Hindawi
Security and Communication Networks
Volume 2019, Article ID 1751285, 8 pages
https://doi.org/10.1155/2019/1751285

Research Article

An Analysis of DDoS Attacks on the Instant Messengers

Mohammad Faisal[1], **Sohail Abbas**[2], **Haseeb Ur Rahman**[1], **Muhammad Zahid Khan**[1], **and Arif Ur Rahman**[3]

[1]*Department of Computer Science and IT, University of Malakand, KP, Pakistan*
[2]*Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, UAE*
[3]*Department of Computer Science, Bahria University, Islamabad, Pakistan*

Correspondence should be addressed to Mohammad Faisal; mfaisal_1981@yahoo.com

Received 18 December 2018; Revised 4 March 2019; Accepted 20 March 2019; Published 30 October 2019

Academic Editor: Petros Nicopolitidis

Latest technologies of voice over IP (VoIP) and mobile messaging for smartphones messengers such as WhatsApp, Viber, Skype, etc., offer free-of-charge facilities of worldwide SMS, MMS, and voice calls to their users, unlike the traditional and expensive cellular or telephone networks' services. Customers of the formerly mentioned messengers are estimated in millions because of the attractive features offered by them. However, these messengers face many cyber security threats and the required security features are either not available at all or are insufficient for efficiently countering the threats. Professionals working in the domain of cyber security are challenged by the devastating effects of distributed denial of service (DDoS) attacks on all major platforms including Apple Macintosh, Windows, Unix, and Linux. In this paper, we demonstrate the effect of DDoS attack on the performance of an IRC server using a test bed. We use a game theoretic model to analyze the feasibility of DDoS attacks on the IRC platform, keeping in view the attacker's objective. The analysis will help the security experts to propose appropriate countermeasures to reduce the attackers' utility, thereby making it less attractive for those attackers to launch the attack.

1. Introduction

Smartphones have various types of connectivity options such as WiFi, Bluetooth, LTE, and ubiquitous computing features, which make them an important part of our daily lives. People are using smartphones for social networking, online banking and shopping, web browsing, tracking locations such as hotels, restaurants, and hospitals, and mail checking, besides the conventional uses including phone calls and messaging services [1]. They have also become the target of hackers and crackers for carrying out malicious activities. They have become an ideal hub for malware, grayware, and spyware developers who can easily exploit the vulnerabilities and insecure communication channels. When a security service is proposed to counter a particular type of breach, in short period of time, a new attacking mechanism is invented by malicious user(s).

One of the important services used in the smart phones is instant messaging (IM) applications that follow customized versions of the Internet Relay Chat (IRC) protocol [2]. The IRC is a client-server, multiuser, and multichannel chatting application that facilitates users to communicate with their peers in real time on the Internet worldwide. A user runs an IRC client that is connected to a server on the IRC network. Servers then forward the messages to other relay servers on the same network. However, there are many vulnerabilities in the IRC protocol, such as weak authentication, channel privacy, and others, that would ultimately pave the path to denial of service (DoS) or distributed denial of service (DDoS) attacks.

The primary objective of DoS and DDoS attacks is to compromise the availability service of a system, thereby weakening the system to an extent that shall deny the legitimate requests. Till date, the most usual DoS attack strategies reported can be categorized into two classes: logic and flooding attacks. The logic attacks use the existing known vulnerabilities as tool to crash the remote servers or at least disrupt their performance till they are not usable by the legitimate users. The flooding-based DoS attack bombards a large volume of data or control packets (sometimes

intended for computationally intensive tasks) on a destination, which would result in exhausted resources, such as bandwidth, memory, and CPU, at the destination. The latter attack type is considered to be more dangerous because typically there is no trivial way to differentiate between bad packets from the good ones. Early DoS attacks were not that sophisticated, i.e., they used a single node to generate the flooding traffic and sent towards the destination server. However, over time, the attack evolved from single source to multiple sources of attacks, which is called DDoS attack. In DDoS, an attacker compromises and controls innocent nodes, called bots, to be part of the attack. The attacker then launches a large-scale attack on a target server using network of bots, called botnets. Ironically, usually in DDoS attack, the actual attacker is hard to be detected. Considering the variety in DDoS attacks and their diverse classification, a single security mechanism cannot be a panacea for DDoS attacks in all scenarios [1, 3, 4].

