PREVOD PRVOG IZDANJA

Bash Shell skriptovanje za testiranje neprobojnosti

Ovladajte veštinom korišćenja komandne linije i unapredite svoje metode testiranja neprobojnosti

STIV KEMPBEL

Predgovor napisao Dejvid Kenedi, osnivač TrustedSec i Binary Defense

kompjuter biblioteka

Bash Shell skriptovanje
za testiranje neprobojnosti

Ovladajte veštinom korišćenja komandne linije i unapredite svoje metode testiranja neprobojnosti

STIV KEMPBEL

kompjuter biblioteka

Packt›

Izdavač:

kompjuter biblioteka

Obalskih radnika 4a
Beograd, Srbija
Tel: 011/2520272
e-pošta: kombib@gmail.com
veb-sajt: www.kombib.rs

Za izdavača:
Mihailo J. Šolajić, urednik

Autor: Stiv Kempbel

Prevod: Nemanja Lukić
Recezent: Miroslav Ristić
Slog: Zvonko Aleksić

Znak Kompjuter biblioteke:
Miloš Milosavljević

Štampa:
Tiraž: 500
Godina izdanja: 2025.

Broj knjige: 586
Izdanje: Prvo
ISBN: 978-86-7310-608-3

Naslov originala:

Bash Shell Scripting for Pentesters
ISBN 978-1-83588-082-1

Packt Publishing Ltd.
Birmingham, UK, packt.com

Bash Shell skriptovanje za testiranje neprobojnosti

Autorizovani prevod sa engleskog jezika.

PREDGOVOR

Tokom višegodišnjeg rada kao član crvenog tima i tester neprobojnosti, jedno od najmoćnijih alata u mom arsenalu bilo je duboko razumevanje Linux operativnog sistema – posebno Bash skriptovanja. Bilo da pretražujem ogromne količine podataka ili razvijam prilagođene alate za iskorišćavanje ranjivosti, svestranost Bash interpretera je nenadmašna. Omogućava fleksibilnost za automatizovanje procesa, rad sa funkcijama sistema i optimizaciju zadataka koji bi inače zahtevali značajnu količinu vremena. U ofanzivnoj bezbednosti, ovladavanje Bash interpreterom nije samo korisno – ono je neophodno. Ova knjiga obuhvata suštinu toga zašto je Bash skriptovanje ključna veština za testere neprobojnosti, pokazujući vam kako da iskoristite njegov puni potencijal.

Ovaj vodič vas vodi kroz svaku fazu testa neprobojnosti, demonstrirajući kako primeniti Bash skriptovanje da biste bili efikasniji. Od izviđanja, gde je prikupljanje informacija ključno, do iskorišćavanja i post-iskorišćavanja, gde preciznost igra presudnu ulogu, ova knjiga pruža praktične primere kako se Bash može koristiti u ofanzivnim bezbednosnim scenarijima. Nije reč samo o praćenju skriptova – već o sticanju znanja potrebnog za pisanje sopstvenih prilagođenih skriptova koji su usmereni na specifične izazove sa kojima se susrećete na terenu. Praktičan pristup koji se ovde koristi osigurava da izgradite i samopouzdanje i veštine u korišćenju Bash interpretera za svaki aspekt napada.

Pored tehničkih veština, ova knjiga vam daje okvir za razmišljanje kao tester neprobojnosti – kako proceniti situacije, prilagoditi se novim izazovima i kreirati rešenja u hodu. Stiv ne pokriva samo tehničke aspekte već vam pruža uvide u to kako pristupiti rešavanju problema, nudeći osnove koje vam omogućavaju da razvijate sopstvene alate kada je to potrebno. Za svakoga ko želi da nauči Bash skriptovanje za ofanzivne operacije, ova knjiga je neprocenjiv resurs. Oprema vas veštinama i načinom razmišljanja neophodnim da budete prilagodljivi, inovativni i, na kraju krajeva, uspešni u svojoj karijeri u oblasti sajber bezbednosti.

Dejvid Kenedi

Osnivač kompanija TrustedSec i Binary Defense

SARADNICI

O AUTORU

Stiv Kembel je tehnički lider u CDW Offensive Security timu. On je penzionisani mornarički veteran koji je ranije radio sa električnim i elektronskim sistemima u avijaciji pre nego što se prebacio na informacionu tehnologiju (IT). Ima više od 19 godina kombinovanog iskustva u IT industriji i testiranju neprobojnosti. Planirao je, koordinisao, vodio i sprovodio testiranja neprobojnosti za različite velike kompanije, uključujući Fortune 500, državne institucije, banke, finansije, zdravstvenu zaštitu i osiguranje, e-trgovinu, pravne firme i klijente iz energetske industrije. Njegova dostignuća uključuju identifikaciju sedam ranjivosti objavljenih kao CVE, kao i doprinose alatima otvorenog koda kao što je Metasploit Framework.

O RECENZENTIMA

Džajaraman Manimaran je iskusni tester bezbednosti sa više od 9 godina iskustva u DevSecOps praksi, testiranju neprobojnosti, simulaciji napada i kombinovanim testovima odbrane i napada. Upravljao je složenostima testiranja usluga za različite sektore, uključujući bankarstvo, finansije i telekomunikacije, donoseći bogatstvo praktičnog znanja u proces evaluacije. Njegova posvećenost širenju znanja ogleda se kroz tehničke blogove, istraživanje bezbednosti i objavljivanje skriptova usmerenih na pojednostavljenje izazova sa kojima se suočavaju testeri bezbednosti. Poseduje sertifikate: CHMRTS, MCRTA, CARTP, CRTP, CRTA, eCPPT, CRT-ID, eWPT, CRT-COL, eJPT, PTP i C|EH.

> *Iskreno zahvaljujem svojoj porodici, posebno svojoj podržavajućoj supruzi, što su uz mene i razumeju moj zahtevan raspored. Posebna zahvalnost autoru i Packt timu na neprocenjivoj prilici da doprinesem ovom izdanju. Vaša podrška i razumevanje učinili su moju ulogu tehničkog recenzenta mogućom.*

Endru Aurand je predavač po ugovoru na Univerzitetu Vilmington. Radio je u IT sektoru 14 godina. Predavao je uvodni Python, uvodni Linux, napredne Linux teme i etičko hakovanje studentima osnovnih studija. Takođe je suosnivač kompanije za sajber bezbednost zasnovane na rešenjima pod nazivom Cipherlock Solutions. Ima master diplomu iz sajber bezbednosti sa Univerziteta Vilmington.

Entoni „RedHetOgust" Radžikevič je iskusni stručnjak za sajber bezbednost sa više od 10 godina iskustva u testiranju neprobojnosti, analizi pretnji i proceni ranjivosti. Kao OSCP sertifikovani stručnjak, predavao je Linux i sajber bezbednost kao gostujući profesor i razvio sigurne Linux distribucije za visoko rizična okruženja. Njegova karijera uključuje rad kao tester bezbednosti za Fortune 100 kompaniju i kao autor sadržaja u OfSek kompaniji, gde je napisao sveobuhvatne obrazovne materijale. Kao posvećeni recenzent i industrijski stručnjak, Entoni pruža pronicljive i pristupačne komentare koji premošćuju tehničke detalje sa angažovanim pristupom za sve čitaoce.

O RECENZENTU ZA SRPSKO IZDANJE

Miroslav Ristić je redovni profesor na Prirodno-matematičkom fakultetu Univerziteta u Nišu, sa preko 25 godina iskustva u razvoju statističkog softvera. Posebno se ističe njegov rad na razvoju grafičkog korisničkog interfejsa R Commander za programski jezik R. Dugi niz godina recenzirao je značajan broj knjiga za izdavačku kuću Springer i časopis Journal of Applied Statistics. Od 2023. godine aktivno recenzira najaktuelnija izdanja izdavačke kuće „Kompjuter biblioteka". Nakon prevođenja, svako izdanje prolazi kroz njegovo stručno vrednovanje i recenziju prevoda, sa ciljem da se osigura da prevodi budu ne samo jasni, precizni i prilagođeni čitaocima, već i da održe visok kvalitet i stručnu relevantnost knjiga.

Predgovor

Bash skriptovanje je osnovna veština u alatima testera neprobojnosti, omogućavajući automatizaciju složenih bezbednosnih procena, analize ranjivosti i zadataka iskorišćavanja. Ova knjiga pruža sveobuhvatan vodič za savladavanje Bash skriptovanja posebno prilagođenog za testiranje neprobojnosti, pokrivajući sve, od osnovnih koncepata skriptovanja do naprednih tehnika za izbegavanje detekcije i integraciju sa modernim tehnologijama poput veštačke inteligencije (AI).

