Beyond the Perimeter: the Need for Early Detection of Denial of Service Attacks

John Haggerty, Qi Shi, Madjid Merabti
Liverpool John Moores University, UK
cmsjhagg@livjm.ac.uk, q.shi@livjm.ac.uk, m.merabti@livjm.ac.uk

Abstract

The threat to organisations from network attacks is very real. Current countermeasures to denial of service (DoS) attacks rely on the perimeter model of network security. However, as the case study and analysis in this paper make apparent, the perimeter model, which relies on firewalls and Intrusion Detection Systems, is unable to provide an effective defence against DoS attacks. Therefore, there is a need for a new approach; one that identifies an attack beyond the perimeter. Within this paper, we present such an approach. We achieve early detection of DoS attacks by the identification of traffic signatures which indicate that an attack is underway. As these signatures can be identified 'outside' the perimeter, appropriate measures can be taken to prevent the attack from succeeding. We use examples of DoS attacks and a case study to demonstrate the applicability of our approach.

1. Introduction

Computer security has three goals; confidentiality, integrity, and availability [19]. Confidentiality requires that assets of a computer system are accessible by only those authorised to access them. Integrity is concerned with ensuring that the system's assets can only be modified by authorised parties only in prescribed ways. Availability refers to ensuring that system services and data are accessible to authorised users when needed. An attack may have an adverse effect on one, or a combination of all the three characteristics. However, there is an imbalance between real-life and research into attacks on confidentiality, integrity, and availability. The CSI/FBI survey 2001 [20] notes that 27% of respondents reported denial of service (DoS) attacks against their systems. In research, partly due to the influence of the military and their focus on secrecy, 90% of research papers deal with confidentiality, 9% with authentication, and 1% with availability [1]. Therefore, research into defences against availability attacks is not congruent with the scale of the DoS problem. However, the commercial world requires cost-effective and workable solutions to the problem.

DoS attacks prevent a legal network user from performing his/her functions [17]. They overwhelm the victim host to the point of unresponsiveness to the legitimate user of that host [5]. As demonstrated by the CSI/FBI survey [20], these attacks are prevalent 'in the wild'. With today's reliance on networks and computing technologies, these attacks can have a serious effect on the victim.

Current countermeasures to DoS rely on the perimeter model of network security. However, this model, which relies on firewalls and Intrusion Detection Systems (IDS), does not provide the defence required against DoS attacks as long as these devices are an internal part of the victim system. This is because they only *respond* to an attack, rather than *prevent* them from being successful. Consequently, when the attacks are detected the services are shut down. This is supported by our real case study presented later in this paper.

The aim of this paper is to present a new approach for early detection of DoS attacks. In this way, attacks can be prevented from succeeding and resource availability continues. In order to achieve this, our approach focuses on the communications medium beyond the organisational perimeter. All traffic, whether valid or invalid, must travel on this medium prior to reaching its destination. In the perimeter model, attack signatures are easily identifiable because they are present within a

finite, controlled and trusted boundary. However, this policy enforcement is not available beyond the perimeter. The novel contribution of this paper is that we provide an alternative approach to the perimeter model for defence against DoS attacks. To achieve this, we present new, generic signatures that can be identified regardless of traffic levels before the attack traffic reaches the perimeter of the intended victim. To demonstrate the applicability of our approach and the signatures required in early detection, we apply them to trace data from real DoS attacks and present our results.

This paper is organised as follows. In section 2, we provide an overview of DoS attacks. In section 3, we discuss the perimeter security model and its weaknesses in the face of DoS attacks. In section 4, we demonstrate these perimeter model weaknesses through a case study and the analysis of a DoS attack. In section 5, we present our approach for early detection of DoS attacks beyond the perimeter, which is based on the communications medium and has different signature requirements from the perimeter model. In section 6, we present our future work. In section 7, we make our conclusions.

2. Denial of service attacks

DoS attacks prevent a legal network user from performing his/her functions [17]. These attacks overwhelm the victim host to the point of unresponsiveness to the legitimate user of that host [5]. With today's reliance on networks and computing technologies, these attacks can have a serious effect on the victim. For example, an attack on a single host, such as a home user, may prevent a transaction from taking place. At the Local Area Network (LAN) level, an organisation may be prevented from conducting its business due to key elements of the LAN infrastructure being affected by an attack. In the extreme [24], an entire organisation may be forced to close operations.

Whilst there are a number of options open to a malicious person wishing to launch a DoS attack, there are two principal classifications of attack; *resource starvation* and *bandwidth consumption* [5, 13, 15, 26]. Resource starvation attacks [16] attempt to consume all resources on their target so that they are unable to process any new requests for legitimate users. For example, Transmission Control Protocol (TCP) SYN flooding [5] uses up all their victim's resources with half-open requests for connection. E-mail subscription attacks [2], where a user is signed up to receive a large amount of junk mail, use up the e-mail resources of the victim. Bandwidth consumption attacks are when an attacker sends more data at the victim host than it is able to deal with, filling all communications channels with data. For example, Internet Control Message Protocol (ICMP) flooding or User Datagram Protocol (UDP) flooding [5], which utilise connectionless protocols to consume bandwidth.

