HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Metodologija penetracijskog testiranja

CCERT–PUBDOC–2008–02–219

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr – nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr – laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Etičan haker (eng. *ethical hacker, white hat*) je stručnjak u području računarstva i računalnih mreža. On napada računalne sustave pokušavajući iskoristiti njihove slabosti, ali, za razliku od zlonamjernih hakera (eng. *hacker, black hat*), on to čini prema nalogu vlasnika ciljanog sustava i upravo s ciljem identifikacije ranjivosti i sprečavanja djelovanja zlonamjernih hakera.

Rad etičnog hakera često se još naziva i penetracijskim testiranjem (eng. *penetration testing*). Penetracijsko testiranje, unatoč uvriježenom mišljenju, ipak nije samo pokretanje nekoliko alata i izrada izvještaja. U ovom je području, naime nasušno potrebna struktura i organizacija koja, osim propisivanja grupe testova, mora osigurati i odgovore na pitanja kvalitete, sveobuhvatnosti i ponovljivosti rezultata. Ta je struktura u ovom području obuhvaćena metodologijom, a, razvija se kako bi od penetracijskog testiranja učinila alat koji će pružiti što je moguće realističniju sliku o aktualnoj sigurnosti testiranog objekta.

U dokumentu dan kratak pregled osnova penetracijskog testiranja s posebnim naglaskom na metodologiju. Dokument opisuje uobičajene faze procesa penetracijskog testiranja, postojeće standarde te se osvrće na prednosti i nedostatke postojećih alata koji automatiziraju ovaj proces.

2. Općenito o penetracijskom testiranju

Penetracijsko testiranje je tehnika procjene sigurnosti računalnog sustava ili mreže koja se temelji na oponašanju stvarnog napada. Prilikom testiranja, ovlašteni ispitivač provjerava metu izvodeći različite vrste napada jednakim tehnikama koje bi koristio i da je stvarni napadač. Cilj mu je uočiti bilo kakvu ranjivost koju je moguće iskoristiti za ostvarenje neovlaštenog pristupa.

Iako ovakav način testiranja daje rezultate i uspijeva pronaći mane, često se precjenjuje njegov značaj. Penetracijsko testiranje ne može dokazati da je sustav potpuno siguran, bez ikakvih mana. Može jedino utvrditi određenu granicu količine znanja i rada neophodnog napadaču za uspješan prodor u sustav. Uz poznavanje spomenutih informacija moguće je jamčiti sigurnost sustava samo u određenim situacijama, a ne nikako općenito i generalno.

U samim počecima razvoja penetracijskog testiranja, vjerovalo se kako sve ranjivosti zadovoljavaju određene prepoznatljive obrasce te su se alati za automatsko provođenje testiranja razvijali upravo u smjeru traženja tih generičkih uzoraka. Nažalost, zbog velike količine različitih računala, operacijskih sustava i programskih jezika (s različitim sintaksnim i semantičkim pravilima), korištenih za izgradnju računalnih sustava, korištenje takvih automatskih alata prilikom penetracijskog testiranja postaje teško i nepraktično. Isto tako, skupo je mijenjati postojeće alate, kako bi odgovarali svim novim okruženjima i programskim jezicima.

Kako su moderni sustavi usmjereni na standarde nekoliko obitelji operacijskih sustava (npr. Unix, Linux, Windows) i ne odviše velik broj programskih jezika (npr. C, C#, Java), budući alati, koji rade na principu raspoznavanja obrazaca ranjivosti, mogli bi biti uspješniji. Počinje se javljati nekoliko specifičnih trendova: antivirusni alati za popularne radne stanice i osobna računala, Unix alati za sigurnosno testiranje, sustavi za detekciju upada u sustav, alati za formalno dokazivanje specifikacije, analizatori ranjivosti, alati za simboličko izvođenje i sl.

Netko se može pitati što to penetracijsko testiranje nudi, a pretraživanje ranjivosti ne. Obje tehnike, naime, mogu dati i daju procjenu mogućih napada na sustav. Ipak, pretraživanje ranjivosti procjenu stvara ne temelju automatiziranog procesa, te ga se može smatrati uvodnim korakom u stvarno penetracijsko testiranje koje potvrđuje ili opovrgava ranjivosti detektirane automatskim procesom. Za razliku od pretraživanja ranjivosti, penetracijsko testiranje daje reducirani skup detektiranih ranjivosti za koje se s izuzetno velikom sigurnošću može tvrditi da se doista nalaze u sustavu.

3. Faze procesa penetracijskog testiranja

3.1. Prikupljanje informacija

Prikupljanje informacija je postupak koji uključuje korištenje Interneta za pronalazak informacija o ciljnom sustavu, organizaciji ili osobi, uz pomoć tehničkih (DNS/WHOIS) i netehničkih metoda (pretraživači, novinske grupe, liste elektroničke pošte i sl.). Kod provođenja bilo kakvog testa na informacijskom sustavu, prikupljanje informacija je ključno i osigurava potrebne preduvjete za nastavak testiranja. Prilikom prikupljanja podataka, važno je biti što domišljatiji te pokušati istražiti sve moguće putove za lakše razumijevanje ciljnog sustava i njegovih resursa. U ovoj fazi testiranja, korisno je sve do čega se može doći: organizacijske brošure, poslovne kartice, prospekti, novinske reklame, interni papiri i dr.

Prikupljanje informacija ne zahtjeva uspostavljanje kontakta s ciljnim sustavom, budući da se informacije prikupljaju iz (uglavnom) javnih izvora na Internetu i organizacija koje drže javne informacije (npr. porezne agencije, knjižnice i dr.).

3.2. Mapiranje mreže

Specifične mrežne informacije prikupljene u prošloj fazi, koriste se i proširuju prilikom izrade vjerojatne mrežne topologije mete. U ovoj se fazi mogu koristiti razni alati i aplikacije, kao pomoć u otkrivanju tehničkih informacija o računalima i mrežama uključenim u testiranje.

Postupak mapiranja mreže obuhvaća:

- pronalazak aktivnih računala,

- pretraživanje priključaka i servisa,
- određivanje vanjskih rubova mreže (usmjerivača, vatrozida),
- identifikaciju kritičnih servisa,
- utvrđivanje informacija o operacijskom sustavu (eng. *OS fingerprinting*),
- identifikaciju ruta korištenjem MIB (eng. *Management Information Base*) baze i
- utvrđivanje informacija o servisima (eng. *Service fingerprinting*).

3.3. Identificiranje ranjivosti

Prilikom identifikacije ranjivosti, analizator detektira slabe točke sustava koje su pogodne za zloporabu. Aktivnosti kojima se postiže takva detekcija uključuju:

- identifikaciju ranjivih servisa korištenjem servisnih poruka (eng. *banners*),
- pretraživanje ranjivosti s ciljem otkrivanja poznatih nedostataka - informacije vezane uz poznate ranjivosti mogu se pronaći u proizvođačevim sigurnosnim oglasima ili u javnim bazama podataka, kao što su *SecurityFocus*, *Secunia* i sl.,
- popisivanje otkrivenih ranjivosti,
- procjenu očekivanog utjecaja (klasifikacija pronađenih ranjivosti) i
- identifikaciju putova napada i scenarija za zloporabu.

3.4. Penetracija

Analizator pokušava ostvariti neovlašten pristup zaobilaženjem sigurnosnih ograničenja, pri tome nastojeći dobiti što veće ovlasti. Ovaj se proces može podijeliti u nekoliko faza:

- Pronalazak programskog koda koji iskorištava ciljane ranjivosti (eng. *exploit*) – moguće je koristiti vlastiti repozitorij ili javno dostupne izvore. Ako je kôd iz vlastitog, sigurnog repozitorija, može se koristiti, a u suprotnom ga slučaju treba prvo testirati u izoliranom okruženju.
- Razvoj alata/skripti - u nekim okolnostima potrebno je kreirati vlastite alate i skripte za testiranje i za povećanje učinkovitosti.
- Testiranje alata/kôda za dokazivanje koncepta.
- Prilagodba alata/kôda za dokazivanje koncepta.
- Potvrda ili pobijanje postojanja ranjivosti - jedino testiranjem ranjivosti analizatori mogu sa sigurnošću potvrditi ili pobiti postojanje ranjivosti.
- Dokumentacija - mora sadržavati detaljan opis putova zloporabe, utjecaja koji je ostvaren na sustav i dokaza postojanja ranjivosti.

3.5. Dobivanje pristupa i povećanje ovlasti

Ukoliko u fazi penetracije nije uspio pokušaj ostvarenja pristupa ovaj korak nudi neke alternativne mogućnosti. Ako je, ipak, penetracija dala određene rezultate, jednako kao i kad to nije slučaj, ovaj korak omogućuje dodatno povećanje ovlasti.

3.5.1. Dobivanje pristupa

Ukoliko pokušaj penetracije nije dao odgovarajuće rezultate analizatoru na raspolaganju stoje još neke mogućnosti probijanja ugrađene zaštite:

- otkrivanje korisničkih imena/zaporki - tzv. napad riječnikom (eng. *dictionary attack*) i napad grubom silom (eng. *brute force attack*),
- otkrivanje praznih ili podrazumijevanih zaporki u sistemskim računima,
- iskorištavanje proizvođačevih podrazumijevanih postavki, kao što su parametri mrežne konfiguracije, zaporke i sl. te
- otkrivanje javnih servisa, koji dozvoljavaju obavljanje određenih operacija na sustavu, kao što je pisanje/stvaranje/čitanje datoteka.

Ukoliko pokušaj zaobilaženja zaporki urodi odgovarajućim rezultatom, slijedi kompromitacija ciljnog sustava. Ona može uključivati i kompromitaciju posrednih sustava, kako bi se zaobišle sigurnosne

mjere i ostvario pristup konačnom cilju. Ti mogući posredni sustavi mogu biti npr. usmjerivači, vatrozidi, domenski poslužitelji, radne stanice i dr.

3.5.2. Povećanje ovlasti

Često se događa da je u prethodnim fazama dobiven pristup sustavu neke niže razine. U tom specifičnom slučaju, potrebno je izvesti mapiranje lokalnih ranjivosti (kao suprotnost ranjivostima baziranim na mreži), iskorištavanje ili razvoj dokaza koncepta (testiranog u izoliranom okruženju i primijenjenog na kompromitirani sustav).
U ovoj je fazi je cilj steći administratorske ovlasti temeljem prethodno stečenih ovlasti manje važnosti. Glavne prepreke tom cilju su primijenjene ispravke koje analizatoru bitno umanjuju broj prisutnih ranjivosti, a time i mogućnost zlouporabe. Dodatnu prepreku predstavljaju alati za provjeru integriteta sustava (uključujući antivirusne alate), koji u nekim slučajevima mogu zaustaviti akcije analizatora.

3.6. Daljnje popisivanje objekata (eng. *Enumerating Further*)

Ova se faza sastoji od sljedećih koraka:

- otkrivanja kriptiranih zaporki za probijanje sustava koji nije spojen na mrežu (npr. kopiranjem datoteka /etc/passwd i /etc/shadow na Linux sustavima),
- otkrivanja zaporki (kriptiranih ili otvorenog teksta) korištenjem *sniffer* alata i drugih tehnika,
- analize prometa,
- prikupljanja kolačića (eng. *cookie*) i njihovog korištenja za iskorištavanje sjednica ili napade na zaporke,
- prikupljanja adresa elektroničke pošte,
- identifikacije ruta i mreža, i
- mapiranja internih mreža i ponovnog izvođenja prethodnih koraka, sa sustavom u danom stanju kao polaznom točkom.

3.7. Kompromitacija sustava

Jedna ranjivost u sustavu dovoljna je za izlaganje čitave mreže, neovisno o tome koliko je sigurna njena periferija. Svaki je sustav jak (u ovom slučaju siguran) onoliko, koliko su jaki njegovi najslabiji dijelovi. Komunikacija između udaljenih korisnika/podružnica i organizacijskih mreža može se zaštititi autentikacijom i enkripcijom, korištenjem tehnologija kao što je VPN (eng. *Virtual Private Network*), kako bi se osigurala autentičnost i privatnost prenošenih podataka. Međutim, to ne jamči pouzdanost krajnjih točaka u komunikaciji.
U takvim situacijama, analizator treba pokušati kompromitirati udaljene korisnike i/ili udaljene podružnice organizacije. Ovaj mu postupak može osigurati pristup internoj mreži.

3.8. Održavanje pristupa i skrivanje tragova

Održavanje pristupa i skrivanje tragova nezaobilazan su dio penetracijskog testiranja, a analizatoru osiguravaju stalnu i trajnu prisutnost na kompromitiranom sustavu bez mogućnosti razotkrivanja. Ovaj korak uključuje primjenu sljedećih alata i tehnika:

- skrivenih kanala,
- stražnjih vrata (eng. *backdoor*),
- tzv. *rootkit* alata,
- prikrivanja tragova,
- skrivanja datoteka,
- čišćenja zapisnika,
- poništenja provjere integriteta i
- poništenja antivirusnih alata.

Ovdje je bino naglasiti da se prikrivanje kanala, instalacija "stražnjih vrata" i *rootkit* alati uglavnom ne koriste kao dio uobičajenog penetracijskog testiranja, zbog rizika od ostavljanja nekog od ovih alata

pokrenutih i nakon kraja testiranja te, na taj način, ostavljanja mogućnosti stvarnom napadaču za njihovo iskorištavanje iskoristi u stvarnom napadu.

4. Postojeći standardi penetracijskog testiranja

Ljudi su često iznenađeni kada uvide mogućnost postojanja strukture i organizacije u procesu penetracijskog testiranja. Unatoč tome, postoji nekoliko izvora koji standardiziraju ovaj proces. Najpoznatiji među njima je onaj s instituta *Institute for Security and Open Methodologies*, poznat pod akronimom OSSTMM (eng. *Open Source Security Testing Methodology Manual*).

