W20D4 – Esame del modulo 5

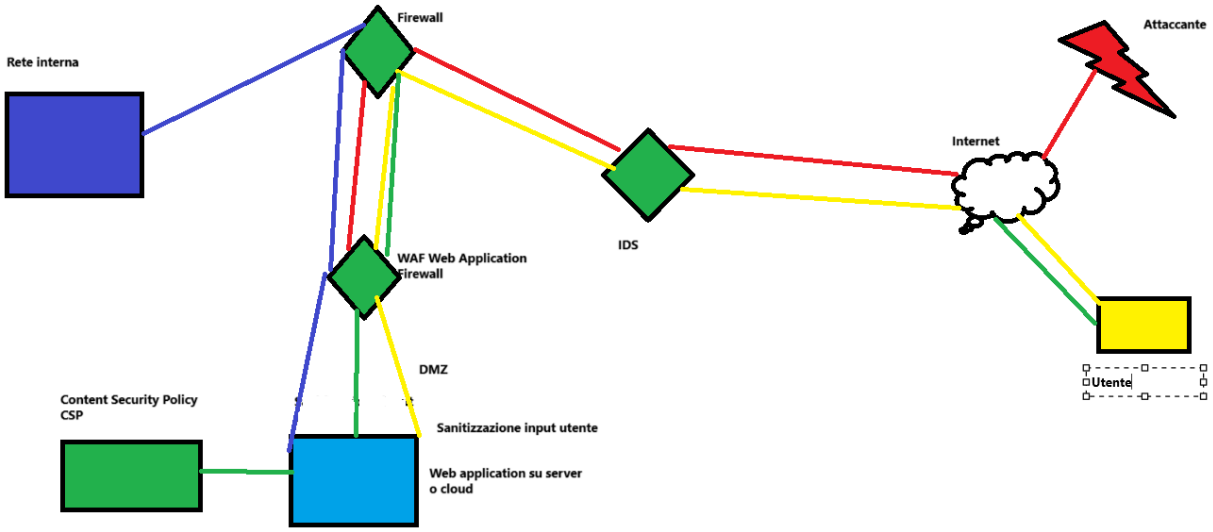
Nota: in questo esercizio ho proposto, nei primi punti, delle soluzioni di base alle problematiche indicate, scegliendo soluzioni efficaci, ma allo stesso non troppo costose e mirate alla risoluzione della problematica indicata. Successivamente, ho iniziato ad unire le varie soluzioni come richiesto nell'esercizio e le differenze tra le reti dimostrano come l'utilizzo delle varie tecnologie sia versatile ed altamente configurabile ed ho cercato di dimostrare come non vi sia una posizione unica nella quale inserire gli elementi della rete. Nell'ultimo grafico, che richiede una modifica più aggressiva dell'infrastruttura, ho provato ad immaginare la rete che proporrei ad un cliente che voglia tutelarsi in modo completo non solo dalle minacce indicate precedentemente ma anche da altre, con una soluzione che richiede un budget non insignificante per essere implementata ma che garantisce un alto livello di sicurezza. Cercherò di spiegare volta per volta i disegni e come e perché ho posizionato gli elementi in un certo modo, aggiungendo anche delle misure non legate alla rete per proteggersi dalle minacce indicate e dei principi per garantire la continuità del business e il pieno recupero dell'operatività. La rete più completa, con il livello di sicurezza più alto e con la maggiore ridondanza è l'ultima soluzione, quella che richiede le modifiche più invasive.

1) Azioni preventive contro XSS ed SQL injection.

Vi sono varie misure preventive da implementare su più livelli per prevenire un attacco basato sul Cross Site Scripting (XSS) o su una SQL injection (SQLi) e queste azioni preventive riguardano sia una modifica del codice sorgente, ad esempio per sanitizzare gli input utenti, sia delle modifiche sull'architettura di rete, aggiungendo per esempio un Web Application Firewall (WAF). Una delle difese fondamentali per prevenire una SQL injection consiste nell'uso di metodi e librerie per la sanitizzazione e la validazione dell'input utente, affinché non venga utilizzata una query SQL per inserire del codice Javascript malevolo o per estrarre dati non legittimi dal database e per fare questo si possono usare librerie di validazione come OWASPI ESAPI. Per prevenire un attacco di SQL injection, è fondamentale, secondo gli studi di OWASP, evitare la scrittura di queries SQL dinamiche con concatenazione delle stringhe, per far sì che una semplice query dinamica non possa essere sfruttata con una query booleana per estrarre più informazioni dal database per scopi malevoli. Si può ricorrere quindi alle "parameterized queries", dette anche "prepared statements", che sono delle queries SQL parametrizzate, cioè precompilate con dei parametri che evitano che un utente malevolo possa cambiare l'intento della query originale, anche se dinamica. Per esempio, viene ricercata nel database una corrispondenza tra i dati salvati e la stringa inserita e, se in questa stringa sono stati inseriti operatori booleani per una SQLi, non vengono estratte informazioni dal database e l'iniezione di codice malevolo non va a buon fine. In questo modo, i valori passati in input vengono trattati come dati e non come possibili comandi SQL. Un'altra opzione valida consiste nell'applicazione di una allow-list input validation, che prevede un lavoro sul codice per specificare da quale tabella nel database prendere i dati necessari, senza lasciare libertà all'utente di inserire la tabella da cui recuperare i dati o di provare una SQL injection. Si può anche esercitare un maggiore