The situation is further complicated by the emergence of Distributed Denial of Service (DDoS) tools, such as "trinoo" [6], "Tribe Flood Network" (TFN) [7], "mstream" [8], etc. Before the emergence of DDoS tools, DoS techniques were focused on simple point-to-point attacks. However, by combining a number of attacking computers in a single attack, an attacker is able to direct more traffic at the victim. This nullifies the fact that the attacker's system was slower than that of the victim. For example, the attack on Yahoo in February 2000 [10] directed approximately 630Gb of data to the site in a 3 hour period. DDoS tools also enable the attacker to switch between attack types during an attack. For example, TFN [7] is capable of ICMP flood, TCP SYN flood, UDP flood, and "Smurf" style attacks.

The effectiveness of DoS attacks has been much reported in recent years, even though organisations continue to employ perimeter model security devices. Cases such as the Cloud Nine incident [24] and the case study to be presented in this paper demonstrate that despite the protection afforded by these devices to technological and information resources, DoS attacks can still occur. Countermeasures supporting perimeter model devices include techniques such as ingress/egress filtering at the ISP level to prevent attack packets from leaving the attacker's network in the first place. The number of attacks observed in [16] demonstrate that despite the implementation of this support filtering, a large number of attacks are still able to cause damage to their victim. Protection of all organisational network assets, including perimeter model devices, is required to combat the DoS problem.

3. The perimeter model and DoS

The perimeter model is an architecture commonly used by today's organisations to protect critical infrastructures. This security model divides network architectures into two distinct groups; *trusted* and *untrusted*. The trusted group is often the finite internal infrastructure, whilst the untrusted consists of infinite external networks. To maintain this segregation of trust and to detect transgression, two types of devices are commonly used; firewalls to control traffic entering and leaving the trusted domain, and IDS to detect transgression of trust within the trusted area boundary.

Firewalls and IDS have distinct but complimentary roles in their protection of network computing resources. Firewalls [3] implement access control and audit functions at the interface between two or more networks, often with different security levels. In effect, they are a conduit that network traffic passes through, both into and out of the network perimeter, and where the security

policies for the organisation pertaining to network traffic are enforced. Firewalls come in varying levels of sophistication [19], from packet filters to bastion hosts. However, network-level firewalls have in common that access to hosts, networks or services is usually controlled using rules based on IP addresses, ports, IP flags, and network interfaces [3].

Intrusion detection [21] is the art of detecting and responding to computer misuse. It attempts to deal with the problem of identifying individuals who are using a computer system without authorisation or those that have legitimate access to the system but are misusing their privileges [25]. In reality, intrusion detection is often satisfied with identifying hosts of attacks rather than human perpetrators behind the attacks. This is due to the latter involving the co-operation of law enforcement agencies. Unlike firewalls, which enforce security policies, IDS detect violations of the security policies within the trusted domain. There are two main types of IDS: *host-based* IDS and *network-based* IDS. A host-based IDS detects attacks by watching for suspicious activity on a single computer system. Network-based IDS [22] are driven by interpretation of raw network traffic. They attempt to detect attacks by watching for suspicious patterns in network traffic within a defined perimeter.

DoS attacks are a potent weapon in an attacker's armoury against perimeter model devices for two reasons. First, in a primary attack, the objective of the attack may be to degrade or halt services of the perimeter model device itself. For example, if the firewall is unable to respond [11], it may degrade or halt Internet access for all its users. Second, they can be used as a diversionary tactic to mask another attack. Normally, attacks are characterised by some noise or other indication of an intruder attempting to compromise the target host or system [14]. Using a DoS attack as camouflage, the real attack is less likely to be observed. Thus, whilst security analysts attempt to stem the flood of packets being sent to the target in the DoS attack, particular packets can be hidden in the flood to break into the target system.

For either reason, DoS attacks are an effective attacker strategy in reaching their attack objective. For example, for either primary or diversionary attacks to work, the attacker must choose a tool to flood the target perimeter model device with as many packets as possible. The aim is not to fill the bandwidth with attack packets and deny legitimate connections, but to make the security device log all attack packets. If the security device logs these packets as suspicious, the attacker can cause the device to fill all hard disk space with audit information. A DDoS tool, "Stick" [12], has already been seen in the wild that specifically targets IDS by sending spurious packets to fill up event logs.

4. DoS - a case study

Recently, we conducted research into the internal threat to a large network [for further details, see 11]. During the research period, a number of security incidents came to light and were captured for analysis. This included a real major DoS incident caused by worm infection. This case study provides us with the opportunity of analysing a real network security event within a defined perimeter to identify issues requiring redress. First, we are able to see the effect that a DoS attack has on perimeter model security devices. Second, it highlights the problem that whilst the perimeter model architecture is in use, an adequate defence against DoS is not possible solely with these perimeter devices.

