

Descrizione del sistema e Analisi del Rischio

Nicolò Marchi

Alessandro Gottoli

Mattia Peretti

21 giugno 2012

Indice

1	Descrizione del Sistema	3
1.1	Panoramica del sistema	3
1.2	Funzionalità del Sistema	3
1.3	Componenti e Sottosistemi	6
1.4	Interfacce	8
1.5	Backdoors	9
1.5.1	Backdoor triviale	9
1.5.2	Backdoor complessa	10
1.6	Materiale Aggiuntivo	10
2	Analisi del Rischio e Misure di Sicurezza	10
2.1	Information Assets	10
2.2	Fonti delle minacce	10
2.3	Rischi e Contromisure	12
2.3.1	Tools	12
2.3.2	Tabelle e descrizione di rischi, probabilità e impatto . . .	14
2.3.3	<i>Valutazione del portale iMovies parte software</i>	15
2.3.4	<i>Valutazione del portale iMovies parte hardware</i>	16
2.3.5	Descrizione dettagliata delle contromisure scelte	17
2.3.6	Risk Acceptance	18

Elenco delle figure

1	Differenza tra il login che viene mostrato al cliente (immagine a sinistra) e il login automatico tramite certificato per l'amministratore.	4
2	Le tabella contenente le informazioni personali modificabili dell'utente <i>a3</i>	4
3	Tabella per il download, la revoca e l'eliminazione dei certificati.	5

4	Tabella per la visualizzazione di tutti i certificati rilasciati con contatori e prossimo serial number.	6
5	Il logo delle librerie Primefaces.	7
6	Il report generato dal tool Wapiti	13

1 Descrizione del Sistema

1.1 Panoramica del sistema

L'assegnamento per il laboratorio di Sicurezza delle Reti consisteva nell'implementazione di una Certificate 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.

Una PKI è una infrastruttura che permette di riconoscere a chi appartengono determinate chiavi pubbliche. In questa infrastruttura vi sarà una Certificate Authority, che firmando certificati certifica che la chiave pubblica *pk* appartiene alla persona *P*. Per firmare questo certificato la CA si occuperà di verificare che la persona che ha richiesto certificazione sia realmente chi dice di essere, dopodiché ne firmerà il certificato. La persona *P* quindi ora può distribuire la propria chiave pubblica certificata.

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. Questi certificati avranno una durata di 6 mesi, con la data di inizio e fine validità che deve essere scelta dall'utente in fase di creazione. Questa scelta è dovuta al fatto che non ci sembrava una scelta ottimale lasciare che si potessero creare o solo un certificato con lo stesso Subject¹, o *n* certificati con Subject uguale e date di validità standard (cioè dal giorno di creazione fino a 365 giorni dopo).

L'architettura del sistema è composta da due parti:

- una macchina virtuale di nome **ServerIMovies**.
- una macchina virtuale di nome **ClientIMovies**.

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 remoto dei backups ci siamo affidati a *FileZilla*

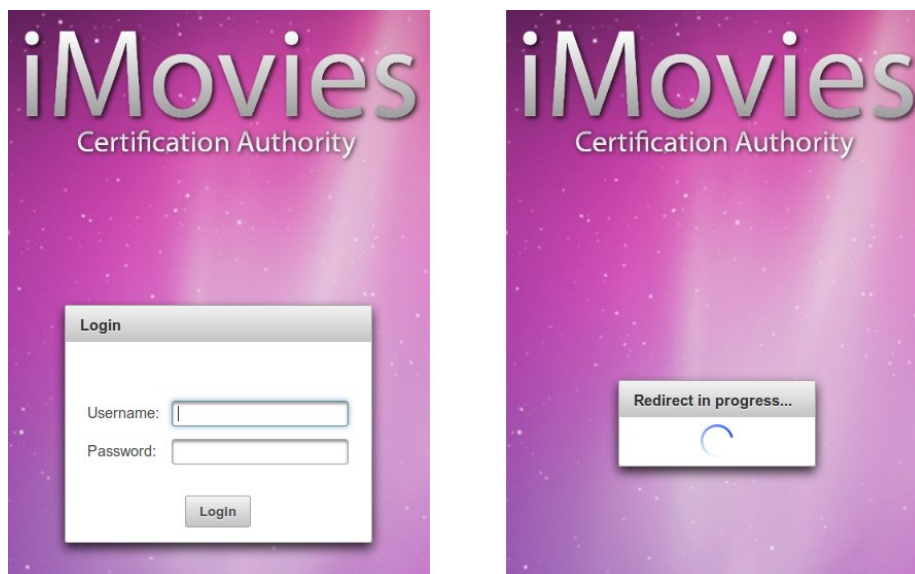


FIGURA 1: Differenza tra il login che viene mostrato al cliente (immagine a sinistra) e il login automatico tramite certificato per l'amministratore.

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

Modifica informazioni personali

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.

¹Il campo Subject identifica l'entità associata alla chiave pubblica memorizzata nel campo Public Key del Subject. Il nome del Subject può essere trasportato nel campo Subject.

²Per maggiori informazioni, <http://www.primefaces.org>

³Implementazione open-source dei protocolli SSL/TLS. <http://www.openssl.org/>

a3's personal information	
Username	a3
First name	<input type="text" value="Andres Alan"/>
Last name	<input type="text" value="Anderson"/>
Email	<input type="text" value="and@iMovies"/>
Once you have done, click one of the buttons below.	

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

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

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

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.

Issued certificates					
<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div>					
Name	Serial	User	Validity	Expiration Date	Revocation Date
02.pem	2	sd	V	2013/06/01 08:32:58	Not Revoked
03.pem	3	a3	R	2013/06/03 16:25:11	2012/06/04 16:16:26
4D.pem	4d	a3	R	2013/06/09 13:54:30	2012/06/16 15:10:17
4E.pem	4e	a3	R	2013/06/09 14:02:20	2012/06/16 15:12:44
52.pem	52	a3	V	2013/06/11 14:11:58	Not Revoked
<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div>					
Issued: 5, Valid: 2, Revoked: 3 The current certificate's serial number is: 5E					

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 number⁴.

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.

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

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. È 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 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

Come già accennato, le macchine virtuali create sono due, installate tramite il software open source *VirtualBox*, e sono suddivise in macchina server e macchina client.

La macchina client si chiama **ClientIMovies**, e consiste in un'installazione della distribuzione Linux *Ubuntu 12.04 LTS* per architetture *amd64*. Le specifiche tecniche della macchina consistono in un hard disk dinamico della dimensione massima di 8 Gigabyte, 1024 MB di RAM e 12 MB di memoria dedicata alla parte grafica. La macchina ha la possibilità di vedere dispositivi USB e di condividere cartelle. Vi è una sola scheda di rete impostata in modo da essere connessa in una rete interna chiamata *INFSEC*. Nella macchina sono presenti due utenti principali:

1. utente *admin*: amministratore del portale iMovies. Nell'utenza 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, e un demone SSH per la connessione remota.
2. utente *client*: rappresenta un cliente del portale iMovies. L'utenza non ha caratteristiche particolari.

La macchina server si chiama **ServerIMovies**, e consiste in un'installazione della distribuzione Linux *Ubuntu Server 12.04 LTS* sempre per architetture *amd64*. Le specifiche tecniche della macchina consistono in un hard disk dinamico della dimensione massima di 8 Gigabyte, 512 MB di RAM e 12 MB di memoria dedicata alla parte grafica. La macchina ha la possibilità di vedere solamente cartelle condivise. Vi è una sola scheda di rete impostata in modo da essere connessa in una rete interna chiamata *INFSEC*. 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.

vsftpd

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 contenente tutte le informazioni dell'utente loggato nel sistema. Da questa pagina è possibile modificare le proprie informazioni personali (ad eccezione dello username), e sulla base di queste informazioni personali generare un certificato, selezionandone la data di inizio e fine, e inserendo una password per la chiave privata.

manageCertificates.xhtml

Questa è la pagina adibita alla gestione dei certificati dell'utente. In questa pagina l'utente può vedere i suoi certificati, anche quelli revocati. Da qui può decidere se cancellare i certificati, revocarli o avviare la procedura di download, che genera il certificato PKCS#12 a partire dalla chiave privata e dal certificato preso in esame.

admin.xhtml

Questa è la sezione dedicata all'amministratore del portale. Da qui l'amministratore può muoversi attraverso un menù, per le varie sezioni dell'area amministrativa.

issued.xhtml

L'amministratore in questa pagina può vedere tutti i certificati creati dagli utenti della CA. Può inoltre visualizzare il numero totale di certificati revocati, e il numero seriale corrente. Da un menù può inoltre muoversi nell'altra sezione dell'amministrazione, che è quella riguardante il controllo degli accessi.

aclog.xhtml

Da questa pagina l'amministratore può vedere in tempo reale gli accessi al sito. Può vedere chi si è connesso, a che ora, e anche chi si è connesso passando attraverso le backdoor. La pagina offre inoltre la possibilità di esportare tutti i dati in formato *.pdf* o *.xls*.

possibile use case

1.5 Backdoors

1.5.1 Backdoor triviale

Questa backdoor consiste nell'avere Apache Tomcat in ascolto su una porta particolare, scelta da noi, e cioè la porta 43567. La porta è rilevabile tramite l'applicazione **nmap** (Network Mapper). Cercando quindi di connettersi alla porta 43567 viene eseguito immediatamente il redirect verso il back-end di amministrazione del portale, dove verrà visualizzata una web shell (*MindTerm*) in un pop-up, connessa in SSH alla macchina server con privilegi di root.

immagine sca nmap (sempre se lo facciamo funzionare)

1.5.2 Backdoor complessa

Questa backdoor consiste nell'implementazione di una semplice password segreta. Citando un film del 1983, "Wargames" diretto da John Badham, la backdoor consiste nell'inserire la parola Joshua nel solo campo password del form di login. La password inserita viene poi trasformata in un hash SHA-1.

All'interno del codice Java, nella classe relativa al controllo delle credenziali passate come input, vi è un campo privato, statico, e costante di nome *magic*. Questo campo contiene l'hash SHA-1 della parola "Joshua". Quando nel form di login viene inserita la password e lo username resta vuoto, viene confrontato lo SHA-1 della parola inserita col campo *magic*. Se il controllo da esito positivo viene fatto il redirect su una pagina contenente un'applet Java che avvia una shell, che connettendosi in SSH con le credenziali di Root, da accesso da super user sulla macchina server.

Riportiamo qui sotto lo spezzone di codice:

```
(...)  
  
private static final String magic="  
    a9a2e8456bf9d58e91fe91cbfe10cad5211216c2";  
  
(...)  
  
if(SHAsum(password.getBytes()).equals(magic)){  
    this.admin=true;
```

```

        log.aclog("backdoor_user", 0);
        adminAccess();
        return;
    }
    (...)

```

1.6 Materiale Aggiuntivo

You may have additional sections according to your needs.

2 Analisi del Rischio e Misure di Sicurezza

2.1 Information Assets

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

2.2 Fonti delle minacce

Le fonti delle minacce sono ciò che può minacciare la sicurezza della CA. Una fonte di minaccia potrebbe essere ad esempio un agente, che vuole qualcosa dalla CA, come ad esempio un certificato. Individuare gli agenti di minaccia è molto importante per poi poter individuare le fonti di minaccia.

Le fonti di minaccia principali per la CA, e anche per la maggior parte dei portali esistenti, si possono suddividere in alcune categorie:

- fonti che non hanno obiettivo specifico: le fonti di minaccia che non hanno un obiettivo specifico possono essere i virus, gli worms, i trojans e le bombe logiche;
- fonti interne: dipendenti, membri dello staff, personale, o chiunque abbia un qualche risentimento verso l'obiettivo;
- fonti criminali: possono essere criminali solitari o associazioni di crimine organizzato, e avranno come obiettivo informazioni di valore per loro e per i loro traffici, come account bancari, carte di credito, o tutto ciò che si può convertire o sfruttare per guadagnare denaro;
- fonte da aziende rivali: aziende che sono impegnate nella guerra delle informazioni o competitive intelligence. I partner e concorrenti rientrano in questa categoria;
- fonte umana non intenzionale: incidenti, negligenza, disattenzione;
- fonte umana intenzionale: Insider, outsider;
- fonte naturale: catastrofi naturali di ogni tipologia, come terremoti, incendi, alluvioni, ecc.

Tenendo conto di tutte queste possibili fonti di minaccia, abbiamo stilato una lista di possibili agenti che possono attaccare o rappresentare fonte di minaccia per la nostra CA. Sono stati elencati in ordine dal più probabile al meno probabile, e sono:

1. L'attaccante motivato: rappresenta il pericolo maggiore; è un attaccante senza scrupoli che vuole a tutti i costi qualcosa dalla Certificate Authority, un qualcosa che non può avere in via "legale". Potrebbe essere un ex-dipendente arrabbiato, o un hacker pagato appositamente da qualcuno, o da qualche organizzazione criminale, per penetrare il sistema.
2. Malware automatico: programma o script, che è alla ricerca di vulnerabilità note, che poi riportano segnalazioni ad un sito di raccolta centrale.
3. Crackers: utenti che cercano di compromettere o eseguire deface ad applicazioni per un guadagno, per la notorietà o per un'idea politica.
4. Hackers veri e propri: un ricercatore di sicurezza o un utente ordinario, che nota qualcosa di sbagliato con l'applicazione, e decide di proseguire nella scoperta della vulnerabilità, per poi creare immediatamente una patch e comunicare agli amministratori la falla trovata e la possibile soluzione.
5. Lo scopritore casuale: un utente normale che si imbatte in un errore funzionale nell'applicazione, semplicemente utilizzando un browser web, e guadagna l'accesso a informazioni privilegiate o funzionalità di livello superiore.

Definite minacce e agenti, possiamo effettuare un'analisi dei rischi della nostra applicazione, basandoci sulle minacce che possono verificarsi contro la nostra Certificate Authority.

2.3 Rischi e Contromisure

In questa sezione cerchiamo di individuare i rischi principali di attacchi possibili alla CA iMovies. Innanzi tutto già in fase di progettazione avevamo preso in considerazione possibili rischi e vulnerabilità di sistema, in modo da cercare di progettare un sistema sicuro già in partenza. Infatti la scelta di utilizzare Java Server Faces è venuta anche da questa valutazione iniziale.

Dopodichè a lavoro ultimato si è deciso di cercare di migliorare ancora di più la sicurezza del sito, testandolo con alcuni tool automatici e con utenti scelti tra amici (come ad esempio Federico De Meo e Alberto Lovato). Oltre ad aiutarci nel testing della Certificate Authority ci hanno anche aiutato a svolgere alcuni attacchi presi dal Common Attack Pattern Enumeration and Classification (CAPEC)⁵.

Presentiamo ora i tool principali che abbiamo utilizzato per cercare eventuali falle nel sistema. Dopodichè nella parte successiva nelle tabelle presenti

⁵<http://capec.mitre.org/>

evidenzieremo come noi abbiamo considerato e valutato i livelli di rischio, probabilità e impatto delle vulnerabilità trovate e per cui abbiamo poi trovato una contromisura.

2.3.1 Tools

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

- **Websecurify**⁶

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 del server.

- **Wapiti**⁷

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*⁸.

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

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*⁹.

⁶<http://www.websecurify.com>

⁷<http://www.ict-romulus.eu/web/wapiti/home>

⁸Un 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.

⁹Questo codice d'errore indica un errore generico avvenuto sul server a seguito di una richiesta impossibile da risolvere (Internal server error).

