An active router approach to defeating denial of service attacks in networks

Journal:	*IEE Proc. Communications*
Manuscript ID:	COM-2005-0441.R1
Manuscript Type:	Research Paper
Keyword:	ACTIVE NETWORKS, PROTOCOLS, Denial of service attacks, SYN, SMURF

powered by ScholarOne
Manuscript Central™

An active router approach to defeating denial of service attacks in networks

Dr Fadi Ali El-Moussa[1], Professor Nigel Linge[2], and Dr Martin Hope[3]

Centre for Networking and Telecommunications Research
Informatics Research Institute,
University of Salford,
Salford,
Greater Manchester, M5 4WT
UK

1. fadi_elmoussa@hotmail.com[i]
2. n.linge@salford.ac.uk
3. m.d.hope@salford.ac.uk

Abstract

Denial of Service Attacks represent a major threat to modern organisations who are increasing dependent upon the integrity of their computer networks. This paper presents a new approach to combating such threats by introducing active routers into the network architecture. These active routers offer the combined benefits of intrusion detection, firewall functionality and data encryption and work collaboratively to provide a distributed defence mechanism. The paper provides a detailed description of the design and operation of the algorithms used by the active routers and demonstrates how this approach is able to defeat a SYN and SMURF attack. Other approaches to network design, such as the introduction of a firewall and intrusion detection systems, can be used to protect networks however, weaknesses remain. Within the paper it is proposed that the adoption of an active router based approach to protecting networks overcomes many of these weaknesses and therefore offers enhanced protection.

Key-words

Denial of Service Attacks, Active networks, SYN, SMURF

[i] Dr Fadi Ali El-Moussa is now employed by BT

1. Introduction

Most organisations are today totally dependent upon the performance and integrity of their network infrastructures. Coupled to this growth in dependency, these organisations have seen a significant increase in the threats posed to such networks, resulting in incidents of serious attack and the ensuing business impacts. Whereas the first generation of attackers required a thorough knowledge of computing, communication protocols and networks, the widespread availability of free web-based attacking tools has brought about a new generation that requires far less detailed knowledge. Consequently since 1988 there has been a 3,000% increase in the number of serious incidents of attack reported by CERT [1].

Denial of Service attacks [2] aim to disrupt a legitimate user's access to the network, servers, or applications by either targeting a server directly or by consuming network bandwidth. A SYN attack [3] aims to target a server's resources by preventing new TCP connections being accepted by repeatedly failing to complete the TCP connection three-way handshake protocol through the withholding of the final SYN ACK PDU. Network resources can also be targeted by the SMURF [4] attack, which is based on the Internet Control Message Protocol (ICMP) where a sequence of ICMP Echo Request PDUs are issued with a fictitious or spoofed IP source addresses. Each host receiving one of these PDUs is required to respond with an ICMP Echo Reply PDU hence, generating excessive traffic that can ultimately overload network resources and greatly impair the throughput available to legitimate network traffic.

Whilst solutions to these attacks have been proposed and implemented, this paper reports on a new approach [5] that creates a network design in which defence intelligence is distributed across a set of active routers that work collectively to offer an integrated approach to the defeat of typical denial of service attacks. We believe that this integrated approach provides a more robust defence against well known attacks and overcomes limitations inherent within current solutions.

2. Defeating Denial of Service Attacks

A significant amount of work has been undertaken to develop techniques for defeating denial of service attacks. The traditional first line of defence is the firewall [6]. However, firewalls do suffer from a number of limitations, which include the inability to detect an attack arising from a host 'behind' the firewall. Once a firewall has been compromised then the whole network becomes vulnerable and because a firewall blocks an attack at the victim side, then a firewall cannot prevent an attacker from initiating further attacks.

Techniques such as traffic rate limiting and trace back are effective at combating denial of service attacks, but do require time to observe network traffic profiles in order to deduce that an attack is in progress. During this time network users must suffer some loss of service or degradation in performance. Equally, when the attacker uses a fictitious IP address it takes longer to trace such attacks back to their point of origin.

Active networks allow the network itself to execute program code up to the Application layer on packet contents. This can be achieved by embedding code within packets which is then executed by routers, or by providing a pointer within a packet to specific executable code within a router. Using this processing capability, active network technology is now being exploited to defeat denial of service attacks through the deployment of active routers within a network infrastructure.

The Intrusion Blocker based on Active Networks (IBAN) [7], transports active code between active nodes so that a scanner function can analyse hosts for vulnerabilities. Based on this analysis, a blocker function is then downloaded to an active router to detect and block known attack profiles. Active Shaping [8] deploys an active component comprising a probe active component (PAC) and a traffic control active component (TAC) to all active routers. Routers maintain three types of queue per interface: normal queue, suspicious queue, and malicious queue. The PAC monitors traffic and if within acceptable traffic flows, passes it to the normal queue. However, if an identifiable traffic flow exceeds some traffic threshold value, then it is regarded as suspicious, transferred to the suspicious queue, and an attack profile configured. The TAC then rate controls this suspicious traffic through a configurable filter function. When the attack is confirmed, traffic is then passed to the malicious queue from which a backtrack message is sent to the up stream router to also apply rate limiting on the flow. This backtrack message will propagate towards the source of the attack. The Active Security System (ASSYST) [9] allows routers to interact through an Active Security Protocol (ASP) to isolate the source of the denial of service attack. The ASP provides a set of messages that propagate between routers to enable them to identify the malicious traffic and adopt a defence strategy. A series of alert messages allow routers to determine the path of the attack and then to employ traffic shaping to limit the suspicious traffic flow.

There are several limitations with current active network solutions namely that there is often an assumption that all routers within a network are active; that the time taken for active code to propagate through the network can be excessive; that large volumes of signalling messages are generated, and there is a limited range of denial of service attack profiles that can be adequately combated.

3. Distributing Defence Intelligence

This paper presents a new approach to the adoption of active networking for the defeat of denial of service attacks. A novel active-router based architecture is presented that offers the integration of a cryptographic algorithm, firewall functionality, and intrusion detection into a series of distributed active routers that communicate using an active protocol. The distributed architecture offers many advantages in defending against well-known types of denial of service attack. The rationale behind the integrated approach is based on the fact that each of these components has limitations but, when combined, these limitations can be overcome by another component function.

