Finite State Machine Detection Model Based on Adaptive TTL Neuron

1Zhang Jun and 2Gao Lei

1Institute of Network, School of Electronics and Information Engineering, Xi’an Jiaotong University, Xi’an, Shaanxi, 710049, People’s Republic of China
2Huawei 3Com Technologies Co. Ltd, Beijing, 100085, People’s Republic of China

Abstract: Without detailed knowledge of the network topology between Network Security Audit System (NSAS) and the end system, NSAS may be unable to determine whether a given packet will even be seen by the end system. Finite state machine detection model based on adaptive TTL neuron is proposed to solve it in this paper. By inspecting the variance of Time To Live (TTL) field in ingress sequence, the adaptive TTL neuron compares the TTL field of new fragment with the average TTL to detect whether the TTL field of new fragment is normal or not. Combining the TTL neuron with the detection state node of finite state machine detection model, ingress flow will be benign and NSAS won’t be confused anymore.

Keywords: Time To Live (TTL), Adaptive TTL neuron, finite state machine, insertion, evasion

INTRODUCTION

Network security audit system (NSAS) just like IDS or content-based audit system (Cao et al., 2002) protects end systems of Intranet online. By exploiting ambiguities in the traffic stream as seen by NSAS, the ability of a skilled attacker can do insertion and evasion (IE) attack (Ptacek and Newsham, 1998; Patton et al., 2001) to end systems. NSAS can accept a packet that an end system rejects, that’s insertion attack; an end system can accept a packet that an NSAS rejects, that’s evasion attack. In the absence of external knowledge (end system implementation details, topology details), exploitable ambiguities can arise in two different ways (Handley et al., 2001): (1) Without detailed knowledge of the end system’s protocol implementation, NSAS may be unable to determine how the end system will treat a given sequence of packets if different implementations interpret the same stream of packets in different ways. Unfortunately, Internet protocol specifications do not always accurately specify the complete behavior of protocols, especially for rare or exceptional conditions. In addition, different operating systems and applications implement different subsets of the protocols. (2) Without detailed knowledge of the network topology between the NSAS and the end system, NSAS may be unable to determine whether a given packet will even be seen by the end system. For example, a packet seen by NSAS that has a low Time To Live (TTL) field may or may not have sufficient hop count remaining to make it all the way to the end system. Figure 1 for an example.

![Fig. 1: IE attack example](image)

In this paper, finite state machine detection model based on adaptive TTL neuron is proposed to detect the latter case. Counting TTL average of ingress as TTL in each connection, the adaptive TTL neuron compares the TTL field of new fragment with the average TTL to detect whether the TTL field of new fragment is normal or not. Combining the TTL neuron with the detection state node of the TTL finite state machine detection model, ingress flow will be benign after filtering and NSAS will no more be confused by the IE attack based on TTL.

Corresponding Author: Zhang Jun, Institute of Network, School of Electronics and Information Engineering, Xi’an Jiaotong University, Xi’an, Shaanxi, 710049, People’s Republic of China
ADAPTIVE TTL NEURON

In the adaptive TTL neuron, the TTL average of ingress fragments as \( \overline{TTL} \) in each connection is counted for the comparison of the TTL field of new fragment to detect whether the ingress is normal or not in TTL field. The result is FALSE for normal flow and TRUE for that it seems to be IE attack.

Each connection of the ingress flow is denoted as tuple \((ID, S_{ip}, S_{port}, D_{ip}, D_{port}, C, \overline{TTL})\) for the incoming fragments. ID is the identification field in the IP header. \(S_{ip}\) is the source IP address. \(S_{port}\) is the source port. \(D_{ip}\) is the destination IP address. \(D_{port}\) is the destination port. \(C\) is total packet which is counted for the connection. \(\overline{TTL}\) is the TTL average of ingress for the connection. Let \(D\) denote the depth of Intranet.

For every tuple, TTL is initialized to be the TTL of first fragment and \(C\) is initialized to be zero. The adaptive TTL neuron runs as follows for a new packet:

**Step 1:** For the new packet, \(\frac{\overline{TTL} \cdot C + TTL_{new}}{C + 1}\), \(C = C + 1\), \(TTL_{new}\) is the TTL field of the new packet.

