Vulnerabilità di Injection e Remote Code Execution: la principale causa di Data Breach e furto di dati

Estratto dalla relazione di Massimiliano Brolli tenutasi al 10° Cyber Crime Conference 2019

Vorrei partire dalla nota frase di un grandissimo inventore greco – Archimede da Siracusa – del quale si narra che, dopo aver avuto l'intuizione della leva, disse: *"Datemi una leva e vi solleverò il mondo"*. Qual è il nesso con la Cyber Crime Conference? Partiamo con un po' di storia.

Sia la grande, sia la piccola e media impresa hanno, nel tempo spostato il loro business su Internet: sono 15-20 anni che assistiamo – abbiamo terminato da poco – al trasferimento dei servizi sul web e, quindi, tutte le applicazioni (e i database) che in precedenza giravano nelle Intranet aziendali sono state in qualche maniera rivisti e portati, attraverso Web Application e app *mobile*, nel web.

Ma il mondo web, lo sappiamo, è un ecosistema complesso, pieno di minacce e che viene costantemente scansionato per indicizzare tutto quanto il contenuto in esso presente e pubblicizzarlo all'interno di *repository* accessibili a chiunque (ad esempio Shodan, zoomeye, ecc…) che possono essere utilizzati come vettori d'attacco per consentire ai cybercriminali l'esfiltrazione di dati dalle nostre aziende.

Mettendo a paragone i *cyberspace* Intranet e Internet, possiamo dire che molto spesso ci siamo focalizzati sul tentativo di

analizzare ciò che era all'interno della nostra Intranet, per prevenire possibili frodi e fenomeni di esfiltrazione dati per diverse finalità (ad esempio, nel caso di dipendenti infedeli). Ma Internet cos'è? Secondo le ultime stime si parla di 4 miliardi di utenti singoli e 24 miliardi di *devices*: in questa stima sono comprese applicazioni web ma anche app *mobile*, ecosistemi IoT e via discorrendo. Quindi, se tenendo in considerazione una Intranet di 10 mila utenze – stiamo già parlando di un'Intranet di grandi dimensioni, per non dire di una *Big Company* a livello internazionale – per ogni potenziale aggressore possiamo dire che, da rete Internet, possono essercene ben 400mila.

Sappiamo che le informazioni saranno il petrolio del nuovo millennio; quindi, ricollegandomi alla citazione d'apertura, ci saranno nuove "leve" che consentiranno di esfiltrare dai sistemi i nostri dati e i nostri "gioielli della corona", che pensiamo di aver messo in sicurezza ma, probabilmente, questo non sempre è stato fatto nella maniera dovuta.



ne cito alcuni: il fenomeno del *phishing,* dell'*SQL injection*, *Remote Code Execution*, i malware in generale. Dobbiamo però partire dal concetto che, purtroppo, oggi tutti i sistemi risultano violabili, dipendentemente dal tempo e soprattutto dall'interesse che hanno i cybercriminali nel produrre attacchi verso le nostre aziende. Nell'era dei *data breach*, questo non può essere più sostenibile; per fortuna si può evitarlo con l'adozione di processi di cybersecurity in grado di risolvere tutte queste carenze. Ma questo richiede costanza e precisione, una precisione di livello militare: stiamo parlando del processo del *patching*, quindi condurre un'efficace attività di discovery delle minacce e *patching* delle applicazioni non appena vengono ufficializzate, da parte dei *vendor*, le vulnerabilità rilevate sulle nostre applicazioni; stiamo parlando del fenomeno

dell'*hardening* – sappiamo tutti che, per sostituire un kit che non funziona bene in una notte burrascosa, i nostri sistemisti le provano tutte fino, magari, a creare all'interno del sistema delle *misconfigurations* che poi non vengono ripristinate (password di default, servizi che prima non c'erano e poi compaiono, magari anche portando nuove vulnerabilità) – e quindi bisogna presidiare il tema dell'*hardening loss*.