For example, a firewall cannot detect attacks (such as SMURF attack) that are embedded within ‘normal’ traffic. Intrusion detection can however, detect this type of attack. Equally, distributing firewall functionality and intrusion detection, allows for the detection and blocking of an attack at different points in the network to overcome some of the traditional firewall limitations. A distributed architecture ensures that the network is no longer a single point of failure, so that an attacker has to pass through different layers of protection before they reach the victim. Distributing a cryptographic algorithm among routers in the network allows them to authenticate packets coming either from the end user or from other routers. This also protects the network against packet sniffing, because all packets traversing through the network are encrypted. In other words, a router is working on behalf of the clients to protect their transmission data depending on a pre-defined security policy inside the router.

4. The Active Router Architecture

Active routers are distributed throughout a network architecture, however, there is no requirement for all routers to become active. These active routers communicate using a dedicated active protocol that has its own defined packet formats, that include active packets, active packet deny, and active packet acknowledgments. In contrast, passive packets are packets issued by non active nodes and in the context of this work are considered to be IP datagrams. Given the fact that not all routers are active, an active router discovery protocol is also defined to allow active routers to determine the address and location of all other active routers within the same overall network architecture. This involves the generation of an active packet that it broadcast over each subnet to which the active router is attached. The generation of these packets is determined by a timer that is configured on the one hand, to minimise excess traffic on the network and on the other, to allow the network to dynamically adapt to the introduction and removal of active routers. These packet also contain a security code that allows active routers to validate each other and more importantly, to detect if an active router has been compromised.

When the active routers are initialised they can commence normal packet processing. In order to pass traffic between active routers, dedicated TCP connections are opened between pairs of active routers to not only ensure that network traffic is protected from packet sniffing attacks, but to also overcome the backtracking message limitation [8]. In this way, active routers do not need to send traffic filters from one hop to another, in order to block the attack. Instead, the active routers can immediately send a block command in an active packet to the other end router with the relevant attack signature. This mechanism can protect the victim behind the active router even if the attacker generates a random spoofed IP source address. Like most existing active network schemes our approach also uses attack pattern recognition and rate limiting to detect and control the attack. Therefore, if an active router detects an attack attempt it will send an active packet to the other edge active router to block the attacking packets. Unlike some other approaches [10,11,12,13,14], this architecture allows the network to work without relying on a central management server.

When an active router receives a passive packet it will process it through an intrusion detection process to check if the packet possesses any attack signatures. If the packet passes that test, then it will be compared against the firewall security policy to determine if this packet complies with the network security policy. Finally, the active router will apply a cryptographic algorithm that is used to specify the type of encryption and authentication that will be applied to the current connection and used to secure data transfer between active routers.

The passive packet process determines how the active router will handle the received passive packet as shown in figure 1. Thus, if the received passive packet is destined to a directly connected subnet, then the passive packet process will send that packet through the SYN and SMURF attack defending processes. In order for the active router to detect a SYN or SMURF attack it applies a rate limit on received connection requests and ICMP Ping packets. If an active router detects an attack it will drop the received packets and send an active packet deny to the edge active router to block the attack. The active packet deny contains the attack signature, and the time during which the attack must be blocked.

If the received passive packet is from a directly connected subnet, then depending on the passive packet type, the active router will either generate an active packet, or apply the required security policy to the passive packet before forwarding it. However, if the passive packet is not from a directly connected subnet then the active router will forward that packet. The active packet process is responsible for processing received active packets to determine how the active router will handle packets. If the active packet contains an encapsulated passive packet that is part of the connection establishment phase, then the active router will extract the passive packet and then either pass the extracted passive packet to the passive packet process or send it to the output queue. If the active packet carries a security policy then the active router will store the relevant information in the active router tables and generate an active packet acknowledgment. When the active packet process receives an active deny packet it will extract the relevant information and adds it to a Deny Table. Whenever an active router receives an active packet acknowledgment, it will compare the received packet with its tables and start forwarding packets from a directly connected client with the agreed security policy applied to them. However, if the active packet acknowledgment is not destined to the active router then it will be forwarded.

To illustrate the basic operation of our scheme, figure 2 shows a client (200.30.10.69) wishing to establish a TCP connection with a server (200.30.10.130) on a different subnet. The two edge routers for these subnets are R4 and R2. When R4 receives a TCP connection establishment request packet on port 2 from the client at time 1, it completes a Log File with the packet header information. R4 then generates an active packet that encapsulates the received passive packet that is received by R2 at time 2. The original passive packet is then extracted and passed through an intrusion detection process before being delivered to the server. When R2 receives a reply from the server at time 3, it will encapsulate this passive packet into an active packet which is sent to R4. It will also complete the TCP handshake with the server by issuing an ACK on behalf of the client. When R4 receives the active packet at time 4, it will extract the passive packet and forward it to the final destination. When the client replies with the ACK packet at time 5, R4 will then intercept the packet and compare the packet header with the contents of the Log File Table. If R4 does not find the packet information in the Log File Table, it will discard the packet; otherwise, it will set the ACK Flag field in the Log File Table.

When R4 receives the first data packet from the client at time 7, it will encapsulate it within an active packet and send it to R2. On receipt, R2 will confirm that this active packet is using the agreed security policy, and generate an active packet acknowledgment and send it back to R4. When this acknowledgement is received R4, it will set the A.P Ack Flag in its Log File Table and delete the active packet entry from its Table of Contents. From now on, R4 will encrypt other passive packets from this traffic flow using the agreed encryption scheme.

When R4 receives a connection termination from the client, it will encapsulate this passive packet into an active packet and send it to R2 that will then forward the original passive packet to the server. The connection termination from the server is received by R2 and forwarded to R4 within an active packet, which completes the transaction.

Clearly active routers used in our integrated approach do add an additional processing overhead to the conventional router's forwarding functionality by passing received packets to the active protocol process. In order to minimise this impact on the overall network architecture, it is only the edge active routers that are required to pass received packets to the active protocol processor. All intermediate active routers, which receive traffic from non-directly connected clients, need only forward the received traffic without passing it to the active protocol process, because a connection has already been established between the edge active routers.

5. Defeating a SYN Attack

To detect a SYN attack attempt, active routers need to count how many connection request packets (SYN =1) have been received from either the same client or from the same router. In order to validate the protocol logic of our scheme, a bespoke C++ simulation environment has been developed. This provides an idealised representation of networks and is able to model packet transfer within a network. The key aim of this simulation is to prove that our scheme has the functionality required to defeat denial of service attacks, rather than to assess overall system performance in terms of throughput or delay. Figure 3 illustrates the network architecture used to assess a range of attack scenarios.