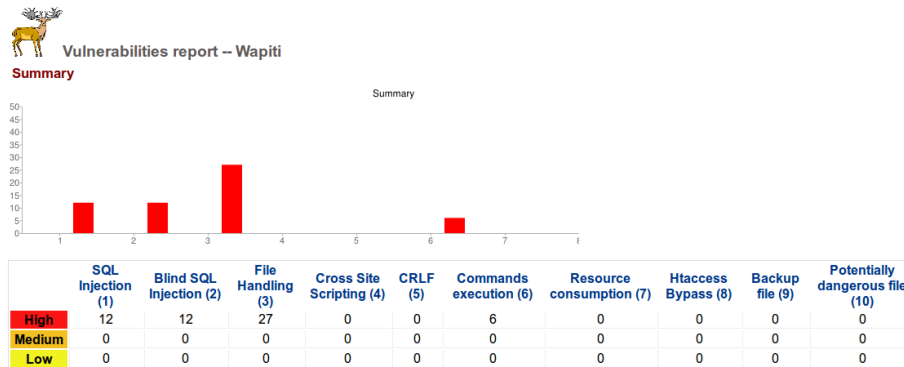


FIGURA 6: Il report generato dal tool Wapiti

- **Exploit-me**¹⁰

Access-me non compatibile con Firefox 13

- **Skipfish**¹¹

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:

```
# 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
```

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 riguardando 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
```

¹⁰<http://labs.securitycompass.com/exploit-me/>

¹¹<http://code.google.com/p/skipfish/>

```

Welcome to skipfish. Here are some useful tips:

1) To abort the scan at any time, press Ctrl-C. A partial report will be written
   to the specified location. To view a list of currently scanned URLs, you can
   press space at any time during the scan.

2) Watch the number requests per second shown on the main screen. If this figure
   drops below 100-200, the scan will likely take a very long time.

3) The scanner does not auto-limit the scope of the scan; on complex sites, you
   may need to specify locations to exclude, or limit brute-force steps.

4) There are several new releases of the scanner every month. If you run into
   trouble, check for a newer version first, let the author know next.

More info: http://code.google.com/p/skipfish/wiki/KnownIssues

NOTE: The scanner is currently configured for directory brute-force attacks,
and will make about 244923 requests per every fuzzable location. If this is
not what you wanted, stop now and consult the documentation.

Press any key to continue (or wait 60 seconds)... █

```

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

```

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.

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


```
skipfish version 2.07b by <lcantuf@google.com>

- 10.236.54.153 -

Scan statistics:

  Scan time : 9:09:41.155
  HTTP requests : 6844717 (207.5/s), 5145131 kB in, 1881612 kB out (213.1 kB/s)
  Compression : 0 kB in, 0 kB out (0.0% gain)
  HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 42376 total (161.6 req/conn)
  TCP faults : 0 failures, 0 timeouts, 1 purged
  External links : 142 skipped
  Reqs pending : 2018

Database statistics:

  Pivots : 179 total, 166 done (92.74%)
  In progress : 2 pending, 5 init, 4 attacks, 2 dict
  Missing nodes : 49 spotted
  Node types : 1 serv, 87 dir, 35 file, 0 pinfo, 9 unkn, 47 par, 0 vall
  Issues found : 128 info, 43 warn, 2 low, 12 medium, 0 high impact
  Dict size : 2258 words (11 new), 110 extensions, 256 candidates
  g nodes : 49 spotted
  Node types : 1 serv, 85 dir, 30 file, 0 pinfo, 9 unkn, 47 par, 0 val
  Issues found : 127 info, 43 warn, 2 low, 12 medium, 0 high impact
  Dict size : 2258 words (11 new), 110 extensions, 256 candidates
```

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

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*¹² di Tomcat. L'idea è quella di creare un'implementazione di un realm (estendendo la classe *RealmBase*) che rifletta i requisiti di: accesso

¹²I *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

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.

Impatto	
Impatto	Descrizione
Alto	Significa che l'impatto dell'evento malevolo porterebbe ad una quasi totale compromissione del sistema, con conseguente perdita di riservatezza, integrità, e disponibilità dei dati.
Medio	Comporta che un evento malevolo potrebbe scatenare una perdita di dati media, cioè potrebbe portare alla perdita di riservatezza dei dati, o di disponibilità, senza intaccarne l'integrità.
Basso	Significa che un evento malevolo etichettato con questo livello di impatto non porterebbe problemi all'integrità delle informazioni del portale, portando a perdita di dati di non elevato interesse, oppure a una perdita di disponibilità temporanea.

