RESEARCH ARTICLE OPEN ACCESS

LEAST FREQUENT SIP PROXY BASED VOIP USING DNS CACHING FOR DOS

[1] Ms. Swathi G.,[2] Ms. Abarna N.,

*1 M.Phil Research Scholar, PG & Research Department of Computer Science & Information Technology, Arcot Sri Mahalakshmi Women's College,,Villapakkam, Vellore, Tamil Nadu, India

*2Assistant Professor, PG & Research Department of Computer Science & Information Technology, Arcot Sri Mahalakshmi Women's College, Villapakkam, Vellore, Tamil Nadu, India

--************************--

Abstract:

We address the issue of a special denial of service (DoS) attack targeting a subcomponent of a Session Initiation Protocol (SIP) based VoIP network. Our focus is fctargeted at attacks that are addressed at the Domain Names Service (DNS). By flooding a SIP element with messages containing difficult-resolvable domain names, it is possible to block the target for a considerable amount of time. We evaluate possibilities to mitigate these effects and show that over-provisioning is not sufficient to handle such attacks. We present results gained from testing with actual SIP providers of a counter solution based on a non-blocking DNS caching solution. Within this cache we evaluate different caching strategies and show that the Least-Frequently-Used caching strategy gives best results to mitigate this kind of attack. (1) LFU policy could help cache to work better than other replacement policies. (2) If there are enough CPU and memory resource, the more parallel processes of SIP proxy, the better performance of SIP proxy. (3) If there are enough CPU and memory resource, the more cache entries, the better performance of SIP proxy. (4) The longer attacking interval, the better performance of SIP proxy

Keywords— **Intrusion Detection, SIP Proxy, Domain Name System, VOIP , LFU, CPU Performance.**

--************************--

I. INTRODUCTION

Denial of service (DoS) attacks present one of the most significant threats to assurance of dependable and secure information systems. Rapid development of new and increasingly sophisticated attacks requires resourcefulness in designing and implementing reliable defenses [2]. This paper presents an overview of current DoS attack and defense concepts, from a theoretical and practical point of view. Considering the elaborated DoS mechanisms, main directions are proposed for future research required in defending against the evolving threat[3],[2].

Latest security incident trend statistics, currently showing an increase in denial of service (DoS) attacks, stress out availability as the security objective of choice when it comes to malicious online activities[7], with goals such as extortion, blackmailing, protest and political activism, but sometimes also plain fun as primary motivation. These types of attacks have been known for a long time, but gained popularity with recent events related to the Anonymous hacktivist collective. Furthermore, DoS has not only grown in sophistication, but is easily pursued with different open source software, underground DoS toolkits etc. One of the most challenging DoS attack characteristics is that, in most cases, specific combinations of legitimate packets are used to produce a malicious, disruptive effect[2],[1]. As DoS schemes have evolved with years, fewer resources are needed to perform devastating attacks, so in certain scenarios a single client is enough to bring down an entire network, as opposed to rudimentary techniques which impose large networked attack resources necessary to deny service to a single target. Considering the progression, further research is needed for successful mitigation[6]. This paper aims to introduce an overview of the most prevailing current issues emerging from DoS in general (i.e. relative attack source quantity), from both attack and defence perspective. Security in mobile ad hoc network is essential even for basic network functions like routing which are carried out by the nodes themselves rather than specialized routers. The intruder in the ad hoc network can come from anywhere, along any direction, and target any communication channel in the network. Compare this with a wired network where the intruder gains physical access to the wired link or can pass through security holes at firewalls and routers [3].

As these attacks are typically performed on a massive scale using a large number of computer resources, they are commonly referred to as distributed denial of service attacks (DDoS) [10],[2]. This is also one of the reasons why most of the academic references use this particular DoS subset terminology, but the main concepts can be viewed in general terms. In order to achieve the necessary quantity [9], a malicious individual must essentially compromise a large number of machines, which are then used to launch DDoS attacks, as well as other operations, including scanning and exploitation of other reachable computers. Network under such control model is called a botnet, built of compromised computers, known as bots or zombies[1].

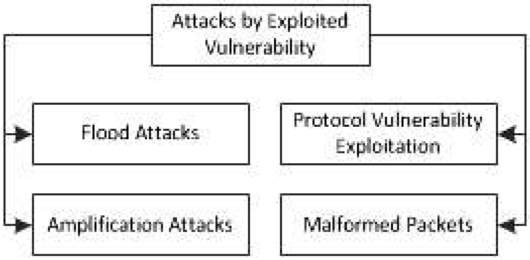

Attacks by Exploited Vulnerability
Flood Attacks
Protocol Vulnerability Exploitation
Amplification Attacks
Malformed Packets

Fig 1.1: DoS attacks classified by exploited vulnerabilities

1.1 DIFFERING ATTACKS BY EXPLOITED VULNERABILITIES

Considering a typical vulnerability being exploited, DOS attacks may be narrowed to four major categories, as described hereinafter.

