Web Application Security: il caso #Hack5Stelle

Date : 4 gennaio 2018

Introduzione

La sicurezza delle applicazioni Web è una delle caratteristiche su cui nel nostro Paese si investe meno e male. La *web application security* è una branca della sicurezza delle informazioni che si occupa della sicurezza di siti Web, delle applicazioni e di tutti i servizi che sono offerti sul Web stesso.

Il Web 2.0 è, da un lato, nuova visione delle tecnologie per la costruzione di siti, dall'altro è veicolo massiccio di informazioni trasmesse. Le differenti visioni dell'insieme delle tecnologie e degli approcci di paradigmi usati per la costruzione del Web, fanno sì che dalle semplici pagine scritte in HTML si passi a interi siti web progettati in linguaggi più articolati come il PHP, Javascript o AJAX. Inoltre il web 2.0 ha portato con sé soprattutto l'aumento esponenziale della condivisione delle informazioni. I social network, l'uso sempre più massiccio del web in ambito commerciale B2B, Business to Business, o B2C, Business to Consumer, hanno dato il via ad una quantità di dati e di informazioni che viaggiano e si propagano sul web.

Numerose sono le aziende, le scuole, gli enti, i movimenti che usano il web per i loro scopi ma, purtroppo, non hanno a riguardo quello che in gergo si chiama *atteggiamento proattivo alla sicurezza* delle infrastrutture.

Affidarsi a professionisti del settore, studiare in modo dettagliato i protocolli e i paradigmi di funzionamento dei sistemi web sono elementi indispensabili per avere coscienza delle vulnerabilità dei propri sistemi e dei rischi connessi nell'effettuare scelte progettuali differenti. Analisi, progettazione e ingegnerizzazione di sistemi informatici sul web devono avere come modello approcci ex-ante caratteristici di una individuazione a priori dei problemi.

Purtroppo gli interessati (imprese, istituti d'istruzione, enti, movimenti) non svolgono questo genere di attività e molto raramente adottano l'approccio corretto, rendendo inefficaci gli sforzi di sicurezza sulle applicazioni web.

Le principali vulnerabilità su cui porre attenzione nella sicurezza di sistemi web sono:

- **Cross-site scripting (XSS)**, vulnerabilità dei siti web dinamici che non adottano un controllo adeguato sugli elementi di input nei form e che, quindi, lasciano aperta una possibilità di inserire, e far eseguire, del codice lato client per scopi non previsti dall'applicazione web. In questo modo si possono visualizzare, raccogliere e perfino modificare dati presenti sui server.
- **Sql Injection**, tecnica attraverso la quale sono forzatamente iniettati comandi di codice SQL malevolo all'interno di elementi di input non controllati del web, così da avere accesso ai contenuti dei database aziendali e di poter dare comandi come se ci si trovasse nella *console* del server stesso.
- **Password inadeguate**, deboli o addirittura lasciate impostate a quelle predefinite.
- **Sessioni utente errate**. Essendo HTTP un protocollo *stateless*, la gestione errata delle sessioni e dei cookie può creare innumerevoli problemi nella sicurezza dei sistemi web.
- **Mancanza di CAPTCHA** (Completely Automated Public Turing test to tell Computers and Humans Apart), cioè non avere un modo per effettuare il test (di Turing pubblico e completamente automatico) per distinguere se chi sta usando quell'interfaccia web è un *bot* (programma di scansione automatica della rete) oppure un essere umano. Una misura di sicurezza che consiste in un test formato da una o più domande a risposta tipicamente grafica e/o visiva, per evitare attacchi di tipo (Distributed) Denial of Service (DDOS).
- **Mancato aggiornamento** dei sistemi operativi, dei servizi e delle applicazioni, che devono essere sempre il più possibile tenuti aggiornati alle ultime versioni perché molto spesso questi aggiornamenti riguardano la sicurezza e non vanno trascurati (vedi ad es. il recente caso del ransomware *WannaCry*).

Caso di studio: #Hack5Stelle

Il caso più significativo di quanto importante sia una efficace progettazione delle piattaforme online e di quanto diventi pericolosa per la privacy degli utenti una visione superficiale della web application security è quello che va sotto il nome di #Hack5Stelle.

