ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Port Scan - A Security Concern

Tariq Ahamad Ahanger
College of Computer Engineering & Sciences, Salman Bin Abdulaziz University, KSA

Abstract: - Port scan is act of systematically scanning a computer's ports. Since a port is a place where information goes into and out of a computer, port scanning identifies open doors to a computer. Port scanning has legitimate uses in managing networks, but port scanning also can be malicious in nature if someone is looking for a weakened access point to break into your computer. Port scans represent a sizable portion of today's Internet traffic. However, there has been little research characterizing port scan activity. The goal of this project is to analyze sample network traces to discover and classify properties of port scans. We hope that this work will help to generate better network intrusion detection systems and increase general network security.

I. INTRODUCTION

Port Scan is the act of systematically scanning a computer's ports [1]. Since a port is a place where information goes into and out of a computer, port scanning identifies open doors to a computer. Port scanning has legitimate uses in managing networks, but port scanning also can be malicious in nature if someone is looking for a weakened access point to break into your computer [2].

Types of port scans:

- *Vanilla:* the scanner attempts to connect to all 65,535 ports.
- *Strobe:* a more focused scan looking only for known services to exploit
- *Fragmented packets:* the scanner sends packet fragments that get through simple packet filters in a firewall.
- *Udp*: the scanner looks for open udp ports.
- *Sweep*: the scanner connects to the same port on more than one machine.
- *Ftp bounce:* the scanner goes through an ftp server in order to disguise the source of the scan.
- *Stealth scan*: the scanner blocks the scanned computer from recording the port scan activities.

II. THE SIX PORT STATES RECOGNIZED BY NMAP

- **Open:** An application is actively accepting tcp connections, udp datagrams or sctp associations on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users. Open ports are also interesting for non-security scans because they show services available for use on the network[3].
- **Closed:** A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up. Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next [3].
- **Filtered:** Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common. This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically [3].
- **Unfiltered:** The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open[3].
- **Open Filtered:** Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response. The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way [3].
- **Closed Filtered:** This state is used when Nmap is unable to determine whether a port is closed or filtered. It is only used for the IP ID idle scan [3].

Port scanning in and of itself is not a crime. There is no way to stop someone from port scanning your computer while you are on the Internet because accessing an Internet server opens a port, which opens a door to your computer. There are, however, software products that can stop a port scanner from doing any damage to your system [4]. Port Scanning is the name for the technique used to identify open ports and services available on a network host. It is sometimes utilized by security

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

technicians to audit computers for vulnerabilities; however, it is also used by hackers to target victims [5]. It can be used to send requests to connect to the targeted computers, and then keep track of the ports which appear to be opened, or those that respond to the request. When a criminal targets a house for a burglary, typically the first thing he or she checks is if there is an open window or door through which access to the home can be gained. A Port scan is similar, only the windows and doors are the ports of the individual's personal computer. While a hacker may not decide to "break in" at that moment, he or she will have determined if easy access is available [6]. Many people feel this activity should be illegal, which it is not, however, due to the fact that the potential attacker is merely checking to see if a possible connection could be made, in most areas, it is not considered a crime. However, if repetitive port scans are made, a denial of service can be created. Hackers typically utilize port scanning because it is an easy way in which they can quickly discover services they can break into. In some cases, hackers can even open the ports themselves in order to access the targeted computer [7]. Hackers also use port scanners to conduct tests for open ports on Personal Computers that are connected to the web [8]. Each individual computer runs on multiple ports. For instance, when a person opens his or her email, the computer's server will open a port through which new mail will be downloaded through a connection to the email server [9]. Certain ports on an individual's personal computer are open continually, making them a target for any potential hacker who is searching for individuals to victimize. This can lead to one's sensitive and personal information falling into the hands of those who intend on using it for criminal activity. Unfortunately, criminals and computer hackers are always looking for new victims to exploit, and port scanning is one of the ways through which this can be accomplished [10]. Ports are like little doors on your system. Most packets leaving your machine come out of a certain door. They are destined for another door on another system. There are two different protocols that use ports: TCP and UDP. Each of these two protocols has 65,536 different ports. Various Internet services listen on certain well-known doors [11]. For example, Web servers usually listen on TCP port 80. Mail servers usually listen on TCP door port 25. An attacker launches a port scan to see what ports are open, with a listening service, on your machine. A port scan attack, therefore, occurs when an attacker sends packets to your machine, varying the destination port. The attacker can use this to find out what services you are running and to get a pretty good idea of the operating system you have [12]. Most Internet sites get a dozen or more port scans per day. As long as you harden your firewall and minimize the services allowed through it, these attacks shouldn't worry you. The Internet today is a complex entity comprised of diverse networks, users, and resources. Most of the users are oblivious to the design of the Internet and its components and only use the services provided by their operating system or applications. However, there is a small minority of advanced users who use their knowledge to explore potential system vulnerabilities. Hackers can compromise the vulnerable hosts and can either take over their resources or use them as tools for future attacks. With so many different protocols and countless implementations of each for different platforms, the launch of an effective attack often begins with a separate process of identifying potential victims [13]. One of the popular methods for finding susceptible hosts is port scanning. Port scanning can be defined as “hostile Internet searches for open ‘doors,’ or ports, through which intruders gain access to computers.” This technique consists of sending a message to a port and listening for an answer. The received response indicates the port status and can be helpful in determining a host’s operating system and other information relevant to launching a future attack [14]. The goal of this research article is to analyze and characterize port scanning traffic. By defining a set of heuristics and applying them to the network trace data, we were able to isolate suspicious packets and group them into sets of scans. These sets were further analyzed to extract properties of the port scanning traffic and to collect relevant statistics.

