Ipotesi di Collaborazione tra Business-e SpA con gli istituti di ricerca

> Business-e S.p.A.
Via L. Braille, 15
48010 Ravenna Italy
Tel. +39.0544.288711
Fax +39.0544.463481

> Sedi operative
Ravenna +39.0544.460413
Roma +39.06.515.9081
Massa +39.0585.790.002
Milano +39. 02.397.10.155

www.business-e.it

> Cap. Soc. int. ver. € 1.001.084,00
Reg. Imprese RA n° 02019960398
R.E.A. n° 164837
C.F.: 02019960398
P.IVA: IT02019960398
Società appartenente al Gruppo Itway
anche ai sensi dell'Art 2497 c.c

ISO 9001
BUREAU VERITAS
Certification
N° 214583
SEDE DI ROMA

Indice

1 Presentazione azienda

Business-e è una società di ingegneria che opera a 360° nei settori della sicurezza informatica, occupandosi di integrazione di prodotti ed architetture, sviluppo di software, tematiche di Ethical Hacking, formazione, certificazioni e standard di sicurezza.

2 Intro

Al fine di inziare a collaborare con i maggiori centri di ricerca in ambito informatico e per dare loro l'opportunità di guardare all'interno di una realtà consolidata come la nostra ma inserita in un settore d'eccellenza e d'avanguardia, abbiamo selezionato una serie di attività per noi estremamente interessanti, che diano modo a dei giovani volenterosi di approfondire durante il corso di studi alcune delle tematiche su cui si sta indirizzando il mercato della sicurezza informatica. Per i seguenti progetti, che possono concretizzarsi in tirocini o tesi di primo livello, od in tesi magistrali/specialistiche, a seconda del candidato e del livello di profondità che lui stesso vuole dare all'argomento, se necessario vengono indicati tecnologie e sistemi operativi di riferimento. Quando tale indicazione è assente, il contesto del progetto va deciso al momento dello start-up, tra l'azienda ed il candidato stesso.

3 Titoli dei progetti

3.1 Stato dell'arte e trend nel settore della Sicurezza Informatica

3.1.1 Obiettivo

Alfabetizzazione al panorama mondiale della sicurezza, tramite individuazione e categorizzazione delle principali tecniche/prodotti nel settore, individuazione di trend nella ricerca in ambito Security ovvero delle piattaforme che sono più oggetto di studio da parte dei ricercatori universitari e non. Individuazione dei siti dei principali gruppi di ricerca universitaria e di hacker, descrizione del loro contenuto, ed analisi delle differenze di approccio alla ricerca nel settore da parte dei

3.1.2 Precondizioni

Nessuna.

3.1.3 Descrizione del progetto

Per effettuare le ricerca delle principali tecniche, è necessario partire dagli articoli delle ultime conferenze/riviste più importanti del settore (Black Hat, CCC, Phrack). A partire da questo, ci si può muovere in ampiezza, analizzando i siti ed i precedenti lavori di tali gruppi. Il lavoro non può andare in profondità su tali argomenti, vista l'enorme mole di informazioni; inoltre, se necessario, deve prevedere delle modalità per correlare i lavori dei vari membri/gruppi, la partecipazione alle conferenze, etc …

3.1.4 Output Attesi

Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati, in particolare le macrocategorie di studio individuate, e dati statistici circa membri, gruppi, tematiche di studio, e piattaforme impattate. Sarà valutata la necessità di articolare le informazioni in un database.

3.1.5 Sistemi Operativi/Tecnologie

Nel caso si implementasse un DB, questo dovrebbe essere consultabile da qualunque laptop aziendale; viste le necessità aziendali, potrebbe essere impossibile utilizzare database orientati ai server come MySQL o postgres.

3.1.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello, che non preveda un output tecnico ma documentale; è quindi adatto a chi vuole cimentarsi in un complesso lavoro di analisi e correlazione.

3.2 Confronto fra tool di Assessment

3.2.1 Obiettivo

Provare un insieme di tool usati in attività di esame della sicurezza di un contesto (generalmente chiamate attività di Assessment o attività di Ethical Hacking).

3.2.2 Precondizioni

Competenze di programmazione Web Application in Java, competenze nell'implementazione di database.

3.2.3 Descrizione del progetto

Creare un semplice ambiente di test che presenti delle vulnerabilità, la cui realizzazione è supportata da un tester che può enucleare le principali necessità; l'ambiente sarà una web application Java con un database a supporto. È a discrezione del candidato se utilizzare framework come Struts, Hibernate, Terracotta, etc… ovviamente a puro scopo esemplificativo. Testing di un numero limitato di tool di assessment (tool per Penetration Test applicativi) per stimarne l'efficacia (numero di vulnerabilità trovate) il numero di falsi positivi individuati ed eventualmente l'impatto sulle prestazioni del server.

