Analiza i izvedba penetracijskog testiranja za povećanje sigurnosti informacijsko-komunikacijskog sustava

Knögel, Nikola

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:670899

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Nikola Knőgel

ANALIZA I IZVEDBA PENETRACIJSKOG TESTIRANJA ZA POVEĆANJE SIGURNOSTI INFORMACIJSKO-KOMUNIKACIJSKOG SUSTAVA

DIPLOMSKI RAD

Zagreb, 2023.

Sveučilište u Zagrebu

Fakultet prometnih znanosti

DIPLOMSKI RAD

Analiza i izvedba penetracijskog testiranja za povećanje sigurnosti informacijsko-komunikacijskog sustava

Analysis and performance of penetration testing to increase the security of the information and communication system

Mentor: prof. dr. sc. Dragan Peraković

Student: Nikola Knőgel

JMBAG: 0135236661

Zagreb, rujan 2023.

ANALIZA I IZVEDBA PENETRACIJSKOG TESTIRANJA ZA POVEĆANJE SIGURNOSTI INFORMACIJSKO-KOMUNIKACIJSKOG SUSTAVA

SAŽETAK

Ovaj diplomski rad temelji se na istraživanju penetracijskog testiranja kao ključne komponente za osiguravanje kibernetičke sigurnosti u različitim okruženjima. Analizirane su različite vrste penetracijskih testova, uključujući crne, bijele i sive kutije, te su uspoređene njihove prednosti i nedostaci. Istražene su različite tehnike napada i metode zaštite od njih, analizirajući ranjivosti i propuste te pružajući strategije za otklanjanje tih rizika. Rad se također bavi primjenom metodologije penetracijskog testiranja u organizaciji, naglašavajući važnost identifikacije ciljeva testiranja, uloge tima i izradu plana testiranja. Proučavanjem primjene penetracijskog testiranja u oblaku, identificirane su ranjivosti i sigurnosni mehanizmi specifični za tu domenu. Analiziran je utjecaj različitih modela u oblaku i razmatrane su metode zaštite u oblaku. Također, istražena je automatizacija penetracijskog testiranja te je napravljena usporedba između automatiziranog i ručnog testiranja i analizirani su različiti alati.

KLJUČNE RIJEČI: penetracijsko testiranje, sigurnost, oblak, tehnike napada, zaštita, metodologija, automatizacija.

SUMMARY

This thesis is based on the research of penetration testing as a key component for ensuring cyber security in different environments. Different types of penetration tests, including black, white and gray boxes, are analyzed and their advantages and disadvantages are compared. Various attack techniques and methods of protection against them are explored, analyzing vulnerabilities and flaws and providing strategies to eliminate these risks. The paper also deals with the application of penetration testing methodology in the organization, emphasizing the importance of identifying testing objectives, the role of the team and creating a test plan. By studying the application of penetration testing in the cloud, vulnerabilities and security mechanisms specific to that domain were identified. The impact of different cloud models is analyzed and cloud protection methods are discussed. Also, the automation of penetration testing was investigated and a comparison was made between automated and manual testing and different tools were analyzed.

KEY WORDS: penetration testing, security, cloud, attack techniques, protection, methodology, automation.

SVEUCILISTE U ZAGREBU
FAKULTET PROMETNIH
ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI
ISPIT

Zagreb, 21. lipnja
2023.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Sigurnost i zastita informacijsko komunikacijskog sustava**

DIPLOMSKI ZADATAK br. 7239

Pristupnik: **Nikola Knogel (0135236661)**

Studij: Promet

Smjer: Informacijsko-komunikacijski promet

Zadatak: **Analiza i izvedba penetracijskog testiranja za povecanje sigurnosti informacijsko-komunikacijskog sustava**

Opis zadatka:

U radu je potrebno prikazati rezultate provedene analiza izvedbe penetracijskog testiranja koje je provedeno za potrebe ovga diplomskoga rada. Prikazati mogucnosti povecanja sigurnosti informacijsko-komunikacijskog sustava.

Mentor:

prof. dr. sc. Dragan Perakovic

Predsjednik povjerenstva za diplomski ispit:

Sadržaj

1. Uvod

U doba ubrzanog tehnološkog napretka i povećane digitalne povezanosti, pitanje kibernetičke sigurnosti izrasta u neizostavan izazov za organizacije širom svijeta. Konstantno izmjenjivanje prijetnji i sofisticirane taktike napadača stavljaju pred organizacije potrebu da proaktivno prepoznaju i neutraliziraju potencijalne sigurnosne rizike. U tom kontekstu, penetracijsko testiranje, kao ključna strategija identifikacije i otklanjanja ranjivosti, postaje ključna komponenta u osiguranju kibernetičke sigurnosti.

Ovaj diplomski rad usmjeren je na sveobuhvatno istraživanje i analizu penetracijskog testiranja kao ključnog alata za osiguravanje kibernetičke sigurnosti u različitim okruženjima. Fokusirajući se na širok spektar aspekata, rad će istražiti različite vrste penetracijskih testova te njihovu primjenu. Kroz usporedbe crnih, bijelih i sivih kutija, cilj je identificirati njihove specifične prednosti i ograničenja.

Kroz sveobuhvatno istraživanje i analizu, ovaj rad ima za cilj pružiti dublje razumijevanje penetracijskog testiranja kao ključne strategije za osiguranje kibernetičke sigurnosti. Ključni ciljevi rada uključuju analizu različitih aspekata penetracijskog testiranja, usporedbu metoda, identifikaciju izazova, evaluaciju tehnika napada i zaštite, primjenu metodologije i analizu oblaka te istraživanje automatizacije u ovom kontekstu. Kroz sve navedeno, ovaj rad teži osnažiti organizacije u razumijevanju i primjeni penetracijskog testiranja kao ključnog koraka u osiguranju kibernetičke sigurnosti.