In order to ascertain the internal threat to the network, an IDS was used on a control machine. BlackICE Defender was chosen as the IDS to be used on the control machine for the duration of the research for two reasons. First, an IDS had to be chosen that ran on Microsoft OSs as the network contained no UNIX machines. Second, the machine had to look like a users' machine and not stand out to the casual observer. Therefore, software that a user may use, rather than an administrator, was seen as more appropriate.

The root of all intrusion detection is based on analysing a set of discrete, time sequenced events for patterns of misuse [21]. These events then form Events of Interest (EOI) [18] within the system. These EOI are possible attacks against the target machine or network. One issue identified within our research was the number of positives versus false positives that were recorded by the system. A *positive* is when the recorded attack equates to an actual EOI, whereas a *false positive* is when an event is recorded as an attack but is not. Over a four-week period, a total of 1493 attacks were recorded against the control computer by the IDS. Of these, 409 attacks were positives and 1084 attacks were false positives.

The 409 positives were caused by a large number of Hyper Text Transfer Protocol (HTTP) port scans caused by an infection within the network perimeter by the Code Red worm. The Code Red worm [4] is a self-propagating malicious code that exploits vulnerabilities in Microsoft Web servers. An upsurge in internal network traffic illustrated by the bottom line in figure 1 is accounted for by the worm attempting to infect other servers by sending a crafted HTTP GET request. This exploits a buffer overflow vulnerability in computers across the network. During the research period, two infections were detected by the upsurge in network traffic activity within the network.

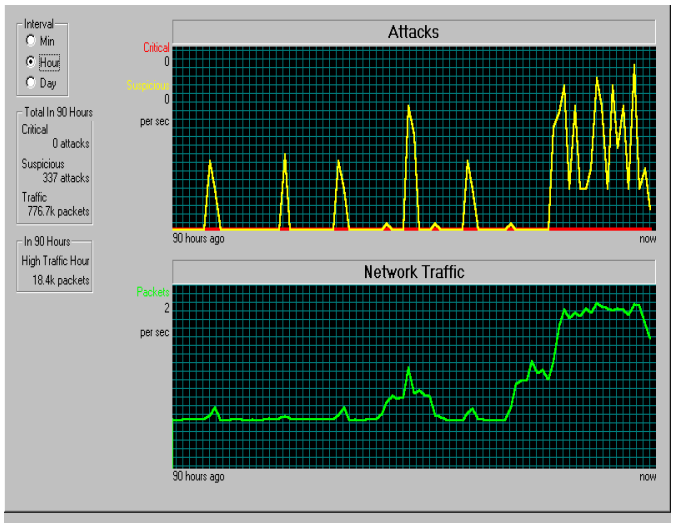

Figure 1. Recorded network traffic by the IDS during the first period of infection by the Code Red Worm. The top line shows the number of attacks against the system, with a proliferation of attacks as the worm is launched.

The first infection by the worm had little or no effect on the principal perimeter model device of the network; the firewall. During the first period of infection, 8 infected machines were identified and patches were then applied by system administrators to remove the infection in accordance with security alerts pertaining to the problem. In total, 219 Code Red HTTP GET requests were received and logged by the control machine IDS. With approximately 330 machines on this network, we can see that a large amount of traffic was generated by requests made to each and every machine. However, services on the network were not seen to be greatly degraded.

A second period of infection by the worm occurred two weeks after the first. A machine that had been infected during the first period of infection had been identified, but patched unsuccessfully. When the Code Red worm was triggered, it attempted to infect other machines both within and beyond the network perimeter. In total, 190 Code Red HTTP connection requests originating from 4 machines were observed against the control machine IDS. This was significantly fewer machines due to earlier successful patching of the system, and resulted in less infection traffic being generated. The infected machines were quickly identified by network traffic monitoring and again taken off-line until patches could be applied.

This second infection resulted in severe internal network disruption. Whilst the attack traffic levels observed were less than the first infection, the entire network and associated sub-nets ground to a halt. As the original infected machine attempted to connect to the Internet, all attack traffic was routed to the firewall as the main perimeter conduit to external networks. The firewall logged all attempts by the attack traffic to connect to the external address as suspicious and did not forward the traffic. However, the number of suspicious requests for external connection caused the firewall's hard disks to fill with audit information, until the point at which the entire hard disk became full and the firewall crashed. This ensured that no traffic, whether valid or invalid requests, could pass through it in either direction. The firewall was unable to provide external users access to key internal servers, such as the main organisational Web server or e-mail servers. Internal users were also unable to connect to external services. This lasted for a further 5 days due to a lack of a timely back-up plan and the resulting system reconfiguration.

5. Early detection of DoS attacks beyond the perimeter

As discussed previously, within the perimeter model, firewalls enforce network security policies and IDS detect misuses of the system. However, both types of security device have distinct disadvantages in the face of a DoS attack because they are located on the target system or network. First, all traffic is required to pass in either direction through the firewall for the security policies to be enforced. When faced with a large amount of traffic, a bottleneck can be formed. This bottleneck then prevents legitimate traffic from passing through the firewall. Second, whilst DoS is a misuse of systems, it will already have achieved its objective of flooding the target system with packets by the time it has been detected.

