L’attività di un Ethical Hacker

Esempi pratici, risultati e contromisure consigliate

Massimo Biagiotti

BUSINESS e
drive your e-success

Information Technology

- Chiunque operi nel settore sa che il panorama dell'IT è in continua evoluzione
  - Nuovi prodotti, Nuove tecnologie, **Nuove esigenze**
- Il mercato è ricco di soluzioni (software ed hardware) che aiutano le organizzazioni a risolvere i loro problemi e tentano di provvedere alle loro necessità
- Generalmente, le organizzazioni mettono in piedi sistemi informativi tramite
  - L'acquisizione e la configurazione di prodotti per le specifiche esigenze e la loro integrazione per giungere alla soluzione desiderata
  - La creazione di soluzioni "ad hoc" tramite lo sviluppo di nuovo software

Information Security

- Da un lato, le soluzioni proposte dai singoli vendor sono sicure "off the shelf"
  - Testate, collaudate ed aggiornabili in tempo reale
  - Non è detto che la configurazione specifica di tali prodotti o la loro integrazione sia altrettanto sicura
  - Il fattore umano è determinante
- Dall'altro, le soluzioni sviluppate "ad hoc" presentano spesso problematiche di sicurezza non trascurabili
  - Non testate, non collaudate, spesso di difficile aggiornamento
- Ogni sistema è, quindi, composto da componenti sicure, ma potenzialmente mal configurate/integrate e componenti non testate, e quindi probabilmente non sicure.
- In questo contesto si collocano le attività di Ethical Hacking

Ethical Hacking

- Il cosiddetto “Ethical Hacker” si occupa di esaminare un sistema per rilevarne i problemi di sicurezza
  - Dovuti ad errori nelle fasi di integrazione
  - Dovuti ad errori nelle fasi di sviluppo
- Con l’attività di “Ethical Hacking” è possibile stimare quanto un sistema sia sicuro
  - Ovvero, capire quanto il livello di sicurezza effettivo si avvicini al livello di sicurezza desiderato
- A valle dell’attività è possibile individuare un insieme di contromisure per rimediare agli eventuali problemi incontrati
  - Contromisure a breve termine (workaround)
  - Contromisure a lungo termine

