#	Rif	Requisito di Verifica	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4	Verifica
1,01	ASVS-1.1	Verificare che siano state identificate tutte le componenti applicative (come file o gruppi, sorgenti, librerie e/o eseguibili) presenti nell'applicazione.	✓	✓	✓	✓	✓	✓	
1,02	ASVS-1.2	Verificare che siano state identificate tutte le compoenti utilizzate dall'applicazione e che non ne fanno parte.			✓	✓	✓	✓	
1,03	ASVS-1.3	Verificare che sia stata definita l'architettura di alto livello dell'applicazione.			✓	✓	✓	✓	
1,04	ASVS-1.4	Verificare che siano state definite tutte le componenti dell'applicazione sia in termini di funzionalità che di sicurezza.					✓	✓	
1,05	ASVS-1.5	Verificare che siano state definite tutte le componenti utilizzate dall'applicazione sia in termini di funzionalità che di sicurezza.					✓	✓	
1,06	ASVS-1.6	Verificare che sia stata effettuata la modellazione delle minacce					✓	✓	
2,01	ASVS-2.1	Verificare che, eccetto per quelle inerentemente pubbliche, tutte le pagine e le risorse accessibili richiedano un controllo di autenticazione.	✓	✓	✓	✓	✓	✓	

2,02	ASVS-2.2	Verificare che tutti i form e i campi di autenticazione abbiamo la funzionalità di autocompletamento disabilitata e che il campo relativo alla password non faccia visualizzare a schermo l'input immesso.	✓	✓	✓	✓	✓	✓
2,03	ASVS-2.3	Verificare che se viene superato un certo numero di tentativi di autenticazione, l'account venga bloccato per un certo lasso di tempo in modo da scoraggiare attacchi di tipo brute-force.	✓		✓	✓	✓	✓
2,04	ASVS-2.4	Verificare che tutti i controlli di autenticazione siano implementati lato server.			✓	✓	✓	✓
2,05	ASVS-2.5	Verificare che tutti i controlli di autenticazione (comprese le librerie che richiedono un servizio di autenticazione esterno) abbiano un'implementazione centralizzata				✓	✓	✓
2,06	ASVS-2.6	Verificare che tutti i controlli di autenticazione falliscano in sicurezza.			✓	✓	✓	✓
2,07	ASVS-2.7	Verificare che la complessità delle credenziali di autenticazione sia sufficiente da evitare attacchi che sono tipici delle minacce dell'ambiente in cui vengono rilasciate			✓	✓	✓	✓

2,08	ASVS-2.8	Verificare che tutte funzioni di gestione dell'account siano resistenti agli attacchi almeno quanto lo è il meccanismo di autenticazione primaria.			✓	✓	✓	✓
2,09	ASVS-2.9	Verificare che gli utenti possano cambiare in modo sicuro le loro credenziali, usando un meccanismo che sia resistente agli attacchi almeno quanto quello di autenticazione.			✓	✓	✓	✓
2,10	ASVS-2.10	Verificare che sia necessario ri-autenticarsi prima di effettuare una operazione particolarmente critica.			✓	✓	✓	✓
2,11	ASVS-2.11	Verificare l'effettiva scadenza delle password dopo la durata massima prevista.			✓	✓	✓	✓
2,12	ASVS-2.12	Verificare che tutti gli eventi relativi all'autenticazione siano correttamente tracciati.				✓	✓	✓
2,13	ASVS-2.13	Verificare che le password vengano cifrate prima di essere memorizzate utilizzando un elemento di salting univoco per ogni account (es: ID interno, data di creazione dell'account).				✓	✓	✓
2,14	ASVS-2.14	Verificare che tutte le credenziali di autenticazione utilizzate per l'accesso a servizi o componenti esterne all'applicazione, siano cifrate e conservate in una locazione sicura e non nel codice sorgente.				✓	✓	✓

2,15	ASVS-2.15	Verificare che tutto il codice da cui dipende l'autenticazione non contenga codice malevolo (vulnerabilità note nelle librerie, eventuali backdoor o hack)						✓
3,01	ASVS-3.1	Verificare che l'applicazione usi l'implementazione di controllo di gestione della sessione predefinita fornita dal framework	✓		✓	✓	✓	✓
3,02	ASVS-3.2	Verificare che la sessione venga correttamente invalidata quando l'utente effettua il login: sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione)	✓		✓	✓	✓	✓
3,03	ASVS-3.3	Verificare che la sessione venga invalidata correttamente dopo un congruo tempo di inattività: sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione).	✓		✓	✓	✓	✓
3,04	ASVS-3.4	Verificare che la sessione venga invalidata correttamente dopo un congruo tempo di vita (un timeout assoluto della sessione), possibilmente configurabile, sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione)					✓	✓
3,05	ASVS-3.5	Verificare che in tutte le pagine nelle quali viene richiesta l'autenticazione sia presente l'apposito collegamento per effettuare il logout	✓		✓	✓	✓	✓

