國立交通大學

資訊管理研究所

碩士論文

一個新的以共同刺激機制為基礎之入侵偵測架構

A Novel Intrusion Detection Architecture Based on the Co-Stimulation Mechanism

研究生：鄭立群

指導教授：羅濟群 教授

中華民國 九十四 年 六 月

一個新的以共同刺激機制為基礎之入侵偵測架構

A Novel Intrusion Detection Architecture Based on the Co-Stimulation Mechanism

研 究 生： 鄭立群　　Student: Li-Chyun Cheng

指導教授： 羅濟群　　Advisor: Chi-Chun Lo

A Thesis
Submitted to Institute of Information Management
College of Management
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Business Administration
in
Information Management
June 2005
Hsinchu, Taiwan, the Republic of China

中華民國　九十四 年 六月

一個新的以共同刺激機制為基礎之入侵偵測架構

研究生：鄭立群　　指導教授：羅濟群 老師

交通大學資訊管理研究所

摘要

False Positive 為現今入侵偵測系統(IDS，Intrusion Detection System)的最大問題之一，它的狀況為” 針對符合入侵規則卻沒有真正入侵的行為發出警報”。這情形使得網路管理員需要耗費大量時間來判斷警報的真假，進而造成管理員對於入侵偵測系統所發出的大量警告產生不信任與無力感。

因此本論文為降低 False Positive 的需求，提出一個入侵偵測的架構。本架構以共同刺激機制(Co-stimulation)之二階段確認的偵測方式為基礎，並配合異常封包分類處理的設計，分別處理「網路型」與「主機型」的異常封包，以達到有效過濾在主機上不會產生異常狀況的攻擊封包，另外亦可偵測出針對 False Positive 的攻擊類型，藉此以提升系統偵測入侵行為的效能。

關鍵字：入侵偵測、誤判率、False Positive、共同刺激

A Novel Intrusion Detection Architecture Based on the Co-Stimulation Mechanism

Student: Li-Chyun Cheng Advisor: Chi-Chun Lo

Institute of Information Management

National Chiao-Tung University

Abstract

Now false positive is one of the most important problems for an Intrusion Detection System (IDS). False positive is the wrong alert sent by IDS when the behavior fit in with the signature of intrusion rule but no real intrusion actions. The wrong alerts will take administrators a lot of time to check that the alerts are valid or not. Let us have no much time to handle other jobs. We also have no confidence about these alerts.

Therefore, in our research, we propose an intrusion detection framework based on a co-stimulation mechanism, which triggers the Monitor Agents on the host system to make sure if there are any real intrusion actions. For filtering invalid alerts efficiently, we classify unusual packets in accordance with two types, network-based and host-based type. Our proposed framework can reduce false positive alerts and increase the rate of correct detection.

Keywords: Intrusion detection、False rate、False Positive、Co-stimulation

誌謝

研究所的兩年是我人生中很特別的學習過程，在這段時間內我不僅學習到專業的知識，也體認了做人做事的方法與態度。因為在「交大資管所」裡不乏有優秀的老師、學生與環境，讓我永遠有優秀且值得學習的對像，一直不斷持續成長，重新認識這個世界。

這兩年來，除了最感謝我的母親與指導教授羅濟群老師的栽培外，還要特別感謝：氣質美女-秀文大姐，見解精闢-俊龍學長，超級用功-俊傑學長，程式神人-永龍學長，Unix-like 魔人-明橋學長、多媒體研究小姐-一濤、建治與網路實驗室全部的同伴等人。非常感謝你們在生活中或課業中都能適時給予我支持與鼓勵，陪我一路走過研究所生活的點點滴滴，如果少了你們我的研究生活一定無法如此順利，也無法充滿溫馨的回憶，衷心祝福往後的日子裡你們也能同樣地幸福。

最後還要感謝我的口試委員林熙禎與劉敦仁老師對於我論文的指導，讓我在研究生生活的最後能畫下完美的句點。

2005/6/27

目次

圖目次

表目次

第一章　緒論

1.1 研究背景與動機

由於網路應用的興盛，資訊交流的速度與廣度也急速成長，這背後的意義代表著電腦與電腦的接觸已經更加親密，另一方面也代表著更多網路安全問題將隨之產生。從 CERT 的安全相關統計數據來看即可知，1995 年至 2004 年的電腦系統弱點數統計從 171 件倍增至 3780 件，而安全事件發生的頻率更是驚人，光 2003 一年就發生了 137,529 件。

伴隨著電子商務、網路通訊與網路服務等相關應用之使用率的成長，更加突顯了網路安全的脆弱性與重要性。當我們正享受網際網路的便利時，網路上的許多威脅卻使我們倍感困擾，我們除了希望電腦能在網際網路的空間正常運作外，也特別在意網路服務的品質、電腦內重要資料的儲存與個人隱私保密等相關問題，這不管對個人或對公司都是能否安心享受網路之利的重要前提。而電腦系統上的漏洞、病毒、網路上散播的駭客資訊與工具等相關威脅，卻更加重了網路安全的挑戰。

一般常見的網路安全機制包含了加解密、防火牆、存取控制、系統稽核、弱點分析等，而入侵偵測系統則是為網路安全中重要的工具之一。入侵偵測系統的概念最早是從 Anderson 於 1980 年時在「Computer Security Threat Monitoring and Surveillance」一文中所提出。它的功能在於能偵測出現行環境下是有否遭受攻擊的狀況，例如，阻斷服務攻擊(DoS, Denial of Service Attack)、後門程式、緩衝區溢位（buffer overflow）等攻擊行為。並且會發佈警報，以告知系統管理員應進行處理並會留下事件記錄。

現今入侵偵測系統最大問題之一就是 False Positive，也就是系統會對於符合入侵規則，卻不是真正的入侵行為(含有入侵規則特徵，卻沒有真正的入侵行為)發出誤判的警報。這使得網路管理員耗費大量時間在判斷警報的真假因而無暇處理其他業務，並且對於入侵偵測系統的大量警告產生了不信任。因此本論文基於共同刺激機制(Co-stimulation)[9]提出一個入侵偵測架構，設計重點在於減少 False Positive，以提升其正確判斷入侵行為的效能。

1.2 研究目的

入侵偵測系統的誤判可分 False Positive 與 False Negative 兩種。False Positive 的定義為「實際上沒有入侵行為，但系統卻產生有入侵行為的判斷」之狀況。

誤判率一直是限制入侵偵測系統效能的主因，也是讓設計廠商與使用者感到相當困擾的問題。在本論文中，特別針對入侵偵測系統(IDS， Intrusion Detection system)討論其誤判率形成的原因與所造成的問題，提出一個能降低 False Positive 的入侵偵測系統架構。

本論文針對降低 False Positive 的需求，以基於兩階段確認之共同刺激(Co-Stimulation)機制為基礎，提出一個入侵偵測的系統架構。其簡單概念為，當網路封包通過第一階段網路型入侵偵測系統後，如果被分析判斷為異常封包時，系統會先把異常封包分類成網路型的異常封包或是主機型的異常封包，以進行分類處理。如果為主機型的異常封包時，即就會觸發第二階段監控代理人(Monitor Agents)的共同刺激機制，在封包所通往的主機上持續監控其系統狀況上的變化。而監控的範圍主要包含了完整性(Integrity)、私密性(Confidentiality)與可用性(Availability)三項。只要上述三項其一狀態在被監控的系統上發現異常時，本系統即可確認其為入侵行為。

由於第二階段的 Monitor Agents 所監控的是主機上實際的狀況，因此能更加提升正確判斷入侵行為的準確性。

1.3 章節規劃

本篇論文目的在於設計一個能有效降低 False Positive 的入侵偵測系統架構。第二章將探討入侵偵測系統的分類和特性，與針對誤判率所造成的影響及攻擊；第三章說明本篇論文所提出的偵測方法與系統架構；第四章介紹系統實作的模型、實驗數據與安全性分析、以及效能分析；第五章則對本系統未來可研究的方向加以討論。

第二章　文獻探討

本章將探討入侵偵測系統的分類、特性、架構、偵測技術與現在未來的發展狀況。接著說明False Positive的相關問題，最後介紹目前所提出降低False Positive的方法與架構。

2.1 入侵偵測系統[15]

入侵偵測系統的功能為偵測惡意的入侵攻擊行為，目的是在主機系統遭受嚴重破壞、系統不正常運作、網路服務癱瘓及重要儲存資料的毀損等重大損失發生時，即時發出遭受攻擊的警告，提醒系統管理員應馬上處理或預先防範以盡可能地降低各種損失。

2.1.1 入侵偵測系統的分類[15]

入侵偵測系統根據監控的對像而言可以分為「網路(Network-Based)型」與「主機型(Host-Based)」。以下將各別地介紹其特色、監控對像與相對的優缺點比較，及各別的偵測限制(表2-1)。

- **網路型入侵偵測系統(Network-Based IDS)**

網路型入侵偵測系統最早是在1990的NSM(Network security Monitor)所提出。主要是監控網路中所傳播的封包之內容為主，例如，封包格式是否合乎網路協定的格式、是否帶有攻擊字串的關鍵字等。

通常網路型入侵偵測系統的建置需要一台獨立主機來負責，且此主機所佈署之位置必需是可收到所防護區域內全部封包的節點。但此會造成一個問題：當網路流量過大時，網路型入侵偵測系統會因過載(overloading)而無法正常運作，造成封包遺失，進而影響偵測的準確率。網路型的優點是採用獨立監控主機來進行監控，相對於主機型的好處是不必考慮不同電腦架構及作業系統的相容性問題。

- **主機型入侵偵測系統(Host-Based IDS)**

主機型入侵偵測系統最早是由J.P.Anderson於1980在「Computer Security Threat Monitoring and Surveillance」[3]一文中所提出。主要概念是經由監控系統的運作情況來判別是否有入侵行為發生，例如，監控系統稽核記錄中是否有不當的使用行為、系統資源是否有不正常的耗損等。

主機型入侵型偵測系統不同於網路型，不需要獨立運作的主機，但必需在每台受保護的主機上皆安裝入侵偵測之軟體。因此，若區域內有多種工作平台及作業系統，就必需考量軟體相容性的問題。

表 2- 1 網路型與主機型入侵偵測系統比較表

特性 \ 種類	網路型入侵偵測系統	主機型入侵偵測系統
監控對像	網路封包	稽核日誌、系統狀態
配置方式	獨立監控主機	每台電腦皆安裝軟體
佈署位置	需設在各網路封包必經過之網路節點	不拘
與作業系統的相關性	低(因只需一台主機)	高
對於硬體的要求	高(因獨立監控全區域)	低
即時性	高	低
監控範圍	區域	主機本身
對加密封包的監控	難	易
入侵的證據抹除	難(因封包會被記錄)	易(入侵者可修改)
佈署成本	低	高
產品	網路硬體設備、軟體	軟體

2.1.2 入侵偵測系統的組成[13]

入侵偵測系統以架構來區分，可分為「集中式架構」以及「分散式架構」兩種類。一般而言，集中式架構之入侵偵測系統是由資料收集模組、入侵分析模組、以及回應模組所組成。若為分散式架構，還會多有個協調模組。下面即各模組之功能介紹。

1. 資料收集模組

資料收集模組主要是提供原生資料(raw data)輸入給入侵偵測系統，以供其判斷是否有入侵事件之發生。以網路型而言，原生資料即為網路封包；而主機型的原生資料為系統日誌檔。

2. 入侵分析模組[14]

入侵分析模組主要的功能是分析處理各種由資料收集模組所提供的原生資料，以判斷其中是否含有攻擊行為模式。常見的入侵判斷分析技術有下列幾種：

A. 入侵規則(Rule-based)：

利用各種語法來描述入侵事件與建立事件特徵的規則資料庫，以利入侵行為的比對。通常使用在不當行為(misuse)的偵測技術上。

B. 異常統計(Statistic Analysis)：

先收集一段正常期間的資料，再利用統計方法來建立基本的比對標準，若偏差值過大即判斷為不正常現象。此法通常使用在異常行為(anomaly)的偵測技術。此方法的優點是計算快速、能處理大量資料且能節省記憶體空間；缺點則為誤判率比一般方法高。

