Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-019-01396-x

ORIGINAL RESEARCH

Distributed denial of service attack defence simulation based on honeynet technology

Xiaoying Wang[1] · Na Guo[1] · Fangping Gao[1] · Jilin Feng[1]

Received: 4 May 2019 / Accepted: 6 July 2019

Abstract
Distributed denial of service (DDoS) is one of the main threats of Internet security, and the detection and prevention of DDoS has always been a hot issue in network security research. DDoS detection and defence systems have many shortcomings such as high false positive rate, low execution efficiency, and lack of linkage between detection and defence. Therefore, eliminating false positives, improving execution efficiency, and enhancing the linkage between detection and defence processes have always been the focuses of research. A preventive defence mechanism based on honeynet technology in the paper is presented without more additional equipment which does not rely on resource advantages, and is equally effective with less effort. Firstly, the in-depth analysis and discussion of detection and defence problems are illustrated by combining with the principle and characteristics of the attack, and systematically analyzing and classifing the detection and defence problems. Next, a distributed denial of service attack defence based on honeynet technology is proposed. Finally, algorithm and the effectiveness of the method are proved by simulation experiments.

Keywords Distributed denial of service · Attack defence simulation · Honeynet technology

1 Introduction

The increasingly specialized testimony of Internet crime (Stalans and Finn 2016) is the distributed denial of service attack (Somani et al. 2016; Saied et al. 2016). Distributed denial of service is an attack against a computer system or network (Khan et al. 2016), which can result in loss of service to users, such as consuming the bandwidth of the victim network or overloading the computer resources of the compromised system. In addition, if DDoS causes a large number of packets per second, the resources on the path will also be exhausted. It is easy to implement DDoS attacks in remote networks by using the tools of the attack, and most of them implement TCP SYN (Mohammadi et al. 2017) and UDP (Kuang et al. 2016) flood attacks. For the connection-oriented Internet TCP protocol (Gomez et al. 2017), the most common method is the TCP SYN flood attack. Because it generates a large number of "half open" (Wen et al. 2017) TCP connections on the target host, which consumes a lot of host resources and make the host no longer accept new connections. For the connectionless UDP protocol, a large number of packets are overloaded with the target host that exhauste network bandwidth and other computer resources to form UDP flooding (Xin et al. 2016). In fact, DDoS attacks are not limited to Web servers. It's possible that any available services on the Internet can be the target of such attacks. Advanced protocols can be used to more effectively increase the load's attack through the use of features, such as running a query that can exhaust resources for an electronic bulletin board or running a recursive HTTP flood attack on a victim site (Prasad et al. 2017). Recursive HTTP flooding refers to the bots starting with a given HTTP link and then recursively accesses all the links on the specified website, also called crawler downloads.

Distributed denial of service attack (DDoS) is an attack that is derived from denial of service attack DoS (Osanaiye et al. 2016) and uses distributed network resources on the Internet to destroy network availability. DDoS prevents the victim system from providing normal services through various means. Initially, because the Dos attack was initiated by a single machine, which was not destructive, so it did not attract enough attention. However, with the development of the Internet and the continuous enrichment of network

✉ Fangping Gao
iccasm@163.com

[1] School of Information Engineering, Institute of Disaster Prevention, Sanhe 065201, Hebei, China

Published online: 13 July 2019

resources, DDoS began to use the distributed attack aircraft group to continuously enlarge the power of DoS attacks. Since the advent of DDoS and the first great power appeared in 1999, DDoS has been used by hackers and has attracted widespread attention from the international community. The initiation of a DDoS attack generally begins with a botnet (Botnet) (Stone-Gross et al. 2011; Anagnostopoulos et al. 2016). Hackers generally use one or more means of communication to infect a large number of hosts with the Bot program, and then use these infected hosts to form a network that can be controlled one-to-many. Botnet contains hundreds of zombie hosts and tens of thousands of zombie hosts, which enables a well-designed DDoS attack to aggregate more than tens of Gbps of attack traffic. Its attack traffic is sufficient to flood any server of bandwidth, such a powerful attack scale cannot be tolerated by any victim host and victim network, and its threat to security can be seen. Honeynet technology requires system security and high controllability to prevent honeynets from being controlled and utilized by hackers. The core requirements of Honeynet technology include: data control, data capture and data analysis. Data control makes it impossible for an attacker to attack other systems outside the honeynet through the compromised system, ensuring that the deployment of the honeynet does not cause more real systems to be attacked, and reducing the risk of honeynet deployment is a prerequisite for system security. Data capture captures and records hacker attacks and processes as much as possible without the hacker's knowledge. Data analysis can use the captured data to analyze the hacker's attack process and the methods and techniques used (Taylor 2019; Hassan et al. 2018; Sombolestan et al. 2018).

To study the detection and defence of attacks (Sharma et al. 2013), we must first clarify the dynamics of development, grasp the evolution process, and understand its characteristics and new developments trends, so that we can know ourselves and know each other. Therefore, t the principle of DDoS in the paper is begun to conduct an in-depth analysis of the defence problem, a preventive defence mechanism based on honeynet technology is given that does not rely on resource advantages and does not require more additional equipment. The effort is equally effective. Firstly, the in-depth analysis and discussion of detection and defence problems is illustrated by combining with the principle and characteristics of the attack,and systematically analyzing and classifing the detection and defence problems. Then, a distributed denial of service attack defence based on honeynet technology is proposed. Finally, the algorithm and the effectiveness of the method are proved by simulation experiments.

2 Honeynet system

2.1 Honeynet system design ideas

To study honeynet technology, we must first understand the details of honeypot technology. Honeypot technology is the foundation of honeynet technology. Lance Spitzner, who is founder of the honeynet project team, defines the honeypot that a honeypot is a safe resource whose value is to be scanned, attacked, and compromised, and to monitor, detect, and analyse these activities (Cross et al. 2017; Dou et al. 2017). Figure 1 shows the structure of a honeynet system.

The definition indicates that the honeypot is a trapping system. From a practical point of view, a honeypot is a computer that does not make any security precautions and is connected to the network, but it is different from a normal computer and internally runs a data logging program and a special-purpose self-exposure program. From the attacker's point of view, the resources are ostensibly the host of its search. The original intention of honeypots is to allow hackers to collect evidence while hiding the real server address, so a qualified honeynet is required to have these functions that to detect attacks, generate warnings, record, deceive, and assist in investigations. Another feature is done by the administrator to sue intruders based on evidence collected by Honeynet when necessary (Yang and Mi 2011; Du et al. 2013). The solution implements the entire solution on a

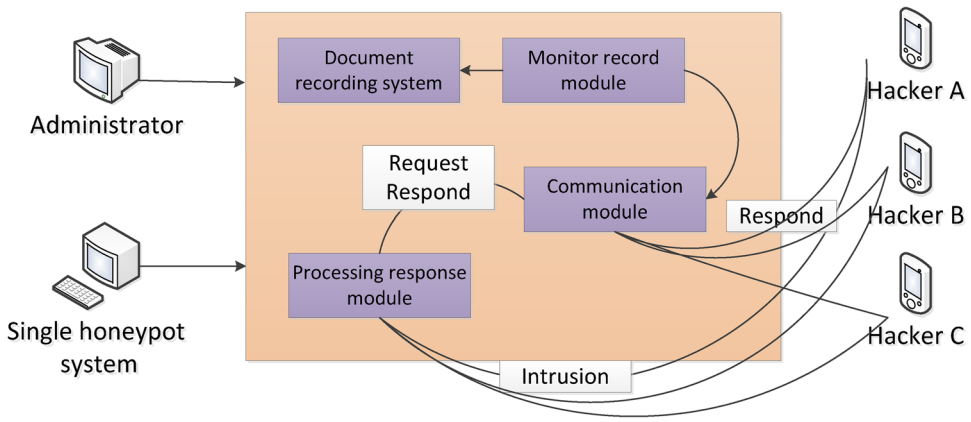

Fig. 1 Schematic diagram of a single honeypot system

single machine. It installs the various components of the honeynet on a single machine, which is easier and cheaper to deploy, lower in cost, and more recoverable. The disadvantage is that the number of physical honeypot hosts that can be deployed is limited and the scalability is not high. Only a small-scale honey network can be deployed, and large-scale attacks cannot be tolerated.