3.2.4 Output Attesi

Documentazione (generalmente la tesi prodotta e la documentazione necessaria per replicare l'ambiente di test su altre macchine) che illustri i risultati dei confronti, e descriva le caratteristiche dell'ambiente di test. Può essere valutata la necessità di articolare le informazioni in un database.

3.2.5 Sistemi Operativi/Tecnologie

Nel caso si implementasse un database per la raccolta dati, questo dovrebbe essere consultabile da qualunque laptop aziendale; viste le necessità aziendali, potrebbe essere impossibile utilizzare database orientati ai server come MySQL o postgres. Per quel che riguarda l'ambiente di test, andrà sviluppato in Windows/Linux + Mysql/Postgres + Tomcat/JBoss.

3.2.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.3 Exploiting: a Case Study

3.3.1 Obiettivo

Effettuare uno studio su una vulnerabilità nota, ed esaminare i passi necessari, le metodologie, etc … per scrivere un exploit per la problematica.

3.3.2 Precondizioni

Buona conoscenza del linguaggio di programmazione in cui è scritto il software oggetto dello studio, e buona conoscenza del sistema operativo, oltre che una minima conoscenza (da approfondire sul campo) del linguaggio macchina dell'architettura.

3.3.3 Descrizione del progetto

A discrezione del candidato, il progetto può articolarsi in due varianti: esame di una vulnerabilità di cui esista già l'exploit, o esame di una vulnerabilità di cui non sia presente un exploit noto. L'azienda, per la natura delle sue attività, è più interessata a vulnerabilità che interessino software closed, ma se in fase di startup dovesse emergere un effort elevato per attività di questo tipo, si potrebbe anche sfruttare una vulnerabilità presente su un software open. Per quanto possibile, si richiede lo sfruttamento di vulnerabilità di tipo remote o local buffer overflow. Per quanto possibile, i software su cui operare debbono essere aggiornati a release di massimo 2 anni fa.

3.3.4 Output Attesi

Documentazione (generalmente la tesi prodotta e la documentazione necessaria a corredo dell'exploit) che illustri in dettaglio la metodologia seguita per l'individuazione e lo sfruttamento della vulnerabilità, il sorgente dell'exploit, ed eventuale altro materiale a corredo.

3.3.5 Sistemi Operativi/Tecnologie

A scelta fra Windows/Linux/*BSD/Solaris.

3.3.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello, se il candidato sceglie di analizzare una vulnerabilità per cui esiste già l'exploit; alternativamente, le tempistiche possono allungarsi, ed il lavoro può anche essere inquadrato nell'ambito di una tesi magistrale.

3.4 Grafi di Attacco

3.4.1 Obiettivo

Analisi degli studi attualmente in corso circa la modellazione dell'azione di un hacker, generalmente tradotti in grafi di attacco.

3.4.2 Precondizioni

Nessuna.

3.4.3 Descrizione del progetto

A partire dall'esame della bibliografia accademica circa l'argomento, categorizzare ed analizzare i singoli lavori proposti nell'ambito delle necessità operative che si riscontrano durante un Penetration Test od a seguito dello stesso (queste informazioni saranno spiegate da tester professionisti). Testare eventuali proof of concept/tool free rilasciati, e sviluppare osservazioni e considerazioni su come poter integrare le necessità di un tester con i lavori già presenti.

3.4.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio i risultati conseguiti, ed eventuali guide per far funzionare i tool provati.

3.4.5 Sistemi Operativi/Tecnologie

Nessuna in particolare.

3.4.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.5 Java Object Injection/Buffer Overflow

3.5.1 Obiettivo

Esaminare lo stato dell'arte delle tecniche di sfruttamento delle vulnerabilità possibili in ambienti basati su Java per poi eseguire applicazioni pratiche su contesti il più realistici possibile.

3.5.2 Precondizioni

Conoscenza pratica di Java ad un medio livello.

3.5.3 Descrizione del progetto

Il progetto può essere diviso in macrofasi sufficientemente indipendenti.
Fase 1: Esame/Organizzazione dello stato dell'arte e delle tecniche esistenti, differenze con le tecniche usuali per linguaggi compilati direttamente in linguaggio macchina (restringendosi ai soli C e C++), esecuzione/testing di eventuali "proof of concept" o exploit noti e reverse engineering degli stessi, strutturazione ed organizzazione della successiva fase, strettamente dipendente da questa.
Fase 2: A partire dalla fase precedente, ricerca (a partire dal codice sorgente) di vulnerabilità e sfruttamento delle stesse su un prodotto Open e Free potenzialmente usato in un contesto reale (come Tomcat). Eventualmente il progetto può anche evolversi in un framework (da descrivere, non da implementare) di studio, progettazione e realizzazione di exploit per Java.
Fase 3: Esame delle fasi 1 e 2 a fronte della differente implementazione delle Java Virtual Machine per i vari kernel lato server (BSD, HP-UX, Solaris….). Questa fase può anche essere indipendente dalle precedenti, ma risulta sicuramente molto ottimizzata se effettuata in seguito a studi che hanno già enucleato risultati od informazioni interessanti.
Va sottolineato come questo progetto possa risultare essere un progetto teso a nuove ricerche nel settore, in quanto è nostra opinione che le tecniche e gli studi nel settore generico del Java Exploiting non abbracciano questa tematica.