C. 有限狀態分析(Finite State)：

利用一群不同的狀態轉換來描述入侵事件。因為不同狀態間之轉移有某程度的相關性，此關係即是觸發狀態轉移之事件。

D. 類神經網路(Neural Network)：

具有自動學習的特色，只要能夠給予適當的訓練，就能使其具有辨別入侵行為的能力。此方法的優點是能偵測未知的攻擊型態；而缺點則是計算費時複雜。

E. 貝氏網路(Bayesian Network)：

依據貝氏機率所衍生出來的方法，具有學習能力。可應用於分散式網路，根據各別入侵分析模組所偵測出的獨立特徵而找出相關入侵行為。

F. 資料探勘(Data Mining)：

利用資料探勘的技術可以有效地將資料分類並且減少不必要的資料比對、找出對應入侵攻擊最匹配之特徵，且能產生新的規則以偵測未知的入侵行為。此分析技術還可配合模糊理論(Fuzzy Theory)來做特徵比對。

表 2- 2 各種分析技術與偵測方式的對應關係[14]

偵測方式 / 分析技術	異常(anomaly)	不當行為
入侵規則		V
異常統計	V	
有限狀態分析		V
類神經網路	V	V
貝氏網路		V
資料探勘	V	V

3. 回應模組

回應模組主要是發出警告，以告知管理員目前系統可能正遭受入侵行為的攻擊，並提供此入侵行為的相關資訊，以做相關的應變措施。一般而言，此模組可以分為被動式的回應與主動式的回應兩種類。被動式的回應即是單純地發出警告訊息給系統管理員；而主動式回應則是會自動針對入侵行為來產生相對的應變動作，例如，調高事件檢視器的靈敏度、配合防火牆直接擋下相關封包、對入侵者發佈警告訊息等相關動作。

4. 協調(coordinator)模組

協調模組為分散式入侵偵測系統才特有的元件。它的功能在於接收各個分析模組所傳送之警告，並在辨別其警告內容後，將該警告轉發到其它分析模組，以協助多個入侵分析模組來共同完成分散式入侵行為的偵測。

2.1.3 偵測方式

一般入侵偵測系統的偵測方式可分為「不當行為(misuse)偵測型」與「異常(anomaly)偵測型」與「混合式偵測(Mixed and hybrid mode detection)」三種方式。以下將分別介紹此三種偵測方式的特徵，以及各偵測方式之間的差別比較(表2-3)。

1. 不當行為偵測型

不當行為偵測型也稱為特徵型偵測(Signature Based Detection)，主要是針對過去已知的各種入侵種類之攻擊樣式(pattern)來建立行為比對的特徵(signature)資料庫，如果當系統偵測出使用者有此行為時，便視為其為異常行為，並且產生警告以告知管理者。

不當行為之偵測方式的重點在於如何定義與建立大量的攻擊特徵，如此才有能力來判別出各種攻擊行為；反之，如果在特徵庫中沒有定義特徵的入侵行為即無法辨別，將視其攻擊為正常合理的行為。因此當新型態的攻擊剛開始出現時，會產生一個問題，即False Negative出現的機率很容易偏高，此時的解決方法便是趕緊定義新攻擊的特徵，並且更新其資料庫。但伴隨著特徵庫的成長，所需比對的特徵量越來越大時，系統的效能也會相對隨之降低。

2. 異常偵測型

異常偵測型的偵測方式剛好與不當行為偵測型相反，此方法不是藉由不當的異常行為來建立比對資料，而是經由一段時間的觀察，及收集系統正常行為與正常狀態的資料來定義正常的行為模式，以產生比對標準。若偵測時發現有不符合此正常行為模式時，便將此行為判斷為異常行為。

採用異常偵測技術的重點在於如何正確有效地定義”正常行為”，而此決定於一開始系統的訓練是否良好、訓練的資料是否正確等，因為這些都將對未來的判斷產生重大的影響。但由於正常行為的定義很困難，常會隨著時間、環境而改變，所以要定義地詳細完整是非常不容易的。因此異常偵測技術有著Positive False出現機率偏高的缺點，容易把合法的行為判斷為不合法。

表 2- 3 異常偵測與不當行為偵測之比較

偵測方式 / 特性	異常偵測	不當行為
比對模式	正常行為	異常的攻擊行為
False Positive 型誤判之機率	高	低
False Negative 型誤判之機率	低	高
是否能判斷未知的攻擊	能	不能
是否常需更新比對資料	不需要	需要
初期是否需要 Training	需要	不需要

3. 混合式偵測

混合式偵測是將不當行為偵測及異常偵測兩種方式混合使用，以對此兩種偵測方式的缺點產生互補作用，期待能達成正確判斷的高偵測率，但同時也能有效地控制誤判率在某一程度以下。雖混合兩種偵測方式，但根本上仍是以其中一種為主要方式，另一種只是輔助，而混合的方式也必然有著高成本與效能較差的缺點。

由上述幾個小節的介紹我們可以大致上對入侵偵測系統有著初步的了解。接下來將介紹的是相關之問題與技術。

2.1.4 Snort[10]

Snort 是一套以「特徵規則為偵測基礎」的網路型入侵偵測系統軟體，它有著(1)免費的開放原始碼 (2)可自由改寫程式 (3)偵測效能出色 (4)眾多外掛模組等優點。

如圖 2-1，Snort 的偵測流程一般可以分為五個步驟，分別是封包解碼、前置處理、偵測引擎、日誌記錄與警報、輸出模組等，每個程序均有各自負責工作，詳細內容如下所述：

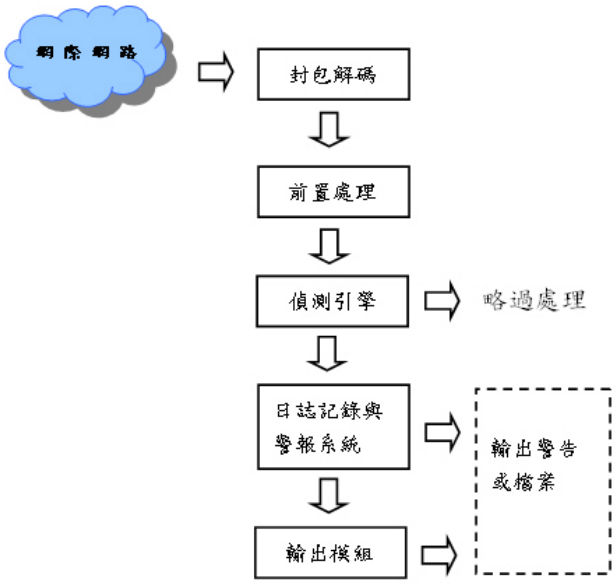

圖 2- 1　Snort 入侵偵測流程[12]

步驟一：封包解碼(Packet Decoder)

從網路介面卡擷取網路上的封包資料，然後再將封包交由前置處理或偵測引擎接著處理。

步驟二：前置處理(Preprocessors)

在封包交給偵測引擎處理之前，前置處理程序可以先針對封包進行重新排序或資料修改，以便符合系統的偵測需求。除此之外，有些前置處理也擁有檢查不正常封包的標頭並且產生警告能力。由於駭客會以各種不同的技巧來欺騙入侵偵測系統的規則比對偵測，因此一些事先的前置處理是必要的。例如，若某入侵規則為”scripts/iisadmin”時，駭客可以透過修改字串的變化，如下列所示，以躲避精確比對特徵每一字元之系統的偵測。

- scripts/./iisadmin
- scripts/examples/../iisadmin
- scripts\iisadmin

- scripts/.\iisadmin

前置處理也可對封包進行暫存與重新整理之處理。因為根據網路協定的定義，若資料大於最大傳輸單位(MTU)的話，封包資料即會被切割。此時若入侵偵測系統要對此被切割封包進行特徵比對就必須具有重組封包的能力。否則，假若當一個特徵字串正好被切割成兩個封包時，系統就會因此而無法偵測出來。

步驟三：偵測引擎(Detection Engine)

偵測引擎是 Snort 的主要核心部分，它透過規則比對的方式來判斷是否有任何入侵活動存在封包之內。一旦封包符合任何的入侵特徵時，系統就會接著做出相對應的動作；至於其它的正常封包，偵測引擎則會略過不做處理。一般而言，偵測引擎所進行比對特徵的部分包含下列所述：

- 封包的 IP 標頭。
- 傳輸層之標頭：TCP、UDP、ICMP 等其它傳輸層標頭。
- 應用層之標頭：DNS、FTP、SNMP、SMTP 等其它應用層標頭。
- 封包承載(Payload)的內容：檢查是否有任何的入侵特徵字串。

步驟四：日誌記錄與警報系統(Logging and Alerting System)

若經偵測引擎判斷為異常封包時，此異常封包的相關資訊將會被系統日誌記錄下來，且若此異常封包相對應的規則回應為警報時，系統就會發出警告，以告知管理者。

步驟五：輸出模組(Output Modules)

此模組也稱為外掛模組(plug-ins)，針對系統的日誌或警報做一些延伸處理，例如，將日誌記錄到資料庫、產生XML格式的輸出、更改路由器或防火牆的設定、送警告訊息到client端等。

2.2 False Positive

入侵偵測系統的誤判可分為False Positive，與False Negative兩種。False Positive的定義與所產生的相關問題會在2.2.1、2.2.2、以及2.2.3小節進行詳細介紹；而False Negative不在本論文的研究目標之內，因此不會進行深入的探討。

2.2.1 False Positive 簡介

入侵偵測系統所會產生的誤判種類可分為False Positive，與False Negative兩種，此兩種誤判的定義如表2-4所敘述。

表 2- 4 入侵偵測系統之誤判率種類

誤判率的種類	說　　明
False Positive	實際上沒有入侵行為，但系統卻產生有入侵行為的判斷。
False Negative	實際上有入侵行為，但系統卻沒有偵測出來。

False Positive一直以來都是限制入侵偵測系統效能的主要因素[5]，也是入侵偵測系統架構設計改良的主要目標之一。為了可以更加了解降低False Positive的重要性，以下首先介紹各種可能伴隨False Positive所而來的各種問題：

- **False Positive是限制入侵偵測系統效能的限制因素**

一般而言，False Positive的產生是由於入侵行為的判斷規則定義得不夠精確所造成。但如果入侵行為的判斷規則定義得太過嚴格又會使系統的False Negative型誤判的機率提高，也就是會造成系統之偵測敏感度下降而遺漏判斷入侵行為的狀況。因此入侵偵測系統的設計往往需要同時考量此兩型誤判所出現的機率，期望能使兩者皆達到合理可接受的標準之上。

- **目前已有針對False Positive來攻擊入侵偵測系統的程式**

此類型的攻擊，最主要的概念是產生大量的偽造封包，而這些封包都有符合”含有攻擊特徵而無真實攻擊行為”之特徵，因此可以觸發大量的誤判警告。接著入侵攻擊者可以把真實攻擊的封包隱藏於這些偽造封包之間，讓系統管理者忙於處理這些大量的警告而無法判別出真正的攻擊行為。更詳細的內容之後會再進一步地介紹。

- **犧牲系統效能以換取更高的偵查靈敏度**

不僅僅是入侵偵測模式的選擇會影響系統的效率與判斷的正確性。同樣地，入侵規則定義的嚴謹程度也會影響入侵偵測系統的偵測效率。較不嚴謹的入侵定義雖能提升系統偵查的靈敏度，偵測出較多可疑的入侵行為，但同時也會提升了False Positive出現的機率。

