Volume 7, numero 4
[Stato di Internet]

API: la superficie di attacco che ci collega tutti

Sommario

Lettera del direttore

Benvenuti nel rapporto di Akamai sullo stato di Internet - Security (SOTI), volume 7, numero 4. Se è la prima volta che leggete questo rapporto o se lo fate con costanza, desideriamo porgervi il benvenuto nella speranza che la nostra ricerca riesca a fornirvi informazioni non disponibili altrove.

Lo sapevate che la prima istanza di un'API (Application Programming Interface) è stata creata da Salesforce.com il 7 febbraio del 2000, secondo quanto riportato nel blog sulle API? Ciò che in origine era un metodo relativamente semplice di comunicazione da un sistema all'altro si è evoluto in uno dei fattori che hanno fatto maggiormente aumentare il traffico in Internet.

Abbiamo contattato Chris Eng, Chief Research Officer di Veracode, per chiedergli di parlare della crescita delle API e delle vulnerabilità a cui espongono le organizzazioni. Chris si è affacciato al settore della sicurezza all'incirca nel periodo in cui veniva scritta la prima API e, per il suo ruolo all'interno di Veracode nell'ambito della sicurezza delle applicazioni, dispone di una profonda conoscenza dell'argomento. Non sorprende il fatto che Chris riesca a scorgere notevoli somiglianze tra gli inizi dello sviluppo di software e l'attuale stato del traffico delle API.

Riteniamo che gli attacchi alle API non siano rilevati e segnalati in modo adeguato, il che li rende una delle minacce più serie che le organizzazioni si trovano ad affrontare. Gli attacchi DDoS e ransomware vengono considerati problemi seri e degni di nota poiché il loro impatto è immediato e visibile. Gli attacchi alle API non ricevono lo stesso livello di attenzione, in larga parte perché i criminali le utilizzano in modo più subdolo rispetto agli attacchi ransomware.

Uno dei nostri punti di forza è rappresentato dall'impegno volto a migliorare il rapporto SOTI. Desideriamo continuare in questo intento per portare in primo piano nuove ricerche e idee interessanti mai trattate prima d'ora. Ci proponiamo di utilizzare diverse fonti di dati, incluse le informazioni raccolte dai nostri partner come Veracode. Non vediamo l'ora di condividere con voi le informazioni da noi raccolte sullo scenario ancora confuso che circonda le API.

Martin McKeay
Editorial Director

SAGGIO

Sicurezza delle API: il passato si ripete

Chris Eng
Chief Research Officer, Veracode

Internet è caratterizzato da un notevole livello di solidità e capacità di autoriparazione in caso di interruzioni casuali, ma più ci si avvicina al livello delle applicazioni e, in particolare, al livello umano, più ci si chiede come fa a funzionare?

Gli attacchi informatici di alto profilo sono diventati sempre più comuni e hanno ampliato la loro portata, in particolar modo i ransomware. Un aspetto che non deve preoccuparvi, ma solo se non vi preoccupate nemmeno di questioni futili quali la benzina, la carne, i viaggi aerei o il backup dei dati.

Questo rapporto è incentrato sulla sicurezza delle API. Se ve ne siete occupati, sapete bene come spesso questo argomento passi in secondo piano.

La prima regola per scrivere software sicuro consiste nel non fare supposizioni sul modo di interazione degli utenti con il prodotto finito. Quando ho iniziato ad affacciarmi a questo settore più di vent'anni fa, come specialista dei test di penetrazione, quasi ogni sito web era facilmente attaccabile perché si ritenevano validi presupposti errati, ad es. "è impossibile cambiare il valore di un menu a tendina", "la convalida lato client funziona" e (questa è la mia preferita) "nessuno lo farà mai" Gli sviluppatori web si trovavano già a così tanti livelli di astrazione, e così lontani dalla tecnologia sottostante che la maggior parte di essi non riusciva a comprendere neanche il protocollo HTTP (e, probabilmente, la cosa è vera anche oggi o forse la situazione è persino peggiorata).

Nel corso del tempo, le applicazioni web sono state lentamente migliorate grazie alla comparsa di strumenti di test più sofisticati e altri SDLC, ma con le API abbiamo visto ripetersi continuamente gli stessi schemi. Ad esempio, Lebin Cheng di CloudVector ha stilato un elenco degli incidenti correlati alle API che si sono verificati nel 2020. Anche se non si tratta di un elenco esaustivo, rappresenta un esempio diretto degli schemi ripetuti presenti dai primi tempi dello sviluppo e dei test delle applicazioni.

Poiché le API si nascondono spesso all'interno delle app per dispositivi mobili, fanno pensare che siano immuni alla manipolazione. Gli sviluppatori partono dal pericoloso presupposto che l'interazione tra utenti e API avvenga solo tramite l'interfaccia utente dei dispositivi mobili.

Rispetto all'elenco OWASP Top 10, l'OWASP API Security Top 10 sostiene di trattare le vulnerabilità e i rischi per la sicurezza specifici delle API, ma, ad un'analisi più approfondita, contiene le stesse vulnerabilità web descritte in un ordine e con termini leggermente diversi. Tanto per rincarare la dose, le chiamate API sono più semplici e veloci da automatizzare (volutamente!), un'arma a doppio taglio da cui possono trarre vantaggio sia gli sviluppatori che i criminali.

Con le API stiamo tutti compiendo gli stessi errori che abbiamo fatto 20 anni fa con i siti web.

Le API per lo sviluppo della sicurezza

Pensando alle API, che consentono ai componenti software di interfacciarsi gli uni con gli altri, viene in mente l'interfaccia che collega i team addetti alla sicurezza e allo sviluppo.

La sicurezza e lo sviluppo non hanno mai realmente parlato la stessa lingua, in parte perché presentano experience, terminologia e priorità molto diverse tra loro. Tuttavia, questa scarsa relazione sta diventando sempre più importante, specialmente con l'inevitabile pressione di offrire più funzioni, accelerare i rilasci e fare tutto nell'ambito del DevOps per seguire l'ultima tendenza in termini di progettazione. Cosa deve cambiare per un miglior allineamento?

Soffermiamoci un attimo. Nella mia esperienza, i criminali sono i migliori difensori. Non voglio suggerire di assumere il personale InfoSec dall'unità PLA 61398 o da Lazarus Group. La mia definizione di criminale è chi ha una mentalità e competenze offensive: uno che non rispetta le regole (o trasgressore).

I trasgressori hanno una comprensione molto migliore dell'arte del possibile. Spesso, quando un trasgressore segnala una vulnerabilità, viene applicata una patch solo per il tempo necessario ad ostacolare la prova di fattibilità, ma nulla di più. Basterà cambiare un carattere da minuscolo a maiuscolo o provare lo stesso attacco su un'altra parte del sito web, e andrà a segno un'altra volta. Perché? L'addetto al sistema di difesa non ha eseguito una verifica accurata. Perché no? Perché l'addetto non conosce realmente il motivo per cui funziona un attacco e come pensa un criminale. Non si tratta semplicemente di dire ad un addetto di pensare come un criminale se non l'ha fatto prima: sarebbe come dire ad un parrucchiere di pensare come un idraulico e aspettarsi che, magicamente, abbia le competenze necessarie per incanalarsi in un pensiero laterale.

Chi non è stato un trasgressore tende anche a qualificare il rischio in maniera errata, solitamente con un atteggiamento pessimista. Per queste persone, gli attacchi sembrano atti di magia nera perché non li hanno mai sferrati in prima persona, pertanto iniziano a pensare che tutto sia un'enorme falla nella sicurezza. In definitiva, si tratta di un atteggiamento improduttivo in quanto, in questo contesto, tutto è una priorità e niente è una priorità.

D'altro lato, i trasgressori puri tendono a non comprendere come funzionano i processi di sviluppo poiché semplificano eccessivamente le complessità e i costi associati alle modifiche al codice e, spesso, non parlano la lingua degli sviluppatori. Probabilmente, non hanno mai dovuto costruire o lanciare un prodotto, non sono in grado di comprendere le complessità degli scambi commerciali e, a volte, non vogliono farlo perché è molto più divertente violare le cose.

Ecco il punto in cui ci troviamo oggi: sono pochissimi i professionisti della sicurezza in grado di stare a cavallo tra questi due mondi in modo efficace. Sono le proverbiali "mosche bianche": persone perfette per il lavoro, ma difficili da trovare

E ora?

Nell'ambito della gestione dei prodotti, a volte si sente parlare di Time to Value (o Time To First Value, TTFV), un'espressione che descrive quanto tempo serve ad un cliente nuovo o potenziale prima di comprendere il valore di ciò che gli è stato effettivamente venduto. Ovviamente, l'obiettivo è farglielo capire "nel più breve tempo possibile".

