実践で学ぶ

一歩進んだサーバー構築・運用術

written by 仙石 浩明

第5回 stone（前編）

これから2回に分けて，アプリケーション・レベルでTCPやUDPのパケットを中継するソフト「stone」を紹介します。ゲートウエイ上でstoneを動作させることにより，内部LAN上のホストと外部ホストの間でパケットをやり取りできます。また，簡易httpプロキシとしても利用することも可能です。

会社を作ります[*1]。3カ月前に転職したばかりで唐突に聞えるかもしれませんが，実は，もともと分社独立を前提にした転職でした。2000年7月中に設立発表を行えるのではないかと思っています[*2]。おかげで現在は，事業計画の立案から会社のロゴのデザイン[*3]まで，設立準備で大忙しです。どうしてこの原稿を書く暇があるのか，自分でも不思議なくらいです[*4]。

もちろん，本業である研究開発の仕事も山のようにあります。さらに，予想していなかった他社動向が明らかになって，その対応でその晩に徹夜をするはめに陥ったりします[*5]。変化が速いこの業界，1カ月先どころか，明日でさえ何が起こるか，全く油断なりません。

パケットを中継するstone

拙作の「stone」は，TCP，あるいはUDPのパケットを中継する，アプリケーション・レベルのパケット・リピータです。特定のホストの特定のポートに対するTCP接続を，他のホストの特定のポートへ中継できるので，便利です。stoneはLinuxをはじめとするUNIX系OSのほか，Windows NTやWindows98/95，OS/2の上でも走らせることができます。stoneの公式ホームページは，http://www.gcd.org/sengoku/stone/Welcome.ja.htmlです。

●TCPの中継

例えば，GCD[*6]の内部LAN上のホスト「takatsu」はプライベート・アドレスであるため，外部からアクセスできません。外部からGCDのゲートウエイ「asao」へアクセスすることはできますし，asaoからtakatsuへアクセスすることもできます。このような場合，asaoでstone を走らせてパケットを中継することにより，外部からtakatsuへアクセスすることが可能になります（**図1**）。

外部からtakatsuへtelnetでアクセスしたい場合は，**図2**のようにasao上でstoneを実行して，asaoのポート「10023」[*7]に対するTCP接続を，takatsuのポート「23」（つまり，telnetのポート）へ中継します。

/etc/servicesに登録されているサービスに関しては，ポート番号を数字で指定する代わりに，サービス名を用いて，**図3**のように実行することもできます。

＊1　私は経営に関しては全くの素人ですから，もちろん社長は私ではありません。私は技術担当（CTO）ということになります。新会社は，次世代携帯電話を対象にコンテンツ関連技術を研究・開発するベンチャ企業です。

＊2　この原稿を執筆している2000年6月中旬現在。

＊3　私はデザインに関しても全くの素人ですから，私はデザイナに描いてもらったサンプルの中から選ぶだけです。

＊4　おまけに，この原稿の締め切り直前に海外出張の予定が入っています。これは飛行機の中で書けっていうことでしょうか。幸い，ビジネス・クラスなので，（その気になれば）原稿の1本や2本，余裕で書けることでしょう… と思っていたら飛行機がやたらに揺れて，原稿を書いていると酔いそうになったので，早々にノートPCの電源を切って寝てしまいました。

＊5　私は徹夜するだけでしたが，新会社の社長（予定）は急きょ翌日に海外出張する羽目になりました。成田発のフライトが取れず，関空発のフライトになったので，その日の最終の新幹線で大阪へ行くというおまけ付きでした。

＊6　筆者が運営する任意団体です。個人で引いたOCNエコノミーの費用の一部を賄うために，さまざまなサービスを提供しています。本連載では，このGCDでのサーバー構築，運用法を例に挙げています。

