A SYN flooding attack detection approach with hierarchical policies based on self-information

Jia-Rong Sun | Chin-Tser Huang | Min-Shiang Hwang

1Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan
2Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA
3Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan

Correspondence
Min-Shiang Hwang, Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan.
Email: mshwang@asia.edu.tw

Funding information
Ministry of Science and Technology, Taiwan (ROC); Ministry of Science and Technology, Grant/Award Numbers: MOST 108-2622-8-468-001-TM1, MOST 108-2410-H-468-023, MOST 109-2221-E-468-011-MY3

Abstract
The SYN flooding attack is widely used in cyber attacks because it paralyzes the network by causing the system and bandwidth resources to be exhausted. This paper proposed a self-information approach for detecting the SYN flooding attack and provided a detection algorithm with a hierarchical policy on a detection time domain. Compared with other detection methods of entropy measurement, the proposed approach is more efficient in detecting the SYN flooding attack, providing low misjudgment, hierarchical detection policy, and low time complexity. Furthermore, we proposed a detection algorithm with limiting system resources. Thus, the time complexity of our approach is only \((\log n)\) with lower time complexity and misjudgment rate than other approaches. Therefore, the approach can detect the denial-of-service/distributed denial-of-service attacks and prevent SYN flooding attacks.

KEYWORDS
denial-of-service (DoS) attack, self-information, SYN flooding attack

1 INTRODUCTION

The SYN flooding attack [1] is a method widely used in network attacks; it is similar to the denial-of-service (DoS) [2] and distributed denial-of-service (DDoS) attacks [3,4]. This attack paralyzes server systems within a short time. The SYN flooding attack, based on transmission control protocol (TCP) [5], is used to exhaust server resources. When the TCP connection is established, three steps must be completed: the three-way handshake [5]. The SYN flooding attack is used to stop the three-way handshake, causing the exhaustion of server resources. So far, many detection methods [6–15] have been proposed to prevent and detect SYN flooding, DoS, and DDoS attacks.

Many SYN flooding detection approaches have been proposed [16–18]. Wang et al. proposed an SYN flooding detection installed at leaf router, flooding detection system. It has these features: stateless and low computation overhead, insensitive to the site, and not undermining the end-to-end TCP performance [16]. Siris and Papaglou analyzed two detection algorithms: an adaptive threshold algorithm and an algorithm based on the cumulative sum (CUSUM) change point detection scheme [17]. They described an adaptive threshold algorithm with high-intensity attack satisfactory performance but not great in the low intensity; the CUSUM has a robust performance and is easy to implement. Yu and others proposed a lightweight and fast detection mechanism using the management information base of simple network management...
protocol to get the raw data [18]. The scheme is aimed at fast detection, and it employs a machine learning approach using the support vector machine to classify attacks. After experiments, this detection mechanism exhibits high efficiency and the low false alarms classified.

Since SYN flooding attacks exhaust the server resources by stopping the three-way handshake, it establishes a connection. If a detection approach can calculate the abnormal and normal values from the successful connection, it can be applied to detect the SYN flooding attack. When a data packet causes an SYN flooding attack or an incomplete TCP three-way handshake, we call the data packet an abnormal packet. The opposite is a normal packet. Therefore, the approach from the integrated entropy measurement used in the e-mail server [19,20] is used to detect a flooding attack, where the packet pair definition is used to define the TCP connection. In addition to the e-mail server [19,20], many articles use the entropy method to detect SYN flooding attacks, just like SYN flooding attack detection based on entropy computing [21]. The email server is used to transmit or receive mails via SMTP, POP3, IMAP, or HTTP protocols. However, the detection approach based on entropy measurement causes the misjudgment problem, which will be described and solved in Section 4.1.

