Descrizione del sistema e Analisi del Rischio

Nicolò Marchi Alessandro Gottoli Mattia Peretti

21 giugno 2012

Indice

Elenco delle figure

1 Descrizione del Sistema

1.1 Panoramica del sistema

L'assegnamento per il laboratorio di Sicurezza delle Reti consisteva nell'implementazione di una Certification Authority (in seguito, CA) riguardante una fittizia compagnia di nome *iMovies*, che vuole offrire ai suoi clienti dei servizi basati su PKI (Public Key Infrastructure). Le direttive per l'implementazione di tale CA sono descritte dal libro di testo[1] adottato.

—Possibile descrizione PKI

La nostra implementazione di iMovies permette agli utenti (già inseriti nel database fornito) la creazione e la firma di certificati che verranno poi usati per la comunicazione sicura tramite e-mail.

L'architettura del sistema è composta da due parti:

- **serverIMovies** - una macchina virtuale a 64 bit con sistema operativo *Ubuntu Server 12.04 server* provvista del contenitore servlet open source *Apache Tomcat 7.0.26*.
- **clientIMovies** - una macchina virtuale a 64 bit con sistema operativo *Ubuntu Server 12.04 client* con doppia utenza (administrator e client).

La web application è stata scritta utilizzando il framework Java Server Faces (JSF) per permettere una maggiore attenzione al backend Java attraverso una più facile implementazione del frontend grafico composto da pagine xhtml (create utilizzando le librerie di componenti grafici *Primefaces*[2].).

Per la gestione del database ci si è affidati al Relational Database Management System MySql.

Per quanto riguarda la creazione, la firma, la revoca, e tutte le operazioni di gestione dei certificati ci si è affidati al toolkit *OpenSSL*[3].

Infine, per gli archivi di backup dei dati è stato usato il semplice tool *tar*, presente in ogni distribuzione Unix. Per lo scheduling dei backup è stato usato il tool *cron*, e per il download dei backup ci siamo affidati a

1.2 Funzionalità del Sistema

Login

Il sistema offre principalmente due possibilità di login. Una possibilità consiste nel connettersi al portale attraverso un certificato PKCS#12 riconosciuto dalla CA iMovies, che permette di bypassare il controllo delle credenziali nel database (dato che si presume che il certificato sia in mano al proprietario dello stesso).

La seconda modalità consiste in un canonico form di login nel quale l'utente è chiamato ad inserire lo username e la password. Per quest'ultimo campo, verrà

[1] "Applied Information Security" di David Basin, Patrick Schaller e Micheal Schläpfer.
[2] Per maggiori informazioni, http://www.primefaces.org
[3] Implementazione open-source dei protocolli SSL/TLS.

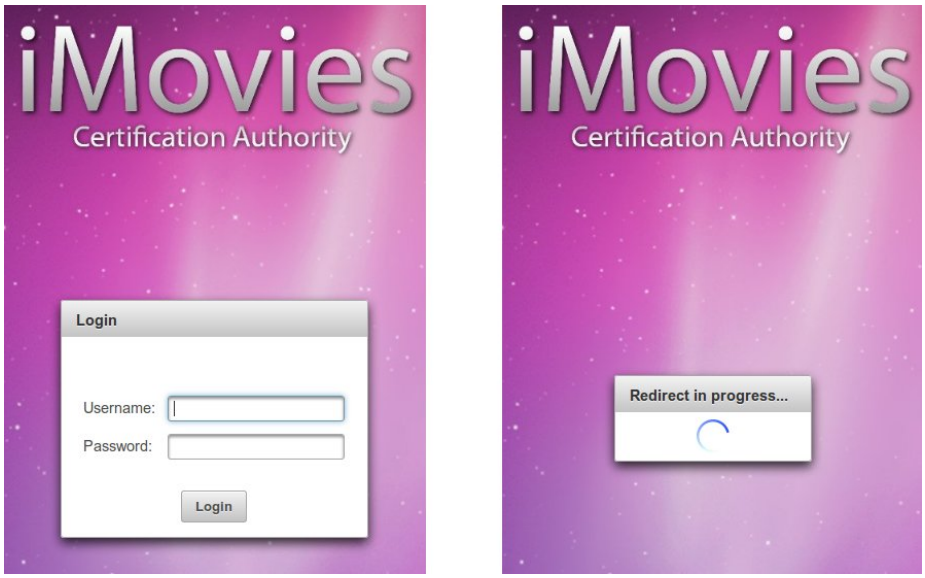


FIGURA 1: Differenza tra il login che viene mostrato al cliente (immagine a sinistra) e il login automatico tramite certificato per l'amministratore.

calcolato il corrispondente hash SHA-1 e tale valore sarà utilizzato nel confronto con gli hash delle password salvati nel database.

