

# IPsecによるマイコンとサーバ間のセキュア通信

総合技術センター

計測・制御技術分野 飯田 仁 (Hitoshi Iida)

## 1. はじめに

数年前より授業出席管理システムにてカードリーダー用中継装置(マイコン)とサーバ間の通信を暗号化する目的でIPsecを用いている。今回サーバの更新に伴い、再度IPsec通信の構築を行ったので報告する。

## 2. IPsecとは

IPsecとは、一言で表現するとネットワーク上を流れる通信内容を改ざんや盗聴などから保護するための暗号化と認証の規格のこと。OSI参照モデルではネットワーク層に該当するIP(Internet Protocol)のパケットを暗号化して送受信するため、より上層のアプリケーションでは意識することなく安全な通信を利用することが可能になる。

IPsecには通信の確立に2つの段階(フェーズ)があり、この確立のための通信も暗号化される。1段階目で共通鍵による認証が行われる。2段階目でどのような暗号方式などを用いてデータ通信を行うか詳細を決定する。2段階目が終了すると以降は安全にデータ通信を行うことができる。

IPsec通信確立の1段階目では、Internet key Exchange(IKE)というプロトコルにより共通鍵の交換が行われ、2段階目のデータ通信ではIP Encapsulating Security Payload(ESP)とIP Authentication Header(AH)の2つのセキュリティプロトコルと、トランスポートモードとトンネルモードの2つの伝送モードを組み合わせた、4方式から選択する。ただしAHプロトコルでは認証のみで暗号化されない。

1. AH-トランスポートモード (非暗号化)
2. AH-トンネルモード (非暗号化)
3. ESP-トランスポートモード
4. ESP-トンネルモード

今回はESP-トランスポートモードを使用す

ることとした。

データをネットワークに送出する際、平文では、図1のように送信データにIPヘッダとTCP/UDPヘッダを付加しパケットとして、ネットワークに送出されるのに対し、ESP-トランスポートモードでは、図2のように平文とは別のIPヘッダ、ESPヘッダ、ESPトレーラ及び認証データを付加し、さらに平文のTCP/UDPヘッダ、送信データ及び付加したESPトレーラを暗号化した上でパケットとして、ネットワークに送出している。なお、図2のオプションである認証データは今回使用していない。

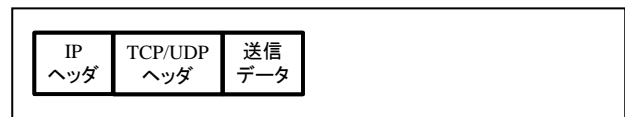


図1 平文のパケット

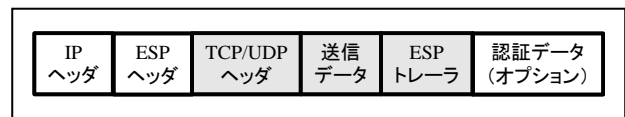


図2 ESP-トランスポートモードのパケット

上記の機能を用いるために、マイコン側にはIPsec暗号化アダプターNS-101(図3)を使用し、サーバにはIPsec用のソフトウェアを導入した。なお、NS-101は縦55×横60mmと非常にコンパクトにまとめられている。



図3 IPsec暗号化アダプターNS-101

マイコンとNS-101及びサーバの接続状況を図4に示す。マイコンからシリアル出力されたデータがXportによりイーサネットのデータに変換され、さらにNS-101にて暗号化されてキャンパスネットワークに送り出される。サーバではNS-101にて暗号化されたデータを復号化した上で、上位の各サービスに受け渡す。このようにキャンパスネットワーク内で安全にデータを送受信できる。

