

La sicurezza informatica nella Pubblica Amministrazione

Webinar 4: Linee Guida «Sviluppo Software Sicuro»

02/12/2021

Linee Guida “Sviluppo Software Sicuro”

Michele Petito, AGID

Linee guida - *Sviluppo del software sicuro*

Introduzione

- **Le linee guida per lo sviluppo del software sicuro** nella pubblica amministrazione si inseriscono nel contesto delle linee guida per la sicurezza ICT delle Pubbliche amministrazioni, aventi lo scopo di fornire indicazioni sulle misure da adottare in ciascuna componente della Mappa del Modello strategico del Piano Triennale
- **L'obiettivo è quello di pervenire a un'architettura della sicurezza** per servizi sia critici che non critici, che definisca i principi e le linee guida del modello architetturale di gestione dei servizi e contestualizzazione rispetto al cluster dei dati gestiti. La sicurezza informatica ha un'importanza fondamentale in quanto oltre ad essere fondamentale per garantire disponibilità, integrità e riservatezza delle informazioni proprie del Sistema informativo della Pubblica amministrazione, è direttamente collegata ai principi di privacy previsti dall'ordinamento giuridico

Linee guida - "Sviluppo del software sicuro"

Struttura del documento

Le linee guida contengono quattro allegati tecnici relativi alle seguenti tematiche:

- **Allegato 1** - *Linee Guida per l'adozione di un ciclo di sviluppo di software sicuro*
- **Allegato 2** - *Linee Guida per lo sviluppo sicuro di codice*
- **Allegato 3** - *Linee Guida per la configurazione per adeguare la sicurezza del software di base*
- **Allegato 4** - *Linee Guida per la modellazione delle minacce e individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design*

Dove si trovano

1) Sito Agid

AGID | Agenzia per l'Italia digitale

Seguici su [f](#) [t](#) [v](#) [m](#)

Cerca nel sito

Agenzia ▾ Piattaforme ▾ Infrastrutture ▾ Sicurezza ▾ Dati ▾ Design servizi ▾ **Linee guida**

Home > Linee guida

2) Sito Cyber Risk Management

sicurezzait.gov.it/cyber/

Linee guida per lo sviluppo del software sicuro

Le linee guida per lo sviluppo del software sicuro nella pubblica amministrazione si inseriscono nel contesto delle linee guida per la sicurezza ICT delle PA, in piena coerenza le indicazioni contenute nel Piano Triennale per l'Informatica nella Pubblica Amministrazione.

Le Linee guida forniscono indicazioni sulle misure da adottare in maniera conforme al Piano Triennale: dalla fase di progettazione del software alla fase di rilascio.

Sicurezza Informatica

- [Allegato 4 - Linee Guida per la modellazione delle minacce e individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design \(2.04 MB\)](#)
Data: 07/05/2020
Pagina del sito: [Linee guida per lo sviluppo del software sicuro](#)
- [Allegato 3 - Linee Guida per la configurazione per adeguare la sicurezza del software di base \(1.57 MB\)](#)
Data: 07/05/2020
Pagina del sito: [Linee guida per lo sviluppo del software sicuro](#)
- [Allegato 2 - Linee Guida per lo sviluppo sicuro di codice \(5.11 MB\)](#)
Data: 07/05/2020
Pagina del sito: [Linee guida per lo sviluppo del software sicuro](#)
- [Allegato 1- Linee Guida per l'adozione di un ciclo di sviluppo di software sicuro \(9.86 MB\)](#)
Data: 07/05/2020
Pagina del sito: [Linee guida per lo sviluppo del software sicuro](#)

Allegato 1

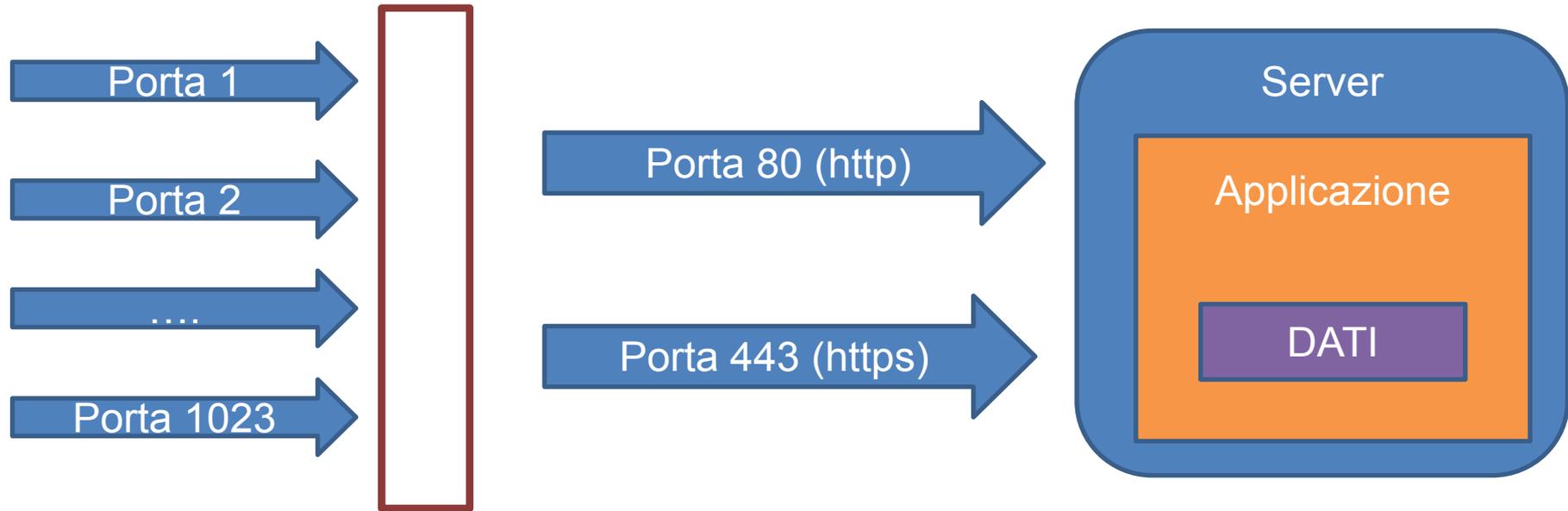
Cap. 4.1 - Il panorama delle vulnerabilità applicative

- Secondo la fonte **Gartner**, negli ultimi tempi **OLTRE IL 75% DEGLI ATTACCHI SONO STATI INDIRIZZATI DIRETTAMENTE VERSO LE APPLICAZIONI.**
- Gli **obiettivi** degli **attacchi** sono le **vulnerabilità** che si celano all'interno delle applicazioni software che forniscono un facile percorso d'ingresso per compromettere i sistemi o lanciare ulteriori attacchi e malware.

Gartner: <https://www.gartner.com/en>

OWASAP: https://www.owasp.org/index.php/Main_Page

Attacchi alle applicazioni



Allegato 1

Cap. 4.1 - Il panorama delle vulnerabilità applicative



Anche la comunità **OWASP**¹ sottolinea la necessità di accrescere la consapevolezza sulla sicurezza delle applicazioni in quanto il SW non sicuro mette a repentaglio le infrastrutture anche più critiche (finanziarie, sanitarie e difensive).

Vulnerabilities By Year



¹ OWASAP: https://www.owasp.org/index.php/Main_Page

² Numero di vulnerabilità per anno - CVE Details: <https://www.cvedetails.com/>

Statistiche Infosec



CERT-AGID
Computer Emergency Response Team
AGID



AGID Agenzia per
l'Italia Digitale

Seguici su 

cerca nel sito 🔍

Documentazione ▾
Chi siamo
Contatti
Strumenti
Glossario ▾
Link utili ▾

Trend Malware

Del 12/01/2021

Emotet

Notizie Vulnerabilità Malware Data breach

CMS HTTPS
18/12/2020

Monitoraggio sul corretto utilizzo del protocollo HTTPS e dei livelli di aggiornamento delle versioni dei CMS nei portali Istituzionali della PA

I servizi digitali erogati dalla Pubblica Amministrazione sono cruciali per il funzionamento del sistema Paese. La minaccia cibernetica cresce continuamente in quantità e qualità, determinata anche dall'evoluzione delle tecniche di ingegneria sociale volte a ingannare gli utenti finali dei servizi digitali sia interni alla PA che fruitori dall'esterno. L'esigenza per la PA di contrastare tali [...]

Statistiche Infosec

- 154847 Vulnerabilities
- 971 CWE(s)
- 463 CAPEC(s)
- 1277 MS Patches
- 38803 Malwares
- 941048 IoC in 19 Blocklists

File	AV
gks-11169-47104	N/A
qd7o1-fl-5139	N/A
22498995-nKH...	N/A
live.php	N/A
x86	N/A
30hCP65	N/A
90652	N/A
lkgdo8n23944...	N/A
djsv1ay8	N/A
sehtd4ixav	N/A

Trend CVE

Del 12/01/2021

- CVE-2020-13559
- CVE-2020-16146
- CVE-2020-23631
- CVE-2020-24025
- CVE-2020-24027

PEC sLoad
11/01/2021

Campagna di malspam sLoad via PEC sfrutta allegato malevolo con doppio livello di compressione ZIP

È stata rilevata in data odierna, ed è ancora in fase di contrasto, una campagna malspam sLoad a tema "Pagamenti" che ha come oggetto Fattura [RAGIONE SOCIALE].La campagna è veicolata via PEC verso altri utenti PEC e fa uso di un allegato ZIP contenente un ulteriore file ZIP. All'interno del secondo ZIP sono presenti due [...]