Modifica informazioni personali

FIGURA 2: Le tabella contenente le informazioni personali modificabili dell'utente *a3*.

Il portale offre agli utenti la possibilità di modificare le informazioni personali precedentemente salvate nel database. Si possono modificare tutti i campi, ad eccezione del campo username, che è fisso e ha il ruolo di primary key nella tabella relativa agli utenti nel database.

Rilascio di certificati

Ad ogni utente viene fornita la possibilità di creare certificati. Alla creazione di un certificato viene generata una chiave privata con crittografia a 4096 bit e crittata con DES3 e una password inserita dall'utente. Dopodiché viene generato e firmato il certificato relativo, con i dati dell'utente salvati nel database.

Revoca dei certificati

Nella sezione di management dei certificati viene fornita la possibilità di revocare selettivamente i certificati dell'utente. Quando un certificato viene revocato, viene generata nuovamente la Certificate Revocation List della Certificate Authority.

Download dei certificati

Viene fornita la possibilità di scaricare i certificati e le relative chiavi private in formato PKCS#12. Quando si richiede il download del certificato il sistema richiederà all'utente la password usata durante la creazione della chiave privata, e una nuova password che sarà usata per l'esportazione del certificato PKCS#12. QUest'ultima password dovrà essere inserita quando si importerà il certificato all'interno di un browser.

Eliminazione dei certificati

Name Of Certificate	Serial	Validity	Expiration Date	Revocation Date	Revoke	Download	Delete
03.pem	3	R	2013/06/03 16:25:11	2012/06/04 16:16:26	Revoke Certificate	Download Certificate	Delete Certificate
4D.pem	4d	R	2013/06/09 13:54:30	2012/06/16 15:10:17	Revoke Certificate	Download Certificate	Delete Certificate
4E.pem	4e	R	2013/06/09 14:02:20	2012/06/16 15:12:44	Revoke Certificate	Download Certificate	Delete Certificate
52.pem	52	V	2013/06/11 14:11:58	Not Revoked	Revoke Certificate	Download Certificate	Delete Certificate

FIGURA 3: Tabella per il download, la revoca e l'eliminazione dei certificati.

Quando un utente sceglie di rimuovere un certificato, innanzi tutto quest'ultimo verrà revocato; dopodiché verrà eliminata la chiave privata associata al certificato, e il certificato stesso.

Amministrazione del portale

L'amministratore del portale accede al frontend di amministrazione solamente con un certificato PKCS#12 già in suo possesso. Attraverso le pagine dell'area amministrativa, l'amministratore può vedere quanti e quali certificati sono stati

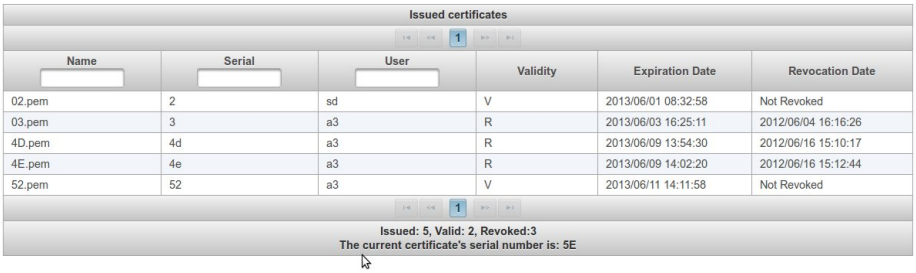


FIGURA 4: Tabella per la visualizzazione di tutti i certificati rilasciati con contatori e prossimo serial number.

rilasciati, quanti e quali certificati sono stati revocati, e il valore corrente del serial number[4].

Inoltre viene effettuato un log di tutti gli accessi al sito, compresi gli accessi effettuati passando attraverso le backdoor.