Analysis of the case study demonstrates that there is a very real requirement for an approach that detects and responds to DoS attacks prior to their reaching the target's perimeter. In this way, we can prevent the attack from succeeding. In order to achieve this, we must focus on the communications medium beyond the perimeter of the trusted networks. However, an advantage of the perimeter model architecture is that we are able to clearly define normal versus abnormal traffic in the organisational security policies. In this way, control over behaviour within the trusted domain is maintained and monitored. We are not afforded this distinction in early detection as we have no such control over behaviour. Therefore, we need to provide a definition of DoS that facilitates a distinction between normal and abnormal traffic beyond the perimeter.

The classifications of DoS attacks presented in section 2 have in common the issue that they are descriptive rather than prescriptive in their approach to

the DoS problem. Whilst they go some way in providing a *description* of the attack objective, they are vague terms when looking for *solutions* to the problem. They provide neither the power nor flexibility with which to achieve effective detection in the communications medium beyond the perimeter as they do not facilitate the distinction between normal traffic and abnormal traffic [14] required for early detection. To redress this, we define a DoS attack as:

***x* distinct packets matching *s* signature in *y* seconds to *h* host**

Figure 2 demonstrates the relationship between elements of the DoS definition for early detection. This diagram provides us with an intuitive illustration of the relationship between the *x*, *y*, and *s* on the target host *h*, when using a single signature. The precise relationship between *x*, *y*, and *s*, and in particular the precise value of *s*, requires further study. Host *h* refers to the target of the DoS attack. Packets *x* are the total number of packets directed at *h*. This is the total of all packets, so includes both normal and abnormal packets. The DoS attack packets *s* are those packets that match a particular signature of this type of attack. These signatures are different from those used in perimeter model devices as we no longer have the clear-cut distinctions that an organisational policy allows. Time *y* is the total period that packets are directed to *h*.

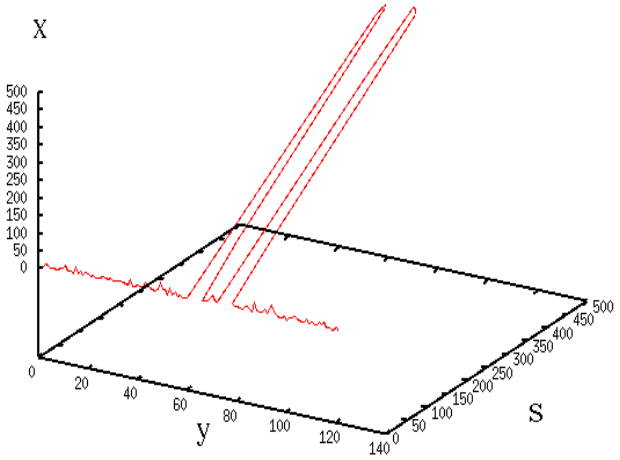

Figure 2. Illustration of the relationship between elements in the DoS definition for a single attack on a particular host.

The advantages of using this definition of DoS attacks are twofold. First, we do not focus on the effect of the attack, but look at the constituent elements of the attack itself. We are therefore able to focus on the communications-level behaviour of DoS attacks for early detection purposes. We can see that key to detection is that packets which match a signature can be defined as DoS attack packets. Second, figure 2 demonstrates the main problem caused by these attacks to perimeter model devices such as firewalls and IDS. Over a period of time, the DoS attack sends more packets to the target. As perimeter model devices log all packets contravening trusted domain policies as intrusion attempts or attacks for audit purposes, valuable processing time and system resources are taken up. Thus, the longer the duration of the attack, the more signatures are recorded, which take up even more system resources on the security device. This may then lead to the unavailability of these devices.

Within a DoS attack, traffic directed at the victim can build up very quickly and be sustained for a period of time. This volume of traffic is the means by which the attacker wishes to cause the target system to become unavailable. We can see the impact of DoS attack traffic on the network within a 2-dimensional view of network activity in figure 3 below. This 2-dimensional view observes traffic in terms of time and number of packets and shows a UDP attack from a single attacker on a small traffic volume LAN for a total duration of two minutes. The sharp increase of traffic at 360 seconds indicates the start of the attack and within a matter of seconds, the attack reaches a plateau of approximately 1300 packets per 5 seconds. This volume of traffic continues for the duration of the attack. In total, over 30,000 attack packets are sent to the target. The yield of the attack will be increased if multiple attacking machines are used. However, this 2-dimensional view only allows us to see a build up of traffic or traffic by protocol. It is therefore insufficient to satisfy us that an attack is underway rather than a “flash crowd” as there is no facility to distinguish normal from abnormal traffic.