- **管理者對系統的不信任**

如果無效警告過的出現於頻繁或是比率過高，這將會使得管理者對於入侵偵測系統的偵測正確性開始產生懷疑，而無法完全相信其判斷結果。

- **False Positive所伴隨產生的警告，會造成系統管理員的負擔**

系統管理者的工作原本就很煩雜，如果入侵偵測系統時常會產生大量之無效警告的話，這會使系統管理者光光是處理這些警報就無能為力去處理其它工作，或是根本就對這些警告的處理產生無力感。

2.2.2 False Positive 攻擊

在「Controlling Intrusion Detection Systems by Generating False Positive：Squealing Proof-of-Concept」[8]一文中，作者W. Yurcik介紹一種與False Positive相關的新攻擊分類，並為之取名為「Squealing」。攻擊的方式是製做符合攻擊特

徵的封包，並選擇性地觸發入侵偵測系統內特定的False Positive警告，讓系統管理者對於某特定入侵警報之判斷力鈍化，或是疲於處理大量的無效警告而忽略了真正的攻擊。由以上嚴重的後果可知，Squealing攻擊類型已經曝露了入侵偵測系統之False Positive所會造成的弱點。

Squealing的攻擊分類可分為公開(overt)攻擊與隱藏(covert)攻擊，而這兩大分類下還可再細分為「狀況攻擊」、「否認攻擊」、「攻擊誤導」、「噪音掩蔽攻擊」、「統計毒藥攻擊」等五項攻擊類別。以下即為各種攻擊類別的詳細介紹：

- **公開攻擊**

1. **狀況攻擊(Conditioning attacks)**

如果找到非惡意的警告的話，系統管理員通常會拿它來過濾因False Positive所產生的無效警告。因此入侵者可以針對系統某特定的活動來進行鈍化程序，鈍化入侵偵測系統的判斷靈敏度，而對於該攻擊活動可達成忽視或視之無效的效果。

2. **否認攻擊(Repudability)**

提高False Positive出現的機率與策略性觸發False Positive警告可使系統管理員分不清楚入侵警報的有效性，因而無法從中判斷出真正攻擊行為，並且懷疑入侵偵測系統的可信度。

3. **攻擊誤導(Attack misdirection)**

發送偽造來源位址的誘騙封包以進行攻擊。例如，駭客可以偽造成攻擊目標的重要客戶或合作廠商之網路位址來進行攻擊，以期造成兩者之間的聯繫阻礙。

- **隱藏攻擊**

1. **噪音掩蔽攻擊(Noise-masking attacks)**

攻擊者能以”不同來源位址”與”符合不同攻擊特徵”之假資訊來產生大量False Positive偽造封包，以針對目標主機進行攻擊，藉此轉移系統管理員的注意力在公開的攻擊上，而無法在一堆隱藏的假攻擊封包中分辨出真正的攻擊。

2. **統計毒藥攻擊(Statistical poisoning attacks)**

這通常發生在入侵偵測系統初期剛在訓練與校調系統，以適應環境的時候。攻擊者在此時可以蓄意餵食不正確的資料給入侵偵測系統，讓系統管理員誤用不正確的統計結果來校調，以造成入侵偵測系統日後的偵測誤差。

隨著Unicode的出現，入侵特徵的判斷越來越困難，但最終的問題仍是如何用運用各種的分析方法來辨別出假扮特徵字串以正確找出入侵活動。

2.2.3 False Positive 攻擊的解決方法

針對False Positive各種的攻擊，Patton與Yurcik在[6]一文中提出了以下兩種可能的解決方案。但由於適應法有著實行上先天的限制，因此作者認為兩方法相較起來，狀態察覺法會是較好的解決方法。

- **適應法(adaptation)：**

入侵偵測系統在運行中可以隨機改變特徵比對的演算法，這可以防止各種False Positive攻擊以基於入侵偵測系統之規則知識而進行的攻擊。但此方法會產生以下兩個問題：

(A) 並非所有的攻擊特徵皆會有多種方式、參數去剖析以符合演算法

(B) 如果攻擊程式能學習適應，這將使Squealing之攻擊更加難以偵測

- **狀態察覺法（state awareness）：**

為了速度與處理能力上的考量，一般入侵偵測系統僅會獨立地檢查個別封包，並且只著重在標頭(headers)與承載(payload)的特徵比對上，而沒有考量先前或事後的事件與次序關性。但如果入侵偵測系統能有個記錄事件脈絡的文件，負責監控異常封包間之相關性的話，則能增加判斷入侵行為的有用資訊，以便過濾一些”從事件記錄間即可判別有異狀的封包”。

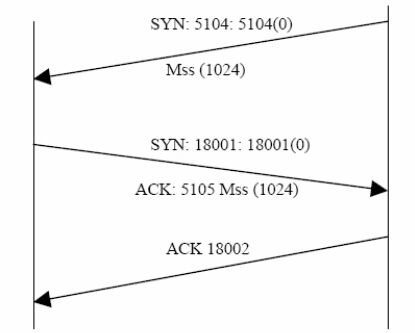

圖 2- 2　TCP 協定之三向交握流程[6]

例如，在TCP協定的三向交握(Handshake)過程中，如圖2-2所示，如果入侵偵測系統能夠記錄封包序號、傳遞的SYN/ACK，SYN及PS/ACK封包等相關資訊，以及資料傳遞的方向時，攻擊者要進行Squealing攻擊就會變地很困難。因為我們可以預期在同樣埠號(port)溝通的封包必定是一來一往，而不會出現僅有進而沒有出的異常現象。

近來，鎮壓False Positive型誤判率的解決方法尚未浮現，但這問題卻隨著新型攻擊的發展而變地越來越嚴重，這代表著入侵偵測系統必須守護越來越多的攻擊類型。然而，由於偵測的不準確，攻擊者可以故意產生誘騙的偽造封包來觸發False Positive，這將使False Positive的問題變地越來越嚴重。

2.2.4 共同刺激機制(Co-Stimulation)

共同刺激機制應用在入侵偵測系統上，最早是在Hofmeyr的LISYS[7]架構(一種網路型的入侵偵測系統架構)中為了讓系統管理員能控制入侵偵測的判斷結果而提出。所謂的共同刺激機制是指進行「二階段確認的偵測方式」：第一階段為網路型入侵偵測系統的偵測，若當此階段偵測到入侵行為時，就必須進行第二階段的偵測，讓系統管理員以E-mail的方式提供第二個偵測判斷。而每一階段的偵測結果可視為一訊號，當兩個階段的偵測訊號皆為真(判斷有入侵行為)時，才會判定確實有入侵行為。

但是在第二階段偵測中以E-mail的方式來提供第二個判斷訊號，不僅會造成系統管理員額外的負擔，也因需要人工的確認而造成警報之延遲。再者，E-mail需要依賴系統管理員的經驗來判斷，這可能造成入侵判斷不夠公正客觀的狀況，而且系統也會因此而缺乏自動化的能力[9]。

基於上述[7]之缺點以及欲使共同刺激機制更能有效降低False Positive，在”A Network IDS with Low False Positive Rate”[9]一文中，作者提出AINIDS架構進行改進。其方法為不再使用「需要系統管理員判斷的E-mail來作為第二階段的判斷」，而是以更客觀、更合理的「入侵定義」自動透過Monitor Agents(輕量型之主機型偵測系統)監控主機上的狀況來判斷辨別。此入侵定義所採用的是最早提出入侵偵測概念的創始人Anderson所作之定義，如下所述：

”***任何意圖危害系統資源的完整性、私密性，以及可用性之行為即為入侵行為(Any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource.)”[3]***

在AINIDS架構下，由於第二階段是採用正確偵測率更高的Monitor Agents來再次確認是否有入侵行為發生，因此可有效地降低False Positive。

共同刺激機制的偵測流程主要可分為以下三個步驟，如圖2-3所示：

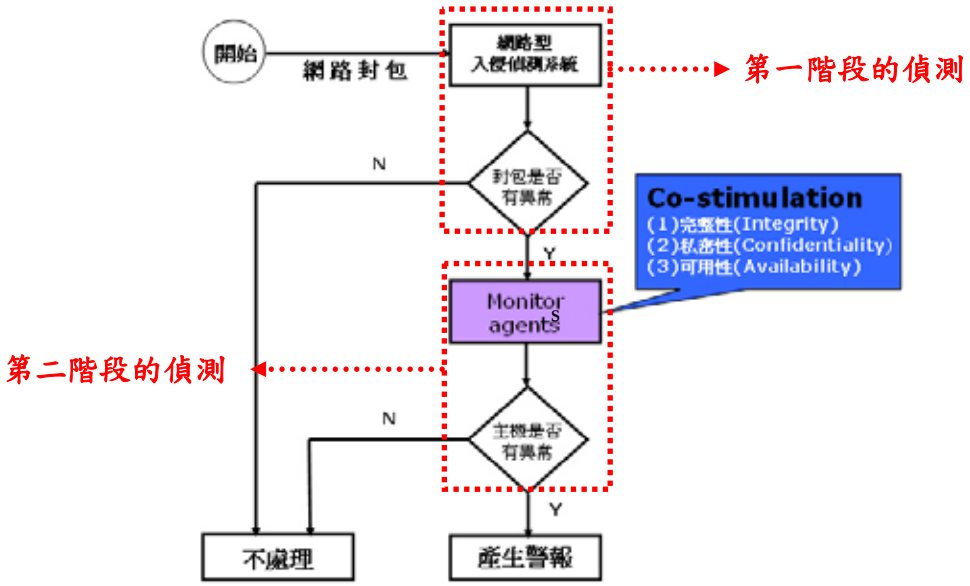

圖 2- 3 共同刺激機制之入侵偵測流程[9]

步驟一：此步驟為共同刺激機制第一階段的偵測。如一般偵測的過程，當網路封包通過網路型入侵偵測系統時，會先被判斷是否為異常之封包。

步驟二:若步驟一判斷為異常封包時,則會觸發受攻擊之主機的Monitor Agents來持續監控該主機之完整性、私密性與可用性等狀態；反之，若為正常封包時，則不會進行任何處理。

步驟三:此步驟為共同刺激機制第二階段的偵測。只要主機上的Monitor Agents判斷出任何異常狀況，即會判斷入侵行為確定成立，並且發出警告；反之，則代表系統在步驟一時為錯誤的判斷，所以不會對此異常封包進行任何處理。

■ **Monitor Agents**

在AINIDS[9]中，使用三種Monitor Agents去監控受保護之主機的系統狀態，此三種Monitor Agents所監控的範圍如下所述：

1.完整性監控代理人(Integrity Monitor Agents)：

專門監控系統的關鍵檔案，如果檔案有任何的改變可以產生相關的報告給系統管理員。例如，檢查系統的二元碼是否有被更改，系統日誌(system log)是否有被刪除，或系統安全性的設定有否受到非預期性的改變。

2.私密性監控代理人(Confidentiality Monitor Agents)：

此監控代理人主要的監視範圍包含了是否有任何違反系統控制政策(control policy)的行為產生，與是否有任何的安全性機密資訊洩漏等。實作上可以監控一些資訊控制列表內較敏感的指令，最主要是採用不當使用偵測(misuse detect)技術。

3.可用性監控代理人(Availability Monitor Agents)：

監控主機是否有發生系統資源不正常損耗的狀況。因為大部分的入侵行為皆會造成系統資源不正常的消耗，例如，記憶體、緩衝暫存器(Buffer)、硬碟空間，中央處理器(CPU)的使用率異常等。

實作上能以「統計異常(statistical anomaly)」的偵測技巧來進行，這樣的技術，就是一開始先定義被監控的主機之正常系統資源消耗的基準線(baseline)，然後再根據實際上與這基準線來的偏差值(deviation)來偵測可能的入侵行為。