Backup dei dati

Il sistema esegue un periodico backup di tutte le chiavi private e di tutti i certificati. I backup sono in realtà due, uno totale che viene eseguito ogni settimana il venerdì alle ore 11.00, mentre un backup incrementale che viene eseguito tutte le ore al minuto 40. In entrambi i casi viene generato un archivio con il comando Unix "tar".

Vi è poi un server ftp che permette ad un amministratore di scaricare da remoto i backups. L'accesso da ftp è limitato solamente alla cartella ove vi sono i backups.

1.3 Componenti e Sottosistemi

Java Server Faces

Come già accennato, per l'implementazione del portale è stata usata la tecnologia di Java Server Faces. JSF, acronimo di Java Server Faces può essere considerata un framework per lo sviluppo di web application basate su Java. É basato sul design pattern architetturale Model-View-Controller (MVC) ed è descritto da un documento di specifiche (JSR 127) alla cui stesura hanno partecipato aziende quali IBM, Oracle Corporation, Siemens e Sun Microsystems. Il suo scopo è di semplificare lo sviluppo dell'interfaccia utente (UI) di una applicazione Web.

A grandi linee il funzionamento del framework JSF si basa su un file di configurazione XML (`faces-config.xml`) in cui vengono definite le viste (sostanzialmente pagine JSP che sfruttano la *taglibrary faces*) e i controllori. Le singole implementazioni sfruttano una servlet di base `FacesServlet` o un filtro il cui

[4]il valore del numero esadecimale che verrà assegnato al prossimo certificato generato

mapping è normalmente `/faces/*` o `*.faces`. La `FacesServlet` deve essere registrata nel file XML (`web.xml`) della web application.

PrimeFaces

FIGURA 5: Il logo delle librerie Primefaces.

Le librerie Primefaces costituiscono una serie di componenti grafici utilizzabili all'interno di una web application Jsf. PrimeFaces è una suite open source utilizzabile con il framework Java Server Faces, esplicitamente pensata per realizzare i componenti presentazionali di una applicazione web enterprise: editor HTML, finestre di dialogo, meccanismi per l'auto-completamento, grafici e calendari, drag & drop, integrazione di mappe google e molto altro.

La suite offre supporto sia ad ajax che al rendering parziale delle pagine web, grazie ad una integrazione nativa con `jquery`. L'aspetto grafico dei componenti si basa su `JQuery UI`: è quindi possibile personalizzarlo attraverso lo skin framework *Theme Roller*, o utilizzare un discreto insieme di temi predefiniti. Da segnalare infine la presenza di uno *User Interface Kit* per la realizzazione di applicazioni Web orientate ai dispositivi mobili (iPhone, Android, etc.), di semplice e veloce configurazione.

Macchine Virtuali

Le macchine virtuali create sono due, create con il software open source *VirtualBox*, e sono suddivise in macchina server e macchina client.

La macchina client consiste in un'installazione della distribuzione Linux *Ubuntu 12.04 LTS* per architetture *amd64*. Nella macchina sono presenti due utenti principali:

1. utente *admin*: è l'utente che funge da admin del portale iMovies. Nell'utenza vi è già installato il certificato PKCS#12 relativo all'amministratore che permette il login rapido alla sezione di amministrazione del portale, un client ftp (più precisamente FileZilla) per il download dei backups da remoto.

2. utente *client*: l'utente client consiste invece in un semplice utente, senza grosse particolarità.

La macchina server invece consiste in un'installazione della distribuzione Linux *Ubuntu Server 12.04 LTS* sempre per architetture *amd64*. Al suo interno troviamo installati tutti i servizi utili al mantenimento e all'esecuzione del portale.

Le macchine virtuali si trovano connesse tra loro in una rete interna, con assegnati come indirizzi IP 192.168.1.11 per la macchina server, mentre 192.168.1.10 per la macchina client.

Web Server

Per poter utilizzare le Java Server Faces, abbiamo dovuto scegliere (in maniera quasi obbligata) il web server *Apache Tomcat 7.0*, ultima versione del noto web container. Tomcat è un contenitore servlet open source sviluppato dalla Apache Software Foundation. Implementa le specifiche Java Server Pages (JSP) e Servlet, fornendo quindi una piattaforma per l'esecuzione di applicazioni Web sviluppate nel linguaggio Java. La sua distribuzione standard include anche le funzionalità di web server tradizionale, che corrispondono al prodotto Apache.

test

————————not finished List all system components, subdivided, for example, into categories such as platforms, applications, data records, etc. For each component, state its relevant properties.

