フォワードプロキシサーバ経由

●HTTPS 通信の場合、HTTP リクエストの 1 つである CONNECT メゾット（プロキシを経由して HTTPS 通信を行う時に使用）を用いて、接続先のホスト名（FQDN 又は IPA）とポート番号を指定する。その際、プロキシの復号機能の有無によって、処理方法が異なる。

①復号機能あり：クライアントとプロキシの間、プロキシとサーバの間には、TCP コネクションと TLS セッションが確立される。よって、TLS に係るプロキシ証明書とサーバ証明書が必要になるが、プロキシ証明書は、サーバの CN と同じものをプロキシで作成する。作成は動的に作成する。ただし、本物ではないため、動的証明書のルート証明書をアクセスしてくる全てのクライアントの端末にインストールしておかないと、警告画面が出てきてしまう。

②復号機能なし：プロキシはトンネリングされて、TLS セッションはクライアントとサーバ間のみとなり、プロキシサーバの証明書は不要となる。TCP コネクション 2 つのまま。

●フォワードプロキシは、代替 HTTP ポート 8080 番で受けて。Web サーバは 443 で受ける。

●HTTPS 以外のトンネルも可能。つまり、プロキシを中継することで、通信元から通信先に全てをスルーしてダイレクトに通信が可能となる。ただし、マルウエアに CONNECT メゾットを悪用されないように接続先のポート番号の制限も必要。

●HTTP 通信の場合、CONNECT ではなく、GET メゾットを用いて、接続先のホスト名（FQDN 又は IPA）とポート番号を指定する。TCP コネクション 2 つのみで、TLS セッションはない。

リバースプロキシサーバ経由

●公開 Web サーバの IPA をリバプロの GIPA として公開していることから、クライアントはリバプロの GIPA を宛先とする。また、ポート番号は 80 や 443 である。この時、フォワプロが介在していても、送信元の IPA がフォワプロの GIPA に変換されるだけである。

●以上のことから、フォワプロとリバプロの両方がある場合、その間の IPA は送信元がフォワプロの GIPA であり、宛先がリバプロの GIPA となる。

●その後、リバプロから Web サーバ間は、新たに TCP コネクションが確立し、送信元がリバプロの **PIPA** で宛先が Web サーバの **PIPA** となる。

HTTPS 対応の負荷分散

●負荷分散装置（リバプロ等）に SSL アクセラレータ機能を付与して HTTPS を復号して中身を見る必要がある。よって、TLS を負荷分散装置で終端しなければならない。なお、暗号範囲は TCP ペイロード（アプリケーション層全体）であり、この部分を復号することになる。

●TLS 付与前　TCP/IP ヘッダ（ポート番号 80）＋ペイロード（アプリケーション全体）

TLS 付与後　TCP/IP ヘッダ（ポート番号 443）＋TLS ヘッダ＋ペイロード（アプリケーション全体）

＋MAC

※MAC はペイロードの MD であり、認証範囲はペイロードのみ。暗号範囲は黄色部分の MAC と MAC 対象部分のペイロード。

●またポートフォワーディング方式は、単純に転送するだけで、通信の終端とならないため、負荷分散はできない。

クライアントにおけるサーバ証明書の検証方法

●Web ブラウザが、サーバ証明書の検証（有効期限、失効状況、CN（Common Name）とアクセス先の FQDN の一致、証明書のパスの正当性等）の際、問題があれば、Web ブラウザは警告メッセージを表示する。

●証明書のパスの正当性は、証明書のパスに従って、ルート CA（Certificate Authority）の署名までの検証を行うことであるが、具体的には以下のとおり。なお、中間 CA は 1 つだけとしている。

①サーバ証明書を発行した中間 CA の公開鍵を使用して、サーバ証明書に付与された中間 CA による電子署名を復号して、証明書の署名対象部分のハッシュ値と一致していれば OK。ただし、中間 CA の公開鍵は中間 CA 証明書にあるが、この時点では中間 CA 証明書の正当性は確認していないため、サーバ証明書の正当性も確認していないことになる。

②中間 CA 証明書を発行したルート CA の公開鍵を使用して、中間 CA 証明書に付与されたルート CA による電子署名を復号して、証明書の署名対象部分のハッシュ値と一致していれば OK。ここで、ルート CA の公開鍵はルート CA が自己署名したルート証明書にあるため、この自己署名を信用することで、パスの正当性が検証されたことになる。

③パスの正当性が検証できたのならば、サーバ証明書が信頼された認証局から発行されたことを確認したことになる。

●上記のパスの正当性が検証されたとしても、証明書が接続先のサーバから送られてきていることは検証できていないため、CN とアクセス先の FQDN の一致を確認しなければならない。

●普通はデジタル証明書を使用する主体（サブジェクト）が鍵ペアを生成して証明書の発行要求をするが、例えば、クライアント証明書の用途が決まっている場合、CA サーバ等で鍵ペアを生成し、クライアント証明書を端末にインストールすることもある。この時、秘密鍵は端末の TPM に格納して保護するようにする。

●TPM（Trusted Platform Module）は、マザーボード上のセキュリティチップであり、耐タンパ性に優れている。そのため、盗み読みや無理な取り出しに対する防御力は大きいことから、クライアント証明書に係る秘密鍵を収納する。

●クライアントが自力で CRL を取得する代わりに、CRL を随時取得している OCSP（Online Certificate Status Protocol）レスポンダサーバに問い合わせる方法もある。

●OCSP は、失効情報のみのチェックであるが、SCVP（Simple Certificate Validation Protocol）は、有効期限や証明書のパスの正当性もチェックしてくれる。

サーバ証明書

証明書の種類	CN (Common Name)	O（Organization Name)	証明時の確認事項
DV 証明書（Domain Validation）：ドメイン認証型	Web サイトの FQDN	記載なし	ドメイン名の使用権
OV 証明書（Organization Validation）：企業認証型		Web サイトの運営団体の組織名	上記に加え組織の法的実在性
EV 証明書（Extended Validation）			上記に加え組織の物理的実在性や組織の運営等