Ethical Hacking

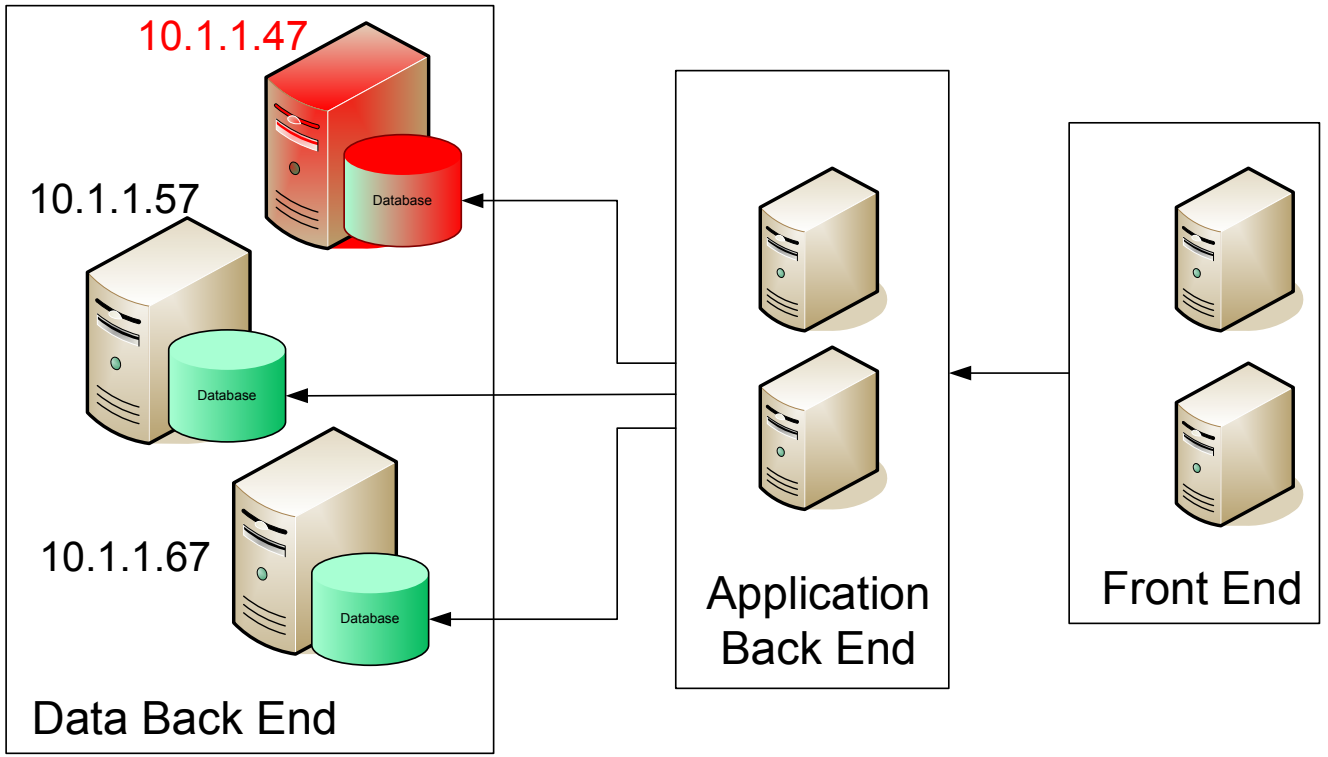
- Le attività non possono essere esaustive per loro natura
  - Vengono rese efficienti ed efficaci dall'esperienza degli ethical hacker
  - Vengono potenziate ed estese ad insiemi molto vasti di componenti tramite l'adozione di strumenti automatici
- Sono condotte simulando le tipologie di attacco più comuni
  - Utente esterno all'organizzazione, attacco condotto da Internet
  - Utente interno all'organizzazione, attacco condotto dalla Intranet aziendale
    - Sulle componenti applicative, come Web Application
    - Sulle componenti infrastrutturali, come apparati di rete, server, etc…
- Vengono ora illustrati due risultati di due distinte attività condotte dalla Intranet
  - Su componenti infrastrutturali e su applicazioni web

Ethical Hacking e Infrastrutture Critiche

- Verranno fatti esempi che non sono riconducibili a realtà particolari per ovvi motivi etici e perché nessuna realtà è più o meno attaccabile rispetto ad un'altra.
- Tutte le infrastrutture sono a rischio se ritenute "interessanti" per qualche motivo e nessuna è considerabile a priori "fuori tiro".
- Esiste un mercato di rivendita delle vulnerabilità, ciò che può non essere immediatamente utile per l'hacker lo può essere per qualcuno disposto a pagare.
- Tale aspetto è tanto più serio per chi gestisce ed è responsabile di infrastrutture "Critiche".
- In occasione del prossimo convegno di Frascati "Critis 2008" ci sarà modo di approfondire il tema.

Esempio 1: Problematiche legate all'integrazione

> Questo primo esempio è l'analisi di parte delle attività effettuate sulla seguente struttura di rete:

