# Cyber Scanning: A Comprehensive Survey

Elias Bou-Harb, Mourad Debbabi, and Chadi Assi

***Abstract*—Cyber scanning refers to the task of probing enterprise networks or Internet wide services, searching for vulnerabilities or ways to infiltrate IT assets. This misdemeanor is often the primarily methodology that is adopted by attackers prior to launching a targeted cyber attack. Hence, it is of paramount importance to research and adopt methods for the detection and attribution of cyber scanning. Nevertheless, with the surge of complex offered services from one side and the proliferation of hackers' refined, advanced, and sophisticated techniques from the other side, the task of containing cyber scanning poses serious issues and challenges. Furthermore recently, there has been a flourishing of a cyber phenomenon dubbed as cyber scanning campaigns — scanning techniques that are highly distributed, possess composite stealth capabilities and high coordination — rendering almost all current detection techniques unfeasible. This paper presents a comprehensive survey of the entire cyber scanning topic. It categorizes cyber scanning by elaborating on its nature, strategies and approaches. It also provides the reader with a classification and an exhaustive review of its techniques. Moreover, it offers a taxonomy of the current literature by focusing on distributed cyber scanning detection methods. To tackle cyber scanning campaigns, this paper uniquely reports on the analysis of two recent cyber scanning incidents. Finally, several concluding remarks are discussed.**

***Index Terms*—Probing, Cyber scanning, Network reconnaissance, Scanning events, Probing campaigns.**

## I. Introduction

CYBERSPACE is the electronic world created by interconnected networks of information technology and the information on those networks. It can be defined as the interdependent network of information technology infrastructures, including the Internet, telecommunication networks, computer systems, and embedded industrial processors and controllers. Cyberspace is a global commons where more than 1.7 billion people are linked together to exchange ideas and services [1]. Moreover, it underpins almost every facet of a modern society and provides critical support for the economy, civil infrastructure, public safety, and national security. Cyberspace is controlled and operated using information and communication technologies. The latter could be considered as the nervous system of our today's world, as critical infrastructures such as telecommunication, transportation, financial services, agriculture, electric grids and public health services profoundly depend on it for their successful operations.

It is evident that individuals, industry and governments are embracing the many advantages that cyberspace offers. According to two recent reports [1, 2], 87% of North American corporations used the cyberspace to conduct business, where the online sales revenue due to that were estimated at $62.7 billion. Moreover in 2009, 74% of households had paid Internet service, 59% of personal tax filings were completed electronically and 67% of north Americans had banked online. Furthermore, governments have also become increasingly dependent on the Internet. The Canadian Federal government alone offers more than 130 commonly used services online, including tax returns, employment insurance forms and student loan applications [3]. Thus, nowadays, the success of cyberspace is an essential asset which demands protection against malicious misuse and other destructive attacks. This task is indispensable yet very challenging.

Recent events demonstrated that cyberspace could be subjected, at high speeds and in full anonymity, to severe attacks with drastic consequences. One particular research revealed that 90% of United States companies have been the target of a cyber attack, with 80% suffering a significant financial loss [4]. In addition, the cyber security report [1] elaborated that in a recent one year period, 86% of large north American organizations had suffered a cyber attack where the loss of intellectual property as a result of these attacks doubled between 2009 and 2011. Moreover, the report alarmed that more than 60% of all the malicious code ever detected, originating from more than 190 countries, was introduced into cyberspace solely in 2011. In general, cyberspace could facilitate the following cyber attacks:

- Distributed Denial of Service (DDoS) [5]: It is an attempt to make a computer or network resources unavailable. It consists of attacks that are deployed to temporarily or indefinitely shutdown services. The timing of such attacks can be coordinated to exploit the availability of critical organization infrastructure by directing enormous flood of Internet traffic towards a small set of targeted Internet Protocol (IP) addresses. By flooding the available bandwidth with intensive traffic, DDoS can effectively bring down a service with potential loss of financial revenue.
- Advanced Persistent Threats (APTs) [6]: APTs typically refer to a group, such as a foreign government, with both the capability and the intent to persistently and effectively target an entity. These cyber attacks possess high stealth techniques and are often target-specific. They are advanced since their operators have a full spectrum of intelligence-gathering techniques at their disposal. APTs assign priorities to specific tasks rather than opportunistically seek information for financial or other gain. The attack is conducted through continuous monitoring and interaction in order to achieve the defined objectives. APT

Manuscript received January 16, 2013; revised May 1, 2013, August 13, 2013, and October 20, 2013.

The authors are with the Concordia Institute for Information Systems Engineering (e-mail: (e_bouh, debbabi, assi)@ciise.concordia.ca).

Digital Object Identifier 10.1109/SURV.2013.102913.00020


Fig. 1. The Anatomy of a Cyber Attack

attacks are executed by coordinated human actions rather than by just relying on automated pieces of code. Their operators are typically very skilled, motivated, organized and well funded.

- Zero-day Attacks [7]: These attacks exploit the observation of newly discovered yet un-patched vulnerabilities to achieve their malicious tasks. While a number of detection mechanisms have been proposed to protect against these attacks, including, access control lists on the edge network, port-knocking and application white-listing [8], these cyber attacks are still very dominant and pose serious issues and challenges.
- Cyber Terrorism: Cyber terrorism refers to the use of cyber attacks in terrorist activities. With the increase of cyber warfare incidents such as Stuxnet [9] and the Russian-Georgian War [10], cyber attacks can shift from targeting corporations to targeting governments and military facilities. With the ever escalating political tension between various world parties, these cyber attacks are becoming a fifth dimension of warfare [11] and a leading advantage to their operators.

Despite efforts to protect the cyberspace, the latest report from senior government officials highlighted that only limited progress has been made in improving the cyber security of crucial networks [12].

Cyber scanning or network reconnaissance is a growing cyber security concern due to the fact that it is the primary stage of an intrusion attempt that enables an attacker to remotely locate, target, and subsequently exploit vulnerable systems. It is basically a core technique and the main facilitator of the above mentioned cyber attacks. Figure 1 depicts the anatomy of a cyber attack where it is seen that cyber scanning plays a major role.

Networks and Internet wide infrastructure are under constant attack from a variety of unproductive or malicious network activity that includes probes from misconfigured and exploited systems, backscattered traffic [13], and automated tools and botnets [14, 15]. Yegneswaran et al. [16] estimated that there exist 25 billion daily global intrusion attempts and this activity continues to increase. Panjwani et al. [17] concluded that a significant 50% of attacks against cyber systems are preceded by some form of network scanning activity. Large numbers of worm-infected systems randomly scan the Internet searching for susceptible systems to exploit. Over the past few years, malicious outbreaks of very large size and severity have rapidly spread across vulnerable systems. In fact, there have been worm outbreaks that have been able to scan and infect 90% of all the vulnerable cyber hosts in less than 10 minutes [18]. Furthermore recently, there have been a flourishing of a cyber phenomenon dubbed as *cyber scanning campaigns*. These are scanning techniques that are extremely distributed, possess composite stealth capabilities and high coordination. Rather than focusing on specific hosts or networks, these campaigns aim at probing and consequently exploiting the Internet's services and infrastructures. Hence, the capability to detect and attribute various scanning activity is a very important task to achieve as this will prevent the actual cyber attack from occurring.

### A. Related Surveys

Although cyber scanning has been studied before, especially its detection techniques, the literature still lacks a survey that provides the readers, coming from different backgrounds, with a comprehensive coverage of the topic. For instance, Barnett et al. [19] solely focused on scanning techniques by providing a taxonomy. Their taxonomy analyzed three main scanning techniques, namely, TCP, UDP and ICMP scans. They presented the techniques using patterns and utilized the scanning speed as an additional attribute. In another survey, Bhuyan et al. [20] elaborated on port scans and their detection methodologies. This survey highlighted on single-source and distributed detection techniques. Moreover, it provided some information on available detection data sets and evaluation metrics. The above two surveys are the closest to the work that we present in this paper. It is noteworthy to mention that the work we present in this paper solely focuses on network scanning or probing. The latter is often the primarily methodology that is adopted by attackers *prior* to launching a targeted cyber attack. Thus, our intention from this work is not to survey cyber attacks or their detection/prevention approaches but rather to highlight on cyber scanning as a precursor technique to launching various cyber attacks. The literature has already provided various taxonomies and discussions of cyber threats [21], including, denial of service attacks [22], botnets [23], malware [24], phishing [25] and spamming [26].

In this paper, we contribute in the following points:

1) By primarily providing a categorization of the entire cyber scanning topic, by discussing cyber scanning's nature, approaches and strategies. This offers the readers a strong, coherent and clear entry point into the topic.
2) By providing a classification for 19 cyber scanning techniques. We thoroughly further discuss this exhaustive and comprehensive list of techniques, and provide their advantages and disadvantages. We as well present a complete summary of those techniques.
3) By developing a unique literature taxonomy of distributed cyber scanning detection methodologies. This

also covers new material after 2010, the year of the latest related survey [20].

4) By highlighting on a new phenomenon of cyber scanning known as cyber scanning campaigns and presenting the analysis of two of its recent incidents targeting two diverse Internet wide infrastructures.

### B. Paper Organization

The rest of this survey is organized as follows. In Section II, we present a categorization of the entire cyber scanning topic. In Section III, we provide an exhaustive discussion on cyber scanning techniques. A literature review and a taxonomy on distributed scan detection techniques is depicted in Section IV. In Section V, we report on two recent cyber scanning campaign incidents. Finally, concluding remarks and lessons learned are stated in Section VI.

## II. Cyber Scanning: Nature, Strategies & Approaches

In this section, we provide a categorization of the entire cyber scanning topic as depicted in Figure 2. We further present a discussion that elaborates on the nature, strategies and approaches of cyber scanning.

### A. Nature of Cyber Scanning

Cyber scanning can be first classified based on its nature; whether the scanning or probing is performed actively or passively. In this section, we present those criteria and consequently discuss their advantages and disadvantages.

*1) Active Scanning:* Active scanning is the process of identifying network services by transmitting certain packets known as the probe packets towards network hosts and devices and subsequently monitoring their responses. Active scanning is typically employed by malicious adversaries to probe a network for certain vulnerabilities. It needs to be noted though that active scanning has a legitimate use as part of a robust network security policy. It allows a network operator to discover the open services in the network in an attempt to check those for known vulnerabilities. The probe packets could either be generic, targeting a specific protocol rather than a certain application, or they can be targeted, focusing on a precise host application. An instance of a generic probe packet could be the typical TCP handshaking procedure [27] for establishing a connection. The latter technique could be used to identify services operating on well-known ports. However, this technique is deficient in two cases. First, this method will only verify the readiness to open a TCP connection and not what service is supported by the connection. Thus, it tends to misinterpret services running on non standard ports. Second, it fails to classify services that have no standard ports, or those that use dynamic port assignment such as services utilizing the remote procedure call (RPC) protocol [28]. UDP probing is another employed approach for active scanning. Certain protocols, especially those running on well-known UDP ports, will successfully respond to a UDP probe packet. Moreover, one can indirectly infer the presence of a UDP service by the lack of a negative response; many hosts automatically generate ICMP port unreachable messages [17] when no process is listening to a given UDP port. Although a lack of response is not definitive, it might indicate the presence of a UDP service.

One example of active scanning is operating systems fingerprinting [29, 30]. This procedure is rendered by the real-time attempt to remotely determine the operating system (type and version) of a particular host of interest. The idea is to send packets to a host so that any responses (or lack of responses) could be analyzed. The responses to these sequences of packets form a signature or a fingerprint for the remote operating system that can be compared against a signature database of known operating system versions. Operating system fingerprinting takes advantage of the observation that each operating system's network stack [27] (i.e., software that implements the TCP/IP protocol) has slight variations in the way it responds to certain packets. These variations offer the ability to determine the type of the remote host operating system. Another example of active scanning is application fingerprinting [31]. It is the real-time action of trying to remotely determine the applications or services running on a particular host of interest. Servers routinely send information about the applications they are running to client systems during normal connection activities. The initial text sent by servers during a connection attempt is known as a banner. The act of harvesting banners during an active identification of network systems and their applications [32] is an interesting and a beneficial concept. For instance, banner grabbing would be routinely performed during vulnerability testing (i.e., penetration testing) of the network. The software versions advertised in application banners can identify potential security issues if it is determined that the software version contains known vulnerabilities.

*2) Passive Scanning:* Passive scanning [33] identifies network services by observing traffic generated by servers and clients as it passes an observation point. Specialized hardware or software could be inserted and installed at the monitoring point to successfully establish passive monitoring. Many routers can 'mirror' ports, sending copies of packets out of another interface to a monitoring host. Furthermore, hardware taps such as optical splitters place no additional burden on the router, but require a brief service interruption to install. Alternatively, Wireshark [34] is one of the most prominent passive software tools.

The detection of well-known services (both TCP and UDP) with passive monitoring is fairly straightforward. An exchange of traffic with a given host indicates an operational service. For TCP, the monitoring host only needs to capture TCP connection setup messages (i.e., the SYN packets [27]); the completion of the three way handshake clearly indicates that a service is available. Under normal operations, the presence of a positive response to a connection request (SYN/ACK) is a sufficient evidence of a TCP service. UDP services can also be identified by observing traffic; however, since UDP is a connectionless protocol, the concept of 'server' and 'client' is not sufficiently clear without application protocol information. In addition, while bi-directional traffic positively indicates a

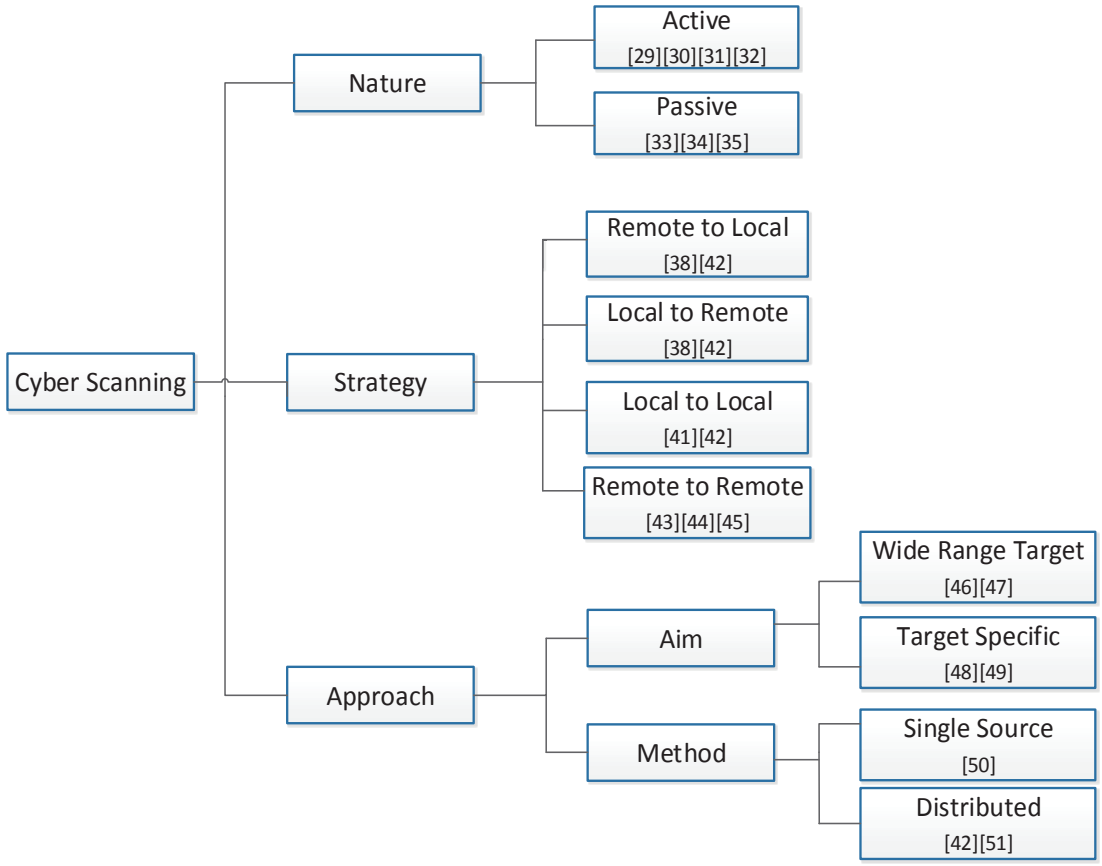



Fig. 2. A Categorization of the Cyber Scanning Topic

UDP service, unidirectional traffic may also indicate a service (since UDP does not mandate a response), but may as well indicate unsolicited probe traffic. As with active probing, passive scanning can not identify services that do not run on well-known ports.

