行政院國家科學委員會補助專題研究計畫 ■ **成果報告** □**期中進度報告**

開放原始碼測試工具:

再造真實網流量於內部實驗室環境

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 94-2218-E-009-029
執行期間：2005 年 08 月 01 日至 2006 年 07 月 31 日

計畫主持人：林盈達 教授
共同主持人：
計畫參與人員：陳一瑋、羅棨鐘、王阜毓、陳李睿、彭偉豪

成果報告類型(依經費核定清單規定繳交)：■精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：交通大學資訊工程學系

中　華　民　國　95　年　10　月　13　日

一、中文摘要

網安產品的複雜度愈來愈高，內部測試結果經常無法符合使用者端的實際狀況，在使用者端可能會發生一些如: 誤擋、當機、效能低落等沒有在內部測試時發現的問題，歸究其原因，便是測試流量不夠真實所造成。解決這樣的問題，可利用流量錄製及播放的技術，到使用者端錄製該點的流量，將流量帶回測試部門內部播放出來對產品進行測試。我們在此計劃中加強了播放技術: 可以支援 NAT 模式下的待測物、搭配效能測試、以及粹取攻擊流量進行播放這個萃取攻擊流量的系統主要有三個重點，第一，本系統利用播放錄製的流量到入侵探測和防護系統來取得警示紀錄。第二，根據警示紀錄從真實流量中找出令入侵探測和防護系統發出警示的最重要封包，藉由前兩個重點，有相同網路特徵值的封包集合則稱為一個網路攻擊連線。然而，一個網路攻擊可能會有多個來源，或者一個來源卻有多條連線，因此，本研究經過分析觀察後設計了第三個重點。第三個重點是藉由內容相似度比對來找出多個來源的攻擊。透過萃取攻擊流量系統所取得的 83%攻擊是不容易受外在影響而變化的，在低變化量攻擊中有 71%的攻擊是可被驗証為完整且無雜質的。

關鍵詞: 網路安全、弱點偵測、網路攻擊、流量萃取、內容相似度比對

二、英文摘要

There are more and more conflicts happened as the in-lab testing results can not reflect what end-users feel in their daily life. They may suffer from the device crash, false-positive or false-negative situation, and bad performance. These problems come from the testing traffic is not “real” enough in the lab. The best choice to this case is “capture and replay” real traffic. That means bring a machine to the real sites to record their daily traffic then bring it back and resend the captured traffic onto the network. We enhance traffic replay tool with: support for NAT mode DUT, Performance testing, and attack traffic replay. There are three main points in the attack traffic replay system. The first thing to confront is attacks identification. This system is replaying the traffic to IDP products to identify attacks because the IDP products can support the rich attack’s identifications. Second, find out the critical packet and others. Before extract attacks, find out the critical packet is necessary because we must know what kinds of attack should be extracted. Although those two steps can find out the packets that have the same network characteristic and merge to a set as a connection of attacks by leverage the logs of IDP products, attacks of multi attackers or multi connections might not extractable because the network characteristics are not the same. Therefore, this system calculates the similarity of packets to find out the attacks because some attacks, i.e. DDoS attacks, have the same or similar packet payload. This work also uses the variation of the extracted attacks to find out the attacks of the multi connections because the variation of attacks should be smoothly. In our experiments, we can extract all attacks that can be alarm by IDP. The 97% of the extracted attacks have low variation. The 99% of the low variation attacks can be verified as completeness and purity.

Keywords: Network Security, Vulnerability Assessment, Network Attacks, Session Extraction, Payload Similarity

三、報告內容

前言

Most NCSec (Network and Content Seucity) Products now are supporting multiple operational modes, ex. transparent, router, and NAT. If the DUT is transparent mode, nothing needs to change within the packets while replaying. If the DUT is router mode, we need to modify MAC address and IP addresses of captured traffic to fit the network environment. However, there is no mechanism to deal with DUT running NAT mode. We need to maintain a corresponding address translation table in the replaying system and make proper translation when the relay process goes on. Besides, we keep track of the number and size of packets been transmitted and received, record the duration of a packet's lifetime to measure the capability of processing speed of the DUT.