I professionisti che si affacciano al settore della sicurezza devono minimizzare il Time to Value.

Relativamente alla portata e all'impatto delle recenti violazioni, Jeremiah Grossman ha ipotizzato che la questione non risiede tanto nelle migliori tecniche offensive, ma piuttosto nella maggiore capacità dei criminali rispetto ai team InfoSec di assumere e formare persone con ruoli di livello base.

Le lacune di competenze nella cybersicurezza di cui si sente tanto parlare dipendono in larga parte da un atteggiamento riluttante (o peggio, un'incapacità sistemica) nel formare le persone. E se ci si impegnasse maggiormente in questa formazione sul lavoro? Ad esempio, apprendere contemporaneamente le competenze di offesa e difesa sfruttando le vulnerabilità degli attacchi SQL injection, quindi scrivere il codice per risolvere questo problema e, infine, capire come eludere la risoluzione. Dopodiché, ripetere l'esercizio su un'applicazione aziendale in circolazione da 10 anni che deve superare tutti i test di integrazione e funzionalità prima di poter distribuire la correzione.

L'altra strategia che possiamo adottare consiste nell'assumere sviluppatori addetti alla sicurezza! Trovate l'opportunità di dimostrare alcuni attacchi e cercate sviluppatori con idee brillanti. Sfruttate la curiosità e cominciate da lì. Immaginate un team addetto alla sicurezza costituito da ex trasgressori che lavorano a fianco di ex sviluppatori. Potrete disporre del meglio dei due mondi e, cosa più importante, potrete iniziare a farli integrare e collaborare tra loro invece di metterli uno contrapposto all'altro.

Esclusione di responsabilità: le opinioni e i punti di vista espressi in questo saggio sono quelli dell'autore e non riflettono necessariamente quelli di Akamai.

Introduzione

Questa edizione del rapporto sullo stato di Internet - Security è incentrata sulla sicurezza delle API (Application Programming Interface). Si tratta di un argomento importante: Gartner ha previsto che, "entro il 2022, l'abuso di API, che ora si verifica raramente, diventerà il più frequente vettore di attacco, causando violazioni di dati per le applicazioni web aziendali".[1]

Oltre alla ricerca da noi condotta, per stilare questo rapporto abbiamo collaborato con Veracode, le cui informazioni sul settore della sicurezza delle applicazioni hanno contribuito fortemente a meglio definire il panorama delle API. Come avete notato, questo rapporto contiene un saggio curato da Chris Eng, Chief Research Officer di Veracode,

Riguardo allo stato degli attacchi online, abbiamo esaminato il traffico degli attacchi in un periodo di 18 mesi tra gennaio 2020 e giugno 2021. A giugno 2021, in un solo giorno, Akamai ha registrato 113,8 milioni di attacchi, una cifra pari a tre volte il numero degli attacchi osservato nello stesso periodo di tempo nel 2020. Con 6,2 miliardi di tentativi registrati, l'attacco SQL Injection (SQLi) rimane in cima all'elenco degli attacchi web, seguito dagli attacchi LFI (Local File Inclusion) con 3,3 miliardi e XSS (Cross-Site Scripting) con 1.019 miliardi.

Il credential stuffing ha rappresentato la fonte di un flusso costante di attacchi registrati finora quest'anno, con flessioni e picchi nei primi due trimestri del 2021. Akamai ha registrato due picchi degni di nota a gennaio e a maggio del 2021, con un numero di attacchi di credential stuffing che ha superato quota 1 miliardo. Casualmente, i due picchi sono stati registrati il secondo giorno di ogni mese, senza alcuna indicazione dei motivi per cui i criminali hanno concentrato i loro sforzi in questo giorno particolare.

Gli Stati Uniti sono stati il bersaglio principale degli attacchi sferrati contro le applicazioni web nel periodo osservato, con un traffico quasi sei volte superiore a quello registrato in Inghilterra, la seconda nazione che ha subito più attacchi. Gli Stati Uniti hanno anche primeggiato come paese di origine degli attacchi, scalzando dalla vetta la Russia, con un traffico quasi quattro volte superiore.

In conclusione, il traffico degli attacchi DDoS è rimasto costante finora nel 2021, con picchi registrati nel primo trimestre di quest'anno. A gennaio, Akamai ha registrato 190 eventi DDoS in un solo giorno, seguito da 183 eventi in un solo giorno a marzo.

Sicurezza delle API

L'interfaccia con il mondo

Le API sono ovunque. Se un'applicazione o un servizio sono disponibili in rete, sicuramente vengono supportati, in qualche modo, da un'API. Oggigiorno, le API supportano le applicazioni mobili, l'Internet delle cose (IoT), i servizi di assistenza clienti basati sul cloud, le applicazioni interne o di partner e molto altro.

Oltre ai problemi di sicurezza derivanti dalle API, bisogna considerare anche l'aspetto delle performance. Akamai esamina i miglioramenti apportati alle performance dalle API su base regolare poiché il traffico delle API passa dai server di origine agli edge server sulla CDN. Questa configurazione accelera gli accessi e garantisce la massima disponibilità,

ma conduce ad un crescente problema. Le organizzazioni che proteggono le API con le tradizionali soluzioni per la sicurezza di rete registrano, al massimo, un moderato successo, quando le cose vanno bene, perché i vecchi standard per la difesa delle reti possono arrivare solo a questo livello. Gli stessi rischi esistenti per siti e applicazioni web riguardano perlopiù anche le API, tuttavia vanno affrontati in modo separato.

Le API espandono notevolmente la superficie di attacco di cui devono preoccuparsi le organizzazioni, pertanto gli addetti ai sistemi di difesa e allo sviluppo devono impegnarsi maggiormente per affrontare queste aree problematiche. Secondo Gartner, "entro la fine del 2021, il 90% delle applicazioni abilitate per il web, passando da una percentuale del 40% nel 2019, aumenterà la propria superficie di attacco per le API pubbliche piuttosto che per l'interfaccia utente".[1]

La buona notizia è che i responsabili aziendali e i team addetti alla sicurezza stanno già rafforzando i propri sistemi di sicurezza relativamente alle pratiche correlate alle API. Tuttavia, il potenziale di crescita è enorme e i criminali sono sicuramente in grado di trarre vantaggio dalle falle presenti nella sicurezza delle API.

Gartner ha previsto che, "entro il 2022, l'abuso di API, che ora si verifica raramente, diventerà il più frequente vettore di attacco, causando violazioni di dati per le applicazioni web aziendali".[1]

La difesa del codice

L'OWASP (Open Web Application Security Project) è una fondazione no profit che si occupa di migliorare la sicurezza dei software, ampiamente nota per la pubblicazione del suo elenco OWASP Top 10, in cui vengono messi in evidenza i principali rischi alla sicurezza affrontati dalle applicazioni web, inclusi gli attacchi injection, la violazione dei dati di autenticazione e controlli degli accessi, l'esposizione di dati sensibili e le configurazioni errate.

L'ultima versione dell'elenco OWASP Top 10, alla data della stesura di questo rapporto, è stata pubblicata nel 2017 (A1-10:2017). L'OWASP ha anche pubblicato un elenco API Security Top 10 (API1-10:2019), la cui sovrapposizione con il primo elenco menzionato è sostanziale. Nel corso del tempo, il settore della sicurezza ha sviluppato una serie di metodi per tenere traccia delle vulnerabilità dei software e per associarli alle guide sulle best practice, come gli elenchi OWASP, incluse le definizioni CWE (Common Weakness Enumeration), che vengono gestite dalla MITRE.

Considerando il numero delle vulnerabilità divulgate pubblicamente e degli attacchi segnalati, è chiaro che le API stanno incontrando gli stessi tipi di problemi affrontati da anni dalle applicazioni basate sul web.

Esaminiamo, ad esempio, le credenziali integrate nel codice sorgente.

Il 29 luglio 2020, Cisco ha distribuito una patch per DCNM (Data Center Network Manager), dopo aver stabilito che le credenziali integrate nel codice sorgente dell'API REST potevano consentire ad un criminale remoto non autenticato di eludere l'autenticazione ed eseguire i comandi desiderati con privilegi di amministratore. Questa vulnerabilità nota come CWE-798 (credenziali integrate nel codice sorgente) può essere direttamente collegata all'API2:2019 e all'A2:2017 tramite l'OWASP nell'ambito della violazione dei dati di autenticazione.

Le API sono considerate versatili e offrono caratteristiche di facilità d'uso e accesso sia per le aziende che per gli utenti finali. La maggior parte delle organizzazioni utilizza le API per scopi interni o esterni, per i clienti o i partner aziendali oppure per una combinazione di questi motivi. Le principali funzioni e aspettative in termini di sviluppo e distribuzione delle API riguardano l'integrazione e l'accesso ai dati, tuttavia l'utilizzo delle API per servizi e linee aziendali digitali (ad es., Checkr, Twilio, Scale, Segment) è aumentato in modo esponenziale negli ultimi anni.