3.5.4 Output Attesi

È bene ricordare che gli output possono subire variazioni dipendenti dallo studio effettuato dal candidato stesso; il progetto, comunque, distinguendo nelle differenti fasi vuole ottenere.
Fase 1: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati, sia teorici che pratici, delle ricerche effettuate.
Fase 2: Documentazione che illustri in dettaglio la metodologia seguita per l'individuazione e lo sfruttamento della vulnerabilità, il sorgente dell'exploit, ed eventuale altro materiale.
Fase 3: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati, sia teorici che pratici, delle ricerche effettuate, e, se è stata intrapresa la strada dell'exploiting di una vulnerabilità, la metodologia seguita per l'individuazione e lo sfruttamento della vulnerabilità, il sorgente dell'exploit, ed eventuale altro materiale.

3.5.5 Sistemi Operativi/Tecnologie

Fase 1: Linux e Windows.
Fase 2: Linux e Windows.
Fase 3: uno o più a scelta fra tutti i sistemi operativi presenti in server utilizzati in ambiti di produzione, come HP-UX, Sun Solaris, IBM AIX.

3.5.6 Tempi

Fase 1: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

Fase 2: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello; se si vuole arrivare al framework, probabilmente le tempistiche e lo spessore teorico del lavoro aumenterebbero decisamente, e quindi in questo secondo caso le tempistiche sono più vicine ad una tesi magistrale.
Fase 3: Il progetto è indicato per una tesi magistrale, ma può anche essere trattato in modalità meno approfondita, ed essere quindi adatto anche a tirocini di primo livello.

3.6 Sviluppo di software per immagazzinamento, correlazione ed analisi di risultati di attività di Penetration Test

3.6.1 Obiettivo

A seguito dell'effettuazione di un numero cospicuo di test, potrebbe essere d'aiuto immagazzinarne i risultati per analisi e considerazioni successive. L'obiettivo del progetto è creare un sistema volto a questo, con anche un'interfaccia web/GUI che permetta solo agli autorizzati di visionarli ed eventualmente inserirne di nuovi.

3.6.2 Precondizioni

Conoscenza sulla programmazione in ambito Java e sulla progettazione/implementazione di database.

3.6.3 Descrizione del progetto Fasi

A partire dall'analisi dei dati in input, progettazione ed implementazione del DB, creazione della logica di business e creazione dell'interfaccia grafica. Va considerato il fatto che gli input per il programma che inserirà i dati nel DB saranno documenti word, potenzialmente di formati differenti.

3.6.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri i passi di creazione del software oltre che le considerazioni fatte sulla natura dei dati. Codice sorgente del software prodotto, ed altri strumenti tecnici (come script del database, guide di configurazione, etc …).

3.6.5 Sistemi Operativi/Tecnologie

Nessuna in particolare, la scelta va valutata a partire dalle esigenze.

3.6.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.7 SOA Security

3.7.1 Obiettivo

Alfabetizzazione su questo settore della security informatica, tramite l'analisi e le considerazioni sullo stato dell'arte circa la sicurezza in ambito SOA (Services Oriented Architecture). Sviluppare osservazioni e considerazioni sui contributi mancanti nel settore e quelli maggiormente presenti, per permettere uno sviluppo successivo del progetto.

3.7.2 Precondizioni

Nessuna.

3.7.3 Descrizione del progetto

A partire dall'esame della bibliografia accademica e non circa l'argomento, categorizzare ed analizzare i singoli lavori proposti nell'ambito delle necessità operative che si riscontrano durante un Penetration Test od a seguito dello stesso (queste informazioni saranno fornite da tester professionisti). Testare i principali tool free, ove possibile. Grazie a questa ricerca, è possibile acquisire una conoscenza di buon livello su un paradigma di programmazione che spesso il contesto accademico può solo accennare, anche nelle lauree specialistiche.

3.7.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio i risultati conseguiti, e gli sviluppi pensati per il proseguo del progetto.

3.7.5 Sistemi Operativi/Tecnologie

Nessuna in particolare.

3.7.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.8 Sviluppo di un SOA Proxy

3.8.1 Obiettivo

Durante le attività di Penetration Test, vengono spesso utilizzati degli strumenti che si pongono a metà fra il tester e l'application server oggetto dell'analisi, che rendono possibile la realizzazione di molteplici scenari di attacco soltanto su applicazioni Web, e quindi soltanto per i protocolli HTTP/HTTPs. L'obiettivo del progetto è creare uno strumento analogo (estendibile) per protocolli proprietari o commerciali in ambito SOA (Weblogic t3, Microsoft XML, Oracle RMI, etc…).