Un esempio, e un caso di studio, illuminante sotto molti aspetti: sia dal punto di vista squisitamente tecnico che in previsione dell'adozione nel nostro Paese del nuovo regolamento generale sulla protezione dei dati personali, il GDPR (Regolamento UE 2016/679) che avrà piena efficacia in Italia dal 25 maggio 2018.

Nota doverosa di premessa: questo articolo non ha nessuna pretesa di essere dettagliato nella web application security del caso di studio proposto. Tantomeno non vi è alcuna pretesa politica o strumentale. Si vuole, invece, ripercorrere in modo critico una serie interminabile di errori e di cattiva gestione che hanno portato ad un caso che sicuramente farà scuola per la gestione della sicurezza della applicazioni sul web, anche in previsione del citato GDPR.

Tutto ha inizio sui *social media* quando alcuni utenti fanno notare delle mancanze estremamente gravi sulla piattaforma "Rousseau" del Movimento 5 Stelle e su alcuni siti vicino al Movimento stesso già all'inizio del 2017.

In prima analisi si evince che il sito del blog del Movimento ha un certificato SSL (Secure

Sockets Layer) non aggiornato o con un basso livello di rating. L'anomalia su tale certificato, indispensabile per rendere sicuro il sovrastante livello applicativo del protocollo HTTPS (accesso web con autenticazione) su cui si basa l'accesso da parte del browser utente, rende il sito web altamente vulnerabile. In poche parole il sito adotta un basso livello di crittografia, esponendosi ad un attacco di tipo "Man In The Middle" (MITM), letteralmente "uomo nel mezzo". In tale tipo di attacco le due parti di una connessione *credono* di parlare direttamente tra di loro, mentre in realtà c'è una terza parte nascosta che si interpone, intercettando e alterando il contenuto dei messaggi scambiati. Alla data di scrittura di questo articolo il sito di www.beppegrillo.it viene identificato dai browser come sito non sicuro (Figura 1).

Figura 1 - Segnalazione del browser all'acceso del sito www.beppegrillo.it

Questa segnalazione di connessione non protetta può esporre la comunicazione anche un altro tipo di attacco hacker denominato POODLE (Padding Oracle On Downgraded Legacy Encryption). Quest'ultimo colpisce soprattutto la versione 3.0 del protocollo SSL (SSLv3), versione non più utilizzata in quanto vecchia di 15 anni. Questa tecnica attacca e danneggia il lato client e non il lato server, consente infatti di decifrare i cookie, corrispondenti ad esempio ai servizi come Twitter e Google, permettendo di entrare negli account degli utenti, senza il bisogno di conoscerne la password di accesso.

Tutto questo è stato palesato e reso pubblico da ricercatori sui social attraverso *Qualys SSL Labs*, lo strumento di analisi sul controllo della sicurezza e configurazione di siti protetti, liberamente accessibile dal sito online: https://www.ssllabs.com/ssltest/.

Inserendo l'indirizzo da verificare, nel nostro caso https://www.beppegrillo.it, si ottiene il seguente esito descritto nella Figura 2.

Figura 2 - Report SSL www.beppegrillo.it

Inutile commentare il ranking di sicurezza del sito web (F), di come il server abbia protocolli di sicurezza obsoleti e la dicitura di come il sistema sia esposto ad attacchi hacker, tra cui spicca il citato POODLE. Ad un occhio inesperto il risultato ottenuto potrebbe non sembrare molto significativo. Ma è di facile intuizione che un indice basso di classificazione relativo alla sicurezza è una esposizione pericolosa dei contenuti sul web.

Stessa situazione di insicurezza all'analisi per l'altro sito collegato al gruppo del Movimento 5 Stelle: https://www.movimento5stelle.it, con la segnalazione del browser illustrata in Figura 3 e l'analisi del report sul certificato digitale (rating C) in Figura 4.