Sintesi riepilogativa delle campagne malevole della settimana 08/01/2021

In questa settimana, il CERT-AGID ha riscontrato ed analizzato, nello scenario italiano di suo riferimento, un totale di 17 campagne malevole attive, di cui 1 generica veicolata anche in Italia e 16 con obiettivi italiani, mettendo così a disposizione dei suoi enti accreditati i relativi 129 indicatori di compromissione (IOC) individuati. Riportiamo in seguito il dettaglio delle tipologie illustrate nei grafici, risultanti dai [...]

Cert-AgID: <https://www.cert-agid.gov.it>

Allegato 1

Cap. 4.2 - Defense-in-Depth

- Allo scopo di proteggere un sistema informativo, è pertanto necessario che ogni sua componente disponga di un proprio meccanismo di protezione.
- La costruzione di **strati multipli di controlli di sicurezza** posti lungo un sistema è definita Defence in Depth.
- La **Defense-in-Depth** è l'approccio alla sicurezza delle informazioni che prevede il raggiungimento di un adeguato livello di sicurezza attraverso l'utilizzo coordinato e combinato di molteplici contromisure. Questa strategia difensiva si fonda sull'integrazione di differenti categorie di elementi: persone, tecnologie e modalità operative.
- La ridondanza e la distribuzione delle contromisure possono essere sintetizzate in una "difesa a differenti livelli" ("**Layered Defenses**"). Il concetto è di derivazione militare e si basa sull'assunto che nel caso in cui un attacco abbia successo, a causa del fallimento di un meccanismo di sicurezza, **altri meccanismi di sicurezza** possono intervenire per consentire un'adeguata protezione dell'intero Sistema.

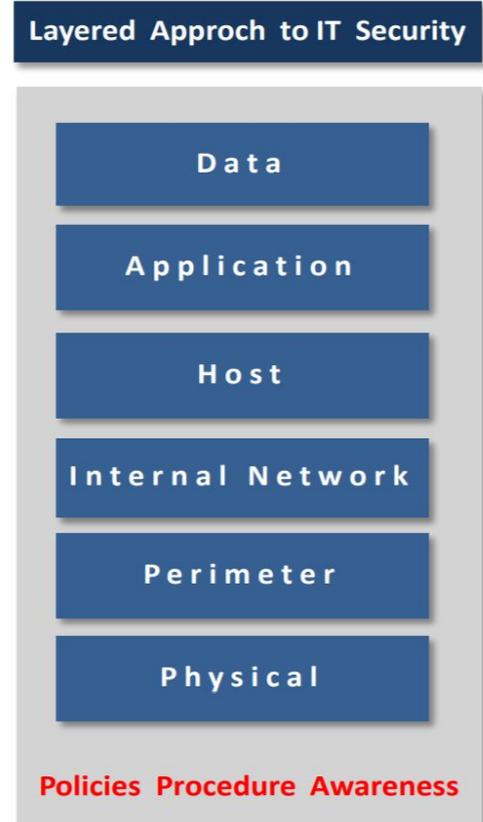
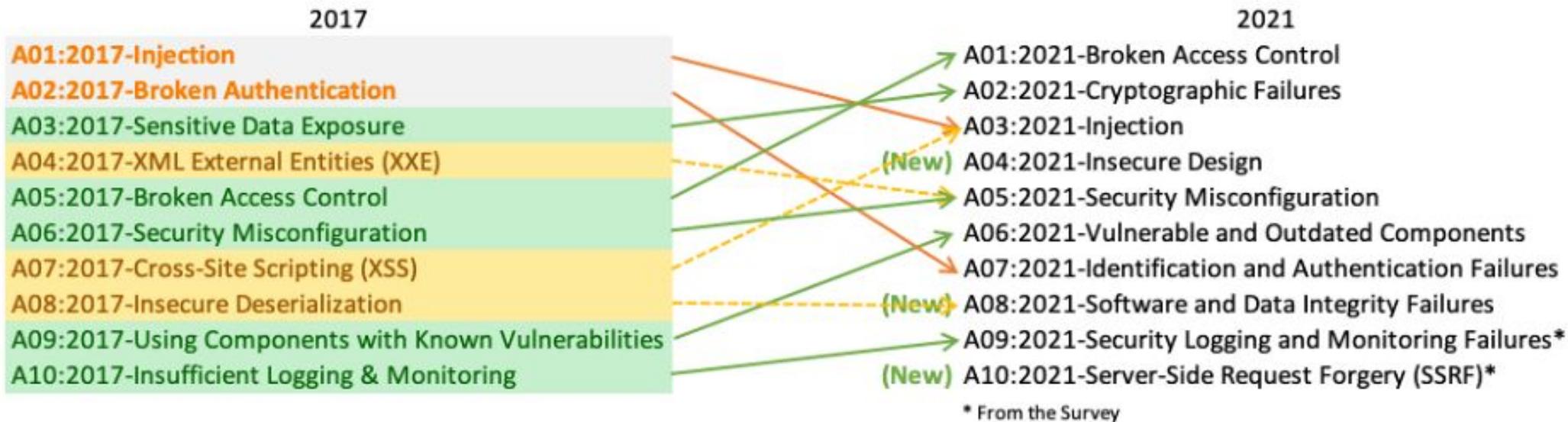