Questa versatilità, tuttavia, rappresenta anche un punto a loro sfavore poiché, spesso, il compromesso tra facilità d'uso e sicurezza crea problemi di gestione.

Un buon esempio di quanto sia difficile bilanciare la facilità d'uso con la sicurezza si può trovare nell'informativa sulla sicurezza delle API pubblicata su Twitter a febbraio 2020. In una comunicazione pubblica, Twitter ha reso noto che un elevato numero di account fittizi stanno sfruttando le relative API e associando nomi utente con numeri di telefono. Le API consentono agli utenti di individuare più facilmente gli amici desiderati con un'apposita funzione, che, tuttavia, è stata sfruttata dai criminali per sottrarre i dati. Questo problema può essere monitorato con i codici CWE-284 e A2:2017/API5:2019 tramite l'OWASP.

Un altro esempio della difficoltà di gestire il compromesso tra disponibilità e sicurezza si può trovare nella pubblicazione di quest'anno (luglio 2021) curata dal ricercatore Muhammad Sholikhin, che è riuscito a identificare i membri di alcuni gruppi chiusi su Facebook tramite le API di questo gigante dei social media. Poiché si suppone che l'iscrizione ad un gruppo chiuso sia riservata, Facebook ha dovuto metterci una pezza e ha ricompensato il ricercatore per il suo lavoro.

GitLab, una popolare piattaforma di gestione di repository Git, ha riscontrato gli stessi problemi osservati su Facebook. Tramite il suo programma Bug Bounty, il ricercatore Riccardo Padovani ha rivelato che è stato possibile visualizzare progetti di gruppi privati tramite le API. Alla fine, GitLab ha tamponato il problema e ha ricompensato il ricercatore per la sua pubblicazione.

I criminali tentano di accedere a Twilio

Anche se i criminali prestano attenzione alle best practice suggerite e ai consigli divulgati in termini di sicurezza, cercando vari modi per sfruttare attività aziendali e servizi che seguono questi suggerimenti, hanno commesso alcuni errori apparentemente di minore importanza (anche se con costose ripercussioni).

Uno di questi servizi è Twilio, un servizio di API che ha riscontrato un enorme successo poiché consente agli sviluppatori di migliorare le user experience con la gestione delle comunicazioni tramite SMS, chat, video e messaggi e-mail.

Al fine di garantire l'efficacia delle credenziali necessarie per accedere a Twilio, il servizio richiede agli sviluppatori di memorizzare il SID (Security Identifier) e il token di autenticazione dell'account Twilio in modo da impedire eventuali accessi non autorizzati. La documentazione aziendale pone l'accento su questo sistema di protezione e suggerisce di adottare variabili di ambiente per implementarlo. Le variabili sono una soluzione logica solitamente associata allo sviluppo delle API, la cui modalità di archiviazione risulta, tuttavia, importante.

Twilio afferma che è possibile memorizzare il SID e i token di autenticazione nei file .env, i quali, tuttavia, andrebbero aggiunti al file .gitignore per evitare di caricare contenuti riservati come testo normale. Gli amministratori dei siti web, spesso, incoraggiano a memorizzare tali file all'esterno della directory radice principale (/www/ o /public_html/) per impedire gli accessi non autorizzati, ma questa operazione non viene sempre eseguita, diventando, pertanto, un costoso errore.

I criminali cercano attivamente di accedere al SID e ai token di autenticazione di Twilio cercando nei siti web nomi di comuni variabili di ambiente, come quelli che finiscono con .env, .dev o anche .prod. Gli honeypot di Akamai hanno rilevato che il 25 agosto 2021 sono state effettuate ricerche per i file "twilio.env" (all'incirca nel periodo in cui questo rapporto è stato finalizzato). Inoltre, poco prima del rilevamento di queste ricerche da parte dei nostri honeypot, SANS ha registrato ricerche simili nel tentativo di accedere a Twilio.

Chi effettua queste ricerche per sottrarre il SID e i token di autenticazione può riuscire nell'intento e attuare una ricognizione passiva sugli account violati per venderli ad eventuali acquirenti, come illustrato nella Figura 1.

Poiché è possibile usare gli account Twilio per scopi generici di spamming o phishing passivo e mirato, nonché per altre frodi, la loro protezione è estremamente importante.

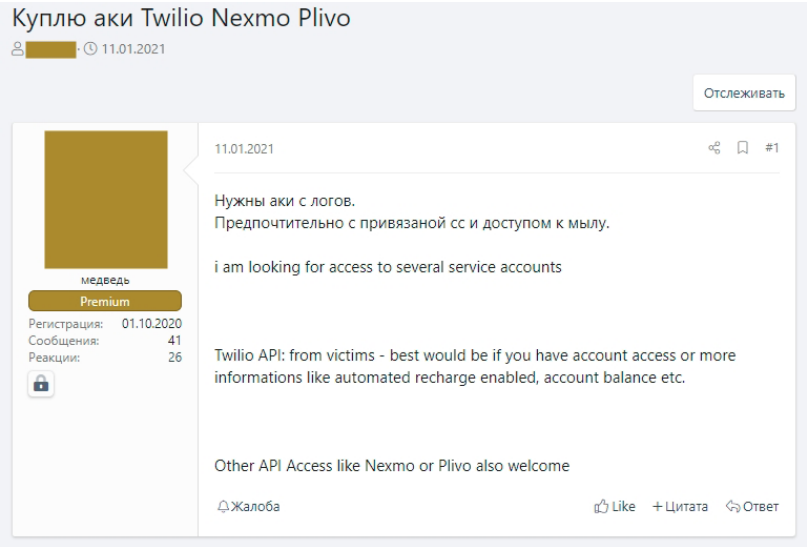

Fig. 1: Un criminale cerca di acquistare le credenziali di accesso a Twilio ed è particolarmente interessato agli account che hanno abilitato la funzione di ricarica automatica

Settore sanitario

Come abbiamo detto in precedenza, l'integrazione e l'accesso ai dati sono le principali aspettative di un'organizzazione per lo sviluppo e la distribuzione delle API. Questa linea di pensiero è più evidente nel settore sanitario che altrove.

Nel 2020, quando la pandemia causata dal COVID-19 ha iniziato a cambiare le nostre vite, la possibilità di utilizzare applicazioni sanitarie mobili (mHealth) per gestire le ricette, le cartelle cliniche e, persino, gli appuntamenti dai medici, ha assunto un'importanza cruciale. Inoltre, le persone hanno continuato ad utilizzare gli strumenti di monitoraggio della salute per la dieta e gli allenamenti.

A febbraio 2021, Alissa Knight, partner di Knight Ink, ha pubblicato un rapporto sponsorizzato da Approov (fornitore di soluzioni per la sicurezza delle API) sulla sicurezza delle API e sulle applicazioni mHealth. I risultati emersi da questa indagine durata sei mesi sono stati sorprendenti.

Premettendo che i risultati non sarebbero stati attribuiti alle singole aziende, Knight è riuscita ad esaminare apertamente 30 diverse applicazioni mHealth. Alcune delle organizzazioni coinvolte nella ricerca di Knight presentavano oltre 1.000 API nelle loro applicazioni. Una delle sfide correlate con la sicurezza delle API consiste nell'identificarne e monitorarne la distribuzione: tanto maggiori sono le dimensioni dell'area di implementazione delle API, tanto più difficile risulta difenderle.

La metà delle API esaminate nel lavoro di Knight ha consentito di accedere ai risultati patologici, radiografici e clinici di altri pazienti, nonché alle cartelle cliniche ospedaliere, superando il livello di autorizzazione concesso.

In tutte le applicazioni mHealth esaminate, le API si sono rivelate vulnerabili alla BOLA (Broken Object Level Authorization). Pertanto, Knight è riuscita a visualizzare le informazioni di identificazione personale (PII) e le informazioni sanitarie protette (PHI) relative ad utenti non assegnati agli account clinici esaminati. Inoltre, quasi l'80% delle applicazioni esaminate ha presentato chiavi API integrate nel codice sorgente (anche senza alcuna scadenza), token, chiavi private e combinazioni di nomi utente e password integrate in fase di progettazione.

"Le pratiche di cybersicurezza di base, ad esempio evitare di inserire nel codice sorgente nomi utente e password e autorizzare tutte le richieste, è un problema endemico nelle applicazioni mHealth", ha concluso il rapporto di Knight.

"Le aziende produttrici di applicazioni mHealth devono adottare un approccio Zero Trust per la sicurezza di app e API al fine di garantire che la semplice autenticazione non sia necessaria per autorizzare un utente ad accedere ai dati richiesti".

