### **Tecniche Comuni di Social Engineering**

1. **Phishing**
   * **Descrizione**: Email, messaggi o siti web fraudolenti progettati per sembrare legittimi e indurre la vittima a fornire credenziali, dati bancari o altre informazioni sensibili.
   * **Varianti**:
     + **Spear Phishing**: Personalizzato per la vittima specifica.
     + **Whaling**: Targetizzazione di figure di alto profilo (es. CEO).
     + **Smishing**: Attraverso SMS o messaggi istantanei.
     + **Vishing**: Attraverso chiamate vocali.
2. **Pretexting**
   * **Descrizione**: Creazione di una storia fittizia per ottenere informazioni private. L’attaccante si spaccia spesso per un'autorità o un collega.
   * **Esempi**: Richiesta di informazioni di login spacciandosi per il reparto IT.
3. **Baiting**
   * **Descrizione**: Offerta di un’esca per attirare la vittima, come dispositivi USB infetti lasciati in luoghi pubblici o file promettenti scaricabili.
   * **Esempio**: Una penna USB etichettata come “stipendi 2024”.
4. **Tailgating e Piggybacking**
   * **Descrizione**: Accesso fisico non autorizzato sfruttando la cortesia altrui. Ad esempio, seguire qualcuno in un'area riservata senza un badge.
5. **Quid Pro Quo**
   * **Descrizione**: Offrire un vantaggio (finto) in cambio di informazioni o accesso.
   * **Esempio**: Un attaccante che si spaccia per un tecnico IT e offre "assistenza gratuita".
6. **Dumpster Diving**
   * **Descrizione**: Raccolta di informazioni utili rovistando nella spazzatura, cercando documenti, fatture o dispositivi non distrutti.
7. **Impersonation**
   * **Descrizione**: L'attaccante si finge un dipendente, un fornitore o un amico per ottenere fiducia.

### **Strategie di Contrasto**

Per difendersi efficacemente dagli attacchi di social engineering, è necessario adottare una combinazione di misure tecniche, organizzative e comportamentali. Ecco alcune strategie e migliori pratiche dettagliate:

### **1. Educazione e Formazione del Personale**

* **Sensibilizzazione regolare**: Organizza sessioni di formazione periodiche per aumentare la consapevolezza sui rischi del social engineering.
* **Simulazioni pratiche**: Conduci test di phishing simulati per allenare il personale a riconoscere attacchi reali.
* **Condivisione di esperienze**: Racconta casi di studio o incidenti reali per dimostrare l’impatto degli attacchi.

### **2. Miglioramento della Sicurezza delle Comunicazioni**

* **Verifica delle fonti**: Insegna a verificare sempre l’identità del mittente di email, messaggi e chiamate.
  + Es. Confermare una richiesta direttamente con il collega o il fornitore attraverso un numero di telefono ufficiale.
* **Segnalazione rapida**: Incoraggia la segnalazione di messaggi sospetti al team di sicurezza.

### **3. Tecnologie di Prevenzione**

* **Filtri anti-phishing**: Utilizza soluzioni software che analizzano email e messaggi per individuare tentativi di phishing (SPF, DKIM e DMARC) .
* **Multi-Factor Authentication (MFA)**: Implementa MFA per proteggere l'accesso a sistemi e dati, rendendo inutile il solo furto delle credenziali.
* **Blocchi su USB e hardware sconosciuti**: Disabilita le porte USB o limita l’utilizzo di dispositivi non autorizzati.

### **4. Definizione di Politiche Aziendali**

* **Minimizzazione dei privilegi**: Applica il principio del “minimo privilegio”, garantendo accesso solo alle risorse necessarie per il lavoro.
* **Gestione dei dati**: Stabilisci regole chiare per il trattamento e lo smaltimento di informazioni sensibili.
  + Es. Uso di tritacarte per documenti e distruzione sicura dei dischi rigidi.

### **5. Protezione Fisica**

* **Controlli di accesso**:
  + Usa badge, codici o autenticazione biometrica per limitare l'accesso a edifici o aree sensibili.
* **Sicurezza dei dispositivi**:
  + Non lasciare mai laptop o dispositivi incustoditi in luoghi pubblici.
* **Prevenzione del tailgating**: Insegna ai dipendenti a non permettere l’accesso a persone senza badge anche se sembrano autorizzate.

### **6. Strategie di Verifica**

* **Verifica telefonica o email**: Prima di fornire informazioni sensibili, verifica sempre con un canale secondario noto e affidabile.
* **Utilizzo di password temporanee (OTP)**: Per le transazioni sensibili, richiedi verifiche aggiuntive come OTP o PIN.

### 

### **7. Creazione di una Cultura della Sicurezza**

* **Promuovi il dubbio sano**: Insegna che è normale sospettare e verificare richieste insolite o urgenti.
* **Team dedicato alla sicurezza**: Nomina un team interno per rispondere rapidamente a segnalazioni e fornire supporto.

### **8. Monitoraggio e Audit**

* **Analisi comportamentale**: Usa sistemi di monitoraggio per individuare attività anomale.
* **Audit regolari**: Controlla l'efficacia delle misure di sicurezza e aggiornale regolarmente per contrastare nuove minacce.