4.1. *Open Source Security Testing Methodology Manual* (OSSTMM)

OSSTMM je priručnik koji detaljno opisuje proces penetracijskog testiranja. Cilj priručnika, koji su njegovi autori postavili još za vrijeme njegovog sastavljanja, je definiranje stroge metodologije penetracijskog testiranja, pri čemu se moraju zadovoljiti tri uvjeta:

- konzistencija,
- ponovljivost i
- pouzdanost rezultata.

OSSTMM je u svojoj srži skup uputa kojima je cilj provođenje iscrpnih penetracijskih testova koji će pokriti sva potrebna područja, pritom pazeći na pridržavanje zakonskih odredbi. Rezultat testiranja učinjenog prema ovom standardu kvantificira stupanj opasnosti, konzistentan je i mora biti ograničen na prezentaciju pronađenih činjenica.

Sam priručnik podijeljen je u šest dijelova (eng. *channels*):

- informacijska sigurnost,
- sigurnost procesa,
- sigurnost Internet tehnologija,
- sigurnost komunikacija,
- sigurnost bežičnih tehnologija i
- fizička sigurnost.

Između ovih dijelova postoji mnogo preklapanja, a informacijska i fizička sigurnost predstavljaju okosnicu cijelog dokumenta.

OSSTMM popisuje tehničke pojedinosti o tome što treba testirati, kao i o tome što treba učiniti prije, za vrijeme i nakon samog testiranja. Kako bi određeno penetracijsko testiranje bilo usklađeno s OSSTMM standardom, ono mora pokriti sve module određenog poglavlja. Ukoliko ne postoji infrastruktura neophodna za izvođenje određene skupine testova, ta se skupina u konačnom izvješću označava oznakom NOT APPLICABLE.

4.2. *National Institute of Standards and Technology* (NIST) standard

Nacionalni institut znanosti i tehnologije Sjedinjenih Američkih Država (eng. *National Institute of Science and Technology* - NIST) načinio je dokument s naslovom *Special Publication 800-42, Guideline on Network Security Testing.*

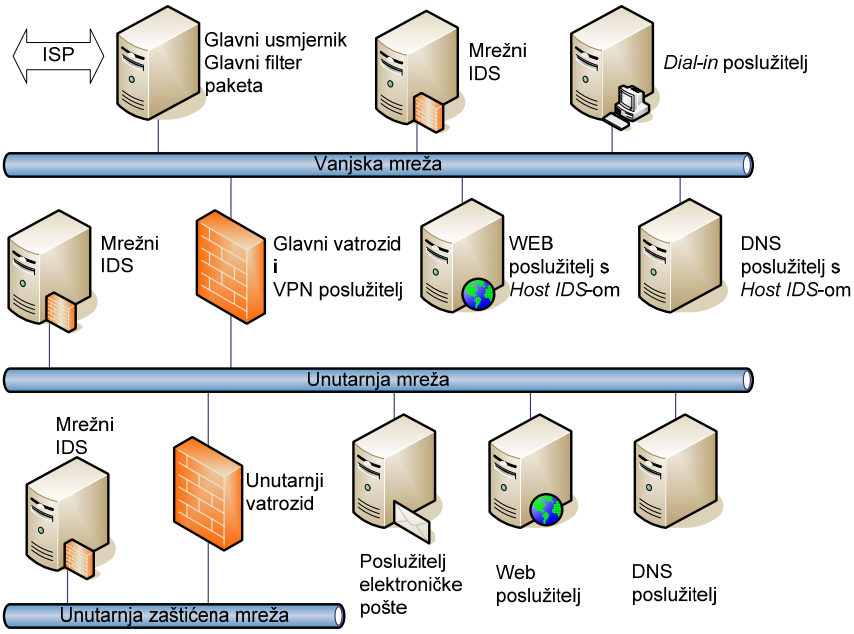
Cilj dokumenta je sistematično propisivanje elemenata testiranja sigurnosti u državnim organizacijama SAD-a. On identificira preduvjete koje je potrebno ispuniti za početak testiranja i preporuča prioritete kojima je moguće testiranje provesti i s ograničenim resursima. Također doprinosi izbjegavanju dvostrukog napora pružajući sustavan pristup cijeloj problematici. Dodatna mu je prednost i postojanje više razina testiranja koje, ovisno o organizaciji nad kojom se postupak primjenjuje, mogu dovesti do značajne uštede sredstava.

