Defending Against Null Calls Stream Attacks by Using a Double-Threshold Dynamic Filter

Haizhi Xu* Changwei Cui** Ying Lin*** Tiejian Luo* Zhanqiu Dong*
** Graduate School, USTC, Beijing, 100039 xuhaizhi@263.net*
*** Beijing Institute of Radio Measurement, Beijing 100854*
**** CTC Communication Development Co. Ltd., Beijing 100088*

Key words: ATM, NCSA attack, Double-threshold filter

Abstract: In this paper, A NCSA (Null Calls Stream Attack) attack that can happen in ATM control plane is pointed out. After analyzing its characteristics, a risk-cost-based solution — double-threshold dynamic filter — is brought forward. The analytical expression and algorithm of the threshold setting are also given. By implementing the algorithm on ATM access equipments, the NCSA attack can be blocked effectively. Finally, the solution is validated in a simulated environment.

1. INTRODUCTION

The Denial-of-service attack is a typical attack in the communication networks. By occupying network resources illegally, it can consume a substantial proportion of network resources. The result is the network's denial of service. Combined with other attacks, the attack will cause worse threats to networks. This paper will discuss a denial-of-service attack in the ATM network — NCSA (Null Calls Stream Attack) attack — and bring forward a risk-cost-based solution — double-threshold dynamic filter — to defend against it.

Although there are not many reports on NCSA attacks in the ATM network, TCP SYN Flooding, a similar Internet attack, is discussed a lot. There is no complete solution for this problem yet, but there are steps that can be taken to lessen its impact [1][2]. The defense for a TCP SYN Flooding attack is mainly to set up packet filters before the TCP connection

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*

end points. The idea of threshold-based packet filter is discussed in [3][4][5][6]. Threshold setting can be based on practical monitoring, while in this way, threshold setting lacks theoretical support and the required security is hard to obtain. A learning algorithm in threshold setting is introduced in [6], but if the attacker increases the TCP SYN Flood gradually, the attack can pass the threshold, which leads to the learning method failure.

According to the difference in null calls streams intensity, we can probabilistically distinguish attacks from normal null calls streams. This method overcomes the vulnerability in the learning algorithm mentioned above. By setting the double-threshold dynamic filter in the ATM access switch, we can effectively block the NCSA Attacks.

This paper first identifies the NCSA attack in the ATM control plane. Then its characteristics are analysed. In the third part, we bring forward a risk-cost-based solution — double-threshold dynamic filter — and give the analytical expression and algorithm of the threshold setting. In the end, the algorithm is validated in a simulated environment.

2. DESCRIPTION OF THE NCSA ATTACK AND ITS CHARACTERISTICS

The ATM is a kind of connection oriented network. According to the ways connections are set, ATM connections can be classified as SVC and PVC. Comparing with PVC, SVC has the merits of flexibility, resources allocation on demand, etc. The NCSA attack in this paper aims at the SVC system. My research work is done on the UNI3.1 protocol[7].

Definition 1. The connection is released immediately before or after the connection is setup successfully. We name such setup requests in the ATM network *Null Calls*. There is no data transfer in the connections established by null calls.

2.1 The description of the NCSA attack

In the normal case, in UNI3.1, the calling end point begins setting up a connection by sending a SETUP message to the called side by the signaling channel. Then the called side acknowledges the SETUP by sending a CONNECT message to the calling side. The calling side then finishes establishing the connection by responding with a CONNECT_ACK message to the neighboring switch. The connection is then open, and data can be exchanged in the virtual channel that is negotiated in the messages.

In the case of attack, after receiving the CONNECT message, the calling side sends out a RELEASE message instead of CONNECT_ACK, or the calling side sends out the RELEASE message before receiving the

CONNECT. The resources of the network and called side (including bandwidth and connection related data structures) are occupied temporarily. Because the resources of the called side, especially of a small ATM terminal, are very limited, frequent null calls will consume most or all of the resources of the called side. The victim of such an attack will have difficulty in accepting any new incoming network connections unless the resources are released. Normally, the called side will release the related resources when it receives a RELEASE message, but the calling side (the attacker) can send out more SETUP messages to occupy the resources. Thus, frequent null calls will lead to the resources occupation ratio near or reach 100% over a period of time. In this case, the attack does not affect existing connections. However, in some cases, the attacked system may exhaust memory, crash, or be rendered otherwise inoperative. Figure 1 illustrates a possible NCSA attack.

