Vol.2013-OS-127 No.3
Vol.2013-EMB-31 No.3
2013/12/3

安全な家庭向けソーシャルルータの実現

櫻井 孝一[1]　海沼 直紀[1]　新城 靖[1]　佐藤 聡[1]　須藤 侑一[1]　肖 焜瑤[1]　中井 央[1]

概要：現在広く利用されている SNS (Social Networking Service) は中央サーバを利用するので、プライバシ侵害などの問題がある。この問題を解決し、SNS ユーザ間で中央サーバを用いず LAN 用アプリケーションを SNS アプリケーションとして利用するための仕組みとしてソーシャルルータが提案されている。従来のソーシャルルータには、専用のソフトウェアを情報端末へ導入する必要があること、および、全ての通信をお互いに受け入れるので安全性が低いという問題があった。本研究では、この問題を解決するために家庭向けソーシャルルータを提案する. 家庭向けソーシャルルータとは家庭用ルータを拡張したもので、異なる家庭内 LAN に所属する SNS ユーザの情報端末間の相互通信を実現する. 家庭向けソーシャルルータでは DNS を利用し、情報端末を SNS のユーザ名を含むホスト名で指定できる. また、このルータはパケットフィルタを用いて、SNS ユーザ間の全ての通信に強固なアクセス制御を提供する. 本研究では Facebook を対象に家庭向けソーシャルルータを実装する.

1. はじめに

今日、数十億もの人々が Social Networking Service(以下 SNS と呼ぶ) を利用している. 多くの利用者がスマートフォンやタブレットなどの携帯情報端末から SNS を利用している. SNS に対応しているアプリケーションにソーシャルアプリケーションがある. ソーシャルアプリケーションには SNS ユーザ同士が協調作業が行えるアプリケーションや SNS ユーザ間でファイルを共有できるサービスが存在する. これらのソーシャルアプリケーションの多くは中央サーバを利用する. そのため、スケーラビリティが低いという問題、サービス終了に伴う保存データの喪失、プライバシの侵害などが起こる可能性がある [1].

この問題を解決するために SNS ユーザ間で中央サーバを介さずに LAN 用アプリケーションを利用したいという要望がある. この要望を満たすための仕組みとしてソーシャルルータが提案されている [2]. ソーシャルルータは SNS ユーザの情報端末から送り出されるパケットを他の SNS ユーザの情報端末へ到達させ、SNS ユーザ間での LAN 用アプリケーションの利用を容易にする. 例えば、NFS(Network File System) を用いたファイルの共有や VNC(Virtual Network Computing) を用いた画面の共有を SNS ユーザ間で容易に利用可能にする.

ソーシャルルータの一実装として Social VPN[2] がある. この Social VPN は SNS ユーザの情報端末 (PC) を P2P 型の VPN で接続する. SNS ユーザは構築されたネットワークで LAN 用アプリケーションを利用できる. この Social VPN の問題点は第 1 に専用のソフトウェアを情報端末 (PC) へ導入する必要があることである. そのため利用できる情報端末は特定のソフトウェアを導入可能な PC に制限される. 従ってスマートフォンやタブレット端末では利用することができない. 第 2 の問題点は Social VPN で構築されたネットワークに参加する SNS ユーザはお互いに全面的に信頼していることが前提となっていることである. これにより各ユーザの情報端末は全ての通信をお互いに受け入れる. もし信頼しているユーザの情報端末がワームに感染していると、他のユーザの情報端末も危険に晒される.

本研究の目的は情報端末の種類に依存せず SNS ユーザの情報端末間で LAN 用アプリケーションが利用できる安全なソーシャルルータを実装することである. そのために一般家庭でインターネットへアクセスするために利用される家庭向けルータに対してソーシャルルータの機能を持たせる.

2. 家庭向けソーシャルルータの要件

本研究で提案する家庭向けソーシャルルータは以下のソーシャルルータとしての要件 (1) から (5)、および、家庭向けルータとしての要件 (6) から (7) の全ての要件を満たすルータである.