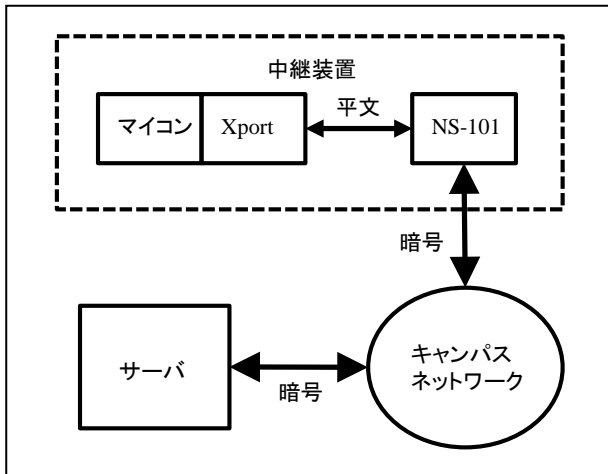


図4 マイコンとの接続状況

このNS-101にはWindowsにて使用可能な専用の設定ソフトがあり、容易に設定が可能であるため設定方法は割愛する。

### 3. サーバの設定

以前より授業出席管理システムにて、マイコン側のXportとサーバ間の通信に利用している暗号化アダプターNS-101であるが、従来利用しているサーバが経年により更新することとなった。サーバOSは最新のものに変更したが、旧サーバのIPsec通信で導入したソフトウェア (ipsec-tools:Racoon)が利用できず、検索を始めた。結果OpenSWANとRacoon2が候補に挙がった。名前からRacoon2は従来使用していたRacoonの後継にあると最終判断して採用することとした。しかし、Racoon2はRacoonの後継ではあるが、機能追加により全くの別ソフトと思われる程異なり、以降の設定作業が難航することとなった。

Racoon2の導入はyumコマンドで可能である。ただし、yumのリポジトリにEPELが導入されている必要がある。設定作業は、インタ

ーネット上に公開されている情報を参考にし、通信の確認作業を行い、暗号通信を確立することができた。以下に設定ファイルの内容を記載する(図5, 図6, 図7)。

```

include "/etc/racoon2/vals.conf";
interface
{
  ike {
    MY_IP port 500;
  };
  kink {
    MY_IP;
  };
  spmd {
    unix "/var/run/racoon2/spmif";
  };
  spmd_password "/etc/racoon2/spmd.pwd";
};
resolver
{
  resolver off;
};
include "/etc/racoon2/default.conf";
include "/etc/racoon2/common.conf";
include "/etc/racoon2/046083.conf";
  
```

図5 racoon2.confの内容

```

ipsec ipsec_adas_esp {
  ipsec_sa_lifetime_time 300 sec;
  sa_index adas_esp_01;
};
sa adas_esp_01 {
  sa_protocol esp;
  esp_enc_alg { aes128_cbc; };
  esp_auth_alg { hmac_sha1; };
};
  
```

図6 common.confの内容

ソフトウェア導入時のデフォルトファイルの内容は割愛した。なお、設定ファイルは/etc/racoon2以下に自動的に配置される。注意として記載内容でIPアドレスに関する部分は念のためプライベートアドレスに変更してい

る。サーバのIPアドレスは192.168.10.3でマイコン側が192.168.46.83である。

```
remote ike_trans_046083 {
  acceptable_kmp { ikev1 };
  ikev1 {
    my_id ipaddr 192.168.10.3;
    peers_id ipaddr 192.168.46.83;
    peers_ipaddr 192.168.46.83 port 500;
    kmp_auth_method { psk };
    pre_shared_key "${PSKDIR}/key.psk";
    exchange_mode main;
    kmp_sa_lifetime_time 360 sec;
    kmp_enc_alg { aes128_cbc };
    kmp_hash_alg { sha1 };
    kmp_dh_group { modp1024 };
  };
  #selector_index ike_trans_046083_in;
  selector_index ike_trans_046083_out;
};
selector ike_trans_046083_out {
  direction outbound;#
  src 192.168.10.3/32 port 80;
  dst 192.168.46.83/32 port any;
  upper_layer_protocol "tcp";
  policy_index ike_trans_policy;
};
#selector ike_trans_046083_in {
# direction inbound;
# src 192.168.46.83.80/32 port any;
# dst 192.168.10.3/32 port 38080;
# upper_layer_protocol "tcp";
# policy_index ike_trans_policy;
#};
policy ike_trans_policy {
  action auto_ipsec;
  remote_index ike_trans_046083;
  ipsec_mode transport;
  ipsec_index { ipsec_adas_esp };
  ipsec_level require;
};
```

図7 046083.confの内容

図7ではtcp通信に関する記述のみであり、

udp通信を利用する場合には一部変更が必要である。また、図7の下線部key.pskの内容は共通鍵に使用する文字列のみが書かれたファイルで、前後にタブや改行などの制御文字が入力されていても通信は確立できない。今回の設定で一番手間取ったところである。

