PASSWORD BASED SCHEME AND GROUP TESTING FOR DEFENDING DDOS ATTACKS

G Dayanandam[1], T V Rao[2], S Pavan Kumar Reddy[3], and Ravinuthala Sruthi[4]

[1]Department of Computer Science Engineering, QIS Institute of Technology (QISIT), Ongole, Andhra Pradesh, India
gdayanand4u@gmail.com
[2]Department of Computer Science Engineering, KL University (KLU), Guntur, Andhra Pradesh, India
drtvrao@kluniversity.in
[3]Department of Information Technology, QIS Institute of Technology (QISIT), Ongole, Andhra Pradesh, India
coolguypavan39@gmail.com
[4]Department of Computer Science Engineering, QIS Institute of Technology (QISIT), Ongole, Andhra Pradesh, India
sruthiravinuthala@gmail.com

ABSTRACT

DOS ATTACKS ARE ONE OF THE TOP SECURITY PROBLEMS AFFECTING NETWORKS AND DISRUPTING SERVICES TO LEGITIMATE USERS. THE VITAL STEP IN DEALING WITH THIS PROBLEM IS THE NETWORK'S ABILITY TO DETECT SUCH ATTACKS. APPLICATION DDOS ATTACK, WHICH AIMS AT DISRUPTING APPLICATION SERVICE RATHER THAN DEPLETING THE NETWORK RESOURCE. UP TO NOW ALL THE RESEARCHES MADE ON THIS DDOS ATTACKS ONLY CONCENTRATES EITHER ON NETWORK RESOURCES OR ON APPLICATION SERVERS BUT NOT ON BOTH. IN THIS PAPER WE PROPOSED A SOLUTION FOR BOTH THESE PROBLEMS BY AUTHENTICATION METHODS AND GROUP TESTING.

KEYWORDS

GRAPHICAL PASSWORD, DDOS ATTACK, GROUP TESTING, APPLICATION RESOURCE, LEGITIMATE TRAFFIC.

1. INTRODUCTION

The Internet currently connects millions of computers around the world that are running on different software and hardware platforms. Every day, our lives become more dependent on the Internet's services that simplify many daily tasks, and every day new users contribute to the Internet's growth. Maintaining correct operation, availability and security of the Internet services is critically important. Just like this high connectivity enables us to develop useful applications, it also provides means to malicious users to recruit and misuse many computers all over the world for various illegal activities.

One type of those malicious activities is *denial of service.* DoS (denial-ofservice) attacks do not aim to alter data or gain unauthorized access, but instead they aim to cripple applications, servers and whole networks, disrupting legitimate users' communication. The attacker either exploits some vulnerability in a target host or network, or he misuses many compromised machines to send huge traffic to the target. The denial of service effect is created by the attacker's traffic interfering with a target's operation, which makes it crash, hang, reboot or do useless work. DoS attacks can be launched from either a single source or multiple sources. Multiple-source DoS attacks are called *distributed denial-of-service* (DDoS) attacks. DDoS attacks can sometimes

employ up to 100,000 compromised computers to perform a coordinated and widely distributed attack. Automated tools that can be easily used by an amateur to generate those attacks are publicly available with detailed specifications about how to use them.

Both DoS and DDoS are a large threat for online services, but DDoS attacks are more difficult to handle because their traffic can be made highly similar to the legitimate traffic. A disruption of a service has many financial consequences for online business. For instance, if an online bank becomes inaccessible for 2 hours this may mean losing business during the outage but also losing customers, prestige and reliability due to a damaged reputation, over a long time.

DDoS attacks are one of the most threatening computer network security problems as they handicap legitimate network usage and cause substantial damage. The devastating negative impact inspires the network security researcher costing great effort to understand and learn them, together with developing and deploying feasible and effective countermeasures against them. Since quantitative DDoS attacks are seldom obtainable from the real network environment, the critical research step is to set up and run the simulated DDoS attacks. Typically, this step requires substantial configuration effort. As the fundamental but popular way, the practical attack tools, which are obtained from the hacker's side, are still being utilized by the security researchers for their researches. If both attack tools and network traffic been configured appropriately, the simulation would have many similarities with the real DDoS attacks. The eight attack tools and making comparisons between them. Nevertheless, the most significant drawback of the real attack tools implementations is that most of the applied attack tools (i.e. TFN2K) have been developed more than ten years ago. Consequently, most of the used up-to-date network devices and software have embedded corresponding countermeasure mechanisms to identify and against such tool generated attack traffic and cause the experiments failure. In addition, the simulated traffic by the old attack tools lacks of capability to model the current sophisticated attack traffic within the large heterogeneous networks.