Naslov diplomskog rada jest *Analiza i izvedba penetracijskog testiranja za povećanje sigurnosti informacijsko-komunikacijskog sustava*, a sam rad strukturiran je kroz 7 poglavlja, uključujući uvod i zaključak:

Seminarski rad sastoji se od 6 poglavlja:

1. Uvod
2. Analiza različitih vrsta penetracijskih testova i njihove primjene u praksi
3. Istraživanje različitih tehnika napada i metoda zaštite od njih
4. Primjena metodologije za penetracijsko testiranje u organizaciji
5. Primjena penetracijskog testiranja u oblaku
6. Automatizacija penetracijskog testiranja
7. Zaključak

U drugom poglavlju opisane su tehnike testiranja bijele, crne i sive kutije. Prikazat će se njihovi nedostaci i prednosti te sami tipovi tehnike testiranja. Nabrojat će se različite vrste penetracijskog testiranja te će se opisati njihove prednosti i koraci za izvođenje penetracijskog testiranja.

U trećem poglavlju istražit će se raznolike tehnike napada koje se koriste za iskorištavanje ranjivosti sustava te analiza učinkovitih metoda zaštite od tih prijetnji.

Cilj istraživanja je dublje razumijevanje prijetnji koje organizacije mogu suočiti te pružanje smjernica za usvajanje odgovarajućih zaštitnih strategija.

U četvrtom poglavlju prikazana je važnost i proces primjene metodologije za penetracijsko testiranje kao ključnog alata u osiguranju kibernetičke sigurnosti organizacije. Fokusirajući se na strukturiran i sustavan pristup, istraživanje će analizirati kako organizacije mogu uspješno implementirati penetracijsko testiranje kako bi identificirale ranjivosti i propuste te osigurale pouzdanu sigurnost sustava i podataka.

U petom poglavlju opisat će se modeli i sama sigurnost usluga u oblaku. Navest će se prednosti i nedostaci penetracijskog testiranja u oblaku te koji su izazovi i sami procesi izvođenja penetracijskog testiranja u oblaku.

U šestom poglavlju prikazana je usporedba između ručnog i automatskog penetracijskog testiranja. Opisat će se alati koji se koriste za automatizaciju penetracijskog testiranja te će se navesti prednosti i izazovi korištenja tih alata.

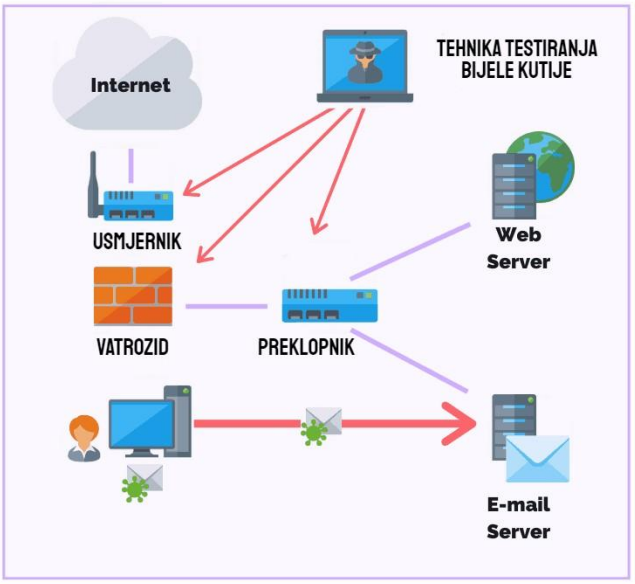
2. Analiza različitih vrsta penetracijskih testova i njihove primjene u praksi

Penetracijsko testiranje pokušava iskoristiti slabosti ili ranjivosti u sustavima, mrežama, ljudskim resursima ili fizičkoj imovini kako bi se testirala učinkovitost sigurnosnih kontrola.

Različite vrste penetracijskih testova uključuju mrežne usluge, aplikacije, klijentsku stranu, bežične mreže, društveni inženjering i fizički napad. Penetracijski test može se izvesti eksterno ili interno za simulaciju različitih vektora napada. Ovisno o ciljevima svakog testa, penetracijski ispitivač može ili ne mora imati prethodno znanje o okruženju i sustavima koje pokušava probiti. To je kategorizirano kao penetracijsko testiranje crne kutije, bijele kutije i sive kutije.

2.1. Tehnika testiranja bijele kutije

Testiranje bijele kutije je metoda dizajna testnih slučajeva koja koristi kontrolnu strukturu proceduralnog dizajna za izvođenje testnih slučajeva. Testiranje bijele kutije može otkriti pogreške u implementaciji kao što je loše upravljanje ključevima analizirajući interni rad i strukturu pojedinog dijela softvera. Na slici 1 prikazana je tehnika testiranja bijele kutije, [3].



Slika 1: Tehnika testiranja bijele kutije

Izvor:[4]

Prednosti tehnike testiranja bijele kutije:

- Temeljitost - potpuna pokrivenost kodom je temeljno načelo testiranja bijele kutije. Osnovni koncept je testirati što više koda, što je daleko temeljitije od standardnog testiranja crne kutije. Temeljitost testiranja bijele kutije također daje poseban

okvir. Pravila testiranja moraju biti precizna, inženjerski utemeljena i dobro definirana. Ovakav način testiranja je transparentan, moguće je napraviti temeljita testiranja koja pokrivaju sve moguće putove kao i kompletnu strukturu i bazu koda. Također procjenjuje unutarnje i vanjske ranjivosti, što bi moglo pomoći u sprječavanju budućih sigurnosnih prijetnji i napada.