(1) ユーザは利用開始時に SNS アカウントで認証する.
(2) ユーザは他の SNS ユーザが利用している情報端末の情報を簡単に取得できる.
(3) SNS ユーザの情報端末が送り出すパケットを他の SNS

[1] 筑波大学

ユーザの情報端末に到達させる.
(4) 情報端末のホスト名は SNS ユーザ名を含むもので指定可能にする.
(5) 必要に応じてルータ間の通信を暗号化する.
(6) 強固なアクセス制御機能がある.
(7) IPv4 NAT(Network Address Translation) が利用可能である.

家庭向けソーシャルルータの要件を満たすためには、ソーシャルルータとしての機能を家庭用ルータにおいて実装すればよい. 重要な要件については以下の節で詳しく述べる.

2.1 SNS ユーザ名を利用した接続先の指定

家庭向けソーシャルルータでは他の SNS ユーザの情報端末の指定には IP アドレスだけではなくホスト名を利用可能にする. 家庭向けルータが持つ DHCP サーバにより情報端末へ配布される IP アドレスは、必ずしも毎回同じとは限らない. 本研究ではホスト名により情報端末を指定可能にすることで IP アドレスが変更されても同じ名前でアクセス可能にする.

家庭向けソーシャルルータで設定する情報端末のホスト名は SNS ユーザ名を含めるようにする. 例えば、SNS ユーザ名が alice なら alice.socialrouter が設定される. また、1 つのアカウントで複数の情報端末を利用する場合には'*devicename.username*.socialrouter' となる. 例えば、alice が pc1 という情報端末を家庭向けソーシャルルータで利用する場合には'pc1.alice.socialrouter' というホスト名が設定される.

2.2 通信の暗号化

家庭向けソーシャルルータではルータ間の通信を必要に応じて暗号化する. これにより通信の暗号化に対応していないアプリケーションは、ソーシャルルータにおいて通信の暗号化がなされるため安全な通信を保てる. 逆に、HTTPS のように、アプリケーション側で暗号化がなされる場合は暗号化を行わないオプションを提供することが望ましい.

2.3 強固なアクセス制御

家庭向けソーシャルルータはアクセス制御を行い、意図するの通信だけを許可する機能を持つ. アクセス制御のためにルータへ設定する必要がある項目としては、友達の情報端末の IP アドレス、自身の情報端末の IP アドレス、使用するポート番号がある.

従来のソーシャル機能がない家庭向けルータでは、友達の情報端末から自身の情報端末へアクセスを許可するには危険を伴う設定を行う必要があった. 従来の家庭向けルータはポートを開放する機能を持っている. この場合、友達の情報端末だけでなく全ての外部のホストからのアクセスを常に許すことになる. これに対して家庭向けソーシャルルータでは、特定の友達の情報端末が、特定の自分の情報端末へ特定のポート番号を使ったアクセスすることだけを許し、その他のアクセスを禁止する.

家庭向けソーシャルルータは、内向け (ingress) アクセス制御だけでなく、外向け (egress) のアクセス制御も行う. 内部のネットワーク内にワームに感染した情報端末がいると SNS ユーザの友達の情報端末へ感染する危険性がある. ソーシャルルータは外向けアクセス制御によりこれを防止する.

アクセスの許可は LAN の SNS ユーザがソーシャルルータを利用している時のみ行われる. SNS ユーザが情報端末の利用を停止するとソーシャルルータはアクセスを遮断する.

2.4 IPv4 NAT への対応

従来の家庭向けルータの NAT 機能を利用する場合に外部からのアクセスに制限が生じる. 家庭向けルータの NAT ではインターネット側の 1 つの IP アドレスを内部の情報端末で共有することが一般的である. そのため、従来の家庭向けルータでは内部に存在する複数の端末で同時にインターネット側の同じポート番号で待ち受けることができない.

家庭向けソーシャルルータではこの問題を解決する. 本ソーシャルルータでは、内部の複数の情報端末を外部からアクセスする時も IP アドレスで区別可能にする. 例えば、複数の情報端末においてポート番号 80 番で Web サーバが動作していたとしても、外部の SNS の利用者の情報端末からは別の IP アドレスでアクセス可能にする.

3. 家庭向けソーシャルルータの概要

本研究では、SNS として Facebook を対象としている家庭向けソーシャルルータ（以下ソーシャルルータと呼ぶ）を実装する. 本ソーシャルルータを SNS ユーザが利用するには SNS ユーザが所有する Facebook アカウントでユーザ認証を行う.