The IRC platform has recently been used as a command-and-control for the large-scale DDoS attacks [2, 5, 6] due to its weak security. In this paper, we demonstrate practically the effect of the attack in terms of bandwidth using a testbed. We also develop a game theoretic model for analyzing the feasibility of DDoS attacks on the IRC platform, keeping in view the attacker's objective. This will help the security experts to propose appropriate mechanisms to reduce the attackers' utility, thereby making it less attractive for those attackers to launch the attack.

The rest of the paper is structured as follows: Section 2 presents DDoS attacks on mobile devices (Android, Symbian, Palm, and iPhone) on smartphone platforms. Section 3 presents the basics of IRC messaging system and its vulnerabilities. In Section 4, we present our game theoretic model to analyze and evaluate the effect of DDoS attack on IRC. Section 5 presents the testbed and experimentation of launching the DDoS attack. The paper is concluded in Section 6 while highlighting the future work.

2. Background and Literature Review

Technologies such as GSM (Global System for Mobile Communications), GPRS (General Packet Radio Services), EDGE (Enhanced Data rates for GSM Evolution), UMTS (Universal Mobile Telecommunication System), and Bluetooth make smartphones a tool of connectivity and a door for malwares, graywares, and spywares to infiltrate and carry out malicious activities. GSM is technology of second generation (2G) that enabled the communication between the smartphones and base stations via switching subsystem. The GSM substituted first generation (1G) analog-based services by a digital, full-duplex, and circuit switched network for voice telephony. GPRS is 2.5 generation technology developed to enhance the data rates and reduce the connection access time of the 2G, implementing the packet switching mechanism and introducing WAP (Wireless Access Protocol) and MMS (Multimedia Messaging Services). EDGE was an effort towards improving the GPRS features with enhancement in its data rates and service reliability. UMTS introduced in 2002 achieved the data rate limit of up to 2 Mbps. Both packet and circuit switching connections were supported at the same time. Multiple services could be accessed simultaneously by the user such as streaming, and conversation with interactive backgrounds. Bluetooth, developed in 1999 based on radio transmission short wavelength standard, is widely used for data communication and personal area networks. It can support short range of communication up to 100 meters with minimal cost and consumption [1]. Before going ahead, here we are going to elaborate potential threats for smart phones. Based on the legality, delivery methods, and user authentication, mobile threats can be classified into three main and defense mechanisms [7].

2.1. Malwares. The focus of malware is usually to frustrate or block the legitimate user from the services they have been authorized to or steal their private data, thereby exploiting the device or platform vulnerabilities. Malware includes virus, worm, trojan horse, root kits, and botnets. Virus is a self-replicating piece of code whereas worm is a self-copying program; trojan horse is a friendly-looking software that is appearing to provide services but a malicious program; root kits install trojans after which then may disable firewalls and antiviruses; and botnets are group of virus-infected devices used for organized crime [1, 7, 8].

2.2. Spyware. To trace the location of the node and retrieve its history for a specific span of time are the main objectives of the spyware. Spywares may be legitimate and illegitimate based on the intention of its use. For example, considering the scenario of someone installing a spyware on their child's or spouse's smartphone. In this scenario, the spyware will not deceive the attacker. However, if the spyware is installed without the user's consent and it successfully gains access to the device and it sends confidential information to the intruder rather than the real author, then it is illegitimate [7].

2.3. Grayware. Collecting user information for the purpose of profiling and then sending marketing information back to the user are the main intentions of grayware. However, the objectives of grayware distributor corporations are not to harm users; rather, they provide some sort of functionality and importance to the host user. Users can complain and block the services of a grayware if the data collection process of a grayware is questionable. The illegal use of grayware is punished by fines rather than any personal statements in many developed countries unlike malwares and spywares. That is why grayware is sometimes called at the boundary of legality and illegality. Depending on the privacy policy sanctioned and the user jurisdiction of complaints, grayware companies have to disclose their compilation practices in their terms and conditions [7].

2.4. DDoS Attacks on Smart Phones. Here, we discuss DDoS/botnet attacks on different cell phone or mobile platforms.

2.4.1. iKee.A. Using the Internet as a source, after scrutinizing the vulnerable IP addresses of phones that have enabled the SSH, the intruder can effectively upload a simple application that makes the phone as part of a botnet that can easily effect other phones via self-propagation [9].