Extracting a complete episode of attacks from an overwhelmingly large amount of recorded traffic is non-trivial. For this goal, by our designing, the real traffic is recorded and then replayed to the intrusion detection and prevention (IDP) products to extract the complete episode of attacks. Such an approach to record traffic and send it to IDP products has been used for evaluation of the performance of the IDP [1], [2]. The IDP products indicate the signs of detected attacks on its logging system, but do not record the attack traffic. This work designs a method to extract attacks according to the logs of IDP products. This method records real traffic and then extracts attacks by associating packets via logs with connections and then with sessions. This session extraction system therefore can extract the desired complete episode of attacks.

研究目的

To provide a more realistic testing environment, we use the mechanism of traffic recording and relaying to do a health examination on the DUT. By recording real world traffic and replaying it in the Lab, DUT will face the same challenge as「BetaSite Testing」. Dangerous situations could be avoided from customers' side like: crash, reboot, slow down. Current Open-Source relaying tool – tcpreplay is somehow not powerful enough to meet all testing requirements for products in NCSec area. We think three features are wroth being developed: NAT mode DUT, Performance testing, and attack traffic replaying. For the replay of attack traffic, we have to 「extract」 the attack sessions from the captured traffic. There several benefits when supporting ASE. First, the extracted attack traffic can be used to do the test of「Attack Recognition」in IPS. Second, we can replay the useful traffic only to save a lot of time. Third, provide a variety of background traffic to do the performance testing in a realer way. Finally, we can because the partial attacks emulated by Nessus can only indicate possible security breaches but does not know whether a real attack will harm or not, this work proposes to extract the complete episode of attacks to make sure the system vulnerability. For this goal, the real traffic is recorded and then replayed to the IDP products to extract the complete episode of attacks. Trivially, the logs of the IDP products can help this work to find out the connection of the detected attacks. However, the attack may have the multiple connections. All related connections of the attack can not be all extracted by the logs of the IDP products because the IDP products only alarm and log the most important connection. Therefore, this work proposes an algorithm to extract multi connections from the attack traffic. We named the multi connections extraction algorithm as the session extraction algorithm.

文獻探討

I. Attack types

This work collects 83 attacks as the samples for the extraction system. These attacks can be divided into three types according to the number of attackers and the number of connections per attack, as presented in Table 1. We assume only one target is in each attack. An attack of the first type involves one attacker and a single connection. An example is the MySQL Authentication Bypass Exploit. This attack can login in a MySQL database without the password. An attack of the second type involves one attacker and more than one connection. An example is the Blaster worm, which establishes three connections when it tries to attack a target. An attack of the third type involves multiple attackers and a single connection from each attacker. A DDoS attack belongs to this type. This observation is helpful to build an extraction system [3], [4].

Number of attackers	Number of connections per attack	Example
1	1	MySQL Authentication Bypass Exploit
1	N	Blaster worm
N	1	DDoS

Table 1: Three types of attack definitions

研究方法

Support DUT runnig NAT mode and performance testing

The basic idea of supporting NAT mode replay is to maintain an IP address mapping table. This table is used to keep track of the variation of source IP address when packets pass through DUT from inside to outside. At this direction, we 「update」 the mapping table according the change of source IP address. On the contrary, we query this mapping table to modify destination IP address. Figure 1 explains this process.