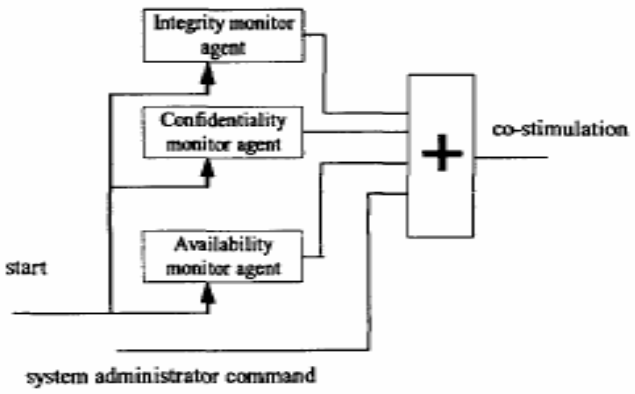

圖 2- 4 共同刺激訊號[9]

如上圖所示，在共同刺激機制中，只要完整性、私密性、可用性監控代理人，三者任一發現系統之異常狀況時，即可確認為入侵行為成立，並且發出警告。而非三個代理人皆必須同時發現系統異常狀況才能判斷為入侵行為成立。

第三章　一個新的以共同刺激機制為基礎之入侵偵測架構

本章主要討論一個能有效降低 False Positive 的入侵偵測系統需求與系統設計考量。由於 Yan Qiao 與 Xie Wei Xin 所提出的「共同刺激機制」[9]之系統架構擁有降低 False Positive 的特性，因此本論文以此機制為基礎，提出一個入侵偵測系統架構，並且融入異常封包分類處理的方式，以增進系統的偵測效率且能更加有效地防範網路上各種入侵攻擊行為。

3.1 一個新的以共同刺激機制為基礎之入侵偵測架構需求

關於 False Positive 的解決辦法，主要可以分為兩個不同的進行方向：一個是針對入侵偵測系統的架構作改良；另一個方向是針對分析模組所選取的分析技術作改進。下表為此兩者的例子說明：

表 3- 1 False Positive 兩個解決方向之說明

False Positive 的解決辦法	說　　明
入侵偵測系統架構上的改良	例如： (1) 結合多個不同種類的入侵偵測系統，共同分析判斷入侵行為。 (2) Cisco Threat Response 的技術：利用虛擬主機來判斷異常狀況，以確認入侵行為。並設有攔截機制，可從中直接阻擋攻擊封包。官方數據說明能有效降低 90%以上的 False Positive 型誤判。
分析技術的改良	例如： (1) Misuse 偵測方式，在第一、二代僅僅是採用 Pattern Marching 的技術，而第三代則加入了網路協定分析技術。 (2) Signature 比對的範圍從 String-based 再增加 Context-based 比對，也就是能針對活動、語意、前後之關系進行判斷。

基於「分析技術」上的改良相對於「入侵偵測系統架構」上的改良需要有較高之運算能力與處理能力，因此較容易因所需偵測的資料量超過負荷而造成系統出現不正常運作的狀況。經評估系統所需要處理的資料量後，本論文以架構上的改進為主要方向。

在 Yan Qiao 與 Xie Wei Xin 所提出的「共同刺激機制」[9]之系統架構擁有可降低 False Positive 的特性。雖然此機制需要進行第二階段額外的 Monitor Agents 監控偵測動作，但此是交由受保護的主機來執行，並不會耗費網路型入侵偵測系統太多資源。因此本論文以此架構為基礎，並考量下列幾點需求，設計一個入侵偵測系統架構。

- **異常封包分類處理的能力**

由 Monitor Agents 所監控的私密性(Integrity)、私密性(Confidentiality)、可用性(Availability)可知 Monitor Agents 其實是輕量級之主機型入侵偵測系統(Light Host-IDS)。由於主機型入侵偵測系統先天上偵測範圍之限制，所以並不是所有網路型入侵偵測系統所能偵測出的攻擊，主機型也都能偵測得出來[15]。此狀況衍生出以下兩個問題：

(1) Monitor Agents 做白工：

Monitor Agents 在系統一開始時並未啟動，而是需等到有異常封包通過第一階段偵測之後，被判定為異常才會被觸發而開始運作。但如果此時 Monitor Agents 監控的是一些超過本身偵測範圍之外，而無法掌控與辨別之異常封包時，就會造成系統資源無效監控的浪費。

(2) 產生 False Negative：

當第一階段網路型入侵偵測系統所判別出的異常封包是在主機上不會產生明顯異常狀況的攻擊封包時，若再將此類可疑封包進行第二階段 Monitor Agents 之判斷處理，則系統最後會將此類異常封包判斷為正常封包，而造成 False Negative 型誤判之狀況，也就是入侵偵測判斷漏失的現象。

針對上述的問題，本論文提出在第一階段被網路型入侵偵測系統判別為異常

的封包都必須先進行過濾的處理程序。將一些在主機上並不會產生明顯異常狀況的攻擊封包抽取出來，以進行另外不同的處理方式。如此才能在共同刺激機制的系統架構中真正地解決上述的兩個問題。

- **能夠抵擋 DoS 的攻擊**

近年來網路上最熱門的攻擊行為除了系統漏洞攻擊、網路蠕蟲、特洛依後門程式外，就屬阻斷服務攻擊(DoS)所造成的危害最為嚴重。此 DoS 可針對系統資源或者是網路頻寬來進行攻擊，讓系統無法正常運作或是無法正常提供服務，因此在設計入侵偵測系統時也需要特別考量如何來針對此類型的攻擊行為進行防範。

主機型的入侵偵測系統可能會因為 DoS 的攻擊而失去作用[15]。因為當主機的 Monitor Agents 發現網路頻寬異常或系統狀況異常時，可能再也無法順利發出警告封包告知防火牆把此類攻擊封包阻擋下來。因此在共同刺激機制之下，此類的攻擊應當交由網路型入侵偵測系統來處理會比較恰當，也才能即時擋下此類異常封包。

因為有攔截 DoS 之攻擊封包的需求，所以在網路型入侵偵測系統架構的設計上需要有封包攔截之模組或是另有防火牆(Firewall)來搭配處理才能夠達成。在系統佈署上，防火牆與網路型入侵偵測系統皆必需配置在網路的線性位置上(Online)，也就是所有封包必需經過之網路節點上，如此一來才能夠完整地監控整個組織的網路狀況。

- **能夠偵測 Squealing 類型之攻擊**

現今入侵偵測系統之設計，大多僅考量該如何降低 False Positive 型誤判的機率，而沒有考慮 Squealing 類型攻擊的偵測與判斷。但 Squealing 攻擊所造成的嚴重性在 2-2-2 小節已經討論過，此問題也已經提報給 CERT，其重要性可知。

再者，降低 False Positive 型誤判的機率並不意味著完全不會有 False Positive 狀況的產生。若入侵者以試探式的攻擊來尋找可能的 False Positive 漏洞時，共同刺激機制僅會把此類攻擊當成無實際攻擊的無效入侵行為，而不會做任何的反應或記錄機制，因此系統管理員並無法察覺有人正試圖攻擊，造成系統潛在的安全風險。由上述可知在設計入侵偵測系統時，對於針對 False Positive 弱點之

Squealing 攻擊類型不得不重視，必須擬定有效的偵測方法。

3.2 一個新的以共同刺激機制為基礎之入侵偵測架構

在 3.1 小節說明了基於共同刺激機制之入侵偵測系統的相關需求。因此在本節我們依據這些需求，提出一個能降底 False Positive 的入侵偵測系統架構，如下圖所示。本論文架構除第一階段網路型入侵偵測系統的偵測與第二階段 Monitor Agents 的監控外還加入了三個模組：異常封包分類模組、DoS 封包攔截模組以及 False Positive 計數器模組，在本小節會一一介紹。

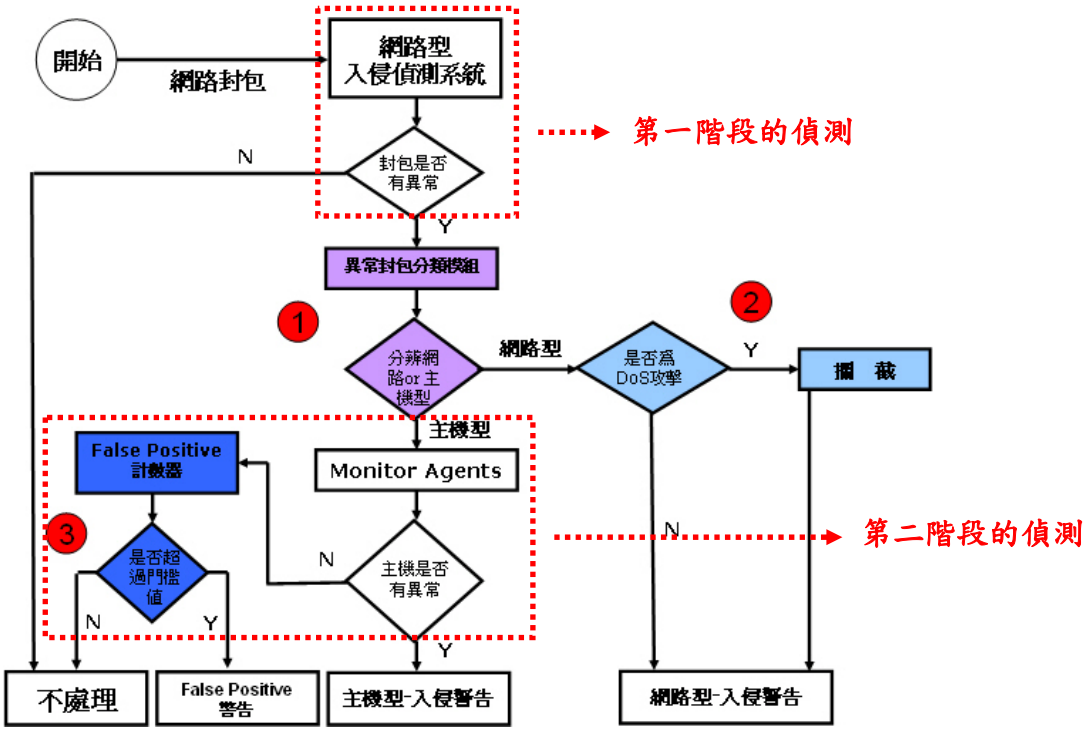

圖 3- 1 以共同刺激機制為基礎之入侵偵測流程

一、異常封包分類模組：

此模組的功能最主要是解決 Monitor Agents 對於第一階段網路入侵偵測系統所偵測出來的某些無法監控之異常封包(因不會在主機上造成異常變化)，或是偵測出來已經無法做出反應(因網路已被癱瘓)的狀況。解決方法是此模組會把被網路型入侵偵測系統所偵測出的異常封包預先分為兩大類：一是「網路型異常封包」，另一類為「主機型異常封包」。以下是網路型與主機型分類原則的介紹：

1.網路型異常封包

因不會在主機上造成明顯異常而超出 Monitor Agents 偵測範圍的異常封包，主要是針對「網路頻寬」與「網路協定的漏洞」進行攻擊。可以分為下列幾種封包類型：

(1)可疑的網路流量封包

(2)掃瞄類型封包

(3)偵測系統漏洞-弱點掃瞄程式

(4)不符網路協定規格之異常封包

2.主機型異常封包

會明顯造成主機上產生異常狀況且可被 Monitor Agents 所偵測的異常封包，主要是針對主機上之「作業系統與應用服務程式的漏洞」進行攻擊。可以分為下列幾種封包類型：

(1)使用異常網路埠(Port)

(2)不符合字串標準

(3)可疑存取行為

(4)異常的網路控制

隨著這兩種不同的異常封包分類會有著不同的處理流程，如圖 3-2 所示。網路型異常封包就同一般的網路型入侵偵測系統的處理流程；而主機型就必須再進一步接受 Monitor Agents 的監控，看看是否有異常狀況發生再做入侵偵測判斷，如此一來即可避免監控代理人處理到無能為力之封包的問題。