3.1 ソーシャルルータへのログイン

SNS ユーザはソーシャルルータの利用を開始する前に Web ブラウザでソーシャルルータが提供する Web ページへログインする. SNS ユーザはログインページから利用する SNS を選択し、SNS が提供する認証ページにユーザ名とパスワードを入力しユーザ認証を行う. ユーザ認証が完了するとソーシャルルータのトップページ (図 1) が表示される. トップページの左側は設定ページへのリンクがあり、右側に SNS ユーザ自身が現在ソーシャルルータで利用している情報端末のリストが表示される. 左側のリストから移動できる設定ページを以下に示す.

- 友達が利用中の端末を表示するページ
- 自身の情報端末を登録するページ

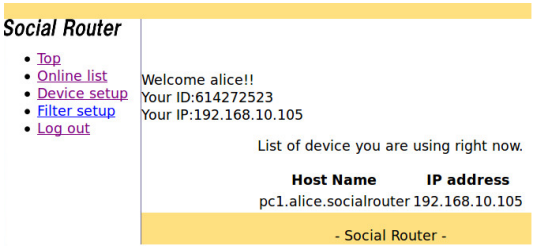

図 1 SNS ユーザのログイン後に表示されるトップページ

```
bob@homepc:~$mount -t nfs pc1.alice.social\
  router:/home/alice/share /mnt/alice-nfs/
```

図 2 alice の NFS サーバが公開している
ディレクトリへマウントするコマンドの例

```
/home/alice/share *.bob.socialrouter(rw,\
  sync,no_root_squash,no_subtree_check)
```

図 3 bob.socialrouter のドメイン名を持つ
ホストのみマウントを許可するルール

- アクセス制御を設定するページ

3.2 ソーシャルルータを利用したアプリケーションの例

本研究の家庭向けソーシャルルータは次の LAN 用アプリケーションを実行することができる.

- NFS を利用したファイル共有
  SNS ユーザは NFS を利用して情報端末間で容量制限や帯域制限のないファイルの共有が可能になる. SNS ユーザはソーシャルルータの要件 (4) よりドメイン名を利用して、NFS サーバを指定することができる. クライアント側では友達が NFS で公開しているディレクトリをマウントするために SNS ユーザ名を含むホスト名を利用できる. この例を図 2 に示す. この図は alice が持つ情報端末の pc1 にあるディレクトリ/home/alice/share をこの情報端末のディレクトリ/mnt/alice-nfs にマウントしている. このようにホスト名として pc1.alice.socialrouter という名前が利用できる. NFS サーバにはアクセス制御の機能がある. NFS サーバは NFS クライアントに対してアクセス可能なディレクトリとオプション (読み書き許可等) を設定できる. たとえば Linux の NFS サーバは/etc/exports ファイルに図 3 のような設定を記述できる. このように記述すると bob が持つ情報端末に/home/alice/share ディレクトリのマウントを許可する.
- IRC を利用したメンバ限定のチャット
  SNS ユーザはソーシャルルータを利用して限られたメンバのみアクセスできる IRC(Internet Relay Chat) サーバを設置することができる. メンバは通常の IRC クライアントから IRC サーバへ接続することで他のメンバとチャットを行う. IRC クライアントとしては、パソコン用のものだけではなくスマートフォンやタブレット用のものも利用可能である. 図 4 は iPhone の IRC クライアントの Mango Lite を利用している.

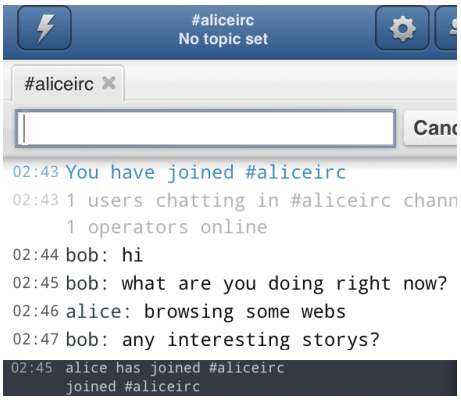

図 4 iPhone の Mango Lite から IRC を利用した図

- VNC を利用したリモートアクセス
  リモートで SNS ユーザの友達の手助けを行う場合, 家庭向けソーシャルルータは相手のアクセスを簡単に受け入れるようにできる. クライアントには PC やスマートフォンのアプリケーションを利用する.