Figura 1 - Defence-in-Depth model for IT

OWASP Top 10



¹ OWASP Top 10: <https://owasp.org/www-project-top-ten>

Allegato 1

Cap. 8 - Secure Software Development Life Cycle (SSDLC)

Diverse sono le iniziative che si sono incentrate sulle problematiche **Secure Development** promuovendo azioni di sensibilizzazione (indirizzate ad aziende e community di sviluppatori) quali:

- la **diffusione delle fondamentali best practices** in materia di sicurezza applicativa (le prime tra tutte riconducibili ad una buona ingegnerizzazione del software);
- **una piena comprensione delle minacce più comuni** (compresi i difetti propri dei linguaggi di programmazione);
- ancora più importante, una **considerazione della problematica fin dalle prime fasi del ciclo di sviluppo**.

L'adozione di un *Secure Software Development Life Cycle* (SSDLC) atto a considerare ed implementare opportune attività di sicurezza nel corso di tutte le sue fasi del ciclo di vita del SW, dalla analisi alla progettazione, sviluppo, test fino alla manutenzione **è una necessità inderogabile** per rispondere alla domanda di sicurezza e per ridurre i costi che comporta trascurarla.

Allegato 2

«*Linee Guida per lo sviluppo di software sicuro*»

- Contiene **best practices da seguire**, al fine **prevenire eventuali problematiche di sicurezza nel codice**
- Fornisce nel contempo uno **strumento utile nell'individuazione di possibili vulnerabilità presenti nel codice** sorgente e le relative contromisure da applicare

Allegato 2

Capitolo 4: Sviluppare applicazioni sicure

La sicurezza informatica, di un'applicazione è il **risultato delle contromisure di sicurezza applicate, nelle diverse fasi che compongono un qualsiasi ciclo di sviluppo adottato**, per ogni livello fisico e logico dell'applicazione stessa.

La figura mostra, a titolo di esempio non esaustivo, uno schema di **modellazione concettuale degli elementi principali che intervengono in tale processo** e sui quali s'indirizzano le linee guida

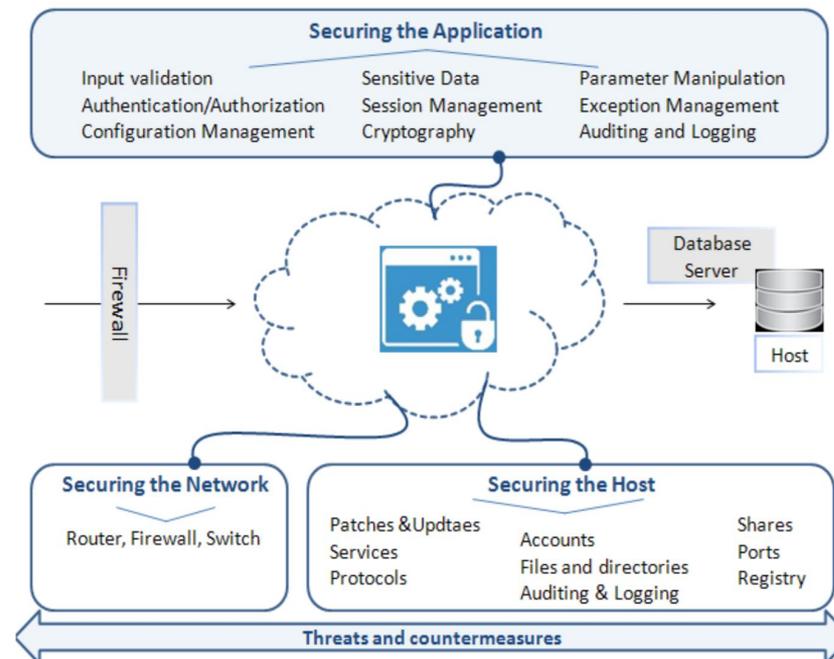


Figura 1: Schema per la sicurezza dell'applicazione

Allegato 2

Capitolo 5: Progettazione e Sviluppo dell'applicazione: Direttive Standard

L'architettura dell'applicazione deve essere progettata e sviluppata secondo i paradigmi standard dell'industria del software, quali: **Architettura monolitica (mainframe)**, **Client / server**, **Service Oriented Architecture (SOA)**, ecc.