Distributed denial-of-service attacks (DDoS) attacks consist of an overwhelming quantity of packets being sent from multiple attack sites to a victim site. These packets arrive in such a high quantity that some key resource at the victim (bandwidth, buffers, and CPU time to compute responses) is quickly exhausted. The victim either crashes or spends so much time handling the attack traffic that it cannot attend to its real work. Thus legitimate clients are deprived of the victim’s service for as long as the attack lasts.

DDOS attacks have adversely affected service to individual machines, major Internet commerce sites, and even core Internet infrastructure services. Occasionally, a very large-scale DDoS attack occurs (usually as the by product of a virus or worm spread), crippling Internet-wide communications for hours. While services are restored as soon as the attack subsides, the incidents still create a significant disturbance to the users and costs victim sites millions of dollars in lost revenue. Furthermore, the Internet is used daily for important communications such as stock trades, financial management and even some infrastructure services. Many of these transactions must be processed in a timely manner and can be seriously delayed by the onset of a DDoS attack. The seriousness of the threat is further increased by the ease on which these attacks are performed. Any unsophisticated user can easily locate and download DDoS tools and engage them to perform successful, large-scale attacks. The attacker runs almost no risk of being caught. All of these characteristics have contributed to a widespread incidence of DDoS attacks ([MVS01] reports more than 12,000 attacks per week).

The first large-scale appearance of distributed denial-of-service (DDoS) attacks occurred in mid-1999. Today, four years later, researchers are still struggling to devise an effective solution to the

DDoS problem. Although many commercial and research defenses have appeared, none of them provide complete protection from the threat. Rather, they detect a small range of attacks that either use malformed packets or create severe disturbances in the network; and they handle those attacks by non-selectively dropping a portion of the traffic destined for the victim. Clearly this strategy relieves the victim from the high-volume attack, but also inflicts damage to legitimate traffic that is erroneously dropped.

There are two DDoS attack features that hinder the design of more effective defenses:

1. DDoS traffic is highly similar to legitimate traffic. The attack usually consists of legitimate packets, generated in high quantity. They blend completely with the small amount of legitimate client traffic, so no differentiation can be made on a packet-by-packet basis. To perform traffic separation, the defense system must group all packets targeting the victim into higher-semantic structures (such as “all traffic exchanged between two IP addresses,” “all HTTP traffic,” “all traffic generated from a given source IP address,” etc.), then keep many statistics on the dynamics of those structures to detect high-volume or anomalous communications. Packets that belong to suspect structures will then be policed, while packets belonging to structures that exhibit legitimate behaviour will be forwarded.

2. DDoS traffic is distributed. Attack streams are generated from numerous attack machines spread all over the Internet and converge only in the proximity of the victim. The defense system must control a large portion of the total attack to alleviate the denial-of-service effect on the victim. This indicates that a system must either be a single-point system located near the victim or a distributed system whose defense nodes cover a significant portion of the Internet.

These two features create contradictory requirements for DDoS defense. In order to perform accurate traffic separation, the defense system requires a lot of resources for record-keeping. Therefore, it can only handle small to moderate traffic volumes. On the other hand, the need to control a large portion of the attack traffic requires placement at points that relay a high traffic volume. Those two requirements can hardly be satisfied at a single deployment point. A majority of DDoS defense systems sacrifice the first goal — traffic separation — to achieve the second goal — control of a large portion of the attack traffic. Those systems are located at or near the victim site, which enables them to detect and control the majority of DDoS attacks, but also places the defense system on the path of high-volume traffic, which impairs its selectiveness.