controllo sui caratteri, ad esempio per evitare che vengano accettati e interpretati come codice dei caratteri speciali tipicamente associati all'inserimento di un codice malevolo e questa misura preventiva si chiama "escaping" dei caratteri. I caratteri vengono trattati come tali e non come parte di codice Javascript o HTML. Per limitare le interazioni con il database, si può pensare di usare un ORM, Object-Relational Mapping e per limitare i caratteri inseriti e consentire solo quelli legittimi, si può ricorrere ad un approccio di whitelisting, che permette di inserire solamente dei caratteri accettabili o dei path, come quello di una email in un campo nel quale si chiede di inserire una mail. In questo modo, si eviteranno caratteri speciali e numerici nel campo in cui inserire un nome e cognome. Si può ricorrere anche ad un hardening del database, si possono usare procedure già salvate per avere accesso al database per certi dati senza dover ricorrere a SQL ed assegnare, come sempre, i privilegi minimi agli utenti. Per prevenire un attacco di tipo Cross Site Scripting, è molto utile far ricorso ad una Content Security Policy o CSP, che limita i contenuti esterni come gli script possibilmente malevoli, definendo una serie di regole e criteri che disciplinano quali contenuti possono essere salvati ed eseguiti e da che fonti devono provenire. Anche contro XSS è infatti necessario implementare delle misure per la sanitizzazione dell'input, per esempio eseguendo un controllo sui caratteri inseriti per evitare che vengano messi dei caratteri speciali o numerici se ad esempio viene richiesto un nome e un cognome. Possono essere vietati esplicitamente dei tag come <script> usati frequentemente negli attacchi XSS o non consentire i commenti HTML. Sono utili anche le tecniche di escaping per fare in modo che i caratteri vengano trattati come tali e non come codice oppure inserire editor del tipo "What-You-See-Is-What-You-Get" (WYSIWYG), per evitare che venga inserito del codice malevolo negli input utenti che poi viene eseguito dal browser. Per evitare il Cross Site Scripting serve implementare misure di sicurezza a protezione dei cookies, come legarli ad un IP specifico per prevenire il furto tramite IP dinamici, oppure ancora è utile attivare il flag http-only per evitare che un codice Javascript possa avere accesso ai cookie e rubarli. Si può anche aggiungere il flag X-XSS-Protection Header nelle risposte http per aiutare i browser ad attivare le loro protezioni contro XSS o ricorrere alle X-Content-Type-Options, da inserire nell'header per evitare che il browser interpreti la risorsa come una tipologia diversa. Tra le azioni fondamentali a livello di network che proteggono sia dagli attacchi SQLi sia XSS c'è l'utilizzo di un Web Application Firewall, detto anche WAF. Questa tipologia di firewall è specificamente pensata per proteggere una web application da attacchi di questo tipo, filtrando il traffico per proteggere appositamente le web application dalle minacce che le riguardano direttamente. Viene filtrato il traffico e vengono rilevati possibili script malevoli legati sia ad una SQLi sia a XSS, evitando che venga eseguito del codice malevolo o che SQL venga sfruttato per fini criminali. Il WAF può essere basato sulla rete, venendo implementato come soluzione hardware, che riduce i tempi di latenza al minimo ma che richiede un costo maggiore. Può anche essere integrato nel software della web application, diventando un WAF basato su host che richiede comunque un uso significativo delle risorse del server locale, oppure ancora può essere basato su cloud, riducendo i costi di gestione e garantendo una risposta immediata, un rapido aggiornamento e una veloce configurazione delle policy. Il WAF è uno strumento molto utile perché protegge anche