3.8.2 Precondizioni

Buona capacità di programmazione in Java, propensione al reverse engineering di architetture anche complesse.

3.8.3 Descrizione del progetto

Dopo una prima fase di analisi delle principali tecnologie SOA, è necessario effettuare il reverse engineering di un protocollo commerciale, od alternativamente studiare il codice sorgente di un protocollo open. A seguito di ciò, si procederà all'implementazione di un software (preferibilmente con GUI Java Swing) che agisca da proxy locale sulla macchina, e permetta di intercettare tutti i pacchetti di quel particolare protocollo, visualizzarne le informazioni ed eventualmente alterarli. Il progetto è estendibile a piacimento, aggiungendo features circa il controllo/analisi del traffico, o plugin su altri protocolli. Deve essere pensato in un'ottica di estendibilità futura.

3.8.4 Output Attesi

Il software, le guide e tutta la documentazione di dettaglio necessaria ad un altro sviluppatore per ampliare il progetto in qualunque senso.

3.8.5 Sistemi Operativi/Tecnologie

Precedenti esperienze di programmazione in Java.

3.8.6 Tempi

Il progetto, vista anche la necessità di uno studio del modello di sviluppo e di una particolare tecnologia SOA, è sicuramente più vicino alle tempistiche di una tesi specialistica che non a quello di un tirocinio e tesi di primo livello.

3.9 Analisi ed eventuale applicazione delle principali tecniche di mascheramento della memoria

3.9.1 Obiettivo

Comprendere lo stato dell'arte sulle principali tecniche legate al mascheramento dei dati in memoria. Con l'aumentare della verticalizzazione delle tecniche di attacco, le realtà enterprise si preoccupano non solo del transito delle informazioni fra due server, o della memorizzazione sicura su storage permanente (hard disk od altri dispositivi magnetici/ottici), ma anche delle informazioni residenti in RAM e della possibilità che esse siano accedute da amministratori o da utenti in grado di ottenere i privilegi amministrativi. È quindi estremamente interessante comprendere se e come sia possibile rendere difficile od addirittura impossibile, da parte di un amministratore, la comprensione di dati particolarmente critici.

3.9.2 Precondizioni

Conoscenza del kernel Linux o del kernel di un sistema operativo Unix server side (HP-UX, Sun Solaris, AIX).

3.9.3 Descrizione del progetto

Raccogliere tutte le informazioni e catalogare la bibliografia e le tecniche esistenti circa l'argomento. In seguito, testare eventuali tool free messi a disposizione, se presenti; se non presenti, potrebbe essere necessario analizzare ed implementare un algoritmo ritenuto interessante, già noto in bibliografia. Comprendere il livello di interazione fra il sistema (kernel, API proprietarie, livello di rete se pertinente, etc…) ed le specifiche tecniche per poter avere una panoramica delle soluzioni e degli ambiti/possibilità di applicazione.

3.9.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio le analisi effettuate, i confronti, e le informazioni ricavate. Se viene implementato un algoritmo, il codice sorgente e le guide/informazioni necessarie ad eventuali altri sviluppatori per esaminarlo e modificarlo.

3.9.5 Sistemi Operativi/Tecnologie

Probabilmente vista l'alta interazione fra software di questo tipo e sistema operativo, l'intero progetto dovrebbe essere svolto su sistemi Linux; importante enucleare concetti/tecniche generalizzabili a tutto il mondo Unix.

3.9.6 Tempi

Il progetto è un progetto di analisi a basso livello, che necessita di una conoscenza di dettagli del sistema operativo e dei processi in esso presenti. In relazione al livello di approfondimento, può essere svolto come un tirocinio e tesi di primo livello, o come tesi magistrale; nell'ultimo caso è necessario implementare un algoritmo e studiarne eventuali varianti, oltre che le interazioni con l'ambiente di esecuzione e le possibili problematiche legate ad aspetti di concorrenza di processi, interazioni con database, accessi illeciti, debolezze intrinseche e di implementazione, etc …

3.10 Studio dei meccanismi di Wrapping di Oracle

3.10.1 Obiettivo

Analisi di una specifica funzionalità di Oracle che riguarda il mascheramento/cifratura di procedure di database. Principalmente, tale analisi è volta ad inserire il candidato nel processo di reverse engineering di un prodotto di cui non si possiede il codice, alla comprensione delle tecniche usate in tal senso, ed allo studio della documentazione a corredo. L'obiettivo di comprendere la procedura di offuscamento/cifratura usata da Oracle in questo contesto può essere raggiunto o meno; è importante per il candidato immedesimarsi nel ruolo di ricercatore di sicurezza, ovvero nel ruolo di un security analyst che possa provare varie strade, più o meno fruttuose, per arrivare all'analisi di un oggetto a lui totalmente ignoto.