3,06	ASVS-3.6	Verificare che l'identificativo di sessione non venga mai mostrato se non nell'intestazione del cookie; in particolare negli URL e nei messaggi di errore, o sui log. Questo implica la verifica che l'applicazione non supporti il rewriting degli indirizzi dei cookie di sessione.		✓		✓	✓	✓
3,07	ASVS-3.7	Verificare che l'identificativo di sessione venga correttamente invalidato dopo il login: sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione)			✓	✓	✓	✓
3,08	ASVS-3.8	Verificare che l'identificativo di sessione venga correttamente invalidata dopo una re-auenticazione: sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione)			✓	✓	✓	✓
3,09	ASVS-3.9	Verificare che l'identificativo della sessione venga correttamente invalidata dopo il logout: sia lato client (quindi resettando il cookie) che lato server (distruggendo la sessione)			✓	✓	✓	✓
3,10	ASVS-3.10	Verificare che l'applicazione riconosca come validi solo gli identificativi di sessione generati dall'applicazione stessa.			✓		✓	✓

3,11	ASVS-3.11	Verificare che gli identificativi di sessione, in particolare dopo l'autenticazione, siano sufficiente validi e casuali da evitare le minacce tipiche dell'ambiente nel quale vengono implementati.					✓	✓
3,12	ASVS-3.12	Verificare che i cookie che contengono le informazioni sulla sessione abbiano gli attributi "domain" e "path" valorizzati con parametri consoni all'applicazione.					✓	✓
3,13	ASVS-3.13	Verificare che tutto il codice da cui dipende la gestione della sessione non contenga codice malevolo (vulnerabilità note nelle librerie, eventuali backdoor o hack)						✓
4,01	ASVS-4.1	Verificare che gli utenti possano accedere solo alle funzioni protette per le quali posseggono una specifica autorizzazione.	✓	✓	✓	✓	✓	✓
4,02	ASVS-4.2	Verificare che gli utenti possano accedere solo agli URL per i quali posseggono una specifica autorizzazione.	✓		✓	✓	✓	✓
4,03	ASVS-4.3	Verificare che gli utenti possano accedere solo ai file per i quali posseggono una specifica autorizzazione.	✓		✓	✓	✓	✓

4,04	ASVS-4.4	Verificare che i riferimenti diretti agli oggetti siano bloccati, in questo modo gli utenti possono accedere solo agli oggetti per i quali sono autorizzati.	✓		✓	✓	✓	✓
4,05	ASVS-4.5	Verificare che, per tutte le directory, sia disabilitato la possibilità di navigazione (directory browsing).	✓		✓		✓	✓
4,06	ASVS-4.6	Verificare che gli utenti possano accedere solo ai servizi per i quali posseggono specifica autorizzazione.			✓	✓	✓	✓
4,07	ASVS-4.7	Verificare che gli utenti possano accedere solo ai dati per i quali posseggono specifica autorizzazione.			✓	✓	✓	✓
4,08	ASVS-4.8	Verificare che l'autorizzazione fallisca in sicurezza.			✓	✓	✓	✓
4,09	ASVS-4.9	Verificare che tutti i controlli di autorizzazione implementati nel livello di presentazione lo siano presenti anche nei livelli inferiori.			✓	✓	✓	✓
4,10	ASVS-4.10	Verificare che l'utente non possa modificare in autonomia il proprio profilo di autorizzazione (o quello di altri) a meno che non sia stato autorizzato.			✓	✓	✓	✓
4,11	ASVS-4.11	Verificare che tutti i controlli autorizzativi siano implementati lato server.			✓	✓	✓	✓

