中文摘要

隨著網際網路的快速開展，網路駭客入侵事件的發生也漸趨頻仍。網路型入侵偵測系統乃用以補防火牆之不足，並提供應用層級的網安保護。網路型入侵偵測系統並可以偵測不論是由網路內部之外部之資源亂用，或是由網路外部至內部之入侵等多種不同的資安問題。通常網路型入侵偵測系統使用字串比對以及靜態分析已進行偵測。但是字串比對已被證明是網路型入侵偵測系統的效能瓶頸所在。

吾人在這篇論文中，提出網路型入侵偵測系統中最重要的偵測引擎之設計與實作。首先針對 SYN Flood 攻擊，吾人提出 FSS 過濾器以成功阻攔攻擊並保護上層之 TCP 狀態引擎。接下來，吾人提出一個 TCP 的濾淨器。將駭客用以迷惑偵測引擎之不明封包移除，以確保字串比對器能夠見到正確無誤的內文。關於網路型入侵偵測系統的心臟，字串比對引擎，這篇文章一共提出三個方法。前兩個設計是以軟體爲基底，而第三個設計是以硬體爲基底。在不同的條件下，這三個比對引擎皆能有良好的處理速度，確保入侵偵測引擎在高準確度的前提下仍能有優異的效能。

Abstract

With growing Internet connectivity comes evolving opportunities for attackers to unlawfully access computers over the network. The Network Intrusion Detection Systems (NIDSes) are designed to identify attacks against networks or a host that are invisible to firewalls, thus providing an additional layer of security. The NIDS aims to detect a wide range of security violations ranging from attempted break-ins by outsiders to system penetrations and abuses by insiders. Generally two main methods are used for intrusion detection, namely Pattern Matching and Statistical Analysis. The former method applies a static set of patterns and alerts to traffic sequences with known signatures. Meanwhile, the latter method detects anomalous events statistically by gathering protocol header information and comparing this traffic to known attacks, as well as by sensing anomalies. Pattern matching tools are excellent at detecting known attacks, but perform poorly when facing a fresh assault or a modification of an old assault. NIDSes that use statistical analysis perform worse at sensing known problems, but much better at reporting unknown assaults. Improved implementation of an NIDS should combine these two methods to improve network protection. Either way, NIDSes rely on exact string matching from network packet payloads against thousands of intrusion signatures.

This dissertation first discusses an efficient and practical mechanism named FSS (First-Seen SYN) filter which can mitigate and block SYN Flood attacks. Then it presents a TCP processing engine which tracks the behaviors of each TCP connection including the state transition, sequence and acknowledgement number, and integrity checking. The most important of all, it eliminates the ambiguities when the attackers use ambiguities in network protocol specifications to deceive network security systems. Then we introduce several fast pattern-matching algorithms since it's the

most computation -intensive task in an NIDS and dominates the performance of an NIDS. Two software-based algorithms and one hardware-based architecture are proposed and proven to be more efficient and high-performance compared to other existing methodologies.