学位論文　博士（工学）

ネットワーク侵入検知・防御システムの実装法に関する研究

2011年度

慶應義塾大学大学院理工学研究科

花岡 美幸

論文要旨

ネットワークを介したリモート攻撃を検出・防止する手段の一つとして，ネットワーク侵入検知・防御システム（NIDS/NIPS: Network Intrusion Detection/Prevention Systems）が広く利用されている．リモート攻撃とは，脆弱性のあるサーバにインターネットを通じて攻撃メッセージを送り，被害を発生させる攻撃である．NIDS/NIPS はサーバに送信されてくるメッセージを検査することによって，攻撃を検知し管理者に警告を発することで，サーバを攻撃から防御する．

リモート攻撃が巧妙になりインターネット上のトラフィックが増大するなど，NIDS を取り巻く環境の変化により，次の 3 つの課題が出てきている．第一に，攻撃検知の精度向上が求められている．攻撃が高度化・巧妙化したことにより，従来の単純なシグネチャ・マッチングでは検知できない攻撃が増えているためである．第二に，NIDS の性能向上が求められている．インターネット・トラフィックの増大や検知手法の高度化に伴い，NIDS の負荷が増加しているためである．第三に，NIDS の障害発生時にも継続して攻撃検知が可能となる，耐障害性が求められている．汎用 PC を元にした構成が多い NIDS が，ハードウェア故障などにより NIDS に障害が起こった場合にも，継続して攻撃検知が行える必要がある．

本論文では，NIDS/NIPS の実装技術として検知精度・性能向上・耐障害性向上を解決する手法を提案する．まず，検知精度向上のためにレイヤ 7 コンテキストを考慮した攻撃検知を行えるようにする．近年，単純なバイトパターンのマッチングでなく，メッセージの順番やフォーマットなどのレイヤ 7 コンテキストを考慮することにより，検知精度を高める NIDS が提案されている．レイヤ 7 コンテキストを考慮した攻撃検知を行うためには，ネットワーク上を流れる個別のパケットをメッセージに再構成する，TCP ストリーム再構成機構が必要となる．レイヤ 7 NIDS/NIPS のための TCP ストリーム再構成機構は次の 4 つの要件を満たさなければならない．すなわち，1) 攻撃メッセージが攻撃対象アプリケーションに届くことがない完全な防御，2) 性能低下が少ないこと，3) NIDS/NIPS の設置に伴ってサーバやクライアントのアプリケーションに変更や再設定が必要ない，アプリケーション透過性，4) 監視する通信の振る舞いを乱さない，すなわち TCP フローや輻輳制御に与える影響が少ないトランスポート透過性，の 4 つである．本論文では，これらを全て満たす TCP ストリーム再構成機構として Store-Through 方式を提案する．Store-Through 方式では，個別のパケットからメッセージに再構築する際，順番が入れ替わったパケットの転送を止めずにコピーをとって転送することでトランス

ポート透過性を保持する．また，攻撃だと判断された時点で後から到着したパケットを破棄することで，攻撃の成功を妨げ完全な防御を達成する．また，IP レベルで実装することで，性能とアプリケーション透過性を達成する．プロトタイプを Linux 2.4.30 上に実装し，実験によって Store-Through 方式によるオーバヘッドは，パケットをそのまま IP 層で転送するだけの場合に比べて，3.8%以下であることを示した．また，実際のネットワークを用いた実験により，トランスポート透過性を保持できていることを示した．

