H. (2004) Fast Portscan Detection Using Sequential Hypothesis Testing. *Proc. SECPRI'04*, Oakland, CA, USA, May 9–12, pp. 211–225. IEEE Computer Society.

[32] Paxson, V. (1998) Bro: A System for Detecting Network Intruders in Real-Time. *Proc. USENIX Security Symp.'98*, San Antonio, TX, USA, January 26–29, pp. 2435–2463. USENIX Association.

[33] Fullmer, M. and Romig, S. (2000) The OSU Flow-Tools Package and Cisco Netflow Logs. *Proc. LISA'00*, New Orleans, LA, USA, December 3–8, pp. 291–303. USENIX Association.

[34] Zhang, Y. and Fang, B. (2009) A Novel Approach to Scan Detection on the Backbone. *Proc. ITNG'09*, Washington, DC, USA, April 27–29, pp. 16–21. IEEE Computer Society.

[35] Sridharan, A., Ye, T. and Bhattacharyya, S. (2006) Connectionless Port Scan Detection on the Backbone. *Proc. IPCCC'06*, Phoenix, AZ, USA, April 10–12, pp. 567–576. IEEE Computer Society.

[36] Kong, S., He, T., Shao, X., An, C. and Li, X. (2006) Scalable Double Filter Structure for Port Scan Detection. *Proc. ICC'06*, Istanbul, Turkey, June 11–15, pp. 2177–2182. IEEE Computer Society.

[37] Gadge, J. and Patil, A.A. (2008) Port Scan Detection. *Proc. ICON'08*, Habitat World, IHC, New Delhi, India, December 12–14, pp. 1–6. IEEE Computer Society.

[38] Zadeh, L.A. (1994) Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, **37**, 77–84.

[39] Basu, R., Cunningham, R.K., Webster, S.E. and Lippmann, R.P. (2001) Detecting Low-profile Probes and Novel Denial-of-Service Attacks. *Proc. IWIAS'01*, West Point, NY, USA, June, pp. 5–10. IEEE Computer Society.

[40] Streilein, W.W., Cunningham, R.K. and Webster, S.E. (2002) Improved Detection of Low-profile Probe and Denial-of-Service Attacks. *Proc. Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, USA, June 11–13, pp. 11–13.

[41] Chen, J.J. and Cheng, X.J. (2009) A Novel Fast Port Scan Method Using Partheno-genetic Algorithm. *Proc. ICCSIT'09*, Los Alamitos, CA, USA, August 8–11, pp. 219–222. IEEE Computer Society.

[42] El-Hajj, W., Aloul, F., Trabelsi, Z. and Zaki, N. (2008) On Detecting Port Scanning Using Fuzzy Based Intrusion Detection System. *Proc. IWCMC'08*, Crete Island, Greece, August 6–8, pp. 105–110. IEEE Computer Society.

[43] Liu, D., Zhang, M.W. and Li, T. (2008) Network Traffic Analysis Using Refined Bayesian Reasoning to Detect Flooding and Port Scan Attacks. *Proc. ICACTE'08*, Phuket, Thailand, December 20–22, pp. 1000–1004. IEEE Computer Society.

[44] Shafiq, M.Z., Farooq, M. and Khayam, S.A. (2008) A Comparative Study of Fuzzy Inference Systems, Neural Networks and Adaptive Neuro Fuzzy Inference Systems for Portscan Detection. *Proc. EVO'08*, Naples, Italy, December, pp. 52–61. Springer.

[45] CS-2001-04 (2001) PHAD: Packet Header Anomaly Detection for Indentifying Hostile Network Traffic. Department of Computer Sciences, Florida Technical Report, Melbourne, FL 32901.

[46] Oke, G. and Loukas, G. (2007) A denial of service detector based on maximum likelihood detection and the random neural network. *Comput. J.*, **50**, 717–727.

[47] Kim, J. and Lee, J.H. (2008) A Slow Port Scan Attack Detection Mechanism Based on Fuzzy Logic and a Stepwise Policy. *Proc. IET ICIE'08*, University of Washington, Seattle, USA, July 21–22, pp. 1–5. IEEE Computer Society.

[48] Conti, G. and Abdullah, K. (2004) Passive Visual Fingerprinting of Network Attack Tools. *Proc. VizSEC/DMSEC'04*, Washington, DC, USA, October, 29, pp. 45–54. ACM.

[49] Fyodor. Nmap. http://nmap.org/.

[50] *FoundStone, a division of McAfee*. Superscan. http://www.foundstone.com/us/resources/proddesc/superscan.htm.

[51] Tenable Network Security Inc. Columbia. Nessus. http://www.nessus.org/nessus/.

[52] Lakkaraju, K., Yurcik, W. and Lee, A.J. (2004) NVisionIP: Netflow Visualizations of System State for Security Situational Awareness. *Proc. VizSEC/DMSEC'04*, Washington, DC, USA, October 29, pp. 65–72. ACM.