### **Checklist Operativa**

* **Prima di cliccare su link o allegati**:
  + Verifica l’URL.
  + Controlla il mittente (email e nome).
  + Fai attenzione agli errori grammaticali o richieste urgenti.
* **Prima di condividere informazioni personali**:
  + Chiediti: questa persona ha il diritto di sapere?
  + Conferma la richiesta tramite un canale indipendente.

**CVE (Shopify)**

Per quanto riguarda la richiesta sulle CVE (Common vulnerabilities and exposures) ho chiesto a chat GPT di analizzare, elencarmi e darmi le best practices per evitare le CVE del software di Shopify, un software as service comunissimo che ha spopolato negli ultimi anni per la costruzione di e-commerce.

. Ecco una panoramica di alcune vulnerabilità recenti, con dettagli su come possono essere mitigate:

### **1. CVE-2024-45036 - Improper Access Control**

* **Descrizione**: Una vulnerabilità nel framework Tophat utilizzato per testare applicazioni mobile. Permette a un attaccante di sfruttare URL malevoli per ottenere il token TOPHAT\_APP\_TOKEN dal file di configurazione locale (~/.tophatrc). Questo token può essere usato per accedere a dati riservati come artefatti di build.
* **Mitigazione**: È stato rilasciato un aggiornamento alla versione 1.10.0 di Tophat che disabilita questo flusso insicuro. Gli utenti devono aggiornare immediatamente e invalidare eventuali token compromessi.
* **Rischio**: Accesso non autorizzato a risorse interne.

### 

### **2. CVE-2022-29230 - Cross-Site Scripting (XSS) in Hydrogen**

* **Descrizione**: Una vulnerabilità XSS nel framework Hydrogen, usato per creare storefront personalizzati, consente a un attaccante di eseguire script maligni su pagine che utilizzano dati controllati dall'utente.
* **Mitigazione**: L'aggiornamento alla versione 0.19.0 di Hydrogen risolve il problema. Non esistono workaround; è necessario aggiornare immediatamente.
* **Rischio**: Compromissione di dati e controllo dell'applicazione da remoto.

### **3. CVE-2020-8176 - XSS in Koa-Shopify-Auth**

* **Descrizione**: Una vulnerabilità in koa-shopify-auth consente l'iniezione di codice malevolo tramite il parametro shop nell'endpoint /shopify/auth/enable\_cookies.
* **Mitigazione**: La vulnerabilità è stata risolta aggiornando alla versione successiva. L'implementazione di filtri di input e output più stringenti è consigliata.
* **Rischio**: Esecuzione di codice JavaScript malevolo da remoto.

### **Migliori pratiche per la mitigazione generale:**

1. **Aggiornamenti regolari**: Applicare sempre le patch fornite dai vendor appena disponibili.
2. **Content Security Policy (CSP)**: Configurare una CSP rigorosa per limitare l'esecuzione di script non autorizzati.
3. **Audit del codice**: Controllare il codice delle applicazioni e delle librerie per vulnerabilità note.
4. **Monitoraggio attivo**: Implementare sistemi di monitoraggio per identificare comportamenti sospetti.
5. **Formazione**: Assicurarsi che i team di sviluppo siano informati su vulnerabilità comuni e best practice di sicurezza.

Queste misure, combinate con l'adozione di principi di sicurezza come "least privilege" e difesa in profondità, possono aiutare a ridurre significativamente i rischi di compromissione.

**Debian Linux**

### **1. CVE-2024-3094**

Questa vulnerabilità è stata scoperta nel pacchetto XZ Utils, incluso in versioni instabili di Debian e altre distribuzioni. Un codice malevolo è stato introdotto tramite una compromissione della supply chain, consentendo potenzialmente l'accesso remoto non autorizzato ai sistemi compromessi. Questo attacco sfrutta un backdoor nascosta nel processo di compilazione di alcune versioni del software.

**Mitigazione:**

* Aggiorna XZ Utils alla versione non vulnerabile (es., 5.4.6).
* Verifica e sostituisci le versioni compromesse.
* Cambia tutte le credenziali per precauzione se i sistemi sono stati esposti.
* Utilizza strumenti di rilevamento come le regole YARA per individuare attività sospette​

### **2. CVE-2024-1086**

Un'altra vulnerabilità critica è stata trovata nel componente netfilter del kernel Linux. La falla è legata a un errore di gestione della memoria (use-after-free) che può permettere un'escalation dei privilegi a livello locale, consentendo agli attaccanti di ottenere un accesso con diritti elevati.

**Mitigazione:**

* Aggiorna il kernel alla versione corretta disponibile nei repository ufficiali.
* Verifica i log di sistema per attività sospette, in particolare se hai visto anomalie nei processi legati a netfilter.
* Configura sistemi IDS/IPS per monitorare l'attività di rete e prevenire exploit in tempo reale​

### **Linee generali per prevenzione:**

1. **Monitoraggio attivo:** Mantieni il sistema aggiornato, utilizzando strumenti di gestione delle patch.
2. **Controllo delle dipendenze:** Verifica sempre la provenienza dei pacchetti software e, quando possibile, utilizza versioni pre-compilate fornite da fonti affidabili.
3. **Backup regolari:** Esegui backup frequenti per minimizzare i danni in caso di compromissione del sistema.
4. **Isolamento delle risorse:** Configura i sistemi con privilegi minimi e utilizza sandboxing per isolare i processi sensibili.