In breve, le applicazioni mHealth esaminate da Knight avevano la necessità di sottoporsi ad un controllo in termini di sicurezza, che si è concretizzato grazie alla cooperazione tra Knight e le società di gestione delle applicazioni.

Spring Boot

Java Spring Framework consente alle organizzazioni di sviluppare applicazioni aziendali che vengono eseguite sulla JVM (Java Virtual Machine). Java Spring Boot (Spring Boot) semplifica e velocizza lo sviluppo di applicazioni e servizi con Spring Framework. Per loro stessa natura, le applicazioni Spring Boot devono includere o interagire con un'API.

Per la stesura di questo rapporto, Veracode ha condiviso con Akamai le informazioni relative alla strategia di sicurezza adottata per 5.000 applicazioni Spring Boot nel corso del tempo. I risultati sono stati desunti dai test sulla sicurezza delle applicazioni condotti in modo statico e dinamico, oltre alle analisi manuali. Nel complesso, il 100% delle applicazioni esaminate ha presentato almeno una vulnerabilità. Anche se non tutte le vulnerabilità sono uguali, il punto è: scrivere il codice senza vulnerabilità non è facile come sembra. Si tratta di un'operazione che richiede tempo e risorse considerevoli, oltre al supporto dei responsabili.

Associazione tra CWE e OWASP

Se si desidera associare i codici CWE all'elenco OWASP Top 10, è disponibile un grafico di mappatura sul sito web di CWE.

Le seguenti associazioni sono correlate ai risultati della ricerca Spring Boot condotta da Veracode:

CWE 73:
A5:2017 tramite OWASP

CWE 80:
A7:2017 tramite OWASP

CWE 89:
API8:2019 e A1:2017 tramite OWASP

CWE 117:
API8:2019 e A1:2017 tramite OWASP

CWE 209:
API7:2019/A6:2017 tramite OWASP

CWE 259:
API2:2019 e A2:2017 tramite OWASP

CWE 327:
A3:2017 tramite OWASP

La maggior parte delle applicazioni Spring Boot esaminate (86%) ha rivelato una vulnerabilità agli attacchi CRLF (Carriage Return and Line Feed) Injection (CWE 117) nei registri. Esistono molti tipi di vulnerabilità CRLF, ma in questo contesto ci riferiamo a quei casi in cui il software in questione non neutralizza (o neutralizza in modo errato) i dati scritti nei registri. In casi simili, un criminale potrebbe riuscire a falsificare i dati del registro o iniettare contenuti dannosi nei registri stessi. È possibile sfruttare i difetti CRLF per sferrare altri attacchi, come gli attacchi XSS. In una nota correlata, il 42% delle applicazioni Spring Boot incluse in questo set di dati è risultato vulnerabile agli attacchi XSS di base (CWE 80). A scopo di chiarezza, tuttavia, durante i test, Veracode segnala anche le vulnerabilità agli attacchi XSS con il codice CWE 79, 83 o 86, a seconda dei casi.

Si sono verificati, inoltre, vari problemi relativi al codice: nel 68% delle applicazioni esaminate, alcune risorse sono state rilasciate in modo errato prima di essere rese disponibili per il riutilizzo (CWE 404), mentre il 50% di esse ha utilizzato algoritmi crittografici violati o rischiosi (CWE 327).

Anche le password integrate nel codice sorgente hanno rappresentato un problema nel 47% delle applicazioni Spring Boot esaminate (CWE 259) per la gestione dei messaggi di errore, in cui erano incluse informazioni sensibili nel 44% delle applicazioni (CWE 209). Alcune delle applicazioni Spring Boot (41%) hanno consentito agli utenti di controllare o influenzare i percorsi utilizzati nelle operazioni del file system (CWE 73), fornendo un livello di controllo sulle applicazioni non necessariamente rispondente alle previsioni. Infine, nel 31% delle applicazioni esaminate, sono risultate vulnerabilità agli attacchi XXE (XML External Entities), che possono essere usate per estrarre dati, eseguire richieste remote da un altro server o lanciare attacchi di tipo Denial-of-Service.

Oltre ai 10 problemi identificati nelle applicazioni Spring Boot, il 21% di esse ha presentato vulnerabilità CWE 89 o SQLi (Veracode segnala anche le vulnerabilità SQLi con i codici CWE 564 e 943). Nella stessa percentuale di applicazioni, inoltre, sono emerse vulnerabilità che ricadono nel codice CWE 601 relativo a problemi di reindirizzamento diretto.

Perché applicazioni e API sono vulnerabili?

Considerando i dati delle applicazioni mHealth e Spring Boot, oltre agli altri esempi menzionati in precedenza, è comprensibile domandarsi perché nelle API si riscontrano gli stessi problemi registrati nelle applicazioni web in passato. Come risulta evidente, alcune di queste applicazioni sono state lasciate consapevolmente vulnerabili. L'anno scorso, l'ESG (Enterprise Strategy Group) ha condotto un sondaggio per conto di Veracode, che ha rivelato come le organizzazioni distribuiscono consapevolmente codice vulnerabile.

Nello specifico, il 48% delle organizzazioni che ha partecipato al sondaggio ha ammesso di distribuire regolarmente codice vulnerabile. All'interno di questo gruppo, il 54% ha affermato che il codice vulnerabile viene distribuito per rispettare una scadenza importante con l'intenzione di risolvere le vulnerabilità note in un rilascio successivo. L'ordine di procedere alla distribuzione viene spesso dato da un team (responsabile dello sviluppo/analista della sicurezza - 28%), un responsabile dello sviluppo (24%), un analista della sicurezza (21%) o singoli sviluppatori che valutano la priorità di ogni problema rilevato (15%).

Tra gli altri motivi elencati che portano alla distribuzione di codice vulnerabile, figura la sensazione di considerare le vulnerabilità a basso rischio (49%) e il rilevamento delle vulnerabilità in una fase del ciclo di sviluppo troppo prossima alla scadenza per cui sia possibile risolverle (45%).

Tutti i motivi elencati sono un classico esempio di un compromesso in termini di sicurezza. Il rischio di violazione è stato accettato per soddisfare le esigenze aziendali, il che potrebbe non sembrare giusto o corretto, ma è la realtà.

La sicurezza è alquanto difficile da realizzare di per sé, ma la sicurezza nello sviluppo è costituita da complesse scelte stratificate che, di solito, riguardano innanzitutto il supporto delle attività aziendali. Se un'azienda deve commercializzare un prodotto e una libreria di codici utilizzata in fase di sviluppo risulta vulnerabile proprio nel momento in cui il prodotto deve essere immesso sul mercato, per molti responsabili aziendali, la richiesta di lanciarlo comunque subito è ragionevole con l'impegno di distribuire una patch nel più breve tempo possibile (purché la vulnerabilità non sia talmente critica da mettere a rischio l'azienda o i suoi clienti/utenti).

Le vulnerabilità a basso rischio, come quelle che richiedono una serie complessa di operazioni, un certo livello di accesso o un set specifico di situazioni da sfruttare, potrebbero perdere la loro priorità a favore del previsto lancio del prodotto in questione. È una decisione rischiosa che le aziende di tutto il mondo affrontano ogni giorno.

Un concetto del tipo *Prima la distribuzione, poi le patch* non significa ignorare completamente la sicurezza, ma semplicemente valutarla in modo da impedirle di interferire con le attività aziendali o le user experience. Ecco perché è importante considerare la sicurezza come parte integrante del ciclo di sviluppo.

La maggior parte dei partecipanti al sondaggio condotto da ESG (78%) ha affermato che gli analisti della sicurezza sono stati direttamente coinvolti con i loro sviluppatori per esaminare le funzioni e il codice (31%), per realizzare modelli di minacce (28%) o per partecipare alle riunioni di sviluppo giornaliere (19%).

Un altro elemento dei problemi di sicurezza riscontrati nello sviluppo di API e applicazioni riguarda la dipendenza dal codice open-source. Il rapporto ESG afferma che, mentre le moderne codebase dipendono notevolmente dal codice open-source, meno della metà dei partecipanti al sondaggio (48%) ha affermato di utilizzare strumenti per monitorare lo stato dei progetti open-source.

Infine, gli sviluppatori stessi, che devono produrre il codice entro una certa scadenza e seguire il concetto *Prima la distribuzione, poi le patch*, solitamente non vengono formati. Il rapporto ESG ha riferito che il 29% dei partecipanti al sondaggio ha affermato che gli sviluppatori non dispongono delle competenze necessarie per mitigare le vulnerabilità identificate nel loro codice. In realtà, il 53% dei partecipanti al sondaggio ha affermato che vengono formati solo gli sviluppatori che si uniscono al team per la prima volta come parte di un programma di formazione annuale; in caso contrario, si suppone che gli sviluppatori debbano provvedere da sé alla loro formazione.