In this paper, we propose a detection approach with a hierarchical policy using packet pairs of self-information [22]. In the hierarchical policy, we define the long-time and short-time domains. The long-time domain contains several short-time domains that produce a layered architecture during our sampling process. Self-information is a method for calculating information values and the probability of occurrences. When an unknown event occurs, it contains a larger information value; otherwise, when a known event occurs, this event will contain fewer information values. Therefore, if the probability of the event occurring is low, it means the event belongs to more unknown events, and the information value will be larger after the event occurs. Otherwise, suppose that the event has a higher probability of occurrence, then the event is less unknown, and the information value will be lower after the event occurs. This method follows the three-way handshake to define packet pairs, including the normal and abnormal packets, and it uses a hierarchical policy algorithm to detect abnormal packets when SYN flooding attacks happen. Furthermore, this method provides lower time complexity, lower misjudgment, and more users than the one proposed by Chen and others [19], and it still detects half-open connections [5].

This paper considers the current management system of the server room (or data center), such as the information security management system (ISMS, ISO 27001:2013). In managing the server room, to ensure the network and host provide services, the administrators monitor the net flow, system resource usage rate of every important host, and stand by to receive the system alarm. When the administrators receive the alarm from different equipment simultaneously, they confirm the alarms individually and keep a record of the events. This course consumes the human resources of server room management. Therefore, our approach makes the resource release time point the detection time point. Once the attack occurs, the administrators confirm the alarms of a flooding attack and system usage rate simultaneously to reduce human resource consumption in the server room management.

This paper is organized as follows. Section 2 describes the detecting flooding attack using the integrated entropy measurement in a server. The details of the approach and proposed algorithm are provided in Section 3. Section 4 compares and analyzes the characteristics between [19] and our proposed approach. Finally, Section 5 presents our conclusions.

## 2 Detecting Flooding Attacks Based on Integrated Entropy Measurement

Chen and others proposed an approach for detecting a flooding attack based on integrated entropy measurement using an e-mail server [19,23]. The e-mail server defined in a previous study [19] includes simple mail transfer protocol (SMTP) [24], post office protocol version 3 (POP3) [20], internet message access protocol version 4 (IMAP4) [25], and hypertext transfer protocol secure (HTTPS) [26]. From the abovementioned e-mail protocols, the message flow has a couple of rounds including request and response messages. Here, they define an entropy pairing, including normal and abnormal packets, and calculate the entropy value for each e-mail protocol. The detection algorithm [19] of this approach is shown below:

\[
X_{K_{n,a}} \in \{x_{K_{n,a},n1}, x_{K_{n,a},n2}, \ldots, x_{K_{n,a},at}\}
\]

and

\[
X_{K_{n,d}} \in \{x_{K_{n,d},a1}, x_{K_{n,d},a2}, \ldots, x_{K_{n,d},at}\}
\]

is a random set value of normal and abnormal packets in a time \(T_n\).

\[
H(x_{K_{n,a}}) \quad \text{and} \quad H(x_{K_{n,d}})
\]

are entropy values set for normal and abnormal packets after entropy calculation; \(K \in \{SMTP, IMAP, POP3, HTTPS\}\). \(P(x_{K_{n,a}})\) and \(P(x_{K_{n,d}})\) are probabilities of normal and abnormal packets of \(K\) in \(T_n\), and \(P(x_{K_{n,a}}) + P(x_{K_{n,d}}) = 1\). The notation and definition of the detecting flooding attacks (DFA) based on the integrated entropy measurement is given in Table 1.

Algorithm 1 shows the entropy values of the normal and abnormal packets calculated during the sampling time and outputs the e-mail server status.
Algorithm 1 The detecting flooding attack based on the integrated entropy measurement

**Input:** $H(x_{K_{tw}})$ and $H(x_{K_{tw}})$ that are under $T_w$, $T_w = 1, 2, 3, ..., t$

**Output:** A cost value $C_{K_{tw}}$ of e-mail server status under $T_w$ of input, and $C_{K_{tw}} \in \{\text{CriticalHigh, VeryHigh, High, Normal, Low, VeryLow}\}$ = \{CH, VH, H, N, L, VL\}