1.4 Interfacce

login.xhtml

La pagina di login si presenta molto semplice, con un semplice form di login dove inserire username e password. Dalla pagina di login si passa sempre e comunque, anche se si ha un certificato PKCS#12 riconosciuto. In quest'ultimo caso nella pagina viene effettuato il controllo del certificato e il redirect alla pagina di amministrazione (in caso di certificato dell'admin) o nella pagina dell'user, con i dati dell'utente.

user.xhtml

La pagina consiste in una semplice pagina di benvenuto dove vi è presente un menù di navigazione e un semplice messaggio di benvenuto. Da questa pagina l'utente può muoversi tra tutte le funzionalità del portale.

edit.xhtml

É la pagina

admin.xhtml

Specify all interfaces and information flows, from the technical as well as from the organizational point of view.

1.5 Backdoors

Describe the implemented backdoors. **Do not add this section to the version of your report that is handed over to the team that reviews your system!**

1.6 Materiale Aggiuntivo

You may have additional sections according to your needs.

2 Risk Analysis and Security Measures

2.1 Information Assets

Describe the relevant assets and their required security properties. For example, data objects, access restrictions, configurations, etc.

2.2 Threat Sources

Name and describe potential threat sources.

2.3 Risks and Countermeasures

List all potential threats and the corresponding countermeasures. Estimate the risk based on the information about the threat, the threat sources and the corresponding countermeasure. For this purpose, use the following three tables.

2.3.1 Tools

Una volta effettuata l'implementazione del sistema, sono stati utilizzati differenti tool per il test delle vulnerabilità del sistema.

- **Websecurify**[5]
  Websecurify è una potente applicazione per rilevare velocemente e in modo accurato le vulnerabilità di una web application.

  L'applicazione è disponibile come plugin per browser e software standalone per sistemi Windows, Mac e Linux.

  Data la release prematura disponibile per sistemi Linux è stato deciso di utilizzare il plugin per Google Chrome.

  Il risultato più importante del report generato è quello relativo al nome del server web e della tecnologia utilizzata che viene trasmesso nelle risposte

[5] http://www.websecurify.com

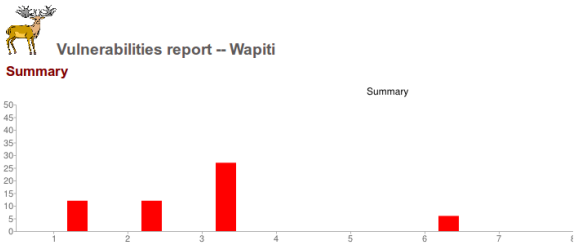
- **Wapiti**[6]
  Wapiti è uno scanner di vulnerabilità di una web application. Non esamina il codice sorgente delle pagine della web application ma si occupa di scansionare le pagine alla ricerca di form o campi di testo attraverso i quali eseguire delle injections. Per questo motivo, è definito come un *fuzzer*[7].

  La release di Wapiti utilizzata per questo test è la 2.2.1. Una volta scaricato l'archivio e decompresso, è stato sufficiente lanciare il comando dalla cartella estratta:

  ```
  python wapiti.py http://www.imovies.org -o report_folder -f html
  ```

  Il software analizza il portale, individua le vulnerabilità e genera un report sotto forma di html all'interno della folder `report_folder`.

  Il report di iMovies è visualizzato nella figura **??**.



	SQL Injection (1)	Blind SQL Injection (2)	File Handling (3)	Cross Site Scripting (4)	CRLF (5)	Commands execution (6)	Resource consumption (7)	Htaccess Bypass (8)	Backup file (9)	Potentially dangerous file (10)
High	12	12	27	0	0	6	0	0	0	0
Medium	0	0	0	0	0	0	0	0	0	0
Low	0	0	0	0	0	0	0	0	0	0

FIGURA 6: Il report generato dal tool Wapiti

  Sono state individuate in tutto 57 vulnerabilità. Ognuna delle vulnerabilità segnala che, ad una richiesta particolare, il server ha risposto con un codice d'errore *500*[8].

