Il business preso in esame, con ricavo lordo annuo da applicativo e-commerce corrispondente a 788.400.000€ annui, necessita di soluzioni attive, passive e preventive di cyber sicurezza. Visto il fatturato e la mancanza di indicazioni esplicite nelle consegne, viene presupposto che il budget della sicurezza non sia un problema e conceda una larga gamma di implementazioni a tutti i livelli: Software, Hardware e Policy.

Il costo di un incidente che invalida il 100% di un asset stimato a 15.000 euro (1.500/minuto per 10 minuti) è di: 15.000 euro per evento | 0,285 euro per anno.

Prima di tutto, vedremo le misure preventive possibili per le tre problematiche evidenziate: SQLi, XSS e DDoS.

XSS:

- **Validazione dell'input utente:** Misura di Secure Software Development Lifecycle, consiste nella bonifica dell'input che arriva da qualsiasi host esterno verso il server che ospita l'applicativo. L'obiettivo è quello di precludere a qualsiasi utente non autorizzato la possibilità di inviare righe di testo interpretabili come codice, che possano potenzialmente aggirare misure di sicurezza ed isolamento di dati, o che possono fornire informazioni sensibili.
  Comunemente consiste nel filtro tramite RegEx di caratteri speciali o di sintassi comune ai linguaggi di web-development usati dall'applicativo da proteggere.

- **Codifica di input e output:** La codifica di plain text in output tramite algoritmi di cifratura moderni ed aggiornati previene l'interpretazione automatica insicura da parte degli applicativi. Una codifica in input inoltre previene l'insorgere di problemi di Data Injection tramite Intercepting Proxy per aggirare la validazione input descritta sopra.

- **Uso di Header appropriati per le risposte:** Una categorizzazione ed assegnazione corretta di header HTTP di "Content-Type" può controllare l'interpretazione da parte dei browser dei contenuti richiesti nei soli modi corretti e sicuri, impedendo l'esecuzione di codice HTML o JavaScript non autorizzate nelle risposte.
  L'uso di header di Content Security Policy limita il numero di script e risorse in scope nel codice eseguito e quindi la sicurezza complessiva della pagina web.

SQLi:

- **Validazione dell'input e codifica degli output**, come sopra, ma volte all'eliminazione di possibili iniezioni di query SQL.

- **Restrizione della base di codice:** Riducendo la superficie attaccabile tramite la disattivazione di funzionalità non strettamente necessarie, l'utilizzo di stored

procedures e il blacklisting di dichiarazioni SQL non validate a priori dagli sviluppatori o amministratori di sistema.

- **Restrizione dell'accesso al database:** Tramite soluzioni perimetrali di rete, come i Firewall, ed anche tramite una buona politica di categorizzazione degli utenti che ne hanno accesso, che vanno protetti da accessi non autorizzati tramite forti password e 2FA.

- **Proper OPSEC:** Non divulgazione di informazioni non necessarie in log, risposte e comunicazioni pubbliche/intercettabili. Valido anche come misura preventiva per una miriade di altri attacchi informatici e non solo.

DDoS:

- **Ridondanza delle infrastrutture di rete:** Una bilanciata distribuzione del traffico tramite più link di rete, e la conseguente elaborazione dei dati su più macchine server, consente di diluire notevolmente la minaccia di un attacco DDoS, che necessiterà sproporzionalmente di più risorse hardware per effettuare un attacco efficace.

- **Sicurezza perimetrale della rete:** Attiva e passiva tramite dispositivi automatici di monitoraggio del traffico della rete e di sicurezza perimetrale. Revisione periodica di log di rete e di sistema per notare eventuali pattern di attacco in anticipo.

- **Incident Response Planning:** Gli attacchi DDoS potrebbero necessitare di tempo per intasare infrastrutture aziendali complesse, il che potrebbe consentire al team interno di sicurezza di riconoscere la minaccia ed attuare manovre pianificate a priori volte a mitigare o annullare i danni di tali attacchi.

Nel caso invece le misure preventive abbiano fallito ed un server contenente il nostro applicativo venisse infettato da un malware, bisogna immediatamente agire, al rilevamento della minaccia da parte di sistemi automatici o utenti, all'isolamento della macchina dalle reti interne, possibilmente ristabilendo il servizio perduto su macchine ridondanti o di backup. La macchina infetta va prontamente disconnessa dalla rete, filtrata per MAC dai router interni, o relegata in una propria zona di quarantena. Una volta accertatosi che la rete interna non sia più raggiungibile, si può procedere a misure volte ad eliminare la minaccia o al ripristino della macchina da zero. A quest'ultima, una buona politica di backup viene ampiamente in soccorso.

Tuttavia, le politiche preventive giocano comunque un ruolo fondamentale sull'efficacia di un'operazione di Disaster Recovery. Ad esempio:

- **Segmentazione della rete:** In sottoreti circostanziali, i cui link sono protetti da dispositivi perimetrali e di end-point al fine di limitare gli spostamenti laterali ed agevolare l'implementazione della quarantena.
- **Sanificazione dei vettori d'attacco:** Tramite una revisione accurata di porte aperte, servizi e software che scambiano informazioni sulla rete.
- **Implementazione di linee guida** e best practice complete a cui attenersi per migliorare la sicurezza delle informazioni e la gestione dei rischi, come il NIST CSF
- **Infrastrutture di rete ridondanti e fault-tolerant.**

Il disegno proposto di rete, considerando tutte le tracce richieste, è il seguente: