DG-Based active defense strategy to Defend against DDoS

Rui guo, [1, 2], Guiran chang [1], Yuhai qin [3], Baojing sun [2], An Liu [2], Bencheng Zhang [2], Dan peng [3]
1(College of information science and engineering, Northeastern University, Shenyang china)
2(Electronic scouting and commanding department, College of Shenyang artillery, china)
3(China criminal police University, Shenyang, china)
Corresponding author: Phn: +86-15942303744, E-mail: happyachilles@sohu.com

Abstract

In this paper, it is advocated that defenders should take active action to stop DDoS attacks. We propose a new model based on Differential Games theory. Four main actors are included, Attacker, Defender, Victim, and Botnet. It is our belief that Victims who experience an attack should cooperate with Defender to defend for a DDoS-attack. The model indicates the minimum number of Bots that should blocked by Defender. A Differential Games model is used to determine how a Defender combats an Attacker and protect the servers. The feasibility and effectiveness of our approach is validated by measuring the performance of an experimental prototype against a series of attacks. The advantages of the scheme are discussed and further research directions are given.

1. Introduction

Denial-of-Service (DoS[1]) attacks use legitimate requests to overload the server, causing it to hang, crash, reboot, or do useless work. The target application, machine, or network spends all of its critical resources on handling the attack traffic and cannot attend to its legitimate clients. Both DoS and DDoS are huge threats to the operation of Internet sites, but the DDoS[2,3] problem is more complex and harder to solve.

With the developing of computer technology, DDoS was first launched by worms or Botnet. Lately, DDoS has shown some new trends. It can be started automatically, controlled by a center computer which distributes the attacks. Because a great lot of computers were infected and controlled by the Attacker, DDoS can congregate more than 1Gbps, by now it can jam any server or network with unwanted traffic.

For DDoS defense, many methods were designed to help victim server survive, such as increasing resources of victim server or authenticating clients before serving. Other ways eliminate the possibility of DDoS, namely, by reducing the number of Bots that are controlled by attackers. Although in some situations the first method can succeed, during a DDoS in progress, who will win the battle depends on which side has more resources. However, since an attacker will only start to attack when it has enough resources, the victim server will often be overwhelmed after all.

The second method is based on the reason of DDoS. DDoS needs a lot of computers synchronously and automatically attack the same server. Attacker needs a toolkit to control many hosts. For stopping DDoS, we need recognize and penetrate this remote control method, and then by using another method, to stop the attacks. Based on this idea, we bring our Differential Games Model (DGM).

We present an implementation of these concepts, along with experimental results from our laboratory testbed. In the rest of this section we give a brief overview of DGM. Section 2 explains the rationale behind our approach. Section 3 presents the architecture based on DGM. Section 4 gives the details of implementation and performance. Section 5, we conclude with a discussion of the deployment options.

2 Active defense strategy

Our defense strategy model(DGM) is composed of four components, namely: Attacker, Defender, Victim and Botnet (see Figure 1）. In our DGM model, four

978-0-7695-3134-2/08 $25.00 © 2008 IEEE
DOI 10.1109/MUE.2008.53

different actors can be distinguished: The first is Attacker. An Attacker uses all kinds of software exploits, worms and malicious code to conquer and control a large amount of computers, called Bots. The second one is Botnet, which is controlled by Attacker. Botnet consists of a large amount of computers, also called zombies or drones. The third actor is Victim. In general, Victim has a good immunity against worms, but is still vulnerable for DDoS attacks, performed by Botnet. The last actor is Defender, which serves as a DDoS protector. Before or during a DDoS attacks, it

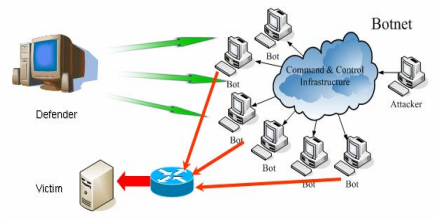

Figure 1. active defense model-

can combat with Attacker or Botnet. As shown in Figure 1, our active defense model describes the battle between Defender and Attacker. Since the main task of Victim is to serve clients, for example as a game server, DNS server or VoIP authentication server, it is unacceptable to be burdened with the extra load of the DDos defense. To overcome this problem, Defender could be a third party server, separated from the Victim. This task can be fulfilled by anything ranging from an internet administration, or governmental organization to commercial services. The aim of Defender is helping Victim to survive the DDoS attacks and keep it running during such an attack. Another task for Defender is to locate and detect the Attacker.