本ソーシャルルータは本研究室の須藤の研究 [3] と肖の研究 [4] の通信路として使われている. 須藤の研究ではソーシャルルータを用いることで SNS ユーザ同士で相互に SQL サーバを利用可能にする. 肖の研究では SNS ユーザ同士がそれぞれ持つネットニュースシステム間で記事の転送を行う.

3.3 ソーシャルルータの設定方法

SNS ユーザは、ソーシャルルータが提供するウェブページから自身の情報端末の登録、自身の情報端末のアクセス許可、および、利用するアプリケーションの設定を行うことができる.

3.3.1 SNS ユーザの友達が利用中の端末を表示するページ

このページは 2 章で述べたソーシャルルータの要件 (2) を満たすためのページである. SNS ユーザは友達が現在利用している端末の情報を図 5 に示したページから確認できる. SNS ユーザはこのページで通信したい友達の情報端末のホスト名を探す. SNS ユーザは図 5 のページの一覧から友達を選択して submit を押す. するとページ上に選択した友達が現在利用している端末のホスト名と IP アドレスが表示される.

3.3.2 自身の情報端末のホスト名を登録するページ

2 章に述べた要件 (4) を満たすために、ソーシャルルータは自身の情報端末のホスト名を登録するページを提供している. SNS ユーザは自身が利用する端末を図 6 の情報端末登録ページからソーシャルルータへ登録する. SNS ユーザはソーシャルルータを利用する情報端末のホスト名と IP アドレスをテキストボックスへ入力する. IP アドレスのテキストボックスには Web ブラウザの IP アドレスが初期値と

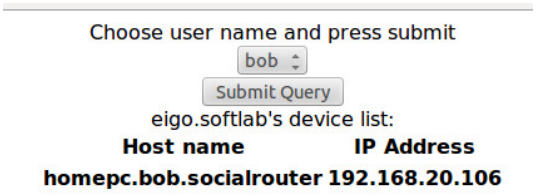

図 5　友達が利用中の端末確認ページ

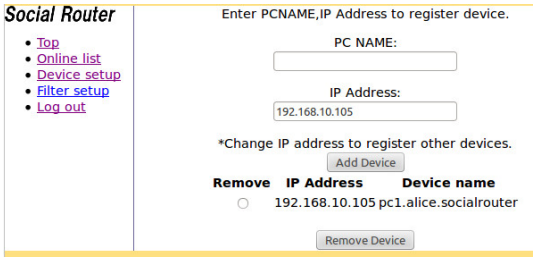

図 6　ソーシャルルータを利用する情報端末の登録ページ

して埋め込まれている. ウェブブラウザを持たない情報端末でソーシャルルータを利用する場合は、テキストボックスの IP アドレスを修正する. ソーシャルルータはこのページから入力を受け付けると情報端末のホスト名と IP アドレスを登録する. 登録が完了すると友達は登録したホスト名で名前解決できるようになる.

3.3.3　アクセス制御設定ページ

2 章で述べた要件 (2) を満たすために本ソーシャルルータはアクセス制御の設定ページを提供する. アクセス制御設定ページは SNS ユーザが利用したい情報端末と友達の情報端末、および、アプリケーションが利用する通信ポートを用いてアクセス許可の設定を行う. アクセス制御設定ページはクライアントとサーバの 2 種類ある. クライアント用アクセス制御設定ページは外向け (egress) アクセス制御、すなわち、自身の情報端末の LAN 外への通信を制御する. サーバ用アクセス制御設定ページは内向け (ingress) アクセス制御、すなわち、外部から自身の情報端末への通信を制御する. クライアント用アクセス制御設定ページを図 7 に示す. サーバ用アクセス制御設定ページを図 8 に示す. 設定ページの操作はクライアント用ページもサーバページも同じである. まず、SNS ユーザは設定ページから自身が利用する情報端末をプルダウンメニューから選択する. 次に利用するアプリケーションをプルダウンメニューから選択する. 最後にクライアント設定ページであれば、アクセスしたい友達の情報端末を選択する. 逆にサーバ用設定ページであれば、アクセスを受け入れる友達の情報端末を選択する. 設定が完了したら submit を押す.

