PENETRATION TESTING OF OPEN SOURCE SOFTWARE

Jakub Hrozek
Master Degree Programme, FIT BUT
E-mail: xhroze01@stud.fit.vutbr.cz

Supervised by: Aleš Smrčka
E-mail: smrcka@fit.vutbr.cz

ABSTRACT

The contribution of the work is the design of a framework based on tools designed for penetration and security testing. A special effort is put to the selection of tools supported by the framework and mainly to the criteria of the selection itself. The goal is to integrate the selected tools in a way that the test is transparent to the user.

1 ÚVOD

Penetračním testováním se rozumí proces, během kterého je simulován útok na reálný systém nebo systém jemu podobný za účelem zjištění jeho slabých míst. Příkladem mohou být neaplikované bezpečnostní záplaty, slabá hesla a podobně. Takový test nemůže zaručit, že systém je bezpečný, k tomu by bylo potřeba znát kompletní seznam zranitelností. Přesto může významnou měrou pomoci ke zlepšení bezpečnosti systému, protože posloupnost kroků testu je podobná té, kterou by použil útočník při reálném napadení sítě.

Existuje několik hledisek, podle kterých lze rozdělit penetrační testování. Například podle úrovně znalostí útočníka o cílovém systému, podle toho, zda útok začíná z vnějšku či uvnitř organizace nebo například podle dopadu na fungování cíle. Podrobnější popis lze nalézt například v [1].

Mnoho dostupných materiálů poskytuje rady jak provést penetrační test, často i s využitím určité metodologie (např. NIST-800-115[1]), která může pomoci vykonání testu do jisté míry formalizovat. Většina takových návodů (např. [2]) ale popisuje čistě manuální test, při kterém se pro každou část útoku využívají jiné aplikace. Výstupy je tedy třeba nejenom ručně interpretovat, ale v neposlední řadě také vhodné aplikace vybrat.

Tento příspěvek seznamuje čtenáře s návrhem systému pro provádění penetračních testů, který integruje aplikace pro jednotlivé fáze testu jako komponenty provázaného celku. Vyvíjený program si klade za cíl proces testování alespoň do jisté míry automatizovat a jednotným uživatelským rozhraním odstínit uživatele od podrobností tak, aby se mohl soustředit pouze na samotný test. Komponenty výsledného programu budou Open-Source aplikace, jedním z cílů je také framework poskytnout jako instalační balíček pro linuxovou distribuci Fedora.

[1] NIST 800-115: Technical Guide to Information Security Testing and Assessment.

2 VÝBĚR KOMPONENT VYVÍJENÉHO NÁSTROJE

Funkcionalita jednotlivých fází penetračního testu je poskytována programy, které tvoří komponenty vytvářeného programu. Nedílnou součástí návrhu je tedy stanovení kritérií, podle kterých jsem hodnotil dostupné aplikace.

2.1 KRITÉRIA VÝBĚRU

Samotná pravidla pro výběr nástrojů vycházení z doporučení pro provádění penetračních testů, které publikovala americká vládní agentura CERT[2]. Dále byla využita některá doporučení pro výběr nástrojů na provádění tradičních testů softwaru. Výsledná kritéria jsem rozdělil do dvou kategorií: (i) nutné podmínky, bez jejichž splnění nemůže být uvažovaná aplikace do kolekce zařazena; (ii) soubor doporučení, které je možné využít pro porovnání programů, sloužících k vykonání stejné fáze testu.

Mezi nutné podmínky patří správná funkctionalita, možnost automatizace a skriptování chodu aplikace a licence kompatibilní s podmínkami distribuce Fedora. Soubor doporučení pro porovnání podobných aplikací obsahuje například to, zda je program aktivně vyvíjen, nebo jestli je možné ho rozšířit pomocí zásuvných modulů. Neméně důležitou vlastností je také formát výstupu, který musi být vhodný pro strojové zpracování.

3 OBECNÉ FÁZE PENETRAČNÍHO TESTU S VYBRANÝMI NÁSTROJI

Konkrétní provedení testu závisí do značné míry na cíli simulovaného útoku a na typu testu. Přesto je ale možné uvést některé kroky, které se používají u převážné většiny penetračních testů.

1. **Průzkum** – nejdříve je třeba zjistit co nejvíce informací o cíli útoku – např. jaké technologie používá, jaký vlastní rozsah IP adres apod. Pro tento průzkum bude sloužit program `hostmap`. Oproti ostatním podobným aplikacím poskytuje zejména možnost výstupu přímo do XML.

2. **Zjištění strojů a služeb** – Skenováním se soustředíme na identifikaci aktivních strojů na síti, zjištění operačního systému a běžících služeb. Během tohoto kroku testu jsou využity aplikace `nmap` (zejm. pro množství konfiguračních voleb) pro aktivní a `p0f` pro pasivní skenování (rozdíl viz. např. [1]). `p0f` pro volbu pasivního sledování provozu na síti .