There are many parameters involved in the setup of a DDoS attack, where the number of Bots is the most important one. An Attacker will raise its number of Bots until it has sufficient resources at its disposal to overwhelm Victim. When the number of Bots reaches the critical point, a DDoS attack is available. Defender, at its turn, will set up honeypots or spread anti-worm software. Upon this we set up our DGM that can be used for controlling the Bots in a Botnet.

In this paper, we discuss the competition between the Attacker and the Defender based on our DGM. The effect of DDoS defense concerns many factors, such as the performance, number of Bots, the condition of networks, etc. In this paper, as an indicator of the performance of the Defender, the number of Bots is necessary to perform a successful DDoS is used.

3 Differential Games Model and DDoS defense

It is supposed that Attacker and Defender, at time *t*, control the number of Bots x(t), y(t). In the beginning *t0=0*, Bots controlled by Attacker and Defender are x(0), y(0). We assume that x(t) and y(t) are Continuous differentiable functions, i.e. changing only slowly. Whether a DDoS will succeed is directly decided by the number of Bots that is deployed. Both Attacker and Defender seek to minimize the use of resources and succeed in their tasks in the process of DDoS. Using these assumptions, we can design a Linear Quadratic Nonzero Sum Differential Games Model [3,4] as follow.

$$\begin{cases} \dot{x} = au_1 - bx \\ \dot{y} = cu_2 - dy \\ x(0) = x_0 \\ y(0) = y_0 \end{cases}$$

$$J_1(u_1,u_2) = -\frac{\xi}{2}(x(T_0))^2 + \frac{\eta}{2}\int_0^{T_0} u_1^{\,2}(t)dt$$

$$J_2(u_1,u_2) = -\frac{\beta}{2}(x(T_0))^2 + \frac{\gamma}{2}\int_0^{T_0} u_2^{\,2}(t)dt$$

$$H_1(x,y,u,\lambda_1) = \frac{\eta}{2}u_1^{\,2} + \lambda_{11}(au_1 - bx) + \lambda_{12}(cu_2 - dy)$$

$$H_2(x,y,u,\lambda_2) = \frac{\gamma}{2}u_2^{\,2} + \lambda_{21}(au_1 - bx) + \lambda_{22}(cu_2 - dy)$$

The necessary condition that $(u_1^{\,*}(t), u_2^{\,*}(t))$ is Open-Loop Nash Equilibrium[5,6,7] is $(u_1^{\,*}, u_2^{\,*})$ and its Curve Track x^*, y^* can be met by equation as follow:

$$\left.\frac{\partial H_1(x^*, y^*, u_1, u_2^*, \lambda_1)}{\partial u_1}\right|_{u_1=u_1^*} = \eta u_1^* + a\lambda_{11} = 0$$

$$\left.\frac{\partial H_2(x^*, y^*, u_1^*, u_2, \lambda_2)}{\partial u_2}\right|_{u_2=u_2^*} = \gamma u_2^* + c\lambda_{22} = 0$$

Where a, c denote the Growth of Bots controlled by Attacker and Defender, and b, d denote decline of Bots because many reasons of network. ξ , η , β , γ are weighting coefficients, describing Attacker and Defender. It should be noted that Hamiltonian functions [8] of Attacker and Defender are defined as follows.

$$\dot{\lambda}_{11} = b\lambda_{11}$$
$$\lambda_{11}(T_0) = -\xi x^*(T_0)$$
$$\dot{\lambda}_{12} = d\lambda_{12}$$
$$\lambda_{12}(T_0) = 0$$
$$\dot{\lambda}_{21} = b\lambda_{21}$$
$$\lambda_{21}(T_0) = 0$$
$$\dot{\lambda}_{22} = d\lambda_{22}$$
$$\lambda_{22}(T_0) = -\beta y^*(T_0)$$

$$\begin{cases} \dot{x}^* = au_1^* - bx^* \\ \dot{y}^* = cu_2^* - dy^* \\ x^*(0) = x_0 \\ y^*(0) = y_0 \end{cases} \quad (\text{Equation 1})$$