contro i bot, Cross-site Forgery CSR e attacchi DoS o DDoS a livello di applicazione e permette di definire delle policy per filtrare il traffico in base a differenti criteri, ad esempio l'origine del traffico, e raccoglie dei log molto preziosi in caso di incidente o semplicemente per il monitoraggio della rete. È collocato vicino alla web application per garantire una maggiore protezione a livello applicativo e per fare questo funge da reverse proxy, per cui il client deve passare per il WAF prima di raggiungere il server, evitando che qualcosa di dannoso possa raggiungere il server e verificando che ciò che arriva al server non crea problemi. Il WAF si basa su una blocklist, che blocca attacchi o fonti note, o su una allowlist, che indica quali fonti possono passare il controllo del firewall. Una funzione protettiva più ampia è data anche dall'uso di IDS ed IPS, che proteggono contro varie tipologie di attacchi e di azioni malevole. Gli IDS, intrusion detection system, filtrano il traffico di rete in tempo reale e, in caso di attività sospette, si limitano a lanciare un alert ai team responsabili, consentendo di ridurre i tempi per rilevare un attacco e di intervenire tempestivamente. Gli IPS, intrusion prevention system, oltre a monitorare il traffico di rete e lanciare degli alert agiscono tempestivamente in automatico per prevenire i danni. La soluzione che ho mostrato è una soluzione efficace ma non troppo impegnativa o costosa che previene un attacco XSS e SQLi: la rete è segmentata e divisa tra la rete interna e la DMZ, che espone la web application e il firewall protegge dal traffico in entrata da internet ed evita che una minaccia possa propagarsi eventualmente dalla DMZ verso gli end point della rete interna. È possibile integrare il firewall con dei sensori IDS, che possono essere piazzati anche tra internet e il firewall interno, fungendo da protezione perimetrale e filtrando già il traffico da internet. Un IPS potrebbe già porre in essere azioni difensive, anziché rilevare solamente del possibile traffico malevolo, ma al momento ho pensato di proporre una soluzione semplice e non troppo onerosa. La posizione ottimale del Web Application Firewall WAF è proprio davanti alla web application, filtrando il traffico tra la DMZ e il firewall interno ed evitando che richieste http ed input utenti malevoli possano raggiungere il server della web application ed ha anche una funzione protettiva nei confronti del resto della rete, filtrando possibili input malevoli. Si potrebbe piazzare anche nel perimetro, prima del firewall interno, filtrando da subito gli input in arrivo da internet, ma questa non è la soluzione più efficiente, perché esegue dei controlli in anticipo e potrebbe sovraccaricare la rete filtrando degli input legittimi. Il WAF direttamente davanti alla web application, invece, filtra gli input nel modo più efficace possibile, proteggendo direttamente la web app ed aggiungendo un livello di protezione ulteriore che si aggiunge al firewall. Per proteggere specificamente contro gli attacchi XSS, si può usare una Content Security Policy (CSP) e l'input utente viene sanitizzato con precisione con le tecniche indicate precedentemente e con l'aiuto del WAF. In questo modo, l'attaccante che prova un XSS o una SQLi viene fermato al limite dal Web Application Firewall, bloccando l'attacco sul nascere e proteggendo sia le connessioni con la rete interna, sia le interazioni con l'utente.

