第 XI 部

IP パケットの暗号化と認証

第 11 部
IP パケットの暗号化と認証

第 1 章　IPsec ワーキンググループ 2006 年度の活動

IPsec ワーキンググループは IPsec（RFC4301[122] 他）に関わる事項（実装、運用など）を扱っており、現在主に活動している項目は以下である。

(1) IPsec で利用する複数の鍵交換プロトコル（IKE、KINK）を利用可能にするアーキテクチャ
(2) 上記アーキテクチャの実装（名称：racoon2）（注：racoon2 のアーキテクチャについては 2004 年度 WIDE 報告書参照）

racoon2 の特徴を以下に挙げる:

- IKEv2、KINK と 2 つの鍵交換プロトコルをサポート
- 複数の OS 上で動作（Linux、NetBSD、FreeBSD）

本年度は、この racoon2 の開発において、従来からサポートしている鍵交換プロトコル IKEv2、KINK の 2 つに加え IKEv1 への対応、IKE の NAT-Traversal サポートを実装し、2 回の一般向けリリースを行った。また、一般向けに各種の情報提供の場として racoon2 のウェブサイト（`http://www.racoon2.wide.ad.jp`）を立ち上げた。

第 2 章　racoon2 リリース

今年度は racoon2 の 4 回目と 5 回目の一般向けリリースを行なった。リリースはソースコードの形で公開された。なお、racoon2 のソースコードは、`ftp://ftp.racoon2.wide.ad.jp/pub/racoon2/` より取得可能である。各リリースでのトピックを次に挙げる。

- 第 4 回目のリリース（7 月 14 日）
  このリリースでは鍵交換プロトコルとして IKE の NAT-Traversal をサポートし、KINK プロトコルは新たに発行された RFC に準拠した。
- 第 5 回目のリリース（12 月 28 日）
  このリリースでは、鍵交換プロトコルとして IKEv1 のサポート及び IKE において証明書による認証をサポートした。

第 3 章　開発した項目

3.1 IKE

racoon2 の IKEv2 部分に関する基本機能の開発は既に終え、安定化作業のフェーズである。今年度はさらに IKEv1 の対応、応用機能として NAT-Traversal 機能を実装した。

3.1.1 IKEv1 サポート

当初の racoon2 の開発アイテムは、IKEv2 と KINK という 2 つのプロトコルであった。このうち IKEv2 は以前のバージョンである IKEv1 と比較して、

- 同じプロトコル名だが、互換部分が少ない
- IKEv1 を使用したい場合は、racoon(v1) を使用すればよい

という考えから、IKEv1 のサポートは考慮に入れず、IKEv2 に開発を注力してきた。しかし、IKEv2 部分の開発も基本的な機能は一通り終えた現在において、racoon2 と接続する可能性のある他の IPsec 関連製品の IKEv2 への移行は、あまり進んでいない。そのため、IKEv1 のみに対応したノードと接続する場合には、racoon2 でなく racoon(v1) を使用する必要がある。その場合、racoon2 と racoon(v1) では設定ファイルの記述方式が異なるなどの理由によりマシン管理などが煩雑になる面があった。よって、利便性を向上させるため、racoon2 のみで IKEv1 と IKEv2 の両方をサポートする必要が生じ、racoon2 でも IKEv1 のサポートをすることになった。

IKEv1 の実装は、基本的には IKEv1 をサポートしている racoon(v1) のコードをできるかぎりそのまま移植して使用することを基本とした。当初は KAME

プロジェクトの racoon(v1) のコードをベースとして移植を始めた。しかし、racoon2 の IKEv2 コードと同様に NAT-Traversal をサポートするため、NAT-Traversal をサポートした ipsec-tools プロジェクト（http://ipsec-tools.sourceforge.net/）の racoon(v1) コードに途中から入れ替えて移植した。ipsec-tools プロジェクトは現在も racoon(v1) のサポートを続けているため、IKEv1 のコードをできるだけ共通にすることでバグフィックスも共有できるということを期待している。

IKEv1 のコードを移植するために行なった作業の主な点は以下である。

1. ログメッセージ関数 plog() 仕様の違いによる変更
2. PF_KEY インタフェースの違いに関する変更
3. 設定ファイル参照のためのインタフェースの違いによる変更

次にその詳細を述べる。

(1) ログメッセージ関数 plog() 仕様の違いによる変更

racoon(v1) から racoon2 への plog() 関数の変更点は以下である。

- ログレベル定数マクロの変更（LLV_ERROR から PLOG_PROTOERR、など）
- ソースコード位置マクロの変更（LOCATION から PLOGLOC へ）
- 一部のログ関数そのものの変更（plog() から isakmp_log() へ）

plog() 関数はソースコード中の色々な箇所で使用されているため変更個所は多いが、作業自体はほぼ単純な置き換えであった。ただし変更個所が多いため、今後 ipsec-tools のソースコードとの比較・バグフィックス追従が面倒になる可能性はある。