Honey nets can be divided into product-type honey nets and research-type honey nets according to their deployment purposes. The purpose of product-type honey nets is to provide security protection for an organization's network, which includes detecting attacks, prevents attacks from causing damage, and helpes administrators to attack timely and correct responses. The general product type honeynet is easier to deploy and does not require a lot of work by the administrator. Research of honey nets are designed to capture and analyse attackers' operations. On the basis of deploying a research honey net, tracking and analysing attackers' attacks can capture the attacker's keystroke records and understand the attackers and attack methods used by the attackers. Research-based honeypots require researchers to invest a lot of time and effort in the process of attacking monitoring and analysis.

The honeynet can also be divided into a low-interaction honeynet and a high-interaction honeynet according to the level of its interaction degree. The degree of interaction reflects the freedom of the attacker to conduct attacks on the honeynet (Gao et al. 2017a, b). Generally speaking, the low-interaction honey network only simulates the operating system and network services, and is easier to deploy and less risky, such as product-type honey nets generally belong to low-interaction honey nets. However, the attacker can have more limited attacks in the low-interaction honey network, so the information can be collected through the low-interaction honey network is compared and limited, because the low-interaction honeynet is usually a simulated virtual honey net, there are more or less fingerprints that are easily recognized by the attacker, The high-interaction honey network completely provides real operating system and network services without any simulation, such as research of honey nets generally belong to high-interaction honey nets. So many attacker attacks can be obtained in the high-interaction honey network. The high-interaction honey network naturally increases the complexity and risk of deployment and maintenance while improving the freedom of attacker activity.

The above describes the technology and classification of the honey net. The effect can be achieved with a single honeynet that is limited. Honeynet technology can only monitor and analyse the attack behaviour against the honey net, and its view is limited, which does not monitor the entire network through intrusion detection systems such as bypass detection. Honeynet technology cannot directly protect vulnerable information systems. At the same time, the honey network by deployed will bring certain security risks. These risks mainly include the fact that the honeynet may be identified by the attacker and the attacker uses the honeynet as a springboard to attack the third party (Yang and Mi 2011; Tapaswi et al. 2014), which is related to the designment of the honeynet target violation.

In view of some of the above the honeynet problems, the honeynet technology has also been proposed. Honeynet technology is a new technology by the Honeynet project team proposed and advocated to deeply analyse various attacks, which builds a highly controllable network structure and provides a series of related tools to control, capture and control network attacks analysis. The honeynet is actually a high-interaction honeynet system that uses the real system—, and the hardware device interacts with the display network. The honeynet is a trapping network that contains several deployed honey nets, which are configured with real systems and applications, is artificially vulnerable to attack and can be captured, monitored and controlled as much as possible all incoming and outgoing network packets.

2.2 The third generation honey network layout structure

The honeynet system for designing botnet DDoS attacks in the paper is based on the third generation honeynet technology. The third generation of honeynet technology has added a lot of core functions to contrast with the previous honeynet. At this time, the biggest improvements are automated updates, data analysis and GUI management, extensive improvements in hardware and international support.

The structure of the third generation honeynet is roughly express the topology diagram that as shown in Fig. 2, which contains several parts of the honey wall, the internal network, and the honeynet network. The honey wall is the device that acts as a gateway in the honeynet, which is the necessary level for all data entering and exiting the honeynet. Therefore, the honey wall is the key to design a honeynet that separates the honeynet from the external network and controls the entire honeynet central hub.

In the honeynet system, the real trapping function is the honeynet internal network, which contains multiple honeynet hosts and connects to the external network through the honey wall. The honey wall is actually a host or server with various services installed. There are three network interfaces on the honey wall, in which eth0 connects the switch to the external network and eth1 connects to the honeynet internal honey network. BR0 is a virtual bridge interface (eth0 + eth1) whose interface is bridged, does not provide an IP address and NIC MAC address, and does not perform TTL decrement and network routing on forwarded network packets. Therefore, the existence of the honey wall does not

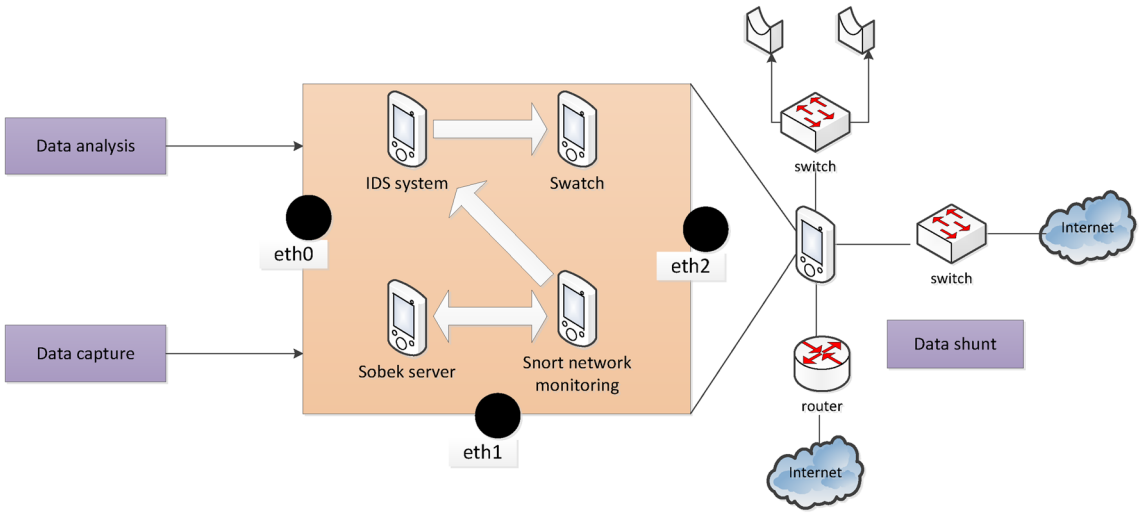

Fig. 2 Schematic diagram of the third generation honey

make any changes to the transmission process of the network data packet, so that the honey wall is extremely difficult to be discovered by the attacker, and this feature provides a guarantee for the concealment of the honey network system. Another network interface for the honey wall eth2 connects to the internal management monitoring network and other normal intranet computers. Security researchers can remotely monitor the honey wall and further analyse the attack data captured by the honey wall. This interface typically uses internal IP and is protected by a strict access control policy. The honey wall is the only connection between the honey network and the external network. All the network traffic flowing into and out of the honey network will pass through it, so the control and capture mechanism of the network data flow can be realized on the honey wall (Ren and Xu 2018).

In the actual layout process, a typical honeynet usually consists of a honey wall and multiple honeynet hosts. In the honeypot network inside the honeynet, any type of system can be placed as a honeynet, such as Solaris, Linux, Windows XP, etc., which creates an environment for the attacker to feel more realistic. At the same time, the various tools and tactics used by attackers can be understand via configuring different services for each system, such as Linux DNS, Windows Server or Solaris FTP server.

The third-generation honeynet's defence strategy can be divided into two types of DDoS attacks, which deceive attackers into the honeynet and research and track DDoS attacks (Qian et al. 2016).

1. The attacker is deceived into the honeynet in the honeynet, and the honeywall system is detected and guides the attacker to enter the honeynet to spoof the network. At this time, the attack events made by the attacker in the honeynet system are all recorded through the data capture system and transmitted to the log server in a hidden way. Furthermore, the data control system is used to suppress the attack behaviour initiated by the attacker.
2. When faced with a malicious attack, such as a botnet or a client-side attack, the attack mode and behaviour through the honey wall are identifid, the information is recorded about the malicious attack through the data capture system, and the botnet or malicious server is tracked to provide information for further research.

3 Analysis of DDoS attacks principle

3.1 DoS and DDoS

The full name of DoS is denial of service. From the various methods of cyber-attacks and the damage caused, DoS is a very simple but effective offensive method. Its purpose is to deny the user's service access, disrupt the normal operation of the organization, and ultimately invalidate some users' Internet connection and network system. There are many ways to attack DoS. The most basic DoS attack is to use a reasonable service request to occupy too many service resources, so that legitimate users cannot get services.

Distributed denial of service is that the English full name is distributed denial of service, which is a special form of denial of service attacks based on DoS, and is a distributed, collaborative large-scale attack. DoS attacks generally use a one-to-one approach. The difference is that DDoS attacks use a batch of downtime to launch an attack and assault the victim on a larger scale. Such a rapid attack is unpredictable and therefore it has powerfully destructive.