Akamai - La parola ai numeri

Come menzionato in precedenza, i dati relativi agli attacchi forniti in questo rapporto riguardano un periodo di 18 mesi, da gennaio 2020 a giugno 2021.

Attacchi web

Akamai ha registrato 113,8 milioni di attacchi sferrati in un solo giorno a giugno 2021, ossia una cifra più che triplicata rispetto al numero di attacchi osservati a giugno 2020 (Figura 2).

L'attacco SQLi si distingue, nella Figura 3, come il principale tipo di attacco sferrato negli ultimi 18 mesi, con 6,2 miliardi di tentativi registrati. Al secondo posto alquanto distanziato dal primo, figura l'attacco LFI con 3,3 miliardi di tentativi registrati e, infine, l'attacco XSS con oltre 1 miliardo di tentativi.

Poiché attualmente la maggior parte del traffico online si basa sulle API, gli attacchi web osservati da Akamai mirano quasi certamente ad organizzazioni con servizi e applicazioni per interazioni pubbliche.

In realtà, quando Akamai ha esaminato i dati condivisi da Veracode, molte delle applicazioni Spring Boot esaminate sono risultate vulnerabili agli attacchi SQLi e XSS. Si è verificata un'enorme sovrapposizione con gli attacchi sferrati contro applicazioni web e API e, come sottolinea il nostro saggio, i problemi del passato iniziano a ricomparire negli odierni cicli di sviluppo.

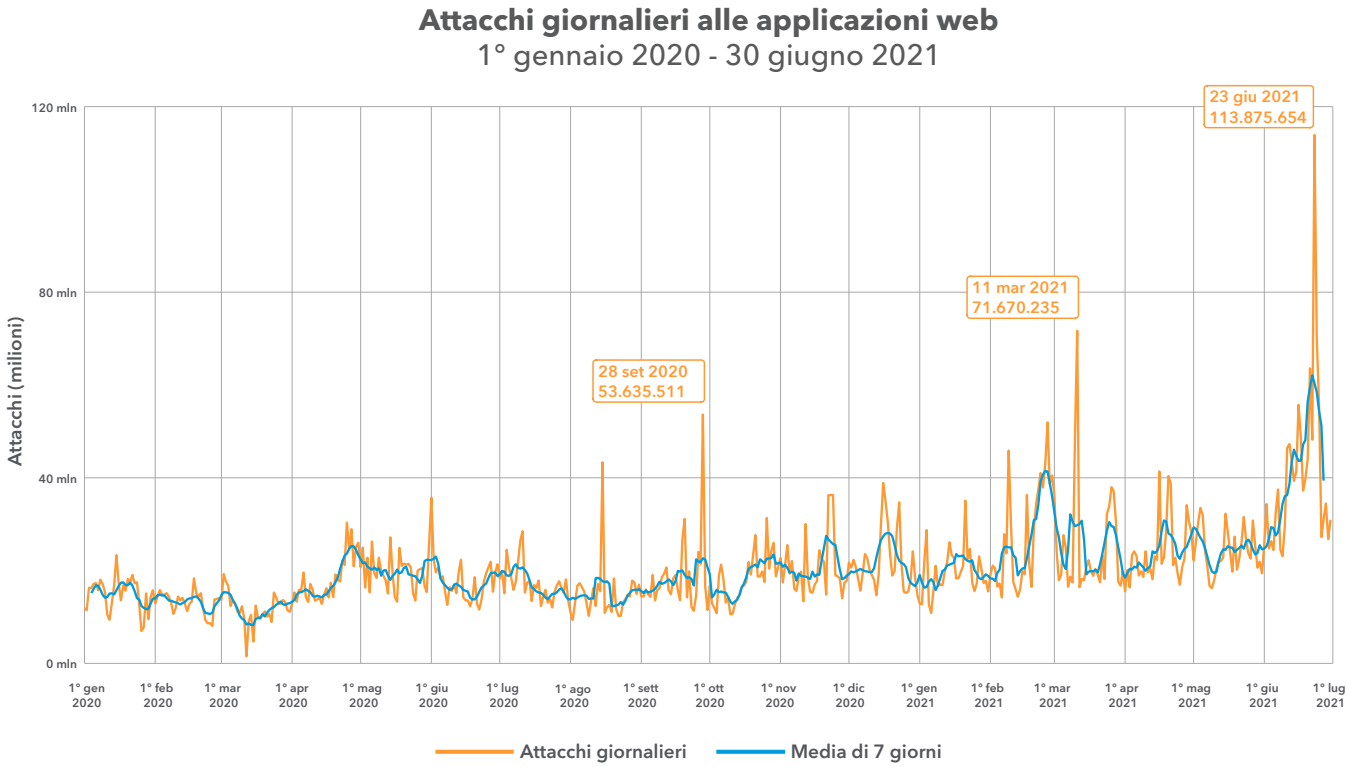

Fig. 2: A giugno 2021, gli attacchi web sono cresciuti in modo esponenziale, raggiungendo un picco di 113,8 milioni di attacchi in un solo giorno

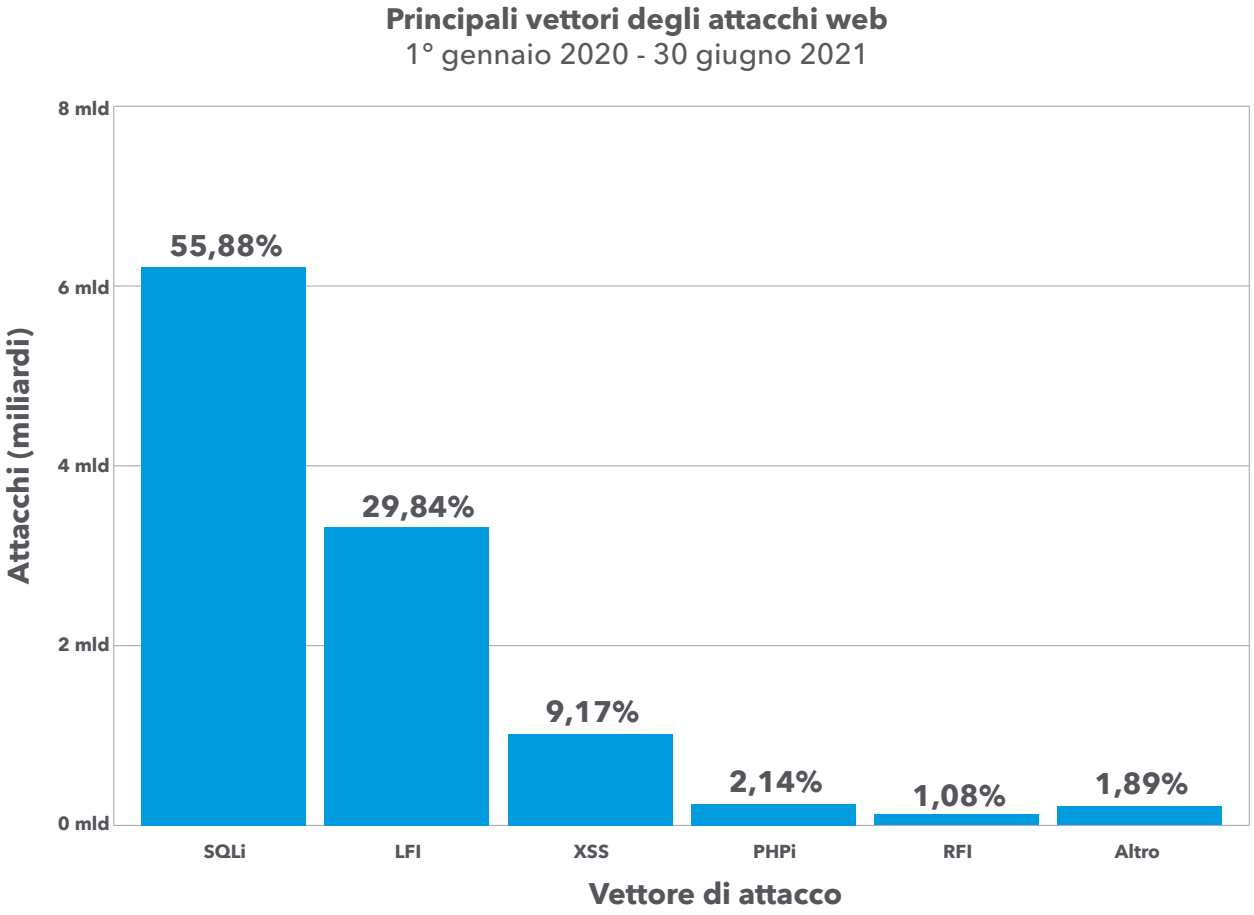

Fig. 3: SQLi rimane il principale vettore degli attacchi web poiché i criminali cercano di sfruttare le applicazioni e le API per accedere alle informazioni sensibili o protette

Abuso di credenziali

Negli ultimi 18 mesi, gli attacchi di credential stuffing sono rimasti stabili, con flessioni e picchi nei primi due trimestri del 2021, seguiti da due attacchi degni di nota registrati a gennaio e a maggio di quest'anno. In questo periodo, gli attacchi di credential stuffing hanno superato quota 1 miliardo al giorno (Figura 5).