＊7　もちろん10023番でなくても未使用ポートであれば何番でも良いのですが，覚えやすい番号の方がより良いでしょう。

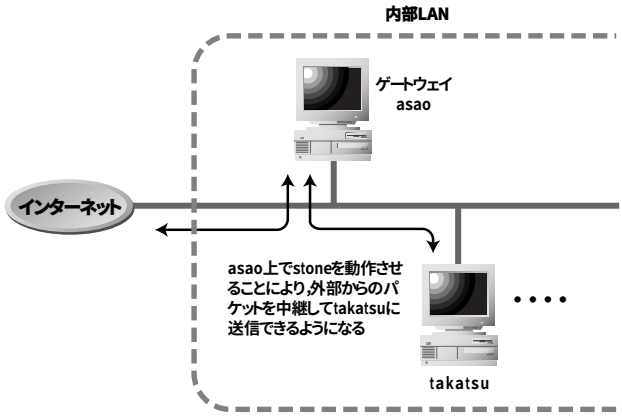

図1　stoneを使うことにより，外部から社内LAN上のマシンにアクセスできるようになる

```
% stone takatsu:23 10023 &
May 29 23:55:15 start (2.1) [22887]
May 29 23:55:15 stone 3: takatsu.gcd.org:telnet <- 10023
```

このマークで改行

図2　stoneの実行例
ゲートウエイのasao上でstoneを実行して，asaoのポート「10023」に対するTCP接続を，takatsuのポート「23」へ中継する。

```
% stone takatsu:telnet 10023 &
```

図3　/etc/servicesに登録されているサービス名を用いても実行できる

```
% telnet asao.gad.org 10023
Trying 210.145.125.162...
Connected to asao.gcd.org.
Escape character is '^]'.

login: sengoku
Password: XXXXXXXX
Last login: Mon May 29 21:31:22 on tty1
takatsu:/home/sengoku %
```

図4　stoneを実行しておくと，asaoのポート10023へアクセスすることにより，takatsuへtelnetできる

このようにstoneを実行しておくと，asaoのポート10023にアクセスすることによって，takatsuへtelnetすることができるようになります(**図4**)。

内部LANがプライベート・アドレスではなかったとしても，多くのファイアウオールは原則としてパケットを通しません。そのような場合にstoneをファイアウオール上で走らせてパケットの中継を行うことにより，ファイアウオールを越えたアクセスが可能になります。

大手企業などでは，複数のファイアウオールが設置されている場合が多いようです(**図5**)。このような場合でも，それぞれのファイアウオールでstoneを走らせることが可能なら，stoneを「飛び石」づたい[*8]にして，内部のホストにアクセスすることが可能になります。

stoneが中継できるパケットは，telnetに限りません。例えば，Xプロトコルを中継すれば，ファイアウオールの外のマシンで走らせたXクライアントを，ファイアウオール内のXサーバーに表示させたり，逆にファイアウオール内のマシンで走らせたXクライアントを，ファイアウオールの外のXサーバーに表示できます。

Xクライアントを実行する外部のマシンのホスト名を「outer」，ファイアウオールを「fwall」，Xサーバーである内部のマシンのホスト名を「inner」とすれば，まずファイアウオール上で次のようにstoneを走らせます。

```
stone inner:6000 6001 &
```

ポート6000番は，Xサーバーのディスプレイ番号0に対応します。以下同様に，ポート6000+n番は，ディスプレイ番号nに対応します。つまり，ファイアウオールの外のマシンで，「DISPLAY」環境変数に「fwall:1」を設定してXクライアントを実行すれば，Xサーバーのinnerに表示されます。

ここでは，ディスプレイ番号1を中継しましたが，ファイアウオール上でXサーバーを実行していなければ（普通は実行しませんね），ディスプレイ番号0を使うこともできますし，逆にディスプレイ番号nまで使用済であれば，n+1を使えばOKです。

stoneを実行する際，環境変数「DISPLAY」が設定済み（例えば「inner:0」）であれば，引数を次のように省略して指定することも可能です。引数「1」はディスプレイ番号「1」を意味します。

```
stone 1 &
```

さて，任意のホストからXサーバーにアクセスできてしまうと，セキュリティ上かなり問題[*9]なので，アクセス可能なホストを制限した方が良いでしょう。

ホストouterからのみ接続を許可するには，次のようにstoneを実行します。

```
stone inner:6000 6001 outer &
```

接続を許可したいホストは，空白文字で区切ることにより複数指定できます。また，IPアドレスが一定の範囲のすべてのホストからの接続を許可することもできます。例えば，IPアドレスが「192.168.1.*（* は任意の

数字）であるホストからの接続を許可する場合には，「192.168.1.0/255.255.255.0」と指定します。

●UDPの中継

stoneはUDPパケットを中継することもできます。例えばファイアウオールの内側にNTP（ネットワーク・タイム・プロトコル）サーバーがなくても，ファイアウオールのfwallでstoneを

```
stone outer:ntp/udp ntp/udp &
```
*10

