Un “hacker” come amico

Quante volte abbiamo sentito parlare di “**alleanze con il nemico**”?

Normalmente ci contorniamo di persone amiche, persone che riteniamo importanti, anche se capita di essere traditi e di comprendere dopo molto tempo che quelle persone alle quali tenevamo così tanto non erano tali. Al contrario degli amici, ci sono persone che vanno contro ogni nostra regola e decisione e lo fanno palesemente, in contrasto su tutto. Conoscere le loro strategie fornisce vantaggi indiscussi perché ci aiuta a comprendere i loro pensieri ed evitare percorsi critici e discussioni su tematiche a rischio. Insomma, capire il ragionamento del “Nemico”, di un antagonista o di un concorrente nel business, ci aiuta nella vita, nel lavoro e non da meno nella sicurezza informatica.

Robert Green riportava, come seconda tra le “48 Leggi del potere”, quanto segue;

Se l’occasione giusta si presenta, chiedi piuttosto il supporto
del nemico di un tempo, dovendo dimostrare la sua lealtà
nei tuoi confronti, sarà più leale di un amico.

Se non hai nemici, createne di nuovi.

Ma anche **Friedrich Nietzsche** recitava, “***Ama i tuoi nemici perché essi tirano fuori il meglio di te***” in quanto la sfida e la collaborazione sono opposti che si attraggono e generano sempre miglioramento, introducendo viste completamente “disruptive” e ignote. Tutto questo dà ritorni non immaginabili nel medio e lungo termine, oltre a portare all’attenzione “quel bivio” che prima non era per nulla ipotizzato.

Ma ora caliamoci nel contesto della sicurezza informatica e

capiamo cosa c’entra tutto questo. I termini “*Security by design*” e “*Privacy by design*” sono slogan da tempo abusati.

Sembrano quegli slogan pubblicitari che tutti conoscono, e spesso li si dice solo perché “genera importanza” (a Roma si direbbe “fa fico”) e fa sembrare di essere attenti a problemi comuni, ma che poi, immancabilmente, **non facciamo nostri**. Quanti di noi si sono chiesti se questo principio di per sé corretto (e mi riferisco principalmente ai progettisti e agli sviluppatori software) lo stanno veramente mettendo in atto nel modo giusto in ogni fase del progetto?

Di fatto la teoria è molto lontana dalla pratica e su questi temi **l’operatività e i dettagli “sono tutto”.** Molti nuovi sistemi risultano carenti di minime implementazioni di sicurezza che portano in fase di controllo a rilevare “impatti reali” importanti e potenziali data-breach dirompenti, ma siamo già troppo tardi.

Gli attacchi informatici e i concetti di cybersecurity oggi **sono sempre più complessi e di difficile comprensione** e il mestiere dell’ingegnere software è totalmente “avulso” e spesso in antitesi a quello dell’ingegnere di sicurezza perché orientato al business e al servizio. Ahimè, sono rari i casi di convivenza di una corretta progettazione con una rigorosa implementazione della sicurezza.

Ci sono poi molte persone che pensano che gli esperti di cybersecurity “di processo” o “high-level” possano garantire un corretto ritorno in termini di esposizione al rischio dei sistemi, risolvendo errori (ma spesso “orrori”) commessi in quella delicata fase che è la progettazione, senza sapere che (facendo un calzante paragone) si stanno facendo **operare al cuore da un chirurgo plastico.**

Per i progetti importanti, chiamate gli “Hacker”!

Ecco un nuovo slogan. Un sistema per non essere violato deve essere progettato come sicuro, ma questo lo sappiamo. È la fase di progettazione, la fase nella quale fare scelte strategiche, calandole nel contesto del business, quella normalmente più importante in un sistema nuovo; anche questo lo sappiamo.

Ma, operativamente, lo stiamo facendo bene?

Avere a bordo un esperto in materia di **“ethical-hacking” che valuti il progetto fin dall’inizio**, nei dettagli (e anche nei processi) in termini di scelta dell’architettura, degli algoritmi di cifratura, della rete, delle librerie software, della corretta gestione dello sviluppo sicuro del codice guardandolo dal punto di vista della sicurezza informatica e delle possibili “violazioni”: **è questo che fa la differenza**.

Perché i “requisiti minimi di sicurezza” **devono essere calati in un contesto complesso** e vanno fatte scelte tecnologiche, ovvero quanto necessario a “metterli in atto”; chi meglio del “nemico” sa come contrastare le minacce, visto che le conosce con precisione?