2.4.2. iKee.B. As like its predecessor, iKee.B authorizes the boot master to command and control (C&C) the shell of all affected phones; this can pass on the C&C of the affected devices to any new location on the Internet. Moreover, it can also extract all the messages from the phone database [9].

2.4.3. Yxes. Sketch the insight of Yxes being the pioneer malware for Symbian operating system 9 targeting to achieve IMEI and IMSI of cell phones, terminating the superfluous applications. However, for its propagation, it uses phishing techniques by sending an SMS to incoming victim phones possessing a link for downloading malwares from the malicious server [10].

2.4.4. Trojan-SMS.AndroidOS.FakePlayer.b. As it is Trojan in nature, after installation by user permission through manual steps, it sends messages via SMS without the user/owner permission, will, and demand. At the end, the result becomes denying the normal service of messaging when required by the owner [1].

2.4.5. ZeuS MitMo. Exploiting the social engineering techniques, the malware first of all sends an SMS containing the link to download and install the software, which can sabotage all the online banking transactions from the cell phone. After this, not only the initial username and password are stolen successfully, but also the malware will become capable of recertifying the message and authenticating the process from the central server of online banking system. So, the malware can now be able to deny the legitimate users from its own operation or services [11].

2.4.6. Sound Miner. Trojan in nature is capable of extracting confidential data from audio sensors installed in Android devices. Sound miner can sniff important information such as credit card number or personal identification number (PIN) on the basis of both speech and voice communication between the users and the phone system, reporting these data to a central malicious party located remotely somewhere else. During installation, it requests for granting access to the microphone of the cell phone. Now, the microphone will obey the malware commands rather than the owner, thereby denying him from regular sound services [12].

2.4.7. Crafted Shell Code. In many Linux-based cell phones, root privileges can be accessed to activate *ptrace()* to inject code to other processes giving wrong messages such as buffer overflow, denying the usual processes and activities of the running application of its legitimate users [13].

2.4.8. Battery Exhaustion Attack. Approximately 22 times faster than the normal battery utility, this attack consumes battery lifetime and badly affects the routine services. Automatically downloads MMS messages exploiting the Internet connection and insecure MMS protocol. In the same way, it also sends UDP packets to its partner zombies and the hit list of the target devices, denying the services of routine users [14].

2.4.9. Curse of Silence Attack. This works by crafting the Symbian S60 phones silent by barring the smart phones to receive SMS messages and reporting to the user that memory is incapable to receive more messages or delete messages to vacate space for incoming messages [15].

2.5. DDoS Attacks on UMTS. In this section, we discuss the vulnerabilities of UMTS due to which possible DoS attack can be launched [16–19].

2.5.1. Authentication Modification. If the radio network controller (RNC) is continuously getting tampered with messages from the intruders, ultimately the connection between the subscriber and the user is terminated, denying the services for the right user.

2.5.2. Dropping the Acknowledgement Signals. A malevolent node observing the Temporary Mobile Subscriber Identity (TMSI) after which trying to drop all the forthcoming messages to compel the system for creating new TMSIs ultimately will result in DoS for all the incoming subscribers.

2.5.3. Replacing the Vulnerable Radio Resource Control (RRC) Messages. An intruder swapping the acceptance of an association sets up an absolute RRC message with a decline RRC message affecting the service quality and for the subscriber creating a DoS.

2.5.4. Demand for Synchronization. An intruder impersonates for repeated and simultaneous requests to resynchronize the data flow for multiple users; it will exaggerate the Home Resource Register and create the DoS for all subscribers.

2.6. DoS Attacks on VOIP and SIP

2.6.1. Deregistration Attack. Highlighting the deregistration attack in SIP by [20], the intruder enforces the legitimate caller either to divert his call to a third party (probably the attacker party) or to re-register again and again with the SIP server/proxy denying the regular services. The attack is powerful enough that it can even bypass authentication procedure.

2.6.2. DoS Attack with Unidentified URLs. Zhang et al. [21] illustrate DoS attack with unnamed and unidentified URLs (uniform resource locator) whose DNS (Domain Name

System) in reality does not exist. They make the server busy in nonsignificant activities and push the legal requests to wait, and especially the attack is effective in synchronous DNS resolution. It is verified experimentally by Zhang et al. that even a well-established synchronous-resolution server can drop calls exceedingly by as few as 1000 messages per second else a single threaded server can even be made shut down and drop calls by one message per second.