Assume that an attacker located on LAN 3 attempts a SYN attack on Server 2 (200.30.10.68) located on LAN 2. The attacker conceals their identity by sending a stream of connection request packets to Server 2 each with a different spoofed IP source address from subnet LAN 3. This scenario shows how active routers R2 and R4 can deal with such an attack and at the same time allow other traffic to pass through. This traffic is generated by client (200.30.10.210) located on LAN 4 sending to server 2 (200.30.10.68) on LAN 2, client (200.30.10.90) located on LAN 2 sending to server 3 (200.30.10.130) on LAN 3, client (200.30.10.144) located on LAN 3 sending to server 1 (200.30.10.45) on LAN 1 and client (200.30.10.40) located on LAN 1 sending to Server 3 (200.30.10.130) on LAN 3.

When the timer used for blocking the attack traffic from LAN 3 to LAN 2 expires in both R2 and R4, these two routers will then start allowing traffic to pass from LAN 3 (200.30.10.140) to Server 2 (200.30.10.68) on LAN 2

Figure 4 illustrates the protocol time sequence diagram generated from our simulation environment. Every time R2 receives a connection request packet, it will encapsulate the passive packet into an active packet and forward it to R4. When R4 receives the active packet, it will extract the connection request packet and fill the intrusion detection table with the packet header information before forwarding the packet to Server 2. For example, when R4 receives the active packet at time 2, it will extract the passive packet, and fill the intrusion detection table as shown. Because this is the first connection request packet received that has an IP source address 200.30.10.140, R4 will then set the SYN from same client field in the intrusion detection table to 1. In addition, because this is the first time R4 receives a connection request packet from a client that is directly connected to R2, R4 will then set the SYN from same Router to 1. When R4 receives another connection request packet at time 3, it will check its header against the intrusion detection table entries, and notice that it is a connection request packet from a new IP source address 200.30.10.131 but that it is from the same active router R2. Therefore, R4 will set the SYN from same Client field to 1 for this packet entry, and will set the SYN from same Router to 2 as shown in the shaded area in the intrusion detection table at time 3.

Every time R4 receives a connection request packet, it will keep incrementing the SYN from the same Router field as long as it does not violate the defined rate limit for receiving a SYN from the same Router. The rate limit for receiving connection request packets from different clients, all coming from the same subnet, is set to 10. Therefore, when R4 receives the tenth connection request packet at time 11 from a client connected to R2, R4 will speculate that these are SYN attack attempts and it will start blocking this traffic. R4 informs R2 to block the suspected SYN attack traffic, by sending an active packet deny.

The data field in the active packet deny then informs the destination edge active router about the attack signature, such as the source and destination address, the type of the attack packets to block and the time period during which the attack profile must be blocked. In this case, the source address is set to any, the destination address is 200.30.10.68 and the time period is 20s. When R2 receives the active packet deny at time 12, it will validate the packet and fill its Deny Table with the attack packet signature as shown in the shaded area in the Deny Table in R2 at time 12. Consequently, when R2 receives connection request packets from two new clients 200.30.10.133 and 200.30.10.145 located on LAN 3 at time 12 and 13 respectively, it blocks their packets from passing through.

Whilst R2 is blocking all clients located on LAN 3 attempting to open a connection with Server 2, it still allows other traffic to pass through. For example, at time 22 client 200.30.10.144 located on LAN 3 sends a connection request packet to Server 1. When R2 receives this packet, it will compare this passive packet with the Deny Table, encapsulate the packet into an active packet and send it to R1. When R1 receives the packet at time 23, it will fill the intrusion detection table and forward the extracted passive packet to Server 1. At time 21, when client 200.30.10.40 on LAN 1 sends a connection request packet to Server 3, this traffic is then allowed to pass through by R1 and R2.

Equally while R4 is blocking any client connected to R2 from sending connection request packets to Server 2, it will also allow clients from other subnets to pass through. For example, at time 14, client 200.30.10.210 on LAN 4 sends a connection request packet to Server 2. When R4 receives the client packet encapsulated into an active packet at time 16, it will extract the connection request packet and fill the intrusion detection table as shown in the shaded area in the intrusion detection table in R4 at time 16. R4 then forwards the packet to Server 2. Traffic passing in the opposite direction to the original attack has not been blocked and when R4 receives a connection request packet from client 200.30.10.90 on LAN 2, it encapsulates the packet and forwards it to R2. When R2 receives the active packet at time 19, it extracts the connection request packet and forwards it to Server 3. Finally, at time 33 the Timer field in both the Deny Table in R2 and the intrusion detection table in R4 expires and traffic from LAN 3 can once again access Server 2.

This example has illustrated that our proposed active router integrated architecture is able to protect the network even if an attacker generates a random spoofed IP source address to accomplish a SYN attack. This attack is blocked close to the attack origin. The system can also automatically reconfigure after detecting and blocking an attack in order to permit other traffic to pass through. In additional it can ensure that blocking traffic from a specific subnet does not affect traffic from other subnets and traffic passing in the opposite direction to an attack continues to flow through the network.

6. Defeating a SMURF Attack

To detect a SMURF attack active routers count how many broadcast ICMP Ping packets have been received from either the same client or from the same router. Our simulation environment has set these limits to 4 and 5 respectively in order to observe the functionality of our scheme.

In figure 5 we assume that an attacker located on LAN 3 sends a stream of broadcast ICMP Ping packets to LAN 1 (200.30.10.63) using a random spoofed IP source address to make it harder for the active router to detect and block their attack traffic. At the same time, a legitimate client 200.30.10.200 located on LAN 4 sends a broadcast ICMP Ping packet to LAN 1.

When R2 receives the first ICMP Ping packet with an IP source address 200.30.10.133 at time 1, it will encapsulate the passive packet into an active packet and forward it to R1. When R1 receives the active packet at time 2, it will extract the ICMP Ping and fill the intrusion detection table as shown in the shaded area in the intrusion detection table at time 2. R1 will set both the ICMP from same Client and the ICMP from same Router field to 1 because it is the first time R1 has received an ICMP Ping packet from a client connected to R2. When R1 receives a broadcast ICMP Ping packet with IP address 200.30.10.131 at time 5, it will compare the packet with the intrusion detection table entries and note that this is the second packet from the subnet directly connected to R2 and increment the ICMP from same Router field. It will continue to do this every time it receives a broadcast ICMP Ping packet from a different client on the directly connected subnet to R2. At time 14, the rate limit for receiving broadcast ICMP Ping packets from the same router is violated and R1 will block the traffic, generate an active packet deny and send it to R2.