III. BACKGROUND AND RELATED WORK

Port scanning is a technique for discovering hosts’ weaknesses by sending port probes. Although sometimes used by system administrators for network exploration, port scanning generally refers to scans carried out by malicious users seeking out network vulnerabilities. The negative effects of port scans are numerous and range from wasting resources, to congesting the network, to enabling future, more serious, attacks [15]. There is a plethora of tools that aim to determine a system’s weaknesses and determine the best method for an attack. The best known and documented tool is nmap by Fyodor from www.insecure.org. Nmap uses a variety of active probing techniques and changes the packet probe options to determine a host’s operating system. Nmap offers its users the ability to randomize destination IPs and change the order of and timing between packets. This functionality can obscure the port scanning activity and thus fool intrusion detection systems. Other port scanners include queso, checkos, and SS. However, these tools do not provide all the capabilities of nmap and thus are not as popular [16]. Several port scan detection mechanisms have been developed and are commonly included as part of intrusion detection systems. However, many of the detectors are easy to evade since they use simple rules that classify a port scan as more than X distinct probes within Y seconds from a single source [17]. Typically, the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

length of Y is severely limited, to keep the amount of state manageable. Spice, a tool developed at Silicon Defense, tries to avoid this drawback. Spice maintains records of event likelihood, from which it generates anomalousness score for each packet. Packets with high scores are stored longer, while state for unsuspicious packets is safely discarded. This heuristic allows Spice to detect stealthy port scans while still being operationally practical. Another approach is employed by Vern Paxon in Bro and emphasizes real time performance and notification, as well as clear separation between mechanism and policy [18].

IV. CLASSIFICATION METHODOLOGY

For the purposes of our analysis, we define a port scan as all anomalous messages sent from a single source during the trace period. We classify port scans into three basic types based on the pattern of target destinations and ports the scan explores.

Vertical Scans

The vertical scan is a port scan that targets several destination ports on a single host. Naively executed, this scan is among the easiest to detect because only local (single-host) detection mechanisms are required.

Horizontal Scans

A horizontal scan is a port scan that targets the same port on several hosts. Most often the attacker is aware of a particular vulnerability and wishes to find susceptible machines. One would expect to see many horizontal scans for a particular port immediately following the publicizing of vulnerability on that port.

Block Scans

Some attackers combine vertical and horizontal scanning styles into large sweeps of the address-port space. This method can yield a hit-list for future exploitation as described in.

Scan Detection

One way to avoid detection is to increase the time between consecutive probes. This technique works since most intrusion detection systems look for X events in a Y-sized time window and can only keep a limited amount of state. We did not have real-time constraints and thus were able to use a time window large enough to detect such stealthy scans.

An attacker can also conceal her IP address by using IP decoys, or "zombie" computers under an attacker's control. Such a scan will appear in our analysis as different scans originating from several IPs. We attempt to quantify this error by combining scans that appear coordinated. If several sources IPs are seen targeting the same set of hosts and ports and these source IPs are in the same /24 network, they are classified as decoys. This is only an approximate solution, which we use for comparison.

Classification Rules

To separate port scanning traffic from other traffic, we looked for probes of two or more {IP address, port number} pairs from a given source within 120 seconds. By using this heuristic we detect the majority of all scans since most port scanning tools set the time between the packets to be much less than 120 seconds. We also keep the state of the destinations by maintaining information about 5000 targets that are not part of currently known scans and keep a 60-second window between the consecutive packets.

