	concentrare l'accesso solo a determinati server interni senza consentire ad es. la manipolazione delle URL o dei parametri di una richiesta http POST in modo tale da ottenere una connessione verso un indirizzo arbitrario della intranet. In conformità al principio della defense-in-depth, il firewall che protegge tale sistema deve essere configurato in maniera puntuale per consentire unicamente le connessioni previste da internet verso il server e dal server verso gli altri server interni (indirizzi ip e porte dei soli server effettivamente previsti). In tal modo l'accesso al portale non deve permettere accessi non autorizzati a reti a cui il sistema è inter-connesso.

Controllo del traffico dati	
Minaccia	- Accesso non autorizzato alle informazioni. - Attacchi all'integrità delle informazioni. - (esempi: Zero-day exploit, Remote File Inclusion)
Contromisure	Impiegare un Web Application Firewall (WAF): - Il web application firewall consente il controllo di tutti i tipi di richiesta HTTP (URL, form, cookie, query string, hidden field e parametri). - L'impiego di una blacklist di URL referenziate consente al WAF di bloccare exploit basati su vulnerabilità applicative "zero-day" (portate lo stesso giorno in cui la vulnerabilità diventa nota).

Controllo del traffico dati	
Minaccia	Accesso non autorizzato alle informazioni - HTML Injection
Contromisure	Utilizzare un Web Application Firewall capace di monitorare la comunicazione tra gli utenti e l'applicazione e creare profili di interazioni HTML consentite.

Controllo del traffico dati	
Minaccia	- Furto di credenziali di autenticazione - Negazione del servizio
Contromisure	Attivare, a livello perimetrale, un dispositivo di sicurezza intelligente di tipo IDS (Intrusion Detection System) o IPS (Intrusion Prevention System) per individuare (IDS) la presenza di codice malevolo e bloccare (IPS) le intrusioni.

Comunicazioni sicure tra differenti Application Server	
Minaccia	- Accesso non autorizzato alle informazioni. - Attacchi all'integrità delle informazioni.
Contromisure	Quando un servizio web utilizza un'architettura distribuita, composta da server web di front-end e application server, ciascun server deve essere dotato di un certificato digitale e deve comunicare con gli altri in HTTPS attraverso TLS 1.2 o successivo. Sui sistemi maggiormente critici ed esposti come front-end su Internet, o usati per transazioni commerciali, la chiave privata deve essere custodita su un dispositivo hardware esterno (HSM).

5.5.2 Hardening

Hardening della piattaforma web	
Minaccia	- Abuso di privilegi - Abuso di risorse - Accesso non autorizzato alle informazioni

	- Accesso non autorizzato al sistema (macchina, configurazione, ecc.)
Contromisure	- Concedere al Web Server i privilegi minimi necessari per completare le operazioni richieste. In particolare dovrà utilizzare un account nominativo diverso da “root” o “administrator”. - Disabilitare gli script, le applicazioni d'esempio, i servizi, le utility non strettamente necessari ed ogni altra funzionalità non pertinente agli scopi della piattaforma web, proposti dalle configurazioni di base del web server. - Limitare l'accesso al file system da parte del web server separando la root directory e le directory virtuali dal resto del file system, facendole puntare su partizioni / mount dedicate. - Disattivare sul web server la possibilità di navigazione del file system. - Disabilitare il “Directory Listing”. - Proteggere con opportune ACL su file system, i file di configurazione e le directory contenenti i siti web i log del server, i suoi eseguibili e i suoi file temporanei. - Modificare i messaggi di sistema eliminando tutte le informazioni atte ad identificare il tipo di server, la versione e la build. - Isolare il servizio web dal sistema che lo ospita e da altri servizi web utilizzando tecniche di “jail” o “chroot”, oppure containers Docker o altre tecniche di virtualizzazione in grado di fornire un efficace isolamento. - Se l’application server lo consente, eseguirlo attraverso una sandbox per proteggere il codice da errori, trojans e codice malizioso (es. eseguire Apache Tomcat attraverso il Security Manager). - Proteggere l’accesso all’interfaccia di amministrazione dell’application server attraverso un firewall o una VPN, in modo da restringere tale accesso ai soli indirizzi IP e utenti autorizzati. Forzare inoltre l’interfaccia amministrativa all’utilizzo di TLS 1.2 o successivi escludendo il semplice http.