4,12	ASVS-4.12	Verificare che sia utilizzato un meccanismo centralizato per la protezione dell'accesso alle risorse di vario tipo (includendo le librerie che richiedono un servizio di autorizzazione esterno).				✓	✓	✓
4,13	ASVS-4.13	Verificare che se sono presenti delle limitazoini imposte delle regole di business (es. limiti giornalieri per un determinato tipo di transazione), questi non possano essere bypassati.			✓	✓	✓	✓
4,14	ASVS-4.14	Verificare che i cambiamenti nello schema di autorizzazione possano essere tracciati, come anche i tentativi.				✓	✓	✓
4,15	ASVS-4.15	Verificare che il codice che gestisce la componente di autorizzazione non sia affetta da codice malevolo.						✓
5,01	ASVS-5.1	Verificare che, al runtime, l'ambiente non sia suscettibile ad attacchi di overflow o che siano stati implementati dei controlli per prevenire tali attacchi.	✓	✓	✓	✓	✓	✓
5,02	ASVS-5.2	Verificare la presenza di un controllo a white list "positivo" per tutti gli input a seconda del caso specifico.	✓	✓	✓	✓	✓	✓
5,03	ASVS-5.3	Verificare che quando in input non rispetta le regole sia rigettato (quindi la richiesta bloccata) oppure sia correttamente sanificato.	✓		✓	✓	✓	✓

5,04	ASVS-5.4	Verificare che sia stato specificato un character-set per tutti gli input (es. UTF-8).			✓	✓	✓	✓
5,05	ASVS-5.5	Verificare che convalida degli input avvenga anche lato server.			✓	✓	✓	✓
5,06	ASVS-5.6	Verificare che sia utilizzato un controllo di convalida per ogni tipologia di dato che viene accettato in input.				✓	✓	✓
5,07	ASVS-5.7	Verificare che siano tracciate le richieste che contengono input non valido.				✓	✓	✓
5,08	ASVS-5.8	Verificare che tutti i dati in input siano normalizzati prima della convalida.					✓	✓
5,09	ASVS-5.9	Verificare che tutto il codice che gestisce componente di input validation non sia affetto da codice malevolo.						✓
6,01	ASVS-6.1	Verificare che tutti gli output che possono contenere (o che siano visualizzati su) applicazioni che utilizzano tecnologie del W3C (elementi o attributi HTML, javascript, CSS, URI o eventuali dialetti come il Bbcode) siano soggetti a tecniche di validazione secondo il contesto.		✓	✓	✓	✓	✓
6,02	ASVS-6.2	Verificare che tutti i controlli di validazione o codifica dei dati vegano applicati lato server.			✓	✓	✓	✓
6,03	ASVS-6.3	Verificare che i controlli di validazione codifichino tutti i caratteri appartenenti a codifiche utilizzate così da non estendere la problematica agli interpreti.				✓	✓	✓

6,04	ASVS-6.4	Verificare che tutti i dati inaffidabili che utilizzati come codice SQL vengano utilizzati tramite prepared statement o API parametrizzate, o che siano soggetti a tecniche di validazione secondo il contesto.				✓	✓	✓
6,05	ASVS-6.5	Verificare che tutti i dati inaffidabili utilizzati da interpreti XML utilizzino API parametriche, ovvero siano soggetti a tecniche di validazione secondo il contesto.				✓	✓	✓
6,06	ASVS-6.6	Verificare che tutti i dati inaffidabili utilizzati tramite query LDAP utilizzino API parametriche, ovvero siano soggetti a tecniche di validazione secondo il contesto.				✓	✓	✓
6,07	ASVS-6.7	Verificare che tutti dati inaffidabili utilizzati come parametri per utilizzare i comandi sistema operativo siano soggetti a tecniche di validazione secondo il contesto.				✓	✓	✓
6,08	ASVS-6.8	Verificare che tutti i dati inaffidabili utilizzati da un qualsiasi interprete siano soggetti a tecniche di validazione secondo la tipologia dell'inteprete.				✓	✓	✓
6,09	ASVS-6.9	Verificare che per ogni tipologia di output gestito dall'applicazione sia presente almeno un controllo per quella specifica tipologia.					✓	✓

6,10	ASVS-6.10	Verificare che tutto il codice che implementa il controllo dell'output non sia affetto da codice malevolo.						✓
7,01	ASVS-7.1	Verificare che tutte le funzioni crittografiche utilizzate siano implementate lato server, a meno che non sia necessaria specificatamente crittografia lato client.			✓	✓	✓	✓
7,02	ASVS-7.2	Verificare che tutti i moduli crittografici falliscano in sicurezza.			✓	✓	✓	✓
7,03	ASVS-7.3	Verificare che i master-secret siano protetti dall'accesso non autorizzato (un master-secret è la password utilizzata dall'applicazione memorizzata in chiaro sul disco che è usata per proteggere l'accesso alle informazioni di configurazione).				✓	✓	✓
7,04	ASVS-7.4	Verificare l'utilizzo del salting quando si implementano funzioni di hash per la protezione delle password.				✓	✓	✓
7,05	ASVS-7.5	Verificare che le problematiche nei moduli crittografici siano tracciate.				✓	✓	✓
7,06	ASVS-7.6	Verificare che tutte le componenti che dipendono da identificatvi causuali (URL, numeri,stringhe, nomi di file, GUID o altro) vengano genate utilizzando generatori di numeri casuali riconosciuti e che non siano suscettibili ad attacchi				✓	✓	✓