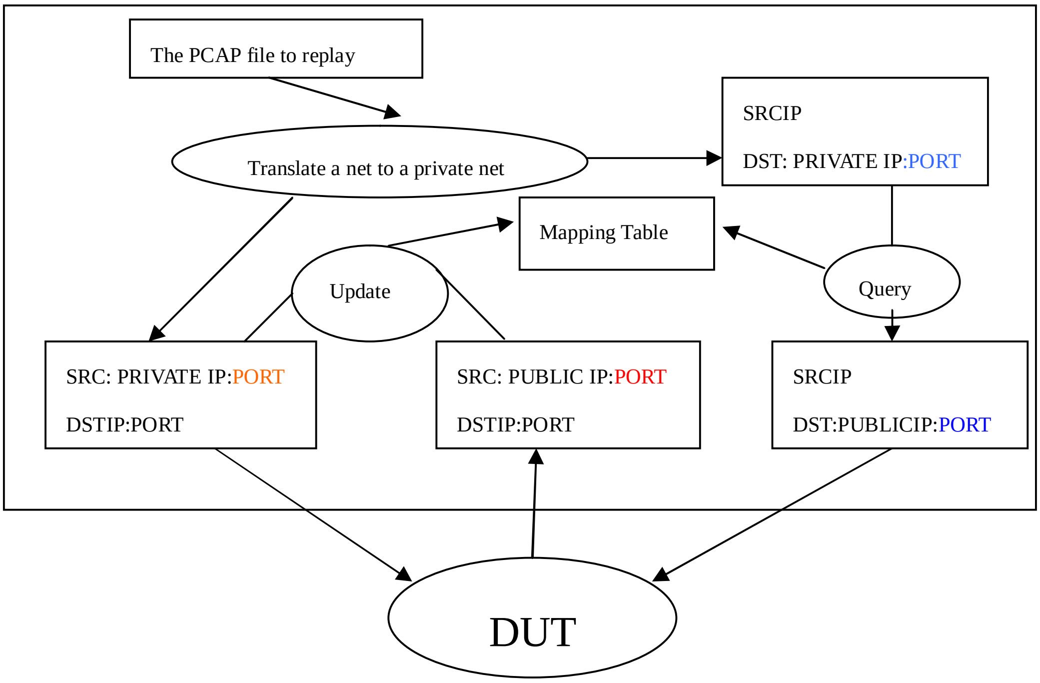

Figure 1: Internal process to support DUT running NAT mode

For the performance testing, we maintain several counters to record the number of packets and the size of data been transmitted and received. Table 2 shows this mechanism:

Counter / Action	Total Packet Send	Total bytes send	Begin Sent Time	Finish Send Time
TCPReplay start			Record	
Send packet	+1	+Send packet length		
TCPReplay finish				Record

Table 2: mechanism of performance testing

Attack Session Extraction

I. Extract attack sessions from recorded traffic

One goal of this work is to extract a complete episode of attacks from a large amount of traffic. The session extraction algorithm is a three-pass algorithm designed for this goal by associating packets, connections and sessions to extract attack sessions. Before the description of the session extraction algorithm, Table 3 shows the definition of the components in session extraction algorithm. The algorithm consists of five steps as follows. Step (i), (ii), (iii) and (v) are trivial works while the step (iv) is the essence of this work.

Names	Descriptions
S_{ip}	Source IP address
S_{port}	Source Port number
D_{ip}	Distance IP address
D_{port}	Distance Port number
Tcp/Udp	The TCP packet or UDP flag
Payload	The content of the packet
P	A TCP or UDP packet in the IP network.
$Tuple(P_i)$	The five-tuple of a packet
A	The anchor packet of the attack
PDA(Possible DoS Attacks)	The data structure that store the packets could be the DoS attacks
PNDA(Possible Not DoS attacks)	The data structure that store the packets could be not the DoS attacks

Table 3: The definition of the components in session extraction algorithm

(i) *Replay real traffic to IDP products by Tcpreplay*.

This algorithm uses the domain knowledge of IDP products, including the well-known Open Source tool, Snort [5]. A IDP product illustrate what attacks have happened with its logs.

(ii) *Find out anchor packets by the first-pass scan*.

This step finds out anchor packets, the critical packets that IDP products alarm when receiving them. There are two tables used herein. One is the alarm log table , which records the alarms of attacks from the replay of attack traffic. The other is the replay log table, which records the time when Tcpreplay replays each packet.(The timestamps from the replay log table are used to mark the attack types by looking for the relation from the alarm log table. The replay log table is then compared with the alarm log table to identify the attack packets.) Time synchronization could be a problem between the replay system and the IDP products. Even if the time has been synchronized, IDP products may not log the times accurately. Therefore, the five-tuple information is used herein. Many IDP products also log the five-tuple information of an attack (some may record fewer than five tuples). The five-tuple information and the timestamp from the alarm log table and the replay log table can locate the anchor packets in the real traffic.

(iii) *Find out the association among attack packets within the same connection by the second-pass scan*.

This step discovers the anchor connection by looking for the relation of the recorded packets with the anchor packets. If the packets have common five tuples with the anchor packet, they belong to the same connection.

(iv) *Find out the association among attack connections within the same session by the third-pass scan*.