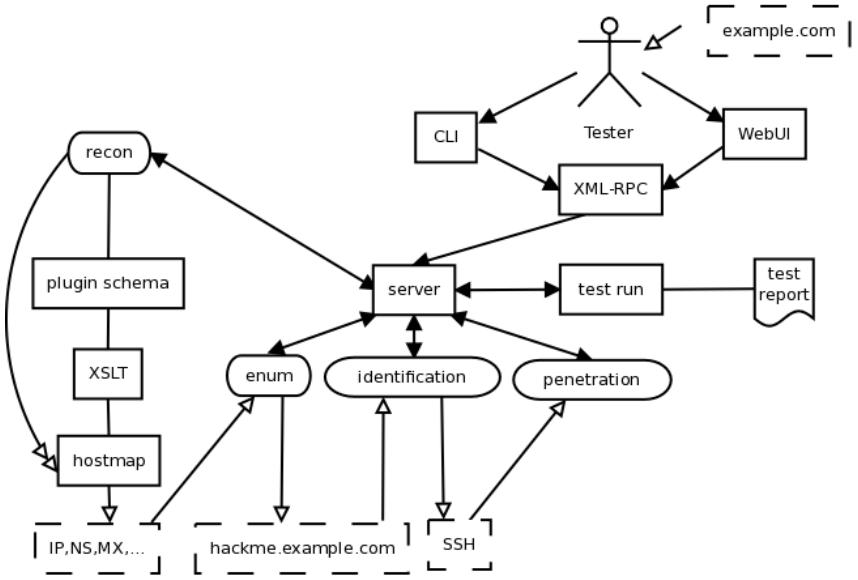
3. **Nalezení slabých míst** – na základě znalosti typů a verzí služeb nebo systémů snažíme zjistit (např. porovnáním s databází publikovaných chyb), která z nalezených služeb by mohla obsahovat obecně známou zranitelnost. Kromě informací zjištěných v přechotím kroku se využívá specializovaný program `Nikto`.

4. **Využití zranitelnosti k provedení útoku** – využití zranitelnosti v praxi znamená, že napadený program vykoná kód dodaný útočníkem, například otevře útočníkovi přístup k systému. Jako komponentu jsem zvolil program `Metasploit` který poskytuje možnost skriptování, databázi s obrovským množstvím programů pro využití bezpečnostních chyb a možnost strojově zpracovatelného výstupu.

[2]Computer Emergency Readiness Team, `http://www.us-cert.gov/`

4 NÁVRH FRAMEWORKU

Aplikace uživateli poskytuje jak řádkové tak webové rozhraní. Tyto komunikují s hlavní částí systému, zde označenou jako server pomocí XML-RPC. Server udržuje data o probíhajícím testu, kontroluje běh jednotlivých zásuvných modulů, představujících fáze testu a v neposlední řadě poskytuje programové rozhraní pro přístup k modulům.



Obrázek 1: Návrh vyvíjené aplikace. Moduly odpovídající obecným krokům penetračního testu jsou zaneseny ovály. Modul pro prvotní průzkum cíle (*recon*) je rozkreslen včetně podrobností. U ostatních části (zjištění strojů a služeb - *enum*, nalezení slabých míst - *identification* a samotný útok - *penetration*) jsou skryty. Čerchované obdélníky odpovídají testovaným prvkům cíle, obdélníky značené plnou čarou značí komponenty systému.

Ačkoliv většina zvolených nástrojů umožňuje výstup do formátu XML, nástroj používá vlastní schéma, do kterého výstup komponent překládá XSLT skripty. To umožňuje záměnu komponenty bez vlivu na formát dat, přenášených uvnitř systému.

5 ZÁVĚR

Tento příspěvek uvedl čtenáře do problematiky penetračního testování a seznámil ho s návrhem integrovaného systému pro provádění penetračních testů.

Momentálně je kromě zde popsaného návrhu systému hotov návrh schémat XML dokumentů, které budou sloužit pro uchovávání a předávání informací mezi jednotlivými fázemi testů. Důraz je kladen na implementaci sémantických analyzárotů či XSLT šablon pro import z formátů, které používají aplikace jako nativní, do schématu použitého vyvíjenou aplikací.

REFERENCE

[1] Hurley, C.: Penetration Tester's Open Source Toolkit, Volume 2, Syngress Publishing, 2007, ISBN 1597492132.

[2] Herzog, P.: Open-Source Security Testing Methodology Manual. ISECOM, 2009-11-23, [URL: `http://www.osstmm.org/` (navštíveno v únoru 2010)].