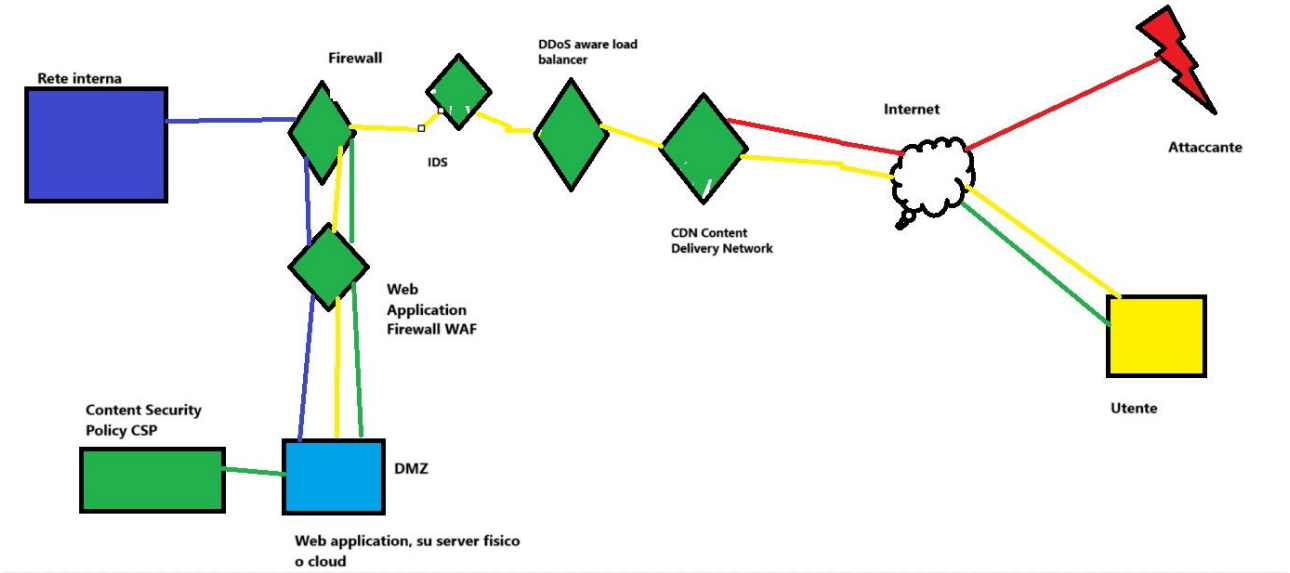


2) DDoS: impatti sul business e azioni preventive.

La formula del CoD, cost of downtime, aiuta a calcolare i danni che avrebbe l'azienda in caso di attacco DDoS sulla web app di e-commerce. Il CoD si ottiene moltiplicando il guadagno per minuto per il tempo di inattività; quindi, se l'azienda guadagna 1500 euro al minuto e l'inattività dura 10 minuti, il cost of downtime è pari a 15.000 euro (1500 x 10) e questo sarebbe l'impatto sul business in caso di non raggiungibilità dell'e-commerce. Queste cifre fanno capire l'importanza di tutelarsi contro un DDoS e potrebbero essere anche più alte, perché gli utenti spendono in media 1500 euro al minuto, ma trattandosi di una media, l'impatto potrebbe essere maggiore se il disservizio si verifica in momenti in cui gli utenti spendono di più della media, si pensi al black Friday o a un momento di saldi e promozioni.

Per la prevenzione di un attacco DDoS, è importante utilizzare un CDN o Content Delivery Network, che riesce a gestire attacchi volumetrici come http flood, SYN Flood o UDP flood distribuendo il carico su vari POP, ossia point of presence, evitando che il traffico malevolo raggiunga direttamente il firewall e la rete aziendali. È necessario che il CDN possa contare su una rete di POP abbastanza ampia, sull'uso della crittografia per i dati in transito e che sia ben configurato, prevedendo una protezione contro vari tipi di attacchi, come attacchi volumetrici, basati sui protocolli oppure contro le applicazioni. Questo aiuta ad evitare il sovraccarico dell'infrastruttura a causa dell'elevato volume di traffico di un DDoS ed usa una cache che memorizza le copie dei contenuti richiesti precedentemente affinché il server abbia un carico minore di richieste nuove da soddisfare. Gli attacchi a protocollo sono attacchi che interessano i livelli 3 e 4 di ISO/OSI, inviando una grande quantità di traffico, basato tra gli altri sul protocollo ICMP, mentre gli attacchi volumetrici sono quelli realizzati magari da botnet con protocolli di rete comuni per saturare le risorse del server. A volte il CDN è integrato con scrubbing centers basati su cloud che aiutano a dirottare il traffico verso altri datacenter, evitando che l'applicazione sia sommersa dal traffico e spesso il CDN prevede anche un anycast routing che instrada il traffico eccessivo verso i nodi server più vicini a livello globale. Questa è una misura di sicurezza perimetrale, per proteggere la rete prima che arrivi al firewall. Anche il Web Application Firewall, posizionato prima della web

