サーバ アプリケーションを保護するアクセス制御メカニズム

Access Control Scheme for Protecting Server Applications

梅澤　健太郎　　高橋　俊成
■UMESAWA Kentaro　■TAKAHASHI Toshinari

ソフトウェアのバグに対するリモートからの攻撃が問題となっている。その対策運用上の問題は，対策パッチの適用を常時迅速に行うことが求められたり，パッチが存在しない未知のバグが存在したりすることにある。これらの問題は，需要が伸びつつあるリモートアクセスサービスにおいて特に重大な脅威であるが，それら攻撃に対する備えは十分といえず，潜在的なリスク要因となっている。

東芝は，これらの攻撃をTCP（Transmission Control Protocol）層でのコネクション確立制御により防御する技術であるTAP（TCP layer Application Protector）を開発した。TAPによりクラッカーやウイルスによるバグへの攻撃からサーバ アプリケーションを保護し，それらサービスの安全性を向上させることができる。

Fixing software vulnerabilities that are exploitable via a network is a matter of urgency for a system administrator. However, sometimes it is difficult to fix such vulnerabilities in a timely manner because there are many administrative problems in the systems operation area and some of the vulnerabilities do not have a program for fixing them at that time. This problem is especially serious in remote access services, which are currently experiencing high demand but have insufficient measures available.

To solve this problem, Toshiba has developed the transmission control protocol (TCP) layer application protector (TAP), which prevents attackers from establishing TCP connections by means of an authentication mechanism at the TCP layer.

1　まえがき

サーバ計算機に対する攻撃は，ソフトウェアのバグ（以下，脆弱（ぜいじゃく）性と記す）を利用して行われることが多い。このような脆弱性への攻撃は，ウイルスやワームなどによって自動化することで容易に実行できるため，迅速な対策が必要である。対策の基本は修正ソフトウェア（パッチ）を適用するという単純な作業であるため，迅速な対応が困難であるとの認識はあまりなく，その脅威をユーザーが軽視する傾向にある。しかしながら，パッチの適用は計算機の再起動を伴ったり，既存のアプリケーションの動作に影響を与えたりする場合も多い。そのため，確実なサービス継続性を必要とするサーバ計算機の管理業務においては，迅速なパッチの適用が難しい状況もある。

このような確実な継続性を要求されるサービスの例として，リモートアクセスやリモートメンテナンスがある。このようなサービスの典型的なものは，SSL/TLS（Secure Socket Layer/Transport Layer Security）を利用したVPN（Virtual Private Network）サービスとして提供される。SSL/TLSは通信データの暗号化などのセキュリティを提供するものであるが，SSL/TLS自体にも脆弱性が報告されている[(1)]ため，それへの攻撃の影響を別に考慮する必要がある。特に，SSL/TLSを利用したシステムの場合，安全性がそれに依存していることが多く，その影響は深刻である。

東芝は，通信コネクションの確立を要求するパケットに認証情報を格納し，それを利用してコネクション確立時にアクセス制御を行う技術であるTAP（TCP（Transmission Control Protocol）layer Application Protector）を提案している。TAPは通信コネクションそのものの確立を制御できることから，サーバ アプリケーションの脆弱性に対する攻撃を事前に防御することができる。その結果，リモートアクセスやリモートメンテナンス用のサーバ アプリケーションなどの安全性が高められ，副次的効果として，パッチの適用に猶予を持たせることができることから，保守コストを下げることも可能となる。ここではTAPを技術的に説明し，作成したモジュールの性能評価結果について述べる。

2　TAPの概要

2.1　原理

WWW（World Wide Web）で利用されるHTTP（HyperText Transfer Protocol）や，安全性を確保するために利用されるSSL/TLSなど，多くの通信アプリケーションはTCPという共通プロトコルで動作している。このTCPにおいては，クライアントがサーバに送信するSYN（SYNchronize）パケット，サーバがそれに応答してクライアントへ送信する