2.6.3. Semantic Level Attack. Conner and Nahrstedt [22] explain an attack in which the user itself automatically calls nonresponding callers in his/her contact list impersonating itself as a legitimate owner, ultimately draining the resources and denying the services from valuable users.

2.6.4. Billing Inconsistencies Attack. Envisage the SIP vulnerability of recording bills in various modes in contrast to VOIP. The impostors primarily attack and focus on to create inconsistencies in billing system, which can either increase the tariff grand amount or to decrease the real consumption payment. Few of these are managed by the man-in-the-middle attack, while some require prior interaction with the victim [23].

3. IRC Messenger

A swift boost in online communication in the recent decades is due to the various social media-based applications and messengers whereby users can share audio, video, and images with each other very easily, almost free of charge, around the world. Although the connection is round the clock with known and unknown users but their privacy is at high risk, and both sender and receivers are vulnerable to security threats [24]. Their security models with respect to their authentication schemes are studied. All of them use their personal cell number or SIM (subscriber identification module) for authentication purposes only, and they are using Internet infrastructure rather than cell phone telecommunication networks for intercommunication and intracommunication. Cell number is entered during the installation phase of the messenger although Android can detect cell number from the phone automatically, while Apple for security reasons makes it manual mandatory. Even devices such as tablets, which are without phone module in architecture, can be activated by exploiting phone numbers of other device through WiFi [24].

3.1. Why IRC Messengers? Few of the vulnerability features of IRC messenger are send and receive text, videos, audio, images, group chats, and sharing cards and contact information. The latest smart phone messengers are the best option for DDoS attack because of its cross-platform (Android, BlackBerry, Symbian, and iPhone) compatibility. No-log-out facility is always available, and once logged in, the user is always connected and online.

3.1.1. Modifying Status Messages. IRC messengers suffer from a vulnerability in which every user can set and reset status that is shared with all users in his/her address book. This sort of facility can create various possible threats. An attacker can change user's status according to his/her own wish although the attacker is unauthorized to do so by sending just an HTTPs request to server containing the user cell number. An attacker can save a number in address book even without a confirmation. Attacker can access all subscribers' status messages for modification.

3.2. IRC Messenger Vulnerabilities. Various security issues such as account hijacking, senders ID spoofing, SMS flooding, enumeration, etc. are open for malicious users to launch DDoS attacks [25–27] presented in Table 1.

3.3. Account Hijacking. During installation of all smartphone messengers, cell/SIM number of the device is used for account establishment. None of the messenger is allowed to enter it automatically. A confirmation message with a PIN code is sent on that specified cell number. User enters that PIN code in the application interface during the installation process for verification purposes. Multiple opportunities are created for hackers that can hijack the account, such as: first, impersonation by hackers to misuse someone cell number. Second, Android can detect the cell number automatically.

3.4. ID Spoofing. Communication between IRC messengers' user identities can be spoofed easily as the users cannot be logout; so, the malicious users can misuse the IRC ID of the real users to send spoof messages. Mistrust among the users can be created, which is very harmful in business context. In such a scenario, eavesdropping is also inevitable.

3.5. SMS Flooding. Sometimes, the messengers generate unwanted messages to their subscribers through their cell phone numbers. In this scenario, the malicious users can generate messages of its own type and can broadcast to the users like business promotion messages without confusing its real identity. Hence, scenarios such as SMS flooding can be occurring. Similarly, replay attacks can be launched by the malicious users that can copy the original messenger message and text it again and again to the IRC users and generate DDoS attack by SMS flooding.

3.6. Enumeration. IRC messenger exchanges the user's contacts with its main server by uploading its own contact list to the server in return to serve and provide the list of all users using IRC messenger along with their details/biodata given, which can be added to the friend of the users contact list automatically. The attacker can get multiple benefits such as information about users: device platform/operating system, cell phone number, geographic location, ethnic origin if given, and picture for confirmation. All these biodata can lead the attacker to launch DDoS attack easily exploiting the vulnerability of any information provided during the enumeration process.

Table 1: Attacks on IRC messenger.