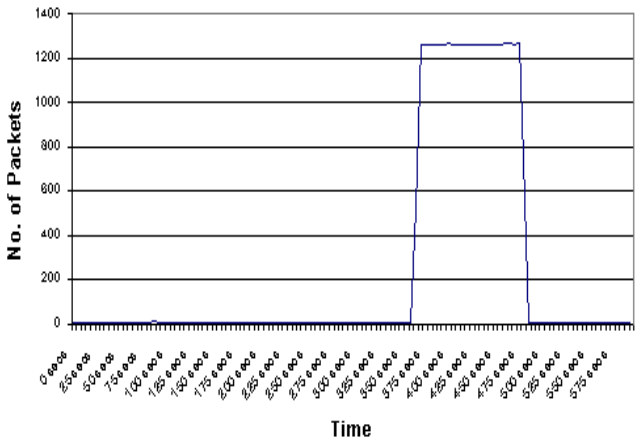

Figure 3. 2-dimensional view of network traffic during a UDP flood attack showing number of packets on the network/5 seconds

Figure 3 represents an important distinction between DoS attacks and other attacks, for example, those that attempt to execute malicious code on their victim, which can be used for early detection. Often (but not always), a DoS attack requires a large amount of data to be sent to the victim. It is the mass of all packets directed at a victim that poses the threat, rather than the packets themselves [9]. For example, it takes 500 packets/second to successfully bring down an Internet server during a SYN flood [27]. Therefore, the focus of our detection shifts from one based on content to one based on volume of packets with corresponding features.

An approach that sifts information in potentially high volumes of traffic can require a great amount of processing power. In work on detecting backdoors [28], it was observed that the more traffic that can be discarded based on information in the TCP/IP headers, the better, as this can reduce the processing load on the monitor. Therefore, our approach looks at the content of packet headers, and the information that they yield, to provide our signatures.

Three characteristics that are present in packet headers aid us in the signature process. First, there is the victim, represented by the destination IP address. Whilst the source IP address of an attack packet may be forged, such as is required by TCP SYN flooding, the destination address cannot. Within the perimeter model, the destination address is not a factor as the attack has already reached its destination by the time the attack is detected. However, for an approach that focuses on traffic *en route*, this is an important factor as it allows us to identify the target of an attack. Second, there is a rate of transfer of particular data types. For example, attacks that subvert UDP or ICMP require a large amount of data to be sent to the victim to consume available bandwidth and prevent legitimate users from accessing resources [5]. Third, related to rate of transfer, we have a characteristic of time. A DoS attack differs from other types of attack as a DoS attack is more successful the longer it affects the victim. Therefore, we will see a high rate of data transfer over a period of time.

When we apply our approach to the same traffic as seen in figure 3, and therefore add the third dimension of our early detection signatures to the traffic view, we can see that the build up of traffic is in effect an attack. As discussed above, the signatures used in early detection will differ from those used in the perimeter model; they are more generic and combine various factors. A combination of features in an attack prove to have a greater distinctive power than any one feature by itself [28]. In figure 4 below, we have the characteristics discussed above; a victim, a high rate of data transfer, and the traffic is sustained at a high level for a period of time. *S* relates to a number of features determining an attack rather than just one single observed signature. Within this attack, in addition to the other characteristics mentioned earlier, we see a fixed packet size, single source IP address, and all traffic sent to a single destination port, that combined form our signature. This port did not relate to a service on the victim. Therefore, the victim machine responded to the UDP attack packets with ICMP messages to inform the attacking machine that the requested service was not available. Although not shown, we observed a large amount of ICMP traffic coming from the victim and directed at the attacker. As we can see in the figure below, these signatures were observed from the start of the attack at 360 seconds, until the attack was terminated at 500 seconds. We can see that the virtually all of the traffic traversing the network (represented by the *x* axis) matches our signature (represented by the *s* axis).

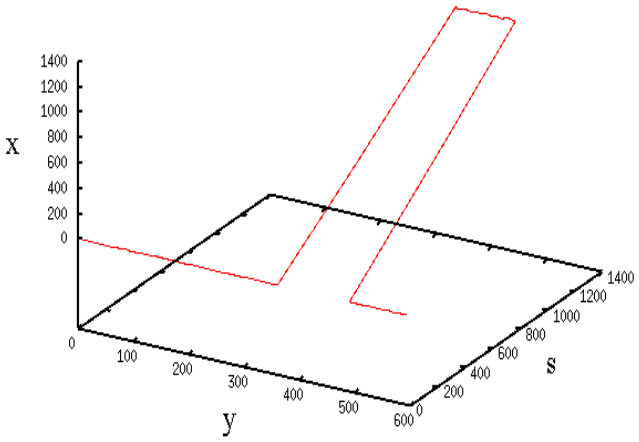

Figure 4. View of network traffic during a UDP flood showing number of packets on the network/5 seconds. By applying the signature axis, we can distinguish normal from abnormal traffic.

One problem faced is that a high rate of data transfer in a short period of time, targeted at a particular host, can occur in legitimate circumstances. For example, in occurrences of "flash crowds" such as [23], a large amount of users attempt to connect to a popular Web server. This large amount of connection attempts is difficult to distinguish from a TCP SYN flood as both phenonema have characteristics in common; sudden volume increase of TCP traffic, large amount of connecting source IP addresses, and sustained time period. However, there is a difference between the two. As discussed earlier, a TCP SYN flood requires that spoofed IP addresses are used to tie up resources on the vicitim by holding half-open connections awaiting a non-existent reply. In a "flash crowd" the demand is so great that not all users wishing to establish a connection can do so. However, some connections are still being established.