We can get the adjoint vectors.

$$\begin{cases} \lambda_{11} = -\xi x^*(T_0)e^{b(t-T_0)}, \lambda_{12} = 0 \\ \lambda_{21} = 0, \lambda_{22} = -\beta y^*(T_0)e^{d(t-T_0)} \end{cases} \quad (\text{Equation 2})$$

The vectors of Investment Strategy are as follow.

$$u_1^*(t) = \frac{a}{\eta}\xi x^*(T_0)e^{b(t-T_0)} \quad (\text{Equation 3})$$

$$u_2^*(t) = \frac{c}{\gamma}\beta y^*(T_0)e^{d(t-T_0)} \quad (\text{Equation 4})$$

From Equation 3, we find that $\frac{a}{\eta}\xi x^*(T_0)$ is the number of Bots that are being deployed by Attacker before DDoS attack.（namely, at *T0*）. This amount is directly proportional to the total number of Bots $x^*(T_0)$. Optimal deployment grows exponentially. Deployment of Bots will occur only gradually, much similar to a rush time on the internet. In this way, in the first day, Bots deployed can be described by $\frac{a}{\eta}\xi x^*(T_0)e^{-bT_0}$, similarly, in the second day it deployed $\frac{a}{\eta}\xi x^*(T_0)e^{-b(1-T_0)}$, Using this principle, Equation 4 can be explained in the same way. From Equation 1 to 4, we can derive an expression for the optimal deployment of Bots for both Attacker and Defender.

$$x^*(t) = x_0e^{-bt} + \frac{a^2\xi}{b\eta}x^*(T_0)e^{-bT_0}\sinh(bt) \quad (\text{Equation 5})$$

$$y^*(t) = y_0e^{-dt} + \frac{c^2\beta}{d\gamma}y^*(T_0)e^{-dT_0}\sinh(dt) \quad (\text{Equation 6})$$

Equation 5 highlights that, with time elapsing, on one hand, the Bots originally used by Attacker (x0) is depleting exponentially, while on the other hand, it is being complemented, which can be described by a sinh law. Furthermore, it is directly proportional to the number of Bots which is the ultimate expectation. In a very long time (for high values of *t*), the rate of change for $x^*(t)$ is close to $a^2\xi x^*(T_0)e^{b(1-T_0)}/(2b\eta)$, namely, the number of Bots grow exponentially. And it is directly proportional to the number of Bots of ultimate expectation $x^*(T_0)$ .With Equation 1 to Equation 6, we can conclude that the equilibrium value of Attacker and Defender Bots:

$$v_1 = J_1(u_1^*, u_2^*) = -\frac{\xi}{2}(x_0 e^{-bT_0} + \frac{a^2\xi}{b\eta} x^*(T_0) e^{-bT_0} \sinh(bt))^2 + \frac{(a\xi x^*(T_0))^2}{4b\eta}(1 - e^{-2bT_0})$$

$$v_2 = J_2(u_1^*, u_2^*) = -\frac{\beta}{2}(y_0 e^{-dT_0} + \frac{c^2\beta}{d\gamma} y^*(T_0) e^{-dT_0} \sinh(dt))^2 + \frac{(c\beta y^*(T_0))^2}{4d\gamma}(1 - e^{-2dT_0})$$

From Equation 5 and 6 we can derive:

$$\begin{cases} x^*(T_0) = \dfrac{b\eta x_0}{b\eta e^{-bT_0} - a^2\xi \sinh(bT_0)} \\ y^*(T_0) = \dfrac{b\eta y_0}{d\gamma e^{-dT_0} - c^2\beta \sinh(dT_0)} \end{cases}$$

Such that

$$u_1^*(t) = \frac{ab\xi x_0}{b\eta e^{bT_0} - a^2\xi \sinh(bT_0)} e^{b(t-T_0)}$$

$$u_2^*(t) = \frac{db\eta y_0}{d\gamma e^{-dT_0} - c^2\beta \sinh(dT_0)} e^{d(t-T_0)}$$