**Begin**

$$x_{K_{tw}} \leftarrow \max \left( -\sum_{i=1}^{t} \frac{P(x_{K_{tw}})}{\log \frac{P(x_{K_{tw}})}{P(x_{K_{tw}})}} \right),$$

$$y_{K_{tw}} \leftarrow \max \left( -\sum_{i=1}^{t} \frac{P(x_{K_{tw}})}{\log \frac{P(x_{K_{tw}})}{P(x_{K_{tw}})}} \right),$$

if $(y_{K_{tw}} - x_{K_{tw}}) > 0$ then return $C_{K_{tw}}$ = CH

else if $(y_{K_{tw}} - x_{K_{tw}}) > 0 \text{ AND}$

$$\left( y_{K_{tw}} - \frac{y_{K_{tw}} + y_{K_{tw}} + y_{K_{tw}}}{3} \right) > 0$$

then return VH;

else return H;

if $(y_{K_{tw}} - x_{K_{tw}}) < 0$ then return $C_{K_{tw}}$ = VL

else if $(y_{K_{tw}} - x_{K_{tw}}) < 0 \text{ AND}$

$$\left( x_{K_{tw}} - \frac{x_{K_{tw}} + x_{K_{tw}} + x_{K_{tw}}}{3} \right) > 0$$

then return L;

else return N.

Table 1: Notation and definition of DFA

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(x_{K_{tw}})$</td>
<td>The probabilities of a normal packet of $K$ in $T_w$.</td>
</tr>
<tr>
<td>$P(x_{K_{tw}})$</td>
<td>The probabilities of an abnormal packet of $K$ in $T_w$.</td>
</tr>
<tr>
<td>$T_w$</td>
<td>A simple time domain.</td>
</tr>
<tr>
<td>$K$</td>
<td>The output of email server status.</td>
</tr>
<tr>
<td>$X_{K_{tw}}$</td>
<td>A set of random values of abnormal packets in a time $T_w$.</td>
</tr>
<tr>
<td>$X_{K_{tw}}$</td>
<td>A set of random values of normal packets in a time $T_w$.</td>
</tr>
<tr>
<td>$H(x_{K_{tw}})$</td>
<td>The entropy values set for the normal packet.</td>
</tr>
<tr>
<td>$H(x_{K_{tw}})$</td>
<td>The entropy values set for the abnormal packet.</td>
</tr>
</tbody>
</table>

Algorithm 1 starts by sampling a time domain $T_w$ and using the entropy of the normal and abnormal packets as the input values and then outputs the current server status, which is categorized as critical high, very high, high, normal, low, very low. When a packet set is sampled, $x_{K_{tw}}$ and $y_{K_{tw}}$, the difference $(y_{K_{tw}} - x_{K_{tw}})$ is calculated. If $(y_{K_{tw}} - x_{K_{tw}}) > 0$, the server status will output critical high; if $(y_{K_{tw}} - x_{K_{tw}}) > 0$, and $y_{K_{tw}}$ is more than the average of the last three times, the server status will output very high; if $(y_{K_{tw}} - x_{K_{tw}}) < 0$, the server status will output very low; if $(y_{K_{tw}} - x_{K_{tw}}) < 0$ and $x_{K_{tw}}$ is more than the average of the last three times, the server status will output low; otherwise, the server status will output the normal status.

This detection approach detects a flooding attack by calculating the entropy values of normal and abnormal packets on an e-mail server and defines the cost values of monitoring the system status of the e-mail server. Simultaneously, they reduce the misjudgment rate and detect the flooding attack quickly and easily using this detection approach and algorithm they proposed. The work process of this algorithm will be described in Section 4.1.

### 3 Detection Approach and Algorithm

Since the SYN flooding attack uses the leaks of TCP three-way handshake to attack a server, it exhausts the server system resources. Therefore, we propose an approach based on self-information for detecting an SYN flooding attack and provide a detection algorithm with a hierarchical policy on a detection time domain. Besides, we refer to the packet pair definition [23] as the TCP three-way handshake message packets and use this definition to distinguish the normal from abnormal packets in the TCP message flow.