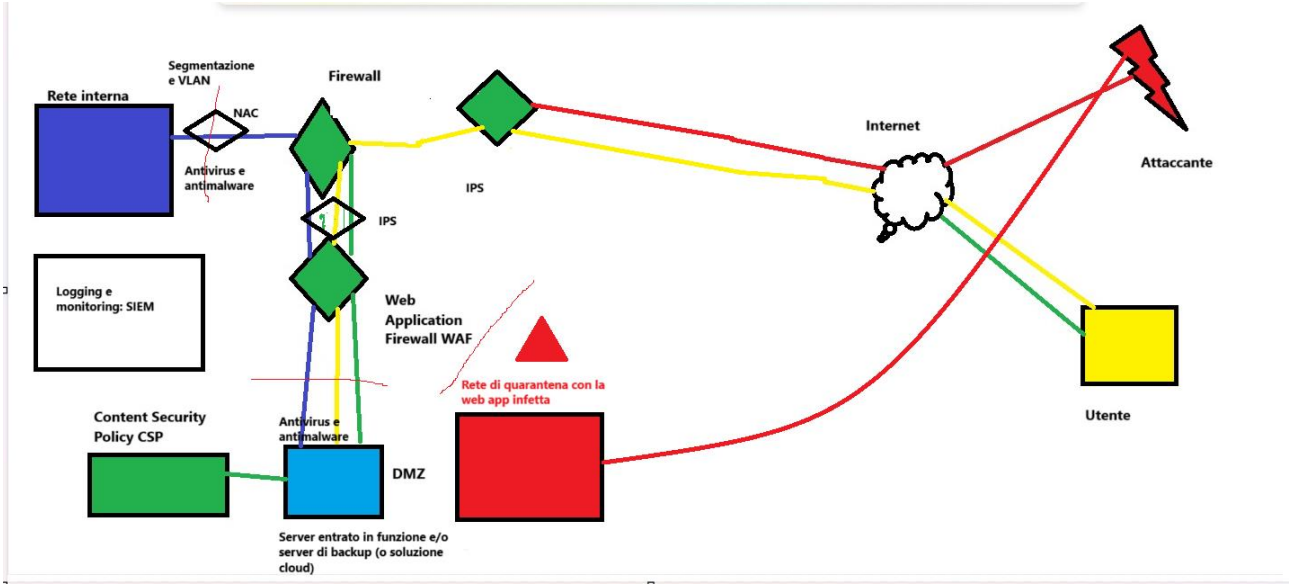
application, aiuta a prevenire un attacco DDoS a livello applicativo, filtrando il traffico http ed individuando dei pattern sospetti, legati ad esempio a richieste ripetitive o con payload sospetti. Tra le varie misure preventive, aggiungere delle rate limiting policies è importante perché aiuta ad evitare l'inondazione di richieste che un DDoS comporta, inserendo un limite di richieste che un singolo indirizzo IP può fare in un determinato lasso di tempo. È una buona idea anche aggiungere un DDoS aware load balancer, in grado di distribuire il traffico per bilanciarlo, per evitare che le troppe richieste mettano in difficoltà la web application, posizionandolo anche come misura di sicurezza perimetrale. Anche un IDS o ancora meglio un IPS sono in grado di prevenire un attacco DDoS e per questo sarebbe una buona idea integrarli nell'architettura di rete. Una strategia preventiva contro un attacco DDoS consiste nell'usare un cluster server, ovvero un gruppo di server che agiscono come se fossero uno e che serve per distribuire il carico di lavoro e di richieste provenienti dagli utenti collegati. Con un failover cluster, invece, se un server viene bloccato dall'attacco DDoS, un altro server viene promosso a nodo attivo e può consentire un rapido recupero dell'operatività, grazie anche a copie di backup che possono essere salvate su un server o nel cloud. Per questi motivi la ridondanza è utile anche contro il DDoS per poter ristabilire l'operatività della web app in fretta. Nella rete che propongo, la sicurezza perimetrale è assicurata dal CDN e dal DDoS aware load balancer, che riescono a gestire il traffico associato ad un DDoS, evitando che l'ondata di richieste arrivi al firewall della rete aziendale, proteggendola da subito. Sempre per garantire un buon livello di sicurezza tra internet e la rete interna, ho inserito un IDS per monitorare il traffico prima del firewall, soluzione più economica di un IPS, tenendo però presente che un IPS sarebbe comunque la soluzione migliore. In virtù del suo effetto protettivo contro i DDoS applicativi, ho mantenuto il WAF a protezione della Web app, in modo che non venga bloccata dall'elevato traffico creato da un DDoS. L'applicazione potrebbe girare su un server cluster per garantire una distribuzione delle richieste e una maggiore resilienza.