- Jedinično testiranje - razumijevanje internog rada aplikacije omogućeno je jediničnim testiranjem. Kao što naziv govori, jedinični testovi ispituju pojedinačne retke koda ili jedinice kako bi se utvrdilo funkcioniraju li kako je predviđeno. Ove testove lako je izvesti programski, omogućujući programerima da brzo utvrde je li nešto slomljeno. Jedinični testovi su koristan alat za određivanje je li se komponenta koja je prethodno radila nedavno slomila.
- Vrijeme - upravljanje vremenom glavna je odgovornost tijekom procesa razvoja softvera jer postoje stalni rokovi koje treba ispuniti. Testiranje bijele kutije ima mogućnost drastičnog ubrzanja procesa testiranja. Programeri često imaju opće razumijevanje problema i najbolji način za njegovo rješavanje čim otkriju grešku. Troškovi komunikacije između programera i osiguranja kvalitete (Quality Assurance – QA) također su eliminirani testiranjem bijele kutije jer programeri mogu identificirati i riješiti probleme bez čekanja na QA.
- Optimizacija - analiza koda odjeljak po odjeljak omogućuje razvojnim programerima da eliminiraju nepotreban kod ili sažimaju već postojeći kod. Dodatno, uklanjanjem prikrivenih problema koji mogu proći neotkriveni tijekom rutinskog testiranja, kod se može učiniti učinkovitijim.
- Introspekcija - testiranje bijele kutije omogućuje programerima da temeljito razmotre implementaciju. Programeri su prisiljeni razmišljati o odnosima između različitih dijelova koda. Možda je postojeća implementacija odgovarajuća, ali se neće dobro mjeriti u budućnosti ili sadrži nepotrebne komponente koje se mogu ukloniti. Razvojni programeri mogu pregledati dizajne i razmotriti kako bi se mogli poboljšati korištenjem testiranja bijeloj kutiji.

Nedostaci tehnike testiranja bijele kutije:

- Veliki troškovi – Testiranje bijele kutije zahtijeva mnogo vremena i novca jer je temeljitije. Iako je to donekle ublaženo jediničnim testiranjem, pisanje jediničnih testova zahtijeva početna ulaganja. Testiranje putem bijele kutije zahtijeva kompetentne ispitivače koji poznaju kod i posjeduju vještinu programiranja, za razliku od testiranja putem crne kutije. To povećava troškove i može obeshrabriti programere da rade na dodatnim značajkama.
- Baza koda koja se brzo mijenja - Ako se baza koda brzo mijenja, automatizirani testni slučajevi su beskorisni. Većina napisanih testnih slučajeva često je beskorisna nakon redizajna ili prerade i zahtijeva ponovno pisanje. Ako se implementacija često mijenja, potrebna je ažurirana skripta za testiranje.
- Velika potrošnja vremena - Kada se za velike aplikacije koristi pristup testiranju bijele kutije, iscrpno testiranje postaje još teže. Testiranje bijele kutije oduzima

puno vremena jer zahtijeva stvaranje širokog raspona ulaznih podataka za testiranje svakog mogućeg puta i okolnosti.

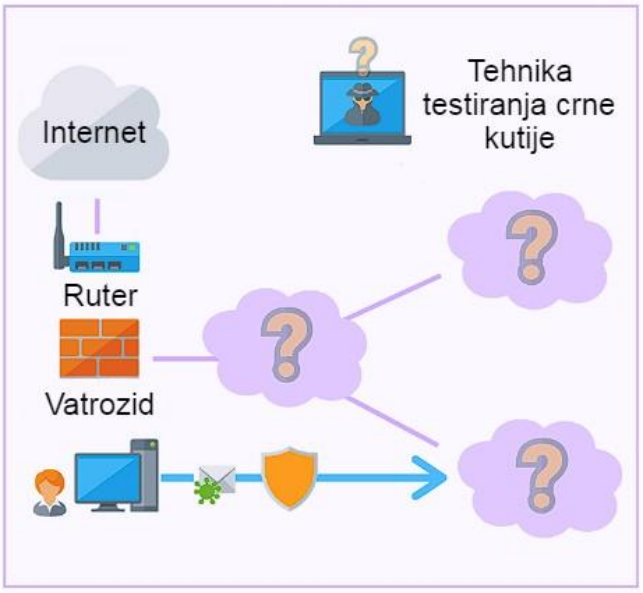
- Više pogrešaka (engl. *Errors*) - Nije realno testirati sve uvjete, stoga bi neki mogli proći netestirani. Uz opću metodu analiziranja svake linije po liniji ili putanje po putanji, postoji mogućnost da se pogreške u kodu možda neće pronaći, [5].
- Tipovi tehnike testiranja bijele kutije:
- Testiranje kontrolnog toka - to je strategija strukturalnog testiranja koja koristi tok upravljanja programom kao tok upravljanja modelom i daje prednost jednostavnijim putevima koji su u većem broju u odnosu na kompliciranje puteve kojih ima manje.
- Testiranje grana – cilj je testirati svaku opciju (istina ili laž) na svakoj kontrolnoj izjavi koja također uključuje složenu odluku
- Testiranje osnovnog puta - testiranje osnovnog puta omogućuje dizajneru testnog slučaja da proizvede logičku mjeru složenosti proceduralnog dizajna i zatim koristi ovu mjeru kao pristup za ocrtavanje osnovnog skupa putova izvršenja.
- Testiranje toka podataka - u ovoj vrsti testiranja graf kontrolnog toka označen je informacijama o tome kako se definiraju i koriste programske varijable.