One example of passive monitoring is Passive Asset Detection System (PADS) [35]. The system is a signature-based software used to passively detect network assets using application fingerprinting. It attempts to provide an accurate and current listing of the hosts and services offered on the network. It utilizes the TCP, ARP, and ICMP protocols [36] to perform its signature matching.

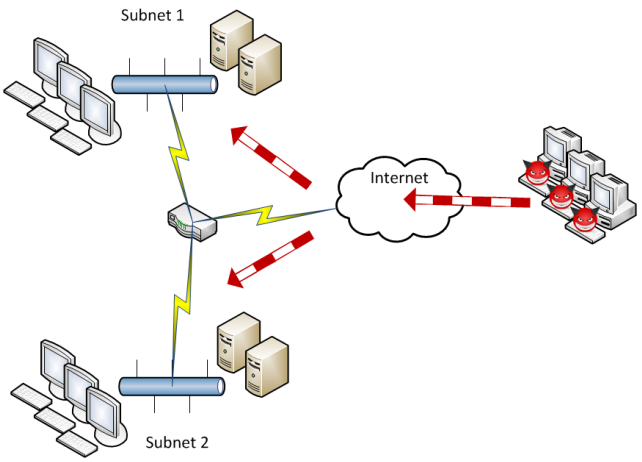
*3) Discussion:* Based on the aforementioned active and passive scanning descriptions, we subsequently discuss their advantages and disadvantages.

In general, active scanning provides a comprehensive report of all open and unprotected ports at the time of the probing. However, it will not detect ports that are filtered by firewalls or obscured by mechanisms such as port knocking [37]. Active scanning typically performs very fast in achieving its task. The main disadvantage of active probing is that it is very intrusive. Active probes solicit a response that would not have been sent otherwise. This can be detected and logged by the host or by intrusion detection systems, particularly if one systematically scans all hosts in a region. A second disadvantage of active scanning is that it does not identify hosts that may be temporarily unavailable at the time of the scan. This disadvantage can be mitigated using multiple active scans, although additional scans may draw further attention and hence increase the probability of being detected.

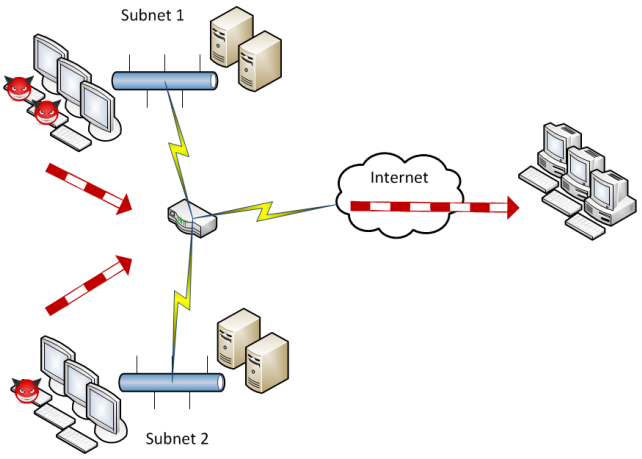
First, passive monitoring has a advantage of being non-intrusive. In fact, it generally cannot be detected by either communication parties. A second advantage of passive monitoring is that it can better detect active services running on transient hosts. Thus, it is specifically effective against machines that are frequently powered off, such as laptops, or hosts temporarily disconnected from the network. Third, passive monitoring can detect services that active probing misses because of firewall configurations. Fourth, passive monitoring can also provide insights into trends and other behaviors which active probing cannot. While monitoring servers, passive monitoring can also track clients, providing extra information such as server popularity and server load. Finally, since passive monitoring consumes no network resources (other than the monitoring host), it can be run on a long-term basis as part of normal network operations. The main disadvantage of passive monitoring is that it only detects services that are active. Therefore, silent servers go unmonitored, even though they may still pose vulnerabilities. Nevertheless, this disadvantage can be mitigated by long term monitoring.

### *B. Cyber Scanning Strategies*

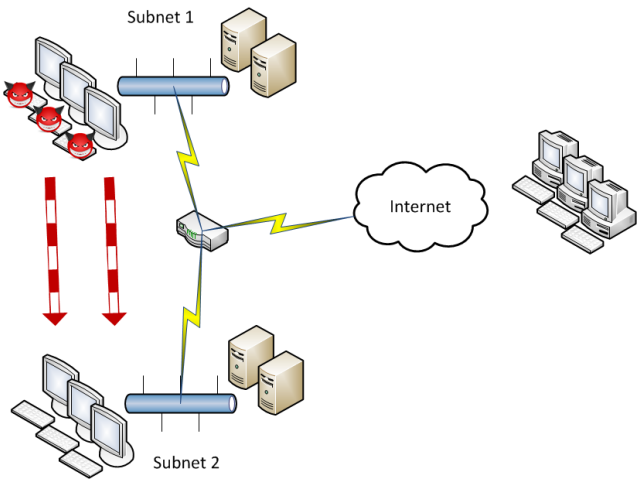
Cyber scanning activities can also be defined by which strategy they adopt. We can classify those strategies into four




(a) Remote to Local




(b) Local to Remote




(c) Local to Local

Fig. 3. Cyber Scanning Strategies

classes; remote to local scanning, local to remote scanning, local to local scanning and remote to remote scanning. The first three classes take into consideration the boundaries of a specific enterprise network and define the direction of the cyber scanning activity. Such activity can be generated by a diverse number of hosts, targeting a number of hosts, and using various cyber scanning methods and techniques.

The remote to local scanning [38] refers to a remote host, outside the boundary of a specific network, performing some sort of cyber scanning on a host inside the enterprise network. This strategy is the most worrisome for enterprise network administrators as they attempt to protect their IT infrastructure from unknown external adversaries. Local to remote cyber scanning occurs when a host, within the administrative control of the enterprise network, scans systems outside the network boundary. In this context, the scanning host performs network reconnaissance against external systems. This strategy may cause serious legal issues against the enterprise network since its infrastructure would be used for malicious purposes against Internet systems [39]. Moreover, local to local cyber scanning [40] refers to a host that scans systems within the boundaries of the enterprise network in which it resides. Topological scanning worms [41, 42] frequently employ this type of scanning strategy. Local to local scanning activity can occur within or between network subnets. Figure 3 summarizes the aforementioned discussed three strategies.

On the other hand, remote to remote scanning [43–45] does not depend on certain boundaries. It can be defined as world wide cyber scanning campaigns. Rather than focusing on a specific enterprise network as a target, this strategy aims at probing and sequentially exploiting the Internet's services. This strategy is often distributed, possesses sophisticated stealth capabilities and is typically highly coordinated. Section V of this paper will provide the analysis of two recent cyber scanning campaign incidents.

### C. Cyber Scanning Approaches

The third classification of cyber scanning, as shown in Figure 2, is based on its approaches, which are composed of aims and methods. The aims specify what is being targeted while the methods state how the cyber scanning is performed.

*1) Wide Range Cyber Scanning:* Wide-range reconnaissance can be defined as the rapid scanning of large blocks of Internet addresses in the search for a specific service or vulnerability. Typically, there is little human interaction in this type of reconnaissance. This is a characteristic of auto-rooters (i.e., a suite of programs designed to automatically scan and attack target machines) [46] and worm propagation. Auto-rooters are composite tools that augment basic port scanning functionality by launching an attack as soon as an open port is located on a target system; they are often used for the rapid enrollment of vulnerable systems into botnets [47] of compromised systems. Simple scanning worms propagate by indiscriminately probing the Internet as rapidly as possible to locate and infect vulnerable systems.

*2) Target-Specific Cyber Scanning:* In contrast, numerous sophisticated scanning techniques allow stealthy, focused scanning of a predetermined target host or network. The following techniques belong to this category:

- Indirect scanning occurs when an attacker uses some systems to scan a target and other systems to attack the same

victim. If the scanning activity from the scanning system is detected, the attacker simply uses another scanning system. A slightly more sophisticated variation uses throw-away scanning systems; previously compromised systems are disposed by the attacker after executing the malicious tasks. In this case, any traced back scanning activity will be attributed to the owner of the compromised system and not to the real attacker.

- Botnet scanning [48] occurs when a collection of compromised systems (bots or zombies) are used to scan a target. The bots are not necessarily on a contiguous set of IP addresses but rather could be very dispersed. For instance, consider a botnet that has an exploit capability against a network service. A botnet of just 254 systems would be able to scan an entire Class C network for that service by sending a single packet from each bot (each with a unique IP address). In this example, perhaps correlating the scanning campaign would be possible, however, it would not reveal the true adversary (the operator of the command and control center) since the bots are basically zombie members.
- Low and slow scanning [49] occurs when an attacker slowly scans a target host or network (i.e., a single scanning campaign may take days, weeks or months). Slow scans may blend into the network noise never exceeding detection thresholds or exhausting detection system state.

*3) Single Source Cyber Scanning:* A single source cyber scanning activity operates from a one (source) to many (targets) fashion. Single source cyber scanning can be classified as belonging to one of four types; vertical, horizontal, strobe and block scans [50]. 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. Note that in general, a vertical scan can be defined as consisting of six or more ports on a single computer, while a horizontal scan as consisting of five or more IP addresses within a single subnet.

*4) Distributed Cyber Scanning:* Distributed scanning [51] occurs when multiple systems act in a union strategy to scan a network or host of interest. Typically, one system will act as a central node and collect the scanning results from all participating systems. Distributing the scanning activity reduces the scanning footprint of any single system and thus decreases the likelihood of being detected.

### D. Summary

In this section, we provided a categorization of cyber scanning. Additionally, we presented a discussion that elaborated on the nature, strategies and approaches of cyber scanning. From the above, we can extract the following:

- Active scanning is efficient but is very intrusive.
- Passive scanning is less intrusive, works well in the presence of firewalls and is optimized to operate effectively with transient hosts.
- Cyber scanning strategies include remote to remote scanning also known as cyber scanning campaigns. The latter possess sophisticated stealth capabilities and are typically highly coordinated.
- Botnet scanning is both a target-specific and a distributed cyber scanning method.

## III. Cyber Scanning Techniques

In this section, we introduce a classification of cyber scanning techniques as shown in Figure 4. We elaborate on this by presenting the techniques and their details in terms of exchanged messages [52] and their scanning abilities. Moreover, we pinpoint and discuss, when applicable, their advantages, their disadvantages and the scenarios when the techniques are best used. Finally, we present a summary of the cyber scanning techniques that includes, but is not limited to, the transport protocol the technique aims to identify, their exchanged messages, and whether the technique is immune to firewall detection. Please note that these techniques could be considered as active scanning, as previously depicted in Figure 2, and could be employed in various strategies and approaches.

### A. Open Scan

Open scan, also known as the vanilla scan [53], is the simplest scanning technique. It refers to the method that follows the same TCP handshake connection that every other TCP-based application uses. Hence, this scanning technique is considered 'Open' since it reacts as a normal TCP connection to determine if a port is available. It utilizes the connect() [54] call functionality that is used by the operating system to initiate a TCP connection to a remote device. The Open scan is illustrated in Figure 5.

This technique utilizes the TCP protocol and the SYN flag to detect TCP ports. When a closed port is targeted, the victim replies with a RST flag. On the other hand, when an open port is detected, the victim replies with an ACK flag. It is worthy to note that this simple technique is easily detected by a firewall. An advantage of this technique is that it can achieve its scan in a very simplistic way without requiring any other functionalities or privileges. The latter is true because this technique utilizes the systems' normal TCP-based methods when connecting to the target. A disadvantage of this scan is apparent when connection logs are examined. Since the Open scan technique requires the completion of a TCP connection, normal application processes immediately follow. Although these applications are directly met with a RST packet, the application has already provided the appropriate login screen or service page. By the time the RST is received, the application initiation process is already well underway and additional system resources have been used. Because this scan technique is evident and easily identified when browsing through applications event logs, it might be considered the TCP scan of last resort. If privileged access is not available and the determination of open TCP ports is absolutely necessary,

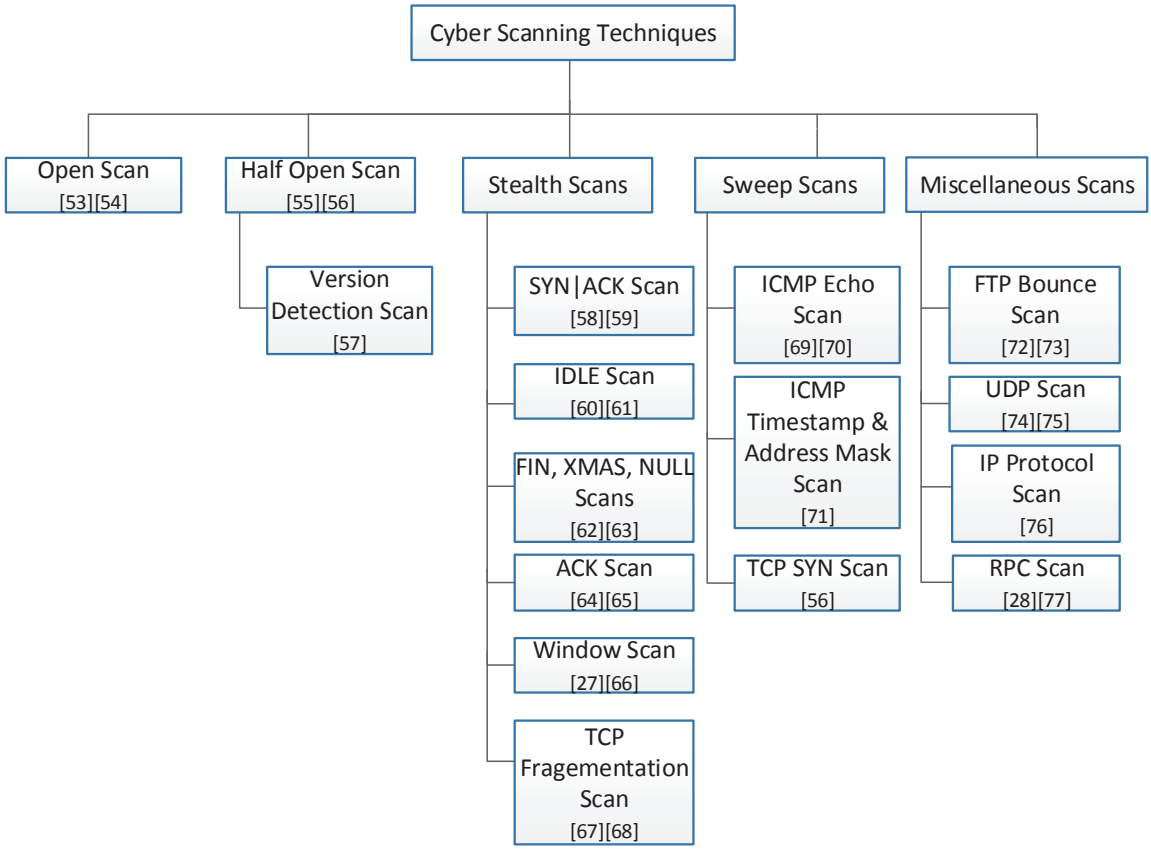



Fig. 4. A Classification of Cyber Scanning Techniques

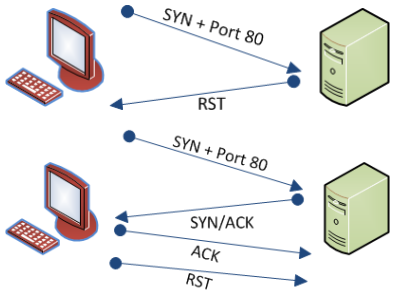



Fig. 5. The Open Scan targeting a closed (5(a)) and an open port (5(b))