Nel corso della fase di progettazione è **necessario garantire un adeguato livello di sicurezza applicativa e infrastrutturale attraverso l'analisi e la modellazione:**

- **delle minacce** inerenti gli applicativi coinvolti;
- **delle interfacce** e degli **agenti** che potrebbero minacciare il sistema.;

Per l'analisi della sicurezza applicativa di una architettura di sistema si adotta un approccio differente a seconda che si tratti di progettazione di applicazioni ex-novo (**approccio Secure by Design**) piuttosto che di reingegnerizzazione di applicazioni esistenti (**approccio Security Control**).

Allegato 2

5.2 - Sviluppo dell'applicazione – Criteri Generali

5.2.1 Performance: Le soluzioni di programmazione impiegate devono ridurre al minimo l'impatto sulle risorse di sistema.

5.2.2 Password: nel codice sorgente I dati di accesso (username/password/nome db/ecc..) ai database o a sistemi di altra natura non devono mai essere inseriti all'interno dei sorgenti.

5.2.3 Privilegi esecutivi minimi: L'applicazione, in esecuzione, non deve utilizzare privilegi amministrativi

5.2.4 Metodo TRACE: Nelle applicazioni Web è obbligatoria la disattivazione lato server del metodo HTTP TRACE.

5.2.5 Assenza di codice malevolo: L'applicazione sviluppata non deve includere al suo interno alcun Trojan horse, spyware o più in generale alcuna componente malware.

5.2.6 Fattore integrità: L'implementazione (e la precedente fase di progettazione) dell'applicazione devono assicurare che: tutti gli errori e le eccezioni insorti durante la fase di processamento ed elaborazione dei dati acquisiti in ingresso, siano correttamente gestiti e non causino il danneggiamento o la perdita di integrità delle informazioni conservate e mantenute dall'applicazione stessa.

5.2.7 Input data validation: L'applicazione deve assicurare, attraverso opportuni meccanismi di convalida, che tutti i parametri in input, specificati dall'utente, siano congruenti a quanto atteso.

5.2.8 Gestione dell'output: L'applicazione deve fornire in output solamente le informazioni rilevanti all'uso delle richieste avanzate dagli utenti, rendendo vana la possibilità di qualsiasi tipo d'information gathering o disclosure non autorizzato

Allegato 2

5.3 – *Formattazione del codice*

5.3.1 Stile e sintassi: la dichiarazione di ogni funzione, metodo o classe deve sempre precedere un commento che riporti scopo della funzione, parametri di input/output, valori di ritorno, tracciamenti delle modifiche al codice (data ultima modifica)

5.3.2 Stile e sintassi: nell'ottica di rendere l'applicazione conforme agli standard internazionali è richiesto l'utilizzo esclusivo di algoritmi riconosciuti nell'industria del software

5.8 – *Password chiavi e certificati*

5.8.1 Gestione delle password, chiavi e certificati: le password mantenute dall'applicazione o le chiavi private dei certificati devono essere conservate in forma cifrata e devono risiedere in un'area di file system o di database differente rispetto a quella dell'applicazione.

5.8.3 Generazione/conservazione delle password nel filesystem/DB: le password memorizzate nel filesystem o nel DB sotto forma di hash (esempio MD5/SHA-1 etc.), devono prevedere l'introduzione di un ulteriore fattore randomico (salt) durante la loro generazione.

Allegato 2

Capitolo 7: Best practice per lo sviluppo in sicurezza

7.2.1 Cross-site scripting (XSS):

- **Reflected XSS.** Si tratta di inoculare e far eseguire script dannosi all'interno di una pagina web attraverso la contraffazione dell'input

- **Stored XSS.** In questo caso il codice HTML o lo script incorporato attraverso l'input viene memorizzato permanentemente sulla pagina e diventa parte integrante di essa. Tutti gli utenti della pagina web saranno potenzialmente vittime (es. blog che consente di inserire commenti in html)

Come difendersi: convalidare tutti gli input in base a una whitelist di valori ammessi. Controllare anche dimensione, tipo e formato dei dati in input. Codificare tutti i dati dinamici (encoding). La codifica dovrebbe essere sensibile al contesto. Definire in modo esplicito la codifica dei caratteri (charset) per l'intera pagina.

Esempio:

Nel codice che segue, un valore preso dalla request viene scritto direttamente sulla response:

```
String nomeUtente = request.getParameter("nome");  
response.getWriter().write("Nome Utente: " + nomeUtente);
```

Il rimedio consiste nel filtrare il valore in input:

```
response.getWriter().write(ESAPI.encoder().encodeForHTML  
( request.getParameter( "nome" ) ));
```

Allegato 2

Capitolo 7: Best practice per lo sviluppo in sicurezza

7.2.2 – Code Injection:

Quando accade: quando l'applicazione utilizza, concatenandole, stringhe in input non bonificate.