[53] Muelder, C., Ma, K.-L. and Bartoletti, T. (2006) Interactive Visualization for Network and Port Scan Detection. *LNCS,*

RAID'05, Seattle, WA, USA, September 7–9, pp. 265–283. Springer, Berlin.

[54] McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T. and Christensen, M. (2004) PortVis: A Tool for Port-Based Detection of Security Events. *Proc. VizSEC/DMSEC'04*, Washington, DC, USA, October 29, pp. 73–81. ACM.

[55] Abdullah, K., Lee, C., Conti, G. and Copeland, J.A. (2005) Visualizing Network Data for Intrusion Detection. *Proc. IEEE IAW'05*, West Point, NY, USA, June, pp. 100–108. IEEE Computer Society.

[56] Musa, S. and Parish, D.J. (2007) Visualising Communication Network Security Attacks. *Proc. IV'07*, Washington, DC, USA, July 4–6, pp. 726–733. IEEE Computer Society.

[57] Lee, A.J., Koenig, G.A., Meng, X. and Yurcik, W. (2005) Searching for Open Windows and Unlocked Doors: Port Scanning in Large-Scale Commodity Clusters. *Proc. CCGRID'05*, Cardiff, UK, May 9–12, pp. 146–151. IEEE Computer Society.

[58] Yurcik, W., Meng, X. and Kiyanclar, N. (2004) NVisionCC: A Visualization Framework for High Performance Cluster Security. *Proc. VizSEC/DMSEC'04*, New York, NY, USA, October 29, pp. 133–137. ACM.

[59] Jiawan, Z., Liang, L., Liangfu, L. and Ning, Z. (2008) A Novel Visualization Approach for Efficient Network Scans Detection. *Proc. SECTECH'08*, Horizon Resort, Sanya, Hainan Island, China, December 13–15, pp. 23–26. IEEE Computer Society.

[60] Whyte, D. (2008) Network scanning detection strategies for enterprise networks. PhD Thesis, School of Computer Science, Carleton University.

[61] Curtis, A.C.J., John, M.D.H., John, R.S. and Udo, W.P. (2000) A Methodology for Using Intelligent Agents to Provide Automated Intrusion Response. *Proc. IEEE SMC IAW'00*, United States Military Academy, West Point, NY, USA, June 6–7, pp. 110–116. IEEE Computer Society.

[62] Zhang, W., Teng, S. and Fu, X. (2008) Scan Attack Detection Based on Distributed Cooperative Model. *Proc. CSCWD'08*, Xi'an, China, April 16–18, pp. 743–748. IEEE Computer Society.

[63] Gelenbe, E. and Loukas, G. (2007) A self-aware approach to denial of service defence. *Comput. Netw.*, **51**, 1299–1314.

[64] Oke, G., Loukas, G. and Gelenbe, E. (2007) Detecting Denial of Service Attacks with Bayesian Classifiers and the Random Neural Network. *Proc. FUZZ-IEEE'07*, London, UK, July, pp. 1964–1969. IEEE, USA.

[65] Loukas, G. and Oke, G. (2007) Likelihood Ratios and Recurrent Random Neural Networks in Detection of Denial of Service Attacks. *Proc. SPECTS'07*, San Diego, CA, USA, July 16–18.

[66] Gelenbe, E. (1989) Random neural networks with negative and positive signals and product form solution. *Neural Comput.*, **1**, 502–510.

[67] Gelenbe, E. (1994) G-networks: A unifying model for neural and queueing networks. *Ann. Oper. Res.*, **48**, 433–461.

[68] Gelenbe, E. and Timotheou, S. (2008) Synchronized interactions in spiked neuronal networks. *Comput. J.*, **51**, 723–730.

[69] Pang, R., Allman, M., Paxson, V. and Lee, J. (2006) The devil and packet trace anonymization. *SIGCOMM Comput. Commun. Rev.*, **36**, 29–38.

[70] Pang, R., Allman, M., Bennett, M., Lee, J., Paxson, V. and Tierney, B. (2005) A First Look at Modern Enterprise Traffic. *Proc. ACM IMC'05*, Berkeley, CA, USA, October, 19–21, pp. 2–2. USENIX Association.

[71] Technologies, C. Winpcap. http://www.winpcap.org.

[72] symantec.com. Symantec security response. http://securityresponse.symantec.com/avcenter.

[73] Axelsson, S. (2000) The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.*, **3**, 186–205.

[74] McHugh, J. (2000) Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur.*, **3**, 262–294.

[75] Ashfaq, A.B., Robert, M.J., Mumtaz, A., Ali, M.Q., Sajjad, A. and Khayam, S.A. (2008) A Comparative Evaluation of Anomaly Detectors Under Portscan Attacks. *Proc. RAID'08*, Cambridge, MA, USA, September 15–17, pp. 351–371. Springer, Berlin.