4.　Facebook を対象とした家庭向けソーシャルルータのシステムの実装

本研究では SNS として Facebook を対象としてソーシャルルータを実装する. 実装した家庭向けソーシャルルータはルータ本体、外部サーバ、SNS の認証サービスの 3 つで構成される。図 9 に構成の全体図を示す. ルータ本体ではパケットフィルタ iptables、DNS サーバ bind9、Web サーバ Apache、Ruby により実装した CGI、および、dRuby のサーバプログラムが動作する. ソーシャルルータには友達情報キャッシュと設定データベースをもつ. **友達情報キャッシュ**は現在オンラインの友達の情報が格納されるキャッシュであり、**設定データベース**はソーシャルルータへ適応される DNS とパケットフィルタの設定情報が格納されるデータベースである. 外部サーバではデータベースに MySQL、Web サーバ Apache、および、dRuby のサーバが動作する. SNS 認証サービスは Facebook が提供する OAuth を利用する.

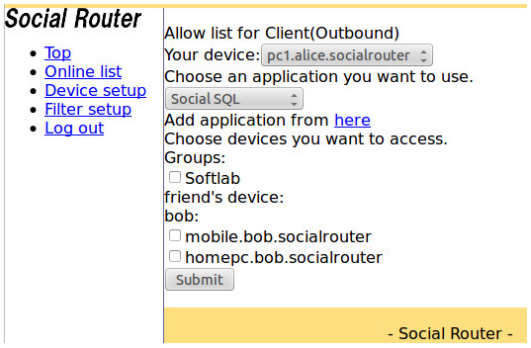

図 7　クライアント用アクセス制御設定ページ

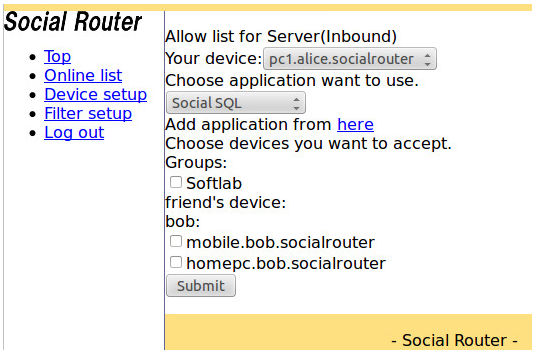

図 8　サーバ用アクセス制御設定ページ

4.1　外部サーバ

家庭向けソーシャルルータを実装するためには以下の情報が必要となる.

- SNS ユーザのプレゼンス情報
- SNS ユーザが利用している情報端末の IP アドレス
- インターネットから到達可能なソーシャルルータの IP アドレス

これらの必要な情報は Facebook では提供していない. 従って家庭向けソーシャルルータの実装では外部サーバを置き、これらの情報を把握する. 外部サーバとは、インターネット上にありソーシャルルータ、および SNS ユーザの

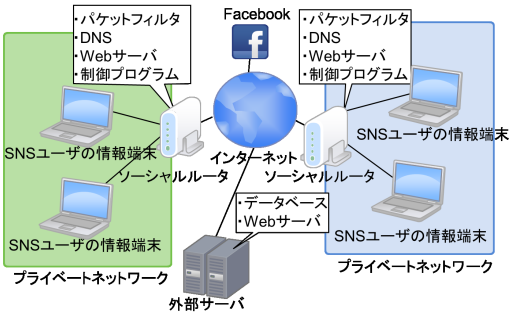

図 9 家庭向けソーシャルルータの全体図

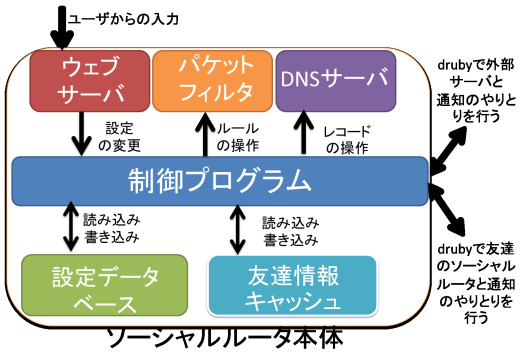

図 10 ソーシャルルータ本体の構成

Web ブラウザから到達可能とする.

