Bug & Debug

L'exploit del mese a raggi X

Gianni Amato, programmatore e "smanettone", è esperto di Sicurezza nel Web

Se i blog diventano virali...

Quando navighi, attento alle Web application a cui dai fiducia. Qualcuna potrebbe tradirti e sfruttare la tua identità per sferrare pericolosi attacchi CSRF. Come ha fatto il noto miniblog Tumblr!

Il **Cross Site Scripting** (XSS), ormai se ne sente parlare ovunque, è una delle tecniche più utilizzate per colpire le Web Application. Per comprendere la gravità del problema basta ricordare "*Nduja Connection*" del ricercatore *Rosario Valotta*, il primo worm, pubblicato sotto forma di Proof of Concept, che dimostra come sia possibile sfruttare le vulnerabilità delle più note webmail italiane per diffondersi e replicarsi. Un po' meno note, ma altrettanto pericolose, sono le vulnerabilità di tipo **Cross Site Request Forgery** (CSRF). L'attacco CSRF viene sferrato per far compiere azioni malevoli, ad un utente lecitamente autenticato, senza che questo le autorizzi. Per meglio comprendere il concetto è necessario un esempio. Un cliente si autentica sul sito Web della propria banca, salvando un cookie di sessione, per controllare il saldo. Nel frattempo, però, con sessione aperta sul sito della banca, visita forum e blog vari. In uno di questi blog potrebbe essere nascosto un tag HTML di questo tipo:

SUL CD

Il plug in **NoScript** che fornisce un filtro contro attacchi XSS

```
<img src="http://bank.site/bonifico?account=mario.rossi&amount=1000&for_iban=0012345678">
```

In questo ipotetico caso l'attaccante riuscirebbe a farsi recapitare sul proprio conto un bonifico di mille euro dal conto della vittima senza che questa se ne accorga minimamente.

La vulnerabilità CSFR

Le vulnerabilità di tipo *Cross Site Request Forgery* sono più diffuse di quanto si creda, ma non è possibile fornire numeri precisi in quanto non esistono metodi affidabili, a parte una scansione manuale per scovarli. Inoltre, per gli sviluppatori non è così semplice difendersi da questo tipo di vulnerabilità se non se ne è pianificata la mitigazione in fase di design. L'utilizzo di token salvati all'interno di cookie e il controllo dell'indirizzo di referer non sono modi sicuri per mitigare il problema. Il primo metodo è facilmente aggirabile in quanto il contenuto del cookie viene inviato ad ogni richiesta fatta dal browser all'applicazione rendendo di fatto l'attacco CSRF completamente funzionante. Il secondo metodo, solitamente più utilizzato, può essere bypassato con varie tecniche tra cui l'uso del seguente tag HTML:

```
<META HTTP-EQUIV="refresh" CONTENT="0;url=http://childporn.site">
```

Altro metodo è quello di utilizzare qualche funzione javascript non documentata nei browser.

Il caso Tumblr

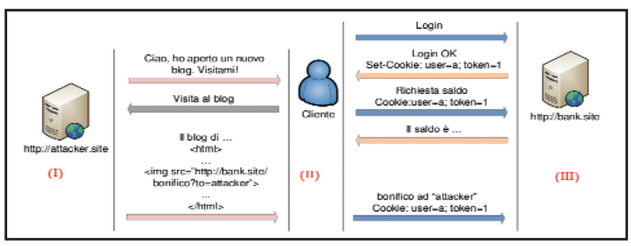
Un esempio pratico è il caso "*Tumblr worm*", un Proof of Concept ideato dal ricercatore italiano *Matteo Carli*. (***www.matteocarli.com/2008/04/tumblr-worm-quando-i-blog-diventano-virali.html***). Sfruttando una vulnerabilità CSRF presente nella dashboard che gestisce i video di Tumblr, era possibile creare un worm in grado di autoreplicarsi attraverso le pagine dei microblog Tumblr (***www.tumblr.com***). I video venivano caricati in un sotto dominio dedicato solo se il referer conteneva la stringa "**www.tumblr. com/new/**" o era completamente vuoto. In questo caso le soluzioni per eludere i controlli erano due: modificare il referer per far credere che la richiesta provenisse dal sito ufficiale o svuotarlo del tutto. Visto che la richiesta era di tipo **POST**, il ricercatore ha optato per la seconda sfruttando uno script javascript che gli permettesse di creare un documento HTML ex-novo al volo.

```
<iframe src="javascript:'<html><body onload=document.forms[0].submit();><form>....</form></body></html>'";></iframe>
```

Per far scattare l'infezione era sufficiente far visitare ad un utente Tumblr, con sessione attiva, una pagina creata ad hoc contenente il codice del worm. Il browser avrebbe interpretato il codice e inviato una richiesta CSRF verso la dashboard di Tumblr creando un nuovo post contenente il codice del worm.

Un rimedio

Gli attacchi CSRF sono un vero problema per gli utenti finali, i quali non possono difendersi facilmente. Nemmeno i browser più moderni forniscono strumenti per mitigare il problema. Un modo semplice per ridurre l'impatto di questo genere di attacchi è l'utilizzo di due browser Web distinti. Il primo utilizzato per la normale navigazione quotidiana e il secondo per la gestione di operazioni che richiedono il login su domini Web "critici". Il logout è uno strumento importantissimo, e va utilizzato non appena si termina il lavoro con il proprio account. Per chi non lo conoscesse, *NoScript* (disponibile nel CD allegato alla rivista), è un plug-in per Firefox che permette di definire delle policy per l'esecuzione di codice Javscript/Flash/Java solamente da domini ritenuti sicuri dall'utente. In più fornisce un filtro contro attacchi XSS.



Attacco CSRF: il cliente (II) di una banca (III) può favorire un server hacker (I)