S. No.	Attack name	Vulnerability	Effects
1	Account hijacking	Account authentication impersonation. Android can detect SIM number automatically.	Masquerading the scenario
2	IRC ID spoofing	IRC user account cannot be logged out	Mistrust among IRC messenger
3	SMS flooding	Unrequested SMS	Reduce IRC reliability
4	Enumeration	User's data exchange	User's privacy badly affected
5	Message modification	PIN code message changes/insertion	Spamming attack

3.7. Message Modification. May be a design error or an opportunity/vulnerability for the attacker, PIN code message of IRC messenger can be modified, simply by sending an HTTPs verification request to the server. This vulnerability is exploited by the hackers for launching spam attack.

In our study of the IRC server (RFC [27]) and literature review, we found multiple vulnerabilities among which the shortlisted vulnerability that we used as an evaluation parameter for our proposed DDoS attack is the message sequence format.

4. Analysis of the DDoS Attack

In this section, we model using game theory (GT) [28] the attacker's feasibility of launching a DDoS attack on the IM server. In the GT context, this is called the attacker's utility. After analyzing the attacker's benefit or utility gained from an attack, the security experts can thwart them by employing strategies to prevent these attacks to happen, thereby reducing the benefits of the attack, so that the attack will become not a lucrative option for the attackers.

4.1. Attacker's Utility. In the GT, a scenario is modelled as a game of n rational players. After each outcome of a game, the players expect a utility and through the game, the players try to devise strategies that give them the highest possible expected payoffs or utilities. Since players are rational, they will take actions after carrying out complete cost-benefit analysis. In this GT modelling, we will not model any specific game, such as zero-sum or prisoners' dilemma, but will follow basic GT to analyze attackers' utilities [29].

Let E represent a set of entities or nodes participating in a particular IM system, maintaining a set of identities I. Let S_e be the set of action profile that an entity $e \in E$ can carry out, called *strategies*. An entity must decide on a unique strategy selected from his knowledge base in order to achieve certain goal. An example strategy for an attacker would be to launch an attack and compromise nodes for some benefits based on whatever resources available, such as malwares and stolen information. However, strategy for benign nodes would be to defend against the malicious attacks using the available resources, such as antivirus, intrusion detection systems, and other methods of information protection. Since there are multiple entities participating in IM system, there is a set of outcomes for $n = |E|$ entities

$$O = S_{e_1} \times S_{e_2} \times S_{e_3} \times \ldots \times S_{e_n}. \quad (1)$$

The combination of the strategies of participating entities completely defines an *outcome*. An outcome $o \in O$ is a selection of one strategy from each of these n sets, that is, o is tuple $(s_{e_1}, s_{e_2}, \ldots, s_{e_n})$ representing the strategy taken by all entities.

Each entity's preferences are denoted by exploiting a utility function that maps outcomes to a utility score. The utility of an outcome o to an entity e is the sum of a *benefit utility* $B_e(o)$ and a *cost utility* $C_e(o)$ determined by payments made by e in outcome o:

$$U_e(o) \equiv B_e(o) + C_e(o). \quad (2)$$

4.2. Attacker's Objective. Let us assume that an attacker m is trying to launch a DDoS attack and let $\rho_q \in S_m$ represent the strategy of compromising q nodes—and doing whatever else is necessary in order to reach this objective.

Let A be the objective (such as launching a successful DDoS on the IM server) that an attacker is trying to achieve. We define an *objective success count* operator $\Phi(o)$, which gives the number of successes by m in the outcome o. For example, one set of objectives is to compromise as many nodes as possible in order to create a botnet. If the number of nodes being compromised increases, so is the value of Φ increased accordingly.

We assume that the attacking entity $m \in E$ values attacks linearly, with the success of a single attack valued at v, so that

$$B_m(o) = v\Phi(o). \quad (3)$$

In general, an attacker's expected utility from launching a successful attack using q compromised nodes is

$$E\left[B_m \mid s_m = \rho_q\right] = \sum_{o \in O} v\Phi(o)\Pr\left[o \mid \rho_q\right]. \quad (4)$$

Now, we model in order to determine the real benefit of an attacker. By real, we mean when benefits gained from an attack exceed its cost—a point we call as attacker's valuation. It is important because when the attacker's valuation of an attack surpasses the cost valuation, it is in the attacker's best interests to launch the attack. The other side of the coin is if the security experts somehow make it reverse, then there will be no benefit for the attackers to launch the attack.