次に，組織ネットワーク内に複数の場所に置かれた NIDS 同士を協調させることで，性能向上と耐障害性向上を行う手法を提案する．大学や企業など多くの組織では，組織内ネットワークとインターネットの境界のみでなく，内側のネットワークの様々な階層に複数の NIDS を設置している場合が多い．本論文で提案する NIDS 協調システム Brownie では，これらの組織ネットワーク内に置かれた NIDS 間でルール設定を交換し合い，定期的に負荷情報をやりとりすることで，NIDS 同士のルール設定を連携させる．そして，性能を向上するため，過負荷になった NIDS の負荷を減少させ，NIDS 間の冗長なルール設定を削除するようにルールを再設定する．また，耐障害性を向上するため，NIDS の障害時には障害が起こった NIDS で有効にしていたルールを，別の NIDS で有効にすることで攻撃検知を代替する．Brownie のプロトタイプを実装し，実験によって，ルールを再設定することで，web サーバベンチマークのスループットが 10%以上向上することを示した．また，NIDS を意図的に停止させ障害を起こした実験では，100 秒程度で別の NIDS でルールが有効になり攻撃が検知されるようになった．この時間は手動で障害回復を行う場合に比べて十分短いと言える．

Abstract

Network intrusion detection/prevention systems (NIDS/NIPS) are widely used for detecting or preventing network-based remote attacks. A remote attacker sends a malicious message to a vulnerable server and causes various damages on it. NIDS/NIPS monitor network traffic for malicious activities, and raise alerts or drop the packets when they detect attacks.

The environmental changes surrounding NIDS expose three issues of current NIDS implementation. First, NIDS needs more accurate detection mechanisms. Since attacks are becoming more complex and sophisticated, some attacks cannot be detected by simple byte-pattern matching. Second, performance improvement is necessary. Because of today's increased traffic volume and sophisticated attacks, NIDS needs enough performance to cope with hi-speed network and complex in-depth analysis to detect attacks. Third, NIDS needs a fault-tolerant mechanism. Although most of current NIDS are developed as software and run on commodity computers, NIDS should continue to detect attacks even under failures.

This dissertation proposes mechanisms to solve the above three issues: detection accuracy, performance, and fault-tolerance. First for detection accuracy, this dissertation proposes a mechanism for layer-7-aware detection with little performance overhead. Although simple string matching is traditionally adopted to detect malicious activities, exploiting layer 7 contexts has been recognized as an effective approach for improving the accuracy of detecting malicious messages in NIDS. Layer-7-aware NIDS requires a TCP stream reassembler which reassembles packets into a message without losing 1) complete prevention, which means the NIPS must be able to prevent target applications from receiving malicious messages, 2) performance efficiency, 3) application transparency, which means the NIDS installation does not require any modification or reconfiguration of the client or server applications, or 4) transport transparency, which means that the NIDS does not impair end-to-end TCP/IP semantics. This dissertation proposes the store-through mechanism which satisfies all the requirements. Store-through preserves transport transparency by forwarding each out-of-order packet immediately after copying the packet. Although the forwarded packet might turn out to be a part of an attack message, the store-through mechanism can successfully defend against the attack by blocking one of the subsequent packets that contain another part of

the attack message and thus provides complete prevention. In addition, IP-level implementation provides performance efficiency and application transparency. Testings of a prototype in Linux kernel 2.4.30 demonstrate that the overhead of store-through is less than 3.8% compared to simply IP forwarding the recieved packets. The experiments over the real Internet also suggest that store-through preserves transport transparency.

For performance and fault-tolerance, this dissertation proposes Brownie, a system which coordinates configurations of already-existing, independently-managed NIDSs in an organization. Our key observation is that most organizations, such as universities or companies, have several NIDSs managed by different administrators inside their internal networks, not only at the network entry point. With our proposed system Brownie, NIDSs exchange their own load status and rule configuration. Then Brownie achieves performance improvement by offloading overloaded NIDS and eliminating redundant rules. For fault-tolerance when a NIDS fails, Brownie enables rules once checked by the failed NIDS so that the other NIDS(s) takes over the failed NIDS. The experimental results with a web server benchmark suggest that Brownie increases the benchmark throughput by more than 10%. The experimental results also show that detections by a failed NIDS are took over by other NIDSs within 100 seconds. This is much faster than recovering manually by administrator.