In figure 5, we see an example of a TCP SYN flood attack. The graph shows two attacks, both lasting 20 seconds each, the first launched at 300 seconds and the second at 360 seconds. In both attacks, we see unusual sudden spikes caused by TCP traffic directed at a particular host on the network. In both attacks, there are approximately 1000 individual IP addresses attempting to connect with the victim within each 20 second attack period. The rise in the packets on the network is caused by traffic that matches our signature: SYN packets that belong to the first part of the TCP 3-way handshake with no resulting connections being established. In this case, we observe that the spikes are caused by SYN packets being sent to the victim, represented by the number of packets on the *s* axis. In the case of a "flash crowd", connections are established, but not enough to satisfy demand. Therefore, we would expect to see a more gradual increase in TCP traffic rather than a sudden co-ordinated traffic spike. We would also expect a more gradual decrease in TCP traffic as connections are made and then ended. In addition, within this particular attack, we are aided by a flaw in the attacker's program that verifies that this is in fact an attack. The sequence number used in the initial SYN packet was constant in all attack packets; 674719801. It is highly unlikely that 2000 individual hosts would have identical sequence numbers.

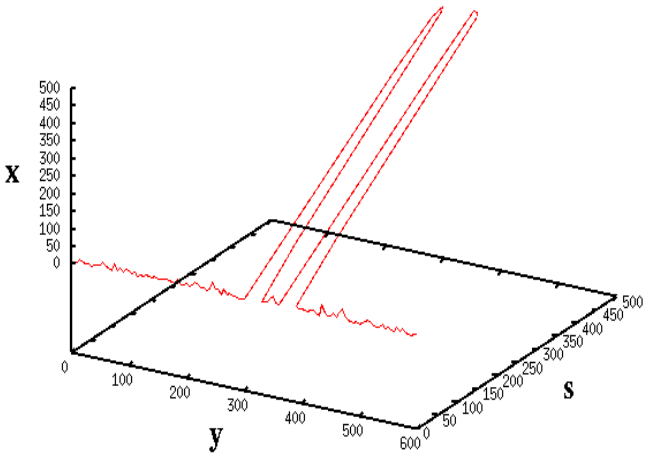

Figure 5. View of network traffic during a TCP SYN flood attack By applying the signature axis to determine that connections are not established we can distinguish attack traffic from "flash crowd" traffic.

The question remains of how we can detect an attack of only a small number of packets, or where the total traffic throughput outweighs the DoS attack traffic. For example, the DoS traffic traversing the Internet may not travel by the same route as other packets of the attack. It is only when they reach the final destination that we may see the upsurge in attack traffic as shown in figure 3. To test our approach's ability to detect abnormal traffic in such situations, we ran tests with a DoS attack, *nuke*, that only sends a small amount of traffic. This traffic causes conflicts in a vulnerable target's kernel running earlier versions of Windows, resulting in the machine crashing. In figure 6 below, we see a 2-dimensional view of network activity over a ten-minute period. Two attacks were launched at 300 seconds and 360 seconds.

As we can see, the 2-dimensional view does not allow us to distinguish the attack traffic from other traffic on the network. Even when viewed by protocol, we can still not clearly differentiate normal from abnormal traffic.

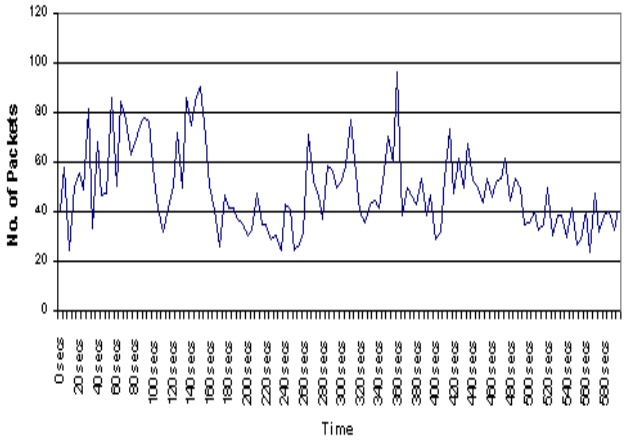

Figure 6. View of network traffic during *nuke* attack. Attacks are indistinguishable and were launched at 300 and 360 seconds.

When we apply our early detection approach to the same data as used in the 2-dimensional view of figure 6, we are able to distinguish attack traffic from normal traffic. In figure 7 below, despite only 9 packets being sent in the attack, we can see that the attack is detected by adding the *s* axis. Due to the small number of TCP packets sent, we are not afforded the number of factors observed in the UDP attack in figure 4 above. However, the main signature of this attack is that traffic is targeted at port 139. As this port is used for internal networking within Microsoft networks it is highly unlikely that external traffic would connect to this port unless for malicious intent. Therefore, we use this port use as an indication of an attack.