It proposes a source-end defense system, called D-WARD, located at networks that are hosting some of the attack machines. D-WARD monitors and polices the outgoing traffic from those networks, thus controlling attacks. Placing the defense at the source-end exposes the defense system to low-to-moderate traffic volumes, thus enabling sophisticated profiling and traffic separation. The system thus provides a highly selective response to DDoS attacks, inflicting almost no damage to legitimate traffic. Source-end defense is not a complete answer to DDoS attacks. Clearly, it can control only the traffic from the networks that deploy the proposed defense system. If the critical mass1 of the attack machines is located in unprotected networks, the attacker will still successfully deny service to legitimate clients of the victim. On the other hand, source-end defense provides two very important features:

- It places the response close to the sources, thus relieving shared Internet resources from the attack as soon as possible.
- It provides a selective response, minimizing collateral damage to legitimate traffic.

Both of these features make source-end defense highly attractive for integration with other DDoS defense systems that are placed closer to the victim. As those systems police all incoming traffic to the victim, they can improve their response selectiveness by detecting and forwarding packets already policed by a source-end defense. This approach creates a dedicated channel to the victim for those legitimate clients that deploy a source-end defense. They feel no denial-of service effect and communicate unhindered with the victim.

Denial-of-service (DoS) and distributed-denial-of-service (DDoS) attacks pose a grave danger to Internet operation. They are, in essence, resource overloading attacks. The goal of the attacker is to tie up a chosen key resource at the victim, usually by sending a high volume of seemingly legitimate traffic requesting some service from the victim. The overconsumption of the resource leads to degradation or denial of the victim's service to its legitimate clients.

In the absence of effective defense mechanisms, the denial-of-service effect lasts for the entire duration of the attack (i.e., as long as key resources are being tied with malicious traffic), and vanishes quickly once the attack is aborted. Since machine resources are usually shared among many applications, the DoS effect inflicts significant damage — not only on client transactions with the victim, but on the victim's total operation. The victim experiences a significant slowdown in all applications sharing the targeted resource, and frequently also connectivity disruption.

Both DoS and DDoS attacks are seemingly simple in design and operate without requiring any special skill or resource for their perpetration. The attack tools can be obtained easily online and the attack goal (resource exhaustion) is attained whenever a sufficiently large amount of malicious traffic is generated. The targeted resource dictates the type and contents of attack packets, e.g. exhaustion of CPU resources requires computation-intensive packets such as CGI or authentication requests, while network resources can be exhausted by any high-volume traffic.

The main difference between DoS and DDoS attacks is in scale — DoS attacks use one attack machine (to generate malicious traffic) while DDoS attacks use large numbers of attack machines. The scale difference also invokes differences in operation modes. The large number of attack machines allows DDoS perpetrators acertain recklessness — they frequently trade sophistication for brute force, using simple attack strategies and packet contents to overload victim resources. However, the simplicity in both attack types arises from convenience, not necessity. The lack of effective defense mechanisms, even for simple attacks, offers no motivation for perpetrators to design more sophisticated ones. Once defences successfully counter one attack class (e.g., like ingress filtering [FS00] has countered random IP source spoofing), attackers quickly deploy slight modifications in their attacks to bypass defensive actions.

There are many attack variations and many dimensions in which attacks can still evolve while preserving the ability to inflict damage on the victim. This feature makes it very challenging to design successful defenses. Due to attack variety, defense systems must maintain a volume of statistical data in order to detect attacks and sieve legitimate from attack traffic. This incurs high operation costs. On the other hand, attackers can easily bypass or trick defenses with slight modifications to their attacks. Any such modifications require added complexity in defense mechanisms (in order to handle the new attack class), thus skyrocketing the cost.

2. DIFFERENT WAYS TO DEFEND

There are four different ways to defend against DoS attacks: (1) authentication; (2) attack detection; (3) attack source identification; and (4) attack reaction.

2.1. AUTHENTICATION

Passwords provide security mechanism for authentication and protection services against unwanted access to resources. A graphical password is one promising alternative of textual passwords. According to human psychology, humans are able to remember pictures easily. This approach aims to improve the global security level by introducing a graphical password based schemes for avoiding ddos attacks.

2.2. ATTACK DETECTION

Attack detection aims to detect DoS attacks in the process of an attack. Attack detection is an important procedure to direct any further action. The challenge is how to detect every attack quickly without misclassifying any legitimate traffic. Here we are using dot defender a web application firewall for filtering the legitimate traffic against DDoS threats.