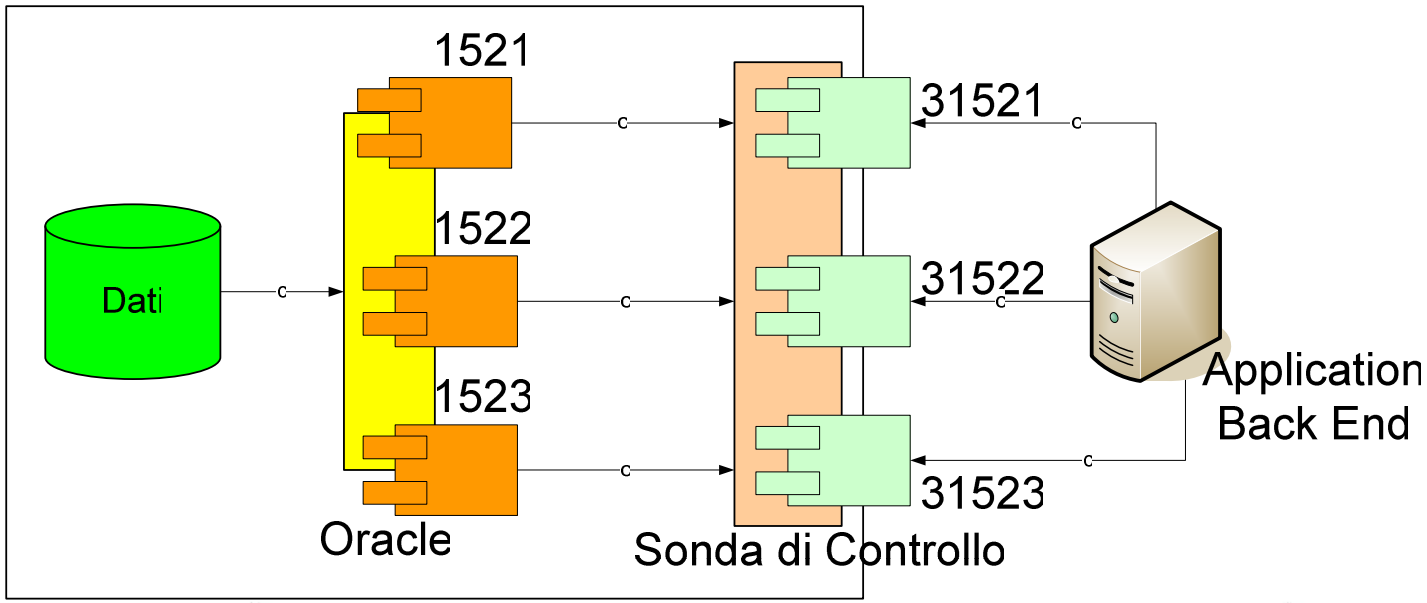


Esempio 1: Problematiche legate all'integrazione

- L'architettura di rete prevede che la comunicazione fra il back end dell'applicazione e i database server avvenga esclusivamente tramite alcune porte:
  - 31521, 31522, 31523
  - I database server presentano tutte le altre porte chiuse, ad eccezione della porta 22 per l'amministrazione remota (SSH)
- Su tutti i database server è installato il DBMS Oracle
  - I database Oracle contengono dati particolarmente critici
- Per proteggere i database è stato installato, su ogni server, un appliance che monitora le query eseguite
  - Una suite che agisce da sonda di controllo rispetto alle richieste inviate al database

Esempio 1: Problematiche legate all'integrazione

- Ogni database server è quindi organizzato secondo lo schema
  - La sonda autorizza le richieste del back-end soltanto se rispettano delle regole (tabelle, caratteri speciali, etc..)
  - La sonda è **responsabile del tracciamento** delle operazioni

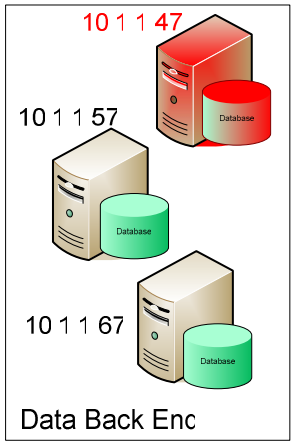


Esempio 1: Problematiche legate all'integrazione

- Oracle, come molti database, utilizza porte TCP per la comunicazione
- In questo caso le porte Oracle (1521,1522,1523) erano aperte soltanto in locale e chiuse all'esterno
- Le porte 31521, 31522, 31523 sono le porte che apre la sonda verso l'esterno, per permettere alle applicazioni di accedere al database
  - La sonda controlla le query inviate ad Oracle
  - L'architettura di rete è pensata per evitare che si acceda ad Oracle senza passare per la sonda

Esempio 1: Problematiche legate all'integrazione

- Dal punto di vista di progettazione, l'architettura è sicura (?)
  - Ma grazie ad un piccolo buco, sono state aggirate le protezioni previste
- Durante il test è stata individuata una credenziale banale nel server 10.1.1.47
  - Username: **alessio**
- Tramite questa credenziale banale si è alterata la struttura di rete
  - Sono state aperte le porte 41521, 41522, 41523 verso l'esterno
  - Il traffico diretto verso tali porte viene inviato **direttamente** ad Oracle
  - In questo modo si **aggira la protezione della sonda**
- Per far ciò, è bastato un semplice SSH Tunnel:
  - `ssh –a –o "GatewayPorts yes" –L 41521:localhost:1521 alessio@10.1.1.47`