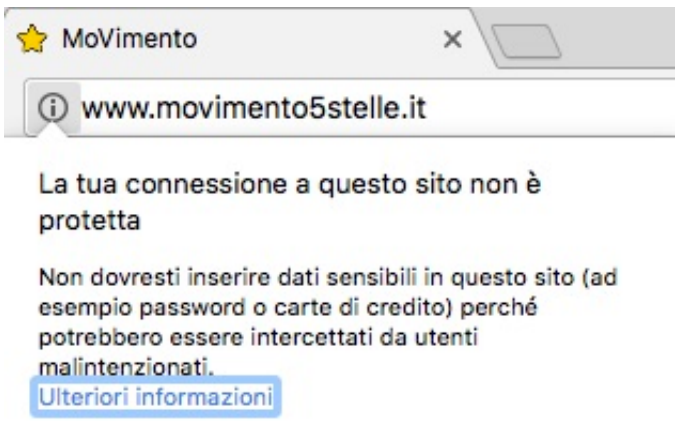

Figura 3 - Segnalazione del browser all’acceso del sito www.movimento5stelle.it

Figura 4 - Report SSL www.movimento5stelle.it

L’analisi del test di affidabilità effettuato sul sito web della piattaforma di voting “sistema operativo Rousseau” https://rousseau.movimento5stelle.it/main.php fornisce un esito ben diverso con elementi dichiarati significanti di sicurezza che, tuttavia, non hanno garantito una corretta *web application security* anzi, come vedremo a breve, è proprio su questo sito web (classificato A+) che si riscontrano le vulnerabilità maggiori.

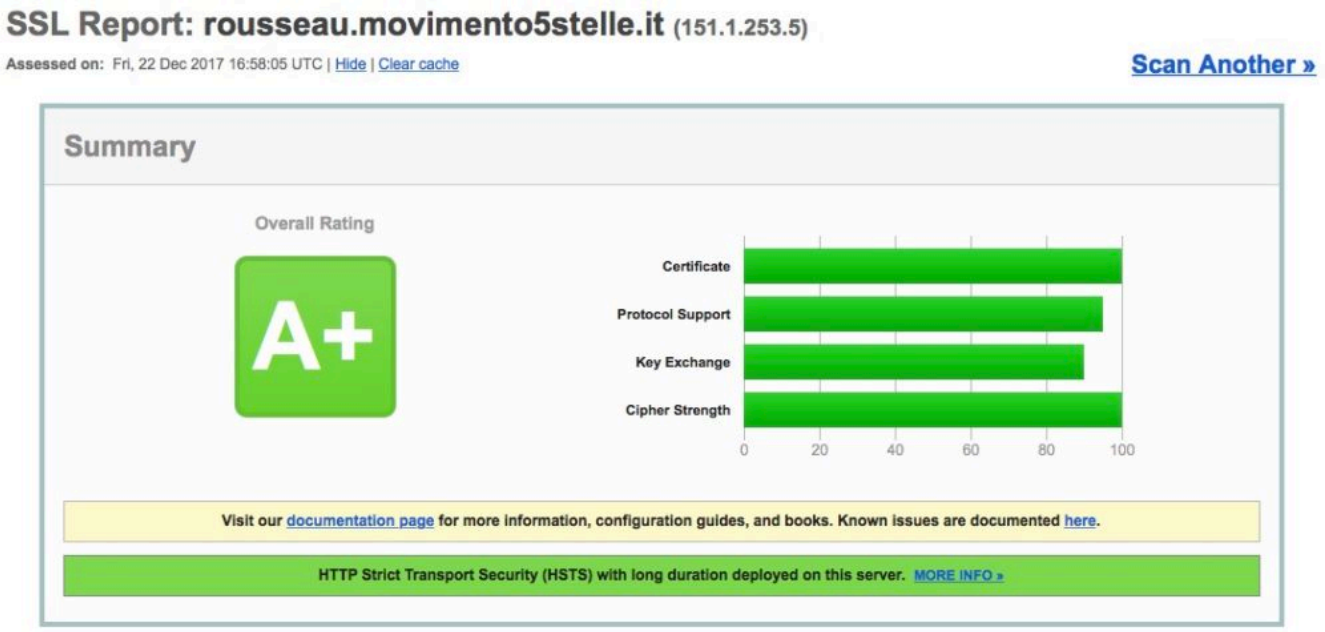

Figura 5 - Report SSL rousseau.movimento5stelle.it

Nonostante un movimento spontaneo di segnalazioni su possibili sicurezze sterili dei server, tutto tace tranquillamente fino a metà anno, prima delle elezioni del candidato premier del Movimento 5 Stelle. In questo periodo Evariste Gal0is, un *WhiteHat* (ricercatore che cerca vulnerabilità nei siti web con lo scopo di aiutare e di migliorare la vita degli utenti e di conseguenza la loro privacy) segnala pubblicamente su un sito web creato appositamente [2] la necessità di proteggere i dati e titola il blog: #Hack5Stelle.