(2) PF_KEY インタフェースの違いに関する変更

カーネルとの PF_KEY インタフェースは、主に、

- カーネルから iked への SADB_ACQUIRE メッセージ
- iked からカーネルへの SADB_ADD/SADB_UPDATE メッセージ
- カーネルから iked への SADB_EXPIRE メッセージ

が存在する。これらのインタフェースに対して、racoon2 仕様にあわせるためにブリッジコードを新たに作成した。PF_KEY インタフェースの SADB_ACQUIRE メッセージの処理に関しては、IKEv1 に関する処理の追加とともに IKEv2 の対応のコードの整理も行い、若干見通しがよくなった。

(3) 設定ファイル参照のためのインタフェースの違いによる変更

設定情報へのアクセスするための方法が racoon(v1) と racoon2 では異なる。racoon(v1)では、コード中で設定情報を格納している構造体のフィールドを直接参照しているが、racoon2 では関数呼び出しに変更している。この変更への対応に関しても、おおむね一対一の機械的な置き換えで済んだが、やはり plog() 関数の変更ほどではないが変更箇所がソースコード中に分散しており、今後のメンテナンスに課題を残している。また、IKEv2 のコードと同様に設定ファイル中の IKEv1 設定の整合性チェックを追加している。

3.1.2 NAT-Traversal サポート

3.1.2.1 はじめに

racoon2 は、鍵交換プロトコルとセキュリティポリシマネジメントを提供する IPsec サブシステムであり、BSD variants や Linux 上で動作するアプリケーションである。

IPsec の利用シーンとしては大きく 3 つに分類することができ、サイト間の情報を保護する Site-to-Site、ホストとサイト間を保護する Host-to-Site、そして 3 つ目はホスト間を保護する Host-to-Host である。Site-to-Site VPN においては通常、固定 IP アドレスが IPsec を終端する端点に付与されているが、それ以外のケース、i.e. Host-to-X についてはその限りではなく、Host が IPsec の端点として利用する IP アドレスは動的にアサインされているものが多い。

racoon2 では、対向する IPsec ノードの端点が固定 IP アドレスを持たない場合の利用シーンにおいても、IPsec を利用できるように対応を行った。また、近年の利用シーンに多くみられる、対向ノードが NAT ボックス配下のプライベートネットワークに属する場合においても IPsec の透過性が得られるように、NAT-Traversal と ESP パケットの UDP カプセル化の対応も行った。

road warrior とは IPsec を用いてリモートアクセスを行うノードで、そのアドレスが動的に変化するノードを指す造語である。よって racoon2 における上述の対応を road warrior 対応と呼ぶことにする。

3.1.2.2 機能要件

racoon2 において road warrior 対応を行った際の機能要件を、大きな粒度で以下に列挙する。

3.1.2.2.1 任意の端点を持つ対向 IPsec ノード対応

(1) 対 road warrior の IKEv2 responder として動作可能であること。
(2) 当該ノードに対するコンフィグレーションが行えること。
(3) 適切な SP をセキュリティポリシマネージャ（SPM）へセットできること。
(4) 適切な SP/SA を BSD variants や Linux のカーネルへセットできること。

3.1.2.2.2 NAT-Traversal

(1) UDP の 4500 番ポートで IKEv2 パケットの送受信が行えること。
(2) UDP の non-ESP marker がハンドリングできること。
(3) IKEv2 の NAT-Detection（NAT-D）ペイロードがハンドリングできること。
(4) IKEv2 の NAT-Keepalive の送受信が可能であること。
(5) ESP の UDP カプセル化と連携できること。

3.1.2.2.3 ESP の UDP カプセル化

(1) IKEv2 の NAT トラバーサルと連携できること。
(2) UDP でカプセル化するための情報をカーネルへセットできること。

3.1.2.3 機能説明

3.1.2.3.1 任意の端点を持つ対向 IPsec ノード対応

当該ノードに対して IKEv2 responder として動作する場合、本来であれば spmd の初期化時に行うセキュリティポリシ（SP）の設定を、IKEv2 ネゴシエーション中へとシフトする必要がある。そのために、spmd 起動時に読み込む設定に当該ノードを示す識別子（IP_RW）を追加し、このような SP は初期化時にカーネルへセットしないようにした。

```
policy rw.foo.com {
        action auto_ipsec;
        remote_index rw.foo.com;
        ipsec_mode tunnel;
        ipsec_index { ipsec_esp; };
        ipsec_level require;
        peers_sa_ipaddr IP_RW;
};
```