Esempio 1: Problematiche legate all'integrazione

- A partire da ciò, è stato possibile:
  1. Esaminare lo stato di sicurezza del database Oracle (che non era a noi raggiungibile in precedenza, grazie alla sonda)
  2. Individuare alcune credenziali banali sul database
  3. Effettuare un escalation di privilegi su tali credenziali tramite exploit noti e disponibili su web:
     - http://rawlab.mindcreations.com/codes/exp/oracle/sys-lt-findricsetV2.sql
  4. **Individuare i dati critici protetti, e violarne la riservatezza**
     - Un hacker non etico avrebbe potuto violarne la disponibilità o, peggio, l'integrità
- Tutto ciò passando inosservati alla sonda di controllo
  - Aggirando completamente le regole di sicurezza impostate
  - Aggirando completamente le procedure di tracciamento

Esempio 2: SQL Injection

- Nel seguito, descriveremo gli impatti di un attacco di tipo SQL Injection, effettuato durante un'attività
- Le problematiche di SQL Injection sono legate ad errori di programmazione
  - Mancata od incompleta validazione dell'input
  - L'applicazione si fida dell'utente e dei valori che egli dà in input al sistema
- L'attività si è svolta su un'applicazione web attestata sulla rete interna del cliente

Esempio 2: SQL Injection

- Il form seguente presenta 4 campi:
  - I primi 3, in verde, sono correttamente verificati e validati rispetto a caratteri speciali
  - Il quarto, in rosso, non è correttamente validato

Esempio 2: SQL Injection

- Dopo alcuni tentativi, è stato possibile alterare il quarto campo inserendo una stringa di SQL Injection efficace :

```
> ' UNION
SELECT
    USERID,password,'','',USERID,TIME_MOD,'','','','','',
    '','',U_Alias,U_Alias,U_Alias,U_Alias,U_Alias,U_Alias
    ,U_Alias,U_Alias,USER_ID,'','','','',''
FROM XXXXXX.XXXX_USER
WHERE USERID>95838550 AND USERID <95838580 –-
```

- Grazie a questa stringa si riesce ad ottenere lo username e la password (memorizzata come hash) di un gruppo di utenti del sistema.

Esempio 2: SQL Injection

- Il risultato
  - Nelle colonne “`Nome e Cognome`” e “`Codice Fiscale`” sono presenti, rispettivamente, lo username e l’hash della password per l’accesso all’applicativo:

Nome e cognome	Codice Fiscale		Registrazione	Data di registrazione	
	T1012daf900128	0604cc91336679f2dbfb2161551		06-09-2007 03:30:21	
	T101143ad00134	1ce38e0bb9c610cd0057d93ca10		06-09-2007 03:30:36	
	T1012cf0b00012	d928d9b2e6cf5fda139d7121dbf4		06-09-2007 03:30:38	
	T1031732c80250	2f970e1157dc1c6d1d7b7ea6c867		06-09-2007 11:38:55	
	T10584a6b80298	3f1855c4f9bb3f5f8e7bddfb14eb		06-09-2007 17:00:18	
	T101816b8801ab	80342d1082a1eff50de0498e422		07-09-2007 09:27:15	
	T102e927e8034c	66604d153bb7bf53a24cfd7c9aa6		07-09-2007 09:43:54	
	T1031df0d80335	ae2b2d9a62bca05e10af95f8081		07-09-2007 10:38:12	
	T1052b9300089c	b51cd6844dc0989ce2cbaddbf67b		07-09-2007 10:47:11	
	T1027d2c2003b7	5a7ff8864bbb6cdf13cf00983402		07-09-2007 12:30:05	