1) Flood Attacks:

Perhaps the most intuitive DOS method is flooding, which means performing a very large number of communication requests towards the target. Even a large enough number of legitimate requests may make the target unresponsive, but the effect is greater with modifications that raise the server workload. One such well known example is SYN flood, in which the attacker, during the establishment of TCP connections [9], sends only SYN messages, i.e. does not complete the three-way handshake with an ACK, therefore maintaining half-open connections and ultimately exhausting the target. A more interesting flooding attack is DNS flood [5], based on excessive legitimate DNS requests and relying on higher workload on the server side per single DNS query.

2) Amplification Attacks:

Since flooding by definition requires significant attacking resources in order to be effective, more clever approaches to this problem have been developed in the form of attack amplification, achieved through utilization of reflectors. Essentially, certain properties may be manipulated to trigger responses significantly larger than the initiated requests, producing increased intensity of the attack. Therefore, given the same initial resource conditions, taking advantage of attack amplifiers [10] produces greater damage than traditional plain overwhelming. One of these attacks is the reflective DNS attack where spoofed DNS requests with the target IP address are issued to DNS servers. This results with redirection of responses to the victim, but also with the amplification due to DNS responses being larger than DNS requests. Another great example in this category is the recursive DNS attack, taking advantage of recursive DNS querying by massively issuing DNS requests with non-existent domain names towards the victim DNS server.

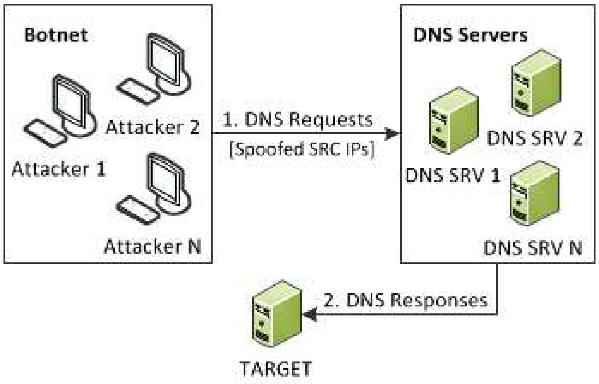


Fig1.2: Reflective DNS attack

Conceptually similar are the most current and increasingly popular NTP amplification attacks. The core difference is that, in this case, NTP servers are used as reflectors, while the attack itself exploits the monlist remote NTP command. Precisely, monlist returns the list of last 600 servers that connected to the related NTP server, with the side-effect of the server response being significantly larger than the client request, having the amplification factor rise up to 200 times to the original request. Targeting these replies using source IP spoofing creates very high DoS potential.

1.2 DNS USAGE OF SIP

1.2.1 Audience

The primary audience for this document consists of application protocol designers, who can reference this document instead of defining their own rules for the representation and verification of application service identity. Secondarily, the audience consists of certification authorities, service providers, and client developers from technology communities that might reuse the recommendations in this document when defining certificate issuance policies, generating certificate signing requests, or writing software algorithms for identity matching..

SIP high availability remains in its nascent stage, resulting in a lack of an industry standard for achieving SIP high availability and reliability. Until now, there has not been a reliable mechanism available to determine the state of a SIP Proxy or a Media server and when calls should be routed away from them. Calls get routed away after a server completely fails, and valuable time is spent probing and connecting to the backup server, which results in long service interruptions.

- ***SIP Proxy Servers:-*** SIP Proxy servers could become unavailable or overloaded in the middle of a session. This would cause long service interruptions as the clients cannot be redirected to an available server.
- ***SIP Media Servers:-*** Media servers could become unavailable or overloaded in the middle of a session. This would cause long service interruptions as the clients cannot be redirected to an available server.
- ***SIP Security:-*** Site security could be compromised because of security attacks against SIP vulnerabilities like open RTP (Real-Time Transport Protocol) and SIP channels.
- ***Site Availability:-*** The entire site could be unavailable because of a link outage. This would cause long service interruptions until the link became available. The site could also become unavailable because of a power outage. This would cause long service interruptions until the power was restored.
- ***Service Quality:-*** Voice and video traffic are extremely sensitive to delays. Long delays cause degradation in the traffic quality rendering the service unusable and unreliable.