The attack connections can be associated with their session. The association may be difficult since the relation among the connections is obscure. Because the attacks have more than one connection, only five tuples and timestamp are insufficient to find out the other connections. The obscurest relation among the connections is the attack of multiple attackers and a single connection from each attacker because the five tuples of the packets from these attackers are different. A common attack of this type is the DDoS or DoS attack. These two types of attacks overwhelm a server to deny its capability of providing services. From our observation, such an attack often has only the TCP ACK or SYN message, as well as a number of packets with the same data payload. The session extraction algorithm is designed based on the above observation. The algorithm parses the recorded traffic packet by packet and extracts an attack session by analyzing the attack types.

After anchor packets of an attack have been found, the algorithm checks each following packet to see if its source IP address or destination IP address is identical to the target IP address of the anchor packet. If not, the packet will be classified to other type of attacks. If the packet belongs to this attack, the algorithm will compare each packet's payload for similarity. The algorithm duplicates a copy in the possible DDoS attack buffer and increases the packet count by one if the similarity is high. The similarity is defined according to the *longest common subsequence* (LCS) of two packet payloads [6]. Formally, given a sequence $X = (x_1, x_2, ..., x_m)$, another sequence $Z = (i_1, i_2, ..., i_k)$ is a *subsequence* of *X* if there exists a strictly increasing sequence $(i_1, i_2, ..., i_k)$ of indices of *X*. given two sequences *X* and *Y*, we say that a sequence *Z* is a *common subsequence* of *X* and *Y* if *Z* is a *subsequence* of both *X* and *Y*. The *longest common subsequence* is the longest subsequence of the all *common subsequence.* Consider the payloads of two packets as two sequences of bytes, S_1 and S_2. The LCS of S_1 and S_2, LCS (S_1, S_2), is the longest sequence of bytes that are subsequences of S_1 and S_2. The similarity is defined by the equation

$$\text{Similarity}(S_1, S_2) = \frac{2 \times |\text{LCS}(S_1, S_2)|}{|S_1| + |S_2|} * 100\% \cdot \quad (1)$$

The similarity threshold is 80% in the proposed algorithm because the packets we collected in the DDoS or DoS attacks are often the minimum Ethernet packets of 64 bytes. Excluding 14-byte MAC header, 20-bytes IP header, 20-bytes TCP header and 4-byte checksum, the payload is only 6 bytes long. From our observation, the packet payloads of the DDoS or DoS attacks we collected are often the same, and the difference is only one byte if the payloads are different. The similarity in this case is 83.33%, so the similarity threshold is set to 80%.

After identifying similar packets, the session extraction algorithm watches the source IP address and the destination IP address at the same time. The step keeps only the packets that come from the attacker and go to the target and those in the opposite direction. The others are simply dropped. This step intends to distinguish the attacks that possibly have one attacker from those that are possibly DDoS attacks.

The algorithm continues to watch the next packet until the end. The algorithm returns the packet count in the possible DDoS attack buffer. The attack might be a DDoS attack if the count is larger than 200, and might be a 1-1 attack otherwise. Figure 2 shows the flowchart of the algorithm.The algorithm can be written as some formulas and pseudo code as follows. We defined the packet P is the set of five-tuple and payload. The $Tuple(P_i)$ is the five-tuple of the packet *i*, $i \geq 1$. The anchor packet A is the set of the five-tuple and payload that the IDP products make alarm when they receive it.

$$P = \{S_{ip}, S_{port}, D_{ip}, D_{port}, Tcp/Udp, Payload\}, \quad (2)$$

$$Tuple(P_i) = (S_{ip}(P_i), S_{port}(P_i), D_{ip}(P_i), D_{port}(P_i), Tcp/Udp(P_i)), \quad (3)$$

Therefore, the session extraction problem turns into a problem to find out the set of packets that have the high similarity of payload with anchor packet A or the same source IP address and distance IP address with anchor packet A. Assume the x is the sequence number of anchor packet in the all packets. The session extraction algorithm can be described as follow.