Knjiga je organizovana u tri dela, vodeći čitaoce od osnovnih koncepata, preko praktične primene u testiranju neprobojnosti, do naprednih tema. Naučićete kako da koristite Bash za izviđanje, testiranje veb aplikacija, procenu mrežne infrastrukture, povećanje privilegija i održavanje pristupa. Fokus knjige je na praktičnom učenju kroz realne primere i scenarije sa kojima se testeri neprobojnosti susreću u svakodnevnom radu.

Za koga je ova knjiga

Ova knjiga je namenjena sledećim ciljnim grupama:

- Stručnjacima za bezbednost i testerima neprobojnosti koji žele da automatizuju svoj rad pomoću Bash interpretera
- Administratorima sistema koji žele da unaprede svoje sposobnosti testiranja bezbednosti
- Istraživačima bezbednosti zainteresovanim za razvoj prilagođenih alata i skriptova
- DevSecOps stručnjacima koji žele da integrišu testiranje bezbednosti u svoje procese
- Studentima i budućim testerima neprobojnosti koji žele da izgrade solidne osnove u automatizaciji

Osnovno poznavanje Linux/Unix sistema i rada u komandnoj liniji je korisno, ali nije neophodno, jer knjiga pokriva gradivo od osnovnih koncepata do naprednih tehnika. Potrebno je imati osnovna znanja i računarske resurse za kreiranje virtuelnih mašina i instalaciju Kali Linux operativnog sistema.

Šta sadrži ova knjiga

Poglavlje 1: Bash komandna linija i okruženje za testiranje – Uvod u osnove Bash skriptovanja u kontekstu testiranja neprobojnosti. Pokriva odabir pravog operativnog sistema, konfiguraciju Bash okruženja i postavljanje osnovnih alata za testiranje neprobojnosti.

Poglavlje 2: Upravljanje datotekama i direktorijumima – Rad sa datotekama i direktorijumima, osnovne komande za navigaciju, rad sa datotekama, dozvolama i povezivanje – ključne veštine za svakog testera neprobojnosti.

Poglavlje 3: Promenljive, uslovni izrazi, petlje i nizovi – Ključni koncepti programiranja u Bash interpreteru, uključujući upotrebu promenljivih, struktura odlučivanja i iteracija kroz podatke.

Poglavlje 4: Regularni izrazi – Uvod u prepoznavanje obrazaca i rad sa tekstom pomoću regularnih izraza, što je ključno za parsiranje izlaznih podataka i automatizovanu analizu podataka.

Poglavlje 5: Funkcije i organizacija skriptova – Kako kreirati modularne i lako održive skriptove pomoću funkcija, od osnovnog kreiranja funkcija do naprednih tehnika kao što je rekurzija.

Poglavlje 6: Bash mrežno upravljanje – Fokus na mrežno skriptovanje, umrežavanje, otkrivanje, rešavanje problema i iskorišćavanje mrežnih usluga.

Poglavlje 7: Paralelna obrada podataka – Tehnike za pokretanje više zadataka istovremeno, što je ključno za efikasno skeniranje i testiranje velikih ciljanih okruženja.

Poglavlje 8: Izviđanje i prikupljanje informacija – Otkrivanje ciljanih resursa pomoću DNS identifikacije, pretrage poddomena i OSINT alata.

Poglavlje 9: Testiranje neprobojnosti veb aplikacija pomoću Bash skriptovanja – Automatizovano testiranje veb aplikacija, uključujući automatizovane zahteve, analizu odgovora i detekciju ranjivosti.

Poglavlje 10: Testiranje neprobojnosti mreža i infrastrukture Bash skriptovanjem – Skeniranje mreža, identifikacija i automatizacija testiranja ranjivosti.

Poglavlje 11: Povećanje privilegija Bash shell skriptovanjem – Identifikacija i iskorišćavanje primena za povećanje privilegija pomoću Bash skriptovanja.

Poglavlje 12: Održavanje pristupa i preusmeravanje napada – Održavanje pristupa kompromitovanim sistemima i proširenje napada preko mrežnog preusmeravanja.

Poglavlje 13: Izveštavanje o testiranju neprobojnosti pomoću Bash skriptova – Automatizacija kreiranja profesionalnih izveštaja o testiranju neprobojnosti.

Poglavlje 14: Prikrivanje aktivnosti i maskiranje – Tehnike izbegavanja detekcije tokom testiranja neprobojnosti.

Poglavlje 15: Integracija sa veštačkom inteligencijom – Kako integrisati mogućnosti veštačke inteligencije u tokove testiranja neprobojnosti.

Poglavlje 16: DevSecOps za testere neprobojnosti - Završava knjigu implementacijom testiranja bezbednosti u CI/CD tokovima i automatizacijom bezbednosnih provera u modernim razvojnim okruženjima.

Kako da najbolje iskoristite ovu knjigu

Da biste maksimalno iskoristili učenje iz ove knjige, trebalo bi da imate sledeće:

- Razumevanje osnovnih bezbednosnih principa.
- Pristup Linux okruženju (Kali Linux) za vežbanje primera.
- Poznavanje osnovnih koncepata virtuelizacije, uključujući mogućnost kreiranja i pokretanja virtuelnih mašina.
- Pristup računaru sa dovoljnim resursima za istovremeno pokretanje dve virtuelne mašine.

SOFTVER/HARDVER POKRIVEN U KNJIZI	ZAHTEVI OPERATIVNOG SISTEMA
Kali Linux	Linux
Bash	

Važno je da imate potrebne resurse i znanje za kreiranje i pokretanje virtuelnih mašina. Ova knjiga ne pokriva instaliranje Linux operativnog sistema ili kreiranje virtuelnih mašina.

Preuzmite datoteke sa primerima koda

Možete preuzeti datoteke sa primerima koda iz ove knjige sa GitHub platforme:

`https://github.com/PacktPublishing/Bash-Shell-Scripting-for-Pentesters`

Ako dođe do ažuriranja koda, ono će biti dostupno u GitHub spremištu.

Takođe, možete pronaći i druge pakete koda iz širokog kataloga knjiga i video zapisa na:

`https://github.com/PacktPublishing Pogledajte ih!`

Konvencije korišćene u ovoj knjizi

Postoji niz konvencija koje su korišćene u ovoj knjizi.

`Kod u tekstu`: Označava delove koda u tekstu, nazive tabela baza podataka, nazive direktorijuma, nazive datoteka, ekstenzije datoteka, nazive putanja, lažne URL adrese, korisnički unos i Twitter korisnička imena. Primer: *„Sada, kada treba da pristupite ovom direktorijumu, možete jednostavno ukucati* cd `$MOJ_DUBOK_DIREKTORIJUM`, *i Bash će vas odmah prebaciti tamo."*

Blok koda je postavljen na sledeći način:

```
#!/usr/bin/env bash
if [ $USER == 'steve' ] && [ -f "/path/to/file.txt" ]; then
    echo "Zdravo, Steve. Datoteka postoji."
elif [ $USER == 'admin' ] || [ -f "/path/to/admin_file.txt" ]; then
    echo "Administratorski pristup odobren ili admin datoteka
postoji."
```

Bilo koji unos ili izlaz komandne linije prikazan je na sledeći način:

```
$ cd /home
```

Podebljano: Označava novi termin, važnu reč ili reči koje vidite na ekranu. Na primer, reči u menijima ili dijaloškim okvirima pojavljuju se **podebljano**. Primer**:** „Na kartici **Model Setting** odaberite model i postavite **Freedom** na **Precise**. Kliknite na dugme **Save**."

> **Saveti ili** važne napomene
> pojavljuju se ovako.

Odricanje odgovornosti

Informacije u ovoj knjizi namenjene su za korišćenje isključivo u etičke svrhe. Nemojte koristiti nijednu informaciju iz knjige ako nemate pismenu dozvolu vlasnika opreme. Ako izvodite nelegalne radnje, postoji mogućnost da budete procesuirani u punoj meri zakona. Ni izdavačka kuća Packt Publishing ni autor ove knjige ne preuzimaju odgovornost za eventualnu zloupotrebu informacija sadržanih u knjizi. Ove informacije moraju se koristiti samo u okruženjima za testiranje uz odgovarajuće pismeno odobrenje odgovornih lica.