The Data field in the active packet deny contains the attack signature of the traffic to block. In this case, the active packet deny specifies that any broadcast ICMP Ping packets coming from the directly connected subnet and destined for LAN 1 (200.30.10.63) should be blocked for a period of time. Thus, when R2 receives a broadcast ICMP Ping packet at time 16, it will block the packet because the packet header matches the contents of its Deny Table.

Note however, that when R3 receives a broadcast ICMP Ping packet from client 200.30.10.200, it will encapsulate the packet and forward it to R1. When R1 receives the packet at time 16, it will check the intrusion detection table to decide either to allow the packet to pass through or not. Then R1 will fill the intrusion detection table with the received packet header information and it will allow the packet to pass through. When R1 receives the

ICMP Reply packets, it will encapsulate them and forward them to R3. When R3 receives the active packets it will extract the ICMP Reply and forward them to the final destination.

This example has illustrated that our proposed active router integrated architecture is able to defeat SMURF attacks by testing received ICMP traffic against defined thresholds. It allows ICMP packets not associated with a suspected attack to pass through the network normally and it can detect a SMURF attack even if the attacker uses a range of spoofed IP source addresses.

7. Conclusions

The challenge facing today's computer networks is to protect them from malicious attacks and threats originating from both inside and outside of an organisation's infrastructure. We have proposed a new approach that deploys active routers within a network to provide a distributed and adaptable defence system.

Each active router integrates firewall functionality, intrusion detection, and a cryptographic algorithm. The firewall and the intrusion detection are used to detect and block attack traffic coming from or going to a network. The cryptographic algorithm is used by the active routers to provide a secure communication between end users on their behalf. In addition, active routers use a dedicated active protocol to control the traffic passing through them, and to detect and to block the attack close to its origin.

We have, through simulation, demonstrated that our proposed architecture has the required functionality to defeat well known attack types. Using a distributed approach overcomes the limitation of conventional techniques that deploy a single firewall or management station to protect an entire network. Each active router provides its own protection of its attached subnets and collectively they are able to offer a strong defence for the whole network. Even if an active router is directly connected to two subnets, it can still protect one subnet from an attacker coming from the other subnet. Should an active router become compromised then the others continue to protect the network. The adoption of an active router approach also allows each one to adapt in real time and reconfigure to block certain traffic profiles whilst allowing others to pass through. Using a distributed array of active routers also means that when an attack is detected then it can be traced back and blocked at the active router that is closest to the point of origin of the attack. Finally, the adoption of data encryption between active routers adds further protection against an attacker originating from within the network.

We therefore believe that this paper has demonstrated that an active router based approach to defeating denial of service attacks is feasible and offers many potential benefits over current approaches being used to protect computer networks.

References

1. CERT Co-ordination Centre, 2003, “CERT/CC Statistics 1988-2003”, on-line http://www.cert.org/stats/cert_stats.html

2. Thomas Daniels and Eugene Spafford, 1999, “Identification of Host Audit Data to Detect Attacks on Low-Level IP Vulnerabilities”, Journal of Computer Security, Vol. 7, No. 1, pp3-35

3. CERT Co-ordination Centre, 2000, “CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks”, on-line http://www.cert.org/advisories/CA-1996-21.html

4. CERT Co-ordination Centre, 2000, “CERT Advisory CA-98.01 SMURF IP Denial of Service Attacks”, on-line http://www.cert.org/advisories/CA-98.01.smurf.html

5. El-Moussa, F.A., Linge, N., 2002, “An Integrated Active Network approach to Intrusion Detection, Network Access Policy and Firewall Functionality”, 3rd Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, PGNet 2002, 17-18th June 2002, EPSRC, Liverpool John Moores University, ISBN 1 902560 086, pp323-327

6. Elizzabeth D. Zwicky, Simon Cooper, and D Brent Chapman, 2000, “Building Internet Firewalls”, 2nd Edition, O’Reilly, ISBN: 1565928717

7. William La Cholter, Priya Narasimhna, Dan Sterne, Ravindra Balupari, Kelly Djahandari, Arvind Mani, Sandra Murphy, and NAI Labs Group, 2002, “IBAN: Intrusion Blcoker based on Active Networks”, Proceedings of DARPA Active Networks Conference and Exposition, San Francisco, USA, pp182-192

8. Dai Kashiwa, Eric Y Chen and Hitoshi Fuji, 2002, “Active Shaping: A Countermeasure against DDOS Attacks”, Proceedings of the 2nd European Conference in Universal Multi-service Networks (ECUMN 2002), pp171-179

9. 110 D Controneo, L Peluso, S P Romano and G Ventre, 2002, “An Active Security Protocol Against DOS Attacks”, Proceedings of the 7th International Symposium in Computers and Communications (ISCC 2002), pp496-501

10. Dan Schanackenberg, Harley Holliday, Randall Smith, Kelly Djahandari and Dan Sterne, 2001, “Cooperative Intrusion Traceback and Response Architecture (CITRA)”, Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX ’01), California, USA, pp56-68

11. Shu Zhang and Paratha Dasgupta, 2003, “Hardened Networks: Incremental Upgrading of the Internet for Attack Resilience”, Proceedings of IEEE Infocom 2003, ASU Technical Report, on-line http://www.public.asu.edu/~shuzhang/

12. Shu Zhang and Paratha Dasgupta, 2003, “Denying Denial of Service Attacks: A Router Based Solution”, Proceedings of IEEE ICDCS 2003, online http:\\www.public.asu.edu/~shuzhang/

13. William La Cholter, Priya Narasimhna, Dan Sterne, Ravindra Balupari, Kelly Djahandari, Arvind Mani, Sandra Murphy, Andrew Purtell and NAI Labs group, 2002, “Active Network Based DDOS Defence”, Proceedings of DARPA Active Networks Conference and Exposition (DANCE ’02), San Francisco, USA, pp193-203

14. Dan Sterne, Kelly Djahandari, Brett Wilson, Bill Babson, Harley Holliday, Travis Reid and NAI Labs group, 2001, “Autonomic Response to Distributed Denial of Service Attacks”, Proceedings of the fourth International Symposium on Recent Advances in Intrusion Detection (RAID 2001), number 2212 in LNCS, pp134-149, online http:\\www.thefengs.com/wuchang/work/cse581-winter2002/papers