のように実行して外部のNTPサーバーouterへ中継させることにより，ntpdateコマンドで時計を合わせられるようになります。

```
ntpdate fwall
```

stoneの引数の書式

stoneの引数の書式をまとめます。基本となる書式は，以下の通りです。

```
stone オプション 石 [ -- 石] &
（[ ] 内は0回以上繰り返し可能）
```

「石」は中継の設定です。「--」で区切ることにより複数設定できます。例えば，telnetとXプロトコルの両方の中継を行いたい場合は，**図6**のように実行します。

以下，オプションについて個別に説明します（**表1**参照）。

●-d オプション

-dオプションを指定すると，stoneはデバッグ情報を出力します。-dを複数指定すればより詳細なデバッグ情報を出力できます。

-nオプションを指定すると，stoneが出力するアクセス・ログ，エラー・ログやデバッグ・ログにおいて，ホスト名やサービス名の代わりにIPアドレスやサービス番号を表示します。例えば図2で示した用例で-nオプションを追加すると，**図7**のように表示されます。

●-u オプション

-uオプションは，同時に記憶できるUDPパケットの発信元の最大数の設定します。

UDP上の大抵のプロトコルでは，パケットを受け取ったホストが発信元のホスト・ポートに対して「返事」パケットを送り返します。

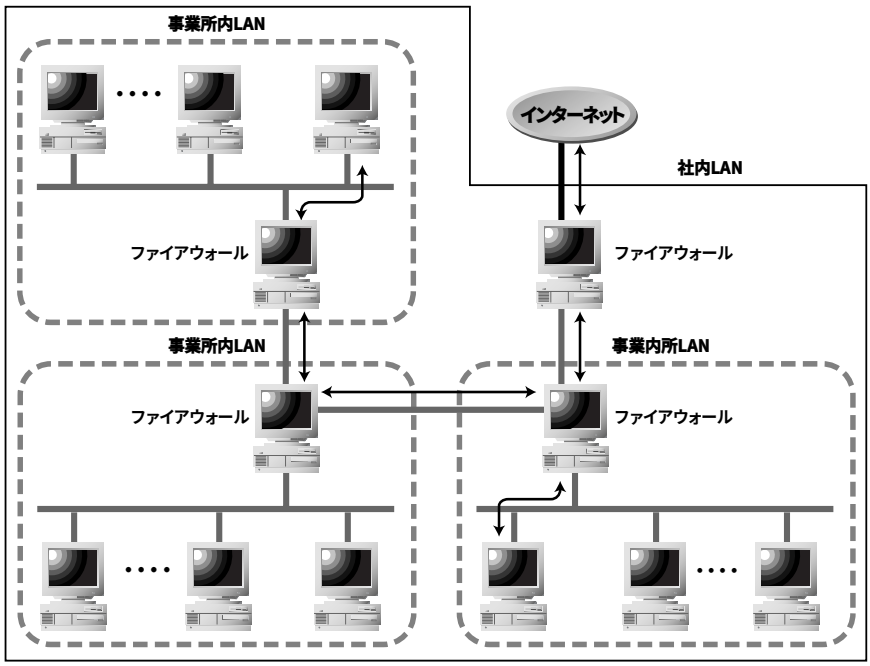

図5　複数のファイアオールを飛び石づたいに，外部から内部のマシンにアクセスできる

```
stone inner:23 10023 -- inner:6000 6001 outer &
```

図6　telnetとXプロトコルの両方の中継を行う場合の実行例

表1　stoneのオプションとその役割

オプション	役割
-d	デバッグ・レベルを増加
-n	ホスト名やサービス名の代わりに，IPアドレスやサービス番号を表示
-u 数値	同時に記憶できるUDPパケットの発信元の最大数の設定
-f 数値	子プロセスの数を設定
-a ファイル名	アクセス・ログの出力先の指定
-L ファイル名	エラー・ログの出力先の指定
-l	エラー・メッセージなどをsyslogへ出力（デフォルト標準出力）
-o 数値	stoneを実行するユーザーIDの指定
-g 数値	stoneを実行するグループIDの指定
-t ディレクトリ	chrootするディレクトリの指定
-z SSLオプション	次号「stone（後編）」で解説予定
-C ファイル名	設定ファイルの読み込み
-P コマンド名	設定ファイルを読み込むときのプリプロセッサの指定