La causa principale di questi picchi di attacchi è ignota, tuttavia è possibile correlarli ad una serie di servizi di credenziali comparsi su vari mercati nel 2020, che continuano ad operare fino ad oggi. Uno di questi servizi, che ha iniziato ad offrire l'accesso a febbraio 2020, promette di fornire aggiornamenti costanti e credenziali sottoposte a dehashing nel formato user:pass, esattamente quello che cercano i criminali quando acquistano o scambiano elenchi di combinazioni.

A onor del vero, alcuni di questi servizi nascondono tentativi di phishing, tuttavia queste operazioni non durano molto poiché i marketplace tendono ad autoregolarsi per contrastare i truffatori. Il rivenditore a cui si fa riferimento nella Figura 4 stava vendendo l'accesso a più di 200 GB di credenziali al momento dell'offerta iniziale. Il servizio offre agli acquirenti varie combinazioni di base o mirate, inclusi i record user:pass con una particolare focalizzazione su retail, gaming, cryptomercati, paesi, ecc.

Inoltre, la sua raccolta viene aggiornata su base alquanto regolare. Ad esempio, nei suoi post, è stato affermato che oltre 430 milioni di combinazioni sono state aggiunte tra febbraio e aprile 2021, mentre a maggio sono state aggiunte 144 milioni di combinazioni.

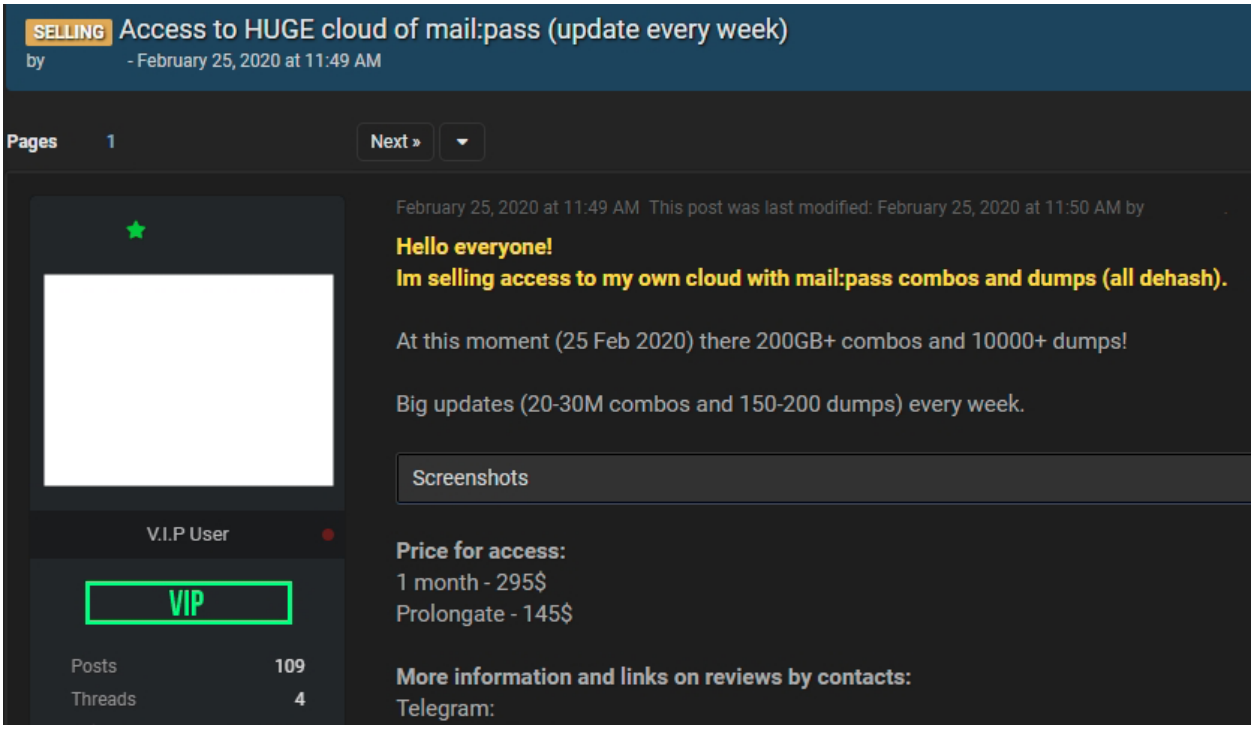

Fig. 4: Un servizio presente in un forum popolare offre un flusso costante di credenziali da poter utilizzare negli attacchi di credential stuffing

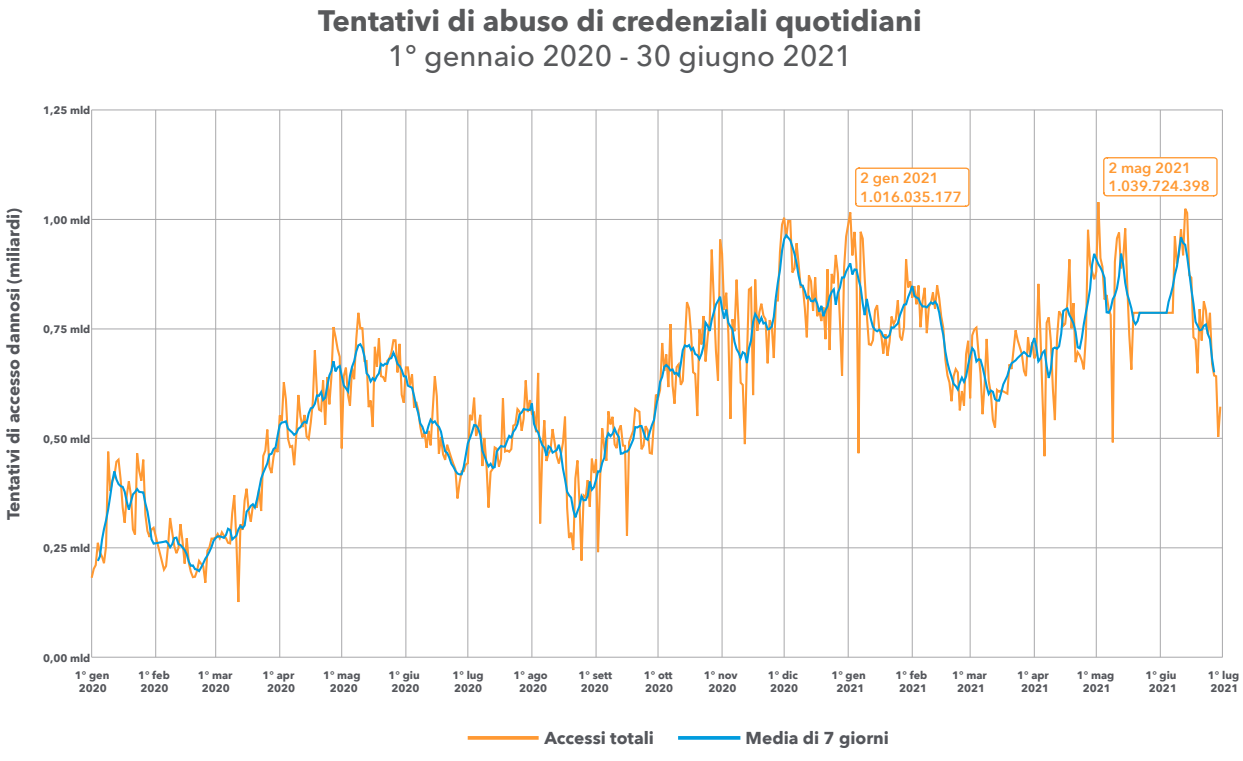

Fig. 5: Negli ultimi 18 mesi, gli attacchi di credential stuffing sono rimasti stabili, con picchi di oltre 1 miliardo di attacchi al giorno registrati in due giorni nel primo e nel secondo trimestre del 2021

Potrebbe trattarsi di una coincidenza, ma i due picchi nei nostri set di dati sono stati registrati il secondo giorno di ogni mese, rispettivamente, a gennaio e maggio 2021, senza alcuna indicazione sui motivi per cui i criminali hanno concentrato i loro sforzi in questo giorno particolare rispetto ad altri giorni degli stessi mesi.