- **Skipfish**[9]
  Skipfish è un software ...

  La versione testata è la 2.07b.

  Per il corretto funzionamento dell'applicazione è stato necessario compilare i sorgenti con i seguenti comandi:

[6] http://www.ict-romulus.eu/web/wapiti/home

[7] Un fuzzer è un software che sfrutta il fuzzing. Quest'ultimo è una tecnica di testing, automatica o semiautomatica, attraverso la quale vengono inviati input invalidi, inaspettati o casuali ad un programma con lo scopo di trovare vulnerabilità attraverso un monitoring delle risposte dello stesso.

[8] Questo codice d'errore indica un errore generico avvenuto sul server a seguito di una richiesta impossibile da risolvere(Internal server error).

[9] http://code.google.com/p/skipfish/

```
# sudo apt-get install build-essential libssl-dev libdn11-dev
# make
```

Una volta compilato Skipfish, è necessario configurare i dizionari. Skipfish costruisce e mantiene automaticamente dei dizionari basati su URL e sui contenuti HTML incontrati nella scansione di un sito. Questi dizionari sono particolarmente importanti per scansioni successive dello stesso sito.

Una volta apprese le caratteristiche di Skipfish, è stato lanciato il comando:

```
# ./skipfish -S dictionaries/complete.wl -W dictionaries/new_dict.wl -o
report_skipfish https://www.imovies.com
```

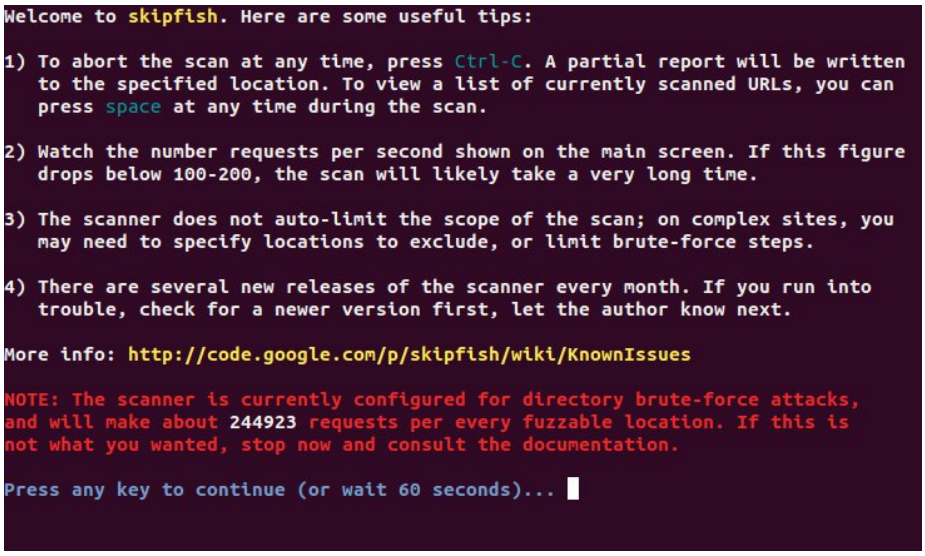


FIGURA 7: La schermata di benvenuto con le prime informazioni sugli attacchi che saranno effettuati.

Tale comando, esegue Skipfish utilizzando un dizionario completo per attacchi di forza bruta a directory, form, campi di testo e parametri della web application. Il dizionario `new_dict.wl` è stato utilizzato come dizionario nel quale inserire parole chiave riguardandi il sito target. Di seguito, il contenuto del file relativo al dizionario dopo la serie di computazioni eseguite da Skipfish.

```
w?  3 2 0 iMovies
w?  3 2 0 jquery
w?  3 2 0 filters
w?  3 2 0 webapps
w?  1 0 0 54
w?  2 1 0 callback
w?  3 2 0 primefaces
w?  3 2 0 jsessionid
w?  2 1 0 tabview
w?  3 2 0 ln
```

```
w?  3 2 0 CSRF_NONCE
w?  2 2 1 157
w?  1 0 0 mousewheel
w?  1 0 0 paginator
w?  1 0 0 lightbox
w?  1 0 0 accordion
w?  1 0 0 jmxproxy
w?  2 1 0 treetable
w?  1 0 0 dot_clear
w?  1 0 0 password-meter
w?  2 2 0 manager-howto
w?  1 0 0 expired
w?  3 2 0 host-manager
w?  3 2 0 RUNNING
w?  2 2 1 27
w?  2 1 0 slider
w?  3 2 0 tomcat-users
w?  2 1 0 datatable
w?  3 2 0 catalina
w?  3 2 0 ROOT
w?  3 2 0 tomcat7
w?  1 0 0 236
w?  1 0 0 153
w?  2 1 0 spinner
w?  2 1 0 resizable
w?  2 2 1 43
w?  2 1 0 rating
w?  3 2 0 primefaces-aristo
w?  3 2 0 ajax-loader
w?  2 1 0 orderlist
w?  3 2 0 tomcat7-common
w?  2 1 0 effect
w?  2 2 1 178
w?  1 0 0 notificationbar
```