SYN/ACK（ACKnowledgement）パケット，それに答えてクライアントがサーバへ送信するACKパケット，から成る3 way handshakeによってコネクションが確立され，その確立したコネクションを利用してクライアントとサーバとでデータを送受信する。

注目すべき点は，サーバはクライアントからのSYNパケットを無条件に受け入れ，それに対して応答している点である。脆弱性を攻撃するためには，TCPコネクションを確立することが前提条件であるため，この段階でクライアントを選別することで攻撃が防御できる。

そこでTAPは，クライアントからサーバへのSYNパケットに格納した認証情報によりクライアント認証を行い，不正なクライアントへのSYN/ACKパケットの送信を行わないことでTCPコネクションの確立を制御し，サーバ アプリケーションの保護を行う。TAPを導入することにより，攻撃者は脆弱性を攻撃することはおろか，SYN/ACKパケットを受信できないことから，そこにサーバ アプリケーションが存在していることすらわからない。そのため，万一ウェブサーバやSSLなどの基幹アプリケーションに脆弱性があっても，攻撃を受けるリスクを最小限にできる。

2.2　接続の手順

SYNパケットによるクライアント認証は，電子署名やメッセージ認証子などの既存の暗号技術を利用して，認証情報をSYNパケット内の未使用フィールドに埋め込んだり，オプションとして付加したりすることで実現できる。しかし，このような変形パケットは，通信経路上にある高機能なルータやファイアウォールを通過できない危険がある。そこでTAPでは，確実にルータを通過するように構成した認証用SYNパケットを，本来のSYNパケットとは別に送信する（**図1**）。

この認証用SYNパケットには，クライアントとサーバの間であらかじめ共有した共有鍵によって生成された認証情報を埋め込むが，ルータなどではじかれる危険性をなくすために，通常のTCPのそれとまったく同じ形式にする必要がある。この制約下では，一つのSYNパケットに埋め込むことのできる情報量が限られる。そこで，複数の認証用SYNパケットに認証情報を分割して埋め込み，サーバ側で認証情報を再構築した後に検証処理を行う。

その具体的な手順は次のようになる。

ステップ1：SYNパケット送信　　TAPクライアント（TAPのクライアント機能が実装されたプログラム）はTAPサーバ（TAPのサーバ機能が実装されたプログラム）に，本来のSYNパケットとともに，複数の認証用SYNパケットに格納した認証情報を送信する。認証情報は，TAPサーバとTAPクライアントの間の共有鍵によって生成されたメッセージ認証子や，リプレイ攻撃防止のための時間情報（図1中のTimestamp）及び複数のSYNパケットから認証情報を再構成するための情報から成る。生成された認証情報は，通信経路上にあるルータやネットワーク制御技術との整合性を考慮し，TCPヘッダのシーケンス番号に埋め込んでいる。

ステップ2：SYNパケット検証　　TAPサーバはTAPクライアントから複数のSYNパケットを受信し，そこに格納された認証情報をTAPクライアントとあらかじめ共有した共有鍵を利用して検証処理を行う。ここで，複数のSYNパケットを送信してきたクライアントの同一性は，送信元IP（Internet Protocol）アドレスや他フィールドの値などを利用して判断する。

ステップ3：SYNパケット通過　　TAPサーバは，TAPクライアントから受信した複数のSYNパケットの認証情報が正しい場合には，受信した本来のSYNパケットを通過させる。その後，そのSYNパケットに対するSYN/ACKパケットの返信処理が行われる。このとき，クライアントの送信したSYNパケットに対する応答処理としてSYN/ACKパケットを返答する。そのため，返信されるSYN/ACKパケットは通信経路上にあるルータを問題なく通過できるうえ，クライアントも通常のSYN/ACKパケットとして受信でき，TCPコネクションを確立できる。

2.3　効果と安全性

攻撃者はTAPで保護されたサービスの存在に気がつかないため，そのサービスが攻撃にさらされる機会が減少する。また，自動化された攻撃スクリプト，ウイルスやワームなどから無差別に攻撃を受けた場合でも，コネクションの確立そのものが防止されているため，サーバ アプリケーションへの攻撃が成功することはない。また，仮にTAPの存在を知りえた攻撃者が手動で攻撃した場合でも，正当な認証情報を含む過去のSYNパケットの再送信（リプレイ攻撃）や認証情報