The DDoS attack is divided into three layers: the attacker, the master and the downtime. The three layers play different roles in the attack. The computer used by the attacker is the master, which can be any host on the network, or even it is an active laptop. The attacker manipulates the entire attack process and sends an attack command to the master. The master is the host used by the attacker and can control a large number of downtimes. The host has specific programs installed on the master, so the special commands can be accepted from attackers and its programs can be sended to the downtime. It only issues commands without participating in the attack. A downtime is a host invaded and controlled by an attacker that run an attacker program on them to accept and operate commands from the master. The downtime is the executor of the attack and actually sends an attack to the victim host. The first step for an attacker to launch a DDoS attack is to find a host with vulnerability on the Internet. After entering the vulnerable host system, the backdoor program is installed. The host is that the attacker's invasion turned into downtime. The second step is to install the attack program on the downtime and attack the victim host under the attacker's command. Because the attacker manipulates behind the scenes through downtime, it will not be tracked during the attack, and is not easy to find identity. Figure 3 is a typical attack schematic.

The reason for adopting such a structure is to isolate network connections and protect the attacker from being tracked by the monitoring system as the attack progresses. At the same time, it can better coordinate the attack, because the number of downtimes is too large, and the command issued by one system will cause the network of the control system to block, affect the suddenness and synergy of the attack. Moreover, a sudden increasing in traffic can easily expose the location and intent of the attacker.

In order to maximize the effectiveness of an attack, an attacker often needs to control as many downtimes as possible. Generally, the process of attacking the host and the implanted program is automatically completed by an attacker's own attack tool.

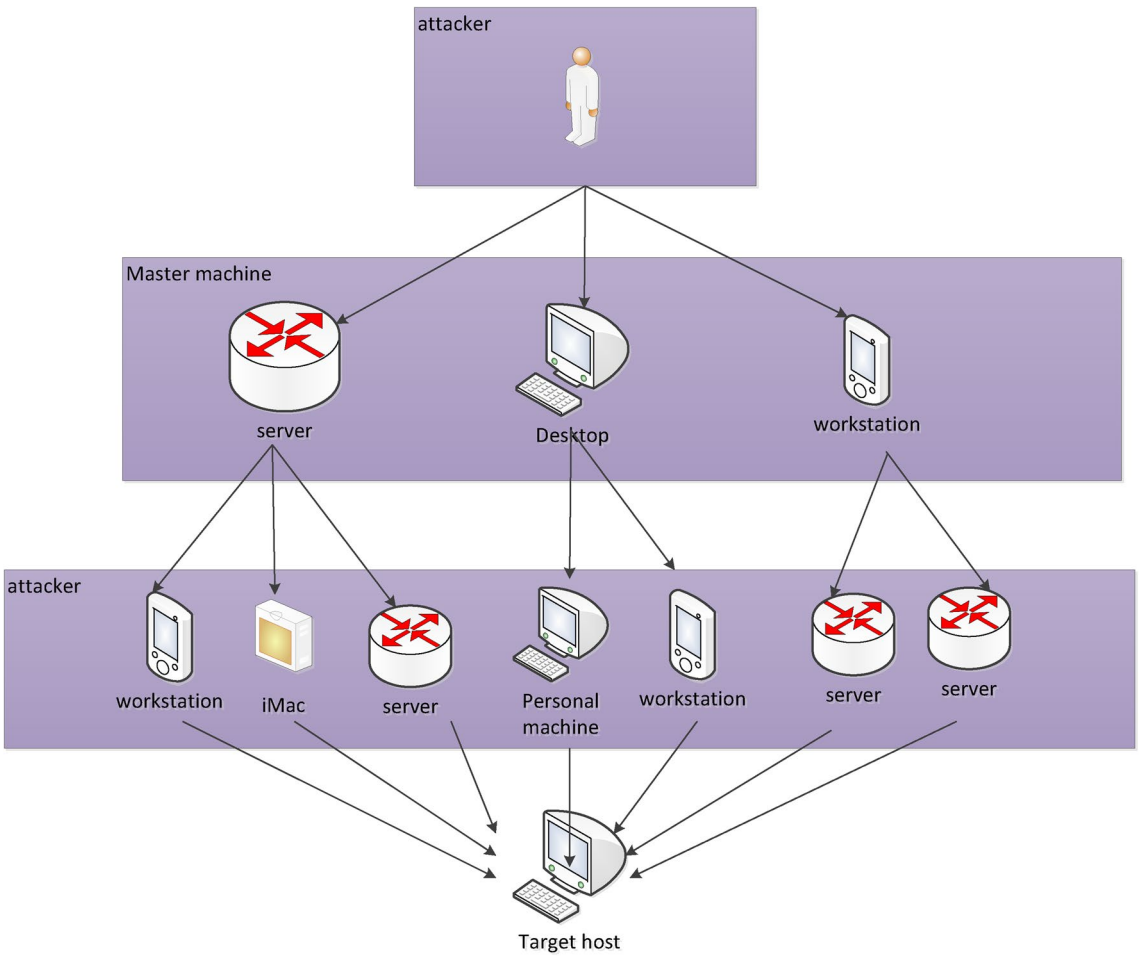

Fig. 3 DDoS attack schematic

3.2 DoS attacks principle and defence method

DoS attacks are the foundation of DDoS. The most basic DoS attack is to use a reasonable service request to occupy too many service resources, which cause the service to be overloaded and unable to respond to other requests. These service resources include network bandwidth, file system space capacity, open processes, and so on. DoS may result in insufficient resources, so normal access requests cannot pass.

3.2.1 DoS attacks principle and classification

Denial of service attacks from the perspective of attack principles can be divided into two categories. One is logical attack, which is vulnerability attack. The other is traffic-based attack, which is also known as flood attack (Qian et al. 2017a, b). Flood attacks are that use the target system to implement vulnerabilities and perform denial of service attacks on the target host, they often do not require attackers to have high attack bandwidth. The defence against this kind of attack only needs to fix the defects in the system. A flood attack refers to the number of service requests that send more than the target system's service capabilities to achieve the purpose of the attack. To defend against such attacks, the attack data must be filtered or shunted under the help of an upstream router. Some attacks have the characteristics of both logical and flood attacks, such as the SYN flood attack. Although the shortcomings of the TCP protocol itself have been exploited, they still need to send a large number of attack requests. There are also some methods of attack that exploit system design flaws to generate higher-traffic communication data than attackers for brute-force attacks. Specifically, the usual DoS attacks are as follows.

1. Smurf, means that broadcast information can be sent to a machine in the entire network by a certain means, for example, via a broadcast address. When a machine sends an ICMP echo request packet by using a broadcast address, some systems respond to an ICMP echo response packet. In this case, sending a packet will receive several response packets. When the source address is the address of the attacking host and the destination address is the packet of the broadcast address, many system responses will send a large amount of information to the attacked host, which result in a DoS attack.
2. Flooding. Flood attacks are the most common and effective means of DoS attacks. DoS flood attacks are divided into three types: TCP-SYN flood, UDP flood, and ICMP flood (Qian et al. 2017a, b). A SYN flood attack is that, when an attacker sends a large number of semi-connected TCP packets, the target server will be overloaded. The UDP flood attack is mainly caused by an attacker using a large number of UDP packets to affect the target server. In most cases, the bandwidth of the server is blocked. An ICMP flood attack is a ping flood attack, which its principle is to send a large number of ICMP packets to a computer, so that the system consumes all resources to respond until the effective network traffic cannot be processed. For a server, the available TCP connections and resources are limited. If a DoS attack occurs, the available TCP connection queues of the server will be blocked quickly, the available resources of the system will be drastically reduced, and the available bandwidth of the network will be rapidly reduced, and then the network will not be able to provide normal services to users.
3. Ping of death. According to the TCP/IP specification, the maximum length of a packet is 65,536 bytes. Although the length of a packet cannot exceed 65,536 bytes, the superposition of multiple fragments divided into one packet can be done. When a host receives a packet longer than 65,536 bytes, it will be attacked by the ping of death.

3.2.2 DoS attacks defense method

For different types of DoS attacks, several strategies for defending against DoS attacks are as follows.

In the defence configuration, the QoS CAR (committed access rate) should be applied to defend against it. The purpose of blocking the network cannot be achieved because of limiting the speed of ICMP packet traffic. Some features of QoS, such as weighted fair queuing (WFQ), general traffic shaping (GTS), and custom queue (CQ), can be used to defend against DoS attacks. For example, when the network is subjected to a remote and mad ping attack, it needs to utilize the WFQ feature to make the access queue of the entire external network more regular and weaken the weight of the crazy ping attack. If the router of the current network has TCP interception, it can also resist DoS attacks and can be well monitored and intercepted when the other party sends the data stream. If the data packet is legal, normal communication is allowed. Otherwise, the router will display a timeout limit to prevent its resources from being exhausted. These can be attributed to the use of device rules to properly shield continuous, high-frequency data impact that is the fundamental principle to prevent DoS attacks.