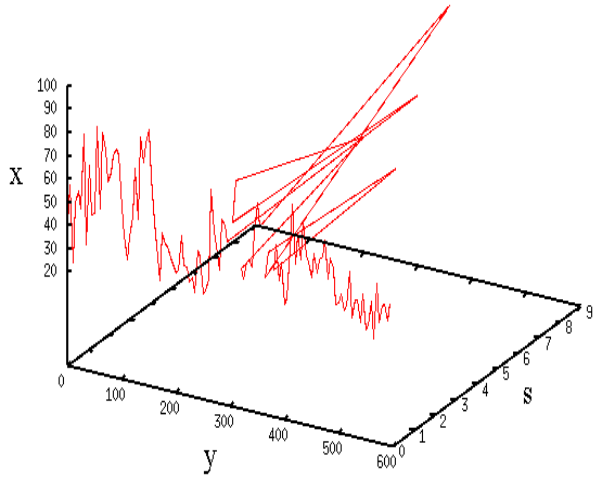

Figure 7. View of network activity during *nuke* attack. The 3-dimensional spikes allow us to determine attack traffic.

The above examples have demonstrated that using our 3-dimensional approach allows us to detect an attack that cannot be seen in a 2-dimensional approach.

6. Future work

The perimeter model provides effective defences against a number of security problems. However, whilst the security devices are located on the victim system, they do not provide an adequate solution against DoS attacks. An alternative approach is to detect and respond to DoS attacks in the communications medium before they are able to reach their intended target. In this way, we are able to provide an effective defence against DoS attacks.

The controlled and finite architecture of the perimeter model allow signatures to be clearly defined. However, with perimeter model detection of DoS attacks, we face a dichotomy. On the one hand, we have definite control over the signatures that we detect, and on the other, by the the time we have detected the attack, it has already achieved its goal. The clear distinction afforded by security policies is not available beyond the perimeter. Therefore, signatures that are applicable to this external environment are required. Our research has already identified a number of generic patterns, or signatures, of DoS attacks that are observed in the communications medium. These patterns are a blending of a number of factors, rather than one definitive signature. Combined, they provide the distinction between normal and abnormal traffic. The more quantitative relationships between pattern constituents will require further

research.

We are currently working on an implementation of our approach. The architecture required for our approach does not attempt to replace the perimeter model, but provides the effective defence against DoS attacks that is required. Our approach must address a number of issues to provide the protection needed. First, as the case study presented in this paper demonstrates, attack detection must balance reports of positives and false positives to be effective. Second, related to the first issue, the approach must provide a response mechanism that does not adversely affect legitimate users. Third, as we are providing early detection, the approach must be scaleable. Fourth, we must consider the administration of our early detection approach. Whilst detection beyond the perimeter has distinct advantages in the face of DoS attacks, it lacks one advantage of the perimeter model: a central control authority. Therefore, issues such as control and administration of the defence mechanisms, responsibility for signature updates when required, and responsibility for policy management require further work. These issues form the basis of our future research work.

7. Conclusions

The threat to organisations from network attacks is very real. Yet, DoS attacks are seriously under-represented in current research. There is a very real need for organisations to protect their technological and information resources from these attacks. Current defences rely on the perimeter security model, consisting of devices such as firewalls and IDS. This model is a commonly used architecture to protect critical infrastructures and relies on the separation of networks into two distinct groups; 'finite and trusted' and 'infinite and untrusted'. However, as the case study and analysis in this paper make apparent, the perimeter model is unable to provide an effective defence against DoS attacks, whilst these security devices are located on the target system.

Therefore, there is a need for a new approach to provide an effective defence against DoS. Within this paper, we have presented a new approach which focuses on detecting DoS beyond the perimeter. This approach detects attacks on the communication medium that all traffic, whether valid or invalid, must traverse to reach the intended destination. Current approaches to DoS provide neither the power nor flexibility required for early detection. We have presented a means by which this early detection of attacks can be achieved. Attack signatures used by the perimeter model cannot be applied to an approach that does not rely on this model. Therefore, we have presented signatures required for early detection. To demonstrate the applicability of our approach, we have used examples of DoS attacks and a case study. Our future work will concentrate on the issues raised in section 6 to develop a cost-effective defence against DoS attacks.

References

[1] Anderson, R. & Lee, J. H., "Jikzi - a new framework for security policy, trusted publishing and electronic commerce," *Computer Communications,* vol. 23, pp. 1621-1626, 2000.

[2] Bass, T., Freyre, A., Gruber, D. & Watt, G., "E-Mail Bombs and Countermeasures: Cyber Attacks and Brand Integrity," *IEEE Network,* vol. 12, pp. 10-17, 1998.

[3] Benecke, C., "A Parallel Packet Screen for High Speed Networks," *Proceedings of the Annual Computer Security Applications Conference* , Phoenix, Arizona, 1999.

[4] CERT, "CERT Advisory CA-2001-19 'Code Red Worm Exploiting Buffer Overflow in IIS Indexing Service DLL," CERT Advisory, http://www.cert.org/advisories/CA-2001-19.html, download 2001, 2001.