通常viなどのエディタでkey.pskファイルを作成するが、この場合文字列の最後に改行文字が自動的に追加される。結果論だが確認の際は、以下に示すように「cat」コマンドを利用し確認すると違いは明らかである。

```
# cat key.psk
Pre_shared_key#
```

図8 正常なkey.pskの場合

```
# cat key.psk
Pre_shared_key
#
```

図9 不良なkey.pskの場合

図8は正常な内容の場合を示し、図9は不良な内容の場合を示す。図8では次のプロンプト「#」が共通鍵の直後にあるが、図9では共通鍵の次の行に表示されている。すなわち、図9では共通鍵の後に「改行文字」が入力されており、Racoon2は改行文字を含めて共通鍵と認識したため、不一致と判断し通信が確立できなかった。ここでは「Pre\_shared\_key」という14文字が共通鍵であるが、実際は異なる。

確実に共通鍵の末尾に改行文字が追加されずにkey.pskファイルを作成する方法を図10に示す（確認も忘れずに行う）。

```
# printf Pre_shared_key > key.psk
または
# echo -n Pre_shared_key > key.psk
#cat key.psk
Pre_shared_key#
```

図10 key.pskファイルの作成方法

このkey.pskファイルはパーミッションをrootのみ(400)としなければ起動できないので変更する必要がある。

「# chmod 400 key.psk」で変更しておくこと。

#### 4. Racoon2の起動スクリプト

Racoon2を導入した際に作成される起動スクリプトは、そのままでは不具合があり起動することができなかった。

図11が自動で作成された起動スクリプトの一部で図12が修正した起動スクリプトの一部である。実行ファイルの名前等が異なっていたので、内容を修正した。

```
exec1="/usr/sbin/spmd"
exec2="/usr/sbin/iked"
prog1="spmd"
rh_status(){
    status $prog
}
```

図11 不具合のあった起動スクリプト

```
exec1="/usr/sbin/racoon2-spmd"
exec2="/usr/sbin/racoon2-iked
    -l /var/log/racoon2/racoon2.log"
prog1="racoon2-spmd"
rh_status(){
    status $prog1
}
```

図12 修正した起動スクリプト

修正するには「# rpm -qlv racoon2」を実行し、インストールされたファイルのリストが表示されるので参考にした。なお、exec2のオプションはログファイルの指定で、追記しない場合、messageに出力される。以上で起動スクリプトの修正は終わったのでred-hat系OSの場合、OS起動時に処理されるchkconfigに登録しておくことと便利である。登録する場合は、上記の起動スクリプトファイル上部に図13を追記する。

```
#!/bin/sh
# chkconfig: 345 48 52
# description: Starts or stops the Racoon2.
# processname: Racoon2
```

図13 chkconfigへの登録

図13において重要なところは、2行目のchkconfigの行で、345は起動するランレベル

を、48は起動順、52は停止順をそれぞれ示している。起動順はIPsecを使用して通信を行うサービスの開始前に起動する必要があるため、停止順は起動とは逆にサービスの終了後に停止する必要があるため、適宜修正が必要になる。停止順の数值は100-(起動順)とすれば良い。3行目と4行目は必要に応じ記載すればよく、なくても登録は可能である。追記が終わると「# chkconfig --add racoon2」で登録できる。「# chkconfig --list | grep racoon2」で実際に登録されたかどうか確認することができる。

#### 5. 動作確認

前節の変更を加えRacoon2を起動した場合、どのようなポリシーによりIPsec通信が設定運用されているか確認する方法を図14に使用コマンドと共に示す。

```
# racoon2-spmdctl policy show
192.168.10.3/32[80] 192.168.46.3/32[any] tcp
    out ipsec
    esp transport require
    created: 作成日時 lastused: 最終使用日時
    lifetime: 0(s) validtime: 0(s)
    selector=ike_trans_028080_out spid=2017
#
```

図14 ポリシー確認

図14ではサーバ(#80ポート使用)からマイコン(すべてのポート)へのポリシーのみを設定し表示されているが、この状態でマイコンとサーバ間の相互通信に問題はない。逆にマイコン(すべてのポート)からサーバ(#80ポート)へのポリシーのみを設定すると全く通信ができなかった。図7にて行頭に"#"のある行はコメント行であるが、マイコン側からサーバへのポリシーとして参考のために記載した。

#### 6. 自動更新(yum)の不具合

本原稿執筆最終時(2014年1月)にyumによりRacoon2の自動更新が行われた。念のため動作確認すると、稼働状況が正常に表示されず、停止操作もできなかった。強制終了させ起動操作を行うと起動はできた。起動スクリプト