We denote the valuation of a DDoS attack (for instance) to be an objective a by ξ_a, defined as follows:

$$\xi_a \equiv \min_q \frac{q}{E\left[\Phi_a \mid \rho_q\right]}, \quad (5)$$

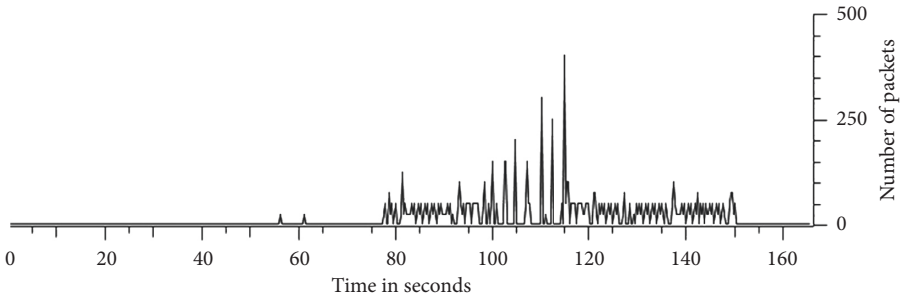

Figure 1: Traffic before the attack.

where $E[\Phi_a \mid \rho_q]$ gives the expected number of successes for an attacker with the objective a launching a DDoS attack compromising or infecting q nodes for its botnet creation.

A security expert, using this measure, may figure out how intrinsically resistant a server is to the DDoS attacks. He/she may independently evaluate and analyze the designed server and the resources installed on it and design strategies to reduce the attacker's utility, thereby making it hard for the attacker to reach such an objective. First, we show that an attacker m only benefits from an attack when their objective valuation is at least ξ_a times their per-compromised machine cost. The attacker's expected utility for the attack with q compromised nodes must be nonnegative for the attack to be rational. So, an attack is rational if and only if

$$\begin{aligned} E\left[B_m \mid \rho_q\right] - q * c \geq 0, \\ v\, E\left[\Phi_a \mid \rho_q\right] \geq q * c, \end{aligned} \tag{6}$$

where c is cost incurred per-compromised node. Since the attacker is rational, he/she would try to optimize (i.e. to minimize) the number of compromised nodes q because it is the related to the cost c. Therefore, according to equation (5) and (6),

$$\begin{aligned} v &\geq \min_q \frac{q * c}{E\left[\Phi_a \mid \rho_q\right]}, \\ v &\geq \xi_a * c. \end{aligned} \tag{7}$$

It is evident from equation (7) that the valuation of an attack is directly proportional to the cost c. To counteract DDoS attacks, it is important for the security experts to devise strategies and design protocols in order to increase the cost c (which is the cost to compromise a single machine as a launching pad for the attack) for the attacker, so that the attack will become not beneficial any more: the induced cost exceeds the benefits gained.

Now, it is important to analyze the number of attempts in creating a botnet of size b in a subregion S of IM network. For a successful DDoS attack, suppose that the attacker requires to compromise b nodes that is the limit where the server may not take on more connections simultaneously. The b nodes may then bombard the targeted server with high volume of traffic in order to overload it and ultimately cripple the server. The probability of attacking a random node is $p = b/S$. Since this is an independent event, we can define the probability of success in t attempts or less by the cumulative distribution function of geometric distribution as:

$$P_{\text{compromise}}(b, t) = cdf_{\text{compromise}}\left(\frac{b}{S}, t\right) = 1 - \left(1 - \frac{b}{S}\right)^t. \tag{8}$$

A very simple case where an attacker creates a botnet of one node, equation (8) defines the success probability of DDoS attack, after t attempts, targeting a single instance of a server and compromising a single node for bot b out of S population.

In order to create a botnet of size r, the attacker needs to compromise r nodes. Since compromising r nodes is an independent event, the probability of compromising r nodes after k attempts should be as follows:

$$P_{\text{compromise}}(b, t, r) = \left(1 - \left(1 - \frac{b}{S}\right)^t\right). \tag{9}$$

As shown in equation (9), the probability of success on compromising r nodes to construct bot b grows exponentially.

5. DDoS Attack Testbed

In this section, we will demonstrate how the performance of IM server degrades after launching DDoS attack against it. We implement an abstract IM server where the DNS is hosted and that is targeted by DDoS attack, whereas the clients are using a JavaScript code to create bots on the server and then keep it busy forever with the help of indefinite loop embedded in the script. The wire shark tool installed on the server is observing the system status and performance before and after the attack.