Kontaktirajte nas

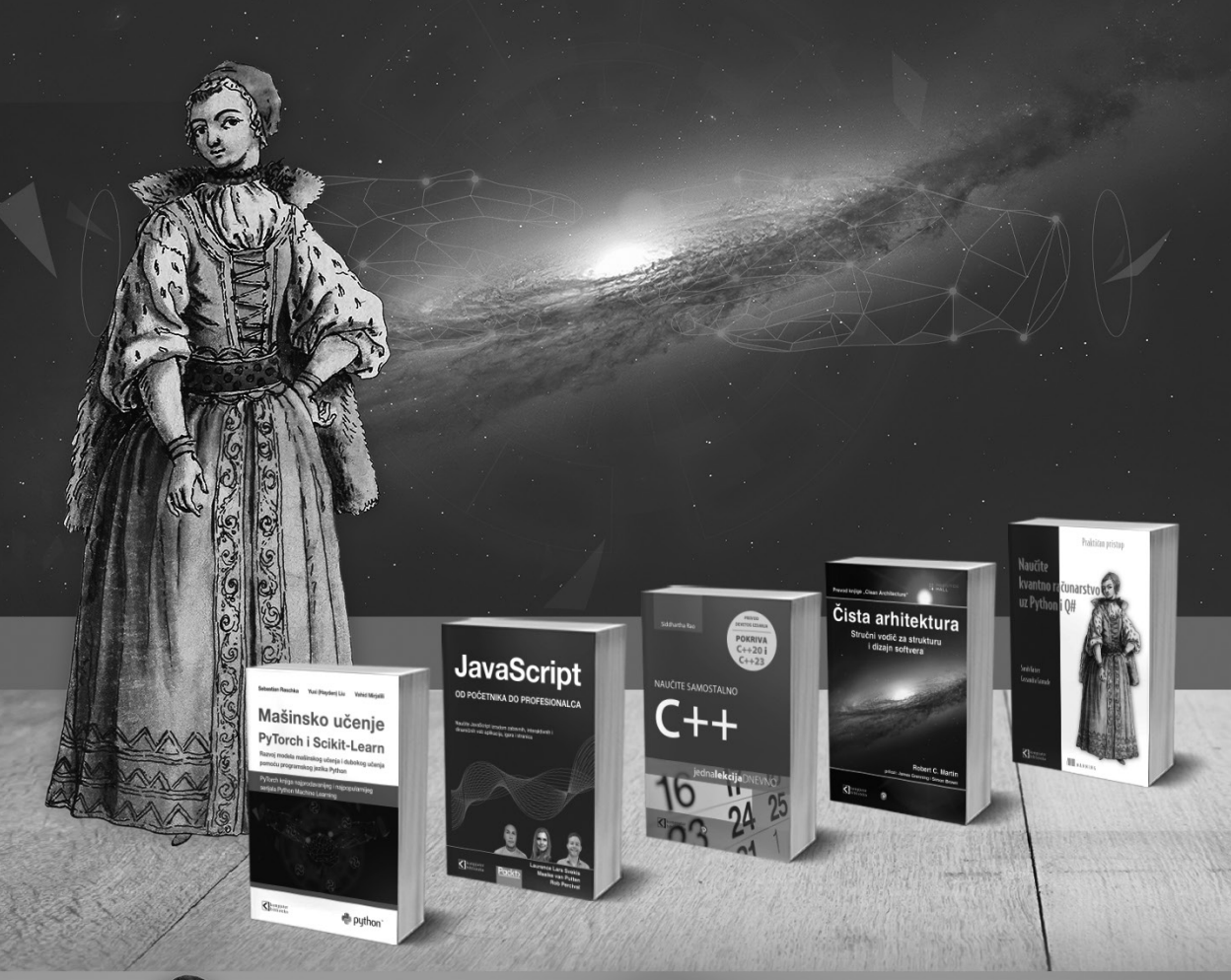
Povratne informacije od čitalaca su uvek dobrodošle.

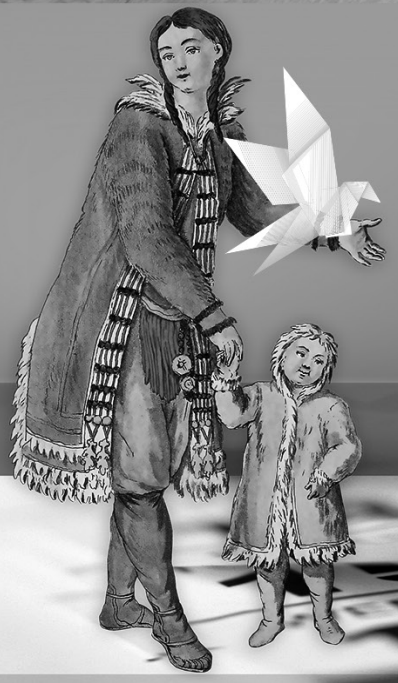
Opšte povratne informacije: Ako imate pitanja o bilo kom aspektu ove knjige, kontaktirajte nas putem e-pošte na: `customercare@packtpub.com`. U naslovu poruke navedite naziv knjige.

Greške u tekstu: Iako smo uložili trud da osiguramo tačnost našeg sadržaja, greške se ponekad mogu desiti. Ako pronađete grešku u ovoj knjizi, molimo vas da nam je prijavite na: `www.packtpub.com/support/errata`.

Piraterija: Ako naiđete na bilo koje nelegalne kopije naših materijala na internetu, bili bismo vam zahvalni ako biste nam dostavili adresu lokacije ili naziv veb sajta. Molimo vas da nas kontaktirate na `copyright@packt.com` sa vezom ka materijalu.

Ako ste zainteresovani da postanete autor: Ako ste stručnjak u određenoj oblasti i zainteresovani ste za pisanje ili doprinos knjizi, posetite `authors.packtpub.com`.





Postanite član Kompjuter biblioteke

Kupovinom jedne naše knjige stekli ste pravo da postanete član Kompjuter biblioteke. Kao član možete da kupujete knjige u pretplati sa 40% popusta i učestvujete u akcijama kada ostvarujete popuste na sva naša izdanja. Potrebno je samo da se prijavite preko formulara na našem sajtu.

Link za prijavu: kombib.rs/kblista.php

Skenirajte QR kod
registrujte knjigu
i osvojite nagradu

Deo 1

Uvod u Bash skriptovanje

Ovaj deo će vam omogućiti da uspostavite čvrste osnove u Bash skriptovanju, posebno prilagođene za testiranje neprobojnosti. Počinje se sa postavljanjem odgovarajućeg radnog okruženja i konfigurisanjem Bash ljuske, a zatim se prelazi na osnovne tehnike upravljanja datotekama i direktorijumima potrebnih za procene bezbednosti. Usmerićete se na savladavanje ključnih programerskih koncepata, uključujući promenljive, uslovne izraze, petlje i nizove, pre nego što se pozabavite prepoznavanjem obrazaca pomoću regularnih izraza – što je ključna veština za parsiranje izlaza bezbednosnih alata. Zatim se prelazi na kreiranje funkcija i organizaciju skriptova, osiguravajući vam da možete izgraditi održive i profesionalne bezbednosne alate. Usmerićete se na osnovne mrežne pojmove i naučićete kako Bash sarađuje sa mrežnim uslugama i protokolima. Ovaj deo se završava tehnikama paralelne obrade, omogućavajući vam da razvijate efikasne skriptove koji mogu istovremeno da obrađuju više zadataka – što je ključna sposobnost za velike procene bezbednosti. Na kraju *prvog dela*, imaćete sve osnovne veštine potrebne za početak pisanja sofisticiranih Bash skriptova fokusiranih na bezbednost.

Ovaj deo sadrži sledeća poglavlja:

- *Poglavlje 1, Bash komandna linija i okruženje za testiranje*
- *Poglavlje 2, Upravljanje datotekama i direktorijumima*
- *Poglavlje 3, Promenljive, uslovni izrazi i nizovi*
- *Poglavlje 4, Regularni izrazi*
- *Poglavlje 5, Funkcije i organizacija skriptova*
- *Pogljavle 6, Bash mrežno upravljanje*
- *Poglavlje 7, Paralelna obrada podataka*

1

Bash komandna linija i okruženje za testiranje

U ovom uvodnom poglavlju, započećete svoje putovanje u svet Bash skriptovanja za **testere neprobojnosti**. Steći ćete jasno razumevanje šta je Bash, zašto je ključan za **testiranje neprobojnosti** i kako da podesite svoje skriptno okruženje. Kroz praktične primere i objašnjenja, postavićete temelje za napredno Bash skriptovanje u kontekstu sajber bezbednosti.

Bash nije samo komandni interpreter – to je alat za automatizaciju složenih i ponavljajućih zadataka koji su česti u oblasti sajber bezbednosti. U rukama neiskusnih korisnika, Bash je poput čekića. Čini se težak, previše složen i neprijatan. Međutim, u rukama onih koji znaju kako da iskoriste njegove mogućnosti, Bash postaje alat visoke preciznosti – skalpel koji omogućava precizno sečenje podataka, kao kod hirurga, i automatizaciju metodologije testiranja neprobojnosti sa efikasnošću inženjera robotike.