Probabilità	
Probabilità	Descrizione
Alta	Significa che vi è un'alta probabilità che un agente malevolo possa facilmente avere l'opportunità di eseguire un attacco e che quest'ultimo vada a buon fine.
Media	Significa che vi è una media probabilità che l'agente abbia le capacità di eseguire determinati attacchi e di portarli a buon termine, e anche in caso vi riesca è molto probabile che le misure di sicurezza diano i propri frutti.
Bassa	Significa che un agente ha poche probabilità di eseguire un attacco (per mancanza di conoscenze o capacità) e di portarlo a buon termine. Bassa probabilità sta anche a indicare che le misure di sicurezza prevengono totalmente un determinato tipo di attacco.

(con prompt di username e password in stile `.htaccess` con hashing della password, con un form definito dall'utente, con user certificate).

Livello di rischio			
Probabilità	Impatto		
	Basso	Medio	Alto
Alta	Basso	Medio	Alto
Medio	Basso	Medio	Medio
Bassa	Basso	Basso	Basso

2.3.2 Valutazione del portale iMovies

Valutazione riguardante la parte software del portale iMovies, quindi tutte le possibilità di attacchi presi in considerazione e le contromisure per evitarli. Inoltre viene inserita una valutazione generale della probabilità, dell'impatto e del rischio dovuto a un determinato attacco. Per farlo ci si è affidati alla Top 10 dei rischi nelle web application del 2010 [?] e al CAPEC.

No.	Threat	Impl./planned countermeasure(s)	L	I	Risk
1	Cross Site Request Forgery	JSF 2.0 ha già incorporato un sistema di prevenzione al CSRF. Si tratta di <code>javax.faces.ViewState</code> , un campo nascosto nel form. Usa valore autogenerato robusto come prevenzione al CSRF.	<i>Medio</i>	<i>Alto</i>	<i>Basso</i>
2	Cross-Site Scripting	JSF 2.0 ha già incorporato un sistema di prevenzione al XSS. JSF fa l'escape nella visualizzazione di un campo inserito dall'utente. <code><h:outputText/></code> e <code><h:outputLabel/></code> hanno un attributo <code>escape=true</code> di default e che quindi si può omettere. Si può anche scrivere <code><p>Welcome, #user.name</p></code> che viene automaticamente sanitizzato.	<i>Medio</i>	<i>Alto</i>	<i>Medio</i>
3	Cross-Site Scripting	Protezione dei cookies. Per proteggere i cookies da Javascript maligni, i web server supportano la feature di permettere all'applicazione di specificare se un determinato cookie può essere acceduto da Javascript o solo da Http. Questa funzionalità è stata abilitata per tutte le app in <code>conf/context.xml</code> aggiungendo <code><Context useHttpOnly=true></code>	<i>Medio</i>	<i>Alto</i>	<i>Medio</i>
4	Sql Injection	Questo tipo di attacco non è responsabilità di JSF. Per ovviarlo usiamo query parametrizzate, utilizzando i <code>PreparedStatement</code> di java per passare i parametri.	<i>Alta</i>	<i>Medio</i>	<i>Alto</i>
5	Broken Authentication and Session Management	Per ovviare a questo tipo di attacco le sessioni sono dotate di un <i>timeout</i> di 5 minuti che invalida e distrugge la sessione. Inoltre al logout dal sistema ogni sessione viene invalidata.	<i>Media</i>	<i>Medio</i>	<i>Alto</i>

6	Security Misconfiguration	Tutte le password standard di Apache Tomcat sono state modificate. Credenziali di default modificate. Errori gestiti in modo che alla richiesta di pagine non esistenti si venga re-indirizzati in una pagina di errore. Attivazione SSL su Apache Tomcat.	<i>Media Medio Alto</i>
7	Failure to Restrict URL Access	compila pure mattia :D	<i>Media Medio Alto</i>
8	Insufficient Transport Layer Protection	Grazie alle <i>navigation rule</i> presenti nel file <code>faces-config.xml</code> dell'applicazione, le regole di redirect sono ben definite da e per ogni pagina.	<i>Media Basso Medio</i>

2.3.3 Valutazione della macchina server iMovies

Piccola valutazione dei problemi che potrebbe avere il software della macchina server.