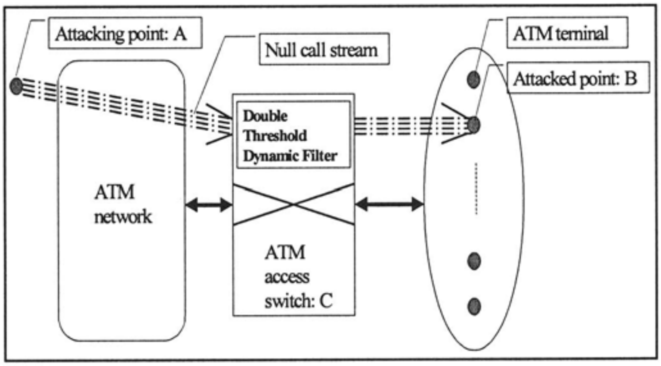

Figure 1. Illustration of a possible NCSA attack

2.2 The characteristics of the NCSA attack

In this paper, we assume:

- The attacker does not arbitrarily revise the ATM address in its own node.
- The attacker does not arbitrarily revise the calling and the called ATM address in the SETUP request.
- A non-attacker may send out very frequent null calls, but the probability of doing so is very small.

Under these assumptions, this paper mainly discusses the NCSA attack with true source address in the SETUP messages. This kind of attack has the following characteristics:

Characteristic 1. The arrival of null calls is frequent *or* the network resources are heavily occupied by null calls.

Characteristic 2. Null calls with characteristic 1 have the same calling address and called address in the SETUP messages.

3. THE DOUBLE-THRESHOLD DYNAMIC FILTER

The basic idea of defending against such attacks is to set up filers. Filters in [3] and [6] are not very satisfying. Here, we propose a filter solution based on risk cost. To the ATM access switch C in figure 1, we implement a double-threshold filter algorithm. By setting up a double-threshold, the algorithm dynamically generates a refusal list. A filter in the call control module filters received SETUP requests according to the list. If the pair of calling address and the called address matches an item in the list, we take the SETUP request as an attack and refuse it without further normal processing; otherwise, the request is to be processed normally.

3.1 The principle and the method of double-threshold setting

3.1.1 The principle of double-threshold setting

Based on the nature of Poisson distribution, our monitoring the network and simulating the NCSA attack, in this paper, we assume that the non-attacking null calls streamξ_0 and the attacking null calls streamξ_1 between nodes A and B obey Poisson distributions with parametersλ_0 andλ_1 respectively.

Guess an arrived null call (we call an arrived null call an *event* in the rest of the paper) as:

1. The event is an attack (H_0);
2. The event is not an attack (H_1);

If the possibilities of the above events are $P(H_0)$, $P(H_1)$ respectively, given a stochastic variant X, according to the Bayes rule[8],

$$P(H_i \mid X) > P(H_j \mid X) \Rightarrow X \in H_i, \quad \text{in which } i, j \in \{0,1\}, i + j = 1. \tag{3.1}$$

In this case, we judge H_i about X is right.

Since the judgement is based on probability, the following 4 situations might happen:

1) Correct warning
2) False warning

3) Failed warning
4) Correct non-warning

Definition 2. *Correct warning* is the case in which the event is an attack and it is judged as an attack. *Correct warning probability* is the probability of correct warning.

Definition 3. *False warning* is the case in which the event is not an attack and it is judged as an attack. *False warning probability* is the probability of false warning.

Definition 4. *Failed warning* is the case in which the event is an attack and it is judged as a non-attack. *Failed warning probability* is the probability of failed warning.

Definition 5. *Correct non-warning* is the case in which the event is not an attack and it is judged as a non-attack. *Correct non-warning probability* is the probability of correct non-warning.