3.10.2 Precondizioni

Conoscenza generica dei DBMS, e di SQL, e dei concetti di base della crittografia; propensione al reverse engineering.

3.10.3 Descrizione del progetto

La modalità con la quale alcune versioni di Oracle offuscano le procedure presenti nel database è tuttora compresa poco dal mondo della ricerca in ambito security. Sono presenti spiegazioni di alto livello che la illustrano, ma, ad esempio, non è presente un software che riesca ad effettuare l'un-wrapping di una procedura (ovvero decodificarla/deoffuscarla). Il candidato dovrebbe, autonomamente, ottenere informazioni sullo stato dell'arte in tal senso, e procedere a simulare il comportamento di un vero ricercatore in ambito security, che voglia ottenere un risultato pratico, ovvero comprendere il meccanismo e scrivere un software in grado di effettuare l'un-wrapping. Questa situazione gli permetterà di approfondire le principali tecniche usate in questi casi (debug real time, analisi e modifica del contenuto della memoria, decompilazione di parti di codice, analisi crittografiche). L'obiettivo finale, ovvero tale software, non è detto che si raggiunga o sia raggiungibile.

3.10.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio lo stato dell'arte, le tecniche affrontate e tentate, il loro esito anche se negativo, la giustificazione delle motivazioni di ciascun tentativo e dell'esito; se presente, del codice sviluppato, oltre che guide ed informazioni da presentare ad uno sviluppatore per proseguire nel lavoro.

3.10.5 Sistemi Operativi/Tecnologie

Linux/Windows; probabilmente è più indicato il secondo sistema in quanto i tool per applicare alcune tecniche sono presenti più per Windows.

3.10.6 Tempi

Se il lavoro tratterà soltanto l'analisi dello stato dell'arte e della comprensione delle tecniche da utilizzare nel contesto, può essere svolto nei tempi e modalità di una tesi e tirocinio di primo livello. Se oltre alla parte teorica, si procede ad una parte pratica di implementazione, allora la complessità e l'eterogeneità del lavoro richiesto rende necessario un approccio più completo tipico, generalmente, di studenti che stanno conseguendo una laurea magistrale.

3.11 Sicurezza in Ajax: stato dell'arte e case study

3.11.1 Obiettivo

Analisi della tecnologia Ajax, principali usi ed applicazioni. In seguito, analisi delle debolezze intrinseche, se presenti, o dei problemi di sicurezza, anche calati in contesti potenzialmente realistici. Applicazione dei risultati in un caso di test, appositamente creato, e discussione dell'output.

3.11.2 Precondizioni

Conoscenza base di Javascript e di Linux.

3.11.3 Descrizione del progetto

Il candidato deve in primo luogo effettuare una ricerca sulla tecnologia Ajax, su come è integrata nei principali application server commerciali e free, e sul come e se risolve le problematiche di sicurezza insite nel fatto che Javascript, agendo lato client, lascia all'utente la possibilità di agire su particolari elementi che possono arrecare danni dal punto di vista della sicurezza. A seguito dell'analisi, il candidato deve implementare un semplice ambiente di test, che presenti una vulnerabilità legata all'integrazione di Ajax con altre componenti, od ad un suo errato uso, e discutere i risultati calando tale realtà di test secondo l'analisi precedentemente effettuata e le esperienze di uso del prodotto in contesti aziendali.

3.11.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio i risultati conseguiti, quindi sia la fase di analisi della tecnologia, delle problematiche eventualmente esistenti e note, del case study e delle considerazioni scaturite a seguito dell'esperimento.

3.11.5 Sistemi Operativi/Tecnologie

Conoscenza delle piattaforme Linux/Windows.

3.11.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.12 Sicurezza di ambienti Mainframe e tecniche/software per emularli/virtualizzarli

3.12.1 Obiettivo

Nonostante l'obsolescenza, le realtà enterprise non sempre sostituiscono i sistemi Mainframe, basati su un'organizzazione fisica e logica totalmente differente rispetto ai classici server Unix/Linux/Widnows; per esempio, nei contesti Mainframe non esiste il concetto di filesystem o processo, così come lo si conosce dall'esperienza su Linux/Windows. Il costo di tali apparati non permette di crearne un laboratorio, ma vi sono tentativi atti virtualizzare tali ambienti. L'obiettivo del progetto è esaminare lo stato dell'arte, le tecniche e gli eventuali prodotti presenti, oltre che, se possibile, tentare di percorrere alcune strade di virtualizzazione.

3.12.2 Precondizioni

Nessuna.

3.12.3 Descrizione del progetto

Il candidato deve effettuare uno studio sullo stato dell'arte con l'obiettivo di enucleare le modalità con le quali i ricercatori in ambito security possono testare o anche solo studiare ambienti mainframe. A partire da questo, è necessario compiere uno studio, potenzialmente sovrapposto al primo, sulle tecniche/piattaforme di virtualizzazione/emulazione di ambienti mainframe, e la loro potenziale usabilità. In ultimo, dovrebbe analizzare in dettaglio una tecnica, ed eventualmente approntare un laboratorio per implementarla e quindi virtualizzare un ambiente Mainframe.