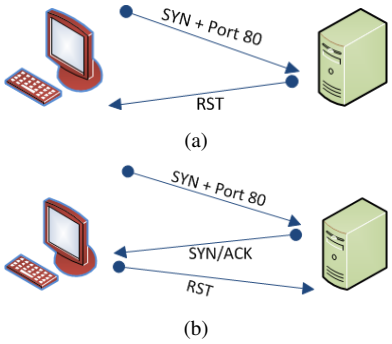



Fig. 6. The Half-Open Scan targeting a closed (6(a)) and an open port (6(b))

this scanning technique may be the only available method to be used.

### B. Half-Open Scan

The Half-Open scan, commonly dubbed as the TCP SYN scan [55, 56], is a common method for port identification that allows the scanner to gather information about open ports without completing the TCP handshake process. When an open port is identified, the TCP handshake is reset before it can be completed. Similar to an Open Scan, a Half-Open scan targeting a closed port will receive a RST packet. However, if the source receives an acknowledgment to a SYN request, meaning a port is open, then the source directly sends a RST frame to reset the session, and therefore the handshake is never completed. This technique is shown in Figure 6.

Since this scan technique never actually creates a TCP session, it is advantageous in two ways. First, it is not logged by destination applications. This point makes the Half-Open method somehow more stealthier than the Open-Scan method, as it is less visible in the destination systems' application logs. Second, it is less stressful to the application service because it does not force the application to initialize or for systems resources to be allocated. On the other hand, this method suffers from one disadvantage. Since there is a need to create new raw packets that do not completely abide by the TCP handshake, the half-open connection process requires some elevated systems privilege (i.e., the modification of network-level packets) at the source to be successful, which is not always feasible. It is significant to mention that this method

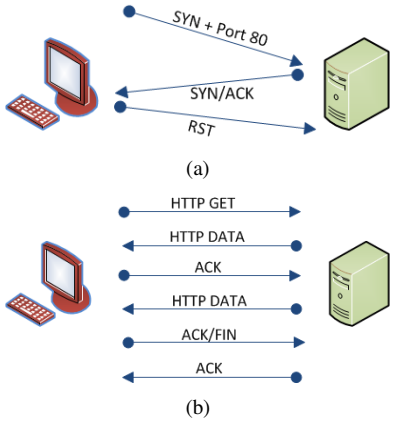



Fig. 7. The Half-Open scan (7(a)) executing prior to the Version Detection Scan (7(b))

can operate on any operating platform and the fact that it only half-opens the TCP connections makes it a very efficient and a streamline method. Nevertheless, since the Half-Open scan technique is structured and uses known TCP flags, it can be detected by an edge firewall rather easily.

*1) Version Detection Scan:* Although the Version Detection scan does not aim to detect open ports as the previous methods, it exploits them by probing the software applications running on remote devices [57]. This method would typically utilize the Half-Open scan technique prior to executing its own scanning method. If open ports are found, the Version Detection scan will begin the probing process by directly communicating with the remote applications on the open ports to uncover as much information as possible. Such information may include the type, the version and the status of that service, the underlying operating system and its version and other services that depend on that running service. This information can be of benefit to a network manager for proper and effective patching purposes or it can be analyzed by the scanner or attacker to exploit a certain known vulnerability of a specific running service. The Version Detection Scan is depicted in Figure 7. In this illustration, the Version Detection Scan technique first executed the TCP SYN scan. After detecting that port 80 is open, it ran its own scan to probe the service on that port. This method poses two advantages from a network management perspective. It can assist in network host applications' version management where hosts showing older software revisions are identified and further action is taken. Second, it assists in locating software that is not compliant with organizational standards. This is also an effective method of verifying the licenses of application services. Nonetheless, this technique possesses two disadvantages. First, it requires significant processing power (less significant using today's machines but still a point to consider) and elevated networking bandwidth since it needs to probe all the services and consequently transmit all their information. Second, since this technique will open numerous sessions with the remote applications, its activity is usually written in application logs which makes it a less stealthier technique.

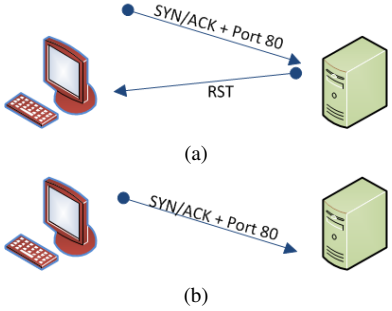



Fig. 8. The SYN|ACK Scan targeting a closed (8(a)) and an open port (8(b))

### C. Stealth Scans

The aforementioned cyber scanning techniques only use the typical SYN flag to investigate open ports. Hence, they are easily detected and logged by intrusion detection systems. In this section, we present and discuss stealth scans. These techniques try to avoid filtering devices by employing certain sets of flags other than SYN to appear as legitimate traffic. All these techniques resort to inverse mapping to determine open ports.

*1) SYN|ACK Scan:* The SYN|ACK scan [58, 59] is a slight modification of the Half-Open scan. Instead of just sending a SYN flag, the source sends a SYN in addition to an ACK flag to the target. For a closed port, the target will reply with a RST flag while a request to an open port will not generate a response. The latter is due to the fact that the TCP protocol requires a sole SYN flag to initiate a connection. This scanning technique is illustrated in Figure 8. This scan technique may generate a notable amount of false positives. For instance, packets dropped due to filtering devices, network traffic, timeouts, etc. can provide an incorrect inference of an open port while the port is in fact closed. However, this is a relatively fast scan method that avoids the three-way handshake and does not utilize a sole SYN flag.

*2) IDLE Scan:* A more complex stealth technique that utilizes the previous SYN|ACK scan and the Half-Open scan is known as the IDLE scan [60, 61]. The technique aims at gathering port information using another station on the network (the zombie) where the scanning process appears as it has been initiated by the zombie IP address instead of the actual source station. This scanning method exploits IP fragmentation identification sequences and implements IP address spoofing. For the scanning process to be executed, the identified zombie machine should satisfy the following two requirements:

- The zombie host must be idle (hence the name 'IDLE' scan). This requirement ensures that the IP identification frames will remain consistent throughout the duration of the scan.
- The zombie host must provide consistent and predictable IP identification (IPID) values. Most operating systems satisfy this requirement.

This technique is clarified in Figure 9. First, in Figure 9(a), the source sends a SYN/ACK flag to the zombie host expecting

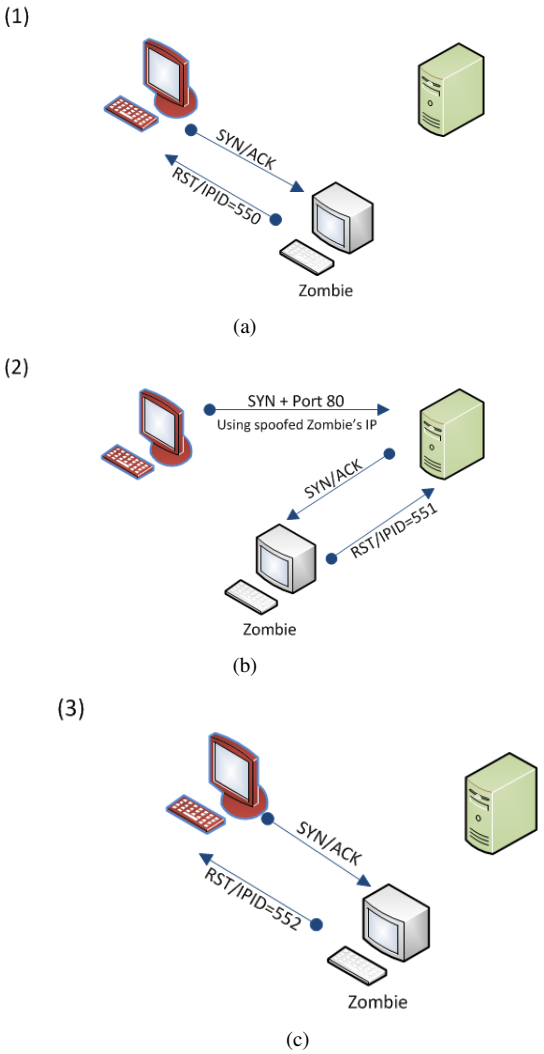



Fig. 9. IDLE scan executing process

a RST flag as a response. This RST packet contains the initial IPID. In Figure 9(b), the source executes a Half-Open scan, using the spoofed IP address of the zombie, targeting the destination host. If that port is open, the destination will reply to the zombie with a SYN/ACK. The zombie, not expecting a SYN/ACK since he/she never sent a SYN, will reply by a RST packet. The latter response will increment the zombie's IPID. Finally, in Figure 9(c), the original host resends the initial SYN/ACK probe to the zombie station. If the IPID has been incremented, then the source will infer that the port that was spoofed in the original SYN frame is open on the destination target. If the IPID has not been incremented, then the source will conclude that the port is closed. The IDLE scan technique of spoofing IP addresses and checking IPIDs allows the source to find open ports from a distance, even if packet filters are in place. The source simply requires any open port to a zombie host to complete the communication process. One of the core advantages of this technique is its stealth factor. A destination station will never identify the IP address of the scanning host. On the other hand, the disadvantages of this technique are three fold. First, there should be a satisfaction of the zombie workstation

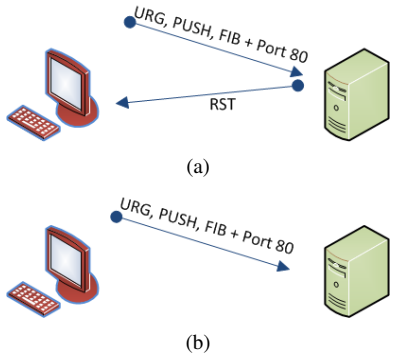



Fig. 10. The Xmas Scan targeting a closed (10(a)) and an open port (10(b))

requirements prior to commencing the scanning process, especially the idle state requirement. Second, although the technique implements source IP address spoofing, the source will still be identified if the technique is used on a local subnet. This last fact is legitimate since the source MAC address on that subnet is not spoofed and hence with some network investigation the source would be pinpointed. The third disadvantage is similar to the disadvantage of the Half-Open scan technique which is rendered by the inability to create raw packets that do comply completely with the TCP handshake procedure without elevation of privileges, which is not always achievable.

*3) FIN, Xmas Tree, and Null Scans:* These three cyber scanning techniques are grouped together since their individual functionality is very similar. They are members of the 'stealth' scans because they send a single frame to a TCP port without any TCP handshaking or any additional packet transfers [62]. They operate identically to the SYN|ACK scan, but they differ by which flags they send. The FIN, the Xmas Tree and the Null scanning techniques send packets with the FIN flag, URG, PUSH, and FIN flags, and packets with empty flags, respectively. In all cases, the closed ports are required to reply to the probe packet with RST, while the open ports must ignore the packet in question [27, 63]. Note that, the Xmas Tree scan takes its name from the flags related to (00101001), which appear similar to the lights of a Christmas tree. The latter technique is depicted in Figure 10, while the FIN and Null scans have similar illustrations as Figure 8 but with different flags as previously mentioned. Since no TCP sessions are created for any of these scans, they are remarkably quiet from the perspective of the remote device applications. Therefore, none of these scans should appear in any of the application logs. These scans are some of the most minimal port-level scans that could be executed. For a closed port, only two packets are transferred. On the other hand, only a single frame is necessary to identify an open port. However, these techniques have two drawbacks. They are ineffective when used against Microsoft machines as all ports will appear to be closed regardless of their actual state. Nonetheless, this provides a backhanded advantage, since any device showing open ports must not be a Windows-based device. The second drawback is related to generating raw packets, which as mentioned earlier in this section, requires elevation of privileges.

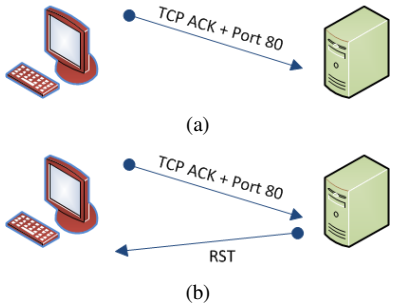



Fig. 11. The Ack Scan targeting a non-reachable (11(a)) and a reachable target (11(b))

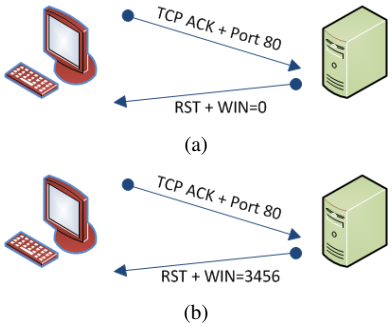



Fig. 12. The Window Scan targeting a closed (12(a)) and an open port (12(b))

*4) ACK Scan:* The ACK scan [64] is not intended to identify an open port. This stealth cyber scanning technique will only provide a filtered (non-reachable) or an unfiltered (reachable) disposition because it never connects to an application to confirm an 'open' state. At a first glance, this appears to be rather limiting but, in reality, the ACK scan can characterize the ability of a packet to traverse firewalls or packet filtered links. Another implementation of the ACK scan can take advantage of the IP routing function to deduce the state of the port from the time to live (TTL) value [65]. An ACK scan operates by sending a TCP ACK frame to a remote port. If there are no responses or an ICMP destination unreachable message is returned, then the port is considered to be filtered. If the remote port returns a RST packet, the connection between the source and the remote target is categorized as unfiltered. Figure 11 demonstrates the ACK scan process.