を確認すると、図12のように修正した部分が図15のように変更されていた。ソフトウェアが更新されても不具合は完全に修正されていないようで、図12の状態に修正し動作確認を行ったところ、起動、停止、及び稼働状況確認を含め、すべて正常に動作した。

```
exec1="/usr/sbin/racoon2-spmd"
exec2="/usr/sbin/racoon2-iked"
prog1="spmd"
rh_status(){
    status $prog
}
```

図15 自動更新後の起動スクリプト

ソフトウェアの自動更新は便利ではあるが思わぬ落とし穴があるかもしれないので、自動更新の後には確認した方が良いと感じた。ただし、今回の不具合は本原稿執筆時に偶然発見できたものであり、影響も少なく幸いであった。

## 7. 通信の確認

以上で導入と起動の設定作業が終了したが、実際に通信を行い暗号化が行われているか確認した。確認方法はサーバとマイコンの間にミラーポート付Hub(FS808TP V1)を設置させ短い時刻データをサーバへ送信し、NS-101なしの平文通信と、NS-101ありの暗号通信をそれぞれ確認(キャプチャ)した。キャプチャにはWire Sharkというフリーのソフトを利用した。

キャプチャ結果の時刻データ送信パケットのみを図16(暗号化前)と図17(暗号化後)に示す。なお、図16と図17では念のため、アドレスが分かる部分にモザイクを実施している。図16が図2と、図17が図3とそれぞれ対応している。送信データはそれぞれ171バイトであるが、平文のパケットサイズは225バイトに対し、NS-101にて暗号化すると278バイトと約50バイト増加するが、暗号化され判読が不可能なことが分かる。

0000	00	d3	aa	ad	40	00	40	06	5f	de	08	00	45	00	.....	.....E.		
0010	00	d3	aa	ad	40	00	40	06	5f	de	08	00	45	00	.....	.....E.		
0020	07	ff	97	6c	00	00	50	4f	53	54	20	2f	41	44	41	53	.....	..^....P.
0030	2f	52	45	47	2f	61	6c	69	76	65	2e	70	68	70	3f	52	.....	...l..PO ST /ADAS
0040	55	3d	30	30	26	6e	6f	64	65	3d	30	78	78	78	78	78	.....	/REG/ali ve.php?R
0050	78	78	2d	78	78	78	78	78	78	78	78	20	48	54	54	50	.....	U=00&nod e=0xxxxx
0060	2f	31	2e	30	0d	0a	43	6f	6e	74	65	6e	74	2d	54	79	.....	xx-xxxxx xxx HTTP
0070	70	65	3a	20	61	70	70	6c	69	63	61	74	69	6f	6e	2f	.....	/1.0..Co ntent-Ty
0080	78	2d	77	77	77	2d	66	6f	72	6d	2d	75	72	6c	65	6e	.....	pe: appl ication/
0090	63	6f	64	65	64	0d	0a	43	6f	6e	74	65	6e	74	2d	4c	.....	x-www-fo rm-urlen
00a0	65	6e	67	74	68	3a	20	33	36	0d	0a	0d	0a	54	3d	31	.....	coded..C ontent-L
00b0	34	30	31	31	34	31	35	33	30	31	37	25	32	43	30	30	.....	ength: 3 6....T=1
00c0	31	34	30	31	31	34	31	35	33	30	31	37	25	32	43	30	.....	40114153 017%2C00
00d0	0a																.....	14011415 3010%2C.
00e0																	.....	.