Il concetto di “Red” e “Blue” si evolve in quanto i “Red” non sono essenziali solo nella fase di produzione, ma occorre averli “all’interno”, nei progetti. Possiamo quindi immaginare **una nuova figura di progetto** (e sbizzarriamoci con il nome) ad esempio “Expert on Cyber-Security & Operational Supervisor” oppure “Expert on Cyber-Security Data-Breach Prevention” **che possa, stando dentro i progetti**, visionare, suggerire, approvare e presidiare le scelte tecnologiche dal punto di vista della Cyber-Security. Di seguito alcuni spunti:

Fase 1: Progettazione

- *Approvare i disegni architetturali e fornire consigli che consentano una corretta segregazione degli ambienti oltre a introdurre, qualora necessarie, ulteriori componenti architetturali se non presenti negli schemi.*
- *Approvare la visibilità in rete dei servizi (sia di erogazione che di amministrazione) in relazione all'esposizione del rischio della piattaforma.*
- *Approvare e indirizzare la scelta degli algoritmi di cifratura/hashing dell'intera piattaforma.*
- *Consigliare la scelta dei sistemi operativi, anche sulla base dei sistemi di patching management automatizzati e indirizzarne la scelta.*
- *Approvare, in collaborazione con il team di sviluppo, il linguaggio di programmazione da utilizzare, gli ambienti, i RAD e la gestione del codice sorgente.*
- *Definire un catalogo delle librerie da utilizzare (in particolare modo se open source) per quel progetto, evitando componenti con ricorrenti problematiche di sicurezza su soluzioni make.*
- *Definire il corretto livello di Log da implementare e le componenti da tracciare.*
- *Verificare che le componenti "embedded" nei software buy (con particolar riferimento all'open source) non abbiano vulnerabilità critiche ricorrenti e che risultino aggiornabili semplicemente (senza riscrittura di codice) in fase di "long-security".*
- *Definire le librerie di depurazione degli input da far utilizzare al team di sviluppo.*
- *Definire linee guida di standard di progetto (sia per gli sviluppatori che per i gestori di esercizio) che assicurino e sensibilizzino, una volta che il sistema è in produzione, i livelli di sicurezza attesi.*

Fase 2: Delivery

- *Controllare e supervisionare la corretta implementazione delle architetture di rete e di piattaforma con test di verifica sulle configurazioni degli apparati di rete.*
- *Controllare che i team di sviluppo adottino le scelte tecnologiche effettuate nella fase di progettazione.*
- *Formare i team di sviluppo per evitare vulnerabilità con impatti gravi non rilevabili dagli scanner statici/dinamici, come ad esempio "broken access control" o "broken authentication" e verificare che abbiano compreso il modello implementativo.*
- *Verificare in delivery lo stato del patching delle piattaforme e assicurarsi che il patching management venga adottato, sia come processo che allo stato dell'arte.*
- *Verificare che le componenti di patching management automatizzato funzionino correttamente, soprattutto sulla superficie internet.*

Fase 3: Pre-Produzione

- *Verificare che i requisiti di Hardening, Patching e Sviluppo Sicuro del codice siano stati correttamente implementati e verificare che i processi siano a regime con un'attività di controllo interno.*
- *Svolgere scansioni automatizzate, sia infrastrutturali che dinamiche, del codice per verificare la presenza di possibili falle di sicurezza.*
- *Svolgere una verifica attraverso tecniche manuali (penetration test) per assicurare che non siano presenti vulnerabilità non rilevabili dagli strumenti automatizzati.*
- *Verificare il patching level della piattaforma, anche effettuando scansioni in white-box.*

L'ultima fase, ovvero la pre-produzione, è quella che porta un

reale vantaggio in quanto l'ethical hacker, essendo stato presente in tutte le fasi di realizzazione del progetto e conoscendo l'infrastruttura, è in possesso di informazioni "privilegiate" ed è a conoscenza di tutti i punti critici dell'applicazione.

Nella fase di avvio della produzione potrà condurre un penetration test con metodologia grey-box per verificare, in modo semplice, la presenza di ulteriori falle fino a quel momento non controllate.

Conclusioni

Le grandi guerre spesso si sono vinte con alleanze strategiche con chi prima era un nemico, beneficiando delle sue strategie, delle sue tecnologie e delle sue armi.

Gli "hacker" conoscono – il fronte – meglio di chiunque altro.
Scovate i migliori "ethical-hacker" tra i vostri fornitori
e portateli nei progetti.

È questo il grande vantaggio e la grande opportunità da cavalcare nella "security by design", ovvero ***"aumentare le competenze cyber utilizzando chi sa violare i sistemi"*** oltre a consentire di progettarli bene, realizzarli, collaudarli per poi gestirli in modo semplice e sicuro nel tempo.

Dobbiamo anche essere consapevoli che, sebbene il nostro **"Risk-appetite" sia alto, la nostra "Risk-tollerace" spesso risulta inferiore alle attese** e se un data-breach viene fatto alle big-company di oltre oceano, noi siamo più a rischio perché la tecnologia non la facciamo noi, ma la usiamo. Occorre quindi prestare massima attenzione a questi aspetti in un'era in cui basta anche un solo data breach per minare la sopravvivenza di un'intera azienda.

Tutto questo per dire che la differenza nell'applicare bene la sicurezza informatica (a parte l'impalcatura di norme, di requisiti e di gestione del rischio e degli slogan spesso

abusati da tutti) la farà sempre più ***“la gestione e il presidio dei dettagli”,*** perché un sistema è come un castello di carte al quale basta una piccola folata di vento per farlo completamente collassare su se stesso.

Tutto sta ad evitare che quella piccola folata ci sia.

Articolo a cura di **Massimiliano Brolli**

https://www.ictsecuritymagazine.com/articoli/un-hacker-come-amico/