*8　これが「stone」の名前の由来です。
*9　例えば，画面に表示している文字などを盗み読みされたり，キーボードで打った文字列を盗聴されたりする危険があります。
*10　「ntp/udp」はUDPのポート123番です。

```
% stone -n takatsu:23 10023 &
May 29 23:55:15 start (2.1) [22287]
May 29 23:55:15 stone 3: 192.168.1.7:23 <- 10023
```

図7　-nオプションを追加して実行した例
ホスト名の代わりにIPアドレスが，サービス名の代わりにポート番号が表示される。

表2　自動的に設定されるマクロ

マクロ	内容
HOST	ホスト名
HOST_ホスト名	1
HOME	ホーム・ディレクトリ

```
#ifdef HOST_asao_gcd_org
-l -dd
-f 2
-o 99 -g 99

takatsu.gcd.org:telnet  1023    outer1 outer2  --
takatsu.gcd.org:6000    6001    210.145.125.160/255.255.255.192  --
#endif
```

図8　設定ファイルの例
ホスト名がasao.gcd.orgである時のみ，「#ifdef ... #endif」部分の指定が有効になる。

したがって往路のパケットをstoneで中継した場合，パケットを受け取ったホストは，stoneに対して返事を送信してきます。したがってstoneはこれを元の発信元ホスト・ポートへ送り返さなければならないわけです。そのためstoneは，返事パケットが返ってくるまで，発信元ホスト・ポートを記憶しています。

ところが，プロトコルによっては返事パケットが返ってくるまでに時間がかかるものや，そもそも返事を返さないプロトコルもあります。UDPなのでパケットが届かないこともあります。返事が返ってくるまで発信元を記憶し続けていると，stoneが消費するメモリー領域がどんどん膨らむ恐れがあります。そこでデフォルトでは記憶する発信元ホスト・ポートの数を10個に制限して，返事がないものについては発信元を次々と忘れるようにしています。

大抵の場合，返事は速やかに返ってくるので，個人的な中継[*11]であれば10個程度で十分と思われます。しかし，返事が返ってくるまで時間がかかるプロトコルの場合や，多人数でstoneを利用する場合は，-uオプションを指定して記憶する送信ホストの数を増やす必要があります。

●-f オプション

stoneのデフォルトの設定では，多数の中継を行う場合でも，単一プロセスとして動作します。これはstoneを実行するホストに対する影響を最小限にする，という配慮からなのですが，多人数でstoneを利用する場合は単一プロセスだと，ソケット数の上限に達してしまったり，処理が追い付かない場合があるかも知れません。このような問題が生じたときは，-fオプションを使って複数プロセスで動作させてください。

●-a, -L, -l オプション

stoneはデフォルトでは，すべてのログを標準出力に出します。-lオプションを指定すると，syslogに出力します。stoneをデーモンとして動かすときは，-lオプションを指定した方が良いでしょう。-a，-Lを指定すると，それぞれアクセス・ログ，エラー・ログを指定したファイルに出力します。

●-o, -g, -t オプション

root特権が必要なポート，すなわち1023番以下のポートを中継する場合は，stoneをroot権限で実行する必要があります。ただし，いったんポートを開いてしまえば，root権限は必要ありません。そこで，-o, -gオプションを使って，stoneを実行するユーザー/グループIDを変更すると良いでしょう。万一stoneのセキュリティ・ホール（現時点ではセキュリティ・ホールの存在は確認されていません）を突く攻撃を受けても，権限のないユーザーIDで実行されていれば，安心です。

さらに，-tオプションを使ってstoneを実行するルート・ディレクトリを変更すれば，被害をそのディレクトリ（犠牲パーティション）に限定できるので，より一層安心です。

●-C, -P オプション

stoneはすべての設定をコマンド・ラインで指定できますが，多数のオプションや「石」をコマンド・ラインから指定するのは面倒かも知れません。そこで設定ファイルにオプションと「石」を書いておき，-Cオプションでstoneに読み込ませられるようになっています。設定ファイルの書式はコマンド・ラインと同じです。設定ファイル中の改行は無視されるので，適当な場所に改行を入れて読みやすくできます。

コンパイル時に設定していれば，stoneはプリプロセッサ（cppなど）を通してから設定ファイルを読み込みます。したがって，#defineなどでマクロを設定して使うことが