II. RELATED WORK

This document attempts to generalize best practices from the many application technologies that currently use PKIX certificates with TLS. Such technologies include, but are not limited to:

- The Internet Message Access Protocol [IMAP] and the Post Office Protocol [POP3]; [USINGTLS]
- The Hypertext Transfer Protocol [HTTP]; [HTTP-TLS][2003]
- The Lightweight Directory Access Protocol [LDAP]; [LDAP-AUTH] and its predecessor [LDAP-TLS][2012]
- The Simple Mail Transfer Protocol [SMTP]; see also [SMTP-AUTH] and [SMTP-TLS][1999]
- The Extensible Messaging and Presence Protocol [XMPP]; [XMPP-OLD][2001]
- The Network News Transfer Protocol [NNTP]; see also [NNTP-TLS][2001]
- The NETCONF Configuration Protocol [NETCONF]; [NETCONF-SSH] and [NETCONF-TLS][2010]
- The Syslog Protocol [SYSLOG]; see also [SYSLOG-TLS] and [SYSLOG-DTLS][2011]
- The Session Initiation Protocol [SIP]; [SIP-CERTS]
- The Simple Network Management Protocol [SNMP]; [SNMP-TLS][2000]

One way of doing call forwarding with ENUM is illustrated in the next figure. The caller uses the telephone to dial the number of another subscriber, which leads to an ENUM lookup (such as is provided by SIP Broker). The DNS responds to the caller by returning a list with NAPTR records for VoIP communication, telephone numbers and email addresses. Next, an attempt will be made, using the VoIP record from this list, to establish a connection with the subscriber. If the subscriber is not online, the next record selected will be that for a connection to a PSTN or mobile telephone. If this attempt fails too, a voice message will be sent to the subscriber via a listed email address. Sub domains of e164.arpa are delegated on a country-code basis by the ITU. Each delegation is normally made to a regulatory body designated by the national government for the country code concerned. What happens at a country level is a National Matter. In general the conventional DNS registry-registrar model is used. The national ENUM registry manages and operates the DNS infrastructure and related systems for country-code.e164.arpa. It takes registration requests from registrars who are agents of the end users, the registrants. Registrars are typically VoIP providers and telcos who bundle an ENUM registration as part of a VoIP service package. People using an ENUM-enabled VoIP service can dial the registrant's existing number and be connected to the registrant's VoIP telephone over the Internet instead of using the PSTN. When they call someone who does not use ENUM, calls complete over the Public Switched Telephone Network or PSTN in the usual manner. Support for .e164.arpa varies widely between countries; many do not support it at all

Identifier type:

A formally defined category of identifier that can be included in a certificate and therefore that can also be used for matching purposes. For conciseness and convenience, we define the following identifier types of interest, which are based on those found in the PKIX specification [PKIX] and various PKIX extensions.

- CN-ID = a Relative Distinguished Name (RDN) in the certificate subject field that contains one and only one attribute-type- and-value pair of type Common Name (CN), where the value matches the overall form of a domain name (informally, dot-separated letter-digit-hyphen labels); see [PKIX] and also [LDAP-SCHEMA]

- DNS-ID = a subjectAltName entry of type dNSName; see [PKIX]

- SRV-ID = a subjectAltName entry of type otherName whose name form is SRVName; see [SRVNAME]

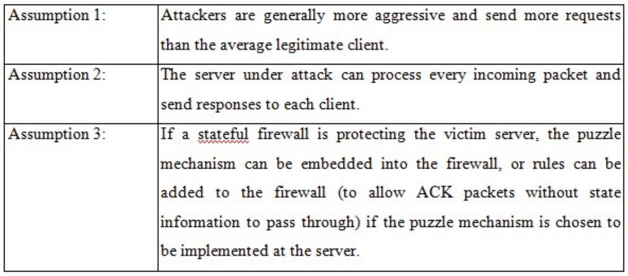
- URI-ID = a subjectAltName entry of type uniformResourceIdentifier whose value includes both

- a ”scheme" and (ii) a "host" component (or its equivalent) that matches the "reg-name" rule (where the quoted terms represent the associated [ABNF] productions from [URI]); see [PKIX] and [URI]

- **Interactive client:** A software agent or device that is directly controlled by a human user. (Other specifications related to security and application protocols, such as [WSC-UI], often referto this entity as a "user agent".)