On one hand, and since the ACK scan does not open any application sessions, the conversation between the source and the remote target is relatively simple. Thus, the scan of a single port is almost invisible, especially when combined with other network traffic. On the other hand, the ACK scan's simplicity is also its largest disadvantage. Because it never tries to connect to a remote device, it can never definitively identify an open port. Although the ACK scan does not identify open ports, it does an impressive job of identifying ports that are filtered through a firewall. This list of filtered and unfiltered port numbers is extremely useful as a reconnaissance method for a future more detailed scan that focuses on specific port numbers and perhaps their vulnerabilities.

*5) Window Scan:* The Window scan, named after the TCP sliding window [27], is a scanning technique used with certain TCP stacks [66]. It is almost identical to the ACK scan, however, it has been found that certain TCP stacks return a window size number when responding to an ACK packet; a RST frame response from a closed port replies with a window size of zero and a RST frame response from an open port replies with a non-zero window size. Figure 12 shows the latter process.

An advantage of the window scan is that it does not open a session, hence there exists no application log associated with the scanning operation. Unless there are additional firewalls or network limits at the operating system level, the scan should go unnoticed. However, the window scan does not operate on all devices, and the number of operating systems vulnerable to this unintended window size consistency is dwindling as operating systems are upgraded and patched. In general, the window scan is useful when looking for open ports while simultaneously maintaining a low level of network traffic. When vulnerable operating systems are identified, the window scan provides a low impact method of locating open ports.

*6) TCP Fragmentation Scan:* This stealth scanning technique can be defined as a process of executing a scan rather than as a scanning technique by itself [67, 68]. It employs either the Half-Open scan or the FIN scan techniques to carry out its scanning methodology. This technique exploits the idea of decomposing the packet header (the probe packet) into smaller packets in an attempt to evade packet filters. This technique is effective since packet filters or intrusion detection systems do not buffer the entire set of packets due to performance issues. Rather, they process the packets individually causing the bypass of the assembled scanning packet. One possible drawback of this technique is rendered by the fact that some destinations do not have the ability to correctly merge the decomposed packets causing dropped probe packets and eventually the failure of this method. We refer the readers to the appendix for more information on how fragmentation (i.e., decomposition) and reassembly (i.e., merging) is implemented in this technique.

### *D. Sweep Scans*

In this section, we present the Sweep scans, which do not aim at identifying active ports but rather at identifying active hosts. They are characterized as performing sweeps, since their purpose is to identify the status of as many hosts as possible instead of focusing on an individual host. In fact, they typically utilize the network's subnet broadcast address as a destination address to target the majority of the hosts. They operate by generating any request that would prompt a remote station's response. They can be defined as cyber scanning facilitators because they pinpoint active hosts just before the actual scanning techniques of active hosts take place.

*1) ICMP Echo Request Scan:* This technique is one of the simplest and most known scanning technique to identify active

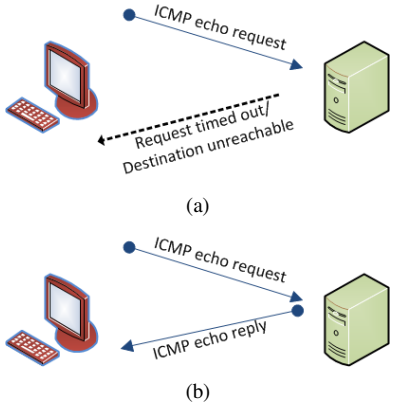



Fig. 13. The ICMP Echo Request targeting a non-active (13(a)) and an active host (13(b))

hosts [69, 70]. It is abundantly used on Windows and Linux machines by invoking the 'ping' command. The idea is to send an ICMP echo request message to the target and wait for a reply. If there is an ICMP Echo Reply message, this indicates that the target is active. Otherwise, it means that either the target is not active (request timed out reply) or that the original request never reached the target (destination unreachable reply) or that the target has discarded the request (dropped the probing packet). Figure 13 portrays this technique.

An advantage of the ICMP echo technique is that it does not depend on any particular application or open port to work. If a remote device communicates via TCP/IP, then it is most often a target candidate for the ICMP echo request scan. A disadvantage of this technique is that ICMP is one of the most filtered protocols in enterprise networks. Since the ICMP protocol has the ability to redirect traffic, identify available workstations, and pinpoint closed ports on a target, when a firewall or a packet filter is first installed, it is a common security guideline to restrict ICMP.

*2) ICMP Timestamp & Address Mask Scans:* These techniques take advantage of the seldom used ICMP messages (Timestamp & Address Mask) to determine if a remote target is active [71]. They function similarly to other ICMP-based scans; the source sends an ICMP Get Timestamp or Get Address Mask messages and waits for an ICMP Send Timestamp or ICMP Send Mask responses. The ICMP Timestamp scan is shown in Figure 14.

Both methods suffer from serious drawbacks. In both techniques, their corresponding probing messages are very rare to appear in a network and, thus, they can be very easily detected. Moreover, both do not achieve promising results when targeting relatively updated operating systems or networking hardware.

*3) TCP SYN Scan:* This cyber scanning facilitator technique is operationally identical to the Half-Open scan technique but the goal in this case is different. In this TCP SYN scan, the source scanner is awaiting a RST packet from a closed port or an ACK packet from an open port. The interesting and effective point of this technique is that either result from the scan will provide the source with a proof that

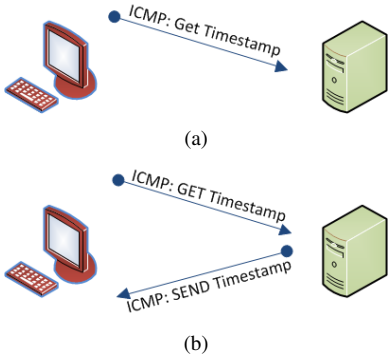



Fig. 14. The ICMP Timestamp scan targeting a non-active (14(a)) and an active host (14(b))

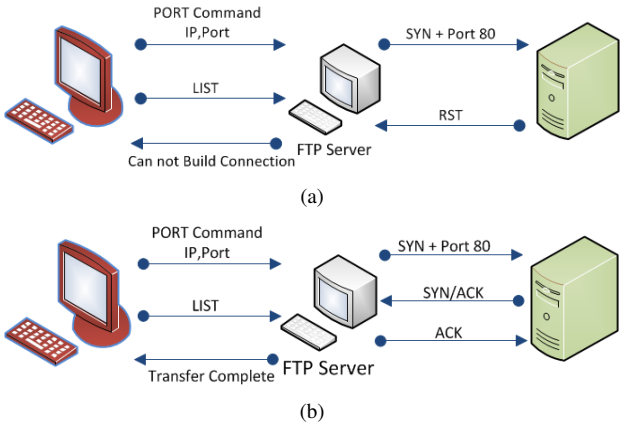



Fig. 15. The FTP Bounce scan targeting a closed (15(a)) and an open port(15(b))

an active system resides at that destination IP address. This technique is advantageous since it accomplishes its goal in just few packets. Such a minimal amount of network traffic appears to be similar to a typical TCP handshake. As a result, this technique can appear as legitimate network traffic and will go undetected.

### *E. Miscellaneous Scans*

This section aims at providing further cyber scanning insights by shedding light on scans that deal with various protocols. These include the FTP bounce, UDP, IP protocol and RPC scans.

*1) FTP Bounce Scan:* Similar to the IDLE scan, the FTP bounce attack [72] employs a third host (the FTP server) to act as a proxy between the source host (scanner) and the destination target. The FTP bounce attack takes advantage of the passive mode FTP [73]. This mode completely separates the command connections from the data connections. This allows the FTP server to be effective in the presence of firewalls since the FTP server would be responsible for building the outbound data connection with the remote host. Furthermore, it allows the source to send a PORT command [73] to the proxy FTP server where the latter will direct the data towards a completely different host (the target). The FTP Bounce scan is shown in Figure 15. The first step of

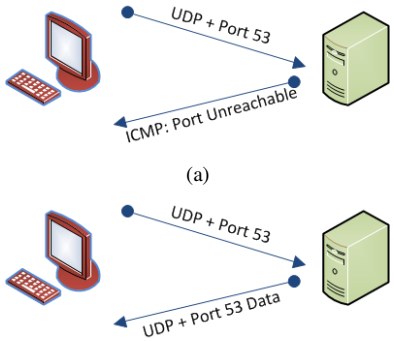



Fig. 16. The UDP scan targeting a closed (16(a)) and an open port (16(b))

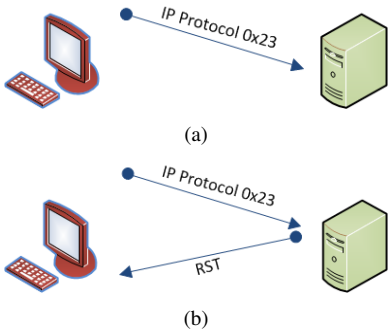



Fig. 17. Unavailable (17(a)) Vs. An Available IP Protocol (17(b))

the FTP Bounce scan occurs when the source connects to the vulnerable FTP server. Subsequently, the source transmits a `PORT` command [73] coupled with the IP address and the port of the target. The FTP server forwards that request to the target. If the intended target's port is closed (Figure 15(a)), the FTP server responds to the source stating the it can not build the connection. On the other hand, if the port is open (Figure 15(b)), the FTP server responds with a message stating that a transfer has been successfully completed. Depending on the reply, the source will infer if the port is open or closed. The advantages of this technique are two fold. First, the technique uses the standard FTP communication to achieve its task. Since the FTP service is found in the majority of enterprise networks, the technique seems feasible almost all the time. Second, it possesses stealth features as the source is using a proxy to direct the scan. The disadvantages of the FTP bounce scan comprise of the following: First, this scan technique is only successful on TCP ports. Since FTP does not connect to remote devices using UDP, it is not possible to retrieve any feedback on the availability of UDP ports. Second, the process of bouncing through an FTP server is slow when compared to other scanning methods. Additionally, the port scanning requests can only check a single port at a time. Third, and since this technique initiates an application session with the FTP server, the FTP servers will log the connection and all its commands making this method vulnerable to being detected. It is significant to note that the FTP Bounce Scan is also possible with secure FTP as long as the scanner is able to use the `PORT` command to request access to the ports. However, nearly all modern FTP servers are by default configured to refuse `PORT` commands that would connect to any host else than the originating host, thwarting FTP bounce attacks.

*2) UDP Scan:* UDP scan [74] does not require any SYNs, FINs, or any other handshaking flags. The lack of a formal communications process in UDP greatly amplifies the effectiveness of this scan technique. The UDP scan is demonstrated in Figure 16.

A closed port will reply with an ICMP port unreachable message while an open port responds with some UDP data. A critical advantage of this technique is that it operates very efficiently on Windows-based devices since they do not usually implement any type of ICMP rate limiting [75].

*3) IP Protocol Scan:* The goal of the IP protocol scan is to inquire about any additional IP protocols in use by the target station, including ICMP, TCP, and UDP [76]. If a router would be scanned, additional IP protocols such as EGP or IGP may be identified. Figure 17 depicts this technique. An unavailable IP protocol (i.e., unused protocol on the target such as for example, EGP on a machine that lacks routing capabilities) does not respond to the scan while an available IP protocol provides a response specific to the protocol type. An IP protocol scan looks fairly obvious if packet traces are investigated; since most networking protocols are based on TCP or UDP, any deviation from those two protocol types is conspicuous.

*4) RPC Scan:* In an attempt to disclose applications' information that operate using the remote procedure call (RPC) [28], the RPC scan sends RPC null messages to previously detected open ports [77]. If any RPC application is running on the target, the reply will include information such as the application's name, version and status. This technique possesses the ability to detect RPC applications running on non-RPC default ports. On the contrary, and since the technique establishes application sessions, its transaction events will be written in application logs, and, thus, the technique is easily detected. Another drawback of this technique is that it relies on previously detected open ports to operate and does not detect open ports by itself.

### *F. Discussion*

This section provides a brief discussion on the role and the effect of IP versions 4 and 6 (i.e., IPv4 & IPv6) and Network Address Translation (NAT) on scanning.

*1) IPv4 & IPv6:* One of the key differences between IPv4 and IPv6 is the much larger address space for IPv6, which also goes hand-in-hand with much larger subnet sizes. This change has a significant impact on the feasibility of TCP and UDP network scanning, whereby an automated process is run to detect open ports (services) on systems that may then be subject to a subsequent attack. Nowadays, many IPv4 sites are subjected to such probing on a recurring basis [17, 78]. Such probing is common in part due to the relatively dense

TABLE I
SUMMARY OF CYBER SCANNING TECHNIQUES

| CST | EOP | ID. T | ID. U | PR | SM | RMC | RMO | IFD |
|---|---|---|---|---|---|---|---|---|
| Open scan | - | √ | - | TCP | SYN | RST | ACK | - |
| Half-Open scan | √ | √ | - | TCP | SYN | RST | ACK | - |
| Version Detection scan | - | - | - | TCP | SYN | RST | ACK | - |
| SYN\|ACK scan | √ | √ | - | TCP | SYN/ACK | RST | - | √ |
| IDLE scan | - | √ | - | TCP | SYN/ACK, SYN | RST/IPID | RST/IPID | √ |
| FIN scan | √ | √ | - | TCP | FIN | RST | - | √ |
| XMAS scan | √ | √ | - | TCP | URG, PUSH, FIN | RST | - | √ |
| NULL scan | √ | √ | - | TCP | - | RST | - | √ |
| ACK scan | √ | √ | - | TCP | ACK | - | RST | √ |
| Window scan | √ | √ | - | TCP | ACK | RST+WIN | RST+WIN | √ |
| TCP Fragm. scan | √ | √ | - | TCP | SYN or FIN | RST | - | √ |
| ICMP Echo scan | - | - | - | ICMP | ICMP Echo | ICMP MSG | ICMP Reply | - |
| ICMP Timestamp scan | - | - | - | ICMP | ICMP Timestamp | - | Timestamp | - |
| ICMP Sub. Mask scan | - | - | - | ICMP | ICMP Sub.Mask | - | Sub. Mask | - |
| TCP SYN scan | - | - | - | TCP | SYN | RST | ACK | - |
| FTP Bounce scan | - | √ | - | TCP | PORT | Error MSG | Conn. Est. | - |
| UDP scan | - | - | √ | UDP | UDP Pkt | ICMP Unreach. | UDP Data | - |
| IP Protocol scan | - | - | - | IP | IP Prot. MSG | - | Protocols | - |
| RPC scan | - | - | - | RPC | RPC NULL | - | RPC App Info | - |

population of active hosts in any given chunk of IPv4 address space. The 128 bits of IPv6 address space is considerably larger than the 32 bits of address space in IPv4. In particular, the IPv6 subnets to which hosts attach will by default have 64 bits of host address space. As a result, traditional methods of remote TCP or UDP network scanning to discover open or running services on a host will potentially become less feasible, due to the larger search space in the subnet [79, 80]. Similarly, worms that rely on off-link network scanning to propagate may also potentially be more limited in impact.

A typical IPv4 subnet may have 8 bits reserved for host addressing. In such a case, a remote attacker only needs to scan at most 256 addresses to determine if a particular service is running publicly on a host in that subnet. Even at only one probe per second, such a scan would take under 5 minutes to complete. On the other hand, a typical IPv6 subnet will have 64 bits reserved for host addressing. In such a case, a remote attacker, in principle, needs to probe $2^{64}$ addresses to determine if a particular open service is running on a host in that subnet. At a very conservative one probe per second, such a scan may take several billion years to complete. A more rapid probe will still be limited to (effectively) infinite time for the whole address space. However, there are a number of ways for the attacker to reduce the address search space to scan against within the target subnet. For more information related to the above discussion, readers are referred to [81, 82].