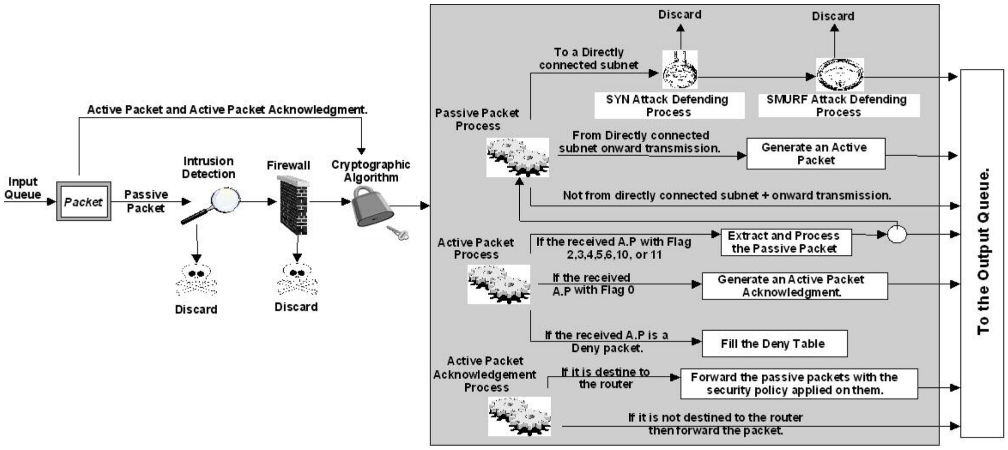

Active Packet and Active Packet Acknowledgment.
Input Queue
Packet
Passive Packet
Intrusion Detection
Firewall
Cryptographic Algorithm
Discard
Discard
Discard
Discard
To a Directly connected subnet
SYN Attack Defending Process
SMURF Attack Defending Process
Passive Packet Process
From Directly connected subnet onward transmission.
Generate an Active Packet
Not from directly connected subnet + onward transmission.
Active Packet Process
If the received A.P with Flag 2,3,4,5,6,10, or 11
Extract and Process the Passive Packet
If the received A.P with Flag 0
Generate an Active Packet Acknowledgment.
If the received A.P is a Deny packet.
Fill the Deny Table
Active Packet Acknowledgement Process
If it is destine to the router
Forward the passive packets with the security policy applied on them.
If it is not destined to the router then forward the packet.
To the Output Queue.

Figure 1 Active Router Internal Packet Processing

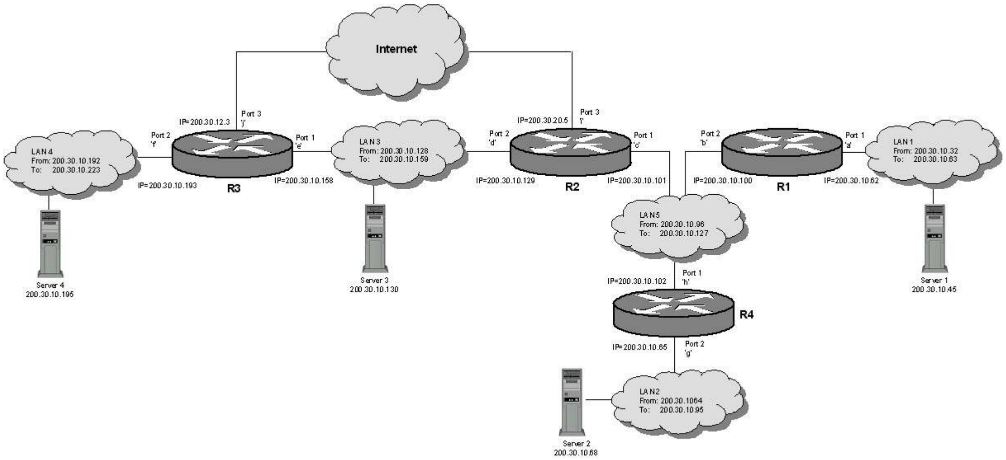
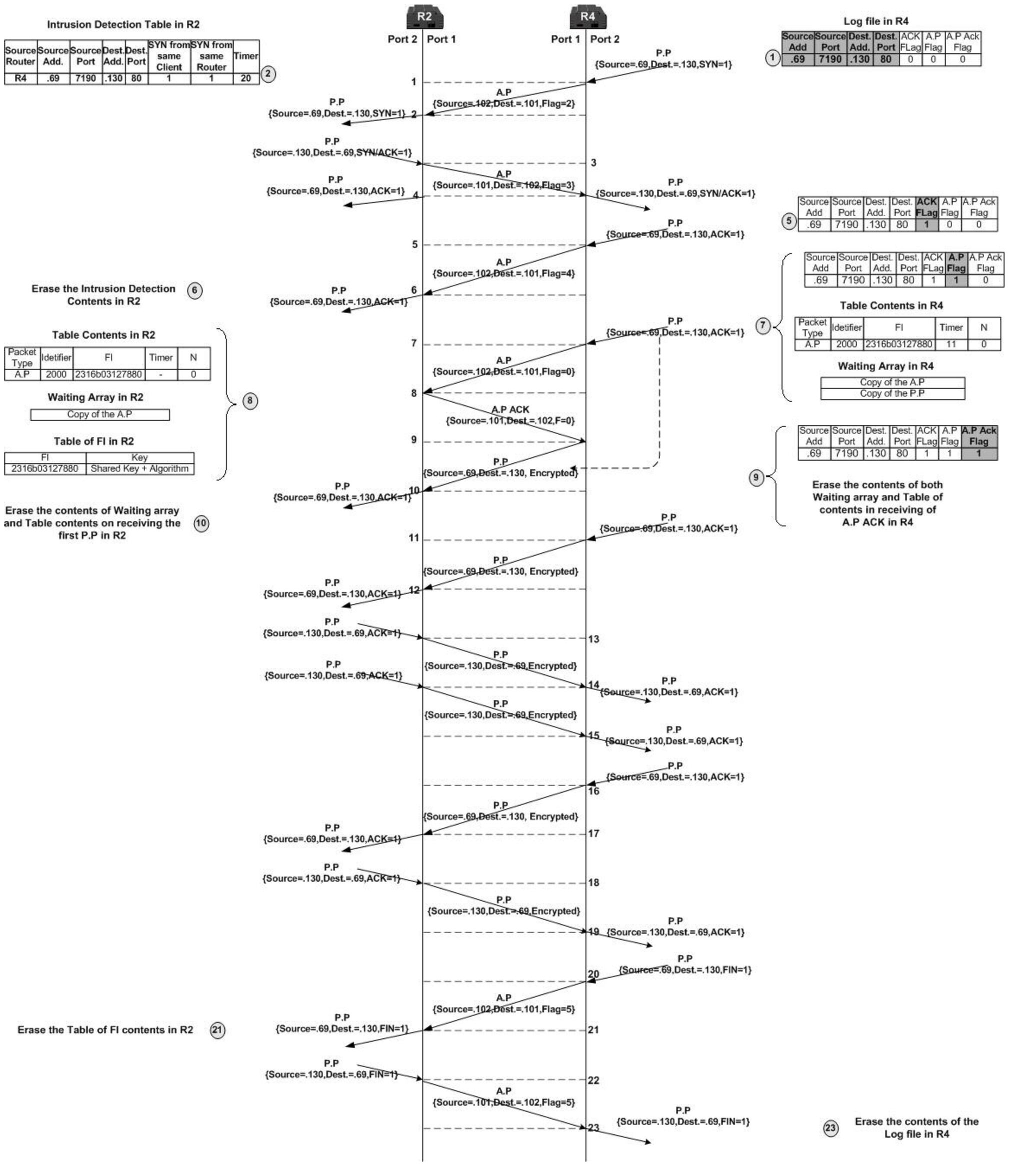


