František Koláček
xkolac12

Souhrn dosavadní práce na BP

Automatizovaná detekce zranitelnosti webových aplikací

Téma své bakalářské práce jsem po schválení vedoucím za společnost RedHat Czech s. r. o. (dále jen RedHat), Janem Rusnačkem a prvotní schůzce, jež se konala dne 1. 10. 2014, začal vypracovávat bakalářskou práci na téma Automatizovaná detekce zranitelnosti webových aplikací.

V úvodu mé práce jsem se zaměřil na teoretický úvod do problematiky a vysvětlení několika pojmů. Dále jsem zmínil aktuálně používané způsoby a možnosti automatizovaného testování bezpečnosti webových aplikací. Webové scannery provádějí black-box testing a tedy neprovádí žádnou analýzu zdrojového kódu. Tyto webové scannery používají pro testování pouze webového rozhraní. Tedy stejného rozhraní, jaké používá legitimní návštěvník při návštěvě či používání této aplikace. V další části práce jsem zmínil projekt OWASP a OWASP foundation, jež je neziskovou organizací, která tento projekt zastřešuje. OWASP také každý rok prezentuje aktualizovaný dokument, jehož obsahem je seznam nejvíce kritických a nejvíce zneužívaných bezpečnostních zranitelností za daný rok. Seznam zranitelnosti za rok 2013 je zde také zmíněn.

V další kapitole se podrobněji zaměřuji na analýzu jednotlivých bezpečnostních scannerů, jež jsou vyvíjené jako opensource a tedy je možné analyzovat jejich zdrojový kód. Pro porovnání jsem vybral tyto opensource scannery: Arachni, w3af, Wapiti, Zap a Revok. Poslední zmiňovaný byl nedávno uvolněn jako opensource. Tento scanner byl dříve vyvíjen jako interní nástroj pro testování bezpečnostni webových aplikací firmy RedHat, avšak byl zveřejněn a jeho zdrojové kódy jsou dostupné prostřednictvím projektu Github.com . Každý z bezpečnostních scannerů byl v této kapitole krátce představen a v závěru této kapitoly byly porovnané některé zajímavé vlastnosti jednotlivých scannerů. Výsledky jsou k nalezení v tabulce na konci kapitoly Analysis of existing solutions.

Následující kapitola shrnuje metodiku, jež byla použita pro vyhodnocení úspěšnosti jednotlivých scannerů. Pro testování byly použity dva různé nástroje. Prvním z nich byl WIVET, jež umožňuje testovat možnosti scannerů použít konkrétní vektory útoku pro webové aplikace. Každý z testovaných scannerů byl spuštěn s konkrétními testovacími moduly a bylo sledováno, kolik z vektorů útoku byl daný scanner schopen rozpoznat a využít. Pro testování byl využit WIVET ve verzi 4, která umožňuje sledovat průběh aktuálního testu a po jeho skončení je schopna v procentech vyjádřit počet nalezených / využitých vektorů útoku. Tato aplikace odlišuje jednotlivé testy pomocí speciální cookie ve webovém prohlížeči jménem PHPSESSID. Každý scanner tedy bylo nutné před spuštěním testů nakonfigurovat tak, aby používal pouze a jen jednu PHPSESSID, jinak by mohl být výsledek testu

zkreslen. Většina z testovaných scannerů umožňuje určitým způsobem obsah této cookies napevno stanovit, avšak některé z nich bylo nutné nakonfigurovat k použití intercept proxy. Jako intercept proxy, přes kterou se datový provoz přeposílal a byl upravován (fixní obsah PHPSESSID), byl použit nástroj Burp. Po získání všech výsledků z nástroje WIVET jsem pokračoval testováním pomocí nástroje WAVSEP, jež poskytuje seznam zranitelných stránek, jež mohou být využity pro testování úspěšnosti detekce jednotlivých útoků.

WAVSEP (ve verzi 1.5.) bohužel neobsahuje žádnou možnost vyhodnocení, tak jak tomu bylo například u nástrje WIVET, bylo tedy zapotřebí výsledky z logů jednotlivých scannerů vyhodnotit samostatně. WAVSEP obsahuje pouze zranitelné stránky a tedy jsem vytvořil sadů skriptů, jež mi pomohli s vyhodnocováním. Prvním krokem bylo vyextrahování všech zranitelných stránek (jejich URL) do lokální databáze zranitelností. Následně rozčlenění těchto zranitelností na kategorie (SQLi, XSS, LFI, RFI a ostatní). V další fázi jsem použil skripty pro vyhodnocení logů z testů jednotlivých scannerů. Tyto skripty hledaly zmínky ohledně nalezených zranitelností a následně je přidávaly do lokální databáze testů. Jakmile byly tyto informace obsažené v databázi, další skript pouze prošel seznam všech možných útoků a útoků, jež daný scanner objevil a spočítal z nich úspěšnost pro daný scanner a daný typ útoku.

V následující části práce byly zmíněné 4 skupiny testovaných zranitelností a to: SQL injection, Cross-Site Scripting, Local File Inclusion a Remote File Inclusion. Každá z těchto zranitelností byla krátce vysvětlena.

V kapitole Results byly představeny výsledky testovaných scannerů. Pro testování možnosti využití různých vektorů útoků byl využit nástroj WIVET a nejlépe si vedla nejnovější verze opensource scanneru Arachni. Nejhůře si naopak vedl scanner Revok, avšak to z důvodu chyby v modulu starajícího se o načítání obsahu stránky. Úspěšnost detekce jednotlivých skupin útoků byla testována pomocí nástroje WAVSEP. Nejlépe si vedly scannery Arachniu a ZAP, naopak nejhůře skončily scannery w3af a Revok.

V poslední (zatím) vypracované kapitole se zabývám návrhem nových a efektivnějších způsobů pro testování určitých zranitelností a jejich plánované implementace do scanneru Revok.

František Koláček
xkolac12

Harmonogram prací

- 18.2 2015 Nejzašší termín pro implementaci patche pro web crawler
- 28. 2. 2015 Opětovné otestování Revoku pomocí WIVET (a vyhodnocení)
- 1. 3. 2015 Pravidelný měsiční report
- 14. 3. 2015 Implementace Jenkins testing environment
- 1. 4. 2015 Pravidelný měsíční report
- 10. 4. 2015 Implementace LFI modulu pro Revok
- 1. 5. 2015 Finální kompletace (obsah CD, dokumentace, podpůrné skripty)
- 20. 5. 2015 Odevzdání BP práce