In addition, due to the nature of the Web service and the TCP protocol itself, there will be a feedback regardless of what instructions the outside world sends to the server, even the wrong feedback. For example, the inability to access the specified page server returns an HTTP 404 error, which is a feedback, and the DoS attack can take advantage of this feature, which can allow the server to accept a large number of instructions, and information congestion caused by the

network. Therefore, in order to deal with this situation, the various patches should be followed up in a timely manner, the key nodes are often scanned and monitored, and the new vulnerabilities are timely repaired. Of course, a firewall is needed to be added to the backbone device because the firewall itself has anti-DoS attack capability. In addition, the unnecessary ports and services are also filleted, and only opening ports that need to provide services.

The above lists several DoS attack defence methods. Relatively speaking, DoS attacks still have more ways to implement defences.

3.3 DDoS attacks structure and method

Compared to DoS attacks, DDoS attacks far outweigh the DoS attacks in terms of size, complexity, means, hazards, detection and prevention. Some methods are effective for DoS attacks but not for DDoS attacks. This chapter will start with the structure and attack mode of DDoS attacks and analyse the principle of DDoS attacks.

3.3.1 DDoS control structure

Thousands of DoS attacks around the world can cause fatal damage to any server or site. Therefore, whether there are many unrelated hosts can be the precondition for the entire attack, the tighter the relationship between these hosts and the target host, the better the attack effect.

The distributed denial of service attack uses a three-layer control structure to launch DoS attacks from many distributed hosts simultaneously, which result in the embarrassment of the attack object. The latest popular way to aggregate host groups of this size is to use botnets.

The topology of the DDoS 3-layer control structure is shown in Fig. 4. The top layer in the figure is the attack initiator, which can be any host on the network or even an active terminal. Its role is to issue attack commands to the layer 2 attack server. layer 2 is an attack indirect terminal that main task is to publish console commands to the attack directly on the terminal. Layer 2 and layer 3 attack executors are installed on some intrusive, unrelated hosts (Jiang et al. 2017). The reason for an attacker to use some hosts indirectly as a springboard is to avoid direct discovery by the control source and still not affect the attack after a layer 2 host is blocked.

The lowest level terminal is the direct originating point of the attack. This layer structure consists of many network hosts, and the platform can be any operating system. The attacker illegally installs the attack program on these terminals, and specifies the address of one or several attack

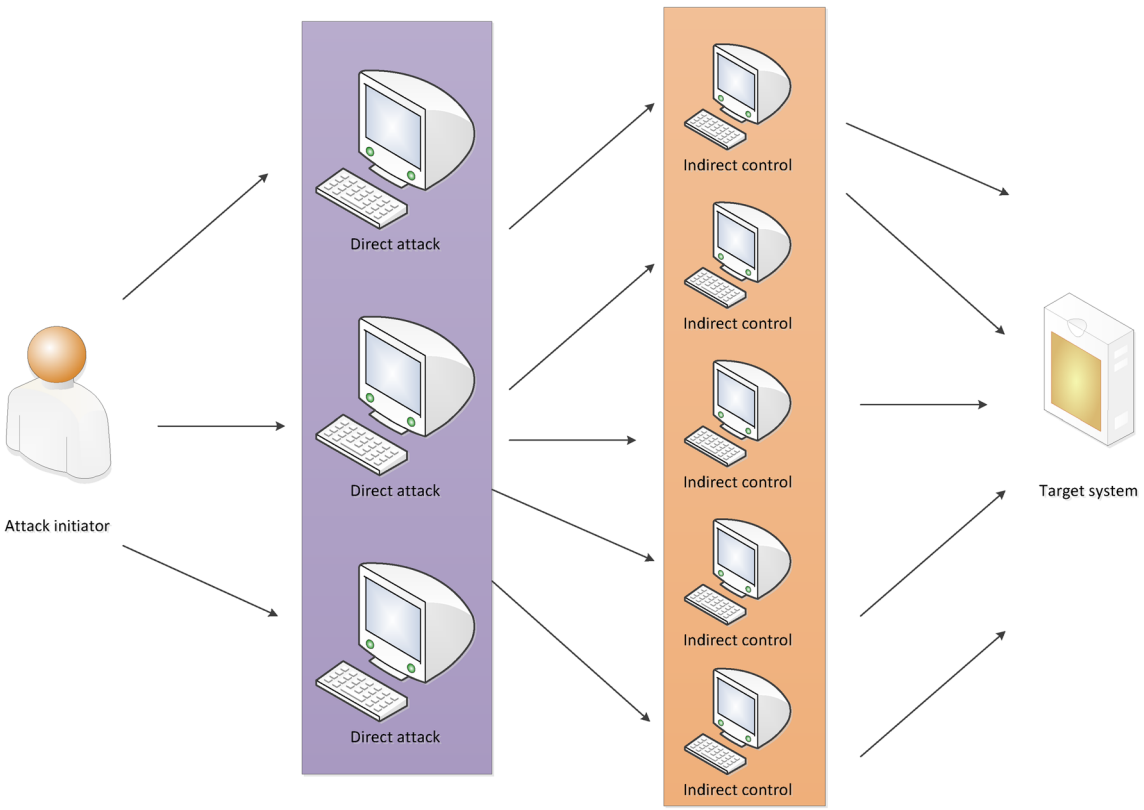

Fig. 4 Three-layer structure of DDoS attack

servers on the upper layer, so that the attack behaviour is directly controlled by the attack server.

The larger the amount of resources an attacker controls, the better the ability to organize DDoS attacks. On the third layer attack terminal, the attacker sends a program by implanting a DDoS attack packet, and uses the program to send a malicious attack packet to the target host. Under the scheduling of the main server during the attack, the DDoS attacker in the agent responds to the command and sends a large number of packets to the target host at high speed, which causes the target host to crash or unable to respond to normal requests (Jiang et al. 2015a, b).

3.3.2 DDoS attack steps

An attacker launching a normal DDoS attack typically takes three steps.

1. At the beginning, the attacker needs to collect information about the target. The following situations are the information that the attacker cares about is the number and address of the target being attacked, the performance of the target host, and the bandwidth of the target. For a DDoS attacker to attack a site on the Internet, it is important to determine how many hosts support the site. The larger site may have many hosts using load balancing technology to provide the same website service. Take a website as an example, it has several addresses to provide services, such as 88.218.71.87; 88.218.71.88; 88.218.71.86 and so on. If you want to do a DDoS attack, you need to attack all servers. If only the 88.218.71.87 machine is paralyzed, but other hosts can still provide the www service, then this DDoS attack will obviously fail. Therefore, if you want to completely crash the site, you must have all the machines with these IP addresses to be effective. In another practical application, the IP address usually represents several machines, the website maintainer can use four or seven layers of switches for load balancing and assign the IP addresses for access using a specific algorithm, which can be transferred to each host of the subordinate (Jiang et al. 2015a, b). At this time, the situation of DDoS attackers is more complicated, and his task is to make the services of many hosts abnormal. Based on the above discussion, the attacker investigates that all specific situations are related to how much downtime is used to achieve the results.
2. The attacker must choose to meet specific machine conditions, such as hosts with good link status, hosts with good performance, and hosts with poor security management. These hosts are often referred to as network broiler, most of which are lacking in self-protection, personal network devices with poor host management and network management. Therefore, exposing your own computer cluster is also a huge hidden danger. After the attacker grasps certain information through scanning or other channels, the host will be controlled to obtain the highest management authority, or at least the account will be gotten that has the authority to complete the DDoS attack task.

The attacker's job is to get information randomly or in a targeted way, and then use the scanner to discover vulnerable machines on the Internet, such as program overflow vulnerabilities, CGI, Unicode, FTP, database vulnerabilities, etc. The scan results indicate that the attacker wants to see it and then attempted to invade and take up downtime. The attacker uploads the DDoS issuer and the control client to the DDoS package, and the attacker can send a malicious attack packet to the victim.