2.3. ATTACK SOURCE IDENTIFICATION

Attack source identification aims to locate the attack sources regardless of the spoofed source IP addresses. It is a crucial step to minimize the attack damage and provide deterrence to potential attackers. The challenge for attack source identification is how to locate attack sources quickly and accurately without changing current Internet infrastructure.

2.4. ATTACK REACTION

Attack reaction aims to eliminate or curtail the effects of an attack. It is the final step in defending against DoS attacks, and therefore determines the overall performance of the defence mechanism. The challenge for attack reaction is how to filter the attack without disturbing legitimate users it is done by using group testing. Our main contributions in this paper are as follows:

- Propose a new graphical password based authentication system.
- Proposing a new security system at server for additional application resource security.
- Propose a defending against network traffic using dot defender software.
- Propose a new size-constrained GT model for practical DoS detection scenarios.

3. AUTHENTICATION THROUGH NEW PASSWORD SYSTEM

In the 1st step we provide authentication to each user. Computer systems and the information they store and process are valuable resources which need to be protected. Current secure systems suffer because they mostly ignore the importance of human factors in security. An ideal security system considers security, reliability, usability, and human factors. A graphical based password is one promising alternatives of textual passwords. According to human psychology, humans are able to remember pictures easily. In this paper, we have proposed a new hybrid graphical password based system, which is a combination of recognition and recall based techniques that offers many advantages over the existing systems and may be more convenient for the user.

Besides application DoS attacks, our defence system is applicable to DoS attacks on other layers, e.g., protocol layer attack-SYN food where victim servers are exhausted by massive half-open connections. Although these attacks occurring different layers and of different styles, the victim machines will gradually run out of service resource and indicate anomaly. Since our mechanism only relies on the feedback of the victims, instead of monitoring the client behaviours or properties, it is promising to tackle these attack types.

3.1. AUTHENTICATION METHOD FOR SECURE PASSWORD

Why we are proposing this authentication method means if the mail account of a person Is opened and he sent a mail to the server at the same time the attacker hacked his password and opened his mail and sent another message to the server at the same time with malicious id then the account or an mail id is blocked by the router so it is not possible for the legitimate user to access server resource so in this paper we proposed a graphical password based system in figure 1 for authentication.