Le parole chiave inserite nel dizionario sono state mantenute secondo il seguente schema: `type hits total_age last_age keyword`.
La colonna `type` indica il tipo di parola inserita nel vocabolario: `w` indica una keyword mentre `g` indica un'estensione di file. Questa colonna è seguita da una lettera che indica se la parola chiave è specifica della tecnologia utilizzata(`s`) oppure generica(`g`). In questo caso, il `?` è del tutto equivalente al letterale `g`. La colonna `hits` indica il numero totale di volte che tale parola è stata individuata senza produrre un errore 404 (pagina non trovata, gestito da iMovies con un redirect). La colonna `total_age` indica il numero totale di cicli di scansione eseguiti sulla parola chiave. Infine, la colonna `last_age` indica il numero totale di cicli di scansione a partire dall'ultimo 'hit'.

È facilmente intuibile che le parole chiave inserite riguardano solo il sito target e, si riesce a notare, come il software esegua un test non solo sul percorso indicato come parametro (`https://www.imovies.org` è puramente indicativo, è stato

utilizzato un percorso composto da indirizzo ip, numero di porta e percorso dell'applicazione) ma anche sull'intero host (troviamo parole chiave come `ROOT`, `tomcat7` o `catalina` che riguardano esclusivamente l'ambiente tomcat). Infatti, tale tool, esegue una scansione sull'intero host, inclusi i servizi presenti su altre porte.

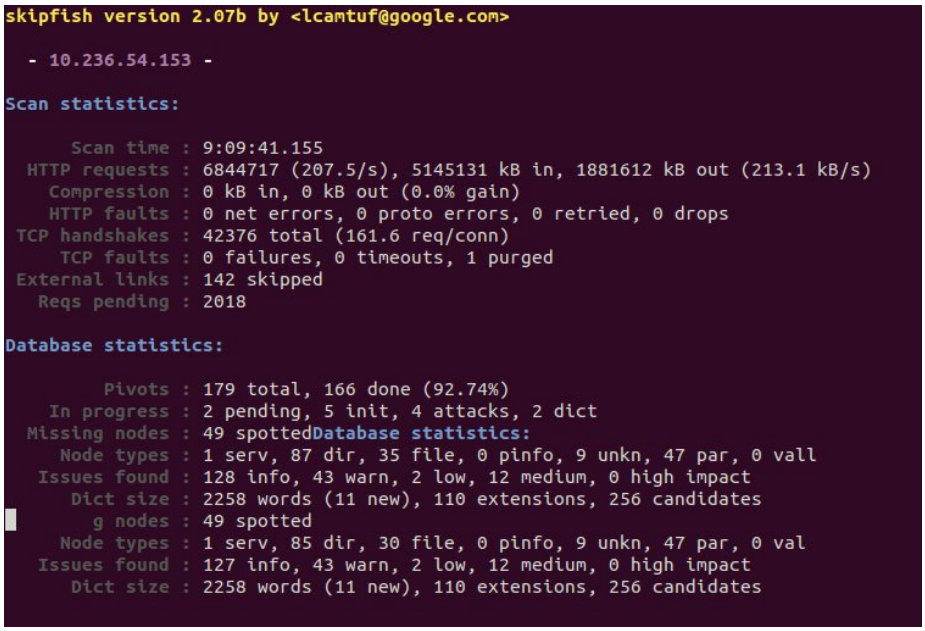


FIGURA 8: L'esecuzione del tool Skipfish al 92.74% con un tempo di scansione di poco superiore alle 9 ore.