3. After the preparation of the above two phases, the attacker can uniformly launch the attack command and initiate the DDoS attack to make the attack target paralyzed. The DDoS tool has a series of mature software products in UNIX or Windows environment, such as Trinoo, TFN, TFN2K, STACHELDRATH, etc. The default settings of these DDoS attack tools are Trinoo client, the default port used between host and agent TCP1524, TCP. 27665, UDP27444, UDP 31335 communicate with the host. When the TFN client, the master, and the agent host communicate with each other, ICMP ECHO and ICMP ECHO REPLY packets are used. TN2K's client, host, and agent host do not use any of the specified ports, which can be specified at runtime or randomly selected by the program, but the UDP, ICMP, and TCP packets are combined to communication, and their information provides clues to sniffing possible DDoS attacks.

Detecting and defending DDoS attacks is much more difficult than defending against DoS attacks. The DDoS method is often used to contact the botnet on a daily basis. In contrast, botnets are easier to control, behaviours are more concealed, and the security risks are unprecedented.

3.4 Self-similarity of network services

Self-similarity means that stochastic processes have the same statistical properties on various time scales. The self-similarity in network communication is manifested over a long period of time, and the statistical characteristics of the number of packets per unit time do not change with time scale. Another important feature of self-similarity is the long-range correlation, which is mainly manifested in the second-order statistical properties of the stochastic process.

A general description is given below. For the meaning of self-similarity of the subject, if the process of the time-dependent process network service satisfies the following conditions, the process is considered to be self-similar.

$$y(t)^d = \alpha^H y(t/\alpha) \tag{1}$$

When $\alpha > 1$, the small sample is amplified to obtain the statistical characteristics of the large sample; when $0 < \alpha < 1$, the large sample is reduced to obtain the statistical characteristics of the small sample.

The sample is self-similar proportional, that is, the time axis and the network flow axis are simultaneously multiplied by two factors to obtain self-similar subsamples. The factor for the timeline is denoted as M_t, and the factor for the network flow axis is denoted as M_y.

Supposing that the total sample be n and the subsample be n', then $M_t, = n/n'$. At the same time, the standard deviation of the total sample is S, and the standard deviation of the subsample is S', so $M_y = S/S'$. Then, the self-similarity parameter H can be expressed as the following formula.

$$H = \frac{1}{2} * \frac{\ln M_y}{\ln M_t} = \frac{1}{2} * \frac{\ln S - \ln S'}{\ln n - \ln n'} \tag{2}$$

The statistical description is given below.

Let $X = \{X_j, j = 1, 2, \ldots\}$ be a stochastic random sequence with a covariance, that is, X has a constant mean $\mu = E[X_i]$ and a finite variance $\sigma^2 = E[(X_i - \mu)^2]$, and its autocorrelation function is only related to k, and has the following form.

$$\begin{aligned} r(k) &= \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2}(k = 1, 2, \ldots) \\ &\sim k^{-\beta} L_1(k), k \to \infty \end{aligned} \tag{3}$$

Among them, $0 < \beta < 1$, L1 meets $\forall x > 0$, there is $\lim_{t\to\infty}(L_1(tx)/L_1(t)) = 1$.

Let $X_k^{(m)} = \frac{(X_{km-m+1}+\cdots+X_{km})}{m}$ be the m-order smoothing process of X, and remember the autocorrelation function of time series $X^{(m)} = (X_1^{(m)}, X_2^{(m)}, \ldots)$ is $r^{(m)}$, m = 1, 2, 3, …, n.

Definition 1 Process X is strictly second-order self-similarity and has a self-similarity coefficient $H = 1 - \beta/2$, if its m-order smoothing process $X^{(m)}$ has the same correlation function as the original process X, i.e. $r^{(m)}(k) = r(k)$ pairs all (m = 1, 2, …, k = 1, 2,…) is established.

Definition 2 Process X is said to be asymptotic second-order self-similar, and has a self-similarity coefficient $H = 1 - \beta/2$, if

$$r^{(m)}(1) \to 2^{1-\beta} - 1, m \to \infty \tag{4}$$

$$r^{(m)}(k) \to (1/2)\delta^2(k^{1-\beta}), m \to \infty \tag{5}$$

$\delta^2(f)$ represents the quadratic difference operator acting on f, i.e.

$$\delta^2(f(k)) = f(k+1) - 2f(k) + f(k-1) \tag{6}$$

The most striking feature of the strict (or asymptotic) self-similar process is that its m-order smoothing process $X^{(m)}$ is non-degenerate when it is related to the $m \to \infty$ correlation function structure, which is related to the traditional Poisson, Markov and other short-term correlation process autocorrelation functions. The exponential decay, that is, when $m \to \infty$, $r^{(m)}(k) \to 0$ is completely different.

Although there are a large number of stochastic models that exhibit self-similarity, the stochastic models considered suitable for sudden traffic modeling are only fractal and ARIMA processes, but they are all based on fractal Gaussian noise. The independent incremental process of fractal Brownian motion is fractal Gaussian noise, and the fractal ARIMA process can be regarded as the result of fractal Gaussian noise filtering by ARIMA parameters as filter coefficients. The autocorrelation function of fractal Gaussian noise satisfies the following formula.

$$r(k) = \frac{\sigma^2}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}) \tag{7}$$

When $0.5 \le H \le 1$, the fractal Gaussian noise is a strict second-order self-similar process with Hurst coefficient H. Because of its simple parameters, it becomes the main tool for self-similar business modelling.

The self-similarity model is the current traffic model that best describes the long-range correlation of data flows in the network. The so-called self-similarity of the data stream means that the data stream on the network does not have an essential burst length. On each time scale, from microseconds to minutes, from minute to hour, the burst period consists of some bursts. The periodic combination of burst sub-cycles consists of some smaller burst sub-cycles, the idea of which is similar to the idea of calculus.

Self-similarity in network services is mainly manifested in the sudden existence of multiple time scales and the same statistical characteristics. This is in contradiction with the fact that the characteristics of the traditional Poisson distribution, that is, the business can be smoothed by statistical averaging.

Since the last century, a large number of network measurements and analysis have proved that real network services have statistical self-similarity. The existence of self-similarity brings some unexpected effects to network performance, which directly affects the design, control and management of the network.

Because this topic needs to deal with a finite data set, for a process that is not self-similar, to some extent, its

aggregation sequence tends to be consistent with the second-order pure noise. The association will eventually show the reduction of the index and the continuity of the density function. For the size of a finite sample, the distinction between these gradations and the characteristics corresponding to the self-similar process is the problem.

4 Design and implementation of defence method based on honeynet

4.1 Design of defence mechanism

In order to track information related to the mobile botnet, the network topology is deployed as shown in Fig. 5.

The system is an improved version of the honeynet system, where the Honeypot is an unpatched Windows 2000 or Windows XP. Because this system is very easy to be attacked, it is easier to receive the favour of hackers and collect the information needed to defend against DDoS attacks. The most important thing in the whole deployment is the honey network gateway HoneyWall that is deployed in bridge mode. It includes three network interfaces, in which eth0 is connected to the external network, eth1 is connected to the honey network, and the two interfaces are connected in a bridge manner. The TTL decrement and network routing of the network data packet are not performed, and the MAC address of the network is not provided, so the attack is performed. For example, HoneyWall is completely invisible and the only connection point between Honeynet and other networks. All network traffic flowing into and out of Honeynet will pass through HoneyWall and be controlled and audited. HoneyWall's other network interface, eth2, connects to the log control server, enables data captured by HoneyWall to be sent to the log server. It also enables remote control of HoneyWall, which typically uses internal IP and is tightly protected.

This improved version of Honeynet uses a multi-level data control mechanism on HoneyWall.

1. Its version includes the use of IPTables to provide outbound traffic restrictions and the use of network intrusion prevention systems to invalidate known attacks. The outbound traffic restriction mechanism restricts the number of connections and traffic rates that each honeypot host can initiate to each other per unit time through IPTables. Once an attacker attempts to use the compromised honeypot master to initiate scanning, such as denial of service attacks, etc., IPTables on HoneyWall outbound packets that exceed the limit will be discarded and a warning notification will be generated so as not to pose a hazard to third-party networks.
2. The network intrusion prevention system is implemented by the snort_inline tool rewritten based on the famous open source network intrusion detection system snort.It is found that it contains known attack features by looking at each out packet, and an alert will be generated and selected according to the configuration. Dropping a packet or modifying a packet invalidates the attack.

The above data control mechanism can minimize the security risks caused by deploying the honeynet, but it cannot be completely eliminated, and still needs to be paid attention.

