Content Filtering e SSL/TLS

Concludiamo la nostra carrellata su SSL/TLS e la navigazione web *sicura*, considerando un altro problema attuale e di non semplice soluzione, anche dal punto di vista normativo/legale. Sappiamo bene che è possibile scaricare da siti web materiale indesiderato (almeno a livello aziendale) o pericoloso. Per questo in molte realtà aziendali sono stati introdotte delle applicazioni dette di *Content Filtering* di cui abbiamo già parlato nei numeri 27 e 28 (Ottobre/Novembre 2004). Queste applicazioni funzionano come Proxy Web e filtrano contenuti quali Active-X, Java, popups, ma anche immagini e quant'altro non si voglia far arrivare dai server (di solito in internet) ai browser web dei propri utenti.

L'utilizzo di una connessione diretta cifrata con SSL/TLS tra il browser ed il server web rende ovviamente impossibile il content filtering. Infatti, poiché di norma il canale cifrato va direttamente dal browser dell'utente al sito finale in internet, il proxy aziendale vede passare solo pacchetti cifrati ed ha generalmente solo due possibilità: o permette qualunque traffico facendo al più un filtro sugli indirizzi IP dei siti ma nessun filtro sui contenuti, o vieta del tutto il traffico cifrato. Si noti come anche la possibilità del *key escrow* (discusso negli articoli precedenti) quasi sempre non sia possibile: ovviamente non è possibile ottenere le chiavi segrete dei server in internet, ed è sicuramente difficile modificare e gestire i browser dei propri utenti in modo che lo permettano.

Per aggirare il content filtering, recentemente siti che distribuiscono materiale spesso non gradito o pericoloso hanno incominciato ad offrire accessi al sito via SSL/TLS. In questo modo i contenuti non permessi possono essere ricevuti dall'utente finale tramite il canale cifrato aggirando così il content filtering aziendale, e rendendo molto più difficile per l'amministratore rendersi conto di cosa sia stato scaricato.

Si pone quindi il problema di come fare content filtering anche su connessioni protette da SSL/TLS in pratica aggirando la protezione end-to-end offerta da SSL/TLS stesso. In questo articolo descriveremo una possibile soluzione applicabile in una tipica situazione aziendale. Se da un punto

di vista tecnico quello che descriveremo non è particolarmente complicato, vogliamo allertare immediatamente chi volesse implementare questa soluzione che è necessario prima di tutto informare i propri utenti della procedura, nel caso ottenere la loro accettazione, ed essere sicuri di sottostare agli obblighi di legge imposti ad esempio dal Testo Unico sulla Privacy e dallo Statuto dei Lavoratori.[1]

Proxy con SSL/TLS MitM

Nei precedenti articoli abbiamo descritto il protocollo SSL/TLS, dovrebbe essere ovvio come poter introdurre content filtering in un proxy web su sessioni SSL/TLS. In pratica si tratta di avere un proxy web che effettua un attacco Man-in-the-Middle alla connessione tra il browser ed il server finale. Quando il browser invia il primo messaggio di una connessione SSL/TLS, il proxy lo intercetta e risponde direttamente dando un proprio certificato digitale. Il browser dell'utente stabilisce così una sessione cifrata con SSL/TLS con il proxy. A sua volta il proxy stabilisce una connessione cifrata con il vero server web. In mezzo, ovvero all'interno del proxy, i dati inviati dal client al server e viceversa, non sono cifrati ed il proxy può quindi applicare loro il content filtering.

Il punto più delicato di questa procedura è nel certificato che il proxy invia al browser dell'utente. Idealmente la procedura dovrebbe essere la seguente. Il proxy intercetta la richiesta SSL/TLS del browser ed invia la propria richiesta al server finale. Quando il proxy riceve il certificato del vero server web e lo valida, crea un proprio certificato con i dati del server web e lo firma con la propria CA. Si noti come per poter fare ciò, sul proxy deve essere installata una CA in grado di generare certificati in tempo reale. A questo punto il proxy invia al browser il certificato appena generato e conclude l'handshake sia con il browser che con il server. In questo modo il browser riceve dal proxy un certificato con gli stessi dati del vero server, ma firmato dalla CA installata sul proxy. Si danno quindi due casi

1 Nel caso si consiglia una consulenza da parte di un esperto legale del settore.

1. il browser riconosce la CA del proxy, quindi accetta il certificato senza dir nulla all'utente
2. il browser non riconosce la CA del proxy, in questo caso chiede all'utente se vuole proseguire o meno e gli mostra il certificato appena generato dal proxy (questo ovviamente accade per qualunque sito protetto con SSL/TLS).

Il secondo caso rende la navigazione "sicura" abbastanza noiosa per l'utente, che può lamentarsi di dover accettare nuovi certificati, emessi sempre dalla stessa CA, ogni qualvolta accede ad un sito SSL/TLS. Per evitare ciò, e rendere la presenza del proxy trasparente all'utente finale, l'amministratore di sistema può installare su tutti i browser degli utenti il certificato della CA del proxy, ricadendo così nel primo caso.

Vogliamo sottolineare come questa *applicazione* sfrutti le debolezze dell'architettura SSL/TLS+CA+PKI che abbiamo descritto nei precedenti articoli, e proprio per questo abbiamo detto che il proxy fa un attacco MitM al client, ovvero al browser dell'utente.

Conclusioni

Alcune considerazioni finali sono utili. Si noti come il proxy, oltre al content filtering sui contenuti inviati dai server, può filtrare anche le richieste dei client, ed ha la responsabilità di verificare i certificati dei veri server web. Visto che l'utente finale non riceve più il certificato del vero server web, non può più verificarlo ed accettarlo o rifiutarlo manualmente nel caso in cui la CA che l'ha firmato non sia riconosciuta dal browser. Il proxy ha quindi il compito di decidere se accettare o meno il certificato, di solito senza alcun intervento da parte dell'amministratore. In pratica l'amministratore del proxy deve decidere quali CA devono essere accettate, opzionalmente quali certificati singoli devono essere accettati e quali no. Ovviamente queste decisioni dell'amministratore per il proxy vengono applicate a tutti gli utenti.

Concludiamo notando che alcuni proxy commerciali offrono questa funzionalità. Per chi comunque fosse interessato ad approfondire l'argomento dal punto di vista tecnico e volesse capire come possa essere implementato in pratica un tale proxy, può essere utile guardare il progetto open (ma ora

abbandonato) *reapoff*. Va anche segnalato che funzionalità simili sono offerte da vari strumenti di analisi del traffico, quali ettercap e dsniff.

Andrea Pasquinucci

pasquinucci@ucci.it

Riferimenti Bibliografici

[1] SSLv3.0: http://wp.netscape.com/eng/ssl3/

[2] TLSv1.0 descritto in RFC-2246

[3] reapoff: http://reapoff.sourceforge.net/

[4] ettercap: http://ettercap.sourceforge.net/

[5] dsniff: http://www.monkey.org/~dugsong/dsniff/