Temeljni naglasak dokumenta stavljen je na pružanje osnovnih informacija o alatima i tehnikama koje stoje na raspolaganju osobi koja želi započeti postupak testiranja sigurnosti. Dokument je, u prvom redu, usredotočen na aspekt mrežne sigurnosti, pri čemu je posebna pažnja posvećena slijedećim sustavima:

- vatrozidima (unutarnjim i vanjskim),
- usmjerivačima i preklopnicima,
- sustavima za zaštitu mrežnog okruženja (primjerice IDS – *Intrusion Detection System* sustavima),

- web, *e-mail* i ostalim aplikacijskim poslužiteljma, te
- ostalim poslužiteljima (DNS – *Domain Name System* , SMB – *Server Message Block*, NFS – *Network File System*, FTP – *File Transfer Protocol* i sl.)

Slijedeća slika prikazuje raspored pobrojanih sustava u uobičajenoj mrežnoj konfiguraciji.



Slika 1. Uobičajen razmještaj temeljnih elemenata računalne mreže

Općenito je preporučljivo testiranje započeti s pobrojanim ključnim elementima, a tek onda nastaviti s testiranjem manje bitnih poslužitelja, radnih stanica i testiranjem sigurnosne osviještenosti uključenog osoblja.

Testovi sadržani u dokumentu primjenjivi su na različite faze razvojnog ciklusa sustava, ali su ipak najkorisniji kao dio rutinskog programa testiranja mrežne sigurnosti koji se provodi nad računalima u njihovom uobičajenom operativnom stanju.

4.3. *Information Systems Security Assessment Framework* (ISSAF)

ISSAF je strukturirani radni okvir koji područje procjene sigurnosti računalnog sustava organizira u različite domene. Osim toga, on detaljno opisuje sasvim specifične testove koji se provode u svakoj od njih.

Iako uključuje vrlo opsežan skup sigurnosnih procedura, još se uvijek smatra standardom u razvoju, te se nije preporučljivo pouzdati u rezultate njegovog provođenja.

5. Alati za automatsko provođenje penetracijskog testiranja

5.1. CoreImpact

CoreImpact je komercijalan alat, proizvođača *Core Security Technologies*, namijenjen automatskom provođenju penetracijskog testiranja. Radi na principu oponašanja stvarnih napada na mrežne poslužitelje i radne stanice, krajnje sustave i web aplikacije. Omogućava pronalaženje i ispravljanje ranjivosti prije nego se dogodi stvaran sigurnosni incident.

Neke od značajki koje podržava su:

- provjera iskoristivih operacijskih sustava i servisa,
- mjerenje reakcije krajnjih korisnika na tzv. *phishing* i *spear phishing* napade, neželjene poruke i druge prijetnje elektroničke pošte i sl.,
- testiranje sigurnosti web aplikacija i demonstraciju posljedica web-utemeljenih napada,
- razlikovanje pravih prijetnji od lažnih,

- konfiguriranje i testiranje učinkovitosti IDS (eng. *Intrusion Detection System*), IPS (eng. *Intrusion Prevention System*) sustava, vatrozida i sličnih infrastruktura,
- potvrdu sigurnosti nadogradnje, izmjena i zakrpa sustava te
- uspostavljanje i održavanje postupaka testiranja ranjivosti.

Osim navedenog, alat omogućava i analizu stanja organizacijske sigurnosti u odnosu na tri najpoznatije metode napada:

- probijanje mrežnih sustava obrane, uz pomoć napada osmišljenih za iskorištavanje ranjivosti u operacijskim sustavima i servisima instaliranim na poslužiteljima, kao i ranjivosti klijentskih aplikacija pokrenutih na stolnim računalima,
- prijevare od strane zaposlenika, dobavljača i drugih krajnjih korisnika, uz pomoć napada socijalnim inženjeringom temeljenim na porukama elektroničke pošte i
- manipulaciju web aplikacija, radi pristupa podacima putem tzv. *SQL injection* napada ili napada uključivanjem udaljenih datoteka.

Najpoznatiji test CoreImpact alata jest brzi penetracijski test (eng. *Rapid Penetration Test*). Uz pomoć tog testa, korisnici mogu saznati sljedeće informacije:

- operacijske sustave i kritične servise,
- aplikacije krajnjih točaka (web preglednike, čitače elektroničke pošte, komunikaciju u stvarnom vremenu (eng. *instant messaging*), alate za reprodukciju multimedijalnih sadržaja, poslovne aplikacije i sl.),
- sigurnosna rješenja krajnjih točaka (antivirusne alate, *anti-phishing*, *anti-malware* alate i sl.),
- informiranost krajnjih korisnika o napadima socijalnim inženjeringom, neželjenim porukama elektroničke pošte i dr. ,
- korištene web aplikacije, kao što su Internet bankarstvo, ERP (eng. *Enterprise Resource Planning*) sustavi i sl.,
- prisutne IDS (eng. *Intrusion Detection System*) i IPS (eng. *Intrusion Prevention System*) sustave, vatrozide i druge sigurnosne alate,
- rezultate pretraživača ranjivosti i
- sigurnosne politike.