To evaluate the multiple requests of home page of a domain name system, here we use JavaScript codes to create the bots and the DNS/website as a platform and wire shark tool for collection of results. RFC: rfc2813 Internet Relay Chat: Server Protocol [26].

Observation 1. The rfc2813 Internet Relay Chat Server Protocol [26] suffers from various vulnerabilities. But, one of them is that intruder can exploit the vulnerability of the multiple requests of home page of the same domain name

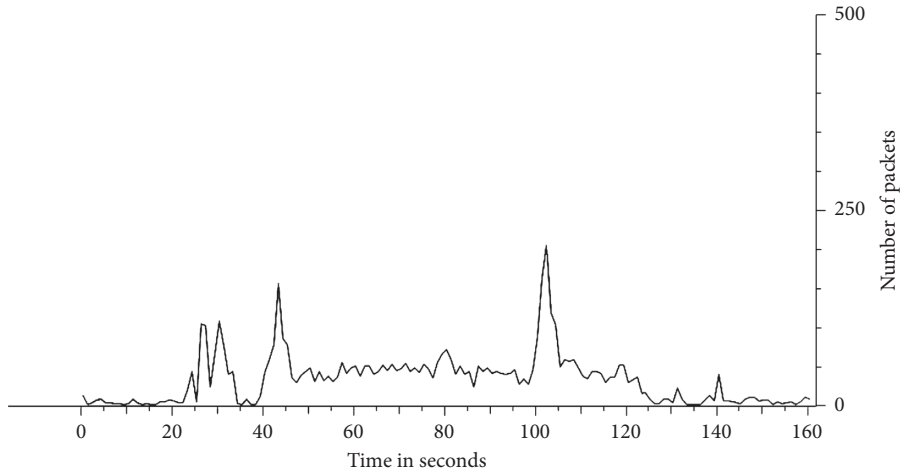

Figure 2: The server operations after the attack.

from different client systems. In this scenario, a different client first sends the script to the server, which creates a bot and keeps the server busy for a long time. The following observations has been detected before and after the DDoS attack as shown in Figures 1 and 2.

In Figure 1, it is observed that after flooding the server by different bots, the number of packets received and processed by the server decreases until the server is completely down.

Observation 2. Exploiting the vulnerability of the multiple requests of home page in various domains, the following observations have been detected before and after the DDoS attack. This time we used the Android platform. As shown in Figure 2, the server goes down around 160 seconds after flooding it by launching DDoS attack.

6. Conclusion and Future Work

In this paper, we discussed the DDoS attacks in general and in the IRC messenger context. We presented a simple scenario of DDoS attack on a server using a real testbed just to realize the effect of the attack on the server resources. We developed a game theoretic model in order to assess the feasibility of the DDoS attack on the IRC platform and also to analyze the attacker's utility. We believe that this will help the security experts to devise appropriate countermeasures to reduce the attackers' utility, thereby making it less attractive for those attackers to launch the DDoS attack.

In our future work, we intend to identify different criteria that affect the attacker's utility and propose countermeasures that make the attack less attractive for the attackers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. L. Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[2] S. Racine, "Analysis of internet relay chat usage by DDoS zombies," Master's thesis, Swiss Federal Institute of Technology Zurich, Zürich, Switzerland, 2004.

[3] I. Ali, M. Faisal, and S. Abbas, "A survey on lightweight authentication schemes in vertical handoff," *International Journal of Cooperative Information Systems*, vol. 26, no. 1, Article ID 1630001, 2017.

[4] S. Abbas, M. Faisal, H. Ur Rahman, M. Zahid Khan, M. Merabti, and A. u. R. Khan, "Masquerading attacks detection in mobile ad hoc networks," *IEEE Access*, vol. 6, pp. 55013–55025, 2018.

[5] D. Goel and A. K. Jain, "Mobile phishing attacks and defence mechanisms: state of art and open research challenges," *Computers & Security*, vol. 73, pp. 519–544, 2018.

[6] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS attacks: trends and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015.

[7] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 3–14, Chicago, Illinois, USA, October 2011.

[8] M. Faisal, S. Abbas, and H. U. Rahman, "Identity attack detection system for 802.11-based ad hoc networks," *EURASIP Journal on Wireless Communications Networking*, vol. 2018, no. 1, p. 128, 2018.