*2) NAT:* In general, networks or devices behind a Network Address Translation (NAT) [83] end-point are somehow protected from probing packets. For instance, if a public (Internet) scanner is performing reconnaissance activity towards a single organization, the scanner will ultimately only probe the NAT end-point for open services. On the other hand, if an internal scanner within an organization performs scanning towards an Internet host, the external host will perceive the probing packets as arriving from the public IP of the NAT end-point. In the case where a public (Internet) scanner performs scanning towards a specific public server within an organization (such as a web server), then the probing packets will be forwarded by the NAT end-point to that server (the reply will also be forwarded from the server to the source). Hence, the NAT end-point will deal with the probing packets, regardless which scanning technique was used, as it typically deals with other network packets.

### G. Summary

The previously discussed cyber scanning techniques are summarized in Table I. The summary includes the cyber scanning technique (CST), whether or not it requires elevation of systems's privileges at the source to operate (EOP), whether it identities TCP or UDP ports (ID. T/ ID. U), the protocol it employs (PR), the messages that it sends (SM), the received messages when the target port is closed or the host is unreachable (RMC), the received messages when the target port is open or the host is reachable (RMO) and, finally, whether the technique is immune to firewall detection (IFD). From the summary table, we can extract the following:

- TCP is the most employed transport protocol used in cyber scanning.
- Although stealth scanning techniques are immune to being detected by a firewall, however, almost all except for the IDLE scan necessitate elevation of systems' privilege at the source to successfully operate.
- To identity UDP ports, only one cyber scanning technique can be utilized.
- The Half-Open scan technique can be used for port-identification as well as for detecting active network hosts.

## IV. LITERATURE REVIEW - DISTRIBUTED DETECTION TECHNIQUES

In this section, we present a review of the recent literature on distributed cyber scanning detection techniques. Distributed cyber scanning, which is illustrated in Figure 19, refers to

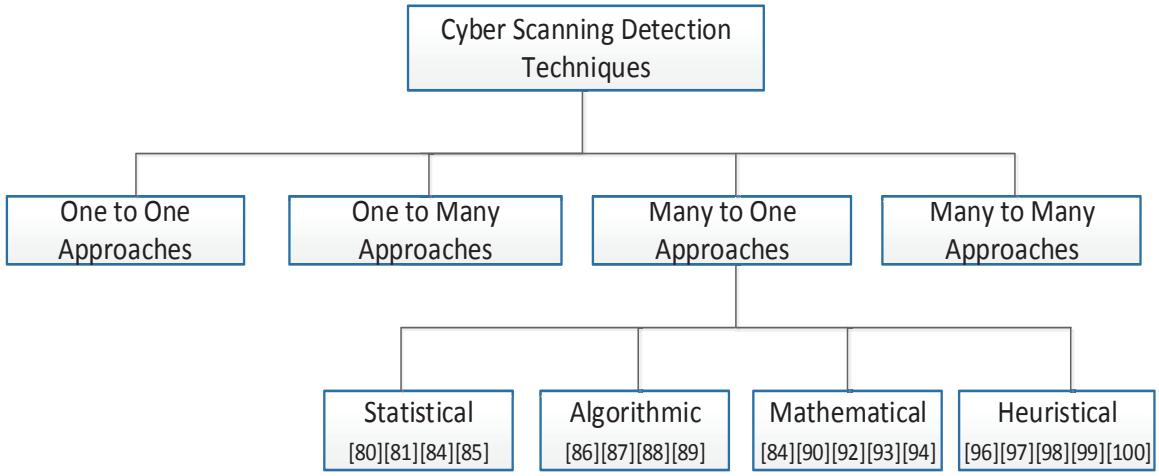



Fig. 18. Taxonomy-Distributed Cyber Scanning Detection Techniques

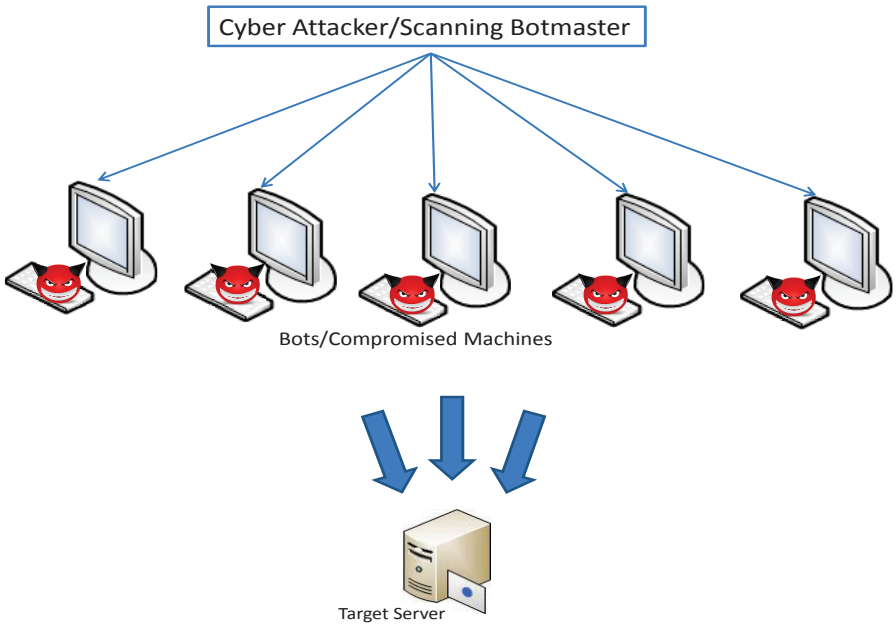



Fig. 19. Executing Distributed Cyber Scanning

the task of decomposing and coordinating the scanning using various compromised systems or bots. Typically, the scanning is controlled by a main attacker dubbed as the scanning botmaster who operates the command and control center and the entire network of bots (or botnet) for coordinated communication, propagation, and other attack activities. Distributed cyber scanning is often thought of as operating in a many (sources) to one (target) fashion, where the target system is often a single entity or a limited number of systems. Moreover, this type of scanning possesses stealth features and could be performed during a prolonged period of time (i.e., a slow scan).

The work presented in this section covers the period from 2001 up to November 2012. The studied literature solely focus on distributed detection techniques (many to one) and excludes single source detection techniques (one to one and one to many). For the latter, please refer to survey [20]. Many to many detection techniques, as briefly discussed in Section II-B, are yet to be investigated in the literature. The generated taxonomy on distributed cyber scanning detection techniques is shown in Figure 18. It is based on the employed approaches to achieve the detection task. The approaches are decomposed into four categories, namely, statistical, algorithmic, mathematical and heuristical.

### A. *Statistical Approaches*

These distributed cyber scanning detection approaches include techniques such as statistical characterization (features) of data samples, extrapolation or interpolation of data based on some best-fit, error estimates of observations, or spectral analysis of a data model.

Zhang et al. [84] proposed a scan detection method based on a distributed cooperative model. Their technique

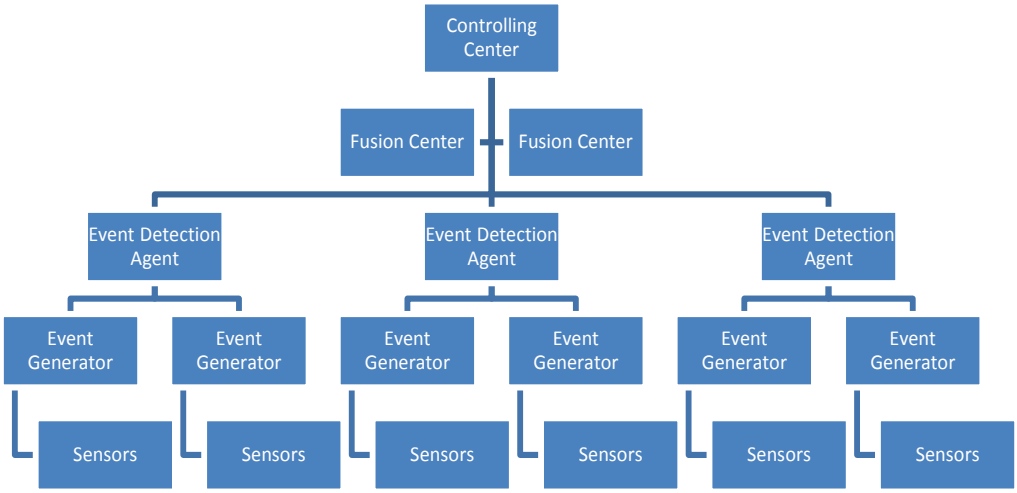



Fig. 20. Distributed Architecture of Cooperative Intrusion Detection [84]

is composed of feature-based detection, scenario-based detection (i.e., a scenario being variants of tuples of source and destination IPs, target ports and protocol flags) and statistic-based detection. Their proposed architecture, which is depicted in Figure 20, is decomposed into 5 layers; sensors, event generators, event detection agents, a fusion center, and a control center. The authors explained that the sensors collect data and system log information. Event generators check and filter data based on normal and abnormal information. Event detection agents detect the integrated data so as to decide whether the event is an intrusion behavior or not. The undetermined data is then sent to a fusion center for further analysis. The fusion center analyzes correlations and performs fusion analysis for the data submitted by event detection agents in order to increase the decision accuracy. Finally, the control center monitors, coordinates and adjusts each event detection agent and its corresponding load. The technique's statistic-based detection is based on predefined thresholds that allow the detection of both scan and denial of service attacks. The authors claim that their method not only can detect those scan attacks with obvious features, but can also detect the attack with stealth features and variants of the attack. To achieve the latter, the authors 1) presented custom SQL queries that are rendered by a predefined packet count threshold, a starting and an end time, and a detection time window and 2) built a feature linked list, used for storing numerous features of scan attempts that could capture variants of a number of different scan attacks.

A positive point of this paper [84] is that the proposed technique is well suited in a distributed large-scale environment. Moreover, the multi-layer architecture exploits the advantages of various approaches, including, statistical and scenario-based. On the other hand, it would have been beneficial if the authors had tested the accuracy of their technique against large-scale distributed scans.

In another work [85], Baig et al. proposed a time independent feature set model (IFSM) for the detection of slow, random and distributed cyber scanning activity. Their proposed technique is based on the observation that scanners, being unaware of systems and network topologies, send most of their probes to inactive hosts or closed ports resulting in many RST and ICMP packets. They designed a database that records information about that case and they took into consideration hosts that are behind Network Address Translation (NAT) [83] routers and those who use the dynamic host control protocol (DHCP) [86] server. They also developed an algorithm that implemented that technique in addition to a pruning method used when system memory runs low. Finally, the authors empirically tested their technique using DARPA's data set [87]. The results demonstrated that their proposed IFSM performs well for detecting slow and fast scans.

The work presented in [85] is a successful example of the usage of statistical feature-based elements in detecting cyber scanning. Nevertheless, this paper appears to suffer from two drawbacks. The technique presented in the paper is solely dependent on RST and ICMP packets. Thus, the technique might only detect scans that actually use or return those packets. Although the latter task can be a common behavior, a significant number of network devices do not allow the propagation of those packets back to the source. Second, it would have been interesting if the authors would have provided some empirical results demonstrating how their technique could be leveraged to detect slow and distributed cyber scanning.

Staniford et al. [88] presented a method for the detection of stealth port scans. Their technique is divided into two layers; the Stealthy Probing and Intrusion Correlation Engine (SPICE) and the Statistical Packet Anomaly Detection Engine (SPADE). Using an entropy-based metric, SPADE determines if a packet is malicious and then passes it to SPICE. The latter engine inserts the packet into a correlation graph, where the nodes represent packets and the connections between nodes contain weights indicating the strength of the relationship

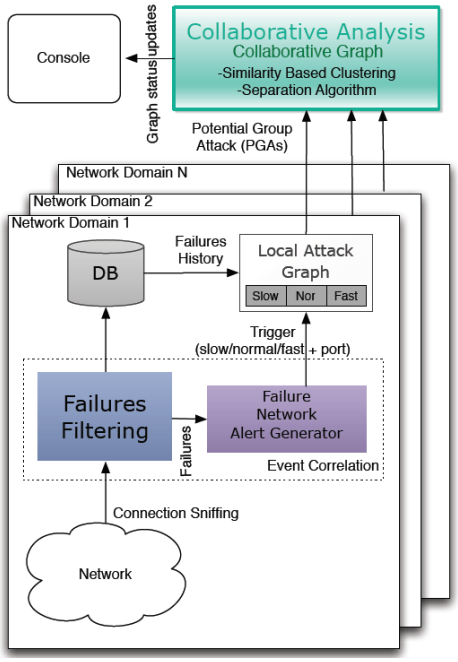



Fig. 21. Proposed Collaborative Architecture [89]

between the packets. The weights are based on a combination of four feature characteristics, namely, equality, proximity, separation, and covariance. In the final graph, all edges with weights less than a certain threshold are dropped, and the remaining subgraphs represent interesting network events.

The work done by Staniford et al. [88] appears to have the following weaknesses. Firstly, SPICE, apparently, was not designed specifically to detect coordinated scanning activity, rather it just forms clusters based on similar properties using the correlation graph. Secondly, it would have been significant if the authors would have reported the true and false negative and positive rates for their approach in addition to providing sufficient details to replicate their results.

### *B. Algorithmic Approaches*

These distributed cyber scanning detection approaches employ step-by-step procedures for calculations, data processing, and formal automated reasoning.

Baldoni et al. [89] proposed a collaborative architecture where each target network deploys local sensors that send alarms to a collaborative layer. This, in turn, correlates this data with the aim of (1) identifying coordinated cyber scanning activity while (2) reducing false positive alarms and (3) correctly separating groups of attackers that act concurrently on overlapping targets. The proposed architecture is illustrated in Figure 21. Locally deployed sensors adopt graph-based clustering algorithms over non-established TCP connections to generate alarms. The collaborative layer employed a similarity approach to aggregate alarms and approximated optimization algorithms to separate distinct group of attackers. The soundness of the proposed approach was tested on real network traces.

The above work, however, might have the following limitations. First, their proposed system is designed to leverage information coming from various network domains to detect distributed scanning. Hence, the collaborative layer appears to be ineffective when the adversary is acting only against one network domain. Second, their system assumed that the target set of an attack contains contiguous IP addresses, which is not always true. Third, if the distributed scanning is being generated by a large number of nodes, where each node only sends one or few packets, then the system might consider those as individual scans rather than correlating them.

In another research work on distributed cyber scanning detection [90], an approach was presented to detect coordinated attacks, based on adversary modeling of the desired information gain. A detection algorithm has also been developed that is based on solutions to the set covering problem, where the aim was to recognize coordinated activity by combining events such that a large portion of the information space is covered with minimal overlap. The author demonstrated the approach by developing a coordinated scan detector, where the targets of a port scan are distributed amongst multiple coordinating sources. The author elaborated that in this case, the adversary wishes to gain information about the active hosts and ports on a particular network. Moreover, the paper provided an algorithm that is capable of detecting horizontal and strobe scans against contiguous address spaces. Finally, the paper presented experimental results of the proposed algorithm in a controlled environment, demonstrating that it has an acceptably low false positive rate.