Conseguenze: L'attaccante potrebbe introdurre script che potrebbero essere eseguiti direttamente nell'application server

Come difendersi: evitare di eseguire codice dinamicamente a partire da input esterno. Verificare sempre l'input impedendo l'inserimento di caratteri potenzialmente dannosi. Designare una white list di valori ammessi.

```
public class CodeInjection {
    static void main(String[] args) {
        System.load(args[0]);
    }
}
```

code fix

```
public class CodeInjectionFixed {
    static void main(String[] args){
        String fileName = null;
        switch(args[0]){
            case "First":
                fileName="First.txt";
                break;
            [ALTRI CASE].....
            default :
                fileName="none.txt";
                System.load(fileName);}}
}
```

Allegato 3

«LG per adeguare la sicurezza del software di base»

Scopo del documento

- **La sicurezza del software di base ed applicativo** richiede di stabilire un processo volto ad identificare rischi e contromisure di sicurezza ad ogni livello (fisico, logico e organizzativo) del contesto in cui tali software operano e sono utilizzati.
- **La configurazione sicura di tali software** (nel seguito tale attività viene spesso indicata con il termine “**hardening**”), è necessario considerare vari elementi, quali le **protezioni perimetrali** (fisiche e logiche), le **architetture di rete** (DMZ, segmentazioni, etc.), le **procedure organizzative** (perché dietro alle tecnologie operano le persone), i **programmi formativi** di “security awareness”, ecc. Partendo da questo presupposto, il presente allegato si pone l’obiettivo di fornire un insieme di indicazioni per affrontare e risolvere correttamente le problematiche legate alla sicurezza del software di base e di individuare le misure da adottare per difendere ogni componente da possibili minacce accidentali e/o intenzionali.

Allegato 3

Struttura del documento

I singoli paragrafi entrano nel dettaglio delle singole componenti (**software di base, middleware, office automation, ecc.**) oggetto di approfondita analisi dal punto di vista delle best practice di sicurezza, e per ognuna forniscono un elenco delle misure di sicurezza da adottare a fronte delle principali minacce, in modo da diminuire l'esposizione ai rischi per la sicurezza delle informazioni e dei servizi erogati.

- Il **paragrafo 4.1** contiene un **elenco delle minacce alla sicurezza** delle informazioni ritenute applicabili nel contesto del presente documento.
- Il **paragrafo 4.2** contiene un **catalogo delle principali tipologie di attacco** rispetto al software di base, al middleware e al software applicativo più comune.
- Il **paragrafo 5.1** fornisce un **insieme di raccomandazioni generali 'trasversali'** che realizzano la base comune per affrontare le problematiche di sicurezza delle specifiche componenti.
- Il **paragrafo 6** contiene in una prima tabella l'**elenco dei riferimenti alle istruzioni operative di hardening** (o benchmarks) messe a disposizione da enti/istituzioni preposte ed affermate a livello internazionale, operanti con il pieno supporto dei rispettivi vendor, e in una seconda tabella l'elenco delle baseline di configurazione e di alcuni strumenti software per l'hardening messi a disposizione direttamente dai vendor.

Allegato 3

Cap. 5 - best practice per la sicurezza del software di base (1/2)

Vulnerability Assessment

Minaccia	<ul style="list-style-type: none"> - Accesso non autorizzato ai sistemi (risorse di sistema, configurazioni, interfacce amministrative, ecc.). - Accesso non autorizzato alle informazioni; - Compromissione delle comunicazioni. - Divulgazione di informazioni riservate. - Negazione dei servizi. - Cancellazione o furto di informazioni. - Attacchi all'integrità dei sistemi (software e configurazioni). - Attacchi all'integrità delle informazioni. - Uso non autorizzato di privilegi. - Violazione della sicurezza, rispetto alle politiche di sicurezza dell'organizzazione. - Violazione di leggi, di regolamenti, di obblighi contrattuali.
Contromisure	<p>Effettuare almeno una volta l'anno un'attività di Vulnerability Assessment (VA) in modo da identificare le eventuali vulnerabilità che possono costituire un canale di accesso non autorizzato ad informazioni. Il VA deve verificare la corretta configurazione delle porte logiche affinché siano attive solo quelle strettamente necessarie. Inoltre, l'attività di VA deve essere condotta avendo come riferimento le ultime vulnerabilità note, pubblicate nelle banche dati di riferimento in tema di sicurezza informatica come, ad esempio, Open Source Vulnerability DataBase (OSVDB) e Common Vulnerabilities Exposures (CVE). In seguito all'attività di VA, per mitigare il rischio associato alle vulnerabilità identificate è necessario:</p> <ul style="list-style-type: none"> - definire i ruoli e le responsabilità per la gestione delle vulnerabilità tecniche; - identificare le azioni da intraprendere (es. disabilitare le funzionalità non utilizzate, inclusi protocolli e servizi; rendere più sicure le impostazioni di configurazione predefinite, ridurre al minimo il numero delle interfacce di amministrazione, ecc.); - revisionare le funzionalità di failover del sistema.