Definition 6. The *Risk Cost Weight* (RCW) is the weight of the risk engendered by the above 4 warnings. C_{00}, C_{01}, C_{10}, C_{11} stand for the RCW of correct warning, false warning, failed warning, correct non-warning respectively. $C_{ij} \in [0,1]$, $i, j \in \{0,1\}$. The larger the risk, the higher the cost.

Suppose the false warning probability is P_f and the failed warning probability is P_l (In figure 2, we illustrated the false warning probability P_f and the failed warning probability P_l in the distribution graph of ξ_0 andξ_1.). Then the overall error probability is:

$$P_e = P_f P(H_0) + P_l\, P(H_1)$$

in which, $P_l = \int_{-\infty}^{V_1} f_{\xi_1}(x)dx$,

$$P_f = \int_{V_0}^{+\infty} f_{\xi_0}(x)dx \qquad (3.2)$$

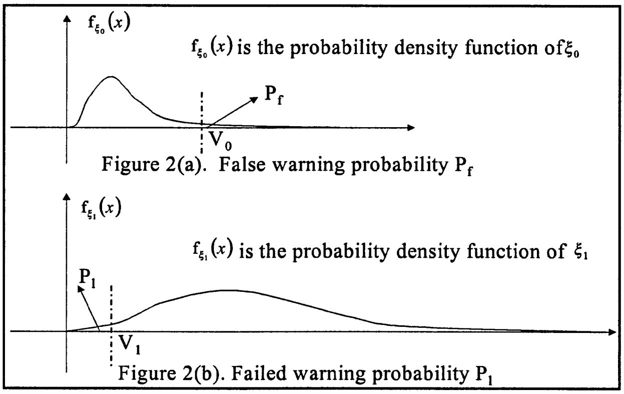

Figure 2(a). False warning probability P_f

Figure 2(b). Failed warning probability P_l

Figure 2. P_f and P_l in the distribution graph of ξ_0 andξ_1 respectively

Definition 7. *Risk cost* is the cost generated by correct warning, false warning, failed warning and correct non-warning. D represents risk cost.

$$\begin{aligned} D &= (C_{10}P_f + C_{00}(1-P_f))P(H_0) \\ &\quad + (C_{01}P_l + C_{11}(1-P_l))P(H_1) \end{aligned} \tag{3.3}$$

Obviously, threshold choosing should minimize the risk cost. In this paper, correct warning and correct non-warning are correct judgements. Their risk cost is the least. Let $C_{00}= C_{11}=0$ simplifying formula (3.3), we get

$$D = C_{10}P_fP(H_0) + C_{01}P_lP(H_1) \tag{3.4}$$

The least risk Bayes judgement is to minimize D in formula (3.4).

If there is not enough knowledge to give reasonable C_{00}, C_{01}, C_{10} and C_{11}, according to the Neyman-Pearson rule[8], generally a permissible false warning probability is given to make the discovery probability (1- P_l) the largest, namely:

$P_f = C$, to let P_l smallest.

To simplify D, let $K_1 = C_{10}$ P(H_0), $K_2= C_{01}$ P(H_1), then:

$$D = K_1P_f + K_2P_l \tag{3.5}$$

Remembering P_f and P_l in formula (3.2), we have:

$$\begin{aligned} D &= K_1 \int_{V_0}^{+\infty} f_{\xi_0}(x)dx + K_2 \int_{-\infty}^{V_1} f_{\xi_1}(x)dx \\ &= K_2 + \int_{V_0}^{+\infty} \left[K_1 \cdot f_{\xi_0}(x) - K_2 \cdot f_{\xi_1}(x)\right]dx \\ &\quad - K_2 \int_{V_1}^{V_0} f_{\xi_1}(x)dx \end{aligned} \tag{3.6}$$

So, the Neyman-Pearson rule comes down to finding ideal thresholds V_0 and V_1. From formula (3.2), we know that if P_f and P_l are given, V_0 and V_1 can be found.