3) Response contro un malware

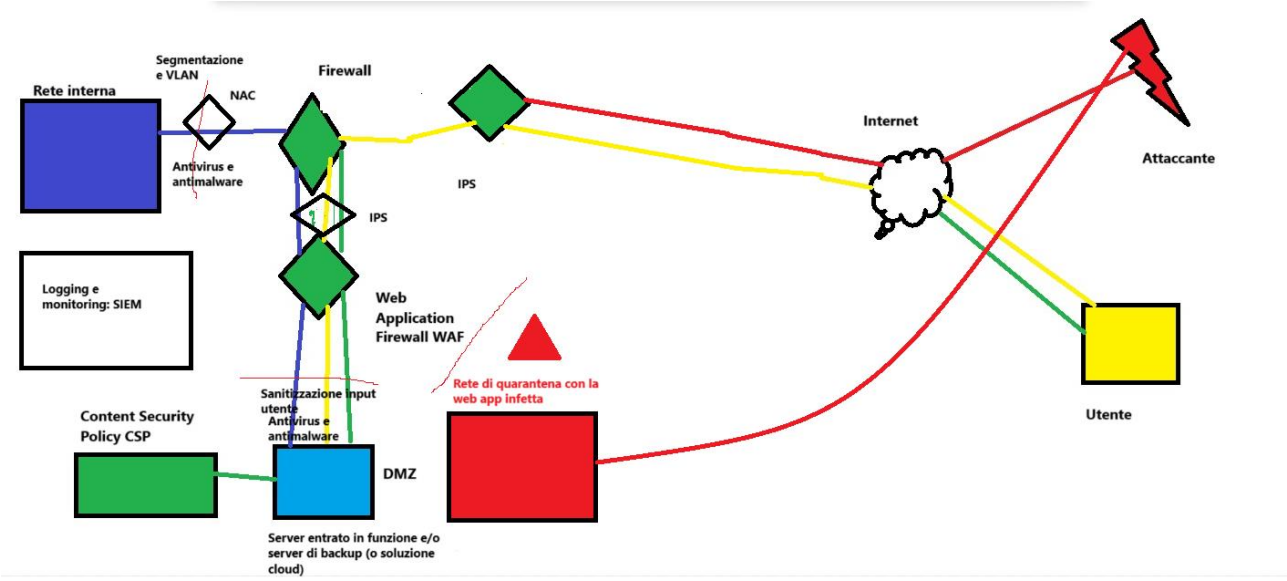
In generale, per porre rimedio ad un incidente legato ad un malware serve scollegare la macchina dalla rete, metterla in modalità sicura, usare uno strumento antimalware per la scansione, il rilevamento ed infine la rimozione del malware e infine pulire la cache ed altri elementi impattati dal malware. Tra le strategie preventive rientrano l'aggiornamento dei sistemi, l'attenzione nel download di materiali o nel collegamento con link sospetti, l'utilizzo di un antivirus con un database di firme aggiornato, l'autenticazione a più fattori e un monitoraggio regolare di possibili IoC. In caso di un attacco per cui la web application è stata infettata con un malware, è necessario isolare subito il server infetto in una rete di quarantena, per evitare che il malware possa propagarsi verso la rete interna ed infettare gli altri dispositivi. In questo caso si può scegliere la soluzione dell'isolamento, per cui il sistema infetto viene posto su una rete di quarantena che viene staccata dalla rete interna, rafforzando allo stesso tempo le policy del firewall per limitare fortemente la possibilità di accesso alla rete di quarantena se non per attività magari forensi o di analisi e per proteggere la rete interna. Bisogna anche rafforzare la segmentazione, isolando le risorse importanti della rete interna e magari inserendo una VLAN per le risorse più importanti. Come richiesto dall'esercizio, ho rappresentato la soluzione dell'isolamento, per cui la macchina infetta è staccata dalla rete ma è ancora raggiungibile tramite internet dall'attaccante, anche se sarebbe utile valutare se l'isolamento è sufficiente, oppure se invece serve pensare ad un approccio di rimozione. Con la rimozione, infatti, la web application infetta è isolata dalla rete interna e non è più raggiungibile nemmeno dall'attaccante tramite internet. Sarebbe quindi utile pensare se l'isolamento è sufficiente o se invece la rimozione rappresenta la soluzione più utile e sicura. A seguito dell'attacco, potrebbe essere una soluzione utile implementare ed inserire dei sensori questa volta IPS sia prima della DMZ, sia prima del firewall interno, come difesa perimetrale e gli IPS, a differenza degli IDS, sono più costosi ma più efficaci e tempestivi nell'agire. Bisogna anche verificare l'aggiornamento di antimalware ed antivirus alla luce della minaccia verificatasi. In termini di business continuity e disaster recovery, per garantire il minore impatto possibile sull'operatività aziendale, si può pensare ad un failover cluster, per cui un secondo server viene promosso a nodo attivo non appena il server della web application infettata viene isolato e smette di erogare servizi. In questo modo, viene ridotto l'impatto sull'azienda e può essere utile pensare anche ad avere un server di backup per il recupero rapido dei dati in caso di necessità. Questi elementi sono sempre parte della DMZ ed aiutano a rispondere adeguatamente all'incidente di sicurezza. Se la web application e il server sono basati su cloud, la risposta all'incidente è ancora più immediata, perché il server infetto viene immediatamente isolato ai primi segnali sospetti e messo in quarantena e si attivano subito le soluzioni di business continuity. La risposta a questo incidente richiede un'attività di indagine forense per capire meglio cosa è successo e dove sono le falle di sicurezza e per questo sono utili i log, sia i log di sicurezza di dispositivi come il firewall, sia i log di sistema del server su cui girava la web app, sia i log dell'applicazione. Si può pensare anche ad un SIEM per rendere più agevole il monitoraggio e la correlazione, magari mettendo il SIEM in un segmento separato dalla rete. Implementare il SIEM è una soluzione adeguata per