Obviously,

$$\left.\frac{\partial H_1(x^*, y^*, u_1, u_2^*, \lambda_1)}{\partial u_1}\right|_{u_1 = u_1^*} = \eta > 0$$

$$\left.\frac{\partial H_2(x^*, y^*, u_1^*, u_2, \lambda_2)}{\partial u_2}\right|_{u_2 = u_2^*} = \gamma > 0$$

Positively, $(u_1^*(t), u_2^*(t))$ is in an Open-Loop Nash Equilibrium. analogously，we conclude that

$$x^*(T_0) = x_0 e^{-bt} + \frac{a^2\xi x_0 \sinh(bT_0)}{b\eta e^{-bT_0} - a^2\xi e^{bT_0} \sinh(bT_0)}$$

$$y^*(T_0) = y_0 e^{-dt} + \frac{c^2\beta y_0 \sinh(dT_0)}{d\gamma e^{-dT_0} - c^2\beta e^{dT_0} \sinh(dT_0)}$$

$$v_1 = -\frac{\xi}{2}\left[\frac{b\eta x_0}{b\eta e^{bT_0} - a^2\xi \sinh(bT_0)}\right]^2 \left[-1 + \frac{a^2\xi}{2b\eta}(1 - e^{-2bT_0})\right]$$

$$v_2 = -\frac{\beta}{2}\left[\frac{d\gamma y_0}{d\gamma e^{dT_0} - c^2\beta \sinh(dT_0)}\right]^2 \left[-1 + \frac{c^2\beta}{2d\gamma}(1 - e^{-2dT_0})\right]$$

During the DDoS attacks, the Defender considers the number of Bots controlled by the Attacker, at the same time, The Defender configures resources by the speed of worm propagation for the best profit.

4 Performance evaluation and comparison

For evaluating our system, we analyze Net-Worm.Win32.Dasher as an example. The W32/Dasher-B worm exploits the vulnerability in Microsoft Windows Distributed Transaction Coordinator (MSDTC), first announced by Microsoft in October. The worm opens a backdoor on vulnerable computers and causes them to connect to a remote server for further instructions. Windows 2000 computers which have not been patched to solve this exploit are most vulnerable for this worm.