3.12.4 Output Attesi

Documentazione (generalmente la tesi prodotta) che illustri in dettaglio lo stato dell'arte, le tecniche presenti, e tutti i passi delle azioni effettuate (anche tramite la produzione di guide, how to, schemi di percorso logico, etc…).

3.12.5 Sistemi Operativi/Tecnologie

Nessuna in particolare.

3.12.6 Tempi

La profondità e la complessità dello studio è probabilmente più adatta ad un candidato che ha terminato il percorso della laurea magistrale.

3.13 Metodologie di sviluppo sicuro del codice

3.13.1 Obiettivo

Raccolta, analisi e confronto delle metodologie principali per lo sviluppo del codice che pongano particolare attenzione ai controlli di sicurezza in fase di developing. A seguito dell'enumerazione e della spiegazione di ciascuna metodologia, è importante definire i potenziali contesti applicativi ed i costi/benefici delle stesse.

3.13.2 Precondizioni

Conoscenza di una metodologia di sviluppo software, e preferibilmente sua applicazione in contesti di sviluppo anche a fini di progetti personali/accademici.

3.13.3 Descrizione del progetto

La ricerca bibliografica di tutte le principali metodologie di sviluppo del codice che mettano in luce gli aspetti di prevenzione relativi alla sicurezza, ovvero includere criteri, linee guida, baseline, osservazioni, progettazioni orientate alla minimizzazione del rate di vulnerabilità nel codice sviluppato, od alla verifica dello stesso. A partire dal processo di creazione del codice, inquadrare la questione in un contesto aziendale, in cui tale processo è solo un elemento del ciclo di sviluppo; in questo modo è possibile procedere alla seconda fase del progetto, ovvero la definizione dei possibili campi d'applicazione di ciascuna metodologia (intesi come i contesti aziendali, l'organizzazione necessaria, il livello di competenza sulla sicurezza medio, etc…), che fanno da prologo alla conclusione del progetto, ovvero un'analisi costi-benefici di ciascun approccio.

3.13.4 Output Attesi

La documentazione di dettaglio dei risultati ottenuti, oltre ai criteri seguiti nell'analisi, sia per la definizione di contesti applicativi sia per l'analisi costi-benefici.

3.13.5 Sistemi Operativi/Tecnologie

Nessuna in particolare.

3.13.6 Tempi

Le tempistiche di svolgimento del progetto sono quelle tipiche di una tesi di primo livello.

3.14 Integrazioni e collaborazione al progetto OWASP - Passwd (Prediction of applications and systems security Within development)

3.14.1 Obiettivo

Creare un modello che aiuti nella predizione e al monitoraggio della sicurezza di un'applicazione.

3.14.2 Precondizioni

Conoscenza delle più comuni tecnologie informatiche e del loro impiego nelle enterprise, conoscenze di sicurezza nelle web-app.

3.14.3 Descrizione del progetto

Il progetto ha come scopo quello di creare un farmework di riferimento per l'analisi dei rischi che lavori in maniera predittiva al fine di dare supporto sin dalle prime fasi del SDLC.
Il fine è quello di supportare uno sviluppo delle web application che possa risolvere sin dalla fase di scrittura del codice applicativo le principali problematiche di sicurezza normalmente presenti sulle tecnologie o caratteristiche comuni alle altre web-app presenti sul mercato.
Fase 1: Il progetto partirà dalla catalogazione tecnologie maggiormente diffuse per lo sviluppo e l'implementazione di web-application ed individuazione dei possibili scenari futuri.
Fase 2: Dopo questa fase si procede all'analisi delle principali metriche adottate per misurare la sicurezza dei programmi e dei sistemi.
Fase 3: Non appena il panorama sarà completo verrà stabilito quale metodologia possa essere la migliore da utilizzare in questo caso per adottarne o adattarne una (sia qualitativa che quantitativa).
Fase 4: In presenza della metodologia di misura, si procederà alla catalogazione delle vulnerabilità normalmente rilevate dai security experts di OWASP per creare un database, il più esteso possibile e facilmente consultabile, che permetta di metterle in relazione tra di loro in termini di rischi associati.
Fase 5: Poi verranno creati dei modelli di riferimento su base tecnologica che permettano di poter valutare quale sia il corretto Key Risk Indicator che in ognuna delle fasi del SDLC permette di passare alla successiva. Alla fine il progetto promuoverà uno standard di riferimento, sulla falsariga degli altri OWASP standards che abbia l'obiettivo di mappare tutti i livelli di sicurezza legati alle web-app all'interno di un'organizzazione che abbia come core business una infrastruttura IT altamente evoluta.
Fase 6: A questo punto sulla base delle considerazioni e metodologie precedenti, progettare e realizzare un'applicazione di test che riesca a simulare una situazione reale dalla quale sia possibile comprendere il rapporto costi/benefici a seconda delle tecnologie adottate. La finalità è quella di individuare la variazione del rischio da accettare nelle varie fasi del SSDLC.
Fase 7: Estendere il test plant anche alle eventuali interazioni con altre web-app per analizzare uno scenario complesso.
Fase 8: Infine agganciare al test plant un cruscotto che riporti tutti gli indicatori predittivi dei rischi sui quali si potrebbe incorrere qualora una particolare web-app, che eroghi dei servizi specifici, qualora venga inserita all'interno di un contesto complesso, possa avere.