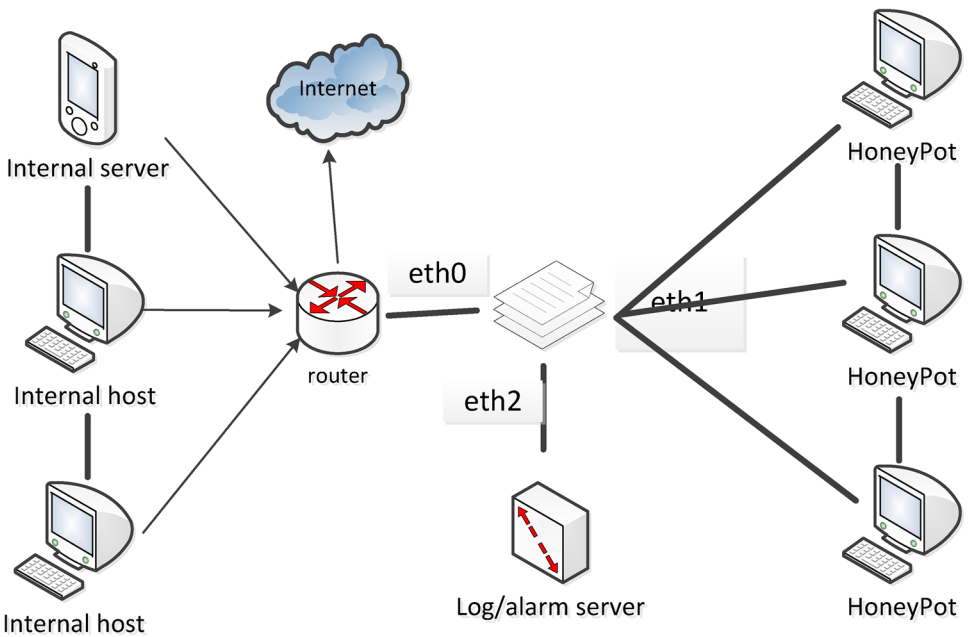

Fig. 5 Honeynet topology diagram

Similarly, in order to meet the data capture requirements of the honeynet system, a multi-level data capture mechanism is used on HoneyWall and each honeypot host to ensure comprehensive and rich attack data for further analysis of attack behaviour.

Firstly, IPTables will log all network connections to the honeynet and record the network connections initiated by the attacker after the honeynet is compromised and the alarms that exceed the number of connections and traffic speed limits.

Secondly, the network intrusion detection system snort deployed on HoneyWall will listen to all the network traffic flowing into and out of the honeynet on the eth1 interface and capture it into the local pcap file, and generate an alarm log for the packets that meet the characteristics of the snort attack. This data provides comprehensive network traffic information for us to track down and restore an attack.

Finally, the attacker usually uses an encrypted channel, such as SSH, to launch an attack command during the attack, and the data capture mechanism provided by HoneyWall cannot understand the attack behavior contained in it, even if all the data packets are intercepted and monitored. However, this encrypted traffic will eventually be received and decrypted on the honeypot host, so a system behavior monitor can be installed on the honeypot host to capture the attacker's attack on the encrypted channel. In the improved version of the honeynet architecture, as shown in Fig. 6, Sebek is used to capture the attacker's further attack behaviour on the honeypot host. The Sebek client is installed on the honeypot host as a kernel module. The attacker discovers the keystrokes and system behaviours and transmits them to Sebek's server through a hidden communication channel. The Sebek server is usually installed on the log server and sent to multiple honeypot hosts. Sebek data is stored.

4.2 Capture information with honeynet

Just like the Windows version of the honeynet mentioned above is an unpatched Windows 2000 or Windows XP. Therefore, this system is very vulnerable, and successful attacks take only a small amount of time. Experiments show that once the robot successfully cracks the honeynet, it tries to connect to the IRC server to get more commands. Since the data capture device is installed on the HoneyWall, the outgoing network connection can be controlled. Snort_inline is used to data control and to replace suspicious connections while on the go. A connection is suspicious if it contains typical IRC messages like "332", "TOPIC", "PRIVMSG" or "NOTICE". Of course, it is also possible to prohibit the bot from accepting valid commands from the server channel. On the other hand, sensitive information about botnets can be extracted from the data captured at that point in time.

HoneyWall's data capture capabilities determine the DNS/IP address and corresponding port number that the bot will want to connect to, and the nickname and identity structure can also be extracted from the data capture log. The server's password, channel name and channel password can also be obtained this way to collect all the necessary information.

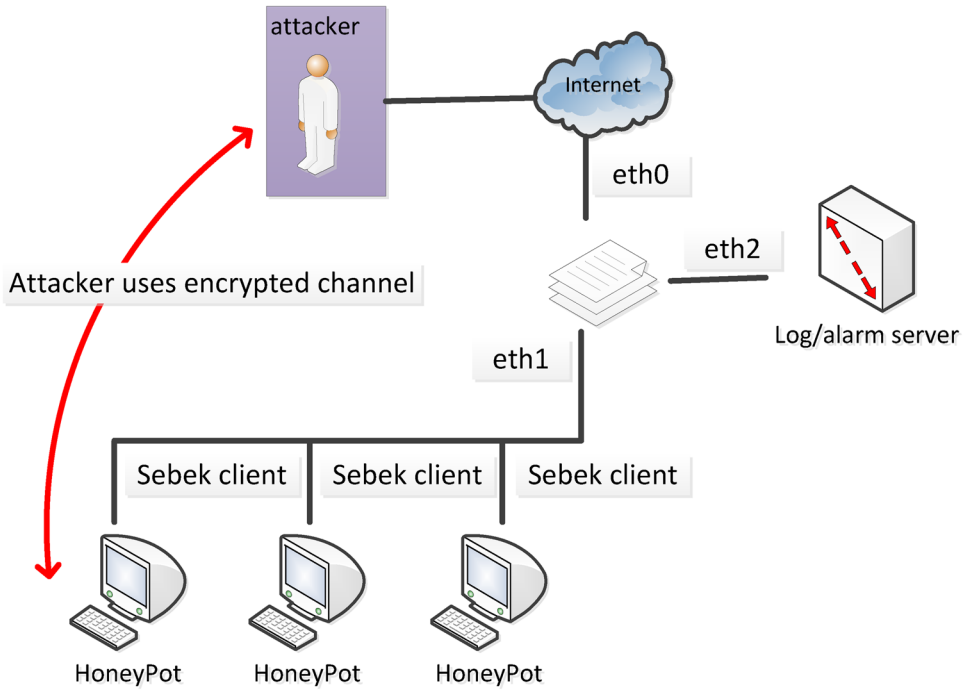

Fig. 6 Schematic diagram of Sebek deployment

4.3 Defence method implementation

Some cases show that most DDoS attacks are initiated by hackers who control large-scale botnets. After gaining the basic features of botnets, there are multiple ways to defend against DDoS attacks caused by botnets. This part is just an important introduction. In fact, the principles of these methods are basically the same. Various methods are applied to make the attacker lose control of the botnet and the weapon to launch the DDoS attack. The complexity and effects of each method vary, and the following are three defence methods.

4.3.1 Simulation controller

The controller is simulated and fully controlled the botnet to achieve DDoS defence. The premise is that the basic information of the botnet mentioned above is mastered. After the analogy controller is authenticated, it can send commands supported by various bots.

(1) Sending update command.

You can download the bot and run your own killing tool. Of course, you need to set a website or file server as a carrier to store the killing tool and also modify the control password, or update the bot to take over the entire botnet and cause the owner of the botnet to fail. The disadvantage of launching a weapon DDoS attack method is that it is necessary to grasp whether the bot authenticates the downloaded program, such as authentication, and this method fails.

(2) Self-delete command.

Making the bot delete itself is that the method also enables the attacker to lose the botnet and achieve the purpose of defence. The disadvantage of this method is that it is only valuable when it is found that the botnet engages in malicious activities. Otherwise, after simply deleting the bot, the vulnerable system will soon be infected by other malicious code. In fact, a host with a bot usually contains multiple other bots simultaneously. In actual cases, some attackers sometimes repair the computer's vulnerabilities after they invade a computer and leave control channels to prevent the computers they control from being snatched by others.