V. EXPERIMENTAL PLATFORM

The network trace data was obtained from CAIDA and was gathered on a very lightly utilized /8 network. Two traces were used, each spanning about a week: February 1st-8th, 2013, and February 11th-17th, 2013. More information about the data can be found in . The data was filtered to exclude all legitimate, outgoing and backscatter traffic. The remainder was mostly port scans with a small percentage of misconfigured traffic. The analysis was conducted on the UCSD Active Web machines. We used Snort to simplify scan detection and logging. Snort is a freeware traffic analyzer much like tcp dump, with the addition of preprocessors that allow for packet sorting based on a set of pre-defined rules. We then used a series of Perl scripts to further analyze the results and generate scan statistics.

VI. RESULTS

Using the previously defined rules, we observed 9927 vertical scans, 5623 horizontal scans, and 2013 block scans.

Packet Types and Distribution

Most of the packets were sent over TCP, with some UDP traffic. The distribution of packet types and protocols can be seen in Figure 1. As shown, most of the packets are TCP SYN packets, with ACK FIN packets a distant second, followed by UDP and TCP ACK RST. All the remaining types combined are only a minute fraction.

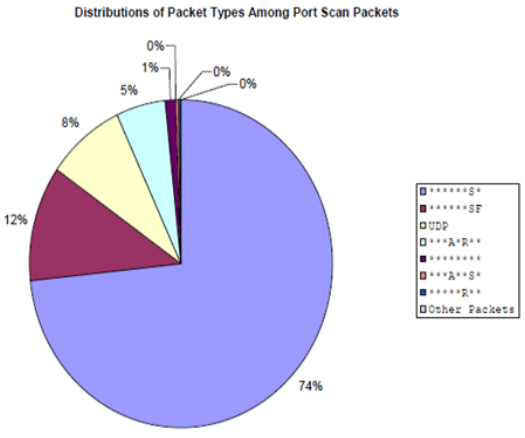

Fig 1 – Packet Types

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

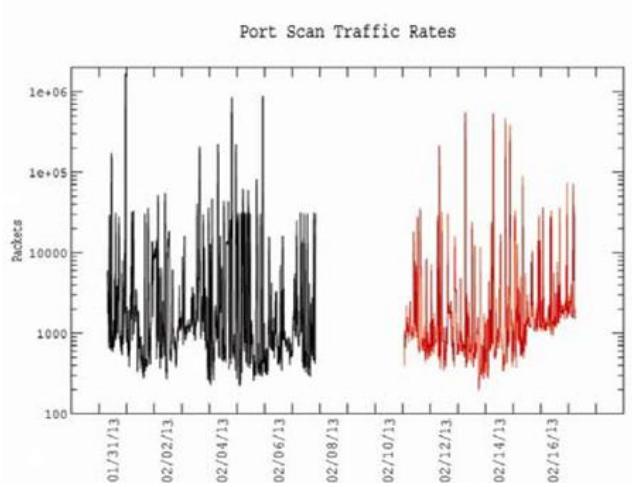

Fig 2 - Port Scan Traffic Rates

Another property we wanted to observe was the time distribution of port scan traffic. Figure 2 does not show a correlation between traffic and the time of the day. We can thus assume that port scanning is a constant activity. This might be due to time zone differences between the attack sources. An interesting metric is a scan size, which gives an indication of the amount of information gathered by the attacker. For vertical scans, we define a scan size as the number of distinct ports scanned. For horizontal scans, it is the number of distinct destination IPs. We observed many scans originating from the same /24 networks that exhibited the same behavior. We believed these scans were a coordinated effort. To roughly quantify the number of scans in this category, we grouped the vertical scans by the source IP if the destination IP and the scanned ports were the same and the source IPs were in the same /24 network. Thus, we assume that most of the source IPs was decoys since the scanning patterns of all of them are so similar. Figure 2 shows the distribution of the vertical scans, both before and after the grouping. We can see that there are a handful of large scans, with the size distribution being dominated by small scans.