[5] Dietrich, S., Long, N. & Dittrich, D., "Analyzing Distributed Denial of Service Tools: The Shaft Case," *14th Systems Administration Conference (LISA 2000),* New Orleans, Louisiana, 2000.

[6] Dittrich, D., "The DoS Project's "trinoo" Distributed Denial of Service Attack Tool," University of Washington Technical Report, http://staff.washington.edu/dittrich/misc/trinoo.analysis, download 2000, 1999.

[7] Dittrich, D., "The "Tribe Flood Network" Distributed Denial of Service Attack Tool," University of Washington Technical Report, http://staff.washington.edu/dittrich/misc/tfn.analysis, download 2000, 1999.

[8] Dittrich, D., Weaver, G., Dietrich, S. & Long, N., "The "mstream" Distributed Denial of Service Attack Tool," University of Washington Technical Report, http://staff.washington.edu/dittrich/miscmstream.analysis.txt, download 2000, 2000.

[9] Gil, T. M. & Poletto, M., “MULTOPS: a Data-Structure for Bandwidth Attack Detection”, *Proceedings of the 10th USENIX Security Symposium* , Washington DC USA, 13-17 August, 2001.

[10] Gonzalez, G., "Congressional Statement on Cybercrime by the Federal Bureau of Investigation," FBI Congressional Report, http://www.fbi.gov/pressrm/congress/congress00/gonza042100.htm, download 2000, 2000.

[11] Haggerty, J., Shi, Q. & Merabti, M., "The Threat from Within: An Analysis of Attacks on an Internal Network", in Ghonaimy, M. A., El-Hadidi, M. & Aslan, H. K. (eds.) *Security in the Information Society Visions and Perspectives* , Kluwer Academic Publishers/IFIP, pp. 133-145, 2002.

[12] Howard, S., "Stick and Network Signature Based Intrusion Detection", SANS Institute Info Sy Reading Room Technical Report, http://www.sans.org/infosecFAQ/threats/stick.htm, download 2001, 11 April 2001.

[13] Leiwo, J.& Zheng, Y., "A Method to Implement a Denial of Service Protection Base," *Vol. 1270 of LNCS,* Information Security and Privacy, Berlin Germany, pp. 90-101, 1997.

[14] Mansfield, G., Ohta, K., Takei, Y., Kato, N., & Nemoto, Y., "Towards trapping wily intruders in the large," *Computer Networks* , vol. 34, pp. 659-670, 2000.

[15] Martin, B., "Have Script, Will Destroy (Lessons in DoS)," Technical Report, http://www.hackernews.com/bufferoverflow/00/dosattack/dosattack.html, download 2000, 2000.

[16] Moore, D., Voelker, G. M. & Savage, S., "Inferring Internet Denial-of-Service Activity," *Proceedings of the 10th Usenix Security Symposium* , Washington DC, USA, 13-17 August 2001.

[17] Muftic, S., Patel, A., Sanders, P., Colon, R., Heijnsdijk, J. & Pulkkinen, U., *Security Architecture in Open Distributed Systems,* John Wiley & Sons, Bath, UK, 1993.

[18] Northcutt, S., *Network Intrusion An Analyst's Handbook* , New Rider Publishing, USA, 1999.

[19] Pfleeger, C., *Security in Computing* , 2nd ed, Prentice Hall International, USA, 1997.

[20] Power, R., "2001 CSI/FBI Computer Crime and Security Survey", Computer Security Institute/Federal Bureau of Investigation Technical Report, vol. 7, no. 1, Spring 2001.

[21] Proctor, P. E., *The Practical Intrusion Detection Handbook,* Prentice Hall, Upper Saddle River, NJ, 2001.

[22] Ptacek, T. H. & Newsham, T. N., "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," Secure Networks Inc. Technical Report, http://www.clark.net/~roesch/idspaper.html, download 2001, January, 1998.

[23] Richardson, T., "Brits flock to the 1901 census site", *The Register* , 2 January 2002, http://www.theregister.co.uk/content/archive/23523.html

[24] Richardson, T., "Cloud Nine blown away, blames hack attack", *The Register,* 22 January 2002, *http://www.theregister.co.uk/content/archive/23770.html*, download 2002

[25] Spafford, E. H. & Zamboni, D., "Intrusion detection using autonomous agents," *Computer Networks,* vol. 34, pp. 547-570, 2000.

[26] Todd, B., "Distributed Denial of Service Attacks," OVEN Digital Technical Report, http://fridge.oven.com/~bet/DDoS/whitepaper.html, download 2000, 2000.

[27] Wang, H., Zhang, D. & Shin, K. G. “Detecting SYN Flooding Attacks”, *Proceedings of INFOCOM2002* , New York USA, 23-27 June 2002.

[28] Zhang, Y. & Paxson, V. “Detecting Backdoors”, *Proceedings of the 9th USENIX Security Symposium* , Denver Colorado USA, August 14-17, 2000.