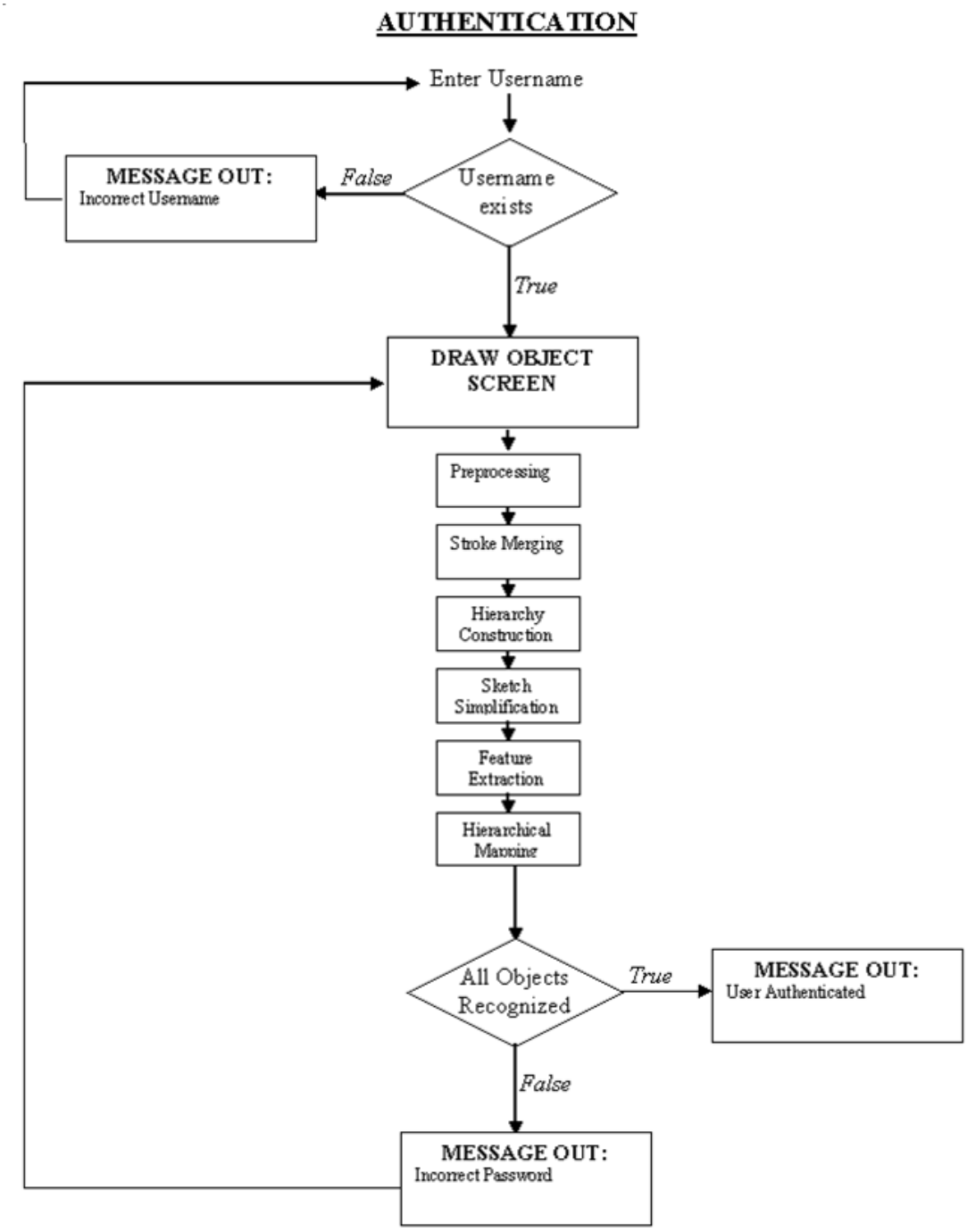

Figure 1. Flow chart for proposed authentication

STEP1: The first step is to type the user name and a textual password which is stored in the database. During authentication the user has to give that specific user name and textual password in order to log in.

STEP2: In this second step objects are displayed to the user and he/she selects minimum of three objects from the set and there is no limit for maximum number of objects. This is done by using

one of the recognition based schemes. The selected objects are then drawn by the user, which are stored in the database with the specific username. Objects may be symbols, characters, auto shapes, simple daily seen objects etc.

STEP3: During authentication, the user draws preselected objects as his password on a touch sensitive screen (or according to the environment) with a mouse or a stylus. This will be done using the pure recall based methods.

STEP4: In this step, the system performs pre-processing.

STEP5: In the fifth step, the system gets the input from the user and merges the strokes in the user drawn sketch.

STEP6: After stroke merging, the system constructs the hierarchy.

STEP7: Seventh step is the sketch simplification

STEP8: In the eighth step three types of features are extracted from the sketch drawn by the user.

STEP9: The last step is called hierarchical matching.

4. ADDITIONAL APPLICATION RESOURCE SECURITY AT SERVER SYSTEM

Beside this authentication method we have to propose another thing that is we define ACTIVE and INACTIVE states for an email account at the server. That is If the mail account of a person is registered in the server and is opened by him at that time in the server his account should be in active state and if another person attacks his ID and want to open it. It sends error message "It is not possible to open". Then by this procedure we can remove fake mails generated by the attackers and it saves application resources.

5. ATTACK DETECTION

Attack Detection is a mechanism which stops the attacks before they actually cause damage. This approach assumes attack traffic is spoofed, which is true in most situations since attackers need spoofed traffic to hide their identities and exploit the protocol vulnerabilities. This approach normally comprises a variety of packet altering schemes, which are deployed at the routers. The packet alters are used to make sure only valid (non-spoofed) traffic can pass through. This greatly reduces the chance of having DDoS attacks. For all the available attack countermeasures, attack prevention is the most preferred approach because it can minimize attack damage. However, it is not easy to specify a altering rule that can differentiate attack traffic from legitimate traffic accurately. Moreover, some types of altering schemes require wide deployment to be Effective. Unfortunately, the Internet is an open community without central administration, which makes prevention a taxing and daunting task. Ingress Filtering is an altering scheme that alters incoming traffic according to a specified rule. We define the customer's network as the network that hosts Internet users of one organization.