3.14.4 Output Attesi

Fase 1: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati delle ricerche effettuate.

Fase 2: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri l'analisi effettuata mettendo a confronto le varie metriche.
Fase 3: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri la metodologia individuata e le motivazioni che ne hanno porto alla stesura.
Fase 4: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri il sistema realizzato. A corredo dovrà essere fornito un software che implementi quanto richiesto e dichiarato nella tesi.
Fase 5: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i modelli individuati e le motivazioni che ne hanno porto alla stesura.
Fase 6: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri le motivazioni per le quali si è scelto di progettare una determinata applicazione piuttosto che un'altra e la relativa analisi sulla variazione del rischio nelle fasi del SSDLC. A corredo dovrà essere fornito un software che implementi quanto richiesto e dichiarato nella tesi.
Fase 7: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri le motivazioni per le quali si è scelto di progettare una determinata applicazione piuttosto che un'altra e la relativa analisi sulla variazione del rischio nelle fasi del SSDLC. A corredo dovrà essere fornito un software che implementi quanto richiesto e dichiarato nella tesi.
Fase 8: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri il sistema realizzato. A corredo dovrà essere fornito un software che implementi quanto richiesto e dichiarato nella tesi.

3.14.5 Sistemi Operativi/Tecnologie

Fase 1: Nessuna.
Fase 2: Nessuna.
Fase 3: Nessuna.
Fase 4: Nel caso si implementasse un database, questo dovrebbe essere consultabile da qualunque laptop aziendale; viste le necessità aziendali, potrebbe essere impossibile utilizzare database orientati ai server come MySQL o postgres.
Fase 5: Nessuna.
Fase 6: Java EE, Tomcat, JBoss, MySql, PostGres.
Fase 7: Java EE, Tomcat, JBoss, MySql, PostGres.
Fase 8: Nel caso si implementasse un database, questo dovrebbe essere consultabile da qualunque laptop aziendale; viste le necessità aziendali, potrebbe essere impossibile utilizzare database orientati ai server come MySQL o postgres.

3.14.6 Tempi

Fase 1: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.
Fase 2: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.
Fase 3: Il progetto è indicato per una tesi magistrale.
Fase 4: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.
Fase 5: Il progetto è indicato per una tesi magistrale.
Fase 6: Il progetto è indicato per una tesi magistrale.
Fase 7: Il progetto è indicato per una tesi magistrale.
Fase 8: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.15 Sviluppo di software per applicare una metodologia di Analisi del Rischio

3.15.1 Obiettivo

In un contesto aziendale, è possibile parlare di sicurezza soltanto quantificando il danno legato al fatto che un determinato dato subisca violazioni rispetto alla sua riservatezza, integrità o disponibilità. Tale quantificazione passa per una attività denominata analisi del rischio. Tale attività può essere svolta seguendo varie metodologie, la maggior parte delle quali è talmente complessa da risultare inapplicabile in contesti realistici in cui non è possibile avere in modo completo tutte le informazioni necessarie in input alla metodologia stessa. Obiettivo di questo progetto è sviluppare un software che consenta l'applicazione di una metodologia che lasci libero l'utilizzatore di adattare i dati passati alla realtà di interesse; altro obiettivo è il confronto fra le metodologie esistenti.

3.15.2 Precondizioni

Conoscenza di Java.

3.15.3 Descrizione del progetto

Analisi delle principali metodologie per l'analisi del rischio, ed implementazione di una di queste, a seguito di una corposa fase di raccolta ed analisi dei requisiti. Uso attento di pattern adatti a gestire in modo intelligente il particolare requisito di permettere all'utente di adattare la metodologia al contesto in cui si trovi ad applicarla, senza però snaturarla. Gestione di tutto il processo di progettazione in maniera autonoma; se il candidato è interessato, sviluppo guidato dal testing (unitario o meno).

3.15.4 Output Attesi

Documentazione prodotta nell'analisi e progettazione del codice (casi d'uso, contratti, diagrammi UML, diagrammi di interazione, diagrammi delle classi, etc …), sorgente sviluppato, ed altra eventuale documentazione a corredo, come commenti sulle scelte tecnologiche/architetturali adottate.