4.3.2 Clearing the program on the host

Users find and locate the computers that are implanted with bots and have them use special software to clear the robots and perform security upgrades, which cause attackers to lose botnets or make botnets smaller, and achieve defence or slow down DDoS the purpose of the attack. It is a huge job and there are many difficulties. It is understood that in 2005, CNCERT/CC mastered more than 100,000 computers controlled by botnets, but there is no channel to contact these computers users; and this information cannot be directly published that result in greater user risk, so some users of the network can only be notified through a collaborative channel. In addition, removing the Bot on the host is not only a procedure that is too large and inefficient, but a mitigation method, because the attacker continues to expand so that new computers can join the botnet.

4.3.3 Cut off remote control

The remote control is cut off, and the connection between the user host and the control server is also cut off, so that the attacker loses control over the botnet and loses the weapon that launches the DDoS attack, thereby achieving the purpose of defense. The premise against the DDoS attack method is to grasp the accurate information of the control server. Network management can cut off the connection between the network user and the network gateway or the control server of the security device, so that users in the network are not controlled by the botnet. The control method can point to the control server by prohibiting the IP or cancelling the domain name used by the botnet, and usually using the dynamic domain name to achieve the purpose of preventing the attacker from losing control of the botnet, so as to achieve the purpose of defence in advance. If the registration authority of the domain name has judicial jurisdiction, the domain name used by the domain name can be revoked in accordance with due process of law. If the control server is located in a country, you can turn off the botnet's control server according to legal procedures. The control state is that the link state has a certain lifetime, ensuring regular updates of the link state and robustness of the protocol. The update of the link state database generates a new route entry, and the injection of the external route also generates a new link state.

4.4 Design and implementation of DDoS attack detection method

In the implementation process, it is necessary to select the most suitable self-similarity model from a large number of models, analyse and compare it with the self-similarity analysis of real network services, and also need to select relevant parameters as the self-similarity of network traffic. A fast calculation algorithm is developed for self-similarity model related parameters, the characteristics of the attack is studied, the real attack through experiments is realized, the impact of the attack on the network self-similarity is researched, and the detection attack method of the attack is established on the network self-similar parameters. The network data capture uses the network processor as a gateway to mirror the data entering and leaving the honeypot network and send it to the data capture host of the honey network

Table 1 Totle specimen value of H

H	Mean	Variance
Aggregate variance method	0.787	0.063
Cycle diagram method	0.877	0.081
R/S method	0.748	0.012
Whittle method	0.847	0.036

monitoring network for storage. The data capture host uses Libpcap to capture and store data packets.

Threshold processing is performed on the real traffic flow of the local area network, which the threshold is $y = 2 \times 10^7$ .The traffic data after the clipping is obtained, so that H is the absolute value of the difference between the limit before and after the limit, and a new H value can be obtained. The H value of the overall sample after clipping is shown in the table below (Table 1).

It can be seen by comparison: on the overall sample, there is no change in the variance of H before and after the clipping. For the aggregation variance method, the period gram method, the Whittle method, the mean value of H has a little change, but for the R/S method, H The mean value is basically unchanged. On the subsamples, before and after the clipping, H has a large change for the aggregation variance and the period gram. For the Whittle method, H has a significant change; for the R/S method, H has no significant change.

From the above, it can be seen that, in the fast-estimation algorithm based on clipping, only the R/S method is a suitable choice. The reason is obvious. Before and after the clipping, the H value of the R/S algorithm tends to be stable and the change is not obvious.

Using the aggregation variance method, the period gram method, and the R/S method, the network traffic data is limited by different thresholds. By comparing the calculated results, it can be found that, the lower the threshold, the more obvious the change of H before and after the clipping. For network traffic anomalies (DDoS), the attack can be accurately detected by the normal and abnormal H changes before and after the clipping. Since the aggregation variance and period gram are more sensitive to the change of the amplitude, and because of the relatively stable characteristics of R/S, we use the above three methods to judge and analyze the attack detection.

Table 2 shows the H value of the normal flow model. The next two tables analyse the total sample of the data. Its data are the values obtained in the continuous attack and the intermittent attack experiment, and their mean and variance, which are shown—Tables 3 and 4, respectively.

Compared the above two tables with Table 2, it is found that the variance of the self-similar Hurst coefficient of the monitored network traffic varies little on the overall

Table 2 Normal traffic model value of H

H	Mean	Variance
Aggregate variance method	0.706	0.083
Cycle diagram method	1.092	0.041
R/S method	0.748	0.015

Table 3 Continuous attack value of H

H	Mean	Variance
Aggregate variance method	0.510	0.152
Cycle diagram method	0.960	0.020
R/S method	0.783	0.013

Table 4 Disconnected attack value of H

H	Mean	Variance
Aggregate variance method	0.652	0.107
Cycle diagram method	1.161	0.038
R/S method	0.742	0.014

sample. Although the aggregation variance method can be used to judge the occurrence of DDoS attacks, the coefficient value curve and the mean value curve cannot reflect the changes brought by the attack, and the strength of the attack cannot be determined, and the attack type cannot be distinguished. In addition, since the overall sample is not conducive to the real-time nature of the detection, we propose the following method.

A real-time limit detection method is that combines network traffic self-similarity with normal model. The details are as follows.

(1) For continuous attacks. Each subsample size is 2000 (the actual time length is 0.555 h). The aggregation variance method, the period gram method and the R/S method are used, $H = H_{disnormal} - H_{normal}$ are also applied, and then the H on the non-repeating interval is aggregated. Finally, the absolute value is made and the H is gained to describe the size of the attack.

When the size of the aggregation interval taken is 10, the H value is calculated by using the aggregation variance method, the period gram method, and the R/S method, as shown in Table 5.

When the size of the aggregation interval taken is 8, the H value is calculated by using the aggregation variance method, the period gram method, and the R/S method, as shown in Table 6.

When the size of the aggregation interval taken is 3, the H value is calculated by using the aggregation variance

Table 5 The size of the sector is 10

H	Subinterval			
	1	2	3	4
Aggregate variance method	0.462	0.389	0.006	0.081
Cycle diagram method	0.452	0.189	0.085	0.004
R/S method	0.029	0.038	0.102	0.035

Table 6 The size of the sector is 8

H	Subinterval				
	1	2	3	4	5
Aggregate variance method	0.437	0.503	0.236	0.223	0.003
Cycle diagram method	0.413	0.372	0.031	0.048	0.005
R/S method	0.022	0.004	0.098	0.041	0.073

Table 7 The size of the sector is 3

H	Aggregate variance method	Cycle diagram method	R/S method
Subinterval			
1	0.110	0.364	0.035
2	0.516	0.384	0.063
3	0.671	0.546	0.027
4	0.024	0.455	0.096
5	1.001	0.276	0.127
6	0.792	0.109	0.043
7	0.222	0.003	0.061
8	0.286	0.127	0.199
9	0.116	0.148	0.024
10	0.201	0.007	0.162
11	0.312	0.020	0.214
12	0.075	0.107	0.126
13	0.067	0.061	0.057
14	0.470	0.003	0.024

method, the period gram method, and the R/S method, as shown in Table 7.

It can be seen from the above comparison that the aggregation variance and the period gram have better discrimination for continuous attacks. The aggregation variance method has larger deviations at the boundary points, the period gram is more accurate, and the R/S discrimination is worse. For continuous attacks, the change of the relative aggregation interval, the smaller the interval, the less obvious the change of the critical value H of the attack.

(2) For intermittent attacks. $H = |H_{before} - H_{after}|$ is used, then H is integrated on the non-repetitive interval, and H is gotten to describe the possibility of attack.

Table 8 The size of the sector is 10

H	Subinterval			
	1	2	3	4
Aggregate variance method	0	0.005	0	0.694
Cycle diagram method	0	0.010	0	0.574
R/S method	0	0	0	NaN

Table 9 The size of the sector is 8

H	Subinterval				
	1	2	3	4	5
Aggregate variance method	0	0	0.005	0	0.694
Cycle diagram method	0	0	0.010	0	0.574
R/S method	0	0	0	0	NaN

Table 10 The size of the sector is 3

H	Aggregate variance method	Cycle diagram method	R/S method
Subinterval			
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0.005	0.103	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0.269	0.090	0
13	0.088	0.245	NaN
14	0.866	0.057	NaN

When the size of the aggregation interval taken is 10, the H value is calculated by using the aggregation variance method, the period gram method, and the R/S method, as shown in Table 8.

When the size of the aggregation interval taken is 8, the H value is calculated by using the aggregation variance method, the period gram method, and the R/S method, as shown in Table 9.