With dot Defender web application firewall shown in figure 2, you can avoid DoS attacks because dot Defender inspects your HTTP traffic and checks their packets against rules such as to allow or deny protocols, ports, or IP addresses to stop web applications from being exploited.

Architected as plug & play software, dot Defender provides optimal out-of-the-box protection against DoS threats, cross-site scripting, SQL Injection attacks, path traversal and many other web attack techniques.

The reasons dot Defender offers such a comprehensive solution to your web application security needs are:

- Easy installation on Apache and IIS servers.
- Strong security against known and emerging hacking attacks.
- Best-of-breed predefined security rules for instant protection.
- Interface and API for managing multiple servers with ease.
- Requires no additional hardware, and easily scales with your business.

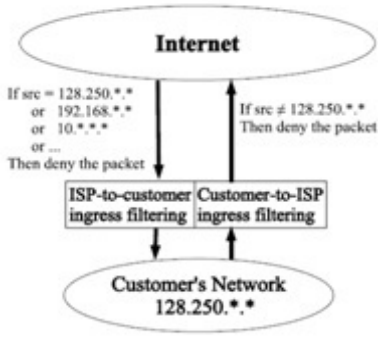

Figure 2. Dot Defender Software Working

So by this firewall technique we can block the port numbers and host IP addresses of the attackers. This reduces the network traffic by allowing only legitimate traffic to the server. By this it filters only legitimate users Host IP addresses and Port numbers.

6. ATTACK SOURCE IDENTIFICATION

The maximum destruction caused by the attacks includes the depletion of the application service resource at the server side, the unavailability of service access to legitimate user, and possible fatal system errors which require rebooting the server for recovery. We assume that any malicious behaviour can be discovered by monitoring the service resource usage, based on dynamic value thresholds over the monitored objects. Data manipulation and system intrusion are out of this scope.

We assume that application interface presented by the servers can be readily discovered and clients communicate with the servers using HTTP/1.1 sessions on TCP connections. We consider a case that each client provides a non spoofed ID (e.g., SYN-cookie) which is utilized to identify the client during our detection period. Despite that the application DoS attack is difficult to be traced, by identifying the Ids of attackers; the router can block the subsequent malicious requests.

The attackers are assumed to launch application service requests either at high inter arrival rate or high workload, or even both. The term "request" refers to either main request or embedded request for HTTP page.

Since the detection scheme proposed will be orthogonal to the session affinity, we do not consider the repeated one-shot attack.

We further assume that the number of attackers d n where n is the total client amount. This arises from the characteristics of this attack. Due to the benefits of virtual servers we employ, this constraint can be relaxed, but we keep it for the theoretical analysis in the current work.

7. ATTACK DETECTION

A detection model based on GT can be assumed that there are t virtual servers and n clients, among which d clients are attackers. Consider the matrix Mtxn specified in figure 3 the clients can be mapped into the columns and virtual servers into rows in M, where M [I, j] =1 if and only if the requests from client j are distributed to virtual server i. With regard to the test outcome column V, we have V[i]=1 if and only if virtual server i has received malicious requests from at least one attacker, but we cannot identify the attackers at once unless this virtual server is handling only one client. Otherwise, if V[i] =0, all the clients assigned to server I are legitimate. The d attackers can then be captured by decoding the test outcome vector V and the matrix M.

$$M = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \overset{\text{testing}}{\Longrightarrow} V = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Figure 3. Binary Matrix M and Test Outcome Vector V

7.1. VICTIM DETECTION MODEL

The victim model in our general framework consists of multiple back-end servers, which can be Web/application servers, database servers, and distributed file systems. We do not take classic multitier Web servers as the model, since our detection scheme is deployed directly on the victim tier and identifies the attacks targeting at the same victim tier; thus, multitier attacks should be separated into several classes to utilize this detection scheme. The victim model along with front-end proxies mentioned below as figure 4.We assume that all the back-end servers provide multiple types of application services to clients using HTTP/1.1 protocol on TCP connections.