3.15.5 Sistemi Operativi/Tecnologie

Nessuna in particolare.

3.15.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.16 Emulazione/virtualizzazione di sistemi operativi/architetture server

3.16.1 Obiettivo

Negli ultimi anni molte aziende stanno centralizzando i propri sistemi su nuovi data center che consentono la virtualizzazione di diversi sistemi operati. Il costo di questi data center di nuova generazione risulta inaccessibile al solo fine di studio, pertanto sorge la necessità di individuare sistemi software che consentano l'emulazione e/o la virtualizzazione di questi ambienti.

3.16.2 Precondizioni

Buona conoscenza dei linguaggi c/c++, conoscenza minima dell'assembly x86 e di altre architetture.

3.16.3 Descrizione del progetto

Il progetto può essere articolato in diverse fasi.
Fase 1: Esamine e organizzare lo stato dell'arte e le tecnologie esistenti al fine di individuare quali software consentano di emulare e virtualizzare differenti sistemi operativi e architetture hardware quali x86, x86_64, ARM, SPARC, SPARC64, PowerPC, PowerPC64, MIPS, MIPS64, m68k (Coldfire), SH-4, Alpha e CRIS. In questa fase devono essere segnalate e confrontate le varie potenzialità di emu./virt. dei singoli software.
Fase 2: A partire dalla fase precedente si dovrà ricercare una possibile soluzione software sulla quale sia possibile implementare l'emulazione dell'hardware, nello specifico la CPU, delle specifiche architetture.
Fase 3: Sulla base della fase precedente implementare una delle possibili architetture hardware sul software di emulazione.

3.16.4 Output Attesi

Fase 1: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati delle ricerche effettuate e dei confronti prodotti.
Fase 2: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri le caratteristiche del software che dovrà consentire l'emulazione. A corredo della documentazione dovrà essere presentato il software di base sul quale dovrà essere possibile implementare altre architetture.
Fase 3: Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri la modalità e le difficoltà incontrate nell'implementare correttamente l'emulazione del'architettura sul software. A corredo della documentazione dovrà essere presentata l'implementazione prodotta con vari test che ne garantiscano l'affidabilità.

3.16.5 Sistemi Operativi/Tecnologie

Fase 1: Linux/Windows, architettura x86/x86_64.
Fase 2: Linux/Windows, architettura x86/x86_64.
Fase 3: Uno o più sistemi operativi/architetture a scelta fra quelli utilizzati negli ambienti di produzione come HP-UX, Sun Solaris, IBM AIX, SPARC e PowerPC.

3.16.6 Tempi

Fase 1: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.
Fase 2: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello; se si vuole arrivare alla produzione del software per l'emulazione con incorporata una specifica architettura, probabilmente le tempistiche e lo spessore teorico del lavoro aumenterebbero decisamente, e quindi in questo secondo caso le tempistiche sono più vicine ad una tesi magistrale.
Fase 3: Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello.

3.17 Sicurezza di ambienti SCADA e tecniche/software per emularli/virtualizzarli

3.17.1 Obiettivo

I sistemi di controllo di supervisione e acquisizione dati distribuiti sono una realtà diffusa oltre che in LAN anche in rete geografica, dove i dati sono maggiormente esposti a manipolazione o visione da soggetti terzi. L'obiettivo del progetto è esaminare lo stato dell'arte, le tecniche e gli eventuali prodotti presenti, oltre che, se possibile, tentare di percorrere alcune strade di virtualizzazione.

3.17.2 Precondizioni

Nessuna.

3.17.3 Descrizione del progetto

L'obbiettivo è quello di individuare la modalità di funzionamento dei sistemi SCADA più noti. La fase successiva è quella di emulare/virtualizzare questi sistemi in modo da ottenere un ambiente in vitro su cui effettuare verifiche per verificarne la sicurezza. Infine identificare quali sono le possibili vulnerabilità di questi sistemi e definirne una soluzione.

3.17.4 Output Attesi

Documentazione (generalmente la tesi prodotta ma anche eventuale altro materiale a corredo) che illustri i risultati delle ricerche effettuati, dando particolare attenzione a quelle che sono le problematiche di sicurezza e le relative soluzioni adottabili. A corredo della tesi dovrebbe essere fornito un ambiente di test, possibilmente realizzato con tecniche di virtuale.

3.17.5 Sistemi Operativi/Tecnologie

Linux/Windows, ambienti di virtualizzazione ed emulazione.

3.17.6 Tempi

Il progetto può essere eseguito con le modalità e le tempistiche di un tirocinio e tesi di primo livello. Se si intende implementare un laboratorio virtuali basato sui sistemi di virtualizzazione/emulazione le tempistiche e lo spessore teorico del lavoro aumenterebbero decisamente, e quindi in questo secondo caso le tempistiche sono più vicine ad una tesi magistrale.