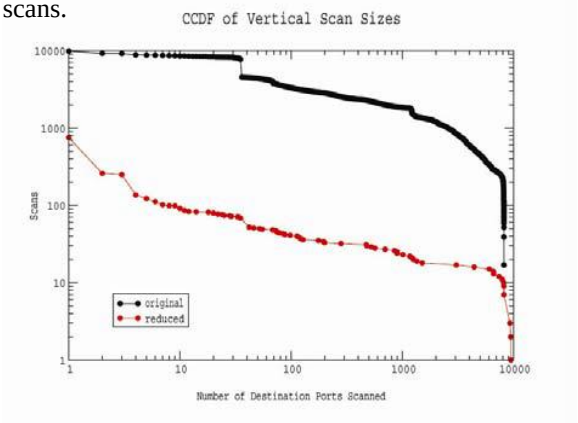

Fig 3 - Complementary Cumulative Distribution Function of the Vertical Scan Sizes

We perform the same grouping on the horizontal scans. The distribution of the horizontal scan sizes is also dominated by the small scans. However, in this case the change between the original and the reduced scan sets is hardly noticeable. It is unclear why vertical scans appear to make more widespread use of decoys; we leave this as an open question. The typical block scan we observed examined the same 2 or 3 ports across a large set of machines. In other words, we did not observe block scans that appeared to be comprehensively covering the address-port space.

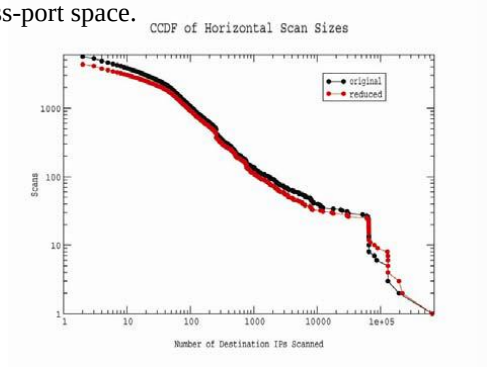

Fig 4 - Complementary Cumulative Distribution Function of Horizontal Scan Sizes

Target Ports

The popularity of target ports was another metric we evaluated. All 2^{16} port numbers were scanned at least once on some host, mostly due to the vertical scans that looked at all the ports on a host. Although some ports were scanned many more times than others, the amount of these scans as a percentage of overall probes is still negligible. Table 1 lists the target ports, the corresponding services and some related statistics.

VI. GEOGRAPHIC DISTRIBUTION OF SCAN SOURCES

The port scans are global phenomena. They originate from a multitude of locations across the world and seem to be correlated only with accessibility of the Internet. To map the location of scan sources, we created a list of unique source IPs and then used CAIDA's IPGeo tool to map IPs to corresponding latitude/longitude pairs. Figure 5 shows the resulting map.

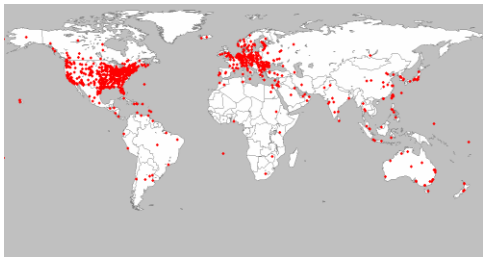

Fig 5 - Geographic Distribution of Port Scan Source IP Addresses

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Scan Patterns

Attackers might try to hide their port scanning activity from naïve detection mechanisms by randomizing the order of destination IP and port probes. From our analysis we saw that most scans did not employ this strategy. Of vertical scans, 58% probed port numbers sequentially, and 91% of horizontal scans traversed the destination IPs sequentially. Scan duration is another metric that helps us to evaluate port scans and design better intrusion detection systems. Figure 6 shows complementary cumulative distribution functions for vertical, horizontal, and block scans. We see a significant variance in the data although short scans tend to be more widespread than the long ones.

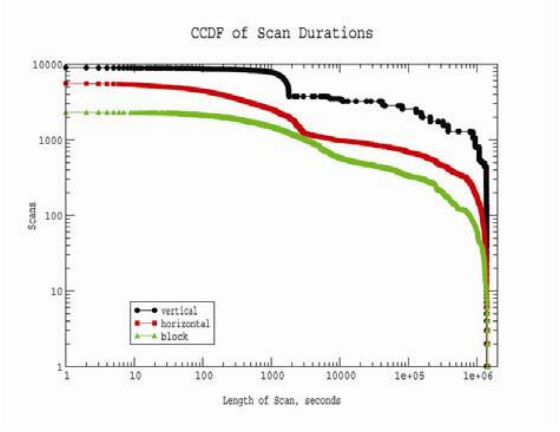

Fig 6 - Complementary Cumulative Distribution of Scan Durations

VII. CONCLUSIONS

The majority of the scans were carried over TCP, with TCP SYNs dominating the traffic. UDP was another protocol that we saw, although it was not very prevalent. Most of the scans were simple vertical or horizontal scans, with vertical scans prevailing by a factor of nearly 2. All the ports were scanned at least once, although even the most frequently scanned ports did not account for a large percentage of the probes. The scan sources originated from a multitude of locations and favored densely populated areas of Europe and North America. Most of the horizontal scans were sequential whereas the vertical scans varied. We observed great diversity in the scan duration.