3.1.2 Setting the thresholds

Consider that normal call requests are a kind of real-time activity. In the case of non-attack, existence of null calls between A and B is related to the SETUP request success ratio. In practice, if a SETUP request is not met in 2-3 seconds, the call might be released and a new SETUP request be sent out. In this case, suppose a null call is generated every 1.5 seconds, remembering the assumption in 3.1.1 that the non-attacking null calls streamξ_0 between nodes A and B obeys Poisson distribution with parametersλ_0, we have:

$$P_{\xi_0}\{X=m\}=\frac{\lambda_0^m}{m!}e^{-\lambda_0} \tag{3.7}$$

in which, $\lambda_0 = E(X) \approx 0.7$

$$P_f=\sum_{k=v_0}^{+\infty}\frac{\lambda_0^k}{k!}e^{-\lambda_0} \tag{3.8}$$

False warning probability can be decided according to network monitoring and security requirements of the protected ATM terminals. Given P_f and formula (3.8), V_0 can be looked up in the accumulating probability table of the Poisson distribution. In a period of time, 1 second for instance, if the frequency of the arrived events (null calls) from A to B $X_{AB}>V_0$, we judge it as attack from A to B.

Similarly, suppose every SETUP in the attacking stream requests CBR bandwidth *d* of the whole bandwidth of the attacked point, and the attacking intensity (the frequency of events happened) ≥1/d, we get:

$$P_{\xi_1}\{X=m\}=\frac{\lambda_1^m}{m!}e^{-\lambda_1}\text{, in which }\lambda_1\geq\frac{1}{d} \tag{3.9}$$

$$P_1=\sum_{k=0}^{v_1}\frac{\lambda_1^k}{k!}e^{-\lambda_1} \tag{3.10}$$

Failed warning probability can also be decided according to network monitoring and security requirements of the protected ATM terminals. Given P_l and formula (3.10), V_1 can be looked up in the accumulating probability table of the Poisson distribution. In a period of time, if the frequency of the arrived events (null calls) from A to B $X_{AB}<V_1$, we judge it as non-attack from A to B.

3.2 The double-threshold filter algorithm

Considering one sampling can not accurately reflect a Poisson process, we successively sample *m* times the stochastic variant X_{AB}——the arrival frequency of null calls from A to B in unit period of time. We make ajudgement by its arithmetical expectation E (X_{AB}). Using the maximum similarity estimation method[8], we can deduce the parameter λ of the Poisson distribution, and:

$$\mathrm{E}(X_{\mathrm{AB}}) = \lambda \approx \overline{X}_{AB} \tag{3.11}$$

As described in 3.1.2, given P_f , P_l and formula (3.8), (3.10), V_0 and V_1 can be looked up in the accumulating probability table of the Poisson distribution.

The double-threshold filter algorithm can be described as consisting of the following 2 parts:

Part 1: Generating a refusal list.

If $\mathrm{E}(Z_j) \geq V_0$, we judge it as H_0. And we add the pair A and B to the refusal list;

If $\mathrm{E}(Z_j) \leq V_1$, we judge it as H_1. And we withdraw the pair A and B from the refusal list if the pair is already in the list;

If $\mathrm{E}(Z_j) \in (V_1, V_0)$, then we continue our judgement by getting the next sample, where Z_j is the frequency of the events happened in the *j*th sampling period.

Considering the error between theory calculation and practice, the thresholds may be revised according to the practical monitoring. We may replace V_0 and V_1 with $V_0' = V_0 + C_{V_0}$ and $V_1' = V_1 + C_{V_1}$ respectively, where, C_{V_0} and C_{V_1} are revised values.

The refusal list is null at the beginning of running. The algorithm will generate a refusal list dynamically.

Part 2: Filtering incoming calls.

A filter in call control module filters received SETUP requests according to the list. If the pair of calling address and the called address matches an item in the list, the SETUP request is refused without further normal process, otherwise, the request is to be processed normally.

4. THE EXPERIMENT

Constructing an experimental environment illustrated in figure 1, we simulated the NCSA attack from node A to node B in the call control layer. We built a double-threshold dynamic filter in the access switch C.

To reduce the space usage in the algorithm, let sampling period be 1 second and m=6. Let P_f=0.05 and P_l=0.05, we deduce the threshold values

V_0=4, V_1=1. Note V_0 and V_1 can be tuned according to the practical conditions.