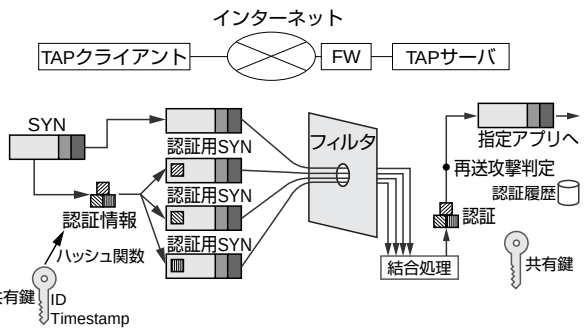

図1．TAP接続の概要 ― TAPクライアントから分割された認証情報を含むSYNパケットが届き，TAPサーバはそれを再構築し，検証する。
Overview of TAP access

の偽造に対して，TAPプロトコルは安全に構成されている。

リプレイ攻撃に対しては，TAPクライアントが送信時点の時刻情報を認証情報に含めて送信し，TAPサーバが検証時に現在時刻を確認してSYNパケットの時刻情報とのずれが許容時間内にあることを確認し防御する。仮に，攻撃者が通信路の盗聴などにより過去のSYNパケットを取得したうえで，その時刻情報を現在時刻に改ざんして送信してきた場合でも，認証情報を共有鍵で検証することにより検知できるため，問題とならない。認証情報の偽造は，その生成時に共有鍵を利用したハッシュ関数を適用することで防いでいる。

なお，TCP自体の既存の問題点として，通信経路上の攻撃者がセッションを乗っ取る可能性が考えられるが，TAPを導入してもこの攻撃を防ぐことはできない。しかしながら，この攻撃が成功するのはかなり限定された状況であり，現実に脅威となる可能性は低いと考えている。

2.4 関連技術

SYNパケットに対する応答制御は，IPアドレスやポート番号を指定するパケットフィルタリング技術として，ファイアウォールなどでも用いられている。しかし，提供するサービスによっては，クライアント端末のIPアドレスが動的に変化するなど，パケットフィルタリングには限界がある。

また，SYNパケットを利用した認証処理としては，Port knocking[(2)]などと呼ばれる運用技術が提案されている。このような技術は，認証情報を格納した複数のSYNパケットを受信した際に，通信コネクションを確立するためのSYNパケットを受け入れるようフィルタリングルールを変更するものである。それに対しTAPは，認証情報を格納したSYNパケットに対するSYN/ACKパケットの返答処理を制御するものであり，TCPコネクションの確立処理と認証処理を一体化させることで，攻撃者が攻撃をしかける余地を減らし，安全性を更に向上させている。

また，サーバアプリケーションにおける脆弱性の対策としては，攻撃用のコードが送信された段階で，侵入検知システムにより防御する方法もある。しかし，この方法では未知の脆弱性を完全には防御できない。このほかに，サーバ計算機に施す対策として，プログラム実行時に異常を検知し，プログラム実行を停止する方法も存在する。しかしながら，この場合はサーバアプリケーション自体の動作も停止させるものが多いため，正規のユーザーにも影響が生じるなど，サーバアプリケーションのサービス継続性に影響がある。

3 TAPの実用性評価

現在までにTAPサーバ，TAPクライアントともにモジュールが完成している（サーバはLinux[(注1)]版のみ，クライアントはLinux版及びWindows®[(注2)]版が存在）。ここでは，TAPの実用性を判断するためにLinux版モジュールで実験を行った。実験では，TAP通信の基本性能，インターネット環境での接続可能性，及び，TAPサーバの耐DoS（Denial of Service）攻撃性能の三つについて調査を行った。