A possible limitation of the work in [90] is that the input for the proposed algorithm consists of single-source port scans. Thus, if an attacker can avoid detection by the single-source scan detector, then he/she also would avoid detection by the developed coordinated scan detector.

In an alternative work, Whyte et al. [91] discussed the notions of darkports and exposure maps. The former are unused ports on active systems while the latter is a technique rendered by passively characterizing the connectivity behavior of internal hosts in a network as they respond to both legitimate connection attempts and scanning attempts. Their proposed technique differs from other scanning detection techniques as they rely on identifying the services offered by the network instead of tracking external connection events. Additionally, they presented some methods to detect advanced cyber scanning activity such as distributed scanning. Finally, they evaluated their approach using three different real data sets.

The above work might have the following limitations. First, the proposed approach requires a prolonged training period (initializing time) to build the network map, possibly decreasing its chances from being operationally feasible. Second, the actual network map populating process is based on observed TCP SYN ACK. Nowadays, there exist a significant number of stealth cyber scanning activity that

never utilizes the TCP SYN ACK. Third, the authors' proposed heuristics to detect, attribute and match distributed cyber scanning is based on source IP grouping. They also considered clusters of three or more remote hosts that target the same destination ports as a distributed scan. This could be ineffective if the sources are spoofed, change regularly due to DHCP usage, or target different ports.

Furthermore, Yegneswaran et al. [92] presented a broad, empirical analysis of Internet intrusion activity using a large set of Network Intrusion Detection Systems (NIDS) and firewall logs. Their breakdown of scan types showed not only a large amount of worm activity but also a substantial amount of scanning activity. To gain insight into the global nature of intrusions, the authors used their data to project the activity across the global Internet. They also presented a high level information theoretic evaluation of the potential of using data shared between networks as a foundation for a distributed intrusion detection infrastructure. Their analysis indicated that small collections of logs from smaller networks may not be sufficient to identify either worst offenders or most popular port targets for attacks. Additionally, their research claimed to detect distributed scanning activity by defining a distributed scan as scans from multiple sources (five or more) aimed at a particular port of destinations in the same /24 subnet within an one hour window.

A potential drawback, related to the authors' definition of coordination or distributed scans, could be withdrawn from the above work. The definition misses several possible coordinated/distributed scans, such as scans from fewer than five sources, or scans where each source scans in a different hour. Additionally, it would have been beneficial if the authors had considered the case where completely unrelated sources might scan the same port on the same /24 subnet within the same hour. Their technique appears to neither report nor detect that case as a distributed cyber scanning activity.

### C. Mathematical Approaches

These distributed cyber scanning detection approaches utilize mathematical models, finite state machines and other algebraic and geometric techniques to achieve their detection task.

Treurniet J. of [93] presented an approach that is based on the idea that anomalous scanning activity can be detected using a finite state machine model that reflects the progression of a TCP connection through a sequence of states via its control flags. By storing such anomalies and applying correlation mechanisms, the author claim that slow and distributed scans could be detected. A proof of concept prototype was implemented which used both DARPA's data [87] and operational data injected with crafted anomalies to test the system. The author reported zero false negative and very few false positives.

The system proposed in this work [93] is evidently advantageous by its space requirements which makes it operationally feasible. On the other hand, this research work might be limited in the following: First, the experimental data was based on filtered data which might not accurately reflect the system's performance. Second, the system's implementation is based on MATLAB [94] which could possibly reduce its operational capabilities from an efficacy point of view. Third, the distributed correlation engine is based on simplistic criteria and appears to operate erroneously when the scan is destined to overlapping targets or ports.

In another work by the same author [95], a new system was proposed that is capable of detecting slow scans and distributed scans. The work was built on previous work [93] by refining the TCP model and adding support for UDP and ICMP. The proposed method is composed of two stages. First, sessions are formed from packet data using simple state machine models of TCP, UDP, and ICMP traffic. Second, common activities are identified in terms of groups of sessions which are referred to as activity patterns. The author verified that the system correctly identifies crafted slow scans injected into real traffic and found that most scans are below current detection thresholds. By combining the detected scans with the session directionality, the author was able to give context to the scan alerts and identify the scans that require immediate attention.

The research work presented above possesses the following advantages. First, it requires no training period and little knowledge of the local network configuration. Second, it successfully attempts to separate backscatter [96] from inverse mapping traffic. On the other hand, and although the author claimed that the system is able to detect distributed scans, the system might inaccurately group targeted distributed scans with other similar un-targeted scans.

Bhuyan et al. [97] presented the adaptive outlier based approach for coordinated scan detection (AOCD). Their proposed approach is based on two techniques. First, the principal component analysis based feature reduction technique was adopted to identify the relevant feature set. These feature sets are used during cluster formation. Second, a variant of the Fuzzy C-means clustering algorithm [98] was also employed to cluster information. Their algorithm also adopts an outlier scoring mechanism for each feature traffic data object and sequentially report it as malicious or not. The authors tested their algorithm using different real-life datasets and against other available literature techniques.

The work in [97] potentially has some limitations. Firstly, it requires a training period and hence 1) its accuracy could be affected when dealing with other new data and 2) it requires some initialization time which is not always feasible in an operational environment. Secondly, it would have been interesting if the demonstrated empirical results were validated with other scan types. Thirdly, their proposed approach assumed that the target of the scanning is a set of contiguous addresses, which might not always be the case.

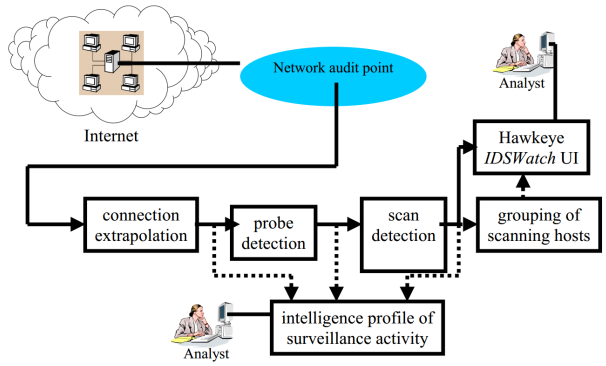



Fig. 22. Architecture of Surveillance Detection [99]

### D. Heursitical Approaches

These distributed cyber scanning detection approaches utilize non-formal expert based analysis including, but not limited to, visualization techniques, filter-based heuristics, previous incident analysis, and multidisciplinary techniques.

Robertson et al. [99] introduced the System Detection's surveillance techniques for enclave environments (ESD) and peering center environments (PSD). The system employed a cascading filter design, as depicted in Figure 22, which coordinated a series of specialized heuristics across connection records, individual probes, scans and coordinated scanning groups. Their proposed approach operates as follows. First, approximate sessions between source and destination IP pairs are extrapolated in accordance with a certain model. Second, each extrapolated session that represents a failed connection attempt is assumed to be a probe. Third, each probing IP is given a score based on the number of unique destination IP/port pairs probed. The IP is in turn considered a scanner if its score is greater than an empirically derived alert threshold. Their system was tested using real-time data and has shown to accurately discover great quantities of surveillance activities, including distributed scans.

The above system is advantageous in being scalable due to data reduction in the used filters and efficient in high bandwidth environments. However, on the other hand, their work assumed, with regards to distributed scanning activity, that a scanner is likely to use several IP addresses on the same subnet to carry on its probing act. 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. This assumption might not always be valid, especially when dealing with botnet scanning (See Section II-C2). Another possible limitation is that their proposed algorithm could be susceptible to decoys intended to cause false positives.

In a different research work, Choi et al. [100] presented the parallel coordinate attack visualization (PCAV) as illustrated in Figure 23.

PCAV displays network traffic on the plane of parallel coordinates using the packet flow information such as the source IP address, destination IP address, destination port and the average packet length in a flow. The parameters are used to draw each flow as a connected line on the plane, where a group of polygonal lines forms a particular shape in case of an attack. From the observation that each attack type possesses a unique pattern, the authors developed nine signatures coupled with their detection mechanisms based on an efficient hashing algorithm. The authors validated their proposed technique on three real network data samples and reported a very low false positive rate.

Although the authors asserted that their technique is able to detect and visualize distributed and coordinated scanning, they did not empirically validate that.

Stockinger et al., in [101], presented a multidisciplinary high-performance query-driven visualization technique for the purpose of anomaly detection. They combined indexing mechanisms with a new approach to visual analytics to efficiently populate visual histograms. Additionally, the authors applied the histogramming technology in conjunction with a specialized visual analytics application for analyzing distributed scans. They tested their system using network connection data that was collected by Bro [102] at a governmental location.

The work presented by Stockinger et al. appears to possess some drawbacks. First, their system is passive in the sense that it might be effective in the analysis of distributed scans but not in real-time detection. Second, since the technique is based on visualization, it is typically hard to provide numerical analysis of the technique's false positive and negative rates. It would have been interesting if the authors had provided some guidelines concerning that issue.

The authors of [103] discussed the application of visualization techniques to the problem of anomaly fingerprinting. This research work explored the application of several visualization techniques and their usefulness towards identification of attack tools and incidents. They used application, network and transport layer information to accomplish the visualization. The authors argued that their technique will aid other detection systems such as those using signature and statistical-based approaches to detect anomalies. Moreover, the authors briefly discussed the effectiveness of their technique in detecting distributed cyber scanning activity.

The work in [103] is interesting since it allows the identification of various scanning tools by visualizing their corresponding traffic. This allows the detection of new attack tools and types without replying on signature based systems that would typically fail in such scenarios. Nevertheless, it would have been valuable if the authors 1) have provided some metrics on how their system would have performed when operating on real-time data and 2) explained how clusters of distributed scanners are formed using their technique.

### E. Summary

This section presented a literature review by solely focusing on many to one cyber scanning detection techniques, commonly referred to as distributed approaches. From what has

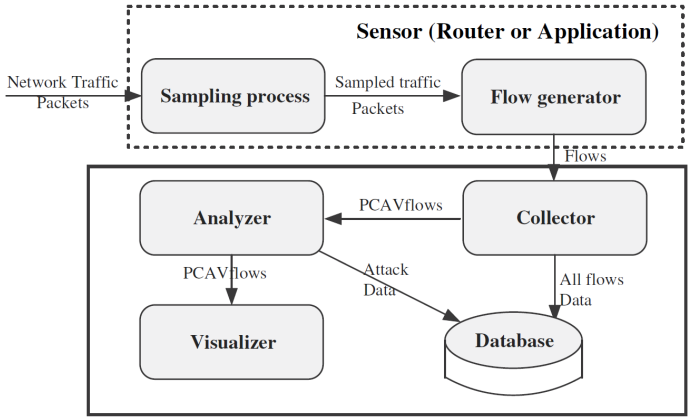



Fig. 23. PCAV System's Design [100]

```
   12:15:18.913184 IP (tos 0x0, ttl 36, id 20335, offset 0, flags [none], proto UDP
  XX.10.100.90.1878 > XX.164.30.56.5060: [udp sum ok] SIP, length: 384
REGISTER sip:3982516068@XX.164.30.56 SIP/2.0
Via: SIP/2.0/UDP XX.164.30.56:5060;branch=1F8b5C6T44G2CJt;rport
Content-Length: 0
From: <sip:3982516068@XX.164.30.56>; tag=14718138184028634232183426 68
Accept: application/sdp
User-Agent: Asterisk PBX
To: <sip:3982516068@XX.164.30.56>
Contact: sip:3982516068@XX.164.30.56
CSeq: 1 REGISTER
Call-ID: 4731021211
Max-Forwards: 70
```

Fig. 24. Snapshot-SIP Scan UDP Header [43]

| | |
|---|---|
| # of probes (1 probe = 1 UDP + multiple TCP pkts) | 20,255,721 |
| #of source IP addresses | 2,954,108 |
| # of destination IP addresses | 14,534,793 |
| % of telescope IP space covered | 86,6% |
| # of unique couples (source IP - destination IP) | 20,241,109 |
| max probes per second | 78.3 |
| max # of distinct source IPs in 1 hour | 160,264 |
| max # of distinct source IPs in 5 minutes | 21,829 |
| average # of probes received by a /24 | 309 |
| max # of probes received by a /24 | 442 |
| average # of sources targeting a destination | 1.39 |
| max # of sources targeting a destination | 14 |
| average # of destinations a source targets | 6.85 |
| max # of destination a source targets | 17613 |

Fig. 25. Snapshot-SIP Scan Campaign-Statistics [43]

been previously discussed, we can extract the following few points:

- In general, very limited work has been done targeting the problem of detecting distributed cyber scanning detection.
- Statistical methods are the least exploited to solve that problem.
- Algorithmic approaches, especially those utilizing clustering mechanisms, are the most effective techniques.
- There exist a lack of effective, accurate and efficient distributed source scanning clustering techniques.

## V. Reports on Cyber Scanning Campaigns

Recently, there has been a noteworthy shift towards a new phenomenon of probing events known as cyber scanning campaigns. These are distinguished from previous scanning incidents as 1) the population of the participating bots is several orders of magnitude larger, 2) the target scope is generally the entire Internet Protocol (IP) address space, and 3) the bots adopt well-orchestrated, often botmaster-coordinated, stealth scan strategies that maximize targets' coverage while minimizing redundancy and overlap. This section reports on the analysis of two recent incidents of cyber scanning campaigns. Recall that, we have defined cyber scanning campaigns in Section II-B, as operating in a many (sources) to many (destinations). The first cyber scanning campaign was recently reported in [43]. The second cyber scanning campaign was discussed online [44], however, to the best of our knowledge, no analysis was reported on it. We contribute in this section by highlighting on the first in addition to providing a preliminary analysis of the second scanning campaign using real one-way data (i.e., darknet data [104]) that we have in our lab.

### A. SIP Scanning Campaign

Dainotti et al. [43] presented the measurement and analysis of a 12-day world-wide cyber scanning campaign targeting VoIP (SIP) [105, 106] servers. Their analysis is based on their collected data using UCSD Network Telescope, a /8 darknet [107]. In a nutshell, a darknet is Internet traffic destined to routable but unused Internet addresses. Since these addresses are unallocated, any traffic targeting them is suspicious and hence need to be investigated. Darknet analysis is an effective method to generate cyber threat intelligence [108–110].

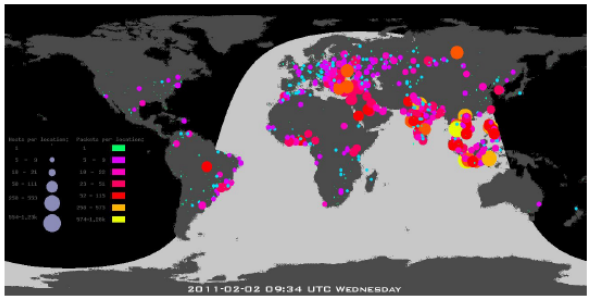



Fig. 26. Snapshot-SIP Scan World Map [43]

As reported by the authors, a partial anatomy of the SIP scanning campaign is depicted in Figures 24 and 25.

The SIP campaign approximately involved a significant 3 million distinct source addresses (scanning bots), generated around 20 million probes and targeted around 14.5 million destinations.

Moreover, the authors created a world map animation of the scanning campaign as illustrated in Figure 26. The snapshot represents, for the first time, an Internet-wide scan conducted by a large botnet.