図16 暗号化前のキャプチャデータ

0000	01	08	aa	c0	40	00	40	32	5f	6a	08	00	45	00	.....	.....E.		
0010	00	f2	a8	84	00	00	00	03	82	3a	56	47	57	51	.....	.....@2 _j.....		
0020	5a	0b	e6	3d	e1	5f	e5	9b	58	a5	ca	e4	87	9f	13	1e	.....	.....:VGWQ
0030	cf	de	33	cf	7d	38	32	61	fd	8e	18	3f	55	6c	12	e7	.....	Z..=.. X.....
0040	e4	dc	2e	c0	9d	4a	a5	d7	90	1d	c5	ce	55	52	f7	f6	.....	..3.}82a ...?U]..
0050	24	be	86	77	33	71	c2	b8	33	cf	08	7e	34	73	94	1f	.....	.....J.. ...UR..
0060	f2	4e	2f	e7	29	ee	b9	d5	ac	9f	54	f5	20	d5	89	c4	.....	\$.w3q.. 3..~4s..
0070	21	6a	49	e8	c2	f5	45	8e	10	8c	da	0f	f6	14	1e	ca	.....	.N/.)... ..T. ...
0080	79	4e	6b	1c	0d	f4	79	9f	fa	38	92	e5	62	bd	55	cc	.....	!jI...E. ....
0090	ed	75	0f	15	57	56	0c	1a	6d	13	8c	97	3d	6d	7e	4b	.....	yNk...y. .8..b.U.
00a0	c5	a3	61	81	72	98	57	99	62	b6	4f	6e	81	ce	95	c0	.....	.u..wV.. m...=m~K
00b0	60	4d	2c	76	74	6c	de	89	b1	45	d1	9a	90	57	83	5d	.....	..a.r.w. b.On....
00c0	77	06	d9	10	fe	7b	7e	c5	28	d3	16	be	47	c3	d2	4c	.....	`M,vt].. (E...w.]
00d0	8a	04	49	a9	92	10	ca	5b	60	88	ff	14	e4	51	77	d8	.....	w...{~. (.G..L
00e0	d8	38	5c	d1	40	81	a0	20	9c	88	e2	d6	38	55	db	49	.....	..I....[ `....Qw.
00f0	0d	e9	a7	31	af	28	54	00	c0	10	dc	06	17	2f	6d	27	.....	.8\.@.. ....8U.I
0100	18	94	c2	37	4f	4c											.....	...1.(T. ....'/m'
0110																	.....	...70L

図17 暗号化後のキャプチャデータ

## 8. エラーメッセージ

今回設定を実施した上で発生した主なエラーメッセージと対処法を記載する。なお、前述した起動スクリプトにて起動時に失敗と表示される場合は、設定ファイルに問題があるため確認すること。

図18は、図9に示すkey.pskによる不具合で今回最も苦労したメッセージである。図19はIPsec通信の設定をしていないホスト(マイコン)からIPsec通信の接続があった場合に出力されるメッセージである。図20は、図7に示す046083.confファイルの内容が悪くsrc/dstに続く192.168.43.0/24などとネットワークアドレスを設定した場合に出力される。ここはネットワークでなくIPアドレス192.168.46.83/32を指定しなければならない。図21はracoon2.confに書かれているincludeファイルが無い場合に出力されるメッセージである(図5下部参照)。この場合はRacoon2の起動に失敗するため判断し易い。

## 9. まとめ

今回サーバOSの更新によりIPsecのソフトウェア導入作業を実施したが、出席管理システムを運用している限り将来にも必要となる。その際今回利用したソフトウェア(Racoon2)が利用できる保証はなく、サーバOS更新時の負担軽減のためにも、サーバの設定が不要で、マイコン側で利用可能なXport+NS-101以外の暗号化デバイスを利用できるように検討を始めておく必要性を強く感じた。

### 参考文献：

- [1] マスタリング TCP/IP 入門編(オーム社)
- [2] すっきりわかった！VPN(アスキー)
- [3]<http://nabe.blog.abk.nu/IPsec>
- [4]<http://lists.freebsd.org/pipermail/freebsd-security/2006-May/003686.html>
- [5]<http://www.unixwiz.net/techtips/iguide-ipsec.html>

```
[PROTO_ERR]: ikev1.c:1259:isakmp_ph1resend(): phase1 negotiation failed due to time up (index xxxx).
```

図 18 PSK エラー

```
[PROTO_ERR]: oakley.c:2323:oakley_skeyid(): couldn't find the pskey for 192.168.46.83.  
[PROTO_ERR]: ikev1.c:728:ph1_main(): failed to process packet.  
[PROTO_ERR]: ikev1.c:407:ikev1_main(): phase1 negotiation failed.
```

図 19 未設定エラー

```
[INTERNAL_ERR]: isakmp_quick.c:1856:get_sainfo_r(): can't find matching selector  
[PROTO_ERR]: isakmp_quick.c:1125:quick_r1recv(): failed to get sainfo.
```

図 20 設定エラー

```
[CRITICAL]: main.c:328:main(): failed reading config
```

図 21 設定ファイルなし