3.1 基本性能

3.1.1 実験環境　サーバ計算機（S）とクライアント計算機（C：C1，C2，C3）を，スイッチングハブを介して接続した。Sの仕様は，OS：Linux2.4.18，CPU：Pentium®4[(注3)]2.4 GHz，メモリ：512 Mバイトであり，Cの仕様は，OS：Linux2.4.18，CPU：Celeron®[(注4)] 1.7 GHz，メモリ：256 Mバイトである。ハブとS及びCは，100Base-TXで全二重通信が可能である。

3.1.2 実験結果　ネットワークベンチマークツールであるNetperfを用いて，SとC1間のTAP通信のトランザクション数（毎回コネクションを確立し，C→S：32バイト，S→C：1,024バイトのデータ転送を行った結果）及びスループットを計測した結果，それぞれ，1,627回/s，94,128 Kビット/sというデータを得た。

同一条件での通常のTCPでは，1,634回/s及び94,129 Kビット/sであり，ほぼ同一の性能を示し，オーバヘッドは存在しないことが確認できた。

3.2 インターネット環境での接続可能性

TAPプロトコルを実利用する場合は，通信経路上に存在するネットワーク機器の構成にかかわらず接続できる必要がある。そこでTAPプロトコルの接続可能性を検証するために，インターネットの複数ISP（Internet Service Provider）を

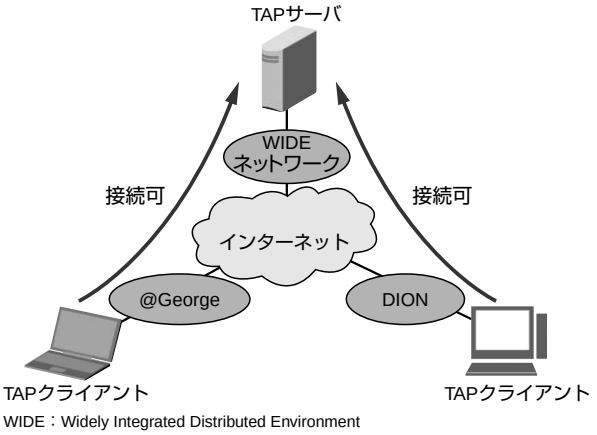

図2．接続実験の概要 ― プロバイダー（DION，@georgeなど）からWIDEに設置されたTAPサーバへの接続が問題なく成功している。
Feasibility test of TAP access via Internet

（注1）　Linuxは，Linus Torvalds氏の米国及びその他の国における登録商標。
（注2）　Windowsは，米国Microsoft Corporatipnの米国及びその他の国における登録商標。
（注3），（注4）　Pentium，Celeronは，米国又はその他の国における米国Intel Corporation又は子会社の登録商標又は商標。

経由した接続実験を行っている(**図2**)。これまでのところ，すべての実験環境においてTAP接続が成功しており，TAPの設計の正しさが確認できたと考えている。今後も様々な環境からの接続実験を継続し，接続可能性に関する情報を収集していく。

3.3 耐DoS攻撃性能

TAPを導入した場合には，攻撃者はサービスの存在そのものを知りえない。しかしここではあえて，TAPによって保護されたサーバアプリケーションに対する大量アクセス型のサービス不能(DoS)攻撃を想定した実験を行った。

この実験では，TAPによって保護されたウェブサーバの存在を知りえた攻撃者が，その可用性を損なうために大量アクセス型のDoS攻撃を行った状況を仮定した。この状況をシミュレートするために，SYNパケットを大量送信するSYN Flood攻撃によりTAPサーバを導入したサーバ計算機に負荷を与えた。そして，その状況でTAPを利用したウェブ閲覧の成功確率を測定した。ここで，SYN Flood攻撃で負荷を与えるのは，TAPはTCPのSYNパケットだけを処理対象とする機構であるため，送信コストの小さいSYNパケットの大量送信がもっとも強力な攻撃方法であることを考慮した。