The pseudo code of the session extraction algorithm

```
PDA = f; // a set of packets, possible the DoS attack
PNDA = f; //a set of packets, possible  not the DoS attack
DDos.packet_number = 0;
Given x,
A = Px;
For all i{
    if (Tuple(Pi).Sip = Tuple(Px).Dip || Tuple(Pi).Dip = Tuple(Px).Dip){
        if (Similarity(Pi.Payload, Px.Payload) ≥ 80%){
        PDA = PDA U Pi;
        DDos.packet_number ++;
        }// End of if
        if ((Tuple(Pi).Sip = Tuple(Px).Sip && Tuple(Pi).Dip ≠ Tuple(Px).Dip)) ||
            Tuple(Pi).Sip = Tuple(Px).Dip && Tuple(Pi).Dip ≠ Tuple(Px).Sip)){
        PNDA = PNDA U Pi;
        }// End of if
    } // End of if
}//end of for
if ( DDos.packet_number ≥ 200){
    return PDA;
}else{
    return PDNA;
}//end of if
```

(v) *Replay the extracted attack session to IDP products to verify whether the same logs are generated. If it is true, the extraction is valid.*

Finally, we replay the extracted attack sessions to IDP products to verify the correctness of the extraction. The extracted session must cause the same alarms as the whole traffic was replayed to the same IDP product. If an IDP product cannot find the attack, the extraction is invalid.

結果與討論（含結論與建議）

Support DUT running NAT mode and performance testing

We use a NAT device and an IDP product to evaluate there two features. If the NAT replay works well, we can see “active” connections displayed on the DUT. After the replaying process completed, the console of In-Lab Live Testing System will show the results of the performance measurement. Also, we use “external” program to watch the effects to system performance when supporting these two features. It seems that there is only a little drop (less than 0.1%) on throughput evaluation.

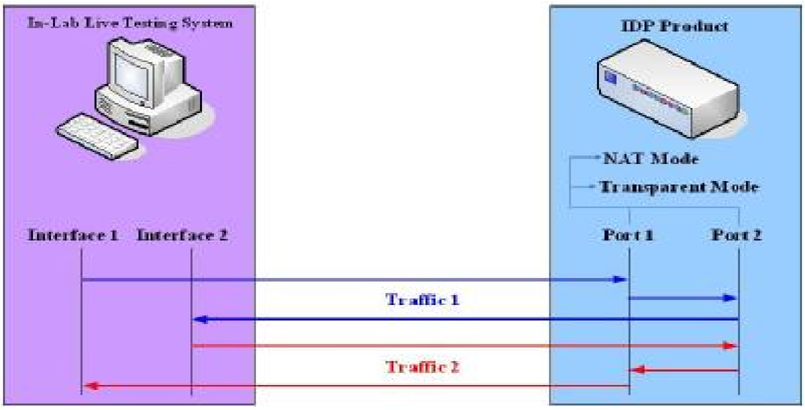

Figure 2: Environment to evaluate two features: NAT replay and Performance measurement

Attack Session Extraction

I. The variation of the session extraction system

The definition of variation in this work is the complement of the probability of the extracted attack’s mode value. The 「mode」 value is the most frequent value.

$$\text{Variation}(\text{Attack}_i) = (1 - P(\text{mode}(\text{Attack}_i))) * 100\% \ . \qquad (4)$$

In our experiment, the different extracted attack sizes for each attack when they could classify as the same attacks come from the result of the comparing the attack size with the size that the most frequent size. The low variation of the session extraction system must be proved if we want to use the results of the session extraction system. In this experiment, we replay 100 attacks and the common real traffic at the same time. We mixed the 100 attacks with 10 different real traffics to observe the variation. Therefore, there are total 10 results (the extracted attacks) of the each attack and the total 1000 results by session extraction system.

Figure 4, 5, and 6 show three cases of the results that we extracted the attacks from the real traffic. The x-axis is 10 extracted attacks of each attack. Figure 4 shows the case one that is the different sizes of attacks less than 3. In this experiment, the 97% of the 100 attacks were in case one. Figure 5 shows the case two that is different sizes of attacks equal to 0. In this experiment, the 52% of the 100 attacks were in case two. Figure 6 shows the case three that is the different sizes of attacks more than 3. In this experiment, the 3% of the 100 attacks were in case three. Figure 3 shows the accumulated number of the attacks of each variation by increasing. The 97% of the extracted attacks is less than 30% variation. The 30% variation could be easy to choose the attack size equal to the size that the most times in our experiment. But, there are also 3% of the extracted attacks could be hard to choose the result of the experiment because they had high variation.