III. TCP BASED DOS WITH SIP

In recent years, TCP client puzzles have been proposed to mitigate attacks on the transport and application layers. In this chapter, the design, implementation details, and simulation results of Puzzle TCP, pTCP, are presented. However, before an introduction to pTCP is given, it is necessary to define the assumptions that are made



Assumption 1:	Attackers are generally more aggressive and send more requests than the average legitimate client.
Assumption 2:	The server under attack can process every incoming packet and send responses to each client.
Assumption 3:	If a stateful firewall is protecting the victim server, the puzzle mechanism can be embedded into the firewall, or rules can be added to the firewall (to allow ACK packets without state information to pass through) if the puzzle mechanism is chosen to be implemented at the server.

Table3.1: Assumption in pTCP

In order for client puzzles to be truly effective, they must be placed below the application layer. Feng documented the need to place client puzzles in the TCP or IP-layer. In the following section, we show how client puzzles can be integrated within the TCP stack to prevent resource-exhaustion DoS attacks. This section also gives a detailed description on how the client and server interact in pTCP.

pTCP implements a similar three-way handshake to establish a connection. When a server receives a packet with the SYN code bit set, it normally replies with a packet with the SYN and ACK code bits set (SYN-ACK packet). However, if the server is experiencing heavy traffic (e.g., flash crowd or DoS attack), then it replies with a challenge to the client which includes a server nonce and difficulty level embedded into the SYN-ACK packet. A server can determine when this is necessary by examining the SYN queue. When the queue reaches near its maximum capacity, puzzles can be turned on. When it is necessary for a server to issue a challenge to a client, the server removes the state information for that client. Thus, for pending connections there are no resources allocated on the server other than the server nonce and current difficulty level, which is common to all potential clients. Therefore, the server is not susceptible to half-open attacks designed to consume server resources. The server nonce is the same for all clients during a 60 second time period. If a client solves a puzzle in the previous time epoch and submits the answer in the following time epoch, the answer is considered incorrect. The client's connection will be reset and the client will need to make another attempt to complete the connection. Since most puzzles take less than a few seconds to solve, the 60 second server nonce period should be adequate.

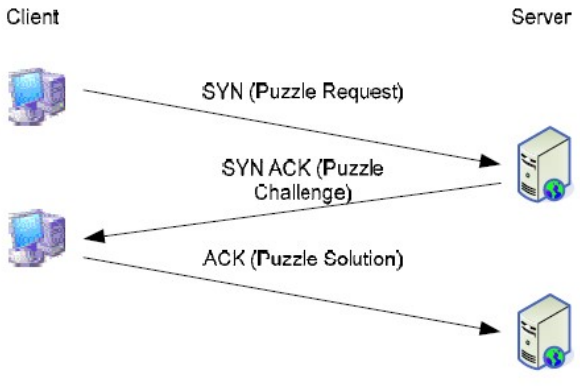


Fig3.1: Client-Server interaction in pTCP

3.1 Implementation of pTCP

The following sections discuss the implementation details of pTCP. The details and algorithms of the design are presented along with other important details including a description of the code changes within the Linux kernel and some of the specific modifications to the existing TCP stack.

pTCP was implemented into the TCP stack in Linux [39]. In pTCP, all of the puzzle information (puzzle request, puzzle challenge, and puzzle solution) is passed using the TCP header. Since data is not normally passed in the three-way handshake in TCP, any additional data needs to be placed within the TCP header. The options field in the TCP header was utilized to pass the puzzle requests, puzzle challenges, and puzzle answers.

