Secure 属性と HSTS

●HTTP レスポンスメッセージの Set-Cookie ヘッダフィールド（←ステータスラインではない。ステータスラインは「HTTP1.1/ 200 OK」等）には、Cookie と共に様々な属性が設定（←横に並べてズラズラ書いていく：Set-Cookie: id=pon; Secure; HttpOnly …）される。この 1 つである Secure 属性は、ブラウザに対して、HTTPS 通信の時にのみ、Cookie を送ることを命令する。

●混同として、STS（Strict Transport Security：HTTP を冠して HSTS という場合もある）があるが、これもヘッダフィールド名であり、Set-Cookie と同次元である。STS は、ブラウザに対して、HTTPS 通信を命令する。

HTTP ヘッダフィールド

●Referrer（Referer で表記する:リクエストヘッダ）は、リンク元の URL の情報

●X-Forwarded-For（リクエストヘッダ）は、実際の送信元の IPA の情報

●Location（レスポンスヘッダ）は、リダイレクト先の URL 情報

Referer ヘッダフィールド（リクエスト時）

●リンク元の URL 情報であるが、クエリストリング情報も一緒に送られるため、クエリストリングにセッション管理情報が含まれる場合には、リンク先に当該情報が漏洩する。そもそも、Referer ヘッダフィールドは、リンク元の情報をリンク先に伝達することが役目である。

●そもそも、Referer に関係なく、GET メソッドでセッション管理情報を渡さなければいいので、Web サーバで URL リライティング機能を無効にしておけばよい。これによって、攻撃者が用意した URL+セッション ID を利用したセッションフィクセーション攻撃の防御にもなる。

HTTP メッセージヘッダとボディ

●空行はヘッダとボディの間だけで、リクエストライン又はステータスライン（ライン）とヘッダの間に空行はない。むしろ、ラインはヘッダの一部とみなされている。

Cookie の前方一致と後方一致

●domain 属性の後方一致、path 属性の前方一致

●domain 属性が正しく機能しないバグを Cookie Monster Bug という。