7,07	ASVS-7.7	Verificare che i moduli crittografici utilizzati siano stati validati dall'organizzazione o da standard internazionali (es. FIPS 140-2 http://csrc.nist.gov/groups/STM/cmvp/validation.html)					✓	✓
7,08	ASVS-7.8	Verificare che tutti i moduli crittografici vengano utilizzati secondo le corrette pratiche di sicurezza (http://csrc.nist.gov/groups/STM/cmvp/validation.html)					✓	✓
7,09	ASVS-7.9	Verificare sia presente una politica espicita sulla gestione delle chiavi di crittografia (es. generazione, distribuzione, revoca, scadenza). Verificare che questa politica sia realmenete applicata.					✓	✓
7,10	ASVS-7.10	Verificare che tutto il codice che supporta o usa un modulo crittografico non contenga codice malevolo.						✓
8,01	ASVS-8.1	Verificare che l'appllicazione non restituisca messaggi di errori dettagliati, stack trace o errori con informazioni che possano supportare un eventuale attaccante, come l'id di sessione, informazioni personali o del sistema.	✓	✓	✓	✓	✓	✓
8,02	ASVS-8.2	Verificare che tutti gli errori siano gestiti lato server.			✓	✓	✓	✓
8,03	ASVS-8.3	Verificare che tutti i controlli di tracciatura siano implementati lato server.			✓	✓	✓	✓

8,04	ASVS-8.4	Verificare che se si verificano errori nella logica applicativa, per default venga bloccata la richiesta.			✓	✓	✓	✓
8,05	ASVS-8.5	Verificare che sia possibile tener traccia sia del successo sia dell'insuccesso degli eventi ritenuti rilevanti per la sicurezza.				✓	✓	✓
8,06	ASVS-8.6	Verificare che un log includa almeno 1. un timestamp da fonte affidabile 2. livello di attenzione dell'evento 3. un evidenza che si tratti di un evento rilevante per la sicurezza (se confuso con altri log). 4. un elemento che permetta di identificare l'utente che ha generato l'evento (se c'è un utente associato) 5. l'indirizzo IP sorgente che ha effettuato la richiesta 6. l'indicazione se la richiesta ha avuto sucesso o meno 7. una descrizione dell'evento				✓	✓	✓
8,07	ASVS-8.7	Verificare che eventuale codice presente nelle richieste non possa essere eseguito dal software di visualizzazione dei log.				✓	✓	✓
8,08	ASVS-8.8	Verificare che i log di sicureza siano protetti da accessi, modifiche e cancellazioni non autorizzate.				✓	✓	✓
8,09	ASVS-8.9	Verificare che la gestione dei log all'interno dell'applicazione sia centralizzata.				✓	✓	✓

8,10	ASVS-8.10	Verificare che l'applicazione non tracci informazioni sensibili che potrebbero essere utili ad un attaccante, come l'identificativo di sessione o informazioni personali.				✓	✓	✓
8,11	ASVS-8.11	Verificare che lo strumento di analisi dei log permetta la ricerca di eventi tramite la combinazione di criteri di ricerca.				✓	✓	✓
8,12	ASVS-8.12	Verificare che tutto il codice che implementa la gestione degli errori e la tracciatura degli eventi non sia affetto da codice malevolo.						✓
9,01	ASVS-9.1	Verificare che tutti i form che contengono informazioni riservate evitino il salvataggio di tali informazioni sul client (es. autocomplete=off, caching delle pagine)	✓	✓	✓	✓	✓	✓
9,02	ASVS-9.2	Verificare che sia stata identificata la lista delle informazioni sensibili trattete dall'applicazione e che siano presenti dei criteri di accesso sia per quel che riguarda il transito che la memorizzazione. Quindi verificare che tali criteri siano implementati in maniera corretta.				✓	✓	✓
9,03	ASVS-9.3	Verificare che ogni informazione sensibile sia trasmessa al server nel corpo del messaggio HTTP (es. i metodi URL non devono mai essere usati per trasmettere dati sensibili)			✓		✓	✓