Un altro vettore importante – qui mi ricongiungo all'altra "leva" prima citata, quindi *phishing* e *SQL injection* – è lo sviluppo sicuro del codice. Siamo tutti convinti che avendo buoni strumenti di scansione statica e dinamica, di metterci al riparo da tutte le vulnerabilità di sicurezza presenti sulle nostre *Web application*. Purtroppo questo non è vero, perché un *best in class* di questi prodotti può rilevare all'incirca 4 su 10 (parlando sempre di *Top 10 owasp,* che più o meno conosciamo) vulnerabilità di sicurezza che, generalmente, sono vulnerabilità di depurazione dell'input: tutte le restanti vulnerabilità presenti sui nostri sistemi sono demandate allo sviluppo. E non è detto che un bravo sviluppatore sia un buon sviluppatore di codice sicuro, quindi dobbiamo fare formazione: in particolar modo chi produce *software* nelle sue *factory* deve insegnare a produrre codice di qualità.

Oltretutto abbiamo assistito, nell'ultimo periodo, al completamento del paradosso della sicurezza informatica: se in precedenza, tra un *client* e il *server* passava un flusso in chiaro e noi potevamo tranquillamente verificare quello che transitava, riuscendo così a vedere (e bloccare) i *payload* dannosi, in seguito – poiché, ovviamente, all'interno di questo flusso in chiaro transitavano anche *user* e password – per garantire una corretta gestione della privacy dell'utente abbiamo implementato i flussi di cifratura. Ovviamente, avendo distribuito tutti questi canali di protocolli cifrati, ci siamo trovati a un certo punto a non

“vedere” più nulla. Allora ci siamo rivolti agli ingegneri di sicurezza, che come soluzione ci hanno proposto di mettere in mezzo uno “scatolotto” (che possiamo chiamare *Web Application Firewall*) cifrando da entrambi i lati, lasciando al centro la possibilità di avere visibilità del traffico in chiaro; quindi, con meccanismi di *Machine Learning*, intelligenza artificiale, correlazione, *SSL inspection* e chi più ne ha più ne metta, possiamo far passare le non-vulnerabilità – quindi tutte le richieste lecite che arrivano al sistema- e bloccare tutto quello che rientra all’interno dei *pattern* di *policy* implementati dentro lo “scatolotto”. Come gestire, invece, tutto ciò che risulta “grigio”, quindi i falsi positivi? Mandiamo tutte le richieste *border* all’interno di un SIEM dove gli analisti del *Security Operation Center* analizzano questi flussi, generano nuove regole e le re-implementano all’interno dello “scatolotto”. Quindi, di fatto, generiamo un meccanismo di *self-improvement* che consente di automatizzare questo processo; il che rende gestibile anche la migrazione sul *cloud,* ma comporta una vertiginosa crescita della complessità. Inoltre, con l’aumento dell’esposizione su internet avremo ulteriori difficoltà; perché il 5G sarà un fenomeno abilitante di nuove tecnologie e nuovi ecosistemi che proietteranno su Internet una quantità industriale di *devices* e di applicazioni

Anche se penso che siamo tutti al corrente di questo fenomeno.

Noi esperti di sicurezza informatica ci troviamo di fronte a una grande sfida nei prossimi anni, perché abbiamo in mano quattro carte che non vanno molto d’accordo con gli obiettivi di cybersecurity: *in primis* la complessità, quindi ecosistemi complessi, *cloud*, protocolli wireless (perché di fenomeni di attacco verso le reti wi-fi ce ne sono, eccome), industry 4.0 in generale e anche il cambiamento da *macro cell* a *small cell* portato dal 5G, che a sua volta genera ulteriore

complessità.

In secondo luogo le minacce, che negli ultimi due anni si sono assestate su circa 3000 CVE annuali critici (senza parlare del fenomeno degli Zero Day, del cyberspionaggio, il cui mercato sta sviluppando una complessità a tale da poter cominciare a suscitare preoccupazioni).

Continua a leggere…

Scarica gratuitamente gli atti del 10° Cyber Crime Conference 2019

https://www.ictsecuritymagazine.com/articoli/vulnerabilita-di-injection-e-remote-code-execution-la-principale-causa-di-data-breach-e-furto-di-dati/