Backup delle informazioni e del software

Minaccia	<ul style="list-style-type: none"> - Negazione dei servizi. - Cancellazione o furto di informazioni.
Contromisure	<p>Effettuare periodicamente copie di backup delle informazioni e del software, in conformità alla politica per il salvataggio dei dati stabilita a livello aziendale. Verificare periodicamente, nel rispetto delle leggi, regolamenti, obblighi contrattuali, l'effettiva memorizzazione, "leggibilità" e integrità delle informazioni registrate, anche al fine di assicurare la pronta disponibilità delle stesse in caso di interruzione dei servizi informativi. Individuare le responsabilità per la gestione delle copie di backup.</p>

Allegato 3

Cap. 5 - best practice per la sicurezza del software di base (2/2)

Hardening della suite Office

Minaccia	<ul style="list-style-type: none"> - Accesso non autorizzato alle informazioni. - Attacchi all'integrità dei sistemi. - Violazione della sicurezza, rispetto alle politiche di sicurezza dell'organizzazione (malware).
Contromisure	<p>Limitare/Disabilitare/Condizionare l'uso di contenuti attivi.</p> <p>Per contenuti attivi si intendono:</p> <ul style="list-style-type: none"> - i controlli ActiveX, - i componenti aggiuntivi quali, ad esempio: <ul style="list-style-type: none"> • Componenti aggiuntivi COM (Component Object Model) • Componenti aggiuntivi Visual Studio Tools per Office (VSTO) • Componenti aggiuntivi di automazione • Server RTD (RealTimeData) • Componenti aggiuntivi di applicazioni, ad esempio file con estensioni wll, xll e xlam • Pacchetti di espansione XML • Fogli di stile XML • Macro VBA
Riferimenti	<ul style="list-style-type: none"> - Pianificare le impostazioni di sicurezza per i controlli ActiveX in Office 2013, https://technet.microsoft.com/it-it/library/cc179076.aspx - Pianificare le impostazioni di protezione per i componenti aggiuntivi per Office

Hardening del browser: configurazione di base per la sicurezza

Minaccia	<ul style="list-style-type: none"> - Attacchi all'integrità dei sistemi. - Cancellazione o furto di informazioni (accidentale o da attacchi come ad es. il ransomware, ecc.). - Violazione della sicurezza, rispetto alle politiche di sicurezza dell'organizzazione (es. malware, ecc.).
Contromisure	<p>La configurazione di default per molti browser web non è sicura. Si raccomandano i passaggi a seguire per rendere maggiormente sicuro il browser web in uso. Tali impostazioni assumono particolare importanza nel caso in cui si utilizza il browser per accedere a sistemi aziendali o più in generale se si utilizza il browser per accedere, inviare o ricevere informazioni sensibili.</p> <ul style="list-style-type: none"> - Impostare il browser di default: <ul style="list-style-type: none"> o Firefox: sia per Mac che per Windows - andare nel menu Firefox > Preferenze (Mac) Opzioni (Windows) > scheda Generale. Selezionare la casella "Controlla sempre se Firefox è il browser predefinito". o Safari: andare nel menu Safari > Preferenze > scheda Generale e clicca sul pulsante "Imposta predefinito....". o Internet Explorer: si raccomanda di non utilizzare IE come browser predefinito. o Google Chrome: andare sulle impostazioni nella sezione "Browser predefinito" e cliccare sul pulsante "Imposta come predefinito" in corrispondenza della voce "Imposta Google Chrome come browser predefinito". - Mantenere il software del browser aggiornato. - Abilitare nel browser gli aggiornamenti automatici e mantenerli in tale stato: <ul style="list-style-type: none"> o Firefox: sia per Mac che per Windows - vai al menu Firefox >

Allegato 4

«Linee Guida per la modellazione delle minacce ed individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy By Design»

Struttura del documento

Il documento è articolato come segue:

- Il **Capitolo 5**, introduce i concetti base di **security e privacy by design**, analizza gli strumenti e i modelli a supporto della fase di progettazione del software sicuro, in essere e in divenire;
- Il **Capitolo 6**, definisce le linee guida per l'**identificazione preventiva delle possibili minacce**, delle relative azioni di mitigazione e per la valutazione e prioritizzazione delle minacce stesse;
- Il **Capitolo 7**, fornisce un **caso d'uso applicativo (Easy Web Site)** in cui vengono impiegate le metodologie e gli strumenti di sicurezza indicati nel capitolo 5.