un’azienda che vuole investire un budget maggiore nella sicurezza. Anche un EDR potrebbe essere un’ottima soluzione per monitorare la sicurezza degli endpoint e per rispondere ad attacchi complessi, ad esempio basati su ransomware. Per studiare il malware che ha infettato la web app potrebbe servire anche una sandbox per poterlo eseguire in un ambiente protetto senza correre il rischio di danneggiare la rete. La parte finale della response a questo incidente consiste nella rimozione del malware e nel pieno recupero del server della web application, anche grazie ad un server di backup per il recupero dei dati. Se si osserva il disegno, si vedono tre sottili linee rosse: la linea verso la DMZ indica un rafforzamento delle politiche del firewall, mentre la linea rossa vicino alla rete di quarantena dimostra che la rete è ben isolata dalla rete interna e ci si può accedere solo tramite canali molto sicuri, magari per indagini fornesi. Per la protezione della rete interna, ho pensato di inserire anche un NAC, network access control, proprio all’ingresso della rete interna. Le politiche del firewall e l’attenta segmentazione della rete, unite alle misure di sicurezza implementate precedentemente, potrebbero far pensare che un NAC non serva; invece, solamente il NAC verifica con attenzione quali dispositivi hanno accesso alla rete, senza monitorare solamente il traffico, ma controllando i requisiti di questi dispositivi e dei loro utenti. Ho aggiunto questa soluzione perché nell’esercizio era indicato il pericolo di un movimento laterale dalla DMZ alla rete interna di un malware e i NAC sono molto abili nel riconoscere possibili tentativi di movimento laterale, rendendoli la soluzione ideale per la protezione della rete interna provenienti dalla DMZ esposta ad internet. Nel disegno si vedono l’utilizzo di un IPS come protezione perimetrale e come ulteriore protezione della DMZ e la sicurezza della web application è sempre garantita dalla presenza del WAF. Si nota anche il segmento di rete gestito dal SOC con il SIEM per il monitoraggio e la gestione dei log.



4) Soluzione completa.

Unendo la soluzione 1 e la soluzione 3, combinando quindi le misure preventive contro gli attacchi XSS e SQLi e le misure prese per rispondere all'incidente di sicurezza derivante dall'infezione della web application con un malware, si ottiene una rete con un livello di sicurezza maggiore contro diverse minacce ed eventualmente con un buon rapporto costi/benefici per un'azienda. La rete di quarantena isola la web application infetta, la quale non rappresenta un pericolo per la rete interna essendo staccata da essa e questo avviene anche grazie ad un rafforzamento della segmentazione e delle policy del firewall. In più la presenza di un IPS e di un WAF tra la DMZ e il firewall interno aiuta a garantire un buon livello di sicurezza e ad evitare che eventuali problematiche nella DMZ, con la web application esposta ad internet, vadano ad impattare la rete interna. Per i motivi di controllo dei dispositivi e di sicurezza al fine di evitare il movimento laterale di un malware, il NAC protegge la rete interna, essendo situato proprio al suo ingresso. Per garantire la continuità del business, c'è sempre un failover cluster, con un server che viene promosso a nodo attivo in caso di problemi e/o con un server di backup per recuperare i dati. Anche su cloud ci sono varie soluzioni vantaggiose per mantenere la continuità del business ed isolare il server infetto. La web application infetta è ancora raggiungibile via internet dall'attaccante, ma è ben staccata dalla rete interna e, come spiegato prima, si può pensare ad una totale rimozione. Viene sempre effettuata la sanitizzazione degli input degli utenti per pervenire attacchi di tipo XSS ed SQLi e il WAF aiuta a garantire una maggiore sicurezza per la web application, filtrano il traffico e concentrandosi sulle minacce specifiche per una web app. I sensori IDS o meglio un IPS proteggono sia esternamente, filtrando il traffico in arrivo da internet, sia il traffico tra la DMZ e il firewall interno, aggiungendo un livello di protezione ulteriore per evitare che eventuali minacce si propaghino alla rete interna, difesa anche da antivirus e antimalware aggiornati, da policy del firewall ben configurate e da una segmentazione attenta. La content security policy previene gli attacchi XSS e tutti i log di sistema, di sicurezza ed applicativi convergono nel SIEM, che effettua correlazioni ed aiuta ad individuare e rispondere ad eventuali incidenti. Come richiesto nell'esercizio, ho unito nel disegno solamente le azioni preventive e di risposta contro i malware e contro XSS ed SQLi, ma questa configurazione di rete offre protezione anche contro i DDoS grazie all'azione dell'IPS e del WAF. Si tratta di una rete adatta ad un'azienda con un buon budget e che deve garantire un buon livello di sicurezza, anche per evitare gli effetti negativi legati alla sfiducia o alla pubblicità negativa, trattandosi di un e-commerce.