To proof that the sources of the scan were not spoofed, Dainotti et al. [43] presented logical and empirical-based evidence. Moreover, they showed that the SIP scanning campaign in fact targeted the entire IPv4 address space by using darknet data from two other sources.

To understand in which fashion the SIP scan accomplished its probing, the authors drew Hilbert's space-filling curves [111]. They discovered that by reversing the order of the three varying bytes (recall that the monitoring network is a /8), the bots perfectly coordinated towards filling the entire address space. Further analysis by the researchers revealed that more than 1 million bots sent a single probe and never participated further in the scan. It is worthy to note that, the latter technique used by the SIP scan would go undetected by all current detection mechanisms.

The authors also elaborated on the phases of the scanning campaign, provided geo-location information about the sources, and finally discussed the results of their binary analysis of the scannning.

We refer the readers to [43] for detailed insights about this SIP cyber scanning campaign.

### *B. MS-SQL Scanning Campaign*

On October 10, 2012, the Internet Storm Center [112] received a report of a large distributed SQL Injection Scan [44]. The report noted that the scanning campaign involved more than 9 thousand bot sources and apparently was targeting Microsoft SQL servers. To the best of our knowledge, no analysis in the literature was reported on it.

TABLE II
STATISTICS-SQL CYBER SCANNING CAMPAIGN

| NoP | USIP | UDIP | MSTD | ASTD | MDTS | ADTS |
|---|---|---|---|---|---|---|
| 216853 | 35 | 204648 | 4 | 1.06 | 181353 | 6195.8 |

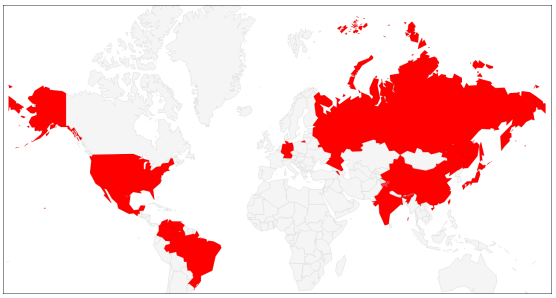


Fig. 27. Sources Heat Map-SQL cyber scanning campaign

To further contribute in this paper, we present in this section a proof and a preliminary analysis of that reported cyber scanning campaign activity targeting the widely deployed SQL database infrastructures.

We have received darknet traffic from many /16 sensors from a trusted third party for the month of October of 2012. We utilize a one day sample (October 3, 2012) of this real network one-way data to report some preliminary findings. By doing this, we concur the evidence of that reported cyber scanning campaign in addition to briefly shedding the light on its details.

Table II summarizes some details about the SQL cyber scanning campaign. The table discloses the number of probes (NoP), the unique number of source IPs (USIP), the unique number of destination IPs (UDIP), the maximum number of sources targeting a destination (MSTD), the average number of sources targeting a destination (ASTD), the maximum number of destinations a source targets (MDTS) and the average number of destinations a source targets (ADTS).

It is revealed that, per that specific darknet one day data sample, the SQL cyber scanning campaign generated approximately 200 thousand probes, from 35 unique sources (bots) and targeted around 205 thousand destinations. Bearing in mind that such statistics are extracted from only a one-day sample, these numbers are considered very significant and concur that this phenomenon is in fact a cyber scanning campaign and not just a simplistic scanning incident.

The heat map of the sources of the campaign is illustrated in Figure 27. Republic of Korea, China and the United States are among the top sources of this cyber scanning campaign.

Moreover, the probe packets, which constituted of ICMP echo requests to ports 80 (web) and 1433 (MS-SQL), have an average time to live (TTL) value of 114.9 and a datagram length of 404. Such features should allow us to build a unique payload signature to identify all the scanning packets. Therefore, we expect to provide an elaborative analysis and

tracking for this SQL cyber scanning campaign in the near future.

### C. Summary

To tackle cyber scanning campaigns, this section reported on the analysis of two recent cyber scanning incidents. The cyber scanning campaigns targeted Internet-wide voice over IP and database infrastructures. The SIP scanning campaign scanned the entire IP address space targeting more than 14 million destinations. Additionally, it used advanced stealth techniques and coordination to achieve its campaign. Using real darknet data, we shed light on the MS-SQL scanning campaign. By this, we demonstrated a proof of the existence of that campaign and presented some preliminary analysis. It is worthy to note that detection mechanisms of such campaigns are yet to be investigated throughout the literature.

## VI. Conclusion and Lessons Learned

The ever increasing population and adoption of cyberspace has been a great asset both socially and economically. The complete embracing of cyberspace technologies allowed the creation and implementation of new ideas that tremendously facilitate everyday tasks. Critical infrastructure heavily depend on information and communication technologies to operate successfully. However, recent events demonstrated that cyberspace could be subjected to amplified, debilitating and disrupting attacks that might lead to severe security issues with drastic consequences.

To tackle the ever increasing concern about cyber scanning, which is the core facilitator for those cyber security incidents, in this paper, we provided a categorization of the entire cyber scanning topic. This offered the readers a strong, coherent and a clear entry point into the topic. Further, we presented a classification for cyber scanning techniques and thoroughly discussed those in addition to their advantages and disadvantages. We also focused and elaborated on distributed cyber scanning detection methodologies, our current research interest. Finally, we contributed by highlighting on a new phenomenon dubbed as cyber scanning campaigns and presented the analysis of two of its recent incidents targeting two diverse Internet wide infrastructures. In this context, we described and pinpointed a reported cyber scanning campaign in addition to performing our own preliminary analysis of an unreported incident using real data samples.

From what has been presented and discussed throughout this paper, we can extract the following points:

- Cyber scanning is a significant and a timely cyber security problem.
- Cyber scanning campaigns present a new paradigm in the area of probing events. The need for a generic approach to automate the detection and identification of such campaigns render a new challenging cyber security problem.
- Cyber scanning could be a precursor of various cyber attacks, including but not limited to, denial of service attacks, malware infections and propagation, phishing and spamming.
- A firewall that employs TCP filtering can prevent or at least detect around 68% of probing activities.
- There still exists a need for more research work targeting the problem of detecting distributed cyber scanning.
- The cyber security community would benefit from more effective and accurate statistical approaches to tackle the problem of detecting distributed cyber scanning.

## Appendix

In this appendix, we provide, for the purpose of paper self-containment, 1) a brief description of the functionality of the TCP flags that were pinpointed in Section III and 2) an explanation on how the TCP Fragmentation Scan, that was discussed in Section III-C6, operates.

1. TCP Flags:

**SYN Flag**: The procedure of establishing a connection between any two hosts necessitates the utilization of the synchronize (SYN) control flag and involves an exchange of three messages; the TCP/IP three-way hand shake. A connection is initiated by an arriving segment containing a SYN flag. The matching of local and foreign sockets determines when a connection has been initiated. The connection is established when sequence numbers have been synchronized in both directions.

**ACK Flag:** The acknowledgment (ACK) flag is used to acknowledge the successful receipt of packets.

**FIN Flag:** The finished (FIN) flag is used to tear down the connection that was previously created using the SYN flag. FIN flags always appear when the last packets are exchanged between any two entities during an established connection.

**RST Flag:** The reset (RST) flag is used when a segment arrives that is not intended for the current connection. In other words, if packets are sent to a host in order to establish a connection, and there was no such service waiting to answer at the remote host, then the host would automatically reject the request and subsequently send a reply with the RST flag set. This indicates that the remote host has reset the connection.

**URG Flag:** The URG flag allows the marking of a segment of data as 'urgent'. Such incoming segments do not have to wait until the previous segments are consumed by the receiving entity but rather are sent directly and processed immediately. In such a scenario, an urgent pointer field specifies exactly where the urgent data terminates.

**PUSH Flag:** The PUSH flag is used to 1) inform the sending application that the data should be immediately sent out and 2) inform the receiving host that the data should

instantaneously be pushed to the receiving application. The PUSH flag is an effective method to avoid the typical TCP buffering mechanism and hence is extensively used in real-time applications.

For more detailed information about the TCP header flags, we refer the readers to [52, 113].

2. TCP Fragmentation Scan:

Three fields in the IP header are used to implement fragmentation and reassembly. These are the 'Fragment Offset', 'Identification' and 'More Fragment' fields. The 'Fragment Offset' specifies the fragment's position within the original datagram, measured in 8-byte units. Accordingly, every fragment except the last must contain a multiple of 8 bytes of data. By default, the 'Fragment Offset' can hold up to 8192 ($2^{13}$) units; the datagram can not contain $8192 \times 8 = 65536$ bytes of data. Since an IP header is at least 20 bytes long, the maximum value for 'Fragment Offset' is restricted to 8189 bytes, which dictates that only 3 bytes remain for the last fragment. An IP transport can be connectionless and thus fragments from one datagram may be interleaved with those from another at the destination. The 'Identification' field uniquely identifies the fragments of a particular datagram. The source system sets the 'Identification' field in each datagram to a unique value for all datagrams, which uses the same source IP address, destination IP address, and protocol values, for the lifetime of the datagram. This way the destination can distinguish which incoming fragments belong to which unique datagram and hence buffer all of them until the last fragment is received. The last fragment sets the 'More Fragment' bit to 0 informing the receiving station to start reassembling the data since all the fragments have been successfully and correctly received.

## ACKNOWLEDGMENT

The authors are grateful for Concordia University and the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this work. The first author is supported by the Alexander Graham Bell Canada Graduate Scholarship (CGS) from NSERC.

## REFERENCES

[1] Government of Canada. Canada's Cyber Security Strategy Report, 2010. http://www.capb.ca/uploads/files/documents/Cyber_Security_Strategy.pdf; Last accessed: 25/10/2012.
[2] The Whitehouse. CyberSpace Policy Review, 2009. http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf; Last accessed: 25/10/2012.
[3] Government of Canada. Service Canada, 2012. http://www.servicecanada.gc.ca/eng/home.shtml; Last accessed: 25/10/2012.
[4] Stephen Hinde. The law, cybercrime, risk assessment and cyber protection. *Computers & Security*, pages 90–95, 2003.
[5] Yoo Chung. Distributed denial of service is a scalability problem. *SIGCOMM Comput. Commun. Rev.*, 42(1):69–71, January 2012.
[6] M.K. Daly. Advanced persistent threat. *Usenix, Nov*, 4, 2009.
[7] Leyla Bilge and Tudor Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proc. 2012 ACM Conf. on Computer and Communications Security*, CCS '12, pages 833–844, New York, NY, USA, 2012. ACM.
[8] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124 – 140, 2010.
[9] Symantec. W32.Stuxnet Dossier, 2012. http://tinyurl.com/36y7jzb; Last accessed: 25/10/2012.
[10] DefenseTech. Cyber War 2.0, Russia v. Georgia, 2012. http://tinyurl.com/8l7cvm8; Last accessed: 25/10/2012.
[11] GovCon Technology. Cyber Attacks Fifth Dimension of Warfare, Says NATO Official, 2012. http://tinyurl.com/yad4z49; Last accessed: 25/10/2012.
[12] The Globe and Mail. Ottawa needs to improve cyber security: Auditor General, 2012. http://tinyurl.com/8n5sl7p; Last accessed: 25/10/2012.
[13] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. Inferring internet denial-of-service activity. *ACM Trans. Comput. Syst.*, 24(2):115–139, May 2006.
[14] Felix Freiling, Thorsten Holz, and Georg Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. In *Computer Security, ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 319–335. Springer Berlin / Heidelberg, 2005. 10.1007/11555827_19.
[15] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proc. 6th ACM SIGCOMM Conf. on Internet measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM.
[16] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. In *Proc. Network and Distributed System Security Symp. (NDSS)*, 2004.
[17] S. Panjwani, S. Tan, K.M. Jarrin, and M. Cukier. An experimental evaluation to determine if port scans are precursors to an attack. In *Proc. Int. Conf. Dependable Systems and Networks, 2005. DSN 2005.* , pages 602 – 611, june-1 july 2005.
[18] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security Privacy*, 1(4):33 – 39, july-aug. 2003.
[19] Richard J Barnett and Barry Irwin. Towards a taxonomy of network scanning techniques. In *Proc. 2008 annu. research conf. South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, SAICSIT '08, pages 1–7, New York, NY, USA, 2008. ACM.
[20] M.H. Bhuyan, DK Bhattacharyya, and JK Kalita. Surveying port scans and their detection methodologies. *The Computer Journal*, 54(10):1565–1581, 2010.
[21] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.
[22] Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666, 2004.
[23] Zhaosheng Zhu, Guohan Lu, Yan Chen, Zhi Judy Fu, Phil Roberts, and Keesook Han. Botnet research survey. In *32nd Annu. IEEE Int. Comput. Software and Applications, 2008. COMPSAC'08.* , pages 967–972. IEEE, 2008.
[24] Nwokedi Idika and Aditya P Mathur. A survey of malware detection techniques. *Purdue University*, page 48, 2007.
[25] Christopher Abad. The economy of phishing: A survey of the operations of the phishing market. *First Monday*, 10(9), 2005.
[26] Henry Stern et al. A survey of modern spam tools. In *Proc. 5th Conf. on Email and Anti-Spam*. Citeseer, 2008.
[27] W.R. Stevens and G.R. Wright. *TCP/IP Illustrated: the protocols*, volume 1. Addison-Wesley Professional, 1994.
[28] R. Thurlow. Rpc: Remote procedure call protocol specification version 2. 2009.
[29] J. Medeiros, A. Brito, and P. Pires. A data mining based analysis of nmap operating system fingerprint database. *Computational Intelligence in Security for Information Systems*, pages 1–8, 2009.
[30] G.F. Lyon. Nmap network scanning: The official nmap project guide to network discovery and security scanning author: Gordon fyodor l. 2009.
[31] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli. Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting. *Computer Networks*, 53(1):81–97, 2009.
[32] D. Stuttard and M. Pinto. *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley, 2011.
[33] Xianghua Xu, Jian Wan, Wei Zhang, Chao Tong, and Changhua Wu. Pmsw: a passive monitoring system in wireless sensor networks. *Int. J. of Network Management*, 21(4):300–325, 2011.
[34] G. Combs. Wireshark network analyzer-user's guide. 2008.
[35] M. Shelton. Passive Asset Detection System, 2008. http://passive.sourceforge.net/about.php; Last accessed: 25/10/2012.
[36] T. Socolofsky and C. Kale. TCP/IP Tutorial. Technical report, RFC 1180, Spider Systems Ltd, 1991.

[37] H. Al-Bahadili and A.H. Hadi. Network security using hybrid port knocking. *IJCSNS*, 10(8):8, 2010.

[38] David Whyte, Paul C. van Oorschot, and Evangelos Kranakis. Tracking darkports for network defense. In *ACSAC*, pages 161–171, 2007.

[39] Elias Bou-Harb, Makan Pourzandi, Mourad Debbabi, and Chadi Assi. A secure, efficient, and cost-effective distributed architecture for spam mitigation on lte 4g mobile networks. *Security and Communication Networks*, 2012.

[40] David Whyte, Evangelos Kranakis, and P Van Oorschot. Dns-based detection of scanning worms in an enterprise network. In *Network and Distributed Systems Symp. (NDSS)*, 2005.

[41] P. Li, M. Salour, and X. Su. A survey of internet worm detection and containment. *IEEE Commun. Surveys & Tutorials*, 10(1):20–35, 2008.