Figure 3 Simulated Network Architecture


R2
R4
Port 2
Port 1
Port 1
Port 2
P.P
{Source=.69,Dest.=.130,SYN=1}
1
A.P
{Source=.102,Dest.=.101,Flag=2}
P.P
{Source=.69,Dest.=.130,SYN=1}
2
P.P
{Source=.130,Dest.=.69,SYN/ACK=1}
3
A.P
{Source=.101,Dest.=.102,Flag=3}
P.P
{Source=.130,Dest.=.69,SYN/ACK=1}
P.P
{Source=.69,Dest.=.130,ACK=1}
4
P.P
{Source=.69,Dest.=.130,ACK=1}
5
A.P
{Source=.102,Dest.=.101,Flag=4}
P.P
{Source=.69,Dest.=.130,ACK=1}
6
P.P
{Source=.69,Dest.=.130,ACK=1}
7
A.P
{Source=.102,Dest.=.101,Flag=0}
8
A.P ACK
{Source=.101,Dest.=.102,F=0}
9
P.P
{Source=.69,Dest.=.130, Encrypted}
P.P
{Source=.69,Dest.=.130,ACK=1}
10
P.P
{Source=.69,Dest.=.130,ACK=1}
11
P.P
{Source=.69,Dest.=.130, Encrypted}
P.P
{Source=.69,Dest.=.130,ACK=1}
12
P.P
{Source=.130,Dest.=.69,ACK=1}
13
P.P
{Source=.130,Dest.=.69,Encrypted}
P.P
{Source=.130,Dest.=.69,ACK=1}
14
P.P
{Source=.130,Dest.=.69,ACK=1}
P.P
{Source=.130,Dest.=.69,Encrypted}
15
P.P
{Source=.130,Dest.=.69,ACK=1}
P.P
{Source=.69,Dest.=.130,ACK=1}
16
P.P
{Source=.69,Dest.=.130, Encrypted}
P.P
{Source=.69,Dest.=.130,ACK=1}
17
P.P
{Source=.130,Dest.=.69,ACK=1}
18
P.P
{Source=.130,Dest.=.69,Encrypted}
19
P.P
{Source=.130,Dest.=.69,ACK=1}
P.P
{Source=.69,Dest.=.130,FIN=1}
20
A.P
{Source=.102,Dest.=.101,Flag=5}
P.P
{Source=.69,Dest.=.130,FIN=1}
21
P.P
{Source=.130,Dest.=.69,FIN=1}
22
A.P
{Source=.101,Dest.=.102,Flag=5}
P.P
{Source=.130,Dest.=.69,FIN=1}
23

Intrusion Detection Table in R2

Source Router	Source Add.	Source Port	Dest. Add.	Dest. Port	SYN from same Client	SYN from same Router	Timer
R4	.69	7190	.130	80	1	1	20

2

Log file in R4

1

Source Add	Source Port	Dest. Add.	Dest. Port	ACK FLag	A.P Flag	A.P Ack Flag
.69	7190	.130	80	0	0	0

5

Source Add	Source Port	Dest. Add.	Dest. Port	ACK FLag	A.P Flag	A.P Ack Flag
.69	7190	.130	80	1	0	0

6 Erase the Intrusion Detection Contents in R2

7

Source Add	Source Port	Dest. Add.	Dest. Port	ACK FLag	A.P Flag	A.P Ack Flag
.69	7190	.130	80	1	1	0

Table Contents in R4

Packet Type	Idetifier	FI	Timer	N
A.P	2000	2316b03127880	11	0

Waiting Array in R4

Copy of the A.P
Copy of the P.P

8

Table Contents in R2

Packet Type	Idetifier	FI	Timer	N
A.P	2000	2316b03127880	-	0

Waiting Array in R2

Copy of the A.P

Table of FI in R2

FI	Key
2316b03127880	Shared Key + Algorithm

9

Source Add	Source Port	Dest. Add.	Dest. Port	ACK FLag	A.P Flag	A.P Ack Flag
.69	7190	.130	80	1	1	1

Erase the contents of both Waiting array and Table of contents in receiving of A.P ACK in R4