BUSINESS

Esempio 2: SQL Injection

- A partire da ciò, è stato possibile:
  1. Raffinare la query per recuperare tutti gli amministratori applicativi
  2. Effettuare un attacco di tipo brute force sugli hash delle password degli amministratori applicativi
  3. Accedere all'applicativo con le credenziali banali così ottenute e creare un nuovo amministratore applicativo
     - Che ha la possibilità di impersonare qualunque utente
     - Che può **visualizzare i dati degli utenti e violarne la riservatezza**
  4. Utilizzare un form di amministrazione precedentemente inaccessibile, vulnerabile ad attacchi di tipo Path Traversal
  5. Accedere a **qualunque file presente su filesystem** dell'Application Server leggibile all'utente che esegue l'Application Server
  6. ….

Contromisure

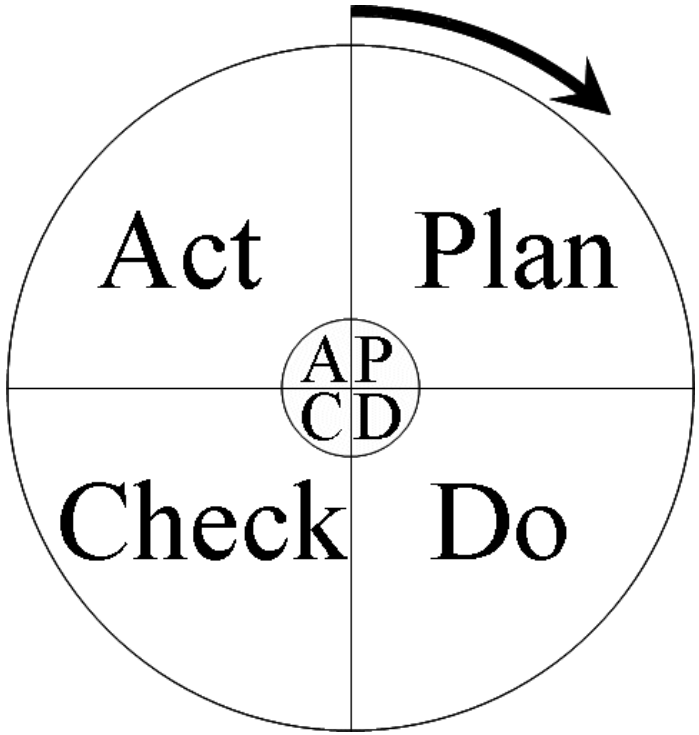
- Gli esempi proposti sono decisamente differenti ma:
  - È stato possibile sfruttare entrambe le problematiche a partire da un errore umano
  - Nell'esempio 1 l'errore è stato commesso in fase di esercizio
  - Nell'esempio 2 l'errore è stato commesso in fase di sviluppo
- Si deve individuare la contromisura adatta
  - Esistono varie contromisure per la stessa vulnerabilità
    - Workaround
    - Riscrittura di parte del codice
    - Aggiunta di moduli hardware
    - ….
  - La contromisura più adatta viene decisa in base alle esigenze ed alla gravità della vulnerabilità
    - Che dipende dal livello di rischio ritenuto accettabile

Conclusioni

- La dinamicità del panorama IT e le nuove esigenze del mercato obbligano le organizzazioni a muoversi verso soluzioni
  - Che integrino prodotti
  - Che presentino moduli sviluppati “ad hoc”
- Nessun sistema è sicuro a priori
  - Errori di integrazione
  - Errori di sviluppo
  - Bug non noti (origine dei cosiddetti “exploit 0-day”)
- Le attività di “Ethical Hacking” sono lo strumento più efficace per comprendere lo stato di sicurezza dei sistemi informativi
  - Cercando gli errori
  - Fornendo le opportune contromisure per risolverli

Processo di Sicurezza

- Permette ad un'organizzazione di migliorare il proprio livello di sicurezza
- Le attività di Ethical Hacking coprono le fasi di
  - *Plan* (applicare in progettazione le best practices)
  - *Do* (sviluppo/implementazione in sicurezza)
  - *Check* (testing)
  - *Act* (contromisure)
- Il processo di sicurezza si deve occupare di tutte le fasi
  - ISO 27001



Thank you!

BUSINESS e
drive your e-success