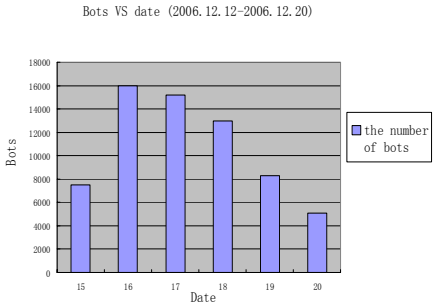

Figure 2. The number of Bots and date

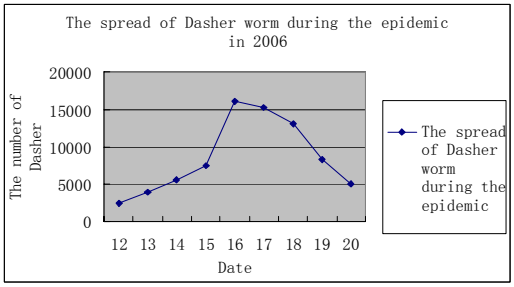

Figure 3. The spread of Dasher worm

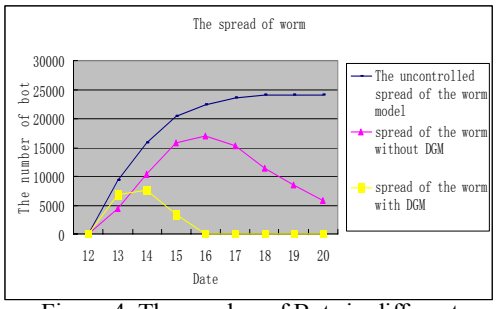

Figure 4. The number of Bots in different model-

If the Defender wants to predominate in

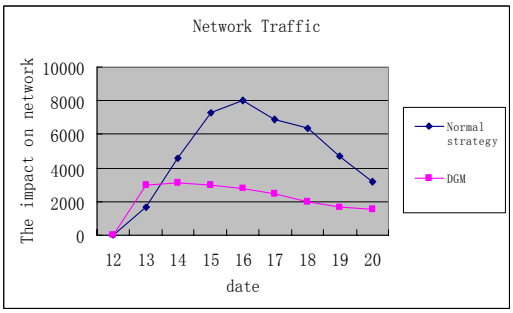

Figure 5. Infection to network

counter-DDoS, he must annihilate the number of Bots below equilibrium. Then it can effectively control the Attacker. If Defender can not control the growth of Bots which are controlled by Attacker, then Bots will proliferate like the 16th. We use ns2[9] to simulate worm, the version 2.27 of ns2 contains the worm applications. With simple changes of ns2, we can simulate the spread of the worm Dasher. For a better explanation of the DGM, we predigest Dasher without influencing the evaluation of our model. We suppose that worm Dasher only sent UDP packages. We modify the bandwidth to simulate the Competition of worms, and the number of infections conquered by worms. For the whole network, we simulate China Education and Research Network, CERNET. In our model, the smallest network cell is a campus network, and the whole network model comprised by 500 campus networks. For the assignment of IP addresses to the network cells, we refer to CERNET. At first, we simulate the spread of Dasher. As shown in Figure 3. It can preferably simulate the spread of Dasher worm during the epidemic in 2006 (Figure 2). For each spread, we examined the impact on the networks. From figure 4, we can see, at the start of each outbreak, the number of Bots, using the DGM is a little more, but in the long run, the total Bots are much less, as compared to other methods. In the every beginning, DGM impacts network more than other models, but in a long-term, it is better than normal strategy (Figure 5).

Our experiments show that our DGM is a useful way to fight off worm infections and control the Botnet. The DGM results in a lower load on the network.

5 Conclusions

The defense mechanism of DDoS attacks, particularly the multi-based, multi-approached and diversified flow method of offensive artifice, simulating the competition of legal users, inhabits a keystone and difficulty in the internet security arena, especially for the attacker using lots of Bots. This paper discusses and implements the use of the Differential Game Model (DGM) to compete with an Attacker.

Our mechanism is characteristically distinct from current methods in the following ways:

(1)It utilizes few resources and does not require participation from all ISP routers, while saving the high burden of DDoS firewalls.

(2) The method is cost-effective, while resulting in a high performance. Furthermore, it is easy to deploy.

(3) The impact of the model on the network is only minor, while the survival of the server during a DDoS attack is greatly improved.

Our future work will focus on other issues in our DGM implementation. One challenging issue for us is to reduce a large number of Bots controlled by Botnet.

References

[1]Jelena Mirkovic,Sven Dietrich,avid Dittrich, Peter Reiher,Internet Denial of Service: Attack and Defense Mechanisms,Prentice Hall PTR,December 30, 2004,1－400

[2] Siris VA, Papagalou F. Application of anomaly

detection algorithms for detecting SYN flooding attacks In: Regency H, ed. Global Telecommunications Conf. (GLOBECOM 2004). Dallas: IEEE, 2004. 2050-2054.

[3] Isaacs, R., Differential Games, John Wiley and Sons, New York, New York, 1965.

[4] Starr, A. W., and Ho, Y. C., Nonzero-Sum Differential Games, Journal of Optimization Theory and Application, Vol. 3, pp.184-206, 1969.

[5] Michael Littman and Peter Stone. A polynomial-time nash equilibrium algorithm for repeated games. ACMEC, San Diego, CA, 2003.

[6] John Nash. Equilibrium points in n-person games. Proc.of the National Academy of Sciences, 36:48–49, 1950.

[7] Nash, J. F. Theory of Games and Economic Behavior (1951) Ph.D. thesis (Princeton University, Princeton).

[8]Sun, Y.Z. Liu, Q.J. Song, Y.H. Shen, T.L. Hamiltonian modelling and nonlinear disturbance attenuation controlof TCSC for improving power system stability. Control Theory and Applications, IEE Proceedings 278- 284

[9] Kevin Fall, Kannan Varadhan, The ns Manual. The VINT Project A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. 2007