外部サーバにはデータベースを用意し、表 1 に示した情報を格納する. ソーシャルルータは友達が利用している情報端末の IP アドレスと友達が利用しているソーシャルルータの IP アドレスを外部サーバから取得する. ソーシャルルータは外部サーバから情報端末の名前と IP アドレスを入手すると自身の DNS に登録し、SNS ユーザ名でその情報端末を指定出来るようにする. ソーシャルルータの IP アドレスはソーシャルルータに情報端末が追加されたこと、削除されたことを通知する際に利用する.

1 章で述べた Social VPN[2] では XMPP(eXtensible Messaging and Presence Protocol) を使用して接続相手を把握する. XMPP はチャット用プロトコルであり、取得できる情報はコンタクトリスト内のユーザに限られる. 本ソーシャルルータの外部サーバは Facebook のアプリケーションとして動作し、SNS ユーザのコンタクトリストの加えて、グループの情報まで取得することが可能である.

表 1 外部サーバ上のデータベースで格納する情報

属性名	説明
uid	SNS 内のユーザ ID
uname	SNS 内のユーザ名
pcname	情報端末のホスト名
ip	情報端末の IP アドレス
routerip	インターネットから到達可能な家庭向けソーシャルルータの IP アドレス

4.2 制御プログラム

図 10 に本実装向けソーシャルルータ本体の構成を示す. 本家庭向けソーシャルルータは既存の DNS サーバとパケットフィルタを利用している. これら既存のプログラムを制御するプログラムを制御プログラムと呼ぶ.

ソーシャルルータは自身に適用される設定のデータベースを持っている。このデータベースを設定データベースと呼ぶ. 設定データベースは制御プログラムがソーシャルルータに設定する情報端末の DNS の設定と情報端末のパケットフィルタの設定を保持している.

ソーシャルルータはオンライン中の友達の情報端末の情報をキャッシュとして持っている. このキャッシュのことを友達情報キャッシュと呼ぶ. 友達情報キャッシュは外部サーバと友達のソーシャルルータから受け取った情報をキャッシュとして保持する.

制御プログラムは外部サーバや他のソーシャルルータからの SNS ユーザの情報の取得や通知に dRuby を利用している.

表 2 に制御プログラムが利用しているイベントの一覧を示す. 以下にそれらのイベントが発生した時の処理を示す.

4.2.1 ユーザの情報端末のログイン

local_login イベントが発生すると、制御プログラムは設定データベースに保存されているその情報端末に関するパケットフィルタとホスト名の設定を DNS サーバ、および、ソーシャルルータのパケットフィルタに登録する. 制御プログラムは外部サーバから SNS ユーザの友達情報を取得し、ソーシャルルータが持つ友達情報キャッシュに追加する. 次にローカルユーザが利用開始した情報端末の情報を外部サーバと友達のソーシャルルータへ通知する.

4.2.2 ユーザの情報端末のログアウト

local_logout イベントが発生すると、制御プログラムはまず、DNS サーバと iptables からその情報端末の情報を消去する. 次にその情報端末がログアウトしたことを外部サーバへ通知する. 制御プログラムはログアウトする SNS ユーザに関連した友達情報を友達情報キャッシュから破棄する. 最後に制御プログラムは友達のソーシャルルータへその情報端末がログアウトしたことを通知する.

4.2.3 ソーシャルルータの設定変更

update イベントが発生すると、制御プログラムは設定データベースを更新する. 制御プログラムは、設定データベースが変更されると、変更された内容をソーシャルルータのパケットフィルタと DNS サーバに適用する。情報端末のホスト名が変更された場合、制御プログラムは外部サーバにそのことを通知する. 最後に友達のソーシャルルータへそのことを通知する.

4.2.4 キャッシュのリフレッシュ

制御プログラムは友達の情報端末の情報を友達情報キャッシュに持っている. ソーシャルルータは refresh イベントが発生すると、友達情報キャッシュのリストに載っている友達の情報端末とソーシャルルータの状態を確認して友達情報キャッシュを更新する.

4.2.5 友達の情報端末のログイン

friend_login イベントが発生すると、制御プログラムはまず、ソーシャルルータが持つ友達情報キャッシュへログインした友達の情報を追加する. 次に、制御プログラムは、通知で送られてきた友達の情報端末の IP アドレスとホスト名をソーシャルルータの DNS サーバへ登録する. もし、ソーシャルルータの設定データベースに友達の情報端末のパケットフィルタの設定が保存されていればそれを適用する.