Evariste Gal0is non fornisce alcuna informazione pratica su come sfruttare la vulnerabilità scoperta, ma indica le problematiche di sicurezza cui è soggetto il sito web, fornendo indicazioni sulla tipologia della vulnerabilità (classificandola come una “SQL Injection”) e, nella prima versione del suo blog, cercando anche di spiegare perché tale tipo di vulnerabilità non sia affatto da sottovalutare.

Un attacco di tipo SQL injection ha come obiettivo quello di interagire in modo non corretto e imprevedibile con il database di un’applicazione web. Questo significa poter aver accesso ai dati presenti nel database in un modo che chi ha scritto l’applicazione non si aspettava. Ad esempio, non essendovi sufficienti controlli sui valori e caratteri immessi in una variabile di input “vulnerabile”, è possibile inviare dei comandi al sito web e ottenere delle risposte dal database, senza avere i privilegi necessari. Questo tipo di attacco molto comune denota che chi ha scritto l’interfaccia non ha usato i criteri di buona programmazione per evitare tali anomalie. L’SQL injection è un campanello di allarme molto importante, perché di solito non esiste un’unica variabile vulnerabile all’attacco. Se questa vulnerabilità viene individuata all’interno di un’applicazione, molto probabilmente la si potrà incontrare ancora in altre sezioni dello stesso sito web. Una volta ricevuto l’attacco, o segnalazione di vulnerabilità, è buona pratica controllare se esistono altre defezioni di progettazione dello stesso tipo nelle altre applicazioni in

esecuzione sul sito.

Dopo qualche ora il sito web e l'account Twitter di Evariste Gal0is vengono disattivati per sua stessa volontà, perché non desidera così tanta notorietà. Il suo sito web riceve un sovraccarico di connessioni. Sulla rete si amplifica la polemica. Vale la pena leggere un puntuale ed esauriente riassunto dell'evoluzione #Hack5Stelle ad opera di Fabrizio Carimati [1].

Varie sono state le prese di posizione dei responsabili dei siti web in oggetto, ma l'analisi di questo aspetto non è assolutamente lo scopo di articolo, che assumerebbe una connotazione politica e non tecnica. Ci preme comunque notare che l'affermazione: "*l'attacco non è avvenuto durante le votazioni*" [4] (3 agosto 2017 ore 18:44) è difficile da sostenere dal punto di vista tecnico, soprattutto visti gli sviluppi seguenti.

Un altro esperto di sicurezza attraverso il suo account Twitter "@R0gue_0" adotta un sistema di pubblicazione delle vulnerabilità totalmente diverso da quello sobrio di Evariste Gal0is. Nella stessa giornata del 3 agosto 2017 alle ore 22:40 egli dichiara non solo di aver avuto accesso al database, ma anche ai loro server. Inoltre informa di averlo fatto costantemente da diverso tempo e, per non essere smentito, fornisce dati estratti dal database (sia dal quello di www.beppegrillo.it sia da rousseau.movimento5stelle.it) come prova.

La parte più importante della segnalazione delle vulnerabilità del *BlackHat* R0gue_0 è che non solo ha avuto possibilità di leggere (dump) l'intero database con le relative password, ma ha anche immesso dati all'interno di esso, senza che nessuno degli amministratori del sito web se ne rendesse conto.

Tale evento è gravissimo, tenendo presente che lo scopo della piattaforma è quello di sviluppare e rendere trasparente il voto elettronico all'interno del Movimento e, come auspicano gli utenti del Movimento, in Italia. Infine il *BlackHat* mette in vendita l'intero database con nomi, cognomi, password, ammontare delle donazioni libere fatte al Movimento 5 Stelle, identità digitale, email, e, dulcis in fundo, anche ogni singolo voto espresso dai singoli utenti all'interno della piattaforma stessa.

Il *dump* (copia) del database delle password, rilasciato come prova da Rogue_0, mostra due situazioni apparentemente differenti: le password del blog sono salvate in chiaro, cioè apertamente consultabili, mentre le password conservate sul database del "sistema operativo Rousseau" *rigorosamente* cifrate.