RPT provodi automatsko penetracijsko testiranje putem sljedećih šest koraka:

1. prikupljanje informacija,
2. napad i penetracija,
3. prikupljanje lokalnih informacija,
4. povećanje ovlasti,
5. čišćenje i
6. generiranje izvještaja.

Slika 2. Šest koraka RPT postupka

5.2. Metasploit

Metasploit je projekt otvorenog koda, namijenjen analizi računalne sigurnosti. Osigurava informacije o sigurnosnim ranjivostima te pomaže u izvođenju penetracijskog testiranja i razvoju IDS potpisa. Njegov najpoznatiji podproizvod jest Metasploit okruženje (eng. *Metasploit Framework*), razvojna platforma za izradu i izvođenje kôda i alata za zloporabu na udaljenim računalima.

Kao i drugi alati namijenjeni zaštiti informacijske sigurnosti, Metasploit se može koristiti za zakonite i neovlaštene aktivnosti.

Metasploit okruženje koriste sigurnosni stručnjaci za izvođenje penetracijskog testiranja, administratori sustava za provjeru instalacija zakrpa, proizvođači za testiranje regresije ranjivosti, ali i ostali sigurnosni istraživači u različite svrhe.

Ovo je okruženje postojalo u tri različite inačice, od srpnja 2003. godine. Treća (3.x) i trenutna inačica je nastala potpunom izmjenom prethodnih i pisana je jezikom Ruby, a razvio ju je Metasploit LLC tim i dostupna je pod *Metasploit Framework* licencom.
Metasploit okruženje sastoji se od alata, biblioteka, modula i korisničkog sučelja. Osnovna funkcija okruženja je učitavanje modula, što korisnicima omogućava konfiguraciju i iskorištavanje modula za napad na ciljni sustav. U nomenklaturi Metasploit okruženja bitno je razlikovati pojmove modula, *exploit*-a i *payload*-a. Modul je dio Metasploit okruženja namijenjen iskorištavanju jedne specifične ranjivosti. Njemu pripada i određeni programski kod (*exploit*) posebno oblikovan iskorištavanju dotične ranjivosti. Cilj iskorištavanja ranjivosti je pokrenuti izvođenje posebno oblikovanog programskog koda na ciljnom računalu. Taj programski kod obično se naziva *payload*, univerzalan je za sve *exploit*-e, a specifičan za svaku računalnu arhitekturu. Metasploit okruženje sadržava nekoliko desetaka takvih kodova namijenjenih različitim funkcijama i oni se na ciljnom računalu koriste u kombinaciji s odgovarajućim *exploit*-om.
Osnovni koraci iskorištavanja ranjivosti sustava korištenjem Metasploit okruženja su:

- Odabir i konfiguracija kôda (eng. *exploit*) kojim se prodire u ciljni sustav uz iskorištavanje neke od poznatih ranjivosti. U Metasploit okruženje uključeno je 200-tinjak različitih *exploit* kôdova za Windows, Unix/Linux i Mac OS X operacijske sustave.
- Provjera osjetljivosti ciljnog sustava na korištenu ranjivost. Ovaj je korak opcionalan.
- Odabir i konfiguracija kôda, čije će izvršavanje biti pokrenuto na ciljnom sustavu u slučaju uspješne zloporabe propusta (eng. *payload*). Primjerice, može se raditi o udaljenoj ljusci (eng. *shell*) ili VNC (eng. *Virtual Network Computing*) poslužitelju.
- Odabir tehnike za kodiranje *payload*-a, tako da ga sustav za detekciju upada (IDS) ne uoči.
- Pokretanje izvođenja *exploit* kôda.

Metasploit je podržan na Linux, Windows, Mac OS X i većini BSD operacijskih sustava te na raznim sklopovskim platformama (od Unix središnjih računala do Nokia n800 dlanovnika). Koristi tzv. *tab-completing* sučelje, Gtk GUI (eng. *Graphical User Interface*) sučelje, skriptno sučelje naredbenog retka i web sučelje s omogućenom AJAX funkcionalnošću.

5.3. Canvas

Immunity Canvas je komercijalan alat koji se koristi kao pomoć timovima za provođenje penetracijskog testiranja, a omogućava korištenje ugrađenih *exploit* kôdova. Baza podataka se nadograđuje jednom tjedno. Alat je otvorenog dizajna i prilagodljiv potrebama korisnika. Sadrži i neke kodove koji nisu dostupni nigdje drugdje.
Podržan je na sljedećim platformama i instalacijama:

- Windows (potreban mu je Python i PyGTK sučelje),
- Linux,
- Mac OS X (potrebno mu je PyGTK sučelje) i
- svim drugim Python okruženjima, kao što su mobilni telefoni i komercijalni Unix sustavi (inačice naredbenog retka).