Each back-end server is assumed to have the same amount of resource. Moreover, the application services to clients are provided by K virtual private servers (K is an input parameter), which are embedded in the physical back-end server machine and operating in parallel. Each virtual server is assigned with equal amount of static service resources, e.g., CPU, storage, memory, and network bandwidth. The operation of any virtual server will not affect the other virtual servers in the same physical machine. The reasons for utilizing virtual servers are twofold: first, each virtual server can reboot independently, thus is feasible for recovery from possible fatal destruction; second, the state transfer overhead for moving clients among different virtual servers is much smaller than the transfer among physical server machines.

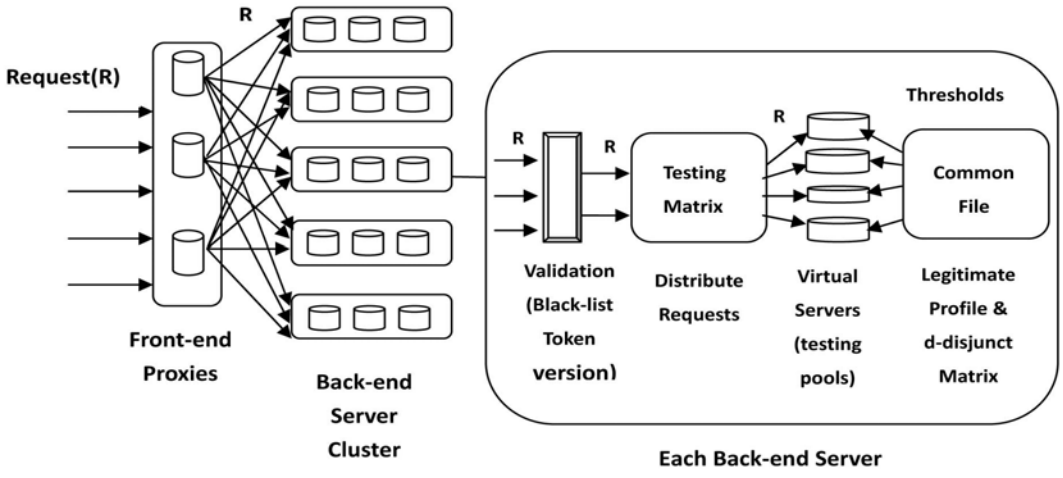

Figure 4. Victim Detection Model

As soon as the client requests arrive at the front-end proxy, they will be distributed to multiple back-end servers for load balancing, whether session sticked or not. Notice that our detection scheme is behind this front-end tier, so the load balancing mechanism is orthogonal to our setting. On being accepted by one physical server, one request will be simply validated based on the list of all identified attacker IDs (black list). If it passes the authentication, it will be distributed to one virtual server within this machine by means of virtual switch. This distribution depends on the testing matrix generated by the detection algorithm. By periodically monitoring the average response time to service requests and comparing it with specific thresholds fetched from a legitimate profile, each virtual server is associated with a "negative" or "positive" outcome. Therefore, a decision over the identities of all clients can be made among all physical servers.

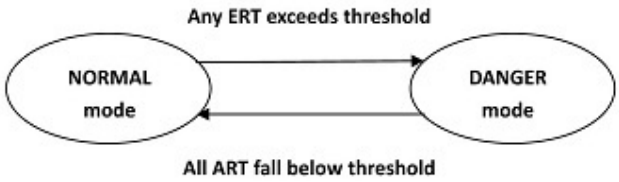

Figure 5. Two-State Diagram of the System

As mentioned in the detection model, each back-end server works as an independent testing domain, where all virtual servers within it serve as testing pools. We discuss about the operations within one back-end server, and it is similar in any other servers. The detection consists of multiple testing rounds.

First, generate and update matrix M for testing in figure 3.

Second, "assign" clients to virtual servers based on M as in figure 4. The back-end server maps each client into one distinct column in M and distributes an encrypted token queue to it. Each token in the token queue corresponds to a 1-entry in the mapped column, i.e., client j receives a token with destination virtual switch. In addition, requests are validated on arriving at the physical servers for faked tokens or identified malice ID[1]. This procedure ensures that all the client

requests are distributed exactly as how the matrix M regulates and prevents any attackers from accessing the virtual servers other than the ones assigned to them.

Third, all the servers are monitored for their service resource usage periodically, specifically, the arriving request aggregate (the total number of incoming requests) and average response time of each virtual server are recorded and compared with some dynamic thresholds to be shown later. All virtual servers are associated with positive or negative outcomes accordingly.