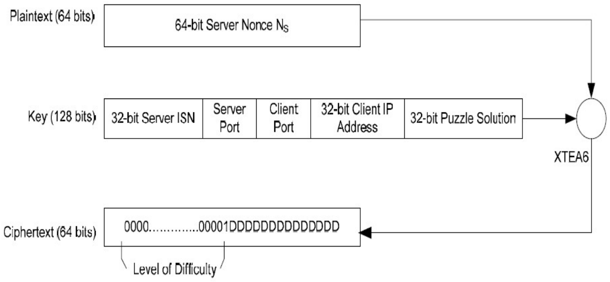


Fig 3.2 : XTEA6 client puzzle

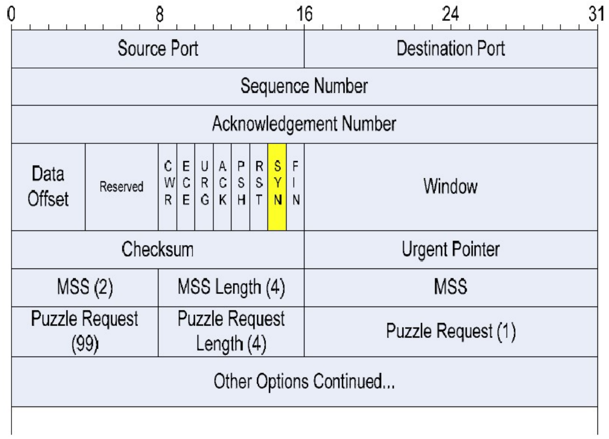


Fig 3.3 : Puzzle request packet

Since the server accepts anonymous ACK packets with puzzle solutions, this can result in another type of an attack. An attacker could be sniffing packets and discover a solution to the puzzle that another computer has already solved. The attacker could then send this puzzle solution as its own. To counteract this, our puzzle uses various parameters from the TCP and IP header. By embedding these values into the puzzle, an attacker cannot use previous solutions. A more likely attack is when an attacker could flood the server with false puzzle solutions. This could potentially be another form of a DoS attack that attempts to exhaust the server's CPU. It is important to always assume that attackers will modify their attack to exploit weaknesses in a modified version of TCP. In this case, an attacker would create an ACKflood tool rather than the traditional synflood tool. Since the client puzzle algorithm is extremely fast, pTCP would be able to discard these false answers quickly and efficiently.

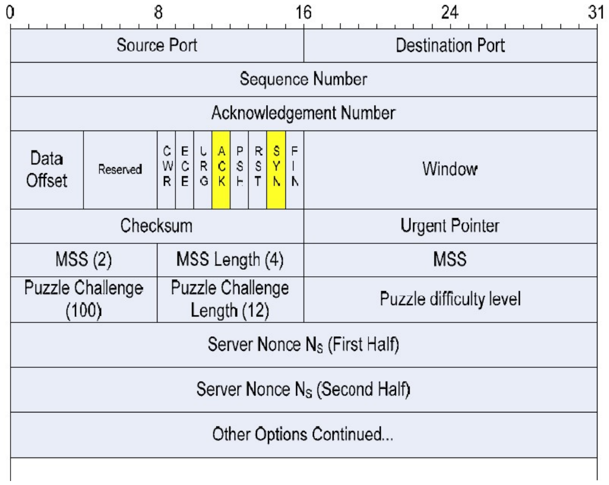


Fig 3.4: Puzzle challenge packet

```
Plaintext = NS
        Key[0] = Server ISN
        Key[1] = Server Port \\ Local Port
        Key[2] = Local IP address
While(Answer is not found)
        Key[3] = get_random_bytes() Ciphertext =
        XTEA6(Plaintext,Key)
If Ciphertext meets difficulty constraint Return
Else
Continue loop
```

3.2 THE PUZZLE ALGORITHM

An important aspect to pTCP is the selection of the cryptographic algorithm used for the client puzzle. In addition to puzzles using XTEA6 and MD5 [36, 41], we also tested SHA-1 [36, 42]. According to our simulation results, the puzzle algorithm based on XTEA6 had the fastest solution verification time among the three that were examined. MD5 was also faster than SHA-1, so only XTEA6 and MD5 were implemented into the kernel.

IV. DETECTING AND PREVENTING DNS DOS ATTACKS

DNS DoS protection is a type of protocol security. DNS attack detection and prevention serves two functions:

- To detect and automatically drop DNS packets that are malformed or contain errors.
- To log unusual increases in DNS packets of any type, including packets that are malformed, packets that contain errors, or packets of any other type that appear to rapidly increase

Creating a custom DNS profile to firewall DNS Traffic

Ensure that you have a DNS security profile created before you configure this system DNS profile.

1. You can create a custom DNS profile to configure the BIG-IP system firewall traffic through the system.
2. 1On the Main tab, click Local Traffic > Profiles > Services > DNS.
3. The DNS profile list screen opens.
4. Click Create. The New DNS Profile screen opens.
5. In the Name field, type a unique name for the profile.
6. In the General Properties area, from the Parent Profile list, accept the default dns profile.
7. Select the Custom check box.
8. In the DNS Traffic area, from the DNS Security list, select Enabled.
9. In the DNS Traffic area, from the DNS Security Profile Name list, select the name of the DNS firewall profile.
10. Click Finished.
11. Assign the custom DNS profile to the virtual server that handles the DNS traffic that you want to firewall

Assigning a DNS profile to a virtual server

1. On the Main tab, click Local Traffic > Virtual Servers.
2. The Virtual Server List screen opens.
3. Click the name of the virtual server you want to modify.
4. From the Configuration list, select Advanced.
5. From the DNS Profile list, select the profile you want to assign to the virtual server.
6. Click Update. The virtual server now handles DNS traffic.