Alat trenutno sadrži oko 150 *exploit* kôdova, a mjesečno ih se dodaje u prosjeku još po četiri. Najveći prioritet alata je iskorištavanje ranjivosti velikog sigurnosnog rizika, npr. udaljeni napadi i iskorištavanje novih ranjivosti. Ranjivosti koje alat provjerava i pokušava iskoristiti također obuhvaćaju sve platforme i aplikacije.

6. Prednosti i nedostaci automatskog provođenja penetracijskog testiranja

Donedavno je penetracijsko testiranje bio vrlo kompleksan ručni proces, koji je moglo izvesti samo nekoliko sigurnosnih specijalista s dugogodišnjim iskustvom. Ispitivači najčešće moraju pisati vlastite kôdove za zloporabu, naučiti dobro koristiti alate iz javne domene te izvoditi mnoge zamorne i vremenski skupe zadatke. Osim što može biti iscrpno, ručno testiranje obično zahtjeva velik tim profesionalaca, s različitim vještinama, a to si većina organizacija ne može priuštiti.
Automatsko provođenje penetracijskog testiranja, s druge strane predstavlja dobar proizvod za izvođenje penetracijskog testiranja. Potrebne alate razvija tim sigurnosnih stručnjaka, koji kombiniraju sigurnu zloporabu i jednostavne pakete. Samo korištenje alata je prepušteno osobama s

daleko nižim stupnjem tehničkog znanja koje testiranjem svih radnih stanica i poslužitelja diljem mreže, mogu načiniti čist i iscrpan pogled na sigurnosno stanje organizacije.
Slijedeća tablica demonstrira prednosti i nedostatke automatskog i ručnog provođenja penetracijskog testiranja.

	Ručno penetracijsko testiranje	Automatsko penetracijsko testiranje
PROCES TESTIRANJA	Radno-intenzivno, nedosljedno i sklono pogreškama, s nespecificiranim standardima kvalitete. Zahtijeva mnogo bitno različitih alata. Rezultati mogu značajno varirati od testa do testa. Općenito zahtjeva stručno sigurnosno osoblje za pokretanje testa i interpretaciju rezultata.	Brzo, jednostavno i sigurno. Eliminira pogreške zamornih ručnih zadataka. Centralizirano i standardizirano s ciljem dobivanja konzistentnih i ponovljivih rezultata. Jednostavno za korištenje i osigurava čiste izvještaje na temelju kojih se može reagirati.
MODIFIKACIJE MREŽE	Često se dogode mnoge izmjene na sustavu.	Sustavi se ne mijenjaju.
ISKORIŠTAVANJE, RAZVOJ I UPRAVLJANJE	Razvoj i održavanje baze zloporaba je vremenski skupo i zahtjeva značajnu stručnost. Javne zloporabe su sumnjive i nesigurne za pokretanje. Ponovno pisanje i unošenje koda je neophodno za više-platformnu funkcionalnost.	Proizvođač proizvoda razvija i održava sve kodove te ih kontinuirano nadograđuje za postizanje maksimalne učinkovitosti. Kodove pišu profesionalci, temeljito ih testiraju i čine ih sigurnima za pokretanje. Pisani su i optimizirani za različite platforme i vektore napada.
ČIŠĆENJE	Ispitivač mora zapamtiti i poništiti sve izmjene. Nakon napada, mogu ostati otvorena stražnja vrata na sustavu.	Vodeći proizvodi nude iscrpno uklanjanje izmjena i stražnja vrata se nikad ne instaliraju.
STJECANJE I POVEĆANJE OVLASTI	Zahtjeva se izmjene sustava, budući da se kôd mora postaviti i prevesti na kompromitiranom računalu.	Korisnici mogu brzo prodrijeti dublje u mrežu. Kôd se nikad ne mora postavljati na ciljno računalo i testovi se mogu provesti udaljeno.
IZVJEŠTAVANJE	Zahtjeva značajan trud, bilježenje i uspoređivanje svih rezultata ručno. Svi izvještaji moraju biti generirani ručno.	Iscrpni prikaz prethodnih događaja i pronađenih mana generiraju se automatski i prilagodljivi su.
ZAPISIVANJE / ANALIZA	Spor, težak, često netočan proces.	Automatski se snima detaljno izvješće o svim aktivnostima.
OBUČAVANJE	Ispitivači moraju naučiti nestandardizirane, *ad-hoc* metode testiranja.	Korisnici mogu naučiti i instalirati alat za manje od jednog dana.