Hardening della piattaforma web	
Minaccia	- Accesso non autorizzato alle informazioni - Path traversal - Violazione della sicurezza, rispetto alle politiche di sicurezza dell’organizzazione - esecuzione arbitraria di codice
Contromisure	- Configurare l'application server in modo tale da rifiutare le URL con sequenze "../", al fine di impedire l'attraversamento di percorsi non protetti. - Bloccare i comandi e le utility di sistema con ACL restrittive.

Hardening della piattaforma web	
Minaccia	Negazione dei servizi.
Contromisure	- Configurare le applicazioni, i servizi e il sistema operativo tenendo sempre presente le possibili esposizioni ad attacchi DoS. - Assicurarsi che i criteri di blocco dell'account predisposti non possano essere sfruttati per bloccare service accounts ben noti. - Assicurarsi che il sistema sia in grado di gestire alti volumi di traffico e che le soglie siano opportunamente impostate per gestire carichi anormalmente elevati. - Configurare il sistema operativo e il server web in modo da evitare il rischio di esaurimento di risorse in presenza di un elevato numero di connessioni non completate (es. TCP SYN COOKIES su kernel Linux/Unix e configurazione opportuna dei timeout sul server web). - Su server web soggetti ad un elevatissimo numero di connessioni, utilizzare applicativi con logica RESTful di tipo connectionless, o affidare l’onere di gestire i parametri della sessione al client attraverso l’inclusione dei parametri di sessione

	in cookies cifrati e non predicibili né manipolabili lato client. - Alcuni Application Server e Web Server espongono semplici interfacce amministrative per lo shutdown remoto che devono essere disabilitate (es. Apache Tomcat sulla porta TCP 8005).

Hardening della piattaforma web	
Minaccia	- Accesso non autorizzato alle informazioni. - Compromissione delle comunicazioni. - Falsificazione di identità. - Violazione della sicurezza, rispetto alle politiche di sicurezza dell'organizzazione (Cross-site scripting, Clickjacking, Hijacking, ecc.).

Contromisure	Configurare sempre una dimensione massima accettabile per l'Header http.
	
	Inoltre, considerare l'adozione di http security headers. Di seguito i principali:
	
	- HttpOnly: istruisce il browser ad impedire che i cookies siano acceduti lato client a mezzo di script;
	- strict-transport-security: forza il browser a comunicare solo su HTTPS;
	- cache-control: impostato a "no-cache, no-store, must-revalidate" (laddove sono in gioco dati sensibili);
	- expires: impostato a 0 (laddove sono in gioco dati sensibili);
	- content-security-policy: definisce quali sono le sorgenti attendibili dei contenuti (script) e quindi caricabili dal browser;
	- x-xss-protection: abilita un filtro sul browser che previene i XSS di tipo reflected;
	- x-frame-options: mette al riparo da un particolare tipo di attacco: il "clickjacking". Di fatto impedisce agli iframes di caricare il sito;
	- public-key-pins: istruisce il browser di associare una opportuna public key con un certo web server. Ciò mette al riparo:
	  o da Man-In-The-Middle attack (tentato con un certificate falso) o
	  o dall'eventualità in cui la certification authority fosse compromessa.
	- x-content-type: impostato a nosniff: mette al riparo da un particolare tipo di attacco: il "mime based attacks". Di fatto impone al browser di attenersi rigorosamente al content type specificato (es. se il server imposta il content come text/html, il browser ne farà il rendering come text/html).
	- expect-ct: impedisce l'utilizzo di certificati emessi in modo errato, consentendo ai siti web di segnalare e, facoltativamente, di imporre i requisiti di trasparenza dei certificati. Quando questa intestazione è abilitata, il sito web richiede al browser di verificare se il certificato appare o meno nei log pubblici della CT[5] (Certificate Transparency).
	- Feature-policy: conferisce la possibilità di consentire o negare l'utilizzo delle funzioni del browser, sia nel proprio frame che nel contenuto di un elemento iframe (<iframe>).
	NB. Non tutti i browser supportano gli http security headers di cui sopra. Anche la scelta del browser è importante.