できます。**表2**に示したマクロは自動的に設定されます。

コンパイル時に設定したプリプロセッサと異なるプリプロセッサを用いる場合は，-Pオプションを利用してください。

設定ファイルの例を**図8**に示します。この例では，ホスト名がasao.gcd.orgである時のみ，マクロ「HOST_asao_gcd_org」が定義されるので，「#ifdef … #endif」部分の指定が有効になります。

この設定ファイルを，例えば /etc/rc.d/stoneというファイル名で保存しておいて，

```
stone -C /etc/rc.d/stone &
```

などと実行します。

「石」の設定方法

中継を設定するための「石」は，**図9**で示したいずれかの形式で定義します。「石」は「--」で区切ることにより，複数個指定できます。

図9の(1)は基本型で，「受けポート」への接続を「あて先ホスト」の「あて先ポート」へ中継します。IP アドレスを複数持つホストでは，「受けポート」の前に特定のアドレス「受けホスト:」を追加することにより，「受けホスト」への接続のみを転送できます。

例えば**図10**のように実行すると，localhostのポート10023番への接続のみを中継します。ループバック・インターフェース(ホスト名localhost，デバイス名lo)は，そのホスト自身からのみアクセスできますから，その他のホストからアクセスできなくなります。

(2)は，Xプロトコル中継のための省略記法です。「ディスプレイ番号」への接続を，環境変数DISPLAYで指定したXサーバーに中継します。

```
(1)　あて先ホスト:あて先ポート　受けポート　接続許可リスト
(2)　ディスプレイ番号　接続許可リスト
(3)　proxy(プロキシ)　受けポート　接続許可リスト
(4)　あて先ホスト:ポート/http　受けポート　リクエスト　接続許可リスト
(5)　あて先ホスト:ポート/proxy　受けポート　リクエスト・ヘッダー　接続許可リスト
```

図9　中継の設定形式

```
stone takatsu:telnet localhost:10023 &
```

図10　特定のインタフェースに届いたパケットのみ中継

(3)は，stoneを簡易httpプロキシとして使う場合の設定です。普通のhttpプロキシに比べると，

- キャッシュ機能がない
- ftpサーバーにアクセスできない

などの機能上の制約があります。機能は限定されていますが，ファイアウオールの外側から内側のLAN上のWebサーバーを参照するといったように，対象が限定されていて，本格的なhttpプロキシだと役不足な場合であれば，stoneを使うという選択も有り得ると思います。

(4)は，httpリクエストに乗せてパケットを中継します。ファイアウオール上でstoneを走らせることができない場合でも，ファイアウオールの内側と外側で通信できるようになります 。これについは後述します。

(5)はhttpパケットを中継する際，指定したリクエスト・ヘッダを追加して転送します。fwallのポート8080番でhttpプロキシが走っていたとすると，例えばlocalhostで**図11**のように実行した上で，localhostでWebブラウザを立ち上げ，httpプロキシとしてlocalhostの9080番を指定します。これにより，Webブラウザが送信するリクエスト・ヘッダーに**図12**の文字列を挿入した上で，fwall上のhttpプロキシへ転送します。「Proxy-Authorization:」は，認証を必要とするhttpプロキシを使う場合に必要なフィールドです。「c2VuZ29rdTpoaXJvYWtp」はユーザーIDとパスワードを「:」でつないだ文字列「sengoku:hiroaki」[*12]をMIME base64で符号化して作った文字列です[*13]。Webブラウザ側に「Proxy-Authorization:」フィールドを送信する機能がない場合は，stone経由でhttpプロキシへアクセスすることにより，パスワードを入力することなくhttpプロキシを利用できます。

(1)において，「受けポート」に「/udp」を付けると，TCPパケットを中継する代わりに，UDPパケットを中継します。

また，(1)において，「あて先ポート」に「/apop」を付けると，POPをAPOPに変換して中継します。例えばAPOPのみを受け付けるPOPサーバーmx.gcd.orgに対して，APOPに対応していないMUA(メール・ユーザー・エージェント)からアクセスしたい場合，MUAを走らせるホスト上で，**図13**のようにstoneを実行しておきます。するとMUAの設定でPOPサーバーをlocalhostに設定すれば，mx.gcd.orgに届いたメールを読むことができます。