#### 3.1 Packet pair definition

The TCP connection consists of a request packet of SYN message from a client, a response packet of SYN-ACK (Synchronize-Acknowledgment), and an ACK (Acknowledgment) from the client. Upon completing the three steps above, the TCP connection will be successful. Therefore, we define the packet pair as (Server response, Client request), which includes a response from a server and a request from a client. Since the first request packet of a client is used to establish the TCP connection, the packet pair only includes a server response of SYN-ACK and a client request of ACK.
Generally, many clients access or are connected to servers, but these clients will not log in or log out of the server simultaneously. When a server providing services to clients is under an SYN flooding attack, the system resources must be expended in a time domain as defined:

i. The complete packet pair \((\text{Server}_{\text{response}_i}, \text{Client}_{\text{request}_i})\) includes a response and a request and denotes a normal packet pair.

ii. The incomplete packet pair \((\text{Server}_{\text{responses}_i}, \phi)\) includes a response and no request; if the steps of the packet pair \((\phi, \phi)\) are not performed in the transport protocol, it will be denoted as an abnormal packet pair.

3.2 | Detection algorithm with hierarchical policy

We use self-information [22] to calculate the self-information associated with the normal and abnormal packet pairs in a time domain. First, we calculate the probabilities of the normal and abnormal packet pairs and then calculate their self-information in a time domain. Next, we subtract the self-information of the abnormal packet pair from that of the normal packet pair to demonstrate whether the server system is safe, normal, abnormal, or overloaded. Algorithm 2 shows how to calculate the normal and abnormal packets during this period by sampling. Furthermore, the current resource usage of the system and the output of the current state of the system is considered. Some notations used in our approach algorithm are given in Table 2.

The multi-inputs are used to create an algorithm with a hierarchical policy. Simultaneously, we define two time domains, \(T_d\) and \(T_r\), to give our algorithm a hierarchical structure.

The detection algorithm uses two time domains to monitor the system; it detects the time domain \(T_d\) and releases the memory resource time domain \(T_r\). \(T_d\) and \(T_r\) can be modified by the administrator using different requirements of the server system. When the server starts working, \(T\) starts to accumulate. If \(T\) reaches \(T_r\) or \(T_d\), it triggers the corresponding work of our algorithm. Once \(T\) becomes multiple of \(T_d\), the system executes detection and calculates the self-information of the normal and abnormal packet pairs and outputs a linguistic value to describe the system status. Also, when \(T\) becomes the multiple of \(T_r\), the system calculates the self-information sum in every \(T_d\) and different values and then releases memory resources and outputs a linguistic value. If \(T\) becomes multiple of \(T_d\) and \(T_r\), the system executes \((T \bmod T_d) = 0\) and \((T \bmod T_r) = 0\). When the server is under the SYN flooding attack, the system resource

**Algorithm 2** The detection algorithm with hierarchical policy (DAHIP)

**Input:** \(S_{(P_r,n)}\) and \(S_{(P_t,s)}\) are under \(T\), \(T = 1, 2, \ldots, t\);

**Output:** Linguistic values include Safe, Normal, and Abnormal under the time domain \(T_d\) in the time domain \(T_r\), linguistic values include Safe, Normal, Abnormal, and Overload.

\[
x_{P,T} \leftarrow \left( -\sum_{i=1}^{t} \log_2 \frac{S_{(P_r,n)}}{S_{P_t}}, T = 1, 2, \ldots, t; \right.
\]

\[
y_{P,T} \leftarrow \left( -\sum_{i=1}^{t} \log_2 \frac{S_{(P_t,s)}}{S_{P_t}}, T = 1, 2, \ldots, t; \right.
\]

while \((T \bmod T_d) = 0\)

if \((x_{P,T} - y_{P,T}) > R_n\) then return Safe
else if \((x_{P,T} - y_{P,T}) \leq R_n\), or \((x_{P,T} - y_{P,T}) > R_n\)
then return Normal
else \((x_{P,T} - y_{P,T}) \leq R_n\) then return Abnormal

end

while \((T \bmod T_r) = 0\)

if \(\frac{\sum_{i=1}^{T_{d}} x_{P,T}}{\sum_{i=1}^{T_{d}} y_{P,T}} > R_n\) then return Safe
else if \(\frac{\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T}}{\sum_{i=1}^{T_{d}} y_{P,T}} \leq R_n\) AND \(\frac{\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T}}{\sum_{i=1}^{T_{d}} y_{P,T}} > R_n\) then return Normal
else \(\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T} \leq R_n\) AND \(\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T} > R_n\) then return Abnormal
else \(\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T} \leq R_n\) AND \(\sum_{i=1}^{T_{d}} x_{P,T} - \sum_{i=1}^{T_{d}} y_{P,T} > R_n\) then return Overload;