3.3.1 実験環境　SにはTAPサーバ及びウェブサーバソフトウェアApacheを，C1にはTAPクライアントを，C2，C3にはSYN Flood攻撃ツールを，それぞれ導入した。そして，3.1.1項で解説したネットワークにおいて，C2，C3がSに向けてSYN Flood攻撃を実施しているなかで，C1はSにTAPを利用したウェブ閲覧を並列に1,000回試みて，その成功確率を測定した。実験では，TAPサーバのコネクション管理機構で利用するコネクションキューのリフレッシュ頻度も変化させ，その違いも測定した。

3.3.2 実験結果　実験の結果を**図3**に示す。ここで，横軸はC2とC3が送信する毎秒当たりのSYNパケットの合計数を，縦軸はTAPを利用したウェブ閲覧の成功確率を表す。

この結果から，リフレッシュ頻度が高いほど同一のパケット数でもウェブ閲覧の成功確率が高いことがわかる。このことは，SYN Flood攻撃に対するコネクションキューの動的管理機構の効率性を表す。そして，リフレッシュ頻度を現在の実装制約上の最高値に設定した場合には，40,000 ppsのSYN Flood攻撃下でも，TAPを利用したウェブ閲覧が100％成功する。この値は，TAP未動作時のSが100 pps程度のSYN Flood攻撃で接続不能状態に陥る(コネクションキューの制限値による)ことと比較しても，極めて安定した動作ができることを示している。

40,000 pps以降は成功確率が徐々に低下し，80,000 ppsのSYN Flood攻撃によって接続不能となる。これだけの頻度の攻撃になると，TAPサーバで利用するLinuxモジュール(libipq)やネットワークの性能限界が支配的となるため，TAPサーバの実装上の問題はまったくないと考えてよい。

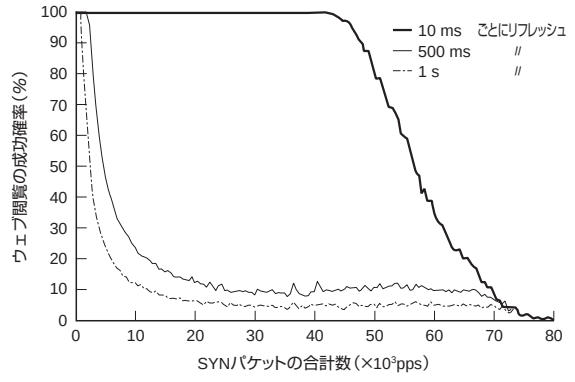

図3．TAP接続によるウェブ閲覧の成功確率 ― 大量アクセス攻撃を受けている状態でも，TAP接続が高い確率で成功している。
Availability of TAP access

4 あとがき

当社は，サーバアプリケーションを攻撃者から不可視とし，脆弱性に対する攻撃を防御するTAPを開発した。この技術は，TCPコネクションの確立制御を行うことにより正規のクライアントとだけコネクションを確立できるようにしたものである。現在までにモジュールの作成を完了し，その効果を実験室レベルでの評価により確認した。この技術により，更に安全にリモートアクセスやリモートメンテナンスなどのサービスの提供ができるようになる。今後は，それらのサービスの運用環境下でTAPサービスの試験運用を行い，実使用における有効性を確認していく。

文　献

(1) CERT Advisory. "CA-2003-26 Multiple Vulnerabilities in SSL/TLS Implementations". <http://www.cert.org/advisories/CA-2003-26.html>, (accessed 2003-12-16).
(2) Krzywinski, M. Port Knocking-Network Authentication Across Closed Ports. SysAdmin Magazine. **10**, 6, 2003, p.12－17.

梅澤　健太郎　UMESAWA Kentaro
研究開発センター コンピュータ・ネットワークラボラトリー。システムセキュリティの研究・開発に従事。情報処理学会会員。
Computer & Network Systems Lab.

高橋　俊成　TAKAHASHI Toshinari
研究開発センター コンピュータ・ネットワークラボラトリー研究主務。プログラミング言語処理系，UNIXセキュリティ及びその応用システムの研究・開発に従事。情報処理学会会員。
Computer & Network Systems Lab.