4.2.6 友達の情報端末のログアウト

friend_logout イベントが発生すると、制御プログラムは通知で受け取ったログアウトする友達の情報端末に関連する DNS のレコードとパケットフィルタのルールをローカルルータから消去する. ソーシャルルータは自身が持つ友達情報キャッシュからログアウトした友達に関する情報を消去する.

表 2 制御プログラムのイベント一覧

イベント名	発生条件
local_login	SNS ユーザがソーシャルルータの Web アプリケーションにある情報端末でログインする.
local_logout	SNS ユーザがソーシャルルータの Web アプリケーションから情報端末でログアウトする.
update	SNS ユーザがソーシャルルータの Web アプリケーションから情報端末の登録情報を変更する、または、アクセス制御の設定を変更する.
refresh	定期的に発生する.
friend_login	友達のソーシャルルータから友達がある情報端末でログインした通知を受け取る.
friend_logout	友達のソーシャルルータから友達がある情報端末でログアウトした通知を受け取る.

4.3 iptables へフィルタリングルールの投入

制御プログラムは local_login イベントや update イベントが発生すると、iptables のルールを設定データベースの情報から生成し、それを system() 関数を用いて iptables コマンドを実行することでルールを追加する. 例えばローカルユーザ Alice の情報端末 (IP アドレス:198.168.10.10) が友達の Bob の Web サーバ (IP アドレス:192.168.20.15 待受ポート番号 tcp80) にアクセスすることを許可する場合、ローカルのソーシャルルータ内の制御プログラムは図 11 のルール生成し system() 関数で iptables コマンドを実行する.

```
iptables -A INPUT -s 198.168.10.10 \
       -d 192.168.20.15 -p tcp -dport 80
```

図 11 制御プログラムが iptables で実行するコマンドの例

```
zone ='socialrouter'
dnsserver ='127.0.0.1'
ns=DDNS.new(dnsserver,zone)
ns.addnamev4(username+"."+zone,deviceip)
```

図 12 実行する DNS レコードの追加コード

4.4 DNS レコードの追加と消去

local_login イベントや update イベントが発生すると、制御プログラムは DNS サーバの DNS レコードを破棄し、新たに設定データベースの情報から DNS サーバにホスト名を登録していく. この処理を行う制御プログラム内の主要部分を図 12 に示す. 変数の username が SNS ユーザ名、変数の deviceip が情報端末の IP アドレスになる. このように Ruby 言語の DDNS クラスを用いて DNS サーバに対してレコードの追加と消去を行っている.

4.5 外部サーバにおけるソーシャルルータの認証

ソーシャルルータが外部サーバへアクセス時にはクライアント認証が必要となる. 外部サーバは認証のために Web サーバを持つ. SNS ユーザはソーシャルルータの利用開始時に、ソーシャルルータと外部サーバの Web サーバで Facebook のアカウントを用いたユーザ認証を行う. 外部サーバは SNS ユーザのユーザ認証完了後、そのソーシャルルータからのアクセスに応じる.

4.6 ソーシャルルータ間のトンネル

本ソーシャルルータはソーシャルルータ間に VPN を利用してトンネリングを行う. 本研究では VPN トンネルとして、IPsec トンネルモード Preshared-key、Freelan[5]、Softeather VPN[6]、および、IP in IP[7] を利用可能にする. ソーシャルルータは必要ならソーシャルルータ間でいずれかの VPN トンネルを構築し、暗号通信を実現する. ソーシャルルータ間で暗号通信が不要な場合は IP in IP[7] トンネルを利用する.

4.7 NAT への対応

本ソーシャルルータは 2.4 節で述べたように NAT に対応する. トネリングで繋がれたネットワーク同士が同じプライベートアドレスの範囲を利用することがある. 本ソーシャルルータではこのように IP アドレスの衝突が起きた場合のは Twice NAT[8] で対応する.