10 Erase the contents of Waiting array and Table contents on receiving the first P.P in R2

21 Erase the Table of FI contents in R2

23 Erase the contents of the Log file in R4

Figure 2 Client-Server Information Flow

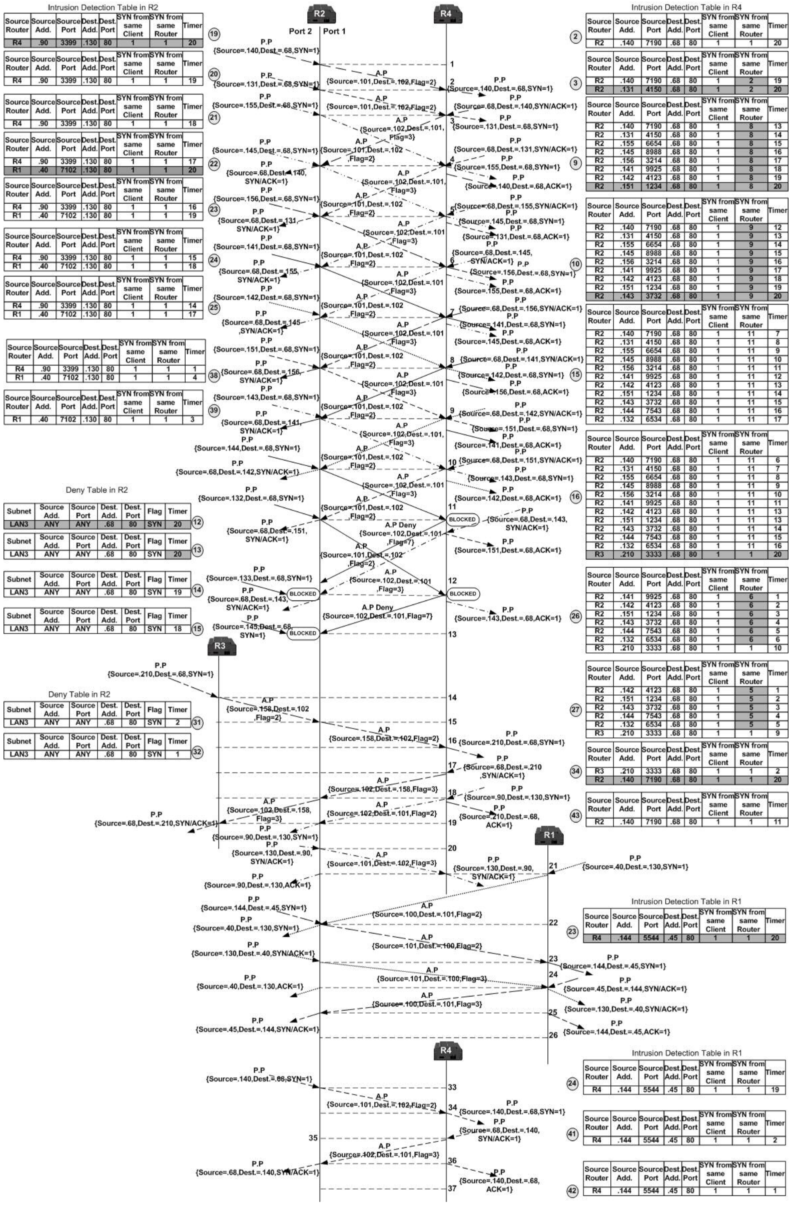
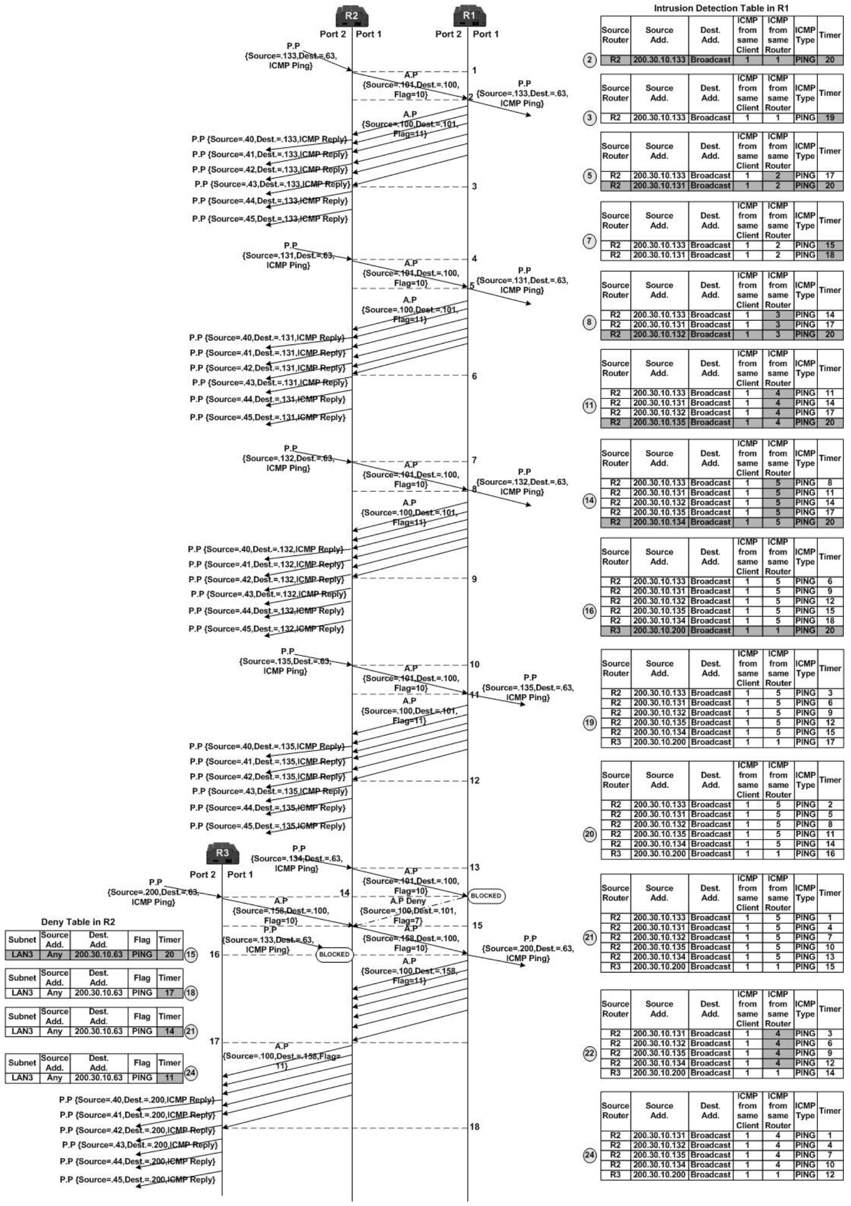