Origini e bersagli

Gli Stati Uniti affermano di essere stati il bersaglio principale degli attacchi sferrati contro le applicazioni web nel periodo osservato, con un traffico quasi sei volte superiore a quello registrato in Inghilterra, la seconda nazione che ha subito più attacchi (Figura 6), a cui seguono nell'elenco India, Austria e Canada. Poiché gli Stati Uniti ospitano i principali bersagli degli attacchi su Internet, la loro prima posizione in questa classifica non ci sorprende.

Gli Stati Uniti hanno anche primeggiato come paese di origine degli attacchi, scalzando dalla vetta la Russia, con un traffico quasi quattro volte superiore (Figura 7).

Le principali aree di destinazione degli attacchi alle applicazioni web
1° gennaio 2020 - 30 giugno 2021

AREA DI DESTINAZIONE	TOTALE DI ATTACCHI	POSIZIONE GLOBALE
Stati Uniti	5.998.188.041	1
Regno Unito	1.021.638.223	2
India	825.061.439	3
Austria	309.373.274	4
Canada	282.846.738	5

Fig. 6: Gli Stati Uniti sono risultati nuovamente il paese maggiormente preso di mira dagli attacchi web, seguiti dall'Inghilterra

Le principali aree di origine degli attacchi alle applicazioni web
1° gennaio 2020 - 30 giugno 2021

AREA DI ORIGINE	TOTALE DI ATTACCHI	POSIZIONE GLOBALE
Stati Uniti	4.019.434.857	1
Russia	1.146.258.871	2
India	910.264.770	3
Paesi Bassi	642.859.781	4
Germania	640.368.111	5

Fig. 7: Gli Stati Uniti sono risultati il principale paese di origine degli attacchi, con un traffico pari quasi al quadruplo di quello originato dalla Russia

La sorgente degli attacchi è una metrica interessante da seguire poiché Akamai può vedere solo la parte finale della catena degli attacchi, ossia il punto in cui il criminale alla fine riesce a stabilire una connessione con il suo bersaglio. Anche se gli Stati Uniti sono in cima a questa classifica, non significa che gli attacchi si siano originati in questo paese. In realtà, i criminali trasmettono il traffico degli attacchi da varie origini con l'obiettivo di nascondere la propria identità e la posizione da cui operano.

DDoS

Il traffico degli attacchi DDoS è rimasto costante finora, con picchi registrati nel primo trimestre del 2021. A gennaio, Akamai ha registrato 190 eventi DDoS in un solo giorno, seguito da 183 eventi a marzo. Si prevede che, durante la fine di quest'anno, potranno registrarsi altri picchi poiché i criminali tendono a preferire gli attacchi DDoS rispetto ad altri tipi di attacchi.

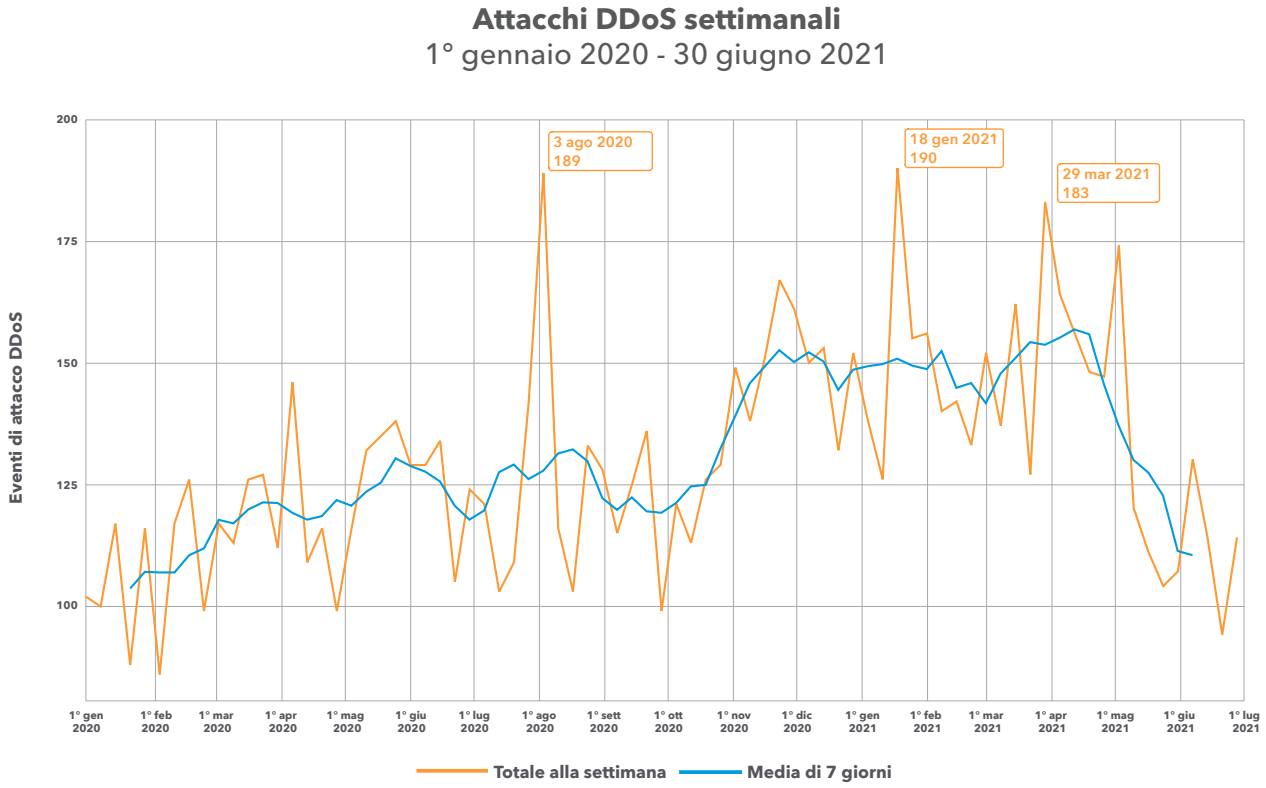

Fig. 8: Il numero degli attacchi DDoS è rimasto costante nel periodo esaminato di 18 mesi, con picchi registrati nel primo trimestre del 2021

Conclusione

La sicurezza delle applicazioni, sia dal lato delle API che dal lato dello sviluppo delle applicazioni web, è una combinazione complessa di funzioni, caratteristiche ed esigenze aziendali. Trovare un equilibro in questo contesto non è semplice come si potrebbe pensare.

Come dimostrano i nostri dati, i criminali sono chiaramente all'opera per sviluppare nuove tecniche e metodi di attacco e le funzionalità delle API rappresentano uno dei loro obiettivi principali. Anche se i team si stanno spostando verso una sicurezza integrata nel ciclo di vita delle sviluppo, il processo richiede tempo, pertanto le organizzazioni vengono lasciate in una situazione difficile, che le costringe, in alcuni casi, a lanciare codici vulnerabili noti poiché vitali per le esigenze aziendali.

Oltre ad alcune pratiche consigliate come riportato nell'appendice A, è chiaro che lo sviluppo delle app e l'implementazione delle API richiedono una formazione più costante: solo in questo modo i team addetti allo sviluppo potranno adempiere alle aspettative in loro riposte in termini di sicurezza.

Appendice A - Best practice: Sicurezza delle API

Per questo rapporto, abbiamo parlato con numerosi esperti e considerato varie fonti di informazioni relative alla sicurezza delle API e delle applicazioni, tra cui Veracode, OWASP, MITRE e 42Crunch.

1. Rilevare le API e monitorarle come in un inventario. Molte organizzazioni hanno riscontrato un problema relativo alle API della cui esistenza non erano consapevoli, pertanto, è fondamentale conoscere dove si trovano le API e per cosa vengono usate. A ciò sono correlate anche le API esterne utilizzate dall'organizzazione. È necessario identificare e proteggere queste API o perlomeno registrarle come possibili fattori di rischio e valutarle.

2. Una volta saputo dove si trovano le API, è necessario sottoporle a test per comprendere quali vulnerabilità nascondono al loro interno. A tal scopo, sono richiesti strumenti di test e una solida preparazione nel campo dello sviluppo, nonché una forma di partnership con i team addetti alla sicurezza.

Sarà necessario discutere sulla tolleranza ai rischi e sui piani approntati per risolvere tempestivamente le vulnerabilità. Per iniziare, è necessario individuare la presenza di chiavi integrate nel codice sorgente e chiamate logiche, oltre a verificare se vi sia la possibilità di compromettere il traffico delle API tramite un attacco di impersonificazione. È inoltre consigliabile verificare nei dispositivi di storage e nei repository la presenza di chiavi da poter usare per compromettere le API o gli elementi ad esse associati.