Allegato 4

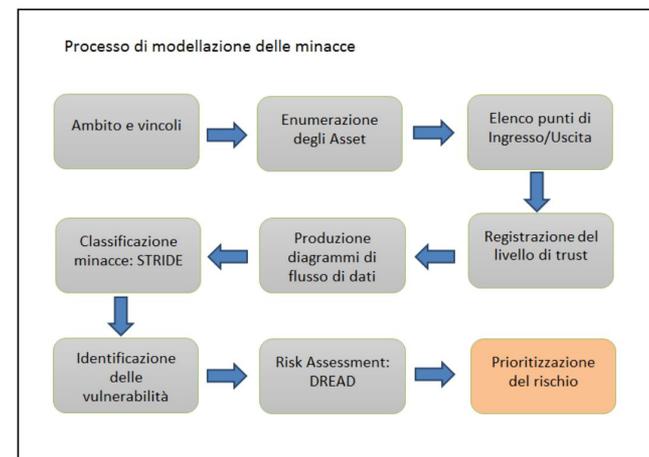
5.1 – *Processi di sviluppo del software sicuro*

- BSA for Secure Software (**BSA**) -
 - framework sviluppato dalla statunitense Software Alliance BSA4
- Software Assurance Maturity Model (**SAMM**)
 - framework supportato da OWASP
- Building Security in Maturity Model (**BSIMM**)
- Comprehensive and Lightweight Application Security Process (**CLASP**)
- Microsoft's Security Development Lifecycle (**SDL**)

Allegato 4

5.1.5 - Microsoft's Security Development Lifecycle (SDL)

- Introduzione del **Threat Modeling**
- Applicabile a metodologie come **Waterfall** e **Agile**, qualsiasi piattaforma e organizzazione
- **STRIDE**: modello di modellazione delle minacce
- **DREAD**: analisi quantitativa del rischio



Allegato 4

5.2: Secure by Design

Definisce un software progettato per essere sicuro e capace di garantire riservatezza, integrità e disponibilità.

- Alcuni **principi del Secure by Design**:
 - *Ridurre al minimo la superficie d'attacco*
 - *Stabilire valori predefiniti sicuri*
 - *Non affidarsi ai servizi di terze parti*
 - *Separare i ruoli*
 - *Segmentare l'infrastruttura di rete*
 - *Ridurre i single point of failure*

Allegato 4

5.2.2 - Altre best practices

- **5.2.2.1** Best practices di secure design per le applicazioni web
- **5.2.2.2** Best practices di secure design per il cloud
- **5.2.2.3** Best practices di secure design per le architetture serverless
- **5.2.2.4** Best practices di secure design per le architetture basate su registri distribuiti (DLT)

Allegato 4

5.2.2.4 - Best practices di secure design per le architetture basate su registri distribuiti (DLT)

DLT = Distributed Ledger Rechnology = Blockchain

- **Tipo di attacchi su DLT:**
 - Il block withholding attack (BWH)
 - Selfish mining
 - Fork after withholding (FAW)
 - **Il famoso attacco del 51%**
 - eclipse attack'
 - consensus delay
 - Finney Attack
- **La maggioranza degli attacchi è orientata verso le applicazioni (smart contracts) che girano sulla DLT e non sulla chain stessa**

Allegato 4

Cap. 6.1.1 Identificazione degli obiettivi di sicurezza (1/2)

- Gli obiettivi specifici di sicurezza sono un sottoinsieme degli obiettivi di progetto e dovrebbero essere utilizzati per guidare gli sforzi impiegati nella modellazione delle minacce. Identificare i principali obiettivi di sicurezza permette di concentrarsi con maggiore attenzione sulle aree da proteggere.
- Ad esempio, se si identificano i dettagli del profilo cliente come dati riservati, che devono essere protetti, è possibile esaminare la modalità di archiviazione sicura di tali dati e il modo in cui l'accesso a tali dati viene controllato e verificato.

Allegato 4

Cap. 6.1.1 Identificazione degli obiettivi di sicurezza (2/2)

Per determinare gli obiettivi di protezione, occorre porsi le seguenti domande:

- **Quali dati occorre proteggere?**
- **Esistono requisiti di conformità?** I requisiti di conformità possono includere criteri di protezione, leggi sulla privacy, regolamenti e standard.
- **Esistono requisiti di qualità specifici del servizio?** I requisiti di qualità del servizio includono tipicamente la disponibilità e i requisiti prestazionali.
- **Esistono beni immateriali che devono essere protetti?** Tali beni includono ad esempio, la reputazione dell'organizzazione, le informazioni commerciali sensibili e la proprietà intellettuale.



AGID

Agenzia per l'Italia Digitale

FormezPA

GRAZIE PER L'ATTENZIONE!

 petito@agid.gov.it