4.1 Methods of Attack Detection

There have been several ideas on achieving DoS traffic detection, each holding various different solutions. One of the options is leveraging statistics to differentiate abnormal behaviour. For instance, malicious traffic may be detected using statistical tests for SYN and ACK messages, thus comparing incoming traffic with a baseline stated as normal activity. A more flexible approach to detecting DoS attacks is using fuzzy, evolutionary and neuro computing, which enables intelligent traffic classification and hard problem solving. Accordingly, radial basis function networks, which are a type of single-layer neural networks, may be used to learn to identify normal traffic and detect anomalies. likely due to simpler development. Typically, it is in the form of signature-based attack detection, which presents comparisons of incoming traffic with already known attack or whitelist patterns (i.e. knowledge). In general, signature-based detection is highly efficient at flagging known threats and attacks, but has no value with new ones, which obviously do not exist in the signature database. These problems are not only true for DoS, but for all other attacks.

4.2 Methods of Attack Prevention

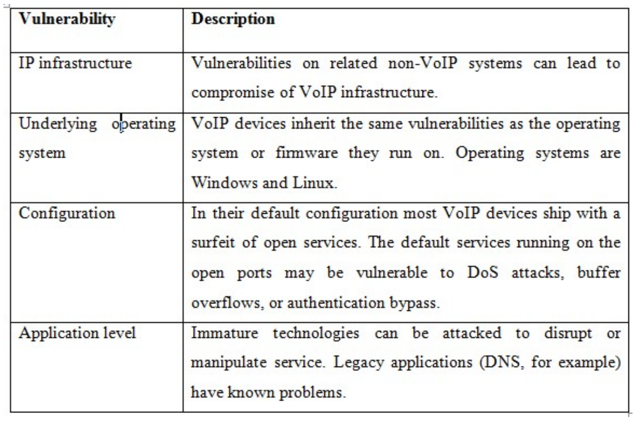
Earlier described DoS detection techniques ideally have follow-up response actions, which mitigate adversary attack effects. However, certain methods can be viewed as primarily prevention-oriented, as they aim at preventing the DoS from occurring in the first place. For example common principles such as system hardening (e.g. running only necessary and patched services), have a role in eliminating certain DoS vectors by solely reducing the attack surface. Also, tolerance-based mechanisms, e.g. simple allocation and combining larger amounts of system resources (e.g. clustering, load balancing, network bandwidth etc.) can thwart some forms of attacks, but obviously this works well only to a certain point.

4.3 Proactive Defence

Information security is continuously in an arms race state, where new attacking schemes are one or more steps ahead of the defence capabilities. This has brought about an increasing need for new measures for pertaining successful cyber defence and information assurance. Solutions for these problems may be achieved through the concepts of active defence (Table 1) as an additional layer in the security and sustainability posture, which are applicable to most cyber threats, including DoS. It is therefore very interesting to consider retaliation and counterattacks, which is still off legal limits for common utilization, but may be of interest in case of cyberwar events. Similar are precautionary attacks, which origin from a military tactic frequently employed in recent past. A significant issue is effective and correct attack attribution,

4.4 Defence Dependability

Along with development of defence systems and protocols, it is also important to perform their evaluation through dependability modeling. The specific case of DoS attacks is probably best suited with Markov chains, useful for both availability and reliability modeling, based on states and state transitions. The discrete-time Markov chains (DTMC) assume state transition dependence is related only to the current state, without any time variance of the transition probabilities. These techniques have significant applicability in DoS research and enable quantitative evaluation, comparisons and improvement of current defence approaches. For example, the random port hopping (RPH) protocol.



Vulnerability	Description
IP infrastructure	Vulnerabilities on related non-VoIP systems can lead to compromise of VoIP infrastructure.
Underlying operating system	VoIP devices inherit the same vulnerabilities as the operating system or firmware they run on. Operating systems are Windows and Linux.
Configuration	In their default configuration most VoIP devices ship with a surfeit of open services. The default services running on the open ports may be vulnerable to DoS attacks, buffer overflows, or authentication bypass.
Application level	Immature technologies can be attacked to disrupt or manipulate service. Legacy applications (DNS, for example) have known problems.