5) Modifica aggressiva dell'infrastruttura.

La seguente soluzione è molto complessa e richiede un importante investimento per un'azienda; tuttavia, offre un alto livello di protezione contro diversi tipi di minacce. La sicurezza perimetrale è affidata ad un Next Generation Firewall NGFW, che analizza il traffico in modo avanzato, avvalendosi anche di elementi comportamentali e che offre da subito un altissimo livello di protezione prima che il possibile traffico malevolo giunga alla rete. In alternativa si può ricorrere anche ad un IPS, che offre un'avanzata funzione protettiva contro gli attacchi e le intrusioni malevole ed agisce non appena rileva una possibile intrusione. Prima che un traffico legato ad un DDoS raggiunga il firewall interno ed impatti la rete, un CDN e/o un DDoS aware load balancer fermano l'elevata quantità di richieste prima che vadano a saturare le risorse interne e, prima del firewall interno, si può inserire un IPS se non è già stato implementato prima. La rete interna è ben segmentata e si avvicina quasi ad un modello di Zero Trust Architecture, per cui ogni comunicazione, anche interna, viene validata ed autenticata per evitare qualsiasi minaccia o movimento laterale di un attaccante che abbia superato tutte le misure di sicurezza perimetrali o che agisca dell'interno della rete stessa. Vi sono infatti dei PEP, ossia Policy Enforcement Points in punti critici, come l'ingresso alla rete interna oppure tra la DMZ e il firewall interno, che richiedono l'autenticazione degli utenti e verificano le comunicazioni in atto. La DMZ è protetta da un IPS e da un WAF, che proteggono anche la rete interna impedendo che qualcosa di malevolo si muova e in tutta la rete gli antivirus e gli antimalware sono aggiornati diligentemente in tutti gli endpoint. Contro gli attacchi XSS o SQLi l'input utente viene sanitizzato ed alla protezione contro XSS ed SQLi si aggiunga la presenza di una content security policy CSP contro gli attacchi XSS. Per garantire la continuità del business è implementata una certa ridondanza. Per esempio, vi è un failover cluster o un server cluster per garantire l'operatività anche in caso di incidenti, vi sono server di backup o copie di backup su cloud per ottimizzare i costi, vi è ridondanza con le configurazioni RAID in eventuali dischi fisici per il salvataggio dei dati e sono in essere dei sistemi per isolare un elemento compromesso o infettato da un malware. Il NAC, anche detto Network Access Control, è un controllo aggiuntivo che serve

per evitare che dispositivi che non rispettano certi standard di sicurezza o comunque non autorizzati abbiano accesso alla rete ed è posto come ulteriore protezione nei confronti della rete interna, che contiene gli end point non esposti sulla rete e che contengono informazioni preziose da proteggere. Il NAC serve anche per proteggere da possibili movimenti laterali di attaccanti dalla DMZ. In alternativa al NAC si potrebbe inserire nuovamente un sistema IPS, in quanto un IPS sarebbe in grado di rilevare eventuali tentativi di movimento laterale dalla DMZ esposta alla rete. Tuttavia, avendo già implementato diverse misure di sicurezza, ho pensato di inserire un NAC per verificare in maniera più puntuale gli accessi alla rete interna, anche perché offre comunque una protezione più efficace contro il movimento laterale che può avvenire dalla DMZ. In questo modo ci sono vari livelli di protezione a difesa della rete interna. In un'architettura di rete così sicura devono per forza esserci delle politiche di backup solide, magari effettuate su cloud e delle politiche di logging e monitoring altrettanto solide. Ho pensato ad un segmento dedicato interamente al SOC, nel quale confluiscono tutti i log di sicurezza, di sistema ed applicativi e che raccolgono i log che inviano tutti i dispositivi. Il lavoro di raccolta, analisi e correlazione lo svolge il SIEM, ma si può pensare anche ad un SOAR per un livello di sicurezza ancora maggiore e per avere delle risposte difensive automatiche prima che intervengano i tecnici. In questo segmento si può aggiungere anche un EDR, ossia EndPoint Detection and Response, che raccoglie i dati dagli endpoint, monitorando costantemente le loro connessioni, li analizza anche grazie alla correlazione e segnala tempestivamente delle attività sospette. Sono molto utili contro i ransomware o contro il phishing, rilevano anche attacchi sofisticati e salvano i dati di log raccolti anche per future analisi. Esistono soluzioni di logging e monitoring anche su cloud.