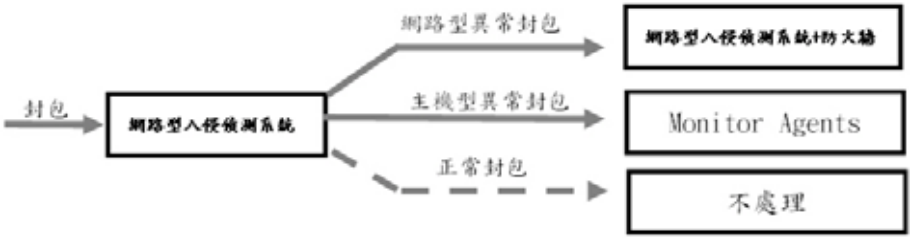

圖 3- 2 異常封包分類處理

二、DoS 封包攔截模組：

如 3.1 小節所說明，關於 DoS 的攻擊應當交由網路型入侵偵測系統配合防火牆來處理會比較恰當，而不須再透過 Monitor Agents 的監控判斷程序，如此才能即時阻擋此類異常封包。下圖為本系統的架構圖，防火牆與網路型入侵偵測系統皆配置於所有網路封包必須經過之位址，以完整監控整個網路狀況。

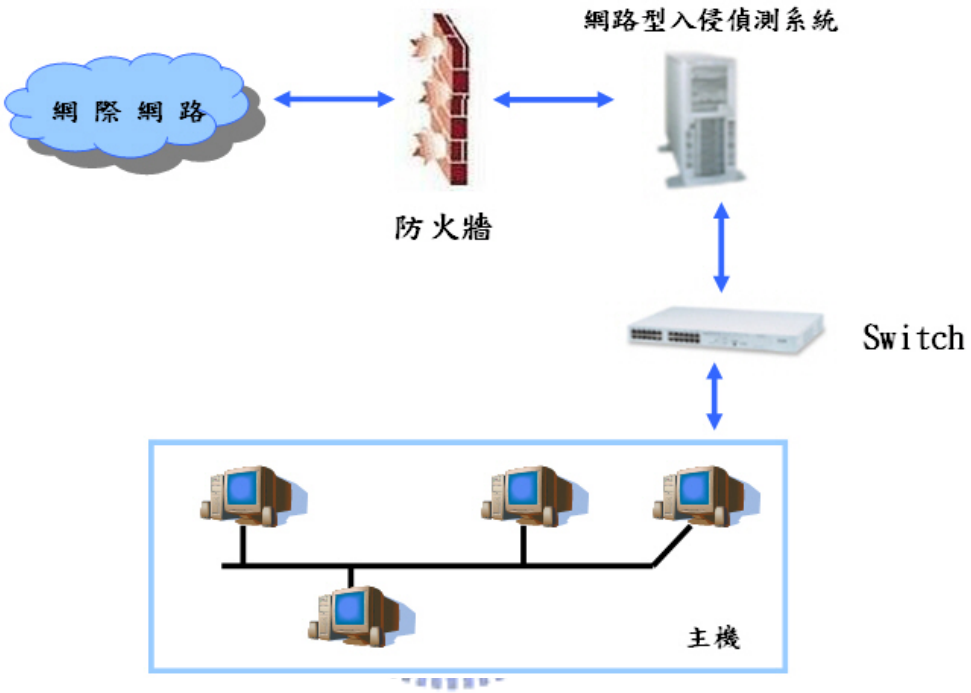

圖 3- 3　系統架構配置圖

此模組判斷是否為 DoS 攻擊封包的方式主要是針對網路型的異常封包來進行處理，下述步驟為整個攔截程序的流程：

步驟 1：若異常封包所符合之特徵規則為 DoS 攻擊類別時，系統會啟動門檻值的機制。

步驟 2：系統會同時監控區網內與區網外的網路流量狀態一段時間，判斷網路流量是否有超過門檻值。

步驟 3：如果當網路流量超過門檻值時，就會觸發動態封包攔截的機制，直接攔截可疑的封包。

步驟 4：當網路流量低於門檻值時，攔截程序即終止。並會返回步驟 1 的狀態下繼續監控。

由上述的步驟可知系統並不會一開始即主動攔截封包，而是經過現實狀況的確認之下，證實了網路流量產生不正常的狀況且已經達到可能會危害系統服務正常運作的程度(門檻值)時，才會立刻觸發自動防護攔截的機制。此作法是犧牲初期短暫遭受攻擊的風險，來換取更準確的封包攔截程序，以降低不正確攔截的情形。

三、False Positive 計數器模組：

當主機型異常封包被 Monitor Agents 監控時，主機若發生任何異常狀況，此主機型異常封包即會被判定為攻擊封包；反之沒有出現任何異常狀況時，該異常封包則會被判定為 False Positive 封包，如圖 3-4 所示。一般 False Positive 如果出現頻率不高時可能屬於正常的狀況，但如果頻率過高時，則表示可能有人在嘗試不當的入侵或探試系統，需要告知系統管理員善加注意。

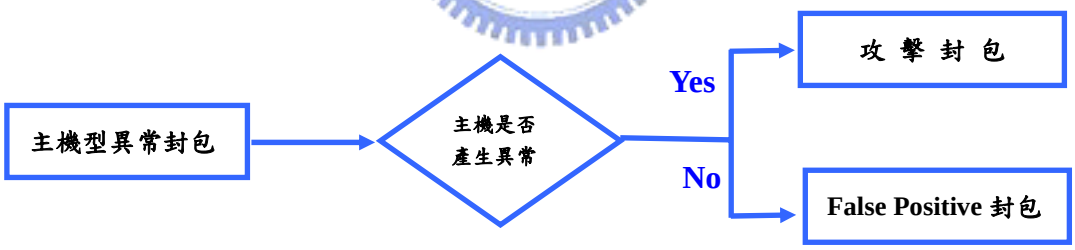

圖 3- 4　False Positive 封包的判斷

針對此點本論文提出一 False Positive 計數器模組。當 False Positive 產生時即開始計數，直到 False Positive 的產生暫停一段時間後才會終止計數。若在計數期間內 False Positive 產生頻率超過了門檻值，此模組會自動產生警告以告知系統管理員。除此之外，此模組還必須可以記錄一些簡易且重要的資訊以供管理員追蹤調查，例如：攻擊位址、時間、False Positive 次數等相關資訊。

3.3 討論

在本章中，我們依目前之網路安全狀況與新的攻擊類型來探討共同刺激機制之入侵偵測系統架構的缺點與安全議題，並且根據這些缺點提出一個新的入侵偵測架構，此新架構加入了封包分類處理模組、False Positive 計數器模組、DoS 封包攔截模組，以期望能達到(1)降低 False Positive 型誤判的機率、(2)能針對 Squealing 的攻擊類型產生警告，以及(3)改善原架構對 DoS 防止的問題等預期理想的實驗成果。

由於在本論文所提出之架構下，主要的偵測範圍還是基於原本系統的分析能力，所以對於網路型入侵偵測系統、Monitor Agents 以及異常封包分類模組的選取與設計，將對偵測結果產生重大影響。其各種可能發生的錯誤判斷之狀況如下表所述：

表 3- 2 本架構可能發生的錯誤判斷狀況

網路型入侵偵測系統	異常封包分類模組	Monitor Agents	結　　果
偵測遺漏	N/A	N/A	False Negative
正確偵測	正確分類	偵測遺漏	False Negative
正確偵測	分類錯誤	正確偵測	False Negative
錯誤偵測	正確分類	錯誤偵測	False Positive

例如，若異常封包分類模組把監控代理人所無法監控異常的網路型異常封包判斷為主機型的封包時，會因此而產生 False Negative，也就是會漏失掉此入侵攻擊警告。

第四章　系統設計與模擬

4.1 測試平台與環境說明

本論文實驗環境建構於交通大學管理二館之網路環境下，硬體設備有四台個人電腦與一台 SWITCH 以進行本論文「基於共同刺激機制之入侵偵測系統」的模擬實驗，詳細之硬體設備清單與網路環境如下所述。

(1)基於共同刺激之入侵偵測系統：

硬體：

CPU：Pentium 4 2.4G

RAM：768M

硬碟空間：30G

系統：

作業系統：Windows XP Professional sp2

資料庫：Mysql Version 4.0.23 for Windows

網頁伺服器：Apache Version_2.0.53 for Windows

(2)受保護之網路節點

硬體：

CPU：CELERON 1.8G/k7 1G/k7 600

系統：

作業系統：WindowsXP Professional/Windows2000 /FreeBSD 4.11stable

(3)開發平台

Windows XP Professional sp2

Visual C++ Version 6

PHP 4

(4)系統實驗架構

下圖 4-1 所示為本論文實驗進行之架構，共有四台電腦，分別擔任主動式入侵偵測系統(IDS + Firewall)、外部攻擊者及兩台在入侵偵測系統所保護區網內之受安全保護的角色。

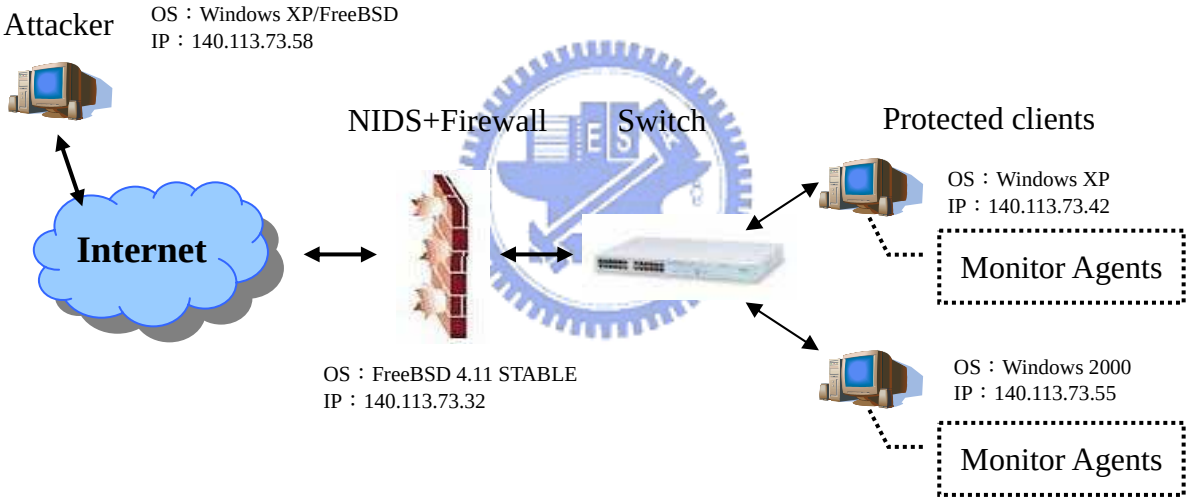

圖 4- 1 模擬實驗之架構

4.2 基於 Snort 之測試流程

依照本論文第三章中所提出的需求與架構，針對 Snort 來進行修改部分如下圖 4-2 所示，主要除了加入共同刺激機制的 Monitor Agents 外，還包括了修改偵測引擎以達成異常封包分類處理、防止 DoS 攻擊之攔截阻擋模組與偵測 False Positive 攻擊之計數器等。