REFERENCES

[1] Richard A. Becker, Stephen G. Eick, and Allan R. Wilks. Visualizing network data. IEEE Transactions on Visualization and Computer Graphics, 1(1):16–28, 1995.

[2] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan. Data mining for network intrusion detection. In Proc. NSF Workshop on Next Generation Data Mining, 2002.

[3] Robert F. Erbacher. Visual traffic monitoring and evaluation. In Proceedings of the Conference on Internet Performance and Control of Network Systems II, pages 153–160, 2001.

[4] L. Girardin and D. Brodbeck. A visual approach for monitoring logs. In Proceedings of the 12th Usenix System Administration conference, pages 299–308, 1998.

[5] Tom Goldring. Scatter (and other) plots for visualizing user profiling data and network traffic. In VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, pages 119–123, New York, NY, USA, 2004. ACM Press.

[6] Jaeyeon Jung, Vern Paxson, Arthur W. Berger, , and Hari Balakrishnan. Fast ports scan detection using sequential hypothesis testing. In Proc. IEEE Symposium on Security and Privacy, 2004.

[7] Teuvo Kohonen. Self-Organization and Associative Memory. Springer-Verlag, Berlin, 3rd edition, 1989.

[8] 8. Kiran Lakkaraju, Ratna Bearavolu, and William Yurcik. NVisionIP—traffic visualization tool for security analysis of large and complex networks. In International Multiconference on Measurement, Modeling, and Evaluation of Computer-

[9] Communications Systems (Performance TOOLS), 2003.

[10] 9. Stephen Lau. The spinning cube of potential doom. Communications of the ACM, 47(6):25–26, 2004.

[11] 10. David J. Marchette, V. Nair, M. Jordan, S. L. Lauritzen, and J. Lawless. Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[12] 11. J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: A tool for port-based detection of security events. In ACM VizSEC 2004 Workshop, pages 73–81, 2004.

[13] 12. K. Mundiandy. Case study: Visualizing time related events for intrusion detection. In Proceedings of the IEEE Symposium on Information Visualization 2001, pages 22–23, 2001.

[14] 13. Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In Proceedings of the Internet Measurement Conference, 2004.

[15] 14. Bryan Parno and Tony Bartoletti. Internet ballistics: Retrieving forensic data from network scans. Poster Presentation, the 13th USENIX Security Symposium, August 2004.

[16] 15. Leonid Portnoy, Eleazar Eskin, and Salvatore J. Stolfo. Intrusion detection with unlabeled data using clustering. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), 2001.

[17] 16. S. Staniford, V. Paxson, , and N. Weaver. How to own the internet in your spare time. In Proceedings of the 2002 Usenix Security Symposium, 2002.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

[18] 17. Soon Tee Teoh, Kwan-Liu Ma, S. Felix Wu, and Xiaoliang Zhao. Case study: Interactive visualization for internet security. In Proc. IEEE Visualization, 2002.

[19] 18. William Yurcik, James Barlow, Kiran Lakkaraju, and Mike Haberman. Two visual computer network security monitoring tools incorporating operator interface requirements. In ACM CHI Workshop on Human-Computer Interaction and Security Systems (HCISEC), 2003.

No.of Hits	*% of Total*	*Port Number*	*Port Services*
6588	0.081%	137	NetBIOS name service (UDP)
5127	0.063%	21	FTP
5103	0.063%	25	SMTP
4960	0.061%	53	DNS
4943	0.061%	17	QOTD
4940	0.061%	113	IDENTD/AUTH
4935	0.061%	105	CSO
4934	0.061%	33	DSP
4932	0.061%	129	PWDGEN – not used for anything, so most likely a port scan
4932	0.061%	29	MSG-ICP
4931	0.061%	1	TCPMUX – test if machine is running SGI Irix
4928	0.060%	13	daytime - Not clearly specified format => used for fingerprinting machines
4928	0.060%	93	DCP
4925	0.060%	41	RAT: Deep Throat - Puts an FTP Service at Port 41
4925	0.060%	85	MIT ML Device
4924	0.060%	97	Swift Remote Virtual File Protocol
4922	0.060%	77	Private Remote Job Execution Services
4920	0.060%	73	Remote Job Services
4919	0.060%	121	Jammerkilla - Encore Expedited Remote Procedure Call
4918	0.060%	37	Time

Table 1 - Top 20: Most Actively Scanned Ports and their Functions