**Step 2:** If \(TTL_{new} \in \mathbb{Z}^{+}\), it's normal, then \(\overline{TTL} = \overline{TTL} + \alpha\), the result is FALSE, else go to step 3. \(\alpha\) is a given parameter, \(\alpha = \text{MIN}(D/2, 2.5)\).

**Step 3:** The ingress may be IE attack, the TTL neuron get result TRUE.

Following above steps, the packet in abnormal TTL field will be detected. The adaptive TTL neuron will get result TRUE for detection of IE attack and FALSE for benign packet.

FINITE STATE MACHINE

Symbol meaning: We denote the finite state machine as FSM = \([I, O, S, \delta, s_0]\):

- **I**: input symbol set
- **O**: output symbol set
- **S**: state set
- **\(\delta\)**: \(S \times I \rightarrow S\) state transition function
- **\(s_0\)**: initial state

With the definition of the finite state machine, the detection model is shown in Fig. 2. The meaning of the parameter in the finite state machine detection model is the following:

(1) Input symbol set \(I = \{X, Y, Z\}\)

Every element of \(I\) has two inputs: One is the \(OR\) function result of the more fragment bit of TTL field and

Fig. 2: Finite state machine detection model

the Fragment Offset field in the IP header, another is the result of TTL neuron. Different inputs have the following cases (the former is the OR function result, the latter is the result of TTL neuron, \(s\) means do not care):

\[X(0, x)\]: The packet isn't fragment, there's no IE attack.
\[Y(1, 0)\]: The packet is fragment, but there's no IE attack.
\[Z(1, 1)\]: The packet is fragment and it seems to be IE attack.

(2) Stat set \(S = \{s_0, s_1, s_2, s_3\}\)

- \(s_0\): detection state
- \(s_1\): the ingress is normal
- \(s_2\): seem to be IE attack

(3) State transition function \(\delta\): \(S \times I \rightarrow I\):

\[
\begin{align*}
\delta(s_0, X) &= s_1 \\
\delta(s_0, Y) &= s_1 \\
\delta(s_0, Z) &= s_1 \\
\delta(s_1, Y) &= s_1 \\
\delta(s_1, Z) &= s_3 \\
\delta(s_2, Y) &= s_1 \\
\delta(s_2, Z) &= s_3 \\
\delta(s_1, X) &= s_1 \\
\delta(s_1, Y) &= s_1 \\
\delta(s_1, Z) &= s_1 \\
\end{align*}
\]

Detection method: Packet passes the initial state \(s_0\) to start detection. If the packet isn't fragment, that will not be IE attack and turn to be state \(s_1\), else turn to be state \(s_2\) for next detection. In state \(s_0\), the fragment will be submitted to state \(s_3\) for format formalization if the IE attack is detected. In state \(s_0\), TTL field of the fragment is modified to be \(\overline{TTL}\). Following is the detail of each state:

(1) \(s_0\): In this state, the finite state machine classifies the packet to be fragment or not by the Flags field and Fragment Offset field of the IP header. In the three bit of Flags field (IETF, 1981), bit 0 is reserved, bit 1 (DF) means don't fragment in value 1 and may fragment in value 0, bit 2 (MF) means there're more fragments in value 1 and to be the last fragment in value 0. Value (0, 1) is get from the OR function of MF and Fragment Offset. FALSE means the packet isn't a fragment, then turn to \(s_1\); and TRUE means it does, then turn to \(s_2\).
CONCLUSIONS

In this study finite state machine detection model based on adaptive TTL neuron is proposed to detect IE attack based on TTL. By inspecting the variance of TTL field in ingress sequence, the adaptive TTL neuron compares the TTL field of new fragment with the average TTL to detect whether the TTL field of new fragment is normal or not. Combining the TTL neuron with the detection state node of finite state machine detection model, ingress flow will be benign and NSAS won't be confused by the IE attack anymore.

REFERENCES


Fig. 3: Detection process