II. The completeness and purity of the session extraction system

The definition of completeness and purity are as the same with the definition of similarity we used before. If the similarity of the extracted attacks and original attack equal to the 100%, we will say the extracted attack is completeness & purity. Otherwise, we will say the extracted attack is not completeness and purity. If the extracted attacks are different with the original attack, we will evaluate the completeness and purity rate (the evaluation of similarity). In our experiment, the extracted attacks of 0% variation are all completeness and purity. The max complete and purity rate of extracted attacks that less than 30% variation is 97.83%, the min rate is 74.77%, and the average rate is 89.58%. The reason of the extracted attacks that are not complete and pure is other connections between attacker and target, i.e. the attacker might also have normal http connections with target. Those normal connections between attacker and target have no association with attacks. If we transfer the IP of the original attacks in our experiment to another IP domain that are different with the mix traffic, the variation will become 0% and completeness and purity will equal 100%. Therefore, we know the reason of the extracted attacks that are not complete and pure is other connections between attacker and target.

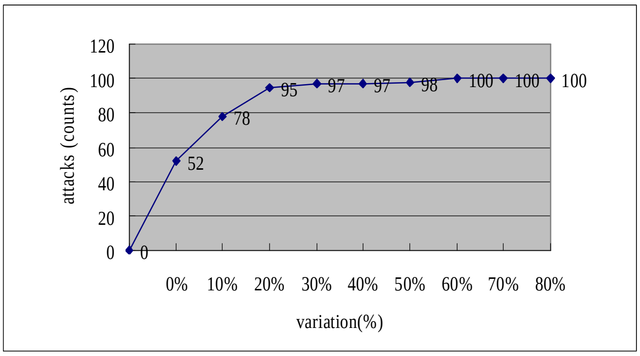

Figure 3: The variation of extracted attacks

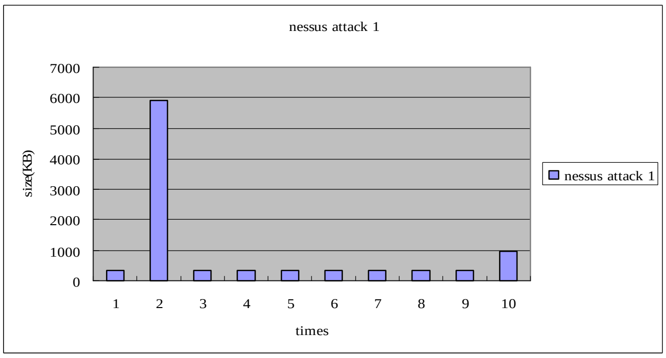

Figure 4: 0% < Variation < 30%

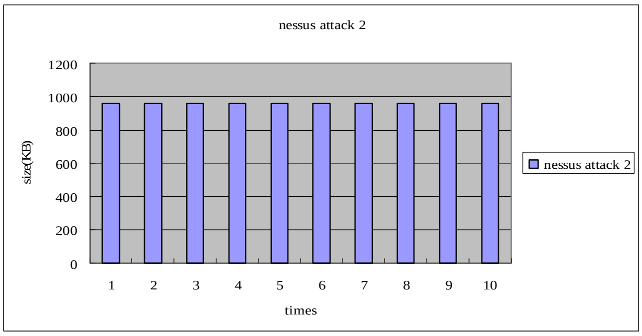

Figure 5: The Variation = 0%

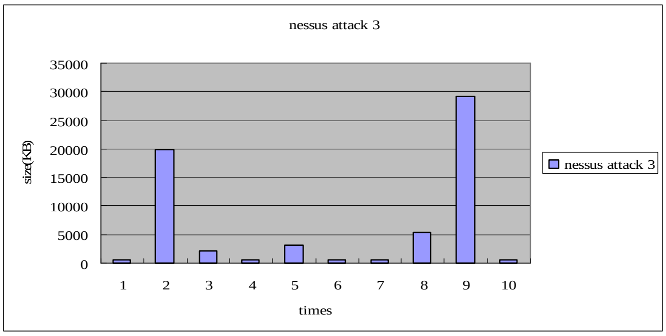

Figure 6: Variation > 30%

參考文獻

[1] H. G. Kayacık and A. N. Zincir-Heywood, “Using Intrusion Detection Systems with a Firewall: Evaluation on DARPA 99 Dataset”, Project in Dalhousie University, [Online]. Available:http://projects.cs.dal.ca/projectx/files/NIMS06-2003.pdf.

[2] DARPA 99 Intrusion Detection Data Set Attack Documentation. [Online]. Available: http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html.