＊11　元々stoneは個人的な中継を目的に開発されました。ファイアウオールの管理者の手を煩わすことなく，1ユーザーがファイアウオール越えを手軽に実現できるようにすることを第一の目的としています。もちろん現在のバージョンでは多人数での使用も考慮されているわけですが，-uオプションのデフォルトが10であり，また-fオプションのデフォルトが0であるあたりに，元々の目的の痕跡が残っています。

＊12　もちろん，こんな安易なパスワードを使ってはいけません。

＊13　MIME base64符号化を行うには，nkfを使うと便利です。

```
stone fwall:8080/proxy localhost:9080 'Proxy-Authorization: Basic c2VuZ29rdTpoaXJvYWtp' &
```

図11　リクエスト・ヘッダを追加して転送する場合の実行例

```
Proxy-Authorization: Basic c2VuZ 29rdTpoaXJvYWtp
```

図12　これが挿入される

```
stone mx.gcd.org:pop/apop localhost:pop &
```

図13　POPをAPOPへ変換しつつ中継

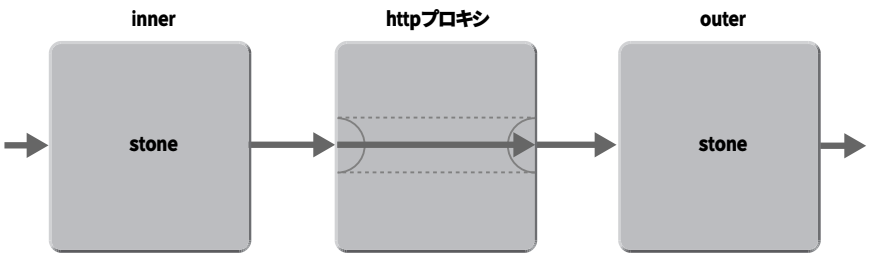

図14　httpプロキシを経由してパケットを転送できる

```
stone fwall:8080/http 10023 'CONNECT outer:443 HTTP/1.0' &
```

図15　httpプロキシを経由してパケットを転送する機能の実行例
httpプロキシの内側にあるinnerで以下を実行する。

```
stone localhost:telnet 443 &
```

図16　httpプロキシの外側のホストouterでは以下を実行する

(1), (3), (4), (5)の形式において,「あて先ポート」に「/base」を付けると, パケットをMIME base64で符号化して中継します。同様に「受けポート」に「/base」を付けると, MIME base64で復号化して中継します。base64はその名の通りアルファベットなど64文字しか用いない符号化ですから, 8ビット・キャラクタすべては通さない通信路を利用することが可能になります。

また, (1), (3), (4), (5)において,「あて先ポート」に「/ssl」を付けると, パケットをSSL(セキュア・ソケット・レイヤー)で暗号化して中継し,「受けポート」に「/ssl」を付けると, SSLで復号化して中継します。SSLについては次号「stone(後編)」で解説する予定です。

●over http

大企業のファイアウオールなどだと, 一般ユーザーはファイアウオール上にアカウントをもらえないことが多く, 自由にstoneを走らせることができない場合がほとんどだと思います。そんな方のために, stoneにはhttpプロキシを経由してパケットを転送する機能もあります(**図14**)。

ほとんどのhttpプロキシは, httpsプロトコルを中継するために,

```
CONNECT outer:443 HTTP/1.0↵
↵
↵は, 0x0D 0x0A の 2 byte
```

を送ると,

```
HTTP/1.0 200 OK ↵
↵
```

などのレスポンスを返した後, ホスト「outer」のポート443番(https)に接続してくれます。本来の用途では, この後SSLによる暗号通信を開始するのですが, 他のプロトコルを流すことも可能です。

つまり,

httpプロキシの内側ホスト(inner):
(1) httpプロキシに対して,「CONNECT outer:443 HTTP/1.0(改行)(改行)」を送信
(2) httpプロキシが返信したレスポンスを無視

httpプロキシの外側ホスト(outer):
(1') outerのポート443番へアクセスが来たら, それを目的のホスト・ポートへ転送

という手続きを踏めば, 後はinnerとouterの間で自由に通信ができます。この手続きは, stoneの「あて先ポート」に「/http」を付けることにより実現できます。

例えば, httpプロキシがファイアウオールfwallのポート8080番で走っている場合, innerで**図15**を実行し, outerで**図16**を実行しておけば, innerのポート10023番がouterのtelnetのポートに中継されます。