2.2. Tehnika testiranja crne kutije

Testiranje crne kutije tretira softver kao "Crnu kutiju" – bez ikakvog znanja o internom radu i ispituje samo temeljne aspekte sustava. Prilikom izvođenja testa crne kutije, ispitivač mora poznavati arhitekturu sustava i neće imati pristup izvornom kodu. Na slici 2 prikazana je tehnika testiranja crne kutije, [3].

Prednosti tehnike testiranja crne kutije:

- Mala potrošnja vremena - u usporedbi s drugim pristupima, pojednostavljena tehnika testiranja oduzima manje vremena za testiranje softvera jer testiranje crne kutije zahtijeva manje pripreme. Nema potrebe za izradom testnih skripti za ispitivače. Znanje o testiranju dovoljno je da se procijeni kako će se određene greške (engl. *Bugs*) otkriti tijekom testiranja; stoga oduzima manje vremena.
- Brza testiranja aplikacija trećih strana - neki softveri ili aplikacije treće strane mogu se testirati jedino uz detaljno testiranje jer je nemoguće stvoriti cjelovito testno okruženje. Testiranje crne kutije može se provesti s dostupnim informacijama i daje uvid u to koliko dobro proizvod radi.
- Kritična procjena - kritični problemi i praznine u funkcionalnosti otkrivaju se u testiranju crne kutije bez prethodnog znanja o internom radu sustava.
- Testiranje crne kutije je nepristrano - kod testiranja crne kutije ispitivač ne zna interno kodiranje softvera jer dizajner i ispitivač rade neovisno. To čini testiranje crne kutije nepristranim i najboljim za testiranje.
- Pruža iskustvo krajnjem korisniku - svi testovi u testiranju crne kutije provode se iz perspektive krajnjeg korisnika.



Slika 2: Tehnika testiranja crne kutije

Izvor:[6]

- Može testirati na različitim preglednicima i operativnim sustavima - testiranje crne kutije nije ograničeno samo na web aplikacije, već se može primijeniti na mobilne aplikacije ili drugi softver gdje je dostupna samo dokumentacija. Zbog ove značajke, testiranje je funkcionalno preko različitih preglednika i operativnih sustava. Ispitivač koji testira mobilnu aplikaciju zahtijeva internetsku vezu (nije moguće stvoriti lokalno okruženje za testiranje). Ova tehnika je najbolja opcija jer ne zahtijeva račun u relevantnoj trgovini aplikacija kao što su Google, Apple itd.
- Testiranje može započeti u ranoj fazi razvoja softvera - prije završetka funkcionalne specifikacije, testni slučajevi mogu se učinkovito dizajnirati u testiranje crne kutije. Ispitivač ne treba čekati završetak koda i može provoditi testiranje tijekom razvojne faze zbog manjeg kašnjenja između trenutka kada se pojavi greška i kad je ta greška uočena.
- Najbolje za testiranje velikih aplikacija - ova tehnika je dobro dizajnirana za testiranje velikih aplikacija. Ne treba posebno okruženje i uglavnom se fokusira na ulaz i izlaz softverskih aplikacija (na temelju softverskih zahtjeva).
- Korisnička perspektiva - u testiranju crne kutije, ispitivači imaju pristup drugačijim informacijama od programera jer obraćaju pozornost na funkcionalnost programa. Time mogu izraziti zabrinutost ili podijeliti komentare iz perspektive korisnika.
- Najbolje je za specifikaciju softvera - ispitivači trebaju pristupiti dokumentima sa zahtjevima i specifikacijama koji mogu pomoći u prepoznavanju dvosmislenog jezika u dokumentaciji. Proizvodni tim razmatra rezultate i komentare ispitivača kako bi izradio bolje specifikacije za krajnje korisnike.
- Štedi troškove i trud u raznim projektima - testiranje crne kutije najbolji je pristup testiranju jer štedi troškove testiranja i truda bilo kojeg velikog ili malog projekta sa svojim raznim značajkama poput toga da ne ovisi o konfiguraciji sustava ili

aplikacije, ne treba tehničke testere cijelo vrijeme. Pronalazak greški prije završetka koda štedi vrijeme i novac poslovanja.

Nedostaci tehnike testiranja crne kutije:

- Nema određivanja prioriteta grešaka - testiranje crne kutije obično identificira neispravne module, ali ne objašnjava u detalje ispunjavaju li zahtjev. Ispitivači moraju dobro poznavati određene greške jer su neke uzrokovane netočnom implementacijom, a neke treba uočiti tijekom testiranja i QA usluga. Određivanje prioriteta grešaka je teško jer je u testiranju crne kutije pristup kodu ili dokumentaciji nemoguć. Pristup je neophodan da bi se znalo što je učinjeno ispravno, a što nije.
- Testiranje crne kutije ne pruža detaljno objašnjenje - testiranje crne kutije je najprikladnije kada se ne traže detalji testiranja. Korisno je kada se ne pojavljuju nedokumentirani zahtjevi.
- Precijenjeni rezultati - zbog nejasnih funkcionalnih specifikacija crno testiranje daje precijenjene rezultate postupka testiranja i ne može se koristiti za složene segmente testiranja koda.
- Izazovno dizajniranje testnih slučajeva - budući da testiranje crne kutije nema jasnu funkcionalnu specifikaciju, ispitivačima stvara poteškoće pri izradi testnih slučajeva.
- Mora pružiti cjelovitu sliku - testiranje crne kutije pruža vanjsko znanje i donosi pretpostavke na temelju toga. Zbog toga se testiranje crne kutije može koristiti s drugim pristupima testiranju poput testiranja bijele kutije i testiranja sive kutije, [7].