When the size of the aggregation interval taken is 3, the H value is calculated by using the aggregation variance method, the period gram method, and the R/S method, as shown in Table 10.

It can be seen from the above that the aggregation variance and the period gram have a better discrimination degree

for the intermittent attack, wherein the aggregation variance method has a large deviation at the boundary point, the period diagram is more accurate, and the R/S discrimination degree is poor. For the intermittent attack, the change of the relative aggregation interval, the smaller the interval, the smaller the difference between the critical values of the judgment attacks.

Comprehensive (1), (2), the appropriate sub-interval size is selected, and the appropriate threshold H is choosen. In the self-similarity analysis, periodic or aggregate variance is used to identify network traffic anomalies to achieve real-time detection of network traffic anomalies (caused by DDoS attacks).

5 Conclusion

For the country and its related security agencies, it is necessary to strengthen the relevant technology platform for state investment. The discovery and monitoring capabilities of BotNet focus on the large-scale intrusions, serious attacks on the Internet such as Trojan horses, and monitoring efforts to detect attacks related to certain BotNets. We strengthen the promotion of BotNet hazards on the website to provide the public with information and solutions about popular Bots, also collect popular Bot malicious code samples on the Internet, join other emergency organizations and security vendors to increase research and analysis, and release security tools. Furthermore, the relevant departments should be actively cooperated to crack down on the criminals who use and transmit Bot. Eliminating the possibility of forming a large-scale botnet from the source is very beneficial for defending against large-scale DDoS attacks.

The threat of DDoS attacks is getting bigger and bigger, and the degree of specialization of crime is getting higher and higher. Since 2004, botnets have received increasing attention from countries around the world, and the research on botnets needs to be further strengthened. Only a deep understanding of botnets can better protect against DDoS attacks caused by botnets. A method for defending against DDoS attacks in the paper is presented, which is based on honeynet technology and does not rely on resource advantages or additional equipment. In order to effectively attack, the attacker needs to control a large number of hosts, so the attacker needs to remotely control the network. The goal of the method is to cut off the remote control network and achieve the purpose of defence by analysing and infiltrating the remote control network. In the next step, we will study multi-miam network collaborative deployment, distributed data capture and log analysis technologies, and establish a multi-level linkage honey network architecture to achieve scale-up simulation, improve processing speed, capture more attack information and more comprehensive analysis of network attacks.

Acknowledgements The work is partially supported by (1) Langfang Science and Technology Research Self-financing Project, Research on Network Abnormal Behavior Analysis Technology Based on Traffic Precursor Observation System Flow Detection (Grant no. 2015013011), (2) Hebei Science and Technology Plan Project, Research on APT Attack Detection Algorithm Based on Big Data Analysis (Grant no. 16210705), (3) Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Research on Path Marking Method of Malicious Code Attack Based on CampusNetwork (Grant no. AGK201704), (4) Research on the basic research business expenses of the central colleges and universities, based on the full-campus network DNS, the key technology of malicious domain name automatic detection (Grant no. ZY20180123).

References

Anagnostopoulos M, Kambourakis G, Gritzalis S (2016) New facets of mobile botnet: architecture and evaluation. Int J Inf Secur 15(5):455–473

Cross M, Dubouis L, Mangin M (2017) Defining flare in osteoarthritis of the hip and knee: a systematic literature review—OMERACT virtual special interest group. J Rheumatol 44(12):161–171

Dou C, Zhang Z, Dong Y (2017) MAS-based hierarchical distributed coordinate control strategy of virtual power source voltage in low-voltage microgrid. IEEE Access 3(2):1–15

Du JW, Zhang X, Zhou Y (2013) Active defense security model in the application of network deception system design. Appl Mech Mater 347–350:2860–2864

Gao HH, Chu DQ, Duan YC (2017a) The probabilistic model checking based service selection method for business process modeling. J Softw Eng Knowl Eng 27(6):897–923

Gao HH, Duan YC, Miao HK, Yin YY (2017b) An approach to data consistency checking for the dynamic replacement of service process. IEEE Access 5(1):11700–11711

Gomez C, Arciamoret A, Crowcroft J (2017) TCP in the Internet of Things: from ostracism to prominence. IEEE Internet Comput 2(9):1–12

Hassan A, Eltayieb N, Elhabob R, Li FG (2018) An efficient certificateless user authentication and key exchange protocol for client-server environment. J Ambient Intell Hum Comput 9(6):1713–1727

Jiang YZ, Chung FL, Ishibuchi H (2015a) Multitask TSK fuzzy system modeling by mining intertask common hidden structure. IEEE Trans Cybern 45(3):548–561

Jiang YZ, Chung FL, Wang ST, Deng ZH, Wang J, Qian PJ (2015b) Collaborative fuzzy clustering from multiple weighted views. IEEE Trans Cybern 45(4):688–701

Jiang YZ, Deng ZH, Chung FL, Wang G, Qian PJ, Choi KS, Wang ST (2017) Recognition of epileptic EEG signals using a novel multi-view TSK fuzzy system. IEEE Trans Fuzzy Syst 25(1):3–20

Khan MA, Khan S, Shams B (2016) Distributed flood attack detection mechanism using artificial neural network in wireless mesh networks. Secur Commun Netw 9(15):2715–2729

Kuang B, Zhao X, Zhou C (2016) The role of UDP-glucuronic acid decarboxylase (UXS) in xylan biosynthesis in Arabidopsis. Mol Plant 9(8):1119–1129

Mohammadi R, Javidan R, Conti M, SLICOTS (2017) An SDN-based lightweight countermeasure for TCP SYN flooding attacks. IEEE Trans Netw Serv Manag 14(2):487–497

Osanaiye O, Choo KKR, Dlodlo M (2016) Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. J Netw Comput Appl 67(C):147–165

Prasad KM, Reddy ARM, Rao KV, BIFAD (2017) Bio-inspired anomaly based http-flood attack detection. Wirel Pers Commun 97(1):281–308

Qian PJ, Jiang YZ, Deng ZH, Hu LZ, Sun SW, Wang ST, Raymond F, Jr Muzic (2016) Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering. IEEE Trans Cybern 46(1):181–193

Qian PJ, Jiang YZ, Wang ST, Su KH, Wang J, Hu LZ, Raymond F, Jr Muzic (2017a) Affinity and penalty jointly constrained spectral clustering with all-compatibility, flexibility, and robustness. IEEE Trans Neural Netw Learn Syst 28(5):1123–1138

Qian PJ, Zhao KF, Jiang YZ, Su KH, Deng ZH, Wang ST, Raymond F, Jr Muzic (2017b) Knowledge-leveraged transfer fuzzy c-means for texture image segmentation with self-adaptive cluster prototype matching. Knowl Based Syst 130:33–50

Ren J, Xu Y (2018) A compartmental model to explore the interplay between virus epidemics and honeynet potency. Appl Math Model 59:86–99

Saied A, Overill RE, Radzik T (2016) Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing 172(C):385–393

Sharma A, Singh R, Pandey G (2013) Detection and prevention from black hole attack in AODV protocol for MANET. Int J Comput Appl 50(5):1–4

Somani G, Gaur MS, Sanghi D (2016) DDoS attacks in cloud computing: collateral damage to non-targets. Comput Netw 109:157–171

Sombolestan SM, Rasooli A, Khodaygan S (2018) Optimal path-planning for mobile robots to find a hidden target in an unknown environment based on machine learning. J Ambient Intell Hum Comput 10(5):1841–1850

Stalans LJ, Finn MA (2016) Understanding how the internet facilitates crime and deviance. Victims Offenders Int J Evid Based Res Policy Pract 11(4):1–8

Stone-Gross B, Cova M, Gilbert B (2011) Analysis of a botnet takeover. IEEE Secur Privacy 9(1):64–72

Tapaswi S, Mahboob A, Shukla AS (2014) Markov chain based roaming schemes for honeypots. Wirel Pers Commun 78(2):995–1010

Taylor SJE (2019) Distributed simulation: state-of-the-art and potential for operational research. Eur J Oper Res 273:37–47

Wen CY, Juan YH, Yang AS (2017) Enhancement of city breathability with half open spaces in ideal urban street canyons. Build Environ 112:322–336

Xin W, Myeongwon O, Katsumi S (2016) Gel-free/label-free proteomic analysis of root tip of soybean over time under flooding and drought stresses. J Proteom 130:42–55

Yang Y, Mi J (2011) Design and implementation of distributed intrusion detection system based on honeypot. Comput Knowl Technol 100:303–308