9,04	ASVS-9.4	Verificare che tutte le informazioni sensibili per le quali sono state effettuate copie nella cache del client siano protetti da accesso non autorizzato e che abbiano dei limiti temporali (es. presenza degli header no-cache, no-store e Cahce-control).				✓	✓	✓
9,05	ASVS-9.5	Verificare che tutte le informazioni sensibili per le quali sono state effettuate copie nel server siano protetti dall'accesso non autorizzato, quindi cancellati dopo il loro utilizzo				✓	✓	✓
9,06	ASVS-9.6	Verificare che sia presente un metodo per rimuovere i dati sensibili all'interno dell'applicazione dopo un ben definito periodo di conservazione.					✓	✓
10,01	ASVS-10.1	Verificare che l'applicazione utilizzi un certificato generato da una CA riconosciuta.	✓		✓	✓	✓	✓
10,02	ASVS-10.2	Verificare che, in caso di problematiche con la connessione TLS, non venga proposta una connessione insicura.			✓		✓	✓
10,03	ASVS-10.3	Verificare che TLS sia utilizzato per tutte le connessioni (anche tra frontend e backend) e che sia autenticato quando vengono trattati dati o funzioni riservate.				✓	✓	✓
10,04	ASVS-10.4	Verificare che siano tracciate eventuali problematiche con TLS.				✓	✓	✓
10,05	ASVS-10.5	Verificare eventuali certificati lato client siano stati generati da CA riconosciute e che non siano stati revocati.				✓	✓	✓

10,06	ASVS-10.6	Verificare che tutte le connessioni verso l'esterno dove vengono trasmesse informazioni sensibili siano autenticate.				✓	✓	✓
10,07	ASVS-10.7	Verificare che tutte le connessioni verso l'esterno che trasmettono informazioni sensibili utilizzino account aventi privilegi minimi.				✓	✓	✓
10,08	ASVS-10.8	Verificare che sia presente un'unica implementazione di TLS da utilizzare per le varie applicazioni (http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf)					✓	✓
10,09	ASVS-10.9	Verificare che siano utilizzati solo codifiche permesse (es. UTF-8)					✓	✓
11,01	ASVS-11.1	Verificare che eventuali redirect non includano dati non verificati.	✓	✓	✓	✓	✓	✓
11,02	ASVS-11.2	Verificare che l'applicazione accetti solo una lista ben definita di metodi HTTP, come ad esempio GET e POST.	✓	✓	✓	✓	✓	✓
11,03	ASVS-11.3	Verificare che ogni risposta HTTP contegna le informazioni rispetto il set di caratteri da utilizzare (es. UTF-8).	✓	✓	✓	✓	✓	✓
11,04	ASVS-11.4	Verificare che il flag "HTTPOnly" sia impostato su tutti i cookie che non richiedano l'accesso tramite Javascript.			✓	✓	✓	✓
11,05	ASVS-11.5	Verificare che il flag "Secure" sia impostato per tutti i cookie che contengono dati sensibili, come ad esempio l'idnetificativo di sessione.			✓	✓	✓	✓

11,06	ASVS-11.6	Verificare che negli header HTTP siano presenti solo caratteri ASCII stampabili.			✓	✓	✓	✓
11,07	ASVS-11.7	Verificare che l'applicazioni generi un token anti-CSRF come parte di un link quando si effettuano operazioni di inserimento, modifica o cancellazione di informazioni. Quindi che verifichi la correttezza del token prima di processare la richiesta.					✓	✓
12,01	ASVS-12.1	Verificare che tutte le informazioni rispetto file di configurazione di controlli inerenti alla sicurezza siano protette dall'accesso non autorizzato				✓	✓	✓
12,02	ASVS-12.2	Verificare che l'applicazione neghi l'accesso se non riesce a leggere i flie di configurazione rispetto la sicurezza.				✓	✓	✓
12,03	ASVS-12.3	Verificare che tutte le modifiche alle impostazioni si sicurezza siano propriamente tracciate.					✓	✓
12,04	ASVS-12.4	Verificare che i file di configurazione possano essere esportati in un formato comprensibile all'uomo per facilitare il processo di audit.						✓
13,01	ASVS-13.1	Verificare che non sia presente codice malevolo all'interno del codice creato o utilizzato dall'applicazione.						✓

13,02	ASVS-13.2	Verificare che siano state utilizzate tecniche di verifica del codice utilizzato, librerie, eseguibili e configurazioni utilizzando checksum o hash						✓
14,01	ASVS-14.1	Verificare che l'applicazione protegga gli utenti e le informazioni relative alle policy di controllo degli accessi da visualizzazioni, modifiche o cancellazioni non autorizzate.					✓	✓
14,02	ASVS-14.2	Verificare che i controlli di sicurezza siano facili da implementare così da essere correttamente utilizzati dagli sviluppatori.						✓
14,03	ASVS-14.3	Verificare che l'applicazione protegga eventuali variabili condivise e le risorse dall'accesso concorrente.						✓