Tipovi tehnike testiranja crne kutije:

- Particioniranje ekvivalencije - može smanjiti broj testnih slučajeva, budući da dijeli ulazne podatke softverske jedinice u particiju podataka iz kojih se mogu izvesti testni slučajevi.
- Analiza graničnih vrijednosti - više se usredotočuje na testiranje na granicama ili gdje su odabrane ekstremne granične vrijednosti. Uključuje minimalne, maksimalne, samo unutar/izvan granica, vrijednosti pogreške i tipične vrijednosti.
- *Fuzzing* - *fuzz* testiranje koristi se za pronalaženje grešaka u implementaciji, korištenjem neispravnog/polu-neispravnog ubacivanja podataka u automatiziranoj ili poluautomatiziranoj sesiji.
- Grafikon uzrok-posljedica - to je tehnika testiranja u kojoj testiranje počinje izradom grafikona i utvrđivanjem odnosa između posljedice i njezinih uzroka. Identitet, negacija, logika lLI i logika I četiri su osnovna simbola koja izražavaju međuovisnost između uzroka i posljedice.
- Testiranje ortogonalnog niza – može se primijeniti na probleme u kojima je ulazna domena relativno mala, ali prevelika da bi se prilagodila iscrpnom testiranju.
- Testiranje svih parova - u tehnici testiranja svih parova, testni slučajevi su dizajnirani za izvršavanje svih mogućih diskretnih kombinacija svakog para

ulaznih parametara. Glavni cilj je imati skup testnih slučajeva koji pokriva sve parove.

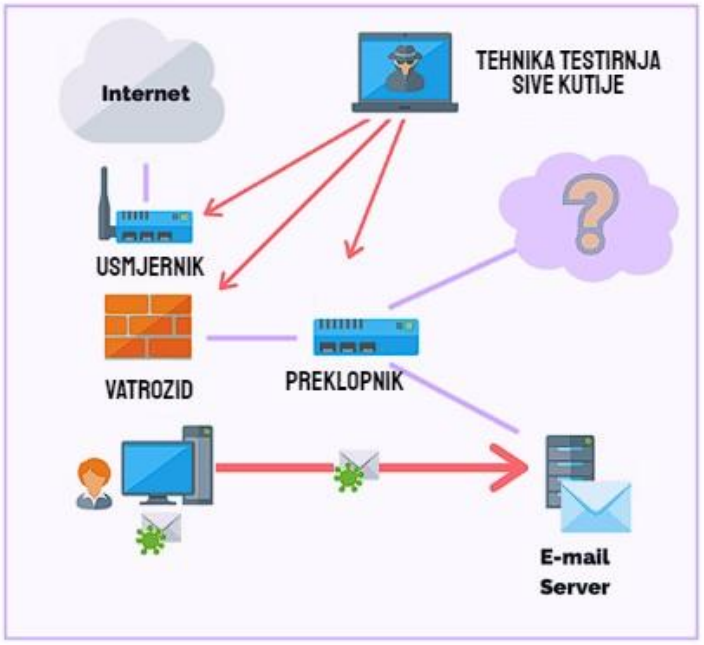
- Testiranje prijelaza stanja: Ova vrsta testiranja korisna je za testiranje stroja stanja i također za navigaciju grafičkim korisničkim sučeljem.

2.3. Tehnika testiranja sive kutije

Tehnika testiranja sive kutije povećat će pokrivenost testiranja dopuštajući nam da se usredotočimo na sve slojeve bilo kojeg složenog sustava koristeći kombinaciju svih postojećih tehnika testiranja bijele kutije i crne kutije. U testiranju u sivoj kutiji ispitivač mora poznavati unutarnje strukture podataka i algoritam, u svrhu dizajniranja testnih slučajeva. Primjeri tehnika testiranja sive kutije su:

- Arhitektonski model
- Jedinstveni jezik za modeliranje (Unified Modeling language - UML)
- Model stanja (stroj konačnog stanja).

U testiranju sive kutije proučavaju se kodovi dvaju modula (metoda testiranja bijele kutije) za dizajn testnih slučajeva, a stvarni test se izvodi u izloženim sučeljima (metoda testiranja crne kutije). Na slici 3 prikazana je tehnika testiranja sive kutije, [3].



Slika 3: Tehnika testiranja sive kutije

Izvor:[1]

Prednosti tehnike testiranja sive kutije:

- Kombinacija crne i bijele kutije - budući da je testiranje u sivoj kutiji kombinacija testiranja u crnoj kutiji i u bijeloj kutiji, ono pruža najbolje od oba svijeta, tj. prednosti obje tehnike testiranja.
- Testni scenariji - poznavanje unutarnjih mehanizama sustava pomaže ispitivaču da opsežnije osmisli testne scenarije.
- Očuvanje izvornog koda - za testiranje sive kutije koriste se funkcionalne specifikacije i drugi projektni dokumenti. Ne zahtijeva korištenje izvornog koda što pomaže u očuvanju izvornog koda od bilo kakvih ometajućih promjena.
- Organizacija poslova - pomaže u odvajanju testera i programera, čime se smanjuju sve nesuglasice između njih.
- Testiranje iz korisničke perspektive - čak i uz djelomično razumijevanje koda, testeri provode testiranje u sivoj kutiji iz perspektive krajnjeg korisnika. To pomaže u identificiranju problema koje su razvojni programeri mogli propustiti tijekom testiranja jedinice.
- Problemi se lako rješavaju – tester ima mogućnost odmah popraviti trenutni problem jer tester može promijeniti djelomično dostupan kod da provjeri rezultate.
- Nije potrebna dodatna edukacija zaposlenika - čak i bez vještina programiranja na visokoj razini, testeri mogu izvesti ovo testiranje.