[9] P. Porras, H. Saidi, and V. Yegneswaran, "An analysis of the iKee.B iPhone Botnet," in *Proceedings of the MobiSec 2010: Security and Privacy in Mobile Information and Communication System*, London, UK, 2010.

[10] A. Apvrille, "Symbian worm Yxes: towards mobile botnets?," *Journal in Computer Virology*, vol. 8, no. 4, pp. 117–131, 2012.

[11] D. Barroso, *ZeuS Mitmo: Man-In-The-Mobile*, 2010, http://securityblog.s21sec.com/2010/09/zeus-mitmo-man-in-mobile-i.html.

[12] R. Schlegel, K. Zhang, X.-y. Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: a stealthy and context-aware sound trojan for smartphones," in *Proceedings of the 18th Annual Network & Distributed System Security Symposium (NDSS '11)*, vol. 11, pp. 17–33, 2011.

[13] L. Liu, X. Zhang, G. Yan, and S. Chen, "Exploitation and threat analysis of open mobile devices," in *Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, pp. 20–29, Princeton, NJ, USA, October 2009.

[14] R. Racic, D. Ma, and H. Chen, "Exploiting MMS vulnerabilities to stealthily exhaust mobile phone's battery," in *Proceedings of the 2006 SecureComm and Workshops*, pp. 1–10, Baltimore, MD, USA, September 2006.

[15] I. A. Iosif, *Mobile Phone Security and Forensics: A Practical Approach*, Springer, Berlin, Germany, 2018.

[16] G. Kambourakis, C. Kolias, S. Gritzalis, and J. H. Park, "DoS attacks exploiting signaling in UMTS and IMS," *Computer Communications*, vol. 34, no. 3, pp. 226–235, 2011.

[17] A. Bose and K. G. Shin, "Proactive security for mobile messaging networks," in *Proceedings of the 5th ACM workshop on Wireless security (WiSe'06)*, Los Angeles, CA, USA, September 2006.

[18] L. Xie, X. Zhang, A. Chaugule, T. Jaeger, and S. Zhu, "Designing system-level defenses against cellphone malware," in *Proceedings of the 28th IEEE International Symposium on Reliable Distributed Systems*, pp. 83–90, 2009.

[19] A. Bose and K. G. Shin, "On mobile viruses exploiting messaging and bluetooth services," in *Proceedings of the SecureComm and Workshops*, vol. 2006, pp. 1–10, London, UK, 2006.

[20] A. Bremler-Barr, R. Halachmi-Bekel, and J. Kangasharju, "Unregister attacks in SIP," in *Proceedings of the 2nd IEEE Workshop on Secure Network Protocols*, pp. 32–37, Santa Barbara, CA, USA, November 2006.

[21] T. Jung, S. Martin, M. Nassar, D. Ernst, and G. Leduc, "Outbound SPIT filter with optimal performance guarantees," *Computer Networks*, vol. 57, no. 7, pp. 1630–1643, 2013.

[22] W. Conner and K. Nahrstedt, "Protecting SIP proxy servers from ringing-based denial-of-service attacks," in *Proceedings of the Tenth IEEE International Symposium on Multimedia*, pp. 340–347, Berkeley, CA, USA, December 2008.

[23] R. Zhang, X. Wang, X. Yang, and X. Jiang, "Billing attacks on SIP-based VoIP systems," in *Proceedings of the first USENIX workshop on Offensive Technologies Article No. 4*, vol. 7, pp. 1–8, Boston, MA, USA, August 2007.

[24] A. Mahajan, M. Dahiya, and H. Sanghvi, "Forensic analysis of instant messenger applications on android devices," *International Journal of Computer Applications*, vol. 68, no. 8, pp. 38–44, 2013.

[25] S. Schrittwieser, "Guess who's texting you? Evaluating the security of smartphone messaging applications," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, San Diego, CA, USA, 2012.

[26] Network Working Group OSPF Version 2: https://www.ietf.org/standards/rfcs/.

[27] Network Working Group IETF RFCs: https://www.ietf.org/standards/rfcs/.

[28] R. B. Myerson, *Game Theory*, Harvard University Press, Cambridge, MA, USA, 2013.

[29] N. B. Margolin and B. N. Levine, "Quantifying resistance to the Sybil attack," in *International Conference on Financial Cryptography and Data Security*, pp. 1–15, Springer, Berlin, Germany, 2008.