Ma un'analisi semplicissima pubblicata su vari blog [2] [3], effettuata dai due esperti di informatica, Antonio Sanso e Evariste Gal0is, mostra come il sito web "Rousseau" abbia adottato come sistema di gestione dei contenuti (CMS, Content Management System) *MovableType*, con una vecchia versione della funzione di cifratura delle password degli utenti che implementa l'ormai obsoleto DES semplice. Tutto questo si traduce nella possibilità di ottenere la conversione delle password cifrate nella relativa forma in chiaro in meno di 12 secondi.

Nonostante il tentativo dei gestori del sito di innalzare il livello di sicurezza utilizzando un

sistema di tipo *two-way authentication* via SMS, l'hacker Rogue_0 dimostra come abbia avuto accesso a svariati profili del "sistema operativo Rousseau". Molto probabilmente, sfruttando la tecnica del *session riding* (cioè una strategia di cyber attack in cui l'utente, correttamente riconosciuto e validato nell'accesso dal sistema di sicurezza a due fattori, una volta instaurata una sessione, riesce a cambiare la propria identità senza che il server se ne accorga). Inoltre Rogue_0 pubblica vari post a sua firma sul blog del Movimento 5 Stelle stesso [8].

Le analisi di come sia stato possibile sfruttare una serie di vulnerabilità esula dallo scopo dell'articolo. Per chi volesse comunque approfondire tutte le evoluzioni di quello che va sotto il nome #Hack5Stelle, il web è ricco di informazioni, soprattutto di persone competenti che ne parlano da tecnici e non da politici [5] [6] [7].

Una serie di situazioni difficili da gestire in cui si sono rincorse analisi, sfide e comunicati stampa. Ma a farne le spese, come sempre, sono stati gli utenti. Un esponente del Movimento 5 Stelle dichiara su La Repubblica del 6 agosto in un articolo a firma di Annalisa Cuzzocrea e Mauro Favale: "*Il problema non siamo noi ma la sicurezza informatica di questo Paese*" (!). Il problema, purtroppo, è che ne va dell'immagine del nostro Paese composto, invece, da professionisti seri che fanno bene il loro lavoro.

Conclusioni

Quali considerazioni si possono trarre da questo evento?

È importante non lasciare cadere l'accaduto nel dimenticatoio. In Italia c'è la necessità di un approccio più ingegneristico e tecnico delle aziende, istituzioni ed enti che mettono online i loro servizi.

Non sono sufficienti controlli generici di *penetration testing* rivolti solo al sistema azienda nel suo complesso. Importante invece è effettuare analisi sistematiche per testare e rilevare le eventuali vulnerabilità, test dedicati a ciascuna singola applicazione software e anche (e forse soprattutto) alle applicazioni progettate per il web.

Tale attività di testing non deve essere naturalmente *esaustiva*, non deve cioè cercare necessariamente tutte le falle e le vulnerabilità dell'applicazione web ma, al contrario, individuare precisamente quali sono quelle più urgenti e critiche a cui dare risposta. Un approccio basato sul calcolo del rischio e sulle verifiche delle vulnerabilità più pericolose, un approccio che sostanzialmente richiama quelle che sono le linee guida del GDPR sulla sicurezza IT in generale.

Sitografia

(siti web con ultimo accesso 22 dicembre 2017)

1. Fabrizio Carimati - https://www.clodo.it/blog/hack5stelle-riassunto/
2. Evariste Gal0is - http://hack5stelle.byethost17.com/
3. Antonio Sanso - http://blog.intothesymmetry.com/2017/08/analisi-dei-dump-di-rousseau-movimento.html?m=1
4. http://www.ilblogdellestelle.it/la_sicurezza_di_rousseau.html
5. David Puente - https://www.davidpuente.it/blog/tag/hack5stelle/
6. Paolo Attivissimo - http://attivissimo.blogspot.it/2017/08/due-parole-sule-vulnerabilita-di.html
7. Matteo Flora – https://www.youtube.com/user/mgpfvideo/search?query=5+stelle
8. Sito web ufficiale M5S - www.movimento5stelle.it

A cura di: **Crescenzio Gallo**, **Michelangelo De Bonis**, **Michele Perilli**
Università di Foggia