3. Sfruttare l'infrastruttura WAF esistente, nonché eventuali soluzioni per la protezione dei dati e la gestione delle identità, insieme a specifici strumenti per la sicurezza delle API, sia durante la fase di sviluppo che durante il lancio dei prodotti. Assicurare, inoltre, che la sicurezza delle API sia un processo costante e non la selezione di una semplice casella durante lo sviluppo. Le nuove vulnerabilità e i nuovi attacchi vengono rilevati continuamente, mentre i controlli singoli lasciano vulnerabile la superficie di attacco.

4. Per quanto riguarda le policy delle API, cercare di evitare di utilizzare policy univoche per ciascuna API, invece di favorire l'adozione di una serie di policy da poter riutilizzare. Inoltre, non inserire le policy del codice direttamente nelle API che richiedono protezione per evitare di violare la separazione dei componenti meccanici, aggiungere complessità non necessarie o un ulteriore sovraccarico di lavoro per gli addetti alla gestione del codice e negare la visibilità per i team che si occupano di sicurezza. Come regola generale, rendere nullo o negare il livello di accesso predefinito per qualsiasi risorsa. In tal modo, viene applicato il privilegio minimo e l'autenticazione diventa un requisito costante.

5. Lo sviluppo delle API deve includere (ad alcuni livelli) diverse parti interessate, tra cui i team addetti allo sviluppo, alle operazioni di sicurezza e rete, alla gestione delle identità (se questa mansione non rientra nei team operativi), i responsabili dei rischi, gli architetti della sicurezza e i team addetti alla questioni di conformità/legali (per garantire che il prodotto sia conforme a tutte le leggi normative e di governance).

Quando si tratta di sicurezza delle API al livello dell'OWASP, APISecurity.io gestisce una scheda di riferimento rapido sviluppata da 42Crunch, la cui lettura può risultare molto interessante.

L'elenco OWASP Top 10 per le applicazioni è disponibile sul sito web dell'OWASP. Ogni voce offre suggerimenti e consigli rivolti agli addetti ai sistemi di difesa.

Appendice B - Metodologie

Attacchi alle applicazioni web

Questi dati descrivono gli avvisi a livello di applicazione generati da Kona Site Defender e Web Application Protector. I prodotti attivano questi avvisi quando rilevano un payload dannoso all'interno di una richiesta a un sito web o un'applicazione protetta. Gli avvisi non indicano una compromissione riuscita. Sebbene questi prodotti consentano un alto livello di personalizzazione, i dati qui presentati sono stati raccolti senza prendere in considerazione le configurazioni personalizzate delle proprietà protette.

I dati sono stati presi da Cloud Security Intelligence (CSI), uno strumento interno per lo storage e l'analisi degli eventi di sicurezza rilevati sull'Akamai Intelligent Edge Platform. Questa è una rete di circa 300.000 server distribuiti in più di 4.000 sedi su oltre 1.400 reti in 135 paesi. I nostri team addetti alla sicurezza utilizzano questi dati, misurati in petabyte al mese, per effettuare ricerche sugli attacchi, segnalare comportamenti dannosi e includere ulteriore intelligence nelle soluzioni Akamai.

Abuso di credenziali

I tentativi di abuso di credenziali sono stati identificati come tentativi di accesso non riusciti ad account che utilizzano un indirizzo e-mail come nome utente. Utilizziamo due algoritmi per distinguere tra tentativi di abuso e utenti reali che non riescono a digitare. Il primo algoritmo è una semplice regola volumetrica che conta il numero di tentativi non riusciti a un indirizzo specifico. Questo metodo differisce da ciò che una singola organizzazione potrebbe essere in grado di rilevare perché Akamai mette in correlazione dati provenienti da centinaia di organizzazioni.

Il secondo algoritmo utilizza i dati provenienti dai nostri servizi di rilevamento dei bot per identificare un abuso di credenziali da botnet e strumenti noti. Una botnet ben configurata può evitare il rilevamento volumetrico distribuendo il suo traffico su numerosi obiettivi, usando un gran numero di sistemi durante la sua operazione di scansione, o distribuendo il traffico nel tempo, solo per menzionare alcuni esempi di elusione.

Anche questi dati sono stati presi dal repository di CSI. Un cliente con un significativo volume di attacchi è stato rimosso da questo set di dati prima del 2020, poiché non disponeva di un intero anno di dati.

È importante notare che la raccolta di dati sull'abuso di credenziali (CRAB) è stata interrotta dal 19 maggio al 7 giugno 2021 e, di conseguenza, non verranno raccolti dati in questo periodo. Per continuare a utilizzare il set di dati CRAB, il numero totale dei tentativi di accesso dannosi è stato calcolato con un semplice valore mediano.

Nel secondo trimestre, questo numero è stato calcolato sulla base dei dati esistenti e attribuito come numero totale dei tentativi di accesso dannosi al giorno in questo periodo di tempo. Questo metodo, insieme a molti altri, è stato verificato sulla base dell'intero set di dati del primo trimestre 2021 e il calcolo del semplice valore mediano è stato stabilito come la stima più vicina al valore totale noto. Non è stato attribuito alcun dato relativo agli account (ossia, segmento verticale principale o secondario, paese di origine/destinazione degli attacchi, regioni) e, pertanto, non saranno disponibili calcoli relativi a tali informazioni nel secondo trimestre 2021.

DDoS

Prolexic Routed difende le organizzazioni dagli attacchi DDoS reindirizzando il traffico di rete tramite gli scrubbing center di Akamai e consentendo solo il traffico pulito. Gli esperti del SOC (Security Operations Center) di Akamai personalizzano i controlli di mitigazione proattivi per rilevare e bloccare immediatamente gli attacchi ed eseguono analisi del traffico rimanente in tempo reale per determinare ulteriori misure di mitigazione, in base alle necessità.

Gli eventi di attacco DDoS vengono rilevati dal SOC o dalla stessa organizzazione mirata, a seconda del modello di implementazione scelto, "always-on" o "on-demand", ma il SOC registra i dati per tutti gli attacchi mitigati. In modo simile al traffico delle applicazioni web, l'origine è determinata dall'origine del traffico IP prima della rete di Akamai.

Riconoscimenti

Responsabili editoriali

Martin McKeay
Editorial Director

Steve Ragan
Senior Technical Writer, Editor

Amanda Goedde
Senior Technical Writer, Managing Editor

Chelsea Tuttle
Senior Data Scientist

Autrice ospite: Chris Eng
Chief Research Officer, Veracode

Marketing

Georgina Morales Hampe
Project Management

Shivangi Sahu
Program Management

All'interno di questo rapporto, Akamai ha menzionato i dati sulla ricerca e le analisi di Gartner, le cui informazioni provengono dal seguente rapporto:

1) ID Gartner: G00404900, 1° marzo 2021, API Security: What You Need to Do to Protect Your APIs (Sicurezza delle API: cosa fare per proteggere le API)

Altri rapporti sullo stato di Internet - Security

Leggete le edizioni precedenti e date un'occhiata ai prossimi rapporti sullo stato di Internet - Security: akamai.com/soti

Altre informazioni sulla ricerca delle minacce Akamai

Restate aggiornati con le ultime novità in termini di intelligence sulle minacce, rapporti sulla sicurezza e ricerche sulla cybersicurezza disponibili sul sito akamai.com/our-thinking/threat-research

Accesso ai dati dal rapporto

Potete visualizzare i grafici e i diagrammi citati in questo rapporto in versioni di alta qualità. L'utilizzo e la consultazione di queste immagini sono forniti a scopo gratuito, purché Akamai venga debitamente citata come fonte e venga conservato il logo dell'azienda: akamai.com/sotidata

Akamai garantisce experience digitali sicure per le più grandi aziende a livello mondiale. L'Akamai Intelligent Edge Platform permea ogni ambito, dalle aziende al cloud, permettendovi di lavorare con rapidità, efficacia e sicurezza. I migliori brand a livello globale si affidano ad Akamai per ottenere un vantaggio competitivo grazie a soluzioni agili in grado di estendere la potenza delle loro architetture multicloud. Più di ogni altra azienda, Akamai avvicina agli utenti app, experience e processi decisionali, tenendo lontani attacchi e minacce. Il portfolio Akamai di soluzioni per l'edge security, le web e mobile performance, l'accesso aziendale e la delivery di contenuti video è affiancato da un servizio clienti di assoluta qualità e da un monitoraggio 24 ore su 24, 7 giorni su 7, 365 giorni all'anno. Per scoprire perché i principali brand mondiali si affidano ad Akamai, visitate il sito www.akamai.com, blogs.akamai.com o @Akamai su Twitter. Le informazioni di contatto internazionali sono disponibili all'indirizzo www.akamai.com/locations. Data di pubblicazione: 10/21.