Figure 4 Defeating a SYN Attack


R2
Port 2
Port 1
R1
Port 2
Port 1
P.P {Source=.133,Dest.=.63, ICMP Ping}
1
A.P {Source=.101,Dest.=.100, Flag=10}
2
P.P {Source=.133,Dest.=.63, ICMP Ping}
A.P {Source=.100,Dest.=.101, Flag=11}
P.P {Source=.40,Dest.=.133,ICMP Reply}
P.P {Source=.41,Dest.=.133,ICMP Reply}
P.P {Source=.42,Dest.=.133,ICMP Reply}
P.P {Source=.43,Dest.=.133,ICMP Reply}
3
P.P {Source=.44,Dest.=.133,ICMP Reply}
P.P {Source=.45,Dest.=.133,ICMP Reply}
P.P {Source=.131,Dest.=.63, ICMP Ping}
4
A.P {Source=.101,Dest.=.100, Flag=10}
5
P.P {Source=.131,Dest.=.63, ICMP Ping}
A.P {Source=.100,Dest.=.101, Flag=11}
P.P {Source=.40,Dest.=.131,ICMP Reply}
P.P {Source=.41,Dest.=.131,ICMP Reply}
P.P {Source=.42,Dest.=.131,ICMP Reply}
6
P.P {Source=.43,Dest.=.131,ICMP Reply}
P.P {Source=.44,Dest.=.131,ICMP Reply}
P.P {Source=.45,Dest.=.131,ICMP Reply}
P.P {Source=.132,Dest.=.63, ICMP Ping}
7
A.P {Source=.101,Dest.=.100, Flag=10}
8
P.P {Source=.132,Dest.=.63, ICMP Ping}
A.P {Source=.100,Dest.=.101, Flag=11}
P.P {Source=.40,Dest.=.132,ICMP Reply}
P.P {Source=.41,Dest.=.132,ICMP Reply}
P.P {Source=.42,Dest.=.132,ICMP Reply}
9
P.P {Source=.43,Dest.=.132,ICMP Reply}
P.P {Source=.44,Dest.=.132,ICMP Reply}
P.P {Source=.45,Dest.=.132,ICMP Reply}
P.P {Source=.135,Dest.=.63, ICMP Ping}
10
A.P {Source=.101,Dest.=.100, Flag=10}
11
P.P {Source=.135,Dest.=.63, ICMP Ping}
A.P {Source=.100,Dest.=.101, Flag=11}
P.P {Source=.40,Dest.=.135,ICMP Reply}
P.P {Source=.41,Dest.=.135,ICMP Reply}
P.P {Source=.42,Dest.=.135,ICMP Reply}
12
P.P {Source=.43,Dest.=.135,ICMP Reply}
P.P {Source=.44,Dest.=.135,ICMP Reply}
P.P {Source=.45,Dest.=.135,ICMP Reply}
R3
Port 2
Port 1
P.P {Source=.134,Dest.=.63, ICMP Ping}
13
A.P {Source=.101,Dest.=.100, Flag=10}
BLOCKED
P.P {Source=.200,Dest.=.63, ICMP Ping}
14
A.P {Source=.158,Dest.=.100, Flag=10}
A.P Deny {Source=.100,Dest.=.101, Flag=7}
15
P.P {Source=.133,Dest.=.63, ICMP Ping}
BLOCKED
16
A.P {Source=.158,Dest.=.100, Flag=10}
P.P {Source=.200,Dest.=.63, ICMP Ping}
A.P {Source=.100,Dest.=.158, Flag=11}
17
A.P {Source=.100,Dest.=.158,Flag= 11}
P.P {Source=.40,Dest.=.200,ICMP Reply}
P.P {Source=.41,Dest.=.200,ICMP Reply}
P.P {Source=.42,Dest.=.200,ICMP Reply}
18
P.P {Source=.43,Dest.=.200,ICMP Reply}
P.P {Source=.44,Dest.=.200,ICMP Reply}
P.P {Source=.45,Dest.=.200,ICMP Reply}

Intrusion Detection Table in R1

2

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	1	PING	20

3

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	1	PING	19

5

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	2	PING	17
R2	200.30.10.131	Broadcast	1	2	PING	20

7

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	2	PING	15
R2	200.30.10.131	Broadcast	1	2	PING	18

8

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	3	PING	14
R2	200.30.10.131	Broadcast	1	3	PING	17
R2	200.30.10.132	Broadcast	1	3	PING	20

11

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	4	PING	11
R2	200.30.10.131	Broadcast	1	4	PING	14
R2	200.30.10.132	Broadcast	1	4	PING	17
R2	200.30.10.135	Broadcast	1	4	PING	20

14

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	5	PING	8
R2	200.30.10.131	Broadcast	1	5	PING	11
R2	200.30.10.132	Broadcast	1	5	PING	14
R2	200.30.10.135	Broadcast	1	5	PING	17
R2	200.30.10.134	Broadcast	1	5	PING	20

16

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	5	PING	6
R2	200.30.10.131	Broadcast	1	5	PING	9
R2	200.30.10.132	Broadcast	1	5	PING	12
R2	200.30.10.135	Broadcast	1	5	PING	15
R2	200.30.10.134	Broadcast	1	5	PING	18
R3	200.30.10.200	Broadcast	1	1	PING	20

19

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	5	PING	3
R2	200.30.10.131	Broadcast	1	5	PING	6
R2	200.30.10.132	Broadcast	1	5	PING	9
R2	200.30.10.135	Broadcast	1	5	PING	12
R2	200.30.10.134	Broadcast	1	5	PING	15
R3	200.30.10.200	Broadcast	1	1	PING	17

20

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	5	PING	2
R2	200.30.10.131	Broadcast	1	5	PING	5
R2	200.30.10.132	Broadcast	1	5	PING	8
R2	200.30.10.135	Broadcast	1	5	PING	11
R2	200.30.10.134	Broadcast	1	5	PING	14
R3	200.30.10.200	Broadcast	1	1	PING	16

21

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.133	Broadcast	1	5	PING	1
R2	200.30.10.131	Broadcast	1	5	PING	4
R2	200.30.10.132	Broadcast	1	5	PING	7
R2	200.30.10.135	Broadcast	1	5	PING	10
R2	200.30.10.134	Broadcast	1	5	PING	13
R3	200.30.10.200	Broadcast	1	1	PING	15

22

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.131	Broadcast	1	4	PING	3
R2	200.30.10.132	Broadcast	1	4	PING	6
R2	200.30.10.135	Broadcast	1	4	PING	9
R2	200.30.10.134	Broadcast	1	4	PING	12
R3	200.30.10.200	Broadcast	1	1	PING	14

24

Source Router	Source Add.	Dest. Add.	ICMP from same Client	ICMP from same Router	ICMP Type	Timer
R2	200.30.10.131	Broadcast	1	4	PING	1
R2	200.30.10.132	Broadcast	1	4	PING	4
R2	200.30.10.135	Broadcast	1	4	PING	7
R2	200.30.10.134	Broadcast	1	4	PING	10
R3	200.30.10.200	Broadcast	1	1	PING	12

Deny Table in R2

15

Subnet	Source Add.	Dest. Add.	Flag	Timer
LAN3	Any	200.30.10.63	PING	20

18

Subnet	Source Add.	Dest. Add.	Flag	Timer
LAN3	Any	200.30.10.63	PING	17

21

Subnet	Source Add.	Dest. Add.	Flag	Timer
LAN3	Any	200.30.10.63	PING	14

24

Subnet	Source Add.	Dest. Add.	Flag	Timer
LAN3	Any	200.30.10.63	PING	11

Figure 5 Defeating a SMURF Attack