end

while \((T \bmod T_r) = 0\) & \((T \bmod T_d) = 0\)
execute \((T \bmod T_d) = 0\), then execute
\((T \bmod T_r) = 0\)

end
will be exhausted quickly. Therefore, the server system is quickly detected in real-time under the shorter time \( T_d \), and the server system status is described in the release resource time \( T_r \) to achieve the comprehensive monitoring system status and detect SYN flooding attacks quickly. Furthermore, \( R_s \), \( R_n \), and \( R_a \) are set by system administrators according to the system loading. The genuine arrival rate can exhaust the resource, but it still belongs to the normal TCP connection. Although the arrival packets can also exhaust the resource, it brings several page views to this server at the same time. Therefore, it is good for the server, and defending the normal connection is not necessary.

We detect the server system frequently in real-time under the shorter time \( T_d \), and describe the server system status in the release resource time \( T_r \) to achieve the comprehensive monitoring system status and detect the SYN flooding attack quickly.

### 4 ANALYSIS AND COMPARISON

This section experiments and calculates the detection results of Chen and others [19] and our proposed approach and shows figures to describe the experiment differences between Chen and others [19] and our approach. Furthermore, we compare the misjudgment rate, detection method, usage range, half-open connections detecting, and time complexity between Chen and others [19] and our approach.

#### 4.1 Experiment analysis

We analyze the correctness of our proposed method, that is, we will check the calculation process to confirm that the results produced using our method are correct. Then, we use the same analytical method to check the correctness of the results produced using the entropy method in Chen and others [19]. Once the calculation process can be misjudged, it affects other analyses, such as precision, recall, and accuracy, and the correctness of the results. Our analysis considers 1 to 100 parameters in our calculation process, and we draw the analysis chart, as shown in Figures 1 and 2. Note that the range of parameters 1 to 100 does not affect the analysis of the graph. Also, we analyze the correctness of our method and the method using entropy [19] in terms of checking the calculation. Only when the verification process is correct, other analysis methods will have meaning.

Suppose there are 100 data packet pairs in a time domain. The normal and abnormal data packet pairs both sum up to 100 data packet pairs. In Figure 1, we calculate the entropy of the abnormal and normal packet pairs. Then, we subtract the value of the abnormal packet pair from that of the normal packet pair, and the calculation result is shown in Figure 2. When the result is positive, the server system is normal; otherwise, it means that it is close to abnormal.

In Figure 1, the entropy values of the normal and abnormal packet pairs are calculated. The numbers of a normal packet pair range from 0–\( n \), and the number of an abnormal packet pair is \((n + a) - n\). Accordingly, the total packet pair equals the sum of the normal and abnormal packet pairs. In this example, the total packet pair equals 100. The recent system status is the entropy value from the normal packet pair minus the abnormal packet pair. If the normal packet pairs are more than abnormal packet pairs, negative values become outputs; otherwise, it will output positive values. The (abnormal-normal) is shown in Figure 2. From Figures 1 and 2, we discover that entropy values correspond to two packet pairs,
The entropy of packet pair value

In Figure 3, we subtract the values of normal packet pairs from abnormal packet pairs after calculating the self-information. When the self-information value is positive, it means a server system is safer; otherwise, it means a server system is abnormal and even dangerous. When the system has no normal packet pairs, it returns to the initial state. Therefore, as shown in Figure 3, the initial and final values are null for the self-information detection method. It does not mean a self-information value corresponds to two packet pair values. In Figure 4, a self-information value can be calculated from a normal packet pair value minus an abnormal packet pair value. In our proposed detection method, since every self-information value only corresponds to a system status, no system misjudgment occurs. Furthermore, when \((T \mod T_d) = 0\), we define three ranges of system status: safe, normal, and abnormal, by calculating \((x_{P,T} - y_{P,T})\). Similarly, when \((T \mod T_r) = 0\), we define four ranges of system status: safe, normal, abnormal, and overload, by calculating \((\sum x_{P,T} - \sum y_{P,T})\). The ranges of these status cannot overlap each other, and a self-information value cannot repeatedly appear in other status ranges when it has existed in different status range. Therefore, our proposed algorithm prevents system misjudgments [19].