[3] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. spafford, Aurobindo Sundaram, Diego Zamboni, "Analysis of a Denial of Service Attack on TCP," sp, p. 0208, 1997 IEEE Symposium on Security and Privacy, 1997.

[4] Vern Paxson, “An analysis of using reflectors for distributed denial-of-service attacks” ACM SIGCOMM Computer Communication Review, 2001

[5] Martin Roesch, “Network Security: Snort - Lightweight Intrusion Detection for Networks”, Proceedings of the 13th USENIX conference on System administration, November. 1999.

[6] T. H. Coemen, C. E. Leiserson, R. L. Rivest, ”Introduction to Algorithms”, pages 314-320, 1990.

[7] T. Ye, D. Veitch, G. Iannaccone and S. Bhattacharyya, “Divide and Conquer: PC-Based Packet Trace Replay at OC-48 Speeds”, IEEE TRIDENTCOM, 2005.

[8] W. C. Feng, A. Goel, A. Bezzaz, W. C. Feng, and J. Walpole. “TCPivo: A high-performance packet replay engine”. ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools), Aug. 2003.

四、計畫成果自評

項目	自評	詳細說明
研究內容與原計畫相符程度	相符	支援 replay 三項功能: (1) NAT mode DUT (2) Performance Testing (3) Attack Traffic Replay
達成預期目標情況	皆有達成	同上
研究成果之學術或應用價值	相當有應用價值	目前與工研院交大網路測試中心(NBL)合作，已有五家以上的廠商使用此工具
是否適合在學術期刊發表或申請專利	適合	已有學生在進行相關技術之論文研究，同時即將把此相關技術技轉給業界的網通代表廠商 CISCO

主要發現或其他有關價值	廠商有更多的需求	需要再支援新的功能,包括 P2P 流量粹取、隱私性保護，已提出第二代計劃，目前也審核通過。
綜合評估	所需之功能有如進度開發出來且實用價值高，但廠商需要更多的功能	同上，已申請計劃，再加以開發、補強。

五、可供推廣之研發成果資料表

□ 可申請專利　■ 可技術移轉　日期：95 年 10 月 13 日

國科會補助計畫	計畫名稱：開放源碼測試工具: 再造真實網流量於內部實驗室環境 計畫主持人：林盈達 教授 計畫編號：94-2218-E-009-029　學門領域：資訊學門一
技術/創作名稱	In-Lab Live Testing System - Attack Session Extraction
發明人/創作人	林盈達、羅榮鐘、陳一瑋
技術說明	中文： 這個萃取攻擊流量的系統主要有三個重點，第一，本系統利用播放錄製的流量到入侵探測和防護系統來取得警示紀錄。第二，根據警示紀錄從真實流量中找出令入侵探測和防護系統發出警示的最重要封包，藉由前兩個重點，有相同網路特徵值的封包集合則稱為一個網路攻擊連線。然而，一個網路攻擊可能會有多個來源，或者一個來源卻有多條連線，因此，本研究經過分析觀察後設計了第三個重點。第三個重點是藉由內容相似度比對來找出多個來源的攻擊。
	英文： The attack session extraction system has the three key points. First, the attack session extraction system is replaying the recorded traffic to IDP products to get alarm logs. Second, the attack session extraction system found out the critical packet that the IDP products make alarm by the alarm logs. The first and second key points of the attack session extraction system can find out the packets that have the same network characteristic and merge to a set as a connection of network attacks. However, a network attack maybe have many attackers or single attacker but multi connections. Therefore, this work analyzed the attacks and designed the third key point. The third key point is using the packet payload similarity to find out the attacks that have the multi attackers.

可利用之產業 及 可開發之產品	可利用之產業: 網路安全產品測試。 可測試的產品: Intrusion Prevention System，無論是 Transparent、Router 或 NAT 模式。 可開發之產品: 利用此技術可進一步加強「弱點偵測」產品。
技術特點	Attack Session Extraction (1) Log parser for different log formats (2) Mark table support for retrieving 5-tuple (3) Packets association for the connection (4) Packets association for the session
推廣及運用的價值	1. 攻擊流量可以用來測試 IPS 產品的特徵值資料庫的正確性 2. 節省測試時間 3. 提供測試時更多不同的背景流量 4. 加強 VA (Vulnerability Assessment)工具的真實性