L'esecuzione del tool è stata un'operazione durata circa 10 ore e, per questo motivo, è stata eseguita di notte.

Il report html generato, in allegato a questa relazione, è accessibile attraverso il file `index.html`.

Le considerazioni finali che si possono fare, in merito all'implementazione di iMovies che è stata eseguita e alla luce dei report generati dai vari tool, sono di vario tipo:

1. Prevenzione dell'errore 500 (internal server error).
   Tale errore generico, segnala la presenza di un comportamento anomalo del server, a seguito di una richiesta particolare. Nel nostro caso, l'errore viene generato quando il campo `javax.faces.ViewState` di tipo *hidden* presente nel form di login viene compilato con un valore che Tomcat non è in grado di gestire. Tale campo viene inserito automaticamente dal framework JSF in ogni form per prevenire attacchi del tipo XSS e CSRF. Gli errori di SQL-Injection individuati dai tool si riferiscono quasi esclusivamente a tale campo.

2. Protezione delle directory.
   I report di Skipfish segnalano chiaramente come le risorse della web application siano facilmente rintracciabili. Una protezione accurata delle cartelle contenenti immagini, pagine da includere oppure aree private sarebbe da implementare. Ad esempio sarebbe possibile, conoscendo il percorso, visualizzare il file xhtml che contiene il menu dell'utente o dell'amministratore (file che viene incluso nelle pagine principali della webapp impossibili da interpretare se l'attaccante non è loggato perché dotate di un controllo di *preRender* attraverso i java bean). Chiaramente, una volta ottenuto tale file, si potrebbe conoscere solo che tipo di azioni è in grado di svolgere l'amministratore ma senza il backend costituito dai java bean e la pagina principale, non si è in grado di fare nulla. Per implementare una funzionalità che permetta una rigida protezione delle directory è necessario affidarsi ai *Realms*[10] di Tomcat. L'idea è quella di creare un'implementazione di un realm (estendendo la classe *RealmBase*) che rifletta i requisiti di: accesso tramite credenziali per il cliente, accesso tramite certificato per amministratore e controllo incrociato su database se necessario. In questo modo, specificato il nuovo realm (che potrebbe chiamarsi *iMoviesRealm*) nel file di configurazione di Tomcat `server.xml` si dovrebbe essere in grado di mantenere la stessa funzionalità del sistema con una rigida protezione di files e cartelle.

3. Protezione del percorso nella barra degli indirizzi.
   Nella barra degli indirizzi sono visualizzate troppe informazioni sul percorso in cui la pagina attualmente visualizzata è situata. Eseguendo un forward anziché un redirect verso la pagina successiva, si evita di fornire troppe informazioni all'utente sulla struttura delle directory della webapp.

Impact	
Impact	Description
High	...
Medium	...
Low	...

Likelihood	
Likelihood	Description
High	...
Medium	...
Low	...

Risk Level			
Likelihood	**Impact**		
	Low	Medium	High
High	Low	Medium	High
Medium	Low	Medium	Medium
Low	Low	Low	Low

[10] I *Realms* sono delle strutture interne di Tomcat create ad-hoc per la protezione e gestione di aree ad accesso limitato. I Realms permettono di creare delle aree protette accessibili attraverso un metodo di autenticazione. Esistono differenti tipi di realms (con accesso a database tramite controllo RBAC-like, attraverso server LDAP, ecc.) e differenti tipi di autenticazione (con prompt di username e password in stile .htaccess con hashing della password, con un form definito dall'utente, con user certificate).

2.3.2 *Evaluation Asset X*

Evaluate the likelihood, impact and the resulting risk, after implementation of the corresponding countermeasures.

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
1	...	...	*Low*	*Low*	*Low*
2	...	...	*Medium*	*High*	*Medium*

2.3.3 *Evaluation Asset y*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
1	...	...	*Low*	*Low*	*Low*
2	...	...	*Medium*	*High*	*Medium*

2.3.4 Detailed Description of Selected Countermeasures

Optionally explain the details of the countermeasures mentioned above.

2.3.5 Risk Acceptance

List all medium and high risks, according to the evaluation above. For each risk, propose additional countermeasures that could be implemented to further reduce the risks.

No. of threat	Proposed countermeasure including expected impact
...	...
...	...