### 4.2 Comparison

In comparison, the misjudgment rate, detection method, wide-range applicability, half-open connections detecting, and time complexity were compared between our approach and that in Chen and others [19]. From the experimental analysis, the misjudgment rate of our approach is lower than that of Chen and others [19]. Our proposed approach uses hierarchical detection, whereas that in Chen and others [19] is linear detection. In the detection method, our approach is multi-input single-output (MISO), using two time values, \(T_d\) and \(T_r\), as the inputs and one system state as the output; the approach of Chen and others [19] is single-input single-output (SISO), using one time value as the input and one system state as the output. In wide-range applicability, since our approach is based on TCP three-way handshake, it can be applied to every transport protocol of TCP. The using range of Chen and others [19] can be used in an E-Mail Server protocol. Both our approach and of Chen and others [19] were used to detect the abnormal traffic flow of half-open connections because both have a definition of packet pairs. From both algorithms and formula, we find the time complexity; our approach is \(O(\log n)\)

\[
\text{Entropy of packet pair value}
\]

\[
\text{Entropy of (abnormal-normal)}
\]

respectively. Therefore, in the entropy detection method [19,22], every entropy value corresponds to two input values. The two input values representing the system status can be either safe or dangerous, causing a system misjudgment. Furthermore, in the algorithm of Chen et al. [19], the system status \((y_{K,T_r} - x_{K,T_r}) \gg 0\) was defined as critical high. However, the most dangerous of the system status is \((y_{K,T_u} - x_{K,T_u}) \equiv 0\) with several abnormal packets instead of the \((y_{K,T_r} - x_{K,T_r}) \gg 0\). Also, the status \((y_{K,T_u} - x_{K,T_u}) \equiv 0\) occurs when abnormal packet pairs are more than normal packet pairs. This status also happens when the abnormal packet pairs are equal to the normal packet pairs, thereby causing a misjudgment. Moreover, \((y_{K,T_u} - x_{K,T_u}) > 0\) and \([y_{K,T_r} - (x_{K,T_r-1} + x_{K,T_r-2} + x_{K,T_r-3})]/3 > 0\) were defined as very high; however, the condition is satisfied when the abnormal packet pairs are more than the normal packet pairs, causing an increase in the very high range. Similarly, when \((y_{K,T_u} - x_{K,T_u}) \ll 0\), \((y_{K,T_u} - x_{K,T_u}) < 0\), and \([x_{K,T_r} - (x_{K,T_r-1} + x_{K,T_r-2} + x_{K,T_r-3})]/3 > 0\), the above misjudgments will happen.
because the values of $x_{P,T}$ and $y_{P,T}$ are calculated from 
\[ \log_2 \left( \frac{S_{(P,T)}}{S_{(P,T)}} \right) \] and \[ \log_2 \left( \frac{S_{(P,T)}}{S_{(P,T)}} \right) \], respectively, and the highest term is $\log n$. Additionally, the time complexity of [19] is under $O(n \log n)$. The comparison table is shown in Table 3. When the time complexity is $(\log n)$, the data are increased $n$ times, and the time is increased $(\log n)$ times. That is, when the data are increased by 1024 times, the time consumption will increase by 10 times $(1024 \times 10 = 10240$ times); when the time complexity is $(\log n)$, the data are increased by $n$ times, and the time consumption is increased by $(n \log n)$ times.

5 | DISCUSSION AND CONCLUSION

This study proposed a detection approach using self-information to detect the SYN flooding attack and provided a detection algorithm with a hierarchical policy. Then, we compared and analyzed the detection using our proposed approach and the entropy method. Results show that our method is more accurate and faster in detecting SYN flooding attacks than the entropy detection method proposed by Chen and others [19] and confirmed that different server systems need to execute detection using a hierarchical policy algorithm. From the experimental results of analysis and comparison, our approach detects the SYN flooding attack efficiently and provides a lower misjudgment rate, hierarchical algorithm policy, wide-range applicability, detecting half-open connections, and lower time complexity. Moreover, the output linguistic value can make the system administrator know the system status.