Nedostaci tehnike testiranja crne kutije:

- Nemogućnost pronalaska kritičnih ranjivosti - tijekom testiranja u sivoj kutiji, testeri nemaju pristup izvornom kodu, tako da postaje teško dobiti potpunu pokrivenost putanje koda, a testeri možda neće primijetiti neke kritične ranjivosti.
- Testiranje algoritama nije moguće jer pristup kompletnoj logici algoritama nije moguć.
- Redundancija - ako je programer već izvršio testni slučaj, izvođenje istog testnog slučaja u testiranju u sivoj kutiji može rezultirati redundancijom, [8].
- Tipovi tehnike testiranja crne kutije:
- Testiranje ortogonalnog niza - ova vrsta testiranja koristi se kao podskup svih mogućih kombinacija.
- Testiranje matrice - u testiranju matrice navodi se izvješće o statusu projekta.
- Regresijsko testiranje - ako se naprave nove promjene u softveru, regresijsko testiranje podrazumijeva izvođenje testnih slučajeva.
- Testiranje uzorka - testiranje uzorka provjerava dobru primjenu za njegovu arhitekturu i dizajn.

2.4. Usporedba crne, bijele i sive kutije

Na sljedećoj tablici bit će prikazana usporedba između crne, bijele i sive kutije:

Tablica 1. Prikaz usporedbe između tehnika testiranja crne kutije, sive kutije i bijele kutije

	Crna kutija	Siva kutija	Bijela kutija
1.	Analizira samo temeljne aspekte, tj. nema dokazane prednosti unutarnjeg rada	Djelomično poznavanje unutarnjeg rada	Potpuno poznavanje internog rada
2.	Zrnatost je niska	Zrnatost je srednja	Zrnatost je visoka
3.	Izvode ga krajnji korisnici, a također testeri i programeri	Izvode ga krajnji korisnici, a također testeri i programeri	Izvode ga programeri i testeri
4.	Testiranje se temelji na vanjskim iznimkama – interno ponašanje programa se zanemaruje	Dizajn testa temelji se na dijagramima baze podataka visoke razine, dijagramima protoka podataka, internim stanjima, poznavanju algoritma i arhitekture	Interni su u potpunosti poznati
5.	Najmanje resursa i vremena potroši	Negdje je između	Najviše resursa i vremena potroši
6.	Može se testirati samo metodom pokušaja i pogrešaka	Podatkovne domene i unutarnje granice mogu se testirati i prekoračiti, ako su poznate	Najbolje testira podatkovne domene i unutarnje granice
7.	Nije prikladno za testiranje algoritama	Nije prikladno za testiranje algoritama	Pogodan je za testiranje algoritama

Izvor: [3]

2.5. Vrste penetracijskog testiranja

Svaka vrsta penetracijskog testa zahtijeva specifično znanje, metodologiju i alate za izvođenje i trebala bi biti usklađena s određenim poslovnim ciljem. Ovi ciljevi mogu varirati od poboljšanja svijesti o napadima društvenog inženjeringa na zaposlenike u cijeloj tvrtki, do implementacije sigurnog razvoja koda za prepoznavanje nedostataka u softverskom kodu u stvarnom vremenu ili ispunjavanja regulatornih obveza ili obveza usklađenosti, [1].

2.5.1. Penetracijsko testiranje mreže

Penetracijsko testiranje mreže radi pomoću simulacije napada iz stvarnog života, pružajući kritične informacije o potencijalnim slabostima koje hakeri mogu iskoristiti kao ulazne točke za pristup mrežama. "Etički hakeri" koriste različite metode kako bi pokušali kompromitirati mrežu ili mreže.

Koraci pri testiranju penetracije mreže su sljedeći:

Planiranje - u fazi planiranja, etički hakeri raspravljaju o opsegu i općem cilju testa s ključnim dionicima. Metode testiranja i metrike uspjeha definirane su u ovoj početnoj fazi rasprave. Nakon što se donese odluka o osnovnom pregledu, hakeri počinju pregledavati sve komponente mreže poduzeća.

Testiranje - u ovoj fazi hakeri koriste statička ili dinamička rješenja testiranja kako bi proučili i razumjeli kako mreža reagira na simulirane napade.

Pristup mrežama - nakon testiranja mreže kako bi razumjeli njezino ponašanje, etički hakeri izvest će razne napade na mrežu, uključujući napade na web aplikacije, strukturiranog jezik upita (Structured query language - SQL) injekcije itd. Ovi napadi pomoći će identificirati ranjivosti ciljne mreže. Ako etički hakeri identificiraju ranjivosti, pokušat će ih stvarno iskoristiti, od pokušaja krađe podataka do eskalacije privilegija do presretanja prometa. Ideja je ovdje odrediti koliko štete mogu prouzročiti. Nakon uspješnog dobivanja pristupa, još jedna metrika od interesa je vidjeti koliko dugo ispitivač može zadržati svoj pristup unutar sustava. Ako hakeri mogu zadržati pristup sustavu kroz dulje vremensko razdoblje, to im daje više mogućnosti za pustošenje i prikupljanje vrijednih osjetljivih podataka.

Analiza - nakon dovršetka aktivnosti testiranja, penetracijski testeri analizirat će svoje rezultate i izraditi izvješće u kojem će biti prikazana njihova analiza. Ovo izvješće pružit će djelotvoran uvid u ranjivosti, stvarnu mogućnost iskorištavanja i priliku za tvrtke da poduzmu potrebne radnje za ispravljanje prije nego što pravi haker dobije priliku iskoristiti njihov sustav.