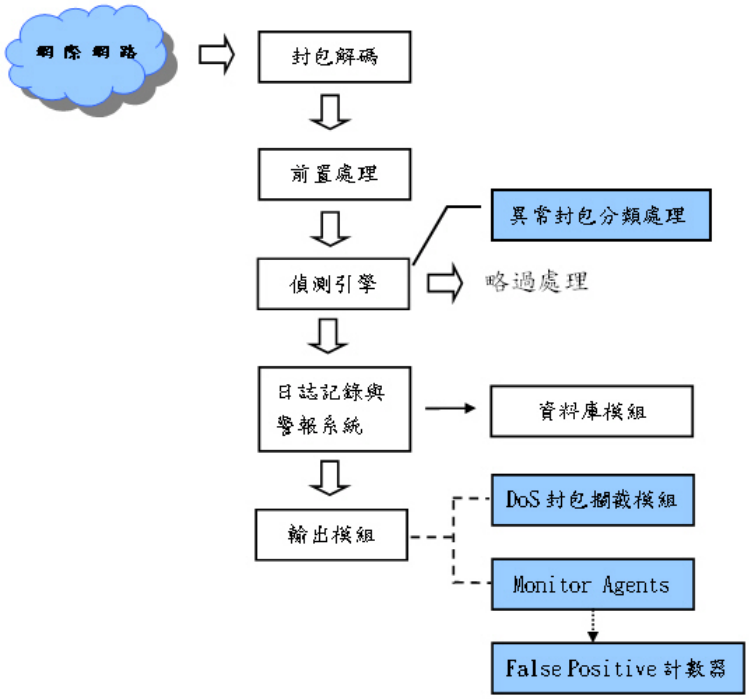

圖 4- 2 基於 Snort 之共同刺激機制入侵偵測流程

4.2.1 異常封包分類模組

異常封包分類模組主要是針對不同的異常封包進行分類，以利往後進行不同的處理活動。本論文針對 Snort 偵測引擎的部分作修改，將原本 Snort 的五類規則動作：Pass、Log、Alert、Activate、Dynamic，新增兩個規則動作，以分別針對 DoS 與主機型的攻擊封包做另外不同的處理。

- 分類原則：

如下圖 4-3，異常封包的分類原則主要可分為兩大類異常封包：「網路型異常封包」與「主機型異常封包」。這兩類異常封包的區別標準以「封包內的不正常活動是否會造成主機上產生異常狀況」為主，若可造成主機上之異常狀況即為主機型異常封包，反之則為網路型異常封包。

另外，網路型異常封包還可細分為「一般網路型異常封包」，以及「DoS 型網路異常封包」。為了遭受 DoS 類型的攻擊時可以直接把攻擊封包阻擋下來，因此 DoS 類型的異常封包必須被區分出來，以利進行不同於一般網路型異常封包的阻擋活動。

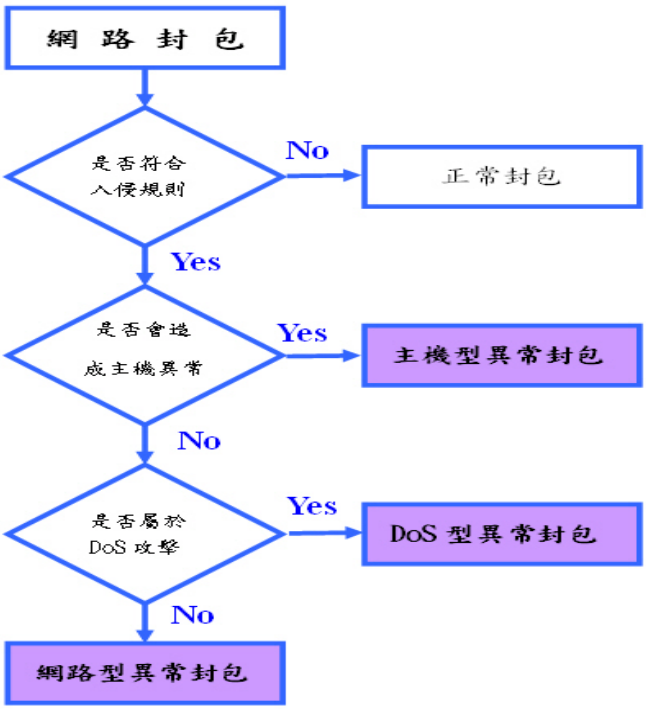

圖 4- 3 異常封包分類流程

- 新增規則活動類別：

圖 4- 4 Snort 規則定義之基本結構[11]

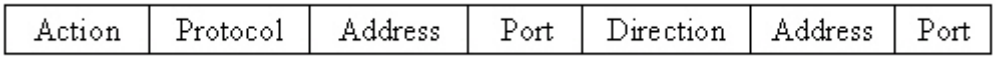

圖 4- 5 Snort 規則定義之標頭結構[11]

如圖 4-4，Snort 的規則定義是由 Rule Header 與 Rule Options 所組成。Rule Header 的內容包括規則活動(Action)、網路協定、來源端位址、埠號及目的端位址與埠號所組成，如圖 4-5 所示。其中所謂的規則活動是指針對不同入侵規則所定義的入侵行為會有不同的處理動作，而此相對應的處理動作即稱為規則活動。Snort 預設的規則活動，如下表 4-1 所描述：

表 4- 1　Snort 預設的規則活動說明[11]

規則活動	說　明
Pass	不處理
Log	記錄下相關資訊
Alert	發出警告並記錄資訊
Activate	發出警告並觸發額外的規則以檢查更多的狀況、更進一步地測試封包
Dynamic	配合 Activate，擔任額外被觸發的規則

為了達成分類處理的需求，如圖 4-6 所示，因此本論文依照 Snort 的格式新增兩類自訂之規則活動，以針對主機型異常封包及 DoS 型異常封包呼叫特別的處理模組，分別對應 Monitor Agents 及 DoS 封包攔截模組來進行特殊需求的處理。此兩類自定的規則活動如下所述：

1. Call Monitor Agents：自訂的規則行動。用來觸發 Monitor Agents 進行監控。

2. Call DoS block module：自訂的規則行動。用來觸發 DoS 動態阻擋模組。

圖 4- 6　異常封包之分類處理

4.2.2 Monitor Agents

此 Monitor Agents 為一主機型的入侵偵測系統，在所有需受保護的主機上都必須安裝。平時 Monitor Agents 並不會主動運作，而是在異常封包被判斷為主機

型時，才會觸發受保護主機上的 Monitor Agents 做監控。監控流程如圖 4-7 所示，監控內容主要是第二章所介紹過的主機之三種特性，分別為：完整性(Integrity)、私密性(Confidentiality)與可用性(Availability)。本論文採用 Norton Intruder Alert 來擔任 Monitor Agents 的角色。

如果在監控時間內主機有發生任何的異常狀況，例如，關鍵檔案被更改或系統資源耗損不正常等任何符合上述三種特性之異常狀況時，Monitor Agents 即會立即判定有入侵行為產生並發出警告；反之，若系統在監控期間內沒有發生任何的異常，系統將會自動結束監控活動並將此主機型異常封包判定為 False Positive 封包，以觸發 False Positive 計數器模組來針對 False Positive 類型之攻擊作額外的監控。

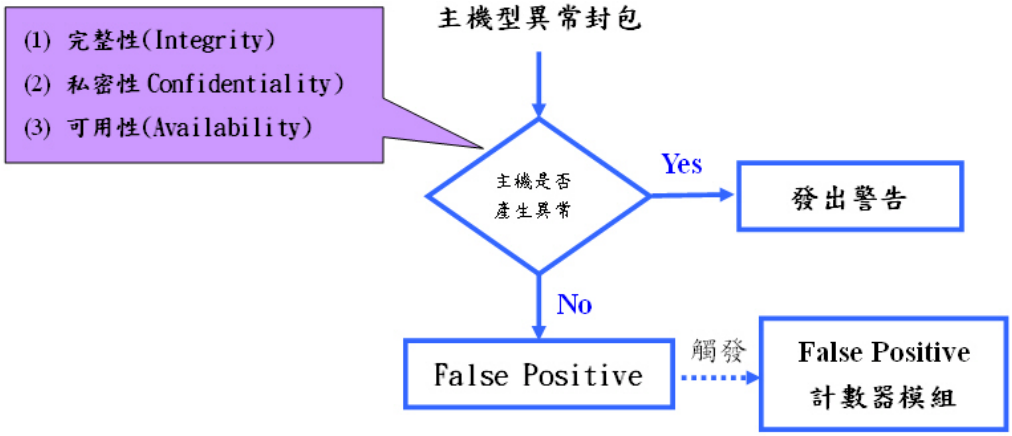

圖 4- 7　Monitor Agents 監控流程

4.2.3 DoS 封包攔截模組

封包攔截的動作是存在風險的，因為有可能因系統之入侵特徵判斷不準確而截斷了正常的網路溝通，進而造成比遭受入侵攻擊更大的損失，因此一般採用攔截機制所處理的異常封包必定含有以下兩個特徵：

1. 肯定此異常封包確實含有入侵行為。

2. 此異常封包必定會造成嚴重的影響。

基於上述因素的考量，本系統所設計之封包攔截模組僅針對 DoS 類型的攻擊封包進行處理，因為此類型的攻擊行為明顯，常伴隨異常的網路流量，且會造

成系統無法正常服務的嚴重影響，因此適合採用。

為了更加確認 DoS 的異常封包是否含有攻擊行為，本模組攔截機制的設計並非一開始即馬上啟動，而是需經過門檻值的再次確認，證實有被攻擊的實際狀況才會啟動。所以要觸發攔截機制必需同時符合下面兩個條件，流程如圖 4-8：

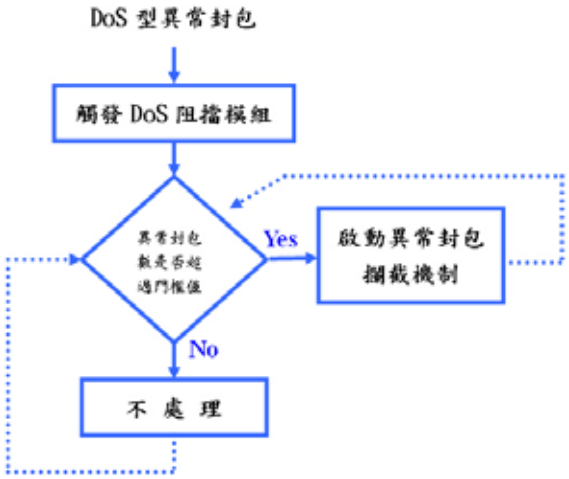

圖 4- 8　DoS 封包攔截模組處理流程

- 必須經「異常封包分類模組」判斷為 DoS 型之異常封包：

一般而言，DoS 攻擊會造成被攻擊的目標主機當機、重新啟動，或是該主機所處的網路壅塞，以致無法正常提供服務。而本 DoS 封包攔截模組僅是針對造成網路壅塞的攻擊封包進行處理，其餘會造成主機狀況明顯影響的類型則屬於 Monitor Agents 的監控範圍。

- 必須超過門檻值

在此，門檻值所指的對象是含有 DoS 特徵之異常封包總數，例如 ICMP 封包與廣播封包等，而非指所有的網路流量狀況，如此可以避免剛好因系統正在進行大量資料傳輸時所可能造成的誤判。

另外，本模組還有一個機制，稱之為「動態攔截」。意指 DoS 異常封包流量一超過門檻值即會啟動封包攔截機制，但如果一段時間過後又低於門檻值時，即又會停止攔截。如此一來，便可讓入侵讓偵測系統在可接受的攻擊風險下適度地調整異常封包的管控政策。例如，當超過門檻值時，即關閉 ICMP Echo Reply 與區網廣播的功能，當低於門檻時又開放此類的封包流通，以達到彈性管理的優點。

4.2.4 False Positive 計數器模組

此模組用來處理經過 Monitor Agents 監控而沒有出現異常狀況的封包，特別是用來偵測 Squealing 類型的攻擊。主要處理的程序分成兩個部分：(1)記錄下表 4-3 所列出的相關資訊以供需要時使用，(2)計算次數判斷是否有超過門檻值，若超過門檻值即發出 False Positive 警報。