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
1	Perdita di dati	Viene eseguito un backup incrementale giornaliero e orario, e uno totale ogni venerdì alle ore 11 di tutte le chiavi private e certificati della CA. I backup vengono inseriti in una cartella all'interno di <code>/var/ftp/admin</code> non accedibile.	<i>Bassa</i>	<i>Alto</i>	<i>Medio</i>
2	Accesso macchina e dati a malintenzionati	problema da sistemare. dobbiamo mettere che solo tomcat e root possono toccare le cartelle.	<i>Media</i>	<i>Alto</i>	<i>Medio</i>
3	Impossibilità di accesso alla macchina server	È stato installato un demone SSH che permette l'accesso da remoto alla macchina server, permettendo operazioni di manutenzione da remoto.	<i>Media</i>	<i>Alto</i>	<i>Medio</i>
4	Impossibilità di accesso alla macchina server	È stato installato un server FTP per il download dei backups da remoto. La cartella dove è in esecuzione il server FTP (che è la cartella contenente i backups) è in <code>chroot()</code> . Significa che l'utente remoto vedrà quella cartella come radice.	<i>Media</i>	<i>Alto</i>	<i>Medio</i>
5	Compromissione Web Server	Apache Tomcat offre la possibilità di eseguire lo shutdown del Web Server da remoto, creando una classe Java che crea una connessione socket sulla porta prestabilita e inviando una determinata password nel buffer.	<i>Media</i>	<i>Alto</i>	<i>Medio</i>
3	dobbiamo pensarci	insieme domani	<i>Media</i>	<i>Alto</i>	<i>Medio</i>
3	dobbiamo pensarci	insieme domani	<i>Media</i>	<i>Alto</i>	<i>Medio</i>

2.3.4 Descrizione dettagliata delle contromisure scelte

La scelta di Java Server Faces non è stata del tutto casuale, o dettata dalla presenza del linguaggio di programmazione Java già conosciuto da tutti noi.

L'utilizzare un framework è stata una scelta dovuta anche al fatto di avere a disposizione uno strumento che già implementasse delle contromisure agli attacchi più diffusi. Questo ci ha permesso oltre che ovviare all'implementazione "personale" di contromisure (come nel caso del CSRF) ma anche di disporre di contromisure già implementate e relativamente sicure, perchè già testate da molti utenti, e create appunto dagli stessi sviluppatori e progettisti.

In particolare avremmo piacere di soffermarci sull'implementazione di JSF di una delle contromisure sul Cross Site Request Forgery.

Spiegazione di un paio di contromisure nel dettaglio.

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.

meo docet: meo ha provato a fare attacco csrf. per farlo sei andato a guardare che tipo di richiesta la pagina edit.xhtml faceva per modificare i dati di un utente. dopo questa analisi ho preso brutalmente la richiesta e l'ho fatta a mano con webscarab e si è visto che veniva soddisfatta. la richiesta aveva da qualche parte dei : non codificati in urlencoded. dopodichè ho provato a replicarla creando una pagina web ad hoc e mi ero arenato in un problema di difficoltà nell'invio del simbolo di : perchè continuava a codificarli e io non li volevo codificati. giunto alla mezza conclusione che non si potesse fare in html ma solo tramite una richiesta javascript ho notato che c'era un campo che si chiama javax. state ecc che conteneva un numeraccio che variava ad ogni richiesta. sono andato a investigare quel campo e a ricercare se jsf implementasse già sistemi contro csrf ho scoperto che lo fa.

Riferimenti bibliografici

- [1] David Basin, Patrick Schaller, Micheal Schläpfer - Springer - *Applied Information Security - A Hands-on Approach*

- [2] OWASP - The ten most critical web application security risk 2010 - OWASP
Top Ten for 2010