次に、responder として CHILD_SA を生成する前処理として、IP_RW の場合は、IKE から spmd を経由してカーネルへ SP をセットする処理を追加した。

```
struct ikev2_child_sa *
ikev2_create_child_responder(...)
{
    ...
    if (pol->peers_sa_ipaddr &&
        rcs_is_addr_rw(pol->peers_sa_ipaddr)) {
        ...
        if (ike_spmif_post_policy_add(...) < 0)
            goto fail_internal;
    }

    sadb_request_initialize(...);

    ikev2_child_getspi(...);
    ...
}
```

また、鍵交換デーモンである IKE も IP_RW を意識する必要があるため、その対応を行った。まず、initiator から IKE_SA_INIT を受信した場合、当該ノードに対しては端点のアドレスにもとづいて行われる remote ディレクティブの検索は常に失敗するため、default ディレクティブへフォールバックし、IKE_SA が生成される。

```
default
{
    remote {
        acceptable_kmp { ikev2; };
        ikev2 {
            kmp_sa_lifetime_time infinite;
            kmp_sa_lifetime_byte infinite;
            kmp_enc_alg { aes128_cbc; 3des_cbc; };
            kmp_prf_alg { hmac_sha1; aes_xcbc; };
            kmp_hash_alg { hmac_sha1; hmac_md5; };
            kmp_dh_group { modp1024; modp1536; modp2048; };
        };
    };
};
```

次に、IKE_AUTH においては ID による検索が行われ、適切な認証方式と鍵を取得することができ、認証フェイズと CHILD_SA 生成を行うことができる。

```
remote rw.foo.com {
    acceptable_kmp { ikev2; };
    ikev2 {
        my_id fqdn "gw.bar.com";
        peers_id fqdn "rw.foo.com";
        peers_ipaddr IP_RW;
        kmp_auth_method { psk; };
        pre_shared_key "${PSKDIR}/rw.foo.com.psk";
    };
};
```

3.1.2.3.2 NAT-Traversal　NAT-Traversal は、鍵交換における端点の間に存在する NAT/NAPT を検知する技術である。通常、IKEv2 は UDP の 500 番ポートを利用してメッセージの交換を行うが、当該ポートを特殊に扱う NAT/NAPT ボックスの存在が危惧されるため、IKEv2 では NAT/NAPT を検知すると同時に速やかにポート 4500 番へ移り、残りのメッセージが交換される。racoon2 においてもこれを実装した。また、カーネルにおいて、UDP カプセル化された ESP か、あるいは IKE パケットなのかを検出する必要があり、そのための技術として non-ESP marker の挿入があり、この対応も行った。

IKEv2 の NAT-Traversal においては、下記の Notify メッセージにより NAT/NAPT の検知が行われる。これらの値は、IKE SPIs、IP アドレスとポートの SHA-1 ハッシュとして表現される。

```
NAT_DETECTION_SOURCE_IP          16388
NAT_DETECTION_DESTINATION_IP     16389
```

これらのメッセージは、IKEv2 の初期交換である IKE_SA_INIT にて搬送される。

Initiator		Responder
HDR, SAi1, KEi, Ni, N(NAT-D[S], NAT-D[D])	→	
	←	HDR, SAr1, KEr, Nr, N(NAT-D[S], NAT-D[D])

racoon2 においてもこれらの Notify メッセージの送受信、そしてこれらの値を比較することにより NAT/NAPT の検知を行う機能を実装した。また、NAT/NAPT ボックスに存在するキャッシュ/マッピングを保持させるための機能である、NAT-Keepalive の送受信の実装も行った。

3.1.2.3.3 ESP の UDP カプセル化　IKEv2 の NAT-Traversal 機能において NAT/NAPT を検知し、その結果、ESP パケットの UDP カプセル化を行う必要がある場合、カーネルに対して UDP カプセル化を行うための情報を渡さなければならない。racoon2 では、PF_KEYv2 フレームワークの NAT 拡張を利用して対応を行った。具体的には、これらのメッセージタイプを利用。

```
SADB_X_EXT_NAT_T_TYPE       20
SADB_X_EXT_NAT_T_SPORT      21
SADB_X_EXT_NAT_T_DPORT      22
```

また、UDP カプセル化のタイプとしては、以下を利用。

```
UDP_ENCAP_ESPINUDP          2
```

BSD variants や Linux カーネルにて一般的に採用されているこれらのタイプを利用することにより、ソースコードのポータビリティを保持することができる。

3.1.2.4 制限事項

現時点で把握している制限事項は、以下のとおり。

(1) selector ディレクティブに IP アドレスを指定しなくてはいけない。例えば、racoon2 が responder として動作する場合、initiator である road warrior 側の selector を IP アドレスで指定する必要がある。他の例としては、racoon2 が responder として動作する場合、NAT/NAPT 配下に位置する initiator の selector IP アドレスを指定する必要がある。いずれのケイスにおいても、トポロジの変化に追従できず、結果として IPsec の通信を阻害する要因となる。