U ovom poglavlju, obrađujemo sledeće ključne teme:

- Uvod u Bash
- Postavljanje radnog okruženja
- Konfiguracija Bash okruženja za testiranje neprobojnosti
- Podešavanje neophodnih alata za testiranje neprobojnosti

Tehnički zahtevi

Za praćenje vežbi u ovom poglavlju, potrebno je Linux okruženje. Ova knjiga podrazumeva da imate dovoljno znanja da instalirate operativni sistem i da ste upoznati sa instalacijom i konfiguracijom okruženja virtuelnih mašina. Ako vam je potrebna pomoć pri postavljanju vašeg laboratorijskog okruženja, korisni resursi uključuju: VirtualBox priručnik na internetu (*Oracle VM VirtualBox User Manual* `https://download.virtualbox.org/virtualbox/UserManual.pdf`) i i nekoliko YouTube video materijala (*VirtualBox YouTube* `https://www.youtube.com/results?search_query=virtualbox`). Srećom, postoji mnogo načina za konfigurisanje okruženja za učenje Bash interpretera.

Svi primeri će biti prikazani koristeći Kali Linux. Međutim, bilo koje Linux ili macOS okruženje će takođe raditi.

„Kali Linux je otvorenog koda, Debian-bazirana Linux distribucija razvijena za različite zadatke sajber bezbednosti, kao što su testiranje neprobojnosti, istraživanje bezbednosti, forenzička analiza računara i obrnuti inženjering."
(Kali Linux, `https://www.kali.org/`*)*

Preporučujem da koristite novu instalaciju Kali Linux virtuelne mašine kako biste pratili vežbe ili izvodili testiranje neprobojnosti. Tokom ove knjige i sopstvenog testiranja neprobojnosti, instaliraćete veliki broj alata i njihovih zavisnosti. Uobičajena praksa je da alati automatski instaliraju zavisne pakete, što je poznato kao *pakleni krug zavisnosti*. To može dovesti do oštećenja vašeg sistema ako nije pravilno podešeno, pa je najbolje koristiti izolovano okruženje kako biste izbegli potencijalne zlonamerne programe.

Kali Linux nudi različita rešenja za instalaciju. Dostupne su instalacione datoteke, virtuelne mašine, slike za okruženja u oblaku i **Windows Subsystem for Linux (WSL)** paketi.

Kali Linux možete preuzeti sa adrese
`https://www.kali.org/get-kali/#kali-platforms`.

Sve komande korišćene u ovom poglavlju mogu se pronaći u GitHub spremištu ove knjige na adresi : `https://github.com/PacktPublishing/Bash-Shell-Scripting-for-Pentesters/tree/main/Chapter01`.

Uvod u Bash

Bash, poznat i kao *Bourne Again Shell*, je komandni interpreter i skript jezik. Razvio ga je Brajan Foks 1989. godine kao besplatnu softversku zamenu za Bourne komandni interpreter, koji je bio vlasnički softver. *(Bash – GNU Project – Free Software Foundation,* `https://www.gnu.org/software/bash/`*).* Bash je najčešće korišćeni komandni interpreter u Linux okruženju. Takođe, omogućava korisnicima da kombinuju više komandi u skriptove koji se mogu pokrenuti unosom samo jedne komande.

Kada otvorite terminal na Linux sistemu i unesete komandu, Bash upravlja interakcijama sa operativnim sistemom, kao i sa pokretanjem izvršnih datoteka i skriptova. Bash komande i Linux izvršne datoteke imaju specifičan odnos, omogućavajući unapređivanje funkcionalnosti i poboljšanje efikasnosti rada. Bash služi kao sloj interakcije između korisnika i Linux jezgra, centralnog dela operativnog sistema. Korisnici unose komande u terminal, koje Bash zatim analizira i izvršava u okviru sistema. Linux izvršne datoteke su programi koji obavljaju različite zadatke. Oni su najčešće binarne datoteke, često napisane u programskim jezicima poput C ili C++, koje su kompajlirane da rade efikasno na Linux sistemima. Kada korisnik unese komandu u Bash interpreteru, često zapravo poziva jedan od ovih izvršnih programa kako bi obavio određeni zadatak.

Bash poseduje sledeće karakteristike:

- **Izvršavanje komandi sa argumentima**: Komande mogu biti izvršne datoteke, ugrađene interpreterske komande i skriptovi.
- **Dovršavanje komandi**: Funkcionalnost koja korisniku pomaže automatskim dovršavanjem delimično otkucanih komandi ili imena datoteka pritiskom na taster *Tab*.
- **Istorija komandi**: Omogućava brzo ponovno korišćenje prethodno unetih komandi u terminalu.
- **Upravljanje procesima**: Slanje komandi u pozadinu i vraćanje u prvi plan.
- **Funkcije i pseudonimi**: Funkcije grupišu povezani kod pod jednim nazivom koji može biti pozvan kada je potreban. Pseudonim omogućava korisniku da skrati složene komande na jednu reč.
- **Nizovi**: Omogućavaju skladištenje elemenata u listu koju kasnije možemo dohvatiti i obraditi.
- **Proširivanje komandi i zagrada**: Proširivanje komandi koristi rezultat jedne komande kao ulaz za drugu. Proširivanje zagrada omogućava generisanje niski.
- **Protočne strukture i preusmeravanje**: Izlaz jedne komande može se koristiti kao ulaz druge komande.
- **Promenljive okruženja**: Dinamičke vrednosti se dodeljuju imenovanim oznakama, koje se često koriste za predstavljanje sistemskih konfiguracija ili skladištenje informacija o okruženju.
- **Navigacija kroz sistem datoteka**: Bash omogućava komande za promenu direktorijuma, prikaz trenutnog direktorijuma i pretragu datoteka i direktorijuma.
- **Pomoć**: Komanda `man`, skraćeno od *manual*, pruža korisniku informacije i primere kako se koriste komande.

Bash skriptovanje je jedna od najvažnijih veština koje sam naučio tokom svoje karijere u testiranju neprobojnosti i koristim je svakodnevno. Kada razvijate aplikacije, u nekom trenutku ćete morati koristiti i drugi skript jezik, poput Python jezika. Međutim, u većini slučajeva, sve što možete raditi u terminalu može se uraditi i u Bash interpreteru, koji orkestrira unos i izlaz podataka i obrađuje podatke iz više alata. Bash je toliko duboko integrisan sa Linux operativnim sistemom da ima smisla naučiti ga pre nego što se pređe na druge skript jezike poput jezika kao što su Python ili Ruby. Iako poznajem više programskih i skript jezika, Bash najčešće koristim zbog njegove čvrste integracije sa okruženjem i mogućnosti brzog dobijanja rezultata čak i pomoću jedne linije ili jedne reči.

Tokom svog rada kao testera neprobojnosti, svakodnevno koristim Bash za analizu podataka ili automatizaciju povezivanja više alata. Kada mi klijent dostavi podatke o opsegu testiranja, često moram da kopiram listu IP adresa ili imena računara iz dokumenta o *pravilima angažovanja*, e-pošte ili Excel tabele i umetnem ih u tekstualnu datoteku. Nemoguće je da podaci ne sadrže neželjene karaktere ili da nisu pravilno formatirani za listu ciljeva skeniranja. Uz pomoć Bash interpretera, mogu očistiti podatke i formatirati ih prema potrebi za testiranje, koristeći samo jednu jednostavnu liniju koda unetu u terminal.

Alati za testiranje neprobojnosti prihvataju podatke u različitim formatima i generišu rezultate skeniranja ili podatke u uobičajenim formatima kao što su XML, JSON ili običan tekst. Izlaz u običnom tekstu može biti formatiran sa višestrukim razmacima, tabulatorima ili njihovom kombinacijom. Sadržaj izvorne datoteke preusmeravam kroz Bash protočnu strukturu kako bih parsirao, očistio, ponovo formatirao i sortirao podatke. Koristim kombinaciju Bash komandi da obavim ove zadatke između izlaza jedne komande i ulaza druge u okviru automatizovanog procesa. Bash je zaista nezamenljiv alat u arsenalu svakog testera neprobojnosti.