Posljednji korak u penetracijskom testiranju jest davanje izvješća s analizom. Uključuje sljedeće ključne stavke:

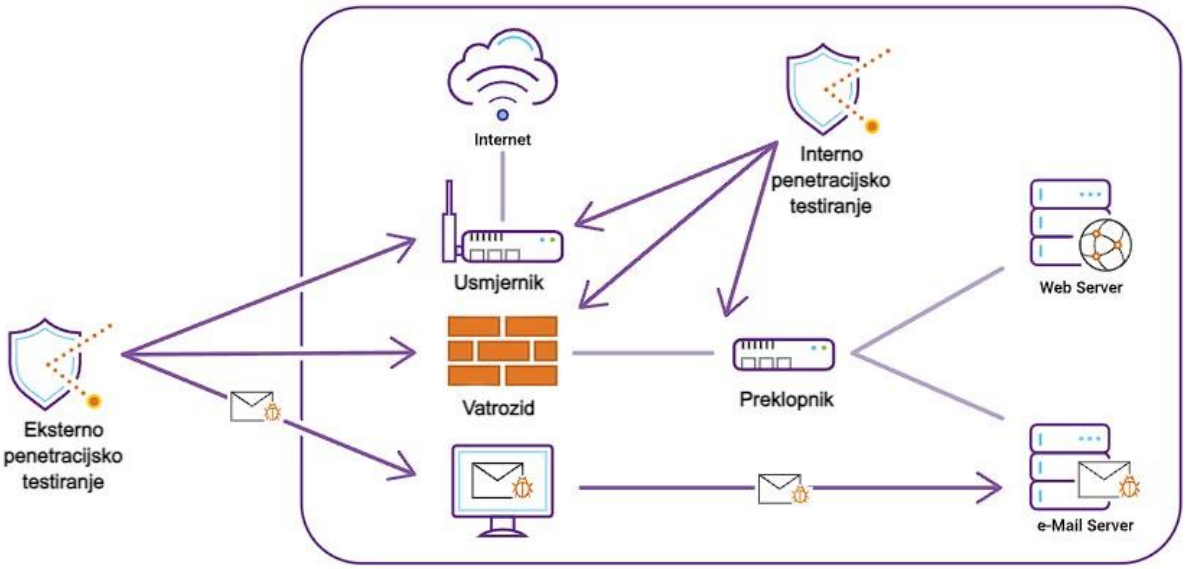
Izvršni sažetak - ovaj sažetak trebao bi ponuditi sažeti opis poslovnog rizika i ukupnog utjecaja pronađenih ranjivosti na poslovanje. Pružanjem laičke i pristupačne analize trenutnog stanja sigurnosti, laički dionici mogu lako razumjeti svoje cjelokupno sigurnosno stanje i lakše pružiti potrebnu podršku.

Analiza rizika - ovaj odjeljak trebao bi proći kroz analizu rizika, pružajući detaljnu analizu otkrivenih rizika i njihovih implikacija.

Analiza utjecaja - detaljan opis koliko je vjerojatno da će se otkrivene ranjivosti iskoristiti i koliko bi razoran/široko rasprostranjen utjecaj bio da se one stvarno iskoriste.

Preporuke za sanaciju – nude se sljedeći koraci koje tvrtka može poduzeti kako bi popravila otkrivene ranjivosti i slabosti.

Sveobuhvatna korist od implementacije penetracijskog testiranja mreže je u tome što omogućuje tvrtki da stekne vrijedan uvid u svoje cjelokupno sigurnosno stanje i daje joj snagu da poduzme mjere za rješavanje problema prije nego što zlonamjerni akter dobije priliku iskoristiti njezine slabosti, [9].



Slika 4: Prikaz penetracijskog testiranja mreže
Izvor:[9]

2.5.2. Penetracijsko testiranje web aplikacije

Penetracijsko testiranje web aplikacije je praksa simulacije napada na sustav u pokušaju pristupa osjetljivim podacima, sa svrhom utvrđivanja je li sustav siguran. Ovi se napadi izvode interno ili eksterno na sustav i pomažu u pružanju informacija o ciljnom sustavu, identificiranju ranjivosti unutar njega i otkrivanju eksploatacija koje bi zapravo mogle ugroziti sustav. To je ključna zdravstvena provjera sustava koja obavještava testere jesu li potrebne sanacije i sigurnosne mjere.

Postoji nekoliko prednosti uključivanja penetracijskog testiranja web aplikacije u sigurnosni program: Pomaže u zadovoljavanju zahtjeva usklađenosti to jest penetracijsko testiranje izričito je potrebno u nekim industrijama, a izvođenje penetracijskog testiranja web aplikacija pomaže u ispunjavanju zahtjeva usklađenosti. Pomaže u procjeni infrastrukture to jest infrastruktura, poput vatrozida i sustav domenskih imena (Domain Name System - DNS) poslužitelja, je javna. Sve promjene u infrastrukturi mogu učiniti sustav ranjivim. Penetracijsko testiranje web aplikacije pomaže identificirati napade iz stvarnog svijeta koji bi mogli uspjeti pristupiti tim sustavima. Identificira ranjivosti i pomaže u potvrđivanju sigurnosnih pravila.