Table 4.1 : VoIP Vulnerabilities

V EVALUATION RESULT

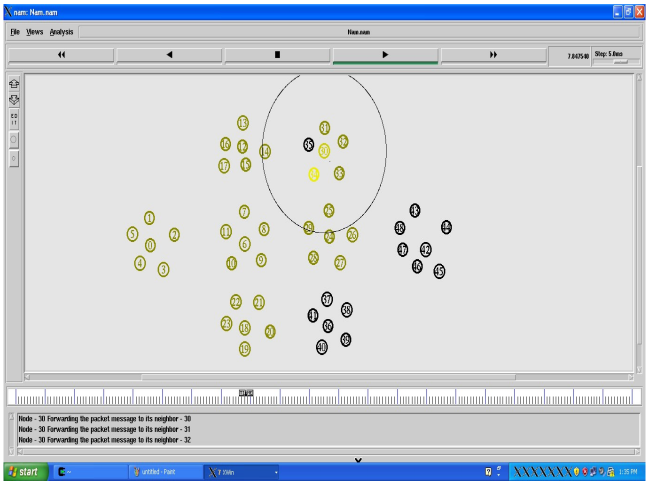
In this section, three experiments were designed to evaluate the performance of the improved clustering algorithm. The simulating program was developed by our team using NS2. In the following experiments, Most of studies only consider that wireless sensor networks are equipped with only Omni-directional antennas, which can cause high collisions. It is shown that the per node throughput in such networks is decreased with the increased number of nodes. Thus, the transmission with multiple short - range hops is preferred to reduce the interference. However, other studies show that the transmission delay increases with the increased number of hops. Found that using directional antennas not only can increase the throughput capacity but also can decrease the delay by reducing the number of hops.

Contains both merits and limitations when people use it to simulate WSNs. To the merits, firstly as a non-specific network simulator, NS-2 can support a considerable range of protocols in all layers. For example, the ad-hoc and WSN specific protocols are provided by NS-2. Secondly, the open source model saves the cost of simulation, and online documents allow the users easily to modify and improve the codes. However, this simulator has some limitations. Firstly, people who want to use this simulator need to familiar with writing scripting language and modeling technique; the Tool Command Language is somewhat difficult to understand and write. Secondly, sometimes using NS-2 is more complex and time-consuming than other simulators to model a desired job. Thirdly, NS-2 provides a poor graphical support, no Graphical User Interface (GUI).

Parameter used in the Simulation

Parameter	Value
Simulation time	300
Number of nodes	46
Traffic model	CBR
Node Placement	Uniform
Performance parameter	Energy consumption, End to End delay
Routing protocol	AODV
No Coding	TCL
Node ID	1
Signal Strength	70%
Mobility Connection	Clustering Head

Table 5.1: Simulation Parameters



Node - 30 Forwarding the packet message to its neighbor - 30
Node - 30 Forwarding the packet message to its neighbor - 31
Node - 30 Forwarding the packet message to its neighbor - 32

Fig 5.2: Proposed Node

In Domain Name System in which each sensor node randomly and alternatively stays in an active mode or a sleep mode. The active mode consists of two phases, called the full-active phase and the semi-active phase. By using cluster node, 48 nodes are created. Then the cluster head is chosen randomly. The amount of energy consumed in MAC is less than MAC because the packets are chosen based on energy level. The X Window System, version 11 (often "X11", or simply "X") is the standard graphical environment under WindowsXP and GNU/Linux; it is also available for other platforms, including Mac OS X and MS-Windows. X applications ("clients") exchange data with an X server (another application). The X server receives and interprets instructions from the clients for displaying the clients' windows, and it collects and transmits keyboard and mouse input events to the clients. The xorg packages available with Cygwin (collectively, "Cygwin/X") provide a high-quality X server,

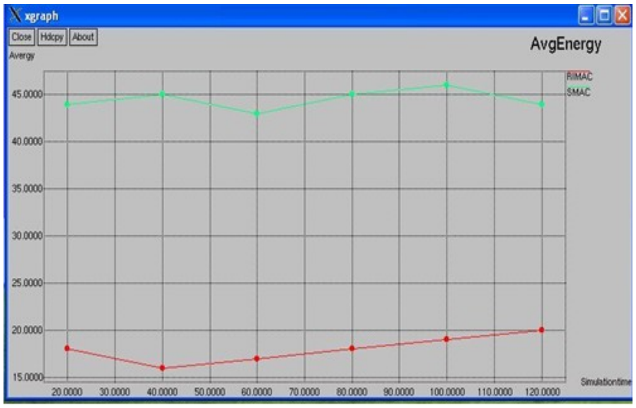


Fig 5.3 : Average Energy

Many protocols in Domain Name System use packet delivery ratio (PDR) as a metric to select the best route, transmission rate or power. PDR is normally estimated either by counting the number of received hello/data messages in a small period of time, i.e., less than 1 second, or by taking the history of PDR into account. The first method is accurate but requires many packets to be sent, which costs too much energy. The second one is energy efficient, but fails to achieve good accuracy. A Sensor Network consists of multiple detection stations called sensor nodes, each of which is small, lightweight and portable. Every sensor node is equipped with a transducer, microcomputer, transceiver and power source.