Evo nekoliko uobičajenih primena Bash skriptovanja u mom radu testiranja neprobojnosti:

- **Automatizovano mrežno skeniranje**: Često obrađujem izlaz Masscan alata, brzog TCP skenera, i prosleđujem ga Nmap alatu za detaljno otkrivanje servisa i skeniranja skriptova.
- **Razbijanje lozinki**: Koristim Bash skript za složene funkcije razbijanja lozinki koje se odnose na Microsoft LM i NTLM heševe, kao i za formatiranje izlaza Hashcat alata radi unosa u alat za izveštavanje.
- **Pretraga teksta**: Pretražujem IP adrese ili druge detalje unutar tekstualnih podataka.
- **Automatizacija opsega testiranja**: Koristim alate za identifikaciju poddomena sa Bash skriptom kako bih osigurao da otkriveni poddomeni budu u skladu sa pravilima testiranja neprobojnosti.
- **Formatiranje podataka**: Upotrebljavam Bash za parsiranje i preformatiranje rezultata Nuclei skeniranja, kako bih izdvojio poddomene i veb aplikacije iz TLS sertifikata i preformatirao podatke za zaobilaženje **mreža za dostavljanje sadržaja** (**CDN**) i **mrežnog zaštitnog zida za veb aplikacije** (**WAF**) kako bih direktno skenirao cilj.
- **Pretraga i sortiranje Nmap izveštaja**: Nakon skeniranja stotina ili čak hiljada IP adresa, koristim Bash za parsiranje `gnmap` datoteka i kreiranje tekstualnih datoteka sa ciljevima organizovanim prema TCP ili UDP portovima za dalja ciljana skeniranja. Na primer, sve IP adrese SMB servera ili HTTP servera izdvajam u posebne datoteke nazvane `smb.txt` i `http.txt`.
- **Sortiranje i uklanjanje dupliranja podataka**: Sortiram jedinstvene IP adrese u datoteku radi uklanjanja duplikata.
- **Konverzija podataka**: Konvertujem imena i prezimena u različite formate za napade raspršivanja lozinki. Ako putem **obaveštajnih podataka iz otvorenih izvora** (**OSINT**) prikupim listu imena zaposlenih, analiziram bilo kakve obrasce koji mogu ukazati na format njihovih Active Directory imena (npr. `i.prezime` ili `ime.prezime`) i koristim Bash za pravilno formatiranje.
- **Filtriranje podataka**: Povremeno moram ukloniti boje sa kodova u izlaznim datotekama zapisa pre nego što ih koristim u izveštavanju, jer sam zaboravio da uključim opciju za onemogućavanje boja ili alat jednostavno ne nudi ovu opciju. Ne želim da u izveštaju klijentima prikazujem podatke sa bojama koje otežavaju čitanje.
- **Iteracija kroz podatke**: Koristim Bash `for` i `while` petlje za iteraciju kroz datoteku i pokretanje komandi nad svakom linijom. Dobar primer za to je kada koristim alat koji može da skenira samo jedan računar odjednom i nema opciju za obradu više ciljeva istovremeno.

Siguran sam da će vam učenje Bash skriptovanja pomoći da budete efikasniji s vremenom i efektivniji u svom poslu. Kada možete automatizovati dugotrajne i dosadne zadatke pomoću Bash interpretera, oslobađate svoje vreme za važnije stvari. Zar ne bi bilo sjajno imati više vremena za učenje ili istraživanje umesto da ga trošite na ručne zadatke koji se lako mogu automatizovati?

Sada kada imamo osnovno razumevanje Bash interpretera i njegove korisnosti u testiranju neprobojnosti, hajde da istražimo kako da postavimo laboratorijsko okruženje u kojem možete bezbedno učiti i pratiti vežbe. U sledećem odeljku ćemo objasniti kako da podesite svoje laboratorijsko okruženje tako da možete pratiti moj rad korak po korak.

Podešavanje laboratorije

Bash nije jedini komandni interpreter za Linux i Unix sisteme, ali je najčešće korišćen. Mnogi drugi interpreteri su pod uticajem Bash interpretera. Na primer, na macOS i Kali Linux sistemima možete naići na Zsh.

Možda se pitate zašto se ova knjiga fokusira na Bash, uprkos tome što su neki operativni sistemi prešli na Zsh. Iako su macOS i Kali prešli na Zsh za nove korisničke naloge, Bash je i dalje instaliran. Većina koda napisanog za Bash će raditi i u Zsh interpreteru uz nekoliko manjih izmena. Možete koristiti **šebeng** liniju u svojim skriptovima kako biste osigurali da Bash interpreter izvrši vaš skript na sistemima gde je instalirano više komandnih interpretera. Tokom bezbednosnih procena, vrlo je verovatno da ćete naići na Linux servere gde je Bash podrazumevani komandni interpreter. Zbog toga je za testera neprobojnosti ključno da razume kako da koristi Bash za iskorišćavanje aplikacija, povećanje privilegija i bočno kretanje kroz sistem.

Srećom, postoji mnogo načina da besplatno pristupite Bash interpreteru. Ovaj odeljak će istražiti različite načine kako da pokrenete Bash interpreter u idealnim uslovima kako biste mogli učiti i koristiti Bash za testiranje neprobojnosti. Takođe ćemo istražiti ranjiva laboratorijska okruženja u kojima možete bezbedno vežbati rad sa Bash interpreterom i alatima za testiranje neprobojnosti.

Virtuelne mašine su najbolji način za praćenje vežbi iz ove knjige, ali i za obavljanje testiranja neprobojnosti. Možda ćete biti u iskušenju da instalirate alate za testiranje neprobojnosti i kod za iskorišćavanje na isti sistem koji koristite za posao ili lične aktivnosti. Međutim, instaliranje softverskih zavisnosti za razne alate može lako oštetiti vaš sistem. Uvek postoji rizik da alati za hakovanje sadrže zlonamerni softver i da zaraze sistem koji koristite svakodnevno za slanje e-pošte ili pregledanje interneta. Virtuelna mašina obezbeđuje pogodno okruženje za testiranje sa svim potrebnim alatima, omogućavajući vam brzo resetovanje ili zamenu testnog okruženja. U svim demonstracijama koristim Kali Linux. Želimo da izbegnemo instaliranje alata za testiranje i koda za iskorišćavanje na isti sistem koji koristimo svakodnevno. Najbolje je koristiti čisto testno okruženje kako bismo izbegli probleme sa softverskim zavisnostima. Kali Linux omogućava lako instaliranje svih potrebnih softverskih paketa za testiranje neprobojnosti.

Virtuelne mašine

Korišćenje **virtuelne mašine** je najbolja metoda. Tokom testiranja neprobojnosti verovatno ćete instalirati veliki broj alata i koda za iskorišćavanje. Takođe ćete čuvati osetljive podatke

o svojim klijentima ili ciljevima testiranja. Virtuelna mašina pruža pogodno okruženje koje možete snimiti, vratiti ili obrisati i zameniti nakon završetka procene bezbednosti.

Postoje brojna besplatna i plaćena rešenja za virtuelizaciju koja mogu zadovoljiti različite potrebe:

- Oracle VirtualBox je besplatan x86 softver za virtuelizaciju. Dostupan je za Windows, macOS (Intel čipset) i Linux. VirtualBox je jednostavan za korišćenje, što ga čini popularnim izborom i za početnike i za profesionalce. Podržava širok spektar gostujućih operativnih sistema i nudi funkcije kao što su snimci stanja, besprekidni režim i deljeni direktorijumi.
- VMware nudi besplatnu verziju svog softvera za virtuelizaciju pod nazivom VMware Workstation Player za nekomercijalnu upotrebu. Kompatibilan je sa Windows i Linux sistemima. Workstation Player je jednostavan za upotrebu, podržava VMDK format virtuelnog diska kompanije CMware i kompatibilan je sa virtuelnim mašinama kreiranim u drugim VMware proizvodima.
- Microsoft Hyper-V je besplatan i dostupan u Windows 10 Pro, Enterprise i Education verzijama. Iako se češće koristi u serverskim okruženjima, Hyper-V može biti dobra opcija i za desktop virtuelizaciju na Windows računarima.

Savet

Za one koji koriste macOS sa Apple procesorom, opcije za virtuelizaciju su UTM, Parallels i VMware Fusion. UTM je jedina besplatna opcija.

Docker kontejneri

Docker kontejneri predstavljaju lakšu alternativu u odnosu na virtuelne mašine. Docker omogućava pokretanje kontejnera na Windows, Linux i macOS sistemima. Kontejneri su efikasniji na hardveru slabijih performansi u poređenju sa virtuelnim mašinama, jer koriste jezgro računara i ne moraju da emuliraju hardver kao što to rade tradicionalni softveri za virtuelizaciju.

Međutim, zbog toga što Docker koristi jezgro računara, ograničeni ste na pokretanje kontejnera koji koriste isti operativni sistem kao i računar. Docker Desktop je alternativno rešenje koje koristi virtuelnu mašinu kako bi omogućilo pokretanje kontejnera sa drugačijim operativnim sistemom od onog na računaru.

Na osnovu mog iskustva, postoje i pozitivne i negativne strane korišćenja Docker kontejnera.

Docker je lakši i predstavlja dobru alternativu tradicionalnim softverima za virtuelizaciju kada raspolažete slabijim hardverom. Minimalni hardverski resursi koje bih dodelio virtuelnoj mašini koja pokreće Kali Linux su 4 GB RAM memorije i 40 GB prostora na disku. Nećete uvek koristiti svih 4 GB RAM memorije i 40 GB prostora, ali ste ograničeni na te vrednosti osim ako ne isključite virtuelnu mašinu, prilagodite RAM i proširite disk. Docker kontejner se pokreće kao prirodni proces računara (izuzev Docker Desktop aplikacije), pa koristi samo onoliko memorije i prostora na disku koliko je potrebno za rad kontejnera.