表 4- 2　False Positive 計數器模組所記錄的資訊

欄 位	說　明
Source_IP	攻擊端的來源位址
Time_stamp	攻擊時間
Attack_type	攻擊的類別
Counter	計數器

False Positive 發生的原因有下列幾項：

- 入侵偵測系統的誤判：

  單純是因入侵偵測之判斷規則不夠準確所造成的誤判。

- 無效的攻擊行為：

  當攻擊封包遇到不對應的作業系統或應用程式時，即會成為無效之攻擊。例如，當專門攻擊 SQL Server 之 Slammer 蠕蟲遇到 Unix 系統或者是沒有安裝 SQL Serve 之主機時，當然無法產生有效之攻擊行為。

- Squealing 攻擊：

  刻意製造出 False Positive 的攻擊類型。一般而言 False Positive 出現的頻率只要不要太高都屬於正常現象，一旦頻率過高時則表示有不正常的行為活動發生。尤其當 Squealing 新攻擊類型被提出後，許多原本拿來測試入侵偵測系統的 False Positive 封包產生程式(例如，stick、sneeze 等相關程式)，也變成了駭客的攻擊工具，這使得入侵偵測系統的設計廠商及組織內的網路管理員不得不正視 False Positive 所產生的相關問題與挑戰。

4.2.5 系統模擬架構

依照前一章節所定義的系統需求與模組設計，模擬一個共同刺激機制的入侵偵測系統架構，系統流程圖如下：

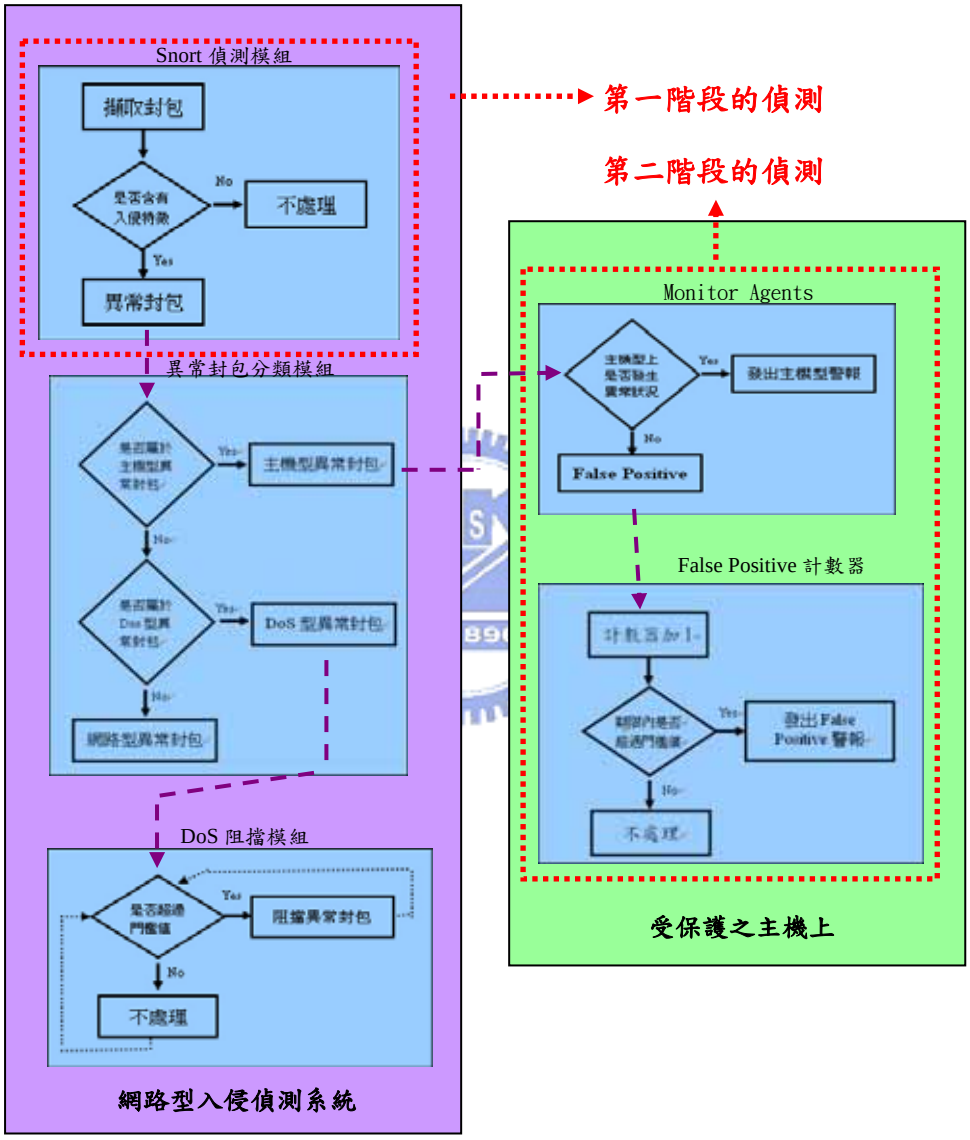

圖 4- 9　共同刺激機制之入侵偵測系統架構模疑圖

如圖 4-9 所示，系統架構共分為五個部分，其中異常封包分類模組與 DoS 阻擋模組存在於 Snort 入侵偵測系統的內部之中，負責擷取、偵測異常以及異常封包分類處理的工作。而 Monitor Agents 模組與 False Positive 計數器模組為一支

獨立的程式，安裝在受保護的主機上，負責監控主機的系統狀況，以判斷主機上是否有入侵行為的產生。

DoS 阻擋模組，會針對 DoS 型的攻擊封包及門檻值判斷，來動態阻擋異常封包，以避免遭受 DoS 型的攻擊。而 False Positive 計數器模組則會針對異常的 False Positive 頻率發出警告，以偵測出 Squealing 類型的攻擊。

4.3 模擬結果與分析

本小節主要針安全性來進行實驗，驗證本論文基於共同刺激機制所提出之入侵偵測架構的安全性。實驗主要分為二個部分：(1)False Positive 攻擊，與(2)一般長期監控狀況。

4.3.1 安全性

- False Positive 攻擊

實驗步驟：

1. 先將 Snort 規則中，屬於主機型攻擊之入侵規則挑選出來，以供測試。
2. 利用 False Positive 之封包產生程式”sneeze”來讀取上步驟所挑選出之主機型攻擊規則，以產生多次偽造封包的攻擊來進行測試。由於這些封包是針對 Snort 的設定檔所產生，所以皆會符合 Snort 的入侵偵測規則，而沒有實際的攻擊行為。
3. 監控被攻擊端主機的狀況，看看是否有符合預期狀況。

實驗結果：

在實驗步驟一中，可能由於”sneeze”程式尚未隨 Snort 版本更新，造成原本 Snort 規則中有 46 種規則類別，經過濾後僅剩 11 種規則類別能正常運作且能產生 False Positive 的無效攻擊封包，而這 11 種規則類別還必需扣除三種不屬於

主機型異常封包所對應的規則類別，最後符合實驗需求的僅剩 8 種規則類別，如表 4-3 所列。

表 4- 3 篩選過後符合需求的規則類別

編號	特徵規則類別名稱	攻擊類別數	警告類別數	總警報數
01	backdoor.rules	2	4	4
02	bad-traffic.rules	1	2	2
03	ddos.rules	1	1	14
04	deleted.rules	3	4	9
05	dns.rules	2	4	8
06	dos.rules	3	4	4
07	rpc.rules	1	1	27
08	snmp.rules	2	6	26
總合		16	26	94

表 4-4 為「步驟二」進行多次 False Positive 攻擊的實驗數據結果。在實驗過程中，所有通過第一階段，由 Snort 判斷含有入侵特徵的異常封包，在經過第兩階段的 Monitor Agents 監控後皆被過濾掉，且沒有產生任何主機型的警告。但隨著 False Positive 攻擊次數的不同，會觸發 False Positive 警告數也不同。

表 4- 4 攻擊的次數與所觸發 False Positive 警告整理表

	1次	5次	10次	20次	50次	100次	200次
False Positive 警告數	0	1	3	6	8	17	17

由表 4-4 的數據我們可以發現 False Positive 警告數一旦達到某個程度時就不會再增加，這是由於本實驗的門檻值設定為 60 次/分鐘，所以當大多數的入侵規則皆達門檻值之後，警報數的增加幅度就會趨緩，因為在固定的期間內，觸發相同規則的 False Positive 警告僅會發出一次。

經由 False Positive 攻擊實驗之結果，可知本論文所提出的共同刺激架構確實能大量過濾掉僅有攻擊特徵而無攻擊行為的警報，並且能針對異常的 False Positive 頻率發告警告。

■ **長期網路監控**

實驗操作：

持續監控 25 天交通大學網路環境下的入侵偵測狀況。並根據所記錄的數據作解讀與分析。

實驗結果：

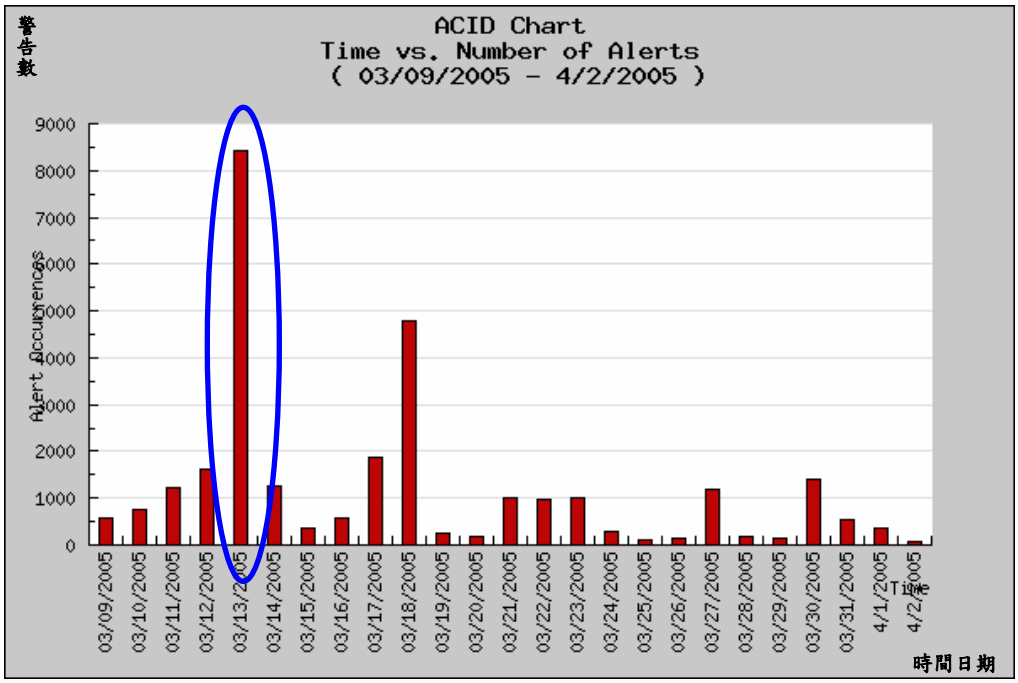

圖 4- 10　監控交通大學內網路異常狀況記錄

上圖為監控期間的每日警報數示意圖，每日主機在網路上遭受異常攻擊的次數大約介於 100 至 8500 間，幅度相當的大。我們可以發現當主機可能成為被攻的目標時，當日的異常記錄會特別地多，例如在監控的第五日時，我們可以看到警報次數記錄異常的高，這是由於主機當日遭受大量的掃描行為所造成，而掃描的動作為一般入侵行為的前置作業，因此當遭遇到此現象時，必須特別留意主機的狀況。

表 4-5 為整個監控期間內的相關統計數據，由此表可知在此監控期間內總共產生了 22344 個網路型的警告，13 個主機型的警告與 4 個 False Positive 警告。由上述可察覺，網路型的警告經常多到超過我們所能夠處理能力之外，因此我們必須有效地減少網路型的警告數，才能不被它所提供的資訊給淹沒。