Our approach uses self-information to detect the SYN flooding attack, and the self-information is used to define the TCP packet pair. Therefore, our approach can be used to monitor memory use. Additionally, our approach can be used in a system that requires a CPU utilization monitor approach.

There are different users and bandwidths among server systems. The suitable defense and prevention policy for each server vary. Hence, this study is focused on detecting the SYN flooding attack not providing defense and prevention policies. Furthermore, the SYN

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Detection method comparison between self-information and entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Detecting TCP flooding attack based on self-information</td>
</tr>
<tr>
<td>Misure rate</td>
<td>Lower</td>
</tr>
<tr>
<td>Algorithm policy</td>
<td>Hierarchical</td>
</tr>
<tr>
<td>Detection method</td>
<td>MISO</td>
</tr>
<tr>
<td>Using range</td>
<td>TCP Server (includes E-Mail Server)</td>
</tr>
<tr>
<td>Half-open connections detecting</td>
<td>Yes</td>
</tr>
<tr>
<td>Time complexity</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>

Abbreviations: MISO, multi-input single-output; SISO, single-input single-output; TCP, transmission control protocol.
flooding attack can be propagated through many ways, for example, DoS, DDoS, and low-rate DDoS/DoS from SYN flooding, and these attacks have corresponding defense and prevention policies. For instance, using SYN cookies to solve SYN floods from the same IP address can be a good policy, but it still needs to detect the attack. The other defense and prevention policies also need to detect first. Therefore, our detection approach is used to detect the occurrence of the SYN flooding attack and then presents the detection results to the system administrator to decide what defense and prevention policies will be used. In other words, we provide a useful and accurate detection approach, so the system administrator only needs to decide what defense and prevention policies are required to prevent the SYN flooding attack on the server system.

ACKNOWLEDGMENTS
The Ministry of Science and Technology partially supported this research, Taiwan (ROC), under contract nos. MOST 109-2221-E-468-011-MY3, MOST 108-2410-H-468-023, and MOST 108-2622-8-468-001-TM1. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

CONFLICTS OF INTEREST
The authors declare that there are no conflicts of interest.

ORCID
Min-Shiang Hwang https://orcid.org/0000-0001-5502-8033

REFERENCES
**AUTHOR BIOGRAPHIES**

**Jia-Rong Sun** received his M.S. and Ph.D. degrees in Computer Science and Information Engineering from Asia University, Taiwan, in 2010 and 2015. He is currently a management consultant of information security and personally identifiable information (PII) in Taiwan. His main research interests include Information security and cybercrime investigation.

**Chin-Tser Huang** received his B.S. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1993, and his M.S. and Ph.D. degrees in Computer Sciences from The University of Texas at Austin in 1998 and 2003, respectively. He is a professor in the Department of Computer Science and Engineering at the University of South Carolina at Columbia. His research interests include network security, network protocol design and verification, and distributed systems. He is the director of the Secure Protocol Implementation and Development (SPID) Laboratory at the University of South Carolina. He is the author (along with Mohamed Gouda) of the book “Hop Integrity in the Internet,” published by Springer in 2005. His research has been funded by DARPA, AFOSR, AFRL, NSF, NEH, and USDOT. He is an NRC Research Associate in 2020 and a recipient of the USAF Summer Faculty Fellowship Award and of the AFRL Visiting Faculty Research Program Award in 2008–2021. He is a senior member of IEEE and ACM.

**Min-Shiang Hwang** received his M.S. in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988 and his Ph.D. degree in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He was a distinguished professor and chairman of the Department of Management Information Systems, NCHU, during 2003–2011. He obtained 1997, 1998, 1999, 2000, and 2001 Excellent Research Awards from the National Science Council (Taiwan). Dr. Hwang was a dean of the College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. His current research interests include information security, electronic commerce, database, data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.