Na Linux računaru možete direktno povezati kontejner sa mrežom računara i otvarati/zatvarati portove po potrebi, pod uslovom da uključite odgovarajuće argumente u komandnoj liniji. Ovo omogućava dinamično otvaranje portova servera za slušanje na mrežnom adapteru računara bez potrebe za zaustavljanjem i ponovnim pokretanjem kontejnera. Takođe možete povezati kontejner sa USB ili serijskim portom radi interakcije sa hardverskim uređajima. Ovu opciju ponekad koristim kada mi je potrebno da pokrenem staru Python 2 aplikaciju za testiranje neprobojnosti koja komunicira sa USB ili serijskim uređajem za radio-frekvencijsko i hardversko hakovanje.

Kada koristite Docker Desktop, koristi se NAT za povezivanje mrežnih portova kontejnera sa mrežom računara, što znači da kontejner mora biti zaustavljen i ponovo pokrenut ako želite da zatvorite ili otvorite dodatne portove. Takođe, sa Docker Desktop aplikacijom nije moguće povezati kontejner sa hardverskim uređajima. Ovo može biti frustrirajuće ako ste već konfigurisali aplikaciju i njene zavisnosti u kontejneru, ali izgubite svoj rad jer morate uništiti kontejner i pokrenuti novu instancu samo da biste otvorili još jedan TCP port za obrnuti slušalac ili serversku aplikaciju.

Moj lični izbor je da koristim Docker isključivo na Linux računaru, a koristim ga za tri specifične potrebe u testiranju neprobojnosti:

- Docker omogućava lako izdvajanje starih aplikacija i izbegavanje problema sa zavisnostima. Postoje zvanični Docker kontejneri za sve Python 2 i 3 verzije.
- Umesto da gubim vreme rešavajući zavisnosti, koristim Docker za pokretanje aplikacija koje ne mogu lako da instaliram na svoj sistem. Na primer, određeni alat za hakovanje dostupan je u Kali spremištu softvera, ali ne i u Ubuntu spremištu. Mogu da kreiram minimalni Kali kontejner koji koristi samo resurse potrebne za pokretanje aplikacije i postavim pseudonim u svojoj `~/.bashrc` datoteci kako bih smanjio dugačku `docker run` komandu na jednu reč koju mogu jednostavno uneti u terminal. Ovo je mnogo brža i lakša opcija u poređenju sa teškom virtuelnom mašinom kada samo želim da pokrenem jednu aplikaciju koja inače ne bi mogla ili bi teško radila na mom računaru.
- Kada želim da testiram ili kreiram alat za iskorišćavanje nedavno objavljene ranjive veb aplikacije, često mogu pronaći Docker kontejner koji mi omogućava da odmah pokrenem ranjivu aplikaciju bez gubljenja vremena na instalaciju i konfiguraciju.

Docker kontejneri su idealni za specifične slučajeve upotrebe, ali su i dalje manje poželjni od virtuelnih mašina. U nastavku ćemo istražiti korišćenje live USB sistema kao alternativu virtuelnim mašinama i kontejnerima.

USB za pokretanje sistema

USB za pokretanje sistema je operativni sistem zapisan na USB disk na način koji ga čini sposobnim za pokretanje sistema. USB za pokretanje sistema je dobra opcija kada vaš računar nema dovoljno hardverskih resursa za pokretanje virtuelne mašine. Možete koristiti softver za kreiranje sistemskih slika da biste zapisali Linux ISO datoteku na USB i pokrenuli Linux operativni sistem direktno sa USB diska. Kada završite rad u Linux operativnom sistemu, jednostavno ponovo pokrenite računar i uklonite USB disk, čime se sistem vraća na prethodno instalirani operativni sistem. Neke Linux distribucije omogućavaju kreiranje trajnog skladišta na USB disku, što znači da nećete izgubiti promene kada ponovo pokrenete računar.

Opšti koraci za pokretanje Linux distribucije sa USB diska za pokretanje sistema:

1. Preuzmite ISO sliku. Neke popularne Linux distribucije za testiranje neprobojnosti uključuju Kali, Parrot Security OS i BlackArch.
2. Kreirajte USB disk za pokretanje sistema. Uobičajeni alati za ovu svrhu su Rufus, balenaEtcher i Linux komanda `dd`.
3. Konfigurišite trajnost (opciono). Ovo obično podrazumeva kreiranje posebne particije na USB disku i podešavanje pokretača operativnog sistema da prepozna i koristi tu particiju. Dokumentovane korake za kreiranje USB Kali sistema možete pronaći na: `https://www.kali.org/docs/usb/usb-persistence/`.

Razmatranja i nedostaci korišćenja USB diska za pokretanje sistema:

- USB skladište je obično mnogo sporije od pokretanja sistema direktno sa SSD diska. Ako koristite USB disk za pokretanje sistema, obavezno koristite standard USB 3.0 ili 3.1 za najbolje performanse.
- Uvek preuzmite ISO sliku sa zvaničnih izvora i proverite njenu kontrolnu sumu pre nego što je koristite.
- Ako planirate da koristite USB za pokretanje sistema za ozbiljniji rad, obavezno koristite šifrovanu trajnost kako biste zaštitili poverljive podatke od neovlašćenog pristupa.

Sada, hajde da pređemo na sisteme zasnovane na oblaku.

Sistemi zasnovani na oblaku

Mnoge platforme zasnovane na oblaku nude besplatne nivoe za pristup Linux sistemima sa dovoljno resursa za manje zahteve. Dobavljači oblaka koji nude besplatne nivoe uključuju: **Google Cloud Platform** (**GCP**), Microsoft Azure i Amazon EC2. Međutim, imajte na umu da besplatni nivo obično ne nudi dovoljno RAM memorije za ozbiljniju upotrebu i neće biti pogodan za pokretanje Kali Linux slike.

Kali Linux nudi dokumentaciju i slike na platformama zasnovane na oblaku kao što su AWS, Digital Ocean, Linode i Azure (`https://www.kali.org/docs/cloud/`).Imam iskustva sa klijentima koji su konfigurisali Kali u oblaku za procenu bezbednosti u cloud okruženjima ili su se povezivali putem VPN mreže sa svojom internom mrežnom infrastrukturom kako bi olakšali testiranje neprobojnosti interne mreže. Ako je interna mreža klijenta već povezana sa dobavljačem oblaka preko VPN mreže, relativno je jednostavno pokrenuti Kali instancu i kreirati pravilo mrežnog zaštitnog zida koje omogućava SSH pristup sa moje IP adrese. Sada kada smo istražili opcije za pokretanje sistema za testiranje neprobojnosti sa Bash interpreterom, hajde da pogledamo neke ranjive sisteme koje možemo koristiti za vežbanje u laboratorijskom okruženju.

Ranjivi ciljevi laboratorije

Dok budete pratili kasnija poglavlja koja se odnose na metodologiju testiranja neprobojnosti, biće korisno da imate pristup ranjivim ciljevima dok izvršavate komande i razvijate

Bash skriptove. Postoji nekoliko odličnih izvora ranjivih ciljeva koje možete koristiti za vežbanje u svojoj laboratoriji.

Metasploitable 2 je ranjiva virtuelna mašina koju pruža Rapid7. Dizajnirana je da prikaže mogućnosti Metasploit Framework alata. Metasploitable 2 takođe predstavlja dobar početnički izazov za razvijanje metodologije hakovanja i učenje Bash interpretera za testiranje neprobojnosti. Projekat zahteva skromne resurse za pokretanje virtuelne mašine i uključuje dokumentaciju o njenim ranjivostima (*Metasploitable 2 | Metasploit Documentation*, `https://docs.rapid7.com/metasploit/metasploitable-2/`).

Game of Active Directory (GOAD) je takođe opcija

„GOAD je laboratorijski projekat za testiranje neprobojnosti Active Directory servisa. Cilj ove laboratorije je da testerima neprobojnosti pruži ranjivo Active Directory okruženje, spremno za upotrebu, kako bi mogli da vežbaju uobičajene tehnike napada." (Game of Active Directory – Orange-CyberDefense, `https://github.com/Orange-Cyberdefense/GOAD`).

Vredno je napomenuti da je GOAD besplatan za korišćenje i da koristi besplatne Microsoft Windows licence koje su aktivirane na 180 dana. GOAD je najbolji resurs koji sam pronašao za vežbanje hakovanja u internim Active Directory mrežnim okruženjima.

MayFly je kreator GOAD projekta. Njihov veb sajt sadrži mnoštvo članaka o tome kako da postavite GOAD na različitim hipervizorima virtuelnih mašina, kao i laboratorijske vodiče za korišćenje pentesting alata u napadima na Active Directory.