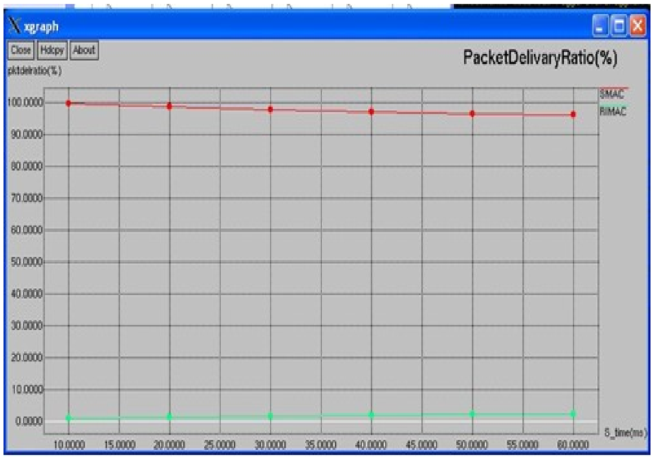


Fig 5.4: Delivery Ratio

CONCLUSION

This research designed and developed the First of all, I would like to answer the three research questions mentioned in the proposal. How to find a proper method to mitigate the effect of DoS attack via DNS request? During the thesis work, I developed the proposed prototype, compared the performance of it with other methods and evaluated in the simulated SIP environment. The result of it is obviously better than other methods from different aspects. Therefore, up to now, the special DNS cache is the best method to solve this problem in our research scope. Which factors of DNS cache and SIP proxy (e.g. caching replacement policy, cache entry These six factors are the most general in SIP architecture despite of hardware and platform. Through these experiments, I found last four factors quite affect the performance of SIP proxy. The relationship of these four factors and the performance of SIP proxy are as follow, (Assume there are not enough cache entries to accommodate all records): (1) LFU policy could help cache to work better than other replacement policies. (2) If there are enough CPU and memory resource, the more parallel processes of SIP proxy, the better performance of SIP proxy. (3) If there are enough CPU and memory resource, the more cache entries, the better performance of SIP proxy. (4) The longer attacking interval, the better performance of SIP proxy.

FUTURE WORK

Enhancement the defense system. For example, in the research, we already know that caching replacement policy, cache entry number, parallel processes number of proxy and attacking interval affect the performance of cache and proxy. We can determine the former three factors from the server side but the attacking interval depends on the attacker, not us. Fortunately, it is possible to decrease attacking speed by some intelligence filter technique. Now we devote to combine this solution with other IDS and IPS in the SIP infrastructure and constructing a scalable defence system.

REFERENCES:

[1] CSI and FBI, “2004 CSI/FBI Computer Crime and Security Survey,” Mar. 2004.

[2] J. Mirkovic et al., Internet Denial of Service: Attack and Defense Mechanisms, Prentice Hall, 2005.

[3] J. Rosenberg et al., “Session Initiation Protocol,” RFC 3261, 2002.

[4] R. Fielding et al., “Hypertext Transfer Protocol — HTTP/1.1,” RFC 2616, June 1999.

[5] A. B. Johnston, SIP — Understanding the Session Initiation Protocol, 2nd ed., Artech House, 2004.

[6] 3GPP Tech. Spec. 3GPP TS 24.228 “Technical Specification Group Core Network; Signaling Flows for the IP Multimedia Call Control Based on SIP and SDP,” 2003.

[7] J. Rosenberg et al., “STUN — Simple Traversal of UDP Through NATs,” RFC 3489, Mar. 2003.

[9] J. Rosenberg, “Request Header Integrity in SIP and HTTP Digest Using Predictive Nonces,” expired Internet

draft, work in progress, IETF, June 2001. draft-rosenberg-sip-http-pnonce-00.txt

[8] J. Franks et al., “HTTP Authentication: Basic and Digest Access Authentication,” Internet Engineering Task Force, RFC 2617, June 1999.

[10] J. Kuthan, “Comparison of Service ------+Creation Approaches for SIP,” Int’l. SIP Conf. 2000, Mar. 2000.

[11] M. Handley et al., “SIP: Session Initiation Protocol,” RFC 2543 (obsoleted), Mar. 1999.

[12] J. Rosenberg and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” RFC 2543, June 2002.