5. 関連研究

Social VPN[2] は SNS を利用しているユーザ間で IP over P2P を利用した VPN を構築し、SNS ユーザ同士の相互通

信を実現する研究である. これにより、SNS ユーザ間で VPN を構築し LAN 用アプリケーションの利用を可能にする. Social VPN のユーザ認証には XMPP に対応した SNS のアカウントを利用する. ユーザ間で VPN を構築するには各ユーザがお互いの XMPP のコンタクトリストに登録されている必要がある. Social VPN ではユーザ同士が相互にコンタクトリストに登録していることでユーザ同士が信頼していると判断しユーザ間でどのような通信も行える.

本研究では SNS ユーザの情報端末間の通信を情報端末単位、および、ポート単位で制限できる点が異なる. さらに、本研究では、スマートフォンやタブレットなどの任意の情報端末を利用できる点が異なる.

Virtual Private Social Networks[9](以下 VPSN と呼ぶ) は既存の SNS を利用し、セキュアな SNS を作成する. VPSN を利用することでセキュアなグループに所属している人にだけユーザの本物のプロフィールの項目やページを表示し、その他の人には異なる情報を閲覧させることが可能になる. ユーザの本物のプロフィールやページはユーザが XML ファイルとして持っており、そのファイルをグループに所属している人にメールで送る. そのメールを受け取った人はユーザの本物のプロフィールやページを閲覧できる.

本研究では外部の SNS ユーザの情報端末が家庭向けソーシャルルータ内の SNS ユーザの情報端末へのアクセスを制御している. VPSN は Web コンテンツへのアクセスを制限しており、家庭向けソーシャルルータとはアクセスを制御する対象が異なる.

6. まとめ

本論文では安全な家庭向けのソーシャルルータについて述べた. 従来のソーシャルルータでは利用可能な情報端末が限られるという問題とアクセス制御機能がないため全てのパケットを情報端末が受け入れてしまうという問題がある. 本家庭向けソーシャルルータではこれらの問題を解決する.

本研究では Facebook を対象に家庭向けソーシャルルータを実装する. 全体の構成はルータ本体、外部サーバ、SNS が提供する認証サーバからなる. 外部サーバは SNS ユーザのプレゼンス情報、SNS ユーザが利用している情報端末の情報、SNS ユーザが利用しているソーシャルルータの情報を把握する. ルータ本体は情報端末のホスト名 (IP アドレス)、ポート番号、および方向 (ingress か egress か) を利用してアクセス制御を行う. 家庭向けソーシャルルータには設定用ウェブページがあり、SNS ユーザはアクセス制御や情報端末の追加をそのページから行う. ソーシャルルータでは必要に応じてルータ間に VPN トンネルを構築し、暗号通信を実現している.

今後の課題としては 2 つある. 1 つ目はソーシャルルータを Facebook 以外の SNS に対応させることである. 2 つ目は異なる SNS を利用しているユーザの情報端末間で通信が出来るようにすることである.

参考文献

[1] A. Datta, S. Buchegger, L.-H. Vu, T. Strufe, and K. Rzadca. Decentralized Online Social Networks. *Springer Science+Business Media pp.349-378*, 2010.
[2] R. Figueiredo, P. Boykin, P. Juste, and D. Wolinsky. Social VPNs: Integrating Overlay and Social Networks for Seamless P2P Networking. *IEEE WETICE/COPS*, 2008.
[3] 須藤侑一, 新城靖, 櫻井孝一, 佐藤聡, 肖焜瑤, 中井央. ソーシャルネットワークを利用した SQL データベースの相互利用. 情報処理学会システムソフトウェアとオペレーティングシステム研 究会 2013-OS-127, 2013.
[4] 肖焜瑤, 新城靖, 櫻井孝一, 佐藤聡, 須藤侑一, 中井央. ソーシャルルータを用いたネットニュースシステムの実現. 情報処理学会システムソフトウェアとオペレーティングシステム研 究会 2013-OS-127, 2013.
[5] freelan - an open-source, multi-platform, peer-to-peer VPN software. http://www.freelan.org/. Accessed: 2013-10-27.
[6] Softether VPN project. http://www.softether.org/. Accessed: 2013-10-27.
[7] C. Perkins. IP Encapsulation within IP. *RFC 2003 (Standards Track), October. 1996.*
[8] P. Srisuresh and M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. *RFC 2663 (Informational), Aug. 1999.*
[9] M. Conti, A. Hasani, and B. Crispo. Virtual Private Social Networks. *The first ACM Conference on Data and Application Privacy pp.39-50*, 2011.