Fourth, decode these outcomes and identify legitimate or malicious Ids. By following the sequence detection with and without packing algorithms all the attackers can be identified within several testing rounds.

To lower the overhead and delay introduced by the mapping and piggybacking for each request, the system is exempted from this procedure in normal service state. The back end server cycles between two states. The two-state diagram of the system is shown in figure 5, which we refer as NORMAL mode and DANGER mode. Once the estimated response time (ERT) of any virtual server exceeds some profile-based threshold; the whole back-end server will transfer to the DANGER mode as shown in figure 6 and execute the detection scheme.

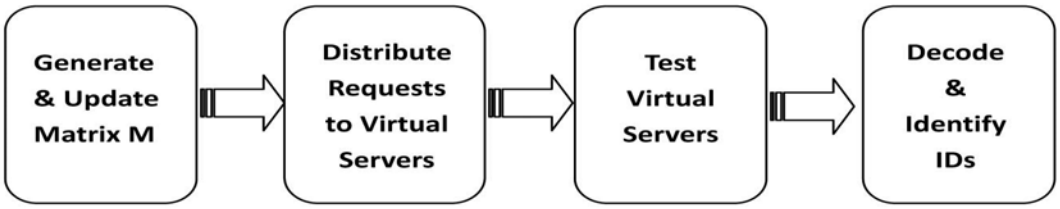

Figure 6. One Testing Round in Danger Mode

Whenever the average response time (ART) of each virtual server falls below the threshold, the physical server returns to NORMAL mode.

ACKNOWLEDGEMENT

In this paper, we developed a solution for both attacks on network resources and on application servers by using new authentication mechanism and group testing. We developed a new hybrid graphical password based system, which is a combination of recognition and recall based techniques that offers many advantages over the existing systems and may be more convenient for the user and by group testing mechanism for locating the attack and for blocking the attack at router itself.

REFERENCES

[1] Ying Xuan, Incheol Shin, My T. Thai, Member, IEEE, and Taieb Znati, Member, IEEE *"Detecting application denial of service attacks a Group-Testing-Based Approach"*

[2] *"Defending against Distributed Denial of Service Attacks"* by Tao Peng.

[3] *"A Graphical password based system for small mobile devices"* byWazir Zada khan, Mohammed Y Aalsalem and Yang Xiang.

[4] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, *"DDos- Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection,"* Proc. IEEE INFOCOM, Apr. 2006.

[5] S. Vries, *"A Corsaire White Paper: Application Denial of Service (DoS) Attacks"*, *http://research.corsaire.com/whitepapers/040405-application-level-dos-attacks.pd f*, 2010.

Authors

Prof.G.Dayanandam B.Tech-CSE, M.Tech-IT, (Ph.D)-CSE, ANU, currently working as Professor & HOD, CSE Dept, QISIT. Having work experience of 12 years, has published two papers in international journal, one paper in National journal and 2 papers in national level conferences. His areas of interest are Cryptography, network security and information security.
Contact: gdayanandam@yahoo.co.in

Dr.T.v.Rao, B.E-ECE, M.E-CS, Ph.D in computer science and engineering, Wayne State University, Detroit, USA, Currently working as Professor in KL University, Vaddeswaram, Guntur Dt. He has more than 32 years of experience and has published many papers in national and international conferences. His areas of interest are multicore and parallel programming
Contact: tv_venkat@yahoo.com

Mr. Suram Pavan Kumar Reddy B.Tech-IT, QISIT, Ongole, India. He is planning to do his Master's in USA. He published an article in International journal of computer and organization trends. His areas of interest are Network Security, Robotics, Web Designing, Ethical Hacking and Networking.
Contact:coolguypavan39@gmail.com

Ms. Ravinuthala Sruthi completed her bachelor's degree in CSE stream from QIS Institute of Technology. She got a Ph.D seat in SASTRA University, Thanjavur. Now going to join in Ph.D programme. Based on the limitations she finds in the project "Detecting application Denial-of service attacks: A Group Testing based approach" she proposed some propositions in this paper. She is interested in the areas of Network security and Web designing.
Contact:sruthiravinuthala@gmail.com