Tablica 1. Usporedba automatskog i ručnog penetracijskog testiranja

Iako bi se iz tablice moglo zaključiti kako je automatsko penetracijsko testiranje apsolutni favorit, tj. da automatsko testiranje ima samo pozitivne strane, to ipak nije tako. Automatskim testiranjem se uistinu dobiva konkretan i pouzdan rezultat, ali automatsko testiranje, za razliku od ručnog testiranja iskusnog čovjeka, ne može osigurati testiranje svih teško iskoristivih ranjivosti. Zbog toga je, kod automatskog testiranja sasvim moguća situacija u kojoj alat ne može pronaći ranjivost sustava koju će stvaran napadač ipak pronaći i uspješno iskoristiti.
Automatsko testiranje, s druge strane nudi dobru obranu za manje kritične servise, jer ih ipak osigurava od većine zlonamjernih ljudi koji nemaju dovoljno znanja da bi iskoristili one "najmanje" nedostatke.

7. Zaključak

Što se tiče budućeg razvoja penetracijskog testiranja, pretpostavlja se da će bitka između stručnjaka koji razvijaju sigurne računalne sustave i hakera, jednako kao i u mnogim sličnim suprotstavljenim područjima ljudske domišljatosti, ostati neriješena. U ovom konkretnom primjeru ova je pretpostavka utemeljena na zatvorenom krugu razvoja sigurnosti koji potiče izgradnju novih tehnologija koje opet nude novi prostor za otkrivanje novih ranjivosti i novih načina zlouporabe.

Unatoč dosad izrečenom pesimizmu, penetracijsko testiranje ipak predstavlja jednu od svega nekoliko tehnika koje se u ovom trenutku mogu suprotstaviti sigurnosnim prijetnjama. Ono je započelo kao ručni proces, koji je u novije vrijeme sve više automatiziran. Ipak, u ovom trenutku vrhunsko penetracijsko testiranje, bez obzira na razvoj tehnologije i mogućnost automatizacije, nikako ne može biti izvedeno bez vrhunski obrazovanih ljudi.

Štoviše, pokazuje se da ni vrhunski kadar, sam za sebe, nije dovoljan za rješenje ovog problema. Sastojak koji nedostaje je standard, sustav koji će unificirati rezultate penetracijskog testiranja, dovesti do njihove veće pouzdanosti i , na koncu, omogućiti njihovo što bolje korištenje u donošenju poslovnih odluka.

Metodologija penetracijskog testiranja upravo se bavi rješenjem tog problema i ona u ovom trenutku predstavlja najslabiju kariku cijelog procesa. Ipak se, jednako tako, u posljednje vrijeme vrlo brzo razvija, što pruža osnovu za optimistična očekivanja vezana uz dominaciju tehnika obrane računalnih sustava nad tehnikama i mogućnostima njihove zlouporabe .

8. Reference

[1] Manual Penetration Testing vs. Automated Penetration Testing, http://www.coresecurity.com/?module=ContentMod&action=item&id=26 , veljača 2008.

[2] Core Impact Overview, http://www.coresecurity.com/?module=ContentMod&action=item&id=32, veljača 2008.

[3] Metasploit Framework, http://metasploit3.com/ , veljača 2008.

[4] Immunity CANVAS professional, http://www.immunitysec.com/products-canvas.shtml , veljača 2008.

[5] http://www.vulnerabilityassessment.co.uk/Framework.png , veljača 2008.

[6] Penetration Testing Methodology, http://www.asociacion-aecsi.es/doc/Auditing/Penetration_Testing_Methodology.pdf, veljača 2008.

[7] Penetration Testing Framework, http://www.vulnerabilityassessment.co.uk/Framework.png, veljača 2008.

[8] Penetration Testing, http://www.acsac.org/secshelf/book001/11.pdf, veljača 2008.

[9] Penetration Testing: Networks and Applications Staying Several Steps Ahead of the Attackers, http://www.sacure.com/pentesting%20-%20whitepaper.pdf, veljača 2008.

[10] Penetration Testing Methodology and Case Study, http://itunit.ipa.ie/ICSCONF2007/Presentations/Hugh%20Callaghan%20-%20Penetration%20testing%20.pdf, veljača 2008.

[11] Information Systems Security Assessment Framework (ISSAF), http://www.oissg.org/issaf/index.php, veljača 2008.

[12] Guideline on Network Security Testing, http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf, veljača 2008.

[13] OSSTMM 2.2. (Open Source Security Testing Methodology Manual), http://isecom.securenetltd.com/osstmm.en.2.2.pdf, veljača 2008.