(2) road warrior の端点が変化した場合、再度、SP と SA を生成するが、旧い SP と SA が残ったままとなる。

3.1.2.5 今後の課題

今後の課題としては、上記 (1) の対応として initiator から提案されるトラフィックセレクタ（TS）を利用してカーネルに SP と SA を生成すること。セキュリティの観点や、複数の initiator を収容した場合の TS 重複等を踏まえ、responder である racoon2 において、ある程度の「制御」を行うことが肝要であり、慎重に設計/実装を行う必要があると考える。また、TS の生成という観点からは、Mobile IPv6 対応への考慮も必要であると思われる。

上記 (2) については、road warrior 対応により動

的に生成した SP をカーネルから削除することにより実現できるが、間違って静的な SP を削除してしまわないような工夫が必要かと思われる。

3.2 KINK

racoon2 の KINK プロトコル部分は、関連インターネットドラフトが RFC 化されたことに伴い準拠仕様を RFC4430、RFC3961 に変更した。この変更には以前の準拠仕様である draft-ietf-kink-kink-06 および draft-ietf-krb-wg-crypto-07 と比べ、key usage number、PRF（Pseudo-Random-Function、疑似乱数関数）、payload type、およびパケットフォーマットが変更されているため互換性がないことに注意が必要である。

第 4 章　Mobile IPv6 サポート

racoon2 では、IPsec を利用するプロトコルの 1 つである Mobile IPv6（MIPv6）への対応を行なっている。昨年度の活動で、慶應大学の山下氏によって、NetBSD、FreeBSD の MIPv6 実装である Shisa への対応が行なわれた。これによって、PF_KEY MIGRATE など MIPv6 をサポートするために必要な機能の多くが実装されたが、すべての機能の動作検証はまだ完了していないのが現状である。本年度も引き続き Mobile IPv6 への対応活動を行なっている。本年度は、一般に Road Warrior と呼ばれる IPsec の終端アドレスが設定時に既知でないノードのサポートが正式に行なわれた。これによって、MN-HA 間の IPsec トンネルをサポートすることが可能になり、MIPv6 をサポートするにあたり必要となる大きな機能が実現されたことになる。

一方で、MIPv6 の初期化時に必要とされる MN-HA 間のトランスポートモードの IPsec SA の鍵交換は、IKE が鍵交換に使用するアドレスが Care Of Address（CoA）である一方で、実際に設定する IPsec SA のアドレスは Home Address（HoA）となることから、そのアドレスをどのように取得するべきかの議論を行っているが、現在のところ明確な結論は出ていない（ちなみに、通常の鍵交換では IKE が使用したアドレスと同じアドレスを使って IPsec SA が設定される）。MIPv6 をサポートする上での今後の課題としては、大きく以下の 3 つが挙げられる。

(1) 上記のトランスポートモード IPsec SA の鍵交換
(2) PF_KEY MIGRATE など個別機能の動作検証
(3) Authorization などセキュリティ機能の検証

また、Linux の MIPv6 実装である MIPL に関しては、IPsec ポリシの管理問題を解決する必要がある。これは、MIPv6 の仕様により Return Home 時には IPsec 機能を停止することが求められており、MIPL では、この機能を実現するために MIPL 側で IPsec ポリシを管理するためである。

第 5 章　まとめ

racoon2 の各鍵交換プロトコル部分に関しては、必要となる機能に関しては実装を終えている。今後、各種設定方法などのユーザビリティの向上や MIPv6 サポートのアップデートなど、応用的な事項について、研究および実装を行っていく予定である。

Glossary

IKE (Internet Key Exchange) v2: RFC4306[118] にて定義される鍵交換プロトコル。

KINK (Kerberized Internet Negotiatoin of Keys): draft-ietf-kink-kink-11 にて定義されるノードの認証に Kerberos を利用した鍵交換プロトコル。

NAT-Traversal (Network Address Translation-Traversal): NAT 配下にいるノードでも IPsec により保護された通信を行なうための技術、RFC3715[1]、RFC3948[92] などにて定義されている。

PF_KEY: RFC2367[139] で定義され、BSD socket API を持つアーキテクチャにおいて SA を管理するためのインタフェース。

SA (Security Association): 単方向の論理的なコネクションでセキュリティサービスを提供するための情報の単位。

第 6 章　論文リスト

本年度の ipsec ワーキンググループの論文リストを以下に示す。

- K. Miyazawa, S. Sakane, K. Kamada, M. Kanda, and A. Fukumoto. Design and Implementation to Support Multiple Key Exchange Protocols for IPsec. In *Proceedings of Linux Symposium Volume Two*, pp. 143–149, July 2006.