Node A simulates NCSA attack on node B. Every thread of the 100 attacking threads in node A generates a Binary value—r_B per second with probability of P_{r_B} =0.25. If the value $r_B = 1$, the thread sends out a null call; if $r_B = 0$, the thread sleeps for 1 second before beginning a new Binary experiment. Every null call applies for CBR bandwidth 1M, 1/25 of the whole 25M bandwidth of the attacked node B. In the experiment of approximately 1000 seconds, node A attacks by sending out 65914 null calls. The distribution of the attack is illustrated by figure 3.

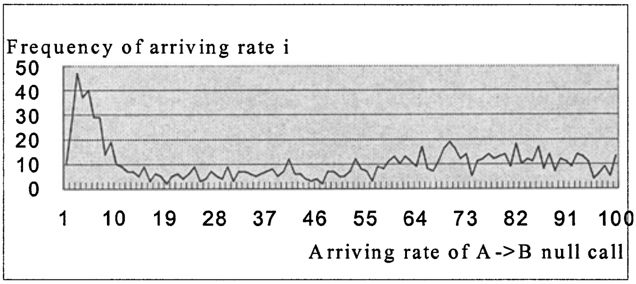

Figure 3. A->B null calls arriving rate distribution viewed at switch C

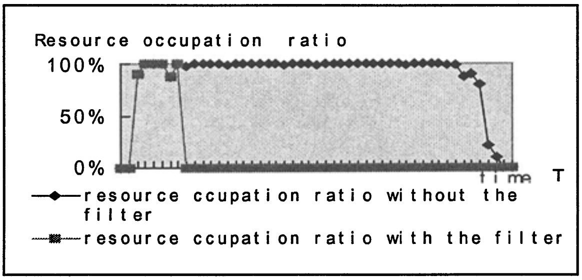

Figure 4. Comparison of the resource occupation ratios of attacked point B with and without the filter

The resource occupation ratios of node B before and after installing the filter are illustrated in figure 4. The experimental result indicates that the double-threshold dynamic filter in the access switch can effectively block the NCSA attacks with true addresses in the SETUP messages.

The double-threshold filter algorithm is not very hard to implement. We implemented it in about 500 lines C source code. And the algorithm does not take much time and space to calculate the refusal list and filter the incoming calls.

5. CONCLUSION

This paper identified the NCSA attack in ATM control plane and brought forward a kind of threshold filter solution to defend it. We have shown the principle of threshold setting and the filtering algorithm. The solution and the algorithm are validated in a simulated environment. The experimental result indicates that the double-threshold dynamic filter in the access switch can effectively block NCSA attacks with true addresses in the SETUP message. Attacks with spoofing addresses will be discussed later.

The solution, double-threshold dynamic filter, is not only useful to defend against NCSA attacks in ATM networks, but also meaningful to block TCP SYN Flooding attacks in the Internet.

REFERENCE

1. CERT Advisory CA-96.21*. TCP SYN Flooding and IP Spoofing Attacks (revised), http://www.cert.org/, 1998.8.
2. CERT Advisory CA-96.21. TCP SYN Flooding and IP Spoofing Attacks, 1996.9.
3. P. Ferguson. Network Ingress Filtering: Defeating Denial-of-service Attacks which employ IP Source Address Spoofing, RFC2267. 1998.1.
4. Anderson, D. et al.: SAFEGUARD FINAL REPORT - Detecting unusual behaviour using the NIDES statistical component, SRI International, 1993.
5. Anderson, D., Frivold, T. and Valdes A. Next Generation Intrusion Detection Expert-System (NIDES) - A Summary, Technical Report SRI-CSL-95-07, SRI International, 1995.
6. Anderson, D. et al.: Detecting Unusual Program Behaviour Using the Statistical Component of NIDES, Technical Report SRI-CSL-95-06, SRI International, 1995.
7. The ATM Forum Technical Committee: "User-Network Interface (UNI) Specification, Version 3.1".
8. Zhou Gairong. Probability and Statistic. Advanced Education Publisher. 1984.3.
9. Haizhi Xu, et.al.: A D&C Mechanism to Solve the PNNI Topology Information Conflicting Problem, accepted for presentation at the SEC2000 conference of the WCC2000 conference, Beijing, 2000.8.