Savet

MayFly je takođe objavio sveobuhvatnu mapu uma za testiranje neprobojnosti Active Directory servisa. Iako imam višegodišnje iskustvo u hakovanju Active Directory servisa, i dalje se dešava da mi ponestane ideja šta sledeće da testiram, pa se oslanjam na ovu mapu uma kada ne znam šta da radim ili želim da budem siguran da nisam propustio nijedan aspekt testiranja. Ova mapa uma je takođe resurs broj jedan koji preporučujem mlađim testerima neprobojnosti koji uče tehnike i alate za hakovanje Active Directory servisa (više detalja možete pronaći na `https://orange-cyberdefense.github.io/ocd-mindmaps/img/pentest_ad_dark_2022_11.svg`).

Ako želite da vežbate svoje Bash skriptove, alate i metodologiju na veb aplikacijama, **OWASP Juice Shop** je odličan resurs.

„OWASP Juice Shop je verovatno najmodernija i najsofisticiranija nebezbedna veb aplikacija! Može se koristiti za obuke iz bezbednosti, demonstracije svesti o bezbednosti, CTF izazove i kao testno okruženje za sigurnosne alate! Juice Shop obuhvata ranjivosti iz celokupne OWASP Top Ten liste (`https://owasp.org/www-project-top-ten/`), *zajedno sa mnogim drugim bezbednosnim propustima koji se mogu naći u stvarnim aplikacijama!" — (OWASP Juice Shop – OWASP Foundation*, `https://owasp.org/www-project-juice-shop/`).

Starija, ali i dalje vrlo relevantna ranjiva veb aplikacija je Mutillidae II.

„OWASP Mutillidae II je besplatna, otvorenog koda, namerno ranjiva veb aplikacija koja služi kao meta za obuku iz oblasti veb bezbednosti. Sa desetinama ranjivosti i ugrađenim sugestijama koje pomažu korisnicima, ovo je jednostavno okruženje za hakovanje veb aplikacija, dizajnirano za laboratorije, entuzijaste sajber bezbednosti, učionice, CTF takmičenja i alate za procenu ranjivosti." — (*OWASP Mutillidae II – OWASP Foundation*, `https://owasp.org/www-project-mutillidae-ii/`).

Jedna od stvari koje volim kod Mutillidae je to što sadrži ugrađene sugestije, uputstva i video uputstva u okviru samog sadržaja. Mutillidae je bio resurs koji sam koristio pre mnogo godina kada sam bio početnik u testiranju neprobojnosti, kako bih naučio tehnike testiranja veb aplikacija. Razlika između aplikacija Juice Shop i Mutillidae je u tome što je Juice Shop moderna veb aplikacija koja koristi JavaScript okvire, dok je Mutillidae tradicionalnija veb aplikacija. Dok Juice Shop sadrži tabelu rezultata i možete pronaći vodiče trećih strana na internetu, Mutillidae ima veliku količinu ugrađenog teksta i video materijala dizajniranih za obuku korisnika.

Svet sajber bezbednosti stalno se menja, a nove ranjivosti se otkrivaju redovno. Laboratorijsko okruženje je idealno mesto za istraživanje i razvoj, omogućavajući vam da bezbedno eksperimentišete sa ovim ranjivostima. Takođe, možete doprineti zajednici sajber bezbednosti otkrivanjem novih ranjivosti ili poboljšanjem postojećih metoda testiranja neprobojnosti.

Sada kada smo istražili ranjive mete za vašu laboratoriju za testiranje neprobojnosti, sledeći korak je prilagođavanje vašeg Bash okruženja kako bi odgovarao vašim potrebama i ličnom stilu rada.

Konfigurisanje vašeg hakerskog okruženja

Ako koristite Kali Linux ili macOS, imajte na umu da vaš terminal podrazumevano koristi Zsh umesto Bash. Zsh ima više funkcionalnosti (napredna automatska dopuna, podrška za teme), ali Bash je šire rasprostranjen i standardan. Bash postoji od kraja 80-ih i smatra se veteranom u svetu komandnih interpretera. Dugo je bio podrazumevani interpreter na većini Linux distribucija i macOS operativnom sistemu (do verzije Catalina, gde je Zsh postao novi standard). Njegova dugovečnost znači da je izuzetno stabilan i dobro podržan.

S druge strane, Zsh je noviji i donosi unapređenja u interaktivnoj upotrebi i snažnijim mogućnostima skriptovanja.

Unesite sledeću komandu u terminal da biste proverili koji interpreter koristite: `echo $SHELL`. Gotovo sav kod prikazan u ovoj knjizi radi u oba komandna interpretera – Bash i Zsh, osim ako nije drugačije naznačeno. Tokom svakodnevnih aktivnosti testiranja neprobojnosti, retko se primećuju razlike između ova dva okruženja. Ako želite da promenite okruženje sa Zsh na Bash, unesite sledeću komandu `chsh -s /bin/bash`. Zatim se odjavite i ponovo prijavite kako bi promena stupila na snagu.

Konfiguracione datoteke za Bash se nalaze u korisničkom direktorijumu, `/home/username`. Pošto sva imena ovih datoteka počinju sa tačkom, nazivaju se *dot datoteke*. Sledeće konfiguracione datoteke se koriste za podešavanje Bash okruženja:

- `~/.bash_profile` – Ova datoteka se izvršava na početku interaktivne prijave i koristi se za inicijalizaciju korisničkog okruženja. Interaktivna prijava podrazumeva prijavljivanje putem komandne linije u tekstualnom terminalu, kao što je SSH sesija.

- `~/.bashrc` – Ova datoteka se koristi za konfiguraciju terminala kada se prijavite preko **grafičkog korisničkog interfejsa** (**GUI**). Sadrži podešavanja kao što su pseudonimi, funkcije, prilagođavanje upita i promenljive okruženja.
- `~/.bash_logout` – Ova datoteka se izvršava kada se sesija završi. Koristi se za izvršavanje zadataka čišćenja okruženja prilikom odjavljivanja.

> **Savet**
>
> Ako ne razumete svrhu znaka tilda (~)i kose crte (/) ispred imena dot datoteka, znak tilda (~) predstavlja korisnički direktorijum. Putanja `~/.bashrc` je ekvivalentna putanji `/home/username/.bashrc`. Ovaj koncept će biti obrađen u *poglavlju 2*.

Najčešće izmene koje ćete želeti da napravite u svojoj `~/.bashrc` datoteci uključuju dodavanje pseudonima, funkcija i prilagođavanje komandnog upita. Pseudonimi su odličan način da skratite duge ili složene komande na jednu reč. Funkcije su složenije – možete ih zamisliti kao kratak skript koji možete uključiti u svoju konfiguraciju komandnog okruženja i pozivati po imenu iz terminala. Funkcije će biti detaljnije obrađene u *poglavlju 5*.

Evo primera pseudonima iz moje `~/.bashrc` datoteke koji koristim za pretragu IP adresa u tekstu:

```
alias grepip="grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'"
```

Možete videti kako bi ovu komandu bilo teško zapamtiti, pa je korisno kreirati pseudonim za svaku složenu komandu koju često koristite.

> **Važna napomena**
>
> Kada napravite izmene u svojim Bash konfiguracionim datotekama, morate se odjaviti i ponovo prijaviti ili pokrenuti datoteku pomoću `source` komande da bi promene bile prihvaćene.

Unesite sledeću komandu da učitate datoteku i odmah primenite promene:

```
$ source ~/.bashrc
```

Sada kada razumete svrhu Bash dot datoteka, pređimo na njihovo uređivanje kako bismo personalizovali okruženje.

Prilagođavanje Bash upita

Upit je mesto gde unosite komande u Bash terminalu. Može biti jednostavan ili složen, u zavisnosti od vaših potreba i ličnih preferencija. Zamislite prilagođavanje svog upita kao način na koji slikar bira svoju paletu boja.

Trenutno podešen Bash upit možete pronaći u datoteci `~/.bashrc`, tražeći liniju koja počinje sa `PS1`. Uobičajen Bash upit može imati vrednost `export PS1="\u@\h \w\$ "`, i ovo bi izgledalo kao `username@hostname ~$` u terminalu. Hajde da objasnimo šta svaki deo znači:

- `\u` – zamenjuje se trenutnim korisničkim imenom.
- `@` –znak „@", koji se pojavljuje odmah nakon korisničkog imena.
- `\h` – zamenjuje se imenom računara do prve tačke.
- `\w` – prikazuje trenutni radni direktorijum, pri čemu se `$HOME` skraćuje na tildu (~).
- `\$` – prikazuje znak `$` za uobičajenog korisnika ili `#` znak za administratorskog korisnika.

Nakon uređivanja `PS1` upita, ne zaboravite da učitate izmenjenu datoteku kako bi promene stupile na snagu.