Hardening della piattaforma web	
Minaccia	Accesso non autorizzato alle informazioni - Remote File Inclusion (RFI)
Contromisure	L'utilizzo di blacklist di IP costruiti sulla base di osservazioni eseguite su avvenuti attacchi (es. di tipo RFI), potrebbero essere usati per bloccare altri tipi di attacchi

[5] https://www.certificate-transparency.org/known-logs

	portati dalla stessa origine. Ove possibile, limitare gli accessi a indirizzi IP o Reti specifiche.

Hardening della piattaforma web	
Minaccia	Crittografia debole o non validata.
Contromisure	Non consentire il fallback a SSL (qualsiasi versione) né a TLS 1.1 o versioni inferiori. Deve essere richiesto l'uso obbligatorio almeno di TLS 1.2.

Hardening della piattaforma web	
Minaccia	Divulgazione di informazioni riservate.
Contromisure	Rimuovere HTTP Response Headers che espongono informazioni sul web server. A titolo di esempio, in ambiente Microsoft, rimuovere: - Server– Specifica la versione del web server version. - X-Powered-By- Indica che il website è "powered by ASP.NET." - X-AspNet-Version- Specifica la versione di ASP.NET usata. Disabilitare il metodo HTTP TRACE. A titolo di esempio: - in ambiente Microsoft, impostare la chiave di registro "EnableTraceMethod" (sotto HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters) a 0 (zero) - in ambiente Apache, configurare "TraceEnable off" in http.conf. - In Apache Tomcat disabilitare l'attributo allowTrace per ogni connettore. Su Apache Tomcat disabilitare inoltre lo Stack Tracing sul client.

Hardening della piattaforma web	
Minaccia	Negazione dei servizi (Buffer overflows).
Contromisure	- Utilizzare linguaggi di programmazione che forniscono controlli automatici sulla dimensione dei buffer di memoria (o a tempo di compilazione o a runtime) come Java, Python o Perl. - Utilizzare le safe libraries (ad es. in C e C++), ovvero librerie di funzioni che implementano protezioni contro il buffer overflow quando tale protezione non è nativamente supportata dal linguaggio di programmazione. - Prevedere che sia il compilatore ad inserire le verifiche sulla dimensione di tutti i buffer nel codice compilato senza richiedere alcuna modifica al codice sorgente (a titolo di esempio, utilizzare il flag /GS per compilare codice sviluppato con Microsoft Visual C ++ ®. Il flag / GS fa sì che il compilatore inietti controlli di sicurezza nel codice compilato).

Integrità del software e dei dati nei sistemi web	
Minaccia	Attacchi all'integrità dei sistemi (software e configurazioni).
Contromisure	Il web server deve proteggere il software, i dati e le informazioni memorizzate sul sistema con meccanismi appropriati per garantire un alto livello di integrità attraverso l'uso di firma digitale o MAC (message authentication codes).

Hardening del sistema operativo che ospita la piattaforma web	
Minaccia	- Accesso non autorizzato al sistema. - Compromissione delle comunicazioni. - Furto di credenziali di autenticazione (es. keylogger). - Violazione della sicurezza, rispetto alle politiche di sicurezza dell'organizzazione. - Violazione di leggi, di regolamenti, di obblighi contrattuali.