表 4- 5 監控交通大學內網路異常狀況記錄

攻擊所屬類別之名稱	各別攻擊類別之警告總數	主機型之異常封包警告	網路型之異常封包警告
unclassified	20314 (69%)	0(0)	20314
web-application-activity	535 (2%)	535(0)	0
misc-activity	1963 (7%)	71(0)	1892
protocol-command-decode	2361 (8%)	2361(1)	0
attempted-admin	2426 (8%)	2426(1)	0
attempted-recon	144 (0%)	12	132
misc-attack	71 (0%)	71	0
web-application-attack	115 (0%)	115	0
shellcode-detect	1438 (5%)	1438(2)	0
non-standard-protocol	3 (0%)	0	3
bad-unknown	3 (0%)	0	3
異常封包總和	29373	7029	22344
網路型警告	22344		
主機型警告	13		
False Positive 警告	4		

一般來說網路型的攻擊警報可透過設定攻擊行為的嚴重等級來過濾降低，例如在本實驗中，一般掃描型的行為約佔警報數的七成，而這些行為實際上並不屬於真正的攻擊行為，而僅僅是攻擊者用來收集情報的相關可疑活動，所以嚴重程度尚不需發出警報。除此之外，還可採用關聯警報的方式，把所有相關的警報融合為單一警報，也可有效降低警報數。

4.3.2 效率分析

由 4.3.1 小節之「長期網路監控實驗」的數據可知，本系統架構能減少因 False Positive 所造成的無效警報數，及減少 Monitor Agents 所需監控之封包數目以提升系統的效率，詳細的介紹如下所述：

- 減少因 False Positive 所造成的無效警報數

  根據 4.3.1 之「長期網路監控實驗」的數據可知，Snort、共同刺激機制、與本論文所提出架構在 25 天內所偵測出的警報數分別為 29373 個，13 個與 22361 個警報(圖 4-11)。相較之下，本論文的架構比 Snort 的警報數減少了 7012 個警報，足足降低了約 23.9%的警報數，且減少的警報數並非為系統所遺漏的警報，而是過濾了因 False Positive 所產生的無效警報。

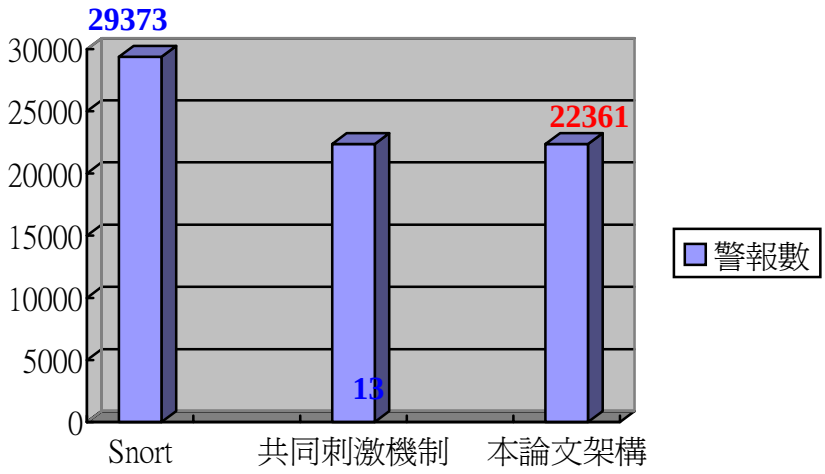

圖 4- 11　警報數比較圖

另外，原共同刺激機制的警報數僅有 13 個，比本論文所提出的架構還大幅減少。這是因為原共同刺激機制並沒有考量到偵測 Squealing 類型的攻擊，以及沒考慮到「並非所有網路型入侵偵測系統所偵測到的異常封包皆能讓 Monitor Agents 偵測出異常狀態」所造成，因此所減少警告的部分是屬於 False Positive 警報與 False Negative 遺漏警告的狀況。由以上的推論可知，本系統的設計確實可以降低 False Positive 而減少無效的警告數，並且降低原共同刺激機制所可能發生 False Negative 的問題。

- 減少 Monitor Agents 所需監控之封包數目

由 4.3.1 之「長期網路監控實驗」的數據可知，原共同刺激機制所需進行第二階段監控的異常封包數目為 29373 個，而本論文利用異常封包分類處理的機制後，所需監控的異常封包大幅度降至 7029 個，整體來說，大約減少了 76%之 Monitor Agents 所需監控封包，如下圖所示。

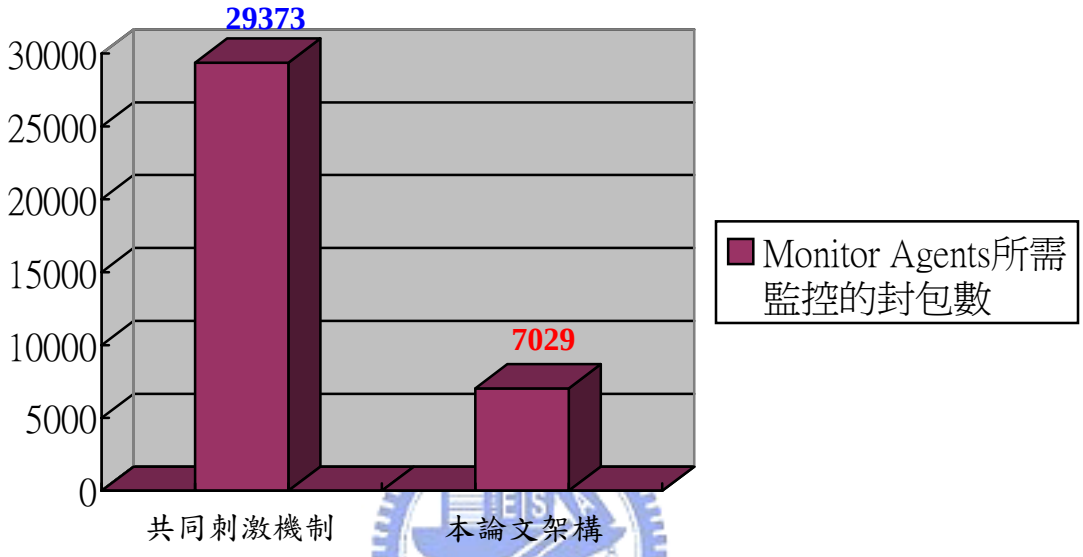

圖 4- 12 Monitor Agents 所需監控封包數目比較圖

4.4 討論

透過本章節的推論與實驗數據分析，我們可以得知本論文基於共同刺激機制所提出之偵測架構確實有成效與可行性，對於入侵偵測系統的安全性與偵測效能皆有正面的幫助。雖然兩階段的偵測架構必需額外在受保護的主機上安裝 Monitor Agents，耗損主機額外的系統資源，但如此的設計，卻可有效地增加入侵判斷的準確性與降低 False Positive 型誤判的機率。

因此，本論文所建構的共同刺激機制之入侵偵測系統架構，對於 False Positive 問題的解決確實是具有可行性與有效性。

第五章　結論及未來發展

5.1 結論

本論文主要貢獻為提出一個基於共同刺激機制的入侵偵測系統架構，內容除了兩階段式之「網路型」與「主機型」的入侵偵測外，還包含了異常封包分類處理、DoS 攔截機制、以及偵測 False Positive 攻擊等。由實驗結果顯示本架構確實可降低系統分析判斷時所造成的 False Positive，並可偵測出 Squealing 新類型的攻擊以增加網路的安全性。與 Yan Qiao 和 Xie Wei Xin 在［9]一文中所提出之架構相較下可減少所需進行第二階段監控的異常封包數目等優點。

由於主要的偵測效率還是基於原本系統的分析能力，所以異常封包分類模組的正確性，將對偵測結果產生重大影響。例如，把監控代理人所無法監控異常的網路型異常封包判斷為主機型的封包時，會因此而產生 False Negative，也就是會漏失掉此入侵攻擊警告。

5.2 未來發展

未來發展的方向可朝系統架構上的改良著手，包含下列幾個方向：

- **聯合防禦的機制**

雖然本文所提出的架構可以避免受保護之主機遭受到 DoS 的攻擊，但由於入侵偵測系統 Online 配置架構上的先天限制，一旦遭受 DDoS(分散式阻斷服務攻擊)類型的攻擊時，系統將會不正常運作。關於此問題的解決，可加入聯合防禦的機制以抵擋 DDoS 類型的攻擊。

- **有限狀態的分析方式**

由於 Monitor Agents 是直接監控主機上的狀況，因此當發現異常時，大多是入侵行為早已發生，接著我們只能做些急救的措施來補救。這和僅僅使用網路型的入侵偵測系統比較起來，網路型入侵偵測系統所做的大多是預測的行為，因為封包尚未送達攻擊目標，所以入侵行為還尚未產生。針對此問題未來可改良

Monitor Agents 的偵測判斷方式，以期可在真正的入侵行為發生之前，提出預告性的警告，例如，改採用有限狀態(Finite State)的分析方式等。

- **偵測新型攻擊的能力**

在攻擊方式不斷起快速翻新的環境之下，若可加入自動更新「入侵特徵資料庫」的機制或是「自動學習入侵規則」的能力，系統偵測入侵行為的能力也才可即時跟上攻擊者的腳步。

參考文獻

[1] D. E. Dening, “An Intrusion Detection Model,” IEEE Transactions On Software ngineering, vol. SE-13, no.2, pp.222-232, Feb. 1987.

[2] E.H．Spafford and D．Zamboni, “Intrusion Detection Using Autonomous Agent,” Computer Networks，vol．34，issues 4，pages 547-570，2000.

[3] J.P.Anderson , "Computer Security Threat Monitoring and Surveillance, " Tech. Rep. , James P Anderson Co. , Fort Washington, PA , Apr. 1980.

[4] Julia Allen，Alan Christie，and William Fithen et al.，“State of the Practice of Intrusion Detection Technologies,” Technical Report CMU/SEI-99-TR-028，CMU/SEI，January 2000.

[5] S. Axelsson,. “The Base-Rate Fallacy and the Difficulty of Intrusion Detection”, ACM Trans. On Info. And SystemSecurity, Aug. 2000, pp. 186-205.

[6] S. Patton, W. Yurcik, and D. Doss, ” An Achilles’ Heel in Signature-Based IDS: Squealing False Positives in SNORT”. Recent Advances in Intrusion Detection (RAID), Univ. of California-Davis, 2001.

[7] Steven A. Hofmeyr, “A Immunological Model of Distributed Detection and its Application to Computer Security” PhD ehesis, Department of Computer Sciences, University of New Mexico, Albuquerque, NM, April 1999

[8] W. Yurcik, “Controlling Intrusion Detection Systems by Generating False Positive：Squealing Proof-of-Concept”, Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, 2002.

[9] Yan Qiao , Xie Wei Xin , ”A Network IDS with Low False Positive Rate”, In Proceedings of the Congress on Evolutionary Computation, Honolulu, HI, May 2002. IEEE.

[10] Snort software, http://www.Snort.org.

[11] Koziol, Jack, “Intrusion detection with Snort”， Sams publishing, 2003

[12] Rehman, Rafeeq Ur. ,” Intrusion detection systems with Snort :advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID /Rafeeq Ur Rehman.”, Prentice Hall PTR, 2003.

[13] 李勁頤，陳亦明，”分散式入侵偵測系統研究現況介紹”，資訊安全通訊 2002，Vol.8

[14] 李駿偉，田筱榮，黃世昆，”入侵偵測分析方法評估與比較”，資訊安全通訊 2002，Vol.8.

[15] 陳培德，賴溪松，”入侵偵測系統簡介與實現”，資訊安全通訊 2002，Vol.8