Možete dodatno prilagoditi svoj upit prema specifičnim potrebama. Na primer, možete dodati trenutnu IP adresu u upit pomoću sledećeg izraza `$(ip a show eth0 | grep -m 1 inet | tr -s ' ' | cut -d ' ' -f 3)`. Ovo može biti korisno za beleženje aktivnosti u zapisima ili izveštajima, kako bi klijenti mogli da povežu vašu aktivnost sa njihovim **sistemima za upravljanje bezbednosnim informacijama i događajima (SIEM)**. Za vizuelno generisanje Bash upita, posetite `https://bash-prompt-generator.org/`, ili pogledajte zvaničnu Bash dokumentaciju za sve dostupne opcije.

Prilagođavanje Bash okruženja je način da optimizujete svoj terminal i učinite ga efikasnijim. Eksperimentišite sa podešavanjima, pronađite ono što vam donosi veću produktivnost i dodajte lični pečat vašem komandnom okruženju. Čak i male promene mogu značajno unaprediti vaš rad u terminalu.

Postavljanje neophodnih alata za testiranje neprobojnosti

U ovom delu ćemo proći kroz postavljanje okruženja za testiranje neprobojnosti, uključujući ažuriranje sistemskih paketa i instalaciju dodatnih alata potrebnih za rad. Većina neophodnih alata već je unapred instalirana u Kali Linux sistemu, tako da ćemo dodati samo nekoliko dodatnih softverskih paketa.

Ažuriranje menadžera paketa

Prvi korak pri korišćenju nove Linux instalacije jeste ažuriranje paketa. Kao što je ranije pomenuto, u svim demonstracijama koristiću Kali Linux. Kali se zasniva na Debian Linux distribuciji i koristi **Advanced Package Tool (APT)** kao upravljač paketa. U suštini, `apt` pojednostavljuje upravljanje softverom. Automatski preuzima, konfiguriše i instalira softverske pakete iz unapred definisanih spremišta. Ova automatizacija ne samo da štedi vreme, već i osigurava da se sve zavisnosti softvera reše bez ručne intervencije.

Pokretanje komande `sudo apt update` osvežava lokalnu bazu podataka dostupnih paketa i njihovih verzija, osiguravajući da imate najnovije informacije iz repozitorijuma. Ovaj

korak je ključan pre instalacije novog softvera ili ažuriranja postojećih paketa, kako biste bili sigurni da dobijate najnovije verzije. Ako koristite **Kali**, **Ubuntu** ili **Debian Linux**, sledeće komande za ažuriranje i nadogradnju će raditi kako je očekivano, jer sve ove distribucije koriste `apt` upravljač paketa:

```
$ sudo apt update && sudo apt upgrade -y && reboot
```

U prethodnoj komandi koristimo `sudo` za podizanje privilegija i `apt` za ažuriranje liste dostupnih paketa. Dvostruki simboli & (`&&`) deluju kao operator logičke konjunkcije; druga komanda za nadogradnju paketa bez potvrde (`-y`) izvršiće se samo ako se prva komanda uspešno završi. Na kraju, sistem se ponovo pokreće kako bi sve promene, uključujući ažuriranja servisa i jezgra, stupile na snagu.

Instalacija ProjectDiscovery alata

ProjectDiscovery nudi odlične alate koje preporučujem za testiranje neprobojnosti (*PDTM – ProjectDiscovery*, `https://github.com/projectdiscovery/pdtm`). Pre nego što ih instaliramo, potrebno je da instaliramo okruženje i biblioteke programskog jezika Go. Pratite sledeće korake za instalaciju:

1. U veb pregledaču idite na **`https://go.dev/dl/`**.
2. Preuzmite odgovarajući paket za svoju Linux distribuciju. Obratite pažnju na arhitekturu procesora. Obično bi to bila vrednost `Archive` za `Kind`, vrednost `Linux` za `OS` i vrednost `x86-64` ili `ARM64` za `Arch`.
3. Otpakujte preuzetu arhivu. Ne zaboravite da promenite verziju paketa tako da odgovara onoj koju ste preuzeli:

   ```
   $ sudo tar -C /usr/local -xzf go1.22.0.linux-amd64.tar.gz
   ```

4. Dodajte `/usr/local/go/bin` u promenljivu okruženja `PATH` u svojoj datoteci `~/.bashrc`. Promenljiva okruženja `PATH` govori vašem Bash okruženju gde da pronađe punu putanju do izvršnih programa kada ne navedete putanju pre komande. Komanda `echo` ispisuje tekst unutar navodnika u terminal, a znak veće od (>) preusmerava izlaz u datoteku. Obratite pažnju da koristimo dva znaka >> za preusmeravanje izlaza. Da smo koristili samo jedan znak (>), sadržaj datoteke bi bio zamenjen novim. Pošto želimo da dodamo novi unos na kraj datoteke, koristimo dva znaka >>:

   ```
   $ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bashrc
   ```

5. Učitajte datoteku kako bi promene stupile na snagu:

   ```
   $ source ~/.bashrc
   ```

6. Proverite da li je `/usr/local/go/bin` dodat u vašu `PATH` promenljivu (pogledajte nakon poslednjeg znaka dvotačke):

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
local/games:/usr/games:/usr/local/go/bin
```

7. Proverite da li je Go pravilno instaliran i može li se pronaći u `PATH` promenljivoj. Verzija i arhitektura mogu varirati:

```
$ go version
go version go1.22.0 linux/arm64
```

8. Instalirajte `pdtm` iz ProjectDiscovery. Ovo je alat koji instalira i upravlja ažuriranjima za sve alate iz ProjectDiscovery:

```
$ go install -v github.com/projectdiscovery/pdtm/cmd/pdtm@latest
```

9. Dodajte `pdtm` u vašu `PATH` promenljivu:

```
$ echo "export PATH=$PATH:$HOME/.pdtm/go/bin" >> ~/.bashrc
```

10. Pokrenite sledeću `pdtm` komandu da instalirate sve alate:

```
$ pdtm -install-all
```

11. Instalirajte `libpcap` za `naabu`:

```
$ sudo apt install -y libpcap-dev
```

To završava instalaciju svih potrebnih ProjectDiscovery alata.

Instalacija NetExec alata

NetExec je alat za iskorišćavanje mrežnih servisa koji pomaže u automatizaciji procene bezbednosti velikih mreža (*NetExec wiki*, `https://www.netexec.wiki/`).

Po mom mišljenju, NetExec je jedan od najkorisnijih alata za testiranje neprobojnosti internih mreža. Podržava većinu mrežnih protokola koji su potrebni tokom testiranja unutrašnje mreže, kao i testiranje Microsoft Active Directory okruženja.

NetExec ima previše funkcionalnosti da bi se sve ovde nabrojale, ali neke od najkorisnijih mogućnosti koje koristim uključuju:

- Skeniranje ranjivosti; NetExec sadrži korisne module za testiranje uobičajenih ranjivosti.
- Napadi grubom silom na autentifikaciju za testiranje slabih lozinki.
- Napad širenjem lozinki ili heširanih lozinki na serverimada bi se pronašlo gde dati akreditivi imaju pristup kao lokalni administrator.
- Izvršavanje komandi.
- Prikupljanje akreditiva.
- Identifikacija SMB deljenih resursa za pristup čitanju/pisanju.

Unesite sledeću komandu da instalirate NetExec:

```
$ sudo apt install -y pipx git && pipx ensurepath && pipx install
git+https://github.com/Pennyw0rth/NetExec
```

Ovim završavamo instalaciju najčešće korišćenih alata za testiranje neprobojnosti koji nisu unapred instalirani.

Rezime

U ovom poglavlju ste upoznati sa neophodnim veštinama Bash skriptovanja, koje su ključne za svakog ko želi da se usavrši u testiranju neprobojnosti. Počeli smo razjašnjavanjem šta je Bash i naglašavanjem njegove važnosti u zadacima sajber bezbednosti. Ovo nije bilo samo puko memorisanje komandi, već način da iskoristimo Bash za automatizaciju ponavljajućih zadataka, obradu podataka i sprovođenje bezbednosnih provera na efikasan način. Nakon toga, prošli smo kroz odabir odgovarajućeg operativnog sistema koji podržava Bash, postavljajući temelj za uspešno skriptovanje. Zatim smo se posvetili podešavanju hakerskog okruženja, prilagođavanju njegovog izgleda i ponašanja prema ličnim preferencijama. Ovo prilagođavanje nije bilo samo radi estetike, već radi stvaranja funkcionalnog i efikasnog radnog okruženja. Na kraju, ovo poglavlje je predstavilo osnovne alate za testiranje neprobojnosti, uz detaljno objašnjenje njihove instalacije i osnovne upotrebe. Do sada imate dobro pripremljeno okruženje i osnovno razumevanje kako Bash skriptovanje može značajno poboljšati vaše sposobnosti u testiranju neprobojnosti.

U sledećem poglavlju bavićemo se tehnikama za rad sa datotekama i direktorijumima.