[42] Cliff Changchun Zou, Weibo Gong, and Don Towsley. Code red worm propagation modeling and analysis. In *Proc. 9th ACM conf. on Comput. and commun. security*, pages 138–147. ACM, 2002.

[43] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescap. Analysis of a "/0" Stealth Scan from a Botnet. In *Internet Measurement Conf. (IMC)*, Nov 2012.

[44] Internet Storm Center. Reports of a Distributed Injection Scan, 2012. http://isc.sans.edu/diary.html?storyid=14251; Last accessed: 28/10/2012.

[45] Internet Census 2012-Port scanning /0 using insecure embedded devices. http://tinyurl.com/c8af8lt.

[46] A. Boulanger. Unauthorized intrusions and denial of service. *Cybercrimes: A Multidisciplinary Analysis*, pages 27–44, 2010.

[47] N. Hachem, Y. Ben Mustapha, G.G. Granadillo, and H. Debar. Botnets: lifecycle and taxonomy. In *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, pages 1–8. IEEE, 2011.

[48] Zhichun Li, Anup Goyal, Yan Chen, and Vern Paxson. Automating analysis of large-scale botnet probing events. In *Proc. 4th Int. Symp. on Information, Computer, and Communications Security*, ASIACCS '09, pages 11–22, New York, NY, USA, 2009. ACM.

[49] K. Ko, H. Jang, B. Park, and Y. Eom. Analysis of the propagation pattern of a worm with random scanning strategy based on usage rate of network bandwidth. *Information, Security and Cryptology–ICISC 2009*, pages 374–385, 2010.

[50] C. Gates. Coordinated scan detection. In *Proc. 16th Annu. Network and Distributed System Security Symp. (NDSS 09)*, 2009.

[51] L. Aniello, G. Di Luna, G. Lodi, and R. Baldoni. A collaborative event processing system for protection of critical infrastructures from cyber attacks. *Computer Safety, Reliability, and Security*, pages 310–323, 2011.

[52] Information Sciences Institute University of Southern California. Darpa Internet Program Protocol Specification, 2012. http://www.ietf.org/rfc/rfc793.txt; Last accessed: 22/4/2013.

[53] Atul Kant Kaushik, Emmanuel S Pilli, and RC Joshi. Network forensic system for port scanning attack. In *Advance Computing Conference (IACC)*, pages 310–315. IEEE, 2010.

[54] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan. Tcp fast open. In *Proc. 7th Conf. on emerging Networking EXperiments and Technologies*, page 21. ACM, 2011.

[55] Atul Kant Kaushik, Emmanuel S Pilli, and RC Joshi. Network forensic system for port scanning attack. In *Advance Computing Conf. (IACC)*, pages 310–315. IEEE, 2010.

[56] Maciej Korczynski, Lucjan Janowski, and Andrzej Duda. An accurate sampling scheme for detecting syn flooding attacks and portscans. In *Int. Conf. Commun. (ICC)*, pages 1–5. IEEE, 2011.

[57] David Brumley, Juan Caballero, Zhenkai Liang, James Newsome, and Dawn Song. Towards automatic discovery of deviations in binary implementations with applications to error detection and fingerprint generation. In *Usenix Security Symp.*, 2007.

[58] Lee Garber. Denial-of-service attacks rip the internet. *IEEE Computer*, 33(4):12–17, 2000.

[59] Phurivit Sangkatsanee, Naruemon Wattanapongsakorn, and Chalermpol Charnsripinyo. Practical real-time intrusion detection using machine learning approaches. *Computer Communications*, 34(18):2227–2235, 2011.

[60] Seong-Cheol Hong, Hong-Taek Ju, and James W Hong. Ip prefix hijacking detection using idle scan. In *Management Enabling the Future Internet for Changing Business and New Computing Services*, pages 395–404. Springer, 2009.

[61] Roya Ensafi, Jong Chun Park, Deepak Kapur, and Jedidiah R Crandall. Idle port scanning and non-interference analysis of network protocol stacks using model checking. In *Proc. of USENIX Security Symp.*, 2010.

[62] J Udhayan, M Muruga Prabu, V Aravinda Krishnan, and R Anitha. Reconnaissance scan detection heuristics to disrupt the pre-attack information gathering. In *Int. Conf. on Network and Service Security, 2009. N2S'09.* , pages 1–5. IEEE, 2009.

[63] W. John, S. Tafvelin, and T. Olovsson. Trends and differences in connection-behavior within classes of internet backbone traffic. *Passive and Active Network Measurement*, pages 192–201, 2008.

[64] Richard Clayton. Failures in a hybrid content blocking system. In *Privacy Enhancing Technologies*, pages 78–92. Springer, 2006.

[65] J. Gadge and A.A. Patil. Port scan detection. In *16th IEEE Int. Conf. on Netw., 2008. ICON 2008.*, pages 1–6. IEEE, 2008.

[66] Jitendra Pahdye and Sally Floyd. On inferring tcp behavior. In *ACM SIGCOMM Computer Communication Review*, volume 31, pages 287–298. ACM, 2001.

[67] Jon Oberheide and Manish Karir. Honeyd detection via packet fragmentation. *Ann Arbor*, 1001:48104, 2006.

[68] Jayant Gadge and Anish Anand Patil. Port scan detection. In *16th IEEE Int. Conf. Netw., (ICON)*, pages 1–6. IEEE, 2008.

[69] Kohei Ohta, Glenn Mansfield, Yohsuke Takei, Nei Kato, and Yoshiaki Nemoto. Detection, defense, and tracking of internet-wide illegal access in a distributed manner. In *Proc. 10th Annu. Internet Society Conf. (INET 2000)*, 2000.

[70] David Mankins, Rajesh Krishnan, Ceilyn Boyd, John Zao, and Michael Frentz. Mitigating distributed denial of service attacks with dynamic resource pricing. In *Proc. 17th Annu. Computer Security Applications Conf., 2001. ACSAC 2001.* , pages 411–421. IEEE, 2001.

[71] Ofir Arkin. Icmp usage in scanning. *The Complete Know-How*, 3, 2001.

[72] Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller, and Robyn Lutz. A software fault tree approach to requirements analysis of an intrusion detection system. *Requirements Engineering*, 7(4):207–220, 2002.

[73] I. Van Beijnum. An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation. 2011.

[74] Avinash Sridharan, Tao Ye, and Supratik Bhattacharyya. Connectionless port scan detection on the backbone. In *25th Int. Conf. on Performance, Computing, and Communications, (IPCCC)*, pages 10–pp. IEEE, 2006.

[75] G. Gont, C. Pignataro, and F. Gont. Recommendations for filtering icmp messages. 2012.

[76] Satish Shetty. Protocol-level malware scanner, August 3 2004. US Patent 6,772,345.

[77] Vinod Yegneswaran, Paul Barford, and Vern Paxson. Using honeynets for internet situational awareness. In *Proc. 4th Workshop on Hot Topics in Networks (HotNets IV)*, pages 17–22. Citeseer, 2005.

[78] Zesheng Chen, Chao Chen, and Chuanyi Ji. Understanding localized-scanning worms. In *Performance, Computing, and Communications Conf.*, pages 186–193. IEEE, 2007.

[79] Viney Sharma. Ipv6 and ipv4 security challenge analysis and best-practice scenario. *Int. J. of Advanced of Networking and Applications*, 1(04):258–269, 2010.

[80] Sean Convery and Darrin Miller. Ipv6 and ipv4 threat comparison and best-practice evaluation (v1. 0). *Presentation at the 17th NANOG*, 2004.

[81] E Davies, Suresh Krishnan, and Pekka Savola. Ipv6 transition/co-existence security considerations. *draft-ietf-v6ops-security-overview-06 (work in progress)*, 2006.

[82] Steven M Bellovin, Bill Cheswick, and Angelos Keromytis. Worm propagation strategies in an ipv6 internet. *LOGIN: The USENIX Magazine*, 31(1):70–76, 2006.

[83] K. Egevang and P. Francis. The ip network address translator (nat). Technical report, RFC 1631, may, 1994.

[84] W. Zhang, S. Teng, and X. Fu. Scan attack detection based on distributed cooperative model. In *12th Int. Conf. on Computer Supported Cooperative Work in Design, (CSCWD)*, pages 743–748. IEEE, 2008.

[85] H.U. Baig and F. Kamran. Detection of port and network scan using time independent feature set. In *Intelligence and Security Informatics*, pages 180 –184, may 2007.

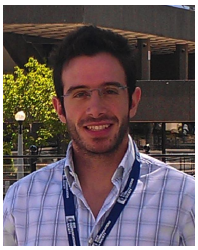
[86] R. Droms. Automated configuration of tcp/ip with dhcp. *IEEE Internet Comput.*, 3(4):45–53, 1999.

[87] MIT. 1999 DARPA Intrusion Detection Evaluation Data Set. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html; Last accessed: 27/10/2012.

[88] S. Staniford, J.A. Hoagland, and J.M. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10(1/2):105–136, 2002.

[89] R. Baldoni, G. Di Luna, and L. Querzoni. Collaborative Detection of Coordinated Port Scans. Technical report, 2012. http://www.dis.

uniroma1.it/~midlab; Last accessed: 27/10/2012.
[90] C. Gates. Coordinated scan detection. In *Proc. 16th Annu. Netw. and Distributed System Security Symp. (NDSS 09)*, 2009.
[91] D. Whyte, P.C. Oorschot, and E. Kranakis. Tracking darkports for network defense. In *Computer Security Applications Conf., 2007. ACSAC 2007. 23rd Annu.*, pages 161–171. IEEE, 2007.
[92] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *ACM SIGMETRICS Performance Evaluation Review*, volume 31, pages 138–147. ACM, 2003.
[93] J. Treurniet. Detecting low-profile scans in tcp anomaly event data. In *Proc. 2006 Int. Conf. on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, page 17. ACM, 2006.
[94] D. Hanselman and B.C. Littlefield. *Mastering MATLAB 5: A comprehensive tutorial and reference*. Prentice Hall PTR, 1997.
[95] J. Treurniet. A network activity classification schema and its application to scan detection. *IEEE/ACM Trans. Netw.*, 19(5):1396–1404, 2011.
[96] A. Dainotti, R. Amman, E. Aben, and K.C. Claffy. Extracting benefit from harm: using malware pollution to analyze the impact of political and geophysical events on the internet. *ACM SIGCOMM Comput. Commun. Review*, 42(1):31–39, 2012.
[97] M.H. Bhuyan, D.K. Bhattacharyya, and J.K. Kalita. Aocd: An adaptive outlier based coordinated scan detection approach. *International Journal of Network Security*, 14(6):339–351, 2012.
[98] J.C. Bezdek, R. Ehrlich, and W. Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.
[99] S. Robertson, E.V. Siegel, M. Miller, and S.J. Stolfo. Surveillance detection in high bandwidth environments. In *Proc. DARPA Information Survivability Conf. and Exposition, 2003.*, volume 1, pages 130–138. IEEE, 2003.
[100] H. Choi, H. Lee, and H. Kim. Fast detection and visualization of network attacks on parallel coordinates. *computers & security*, 28(5):276–288, 2009.
[101] K. Stockinger, E. Bethel, S. Campbell, E. Dart, and K. Wu. Detecting distributed scans using high-performance query-driven visualization. In *Proc. 2006 ACM/IEEE Conf. on Supercomputing*, page 82. ACM, 2006.
[102] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23):2435–2463, 1999.
[103] G. Conti and K. Abdullah. Passive visual fingerprinting of network attack tools. In *Proc. 2004 ACM workshop on Visualization and data mining for computer security*, pages 45–54. ACM, 2004.
[104] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. Internet background radiation revisited. In *Proc. 10th ACM SIGCOMM conf. on Internet measurement*, IMC '10, pages 62–74, New York, NY, USA, 2010. ACM.
[105] B. Goode. Voice over Internet protocol (VoIP). *Proc. IEEE*, 90(9):1495 – 1517, Sep 2002.
[106] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, et al. Sip: session initiation protocol. Technical report, RFC 3261, Internet Engineering Task Force, 2002.
[107] The Cooperative Association for Internet Data Analysis. The UCSD Network Telescope, 2012. http://www.caida.org/projects/network_telescope/; Last accessed: 29/10/2012.
[108] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson, et al. The internet motion sensor: A distributed blackhole monitoring system. In *Proc. 12th ISOC Symp. on Network and Distributed Systems Security (SNDSS)*, pages 167–179, 2005.
[109] Vinod Yegneswaran, Paul Barford, and Dave Plonka. On the design and use of internet sinks for network abuse monitoring. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *Recent Advances in Intrusion Detection*, volume 3224 of *Lecture Notes in Computer Science*, pages 146–165. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-30143-1_8.
[110] T. Cymru. The darknet project. *Internet: http://www. cymru. com/Darknet*, 2004.
[111] H. Sagan. *Space-filling curves*, volume 2. Springer-Verlag New York, 1994.
[112] Internet Storm Center, 2012. https://isc.sans.edu/; Last accessed: 29/10/2012.
[113] Alberto Leon-Garcia and Indra Widjaja. *Communication networks*. McGraw-Hill, Inc., 2003.



**Elias Bou-Harb** is a network security researcher pursuing his Ph.D. in Computer Science at Concordia University, Montreal, Canada. Previously, he has completed his M.A.Sc. degree in Information Systems Security at the Concordia Institute for Information Systems Engineering. He is also a member of the National Cyber Forensic and Training Alliance (NCFTA), Canada. His research interests focus on the broad area of cyber security, including operational cyber security for critical infrastructure, LTE 4G mobile network security, VoIP attacks and countermeasures and cyber scanning campaigns. He is supported by the prestigious Alexander Graham Bell Canada Graduate Scholarship (CGS) from the Natural Sciences and Engineering Research Council of Canada (NSERC).

**Mourad Debbabbi** holds Ph.D. and M.Sc. degrees in computer science from Paris-XI Orsay University, France. He has published more than 70 research papers in international journals and conferences on computer security, formal semantics, mobile and embedded platforms, Java technology security and acceleration, cryptographic protocol specification, design, and analysis, malicious code detection, programming languages, type theory, and specification and verification of safety-critical systems. He is a full professor and the director of the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada. He has served as a senior scientist at the Panasonic Information and Network Technologies Laboratory, Princeton, New Jersey; associate professor at the Computer Science Department of Laval University, Quebec, Canada; senior scientist at General Electric Research Center, New York; research associate at the Computer Science Department of Stanford University, Palo Alto, California; and permanent researcher at the Bull Corporate Research Center, Paris, France.

**Chadi Assi** received his B.Eng. degree from the Lebanese University, Beirut, Lebanon, in 1997 and his Ph.D. degree from the City University of New York (CUNY) in April 2003. He is currently a full professor with the Concordia Institute for Information Systems Engineering, Concordia University. Before joining Concordia University in August 2003 as an assistant professor, he was a visiting researcher with Nokia Research Center, Boston, Massachusetts, where he worked on quality of service in passive optical access networks. His research interests are in the areas of networks and network design and optimization. He received the prestigious Mina Rees Dissertation Award from CUNY in August 2002 for his research on wavelength-division multiplexing optical networks. He is on the Editorial Board of IEEE Communications Surveys & Tutorials, IEEE Transactions on Communications, and IEEE Transactions on Vehicular Technologies.