Postoje tri koraka za izvođenje penetracijskog testiranja na web aplikacijama:

1. Konfiguriranje testova - prije početka, važno je definirati opseg i ciljeve projekta testiranja. Određivanje je li cilj ispunjavanje potreba usklađenosti ili provjera ukupne izvedbe bude usmjerilo ispitivača prema testovima koje treba provesti. Nakon što ispitivač odluči cilj testiranja, potrebno je prikupiti ključne informacije koje su potrebne za izvođenje testova. To uključuje mrežnu arhitekturu, informacije o stvarima kao što su aplikacijsko programsko sučelje (Application Programming Interface - API) i opće informacije o infrastrukturi.
2. Izvršenje testova - obično će testovi biti simulirani napadi koji pokušavaju vidjeti može li haker doista pristupiti aplikaciji. Dvije ključne vrste testova koji se mogu pokrenuti uključuju:
    - Eksterne penetracijske testove koji analiziraju komponente dostupne hakerima putem interneta, poput web-aplikacija ili web-mjesta
    - Interne testove prodora koji simuliraju scenarij u kojem haker ima pristup aplikaciji iza tvrtkinog vatrozida
3. Analiza testova - nakon završetka testiranja potrebno je analizirati rezultate. Treba raspraviti o ranjivostima i izloženosti osjetljivim podacima. Nakon analize mogu se implementirati potrebne promjene i poboljšanja, [10].

2.5.3. Penetracijsko testiranje na strani klijenta

Penetracijsko testiranje na strani klijenta koristi se za otkrivanje ranjivosti ili sigurnosnih slabosti u aplikacijama na strani klijenta. To mogu biti programi ili aplikacije kao što su Putty, klijenti e-pošte, web preglednici (npr. Chrome, Firefox, Safari itd.), Macromedia Flash i drugi. Programi poput Adobe Photoshop-a i Microsoft Office Suite-a također su podložni testiranju. Testovi na strani klijenta provode se kako bi se identificirali specifični kibernetički napadi:

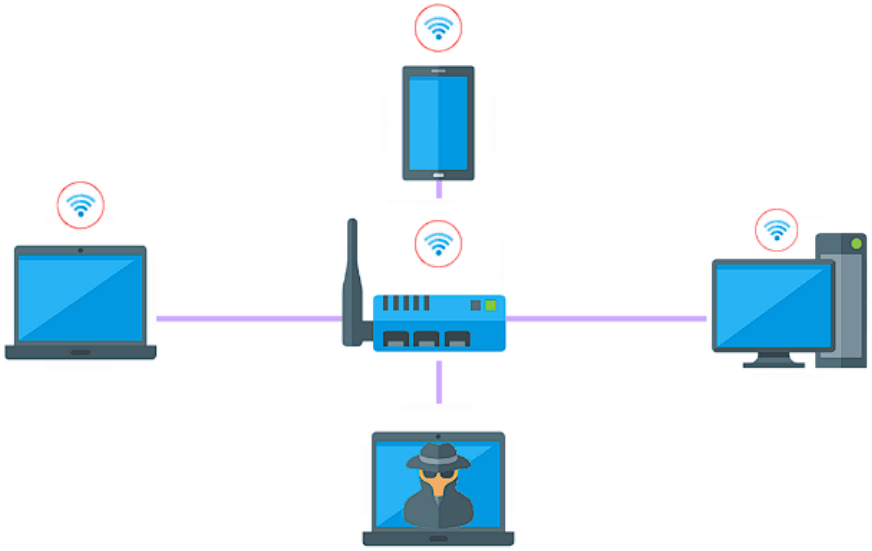
- *Cross-Site skripting* napadi
- *Clickjacking* napadi
- Dijeljenje izvora s više izvora (Cross-Origin Resource Sharing - CORS)
- Otimanje obrazaca
- HTML (HyperText Markup Language) injekcija
- Otvoreno preusmjeravanje
- Infekcija zlonamjernim softverom (engl. *Malware*).

2.5.4. Bežično penetracijsko testiranje

Bežično penetracijsko testiranje uključuje prepoznavanje i ispitivanje veza između svih uređaja povezanih na WiFi tvrtke. To su uređaji poput prijenosnog računala, tableti, pametni telefoni i svi ostali uređaji interneta stvari (Internet of Things - IoT). Bežično penetracijsko testiranje obično se izvodi na licu mjesta jer ispitivač mora biti u dometu bežičnog signala da bi mu pristupio. Alternativno, sljedeća jedinica računanja (Next Unit of Computing - NUC) i WiFi ananas (engl. *Pineapple*) mogu se postaviti na licu mjesta za daljinsko izvođenje testa.

Bežične komunikacije nevidljivo su pokrenute usluge koje omogućuju protok podataka u i iz mreže. Stoga se ova bežična mreža mora zaštititi od bilo kakvih slabosti poput neovlaštenog pristupa ili curenja podataka. Prije izvođenja bežičnog penetracijskog testa treba se razmotriti sljedeće:

- Jesu li sve pristupne točke identificirane i koliko njih koristi loše metode šifriranja?
- Jesu li podaci koji ulaze i izlaze iz mreže šifrirani i ako jesu, na koji način?
- Postoje li nadzorni sustavi za prepoznavanje neovlaštenih korisnika?
- Postoji li ikakva mogućnost da je informacijska tehnologija (Information technology - IT) tim krivo konfigurirao ili duplicirao bežičnu mrežu?
- Koje su trenutne mjere za zaštitu bežične mreže?
- Koriste li sve bežične pristupne točke Wi-Fi zaštićeni pristup (Wi-Fi Protected Access - WPA) protokol, [1]?



Slika 5: Prikaz bežičnog penetracijskog testiranja

Izvor:[11]

2.5.5. Penetracijsko testiranje društvenog inženjeringa

Penetracijsko testiranje društvenog inženjeringa usredotočeno je na ljude i procese te ranjivosti povezane s njima. Cilj ovog testa je identificirati slabosti u osobi, grupi ljudi ili procesu i identificirati ranjivosti s jasnim putem za ispravljanje. Uobičajene vrste napada društvenim inženjeringom koje koriste ispitivači uključuju:

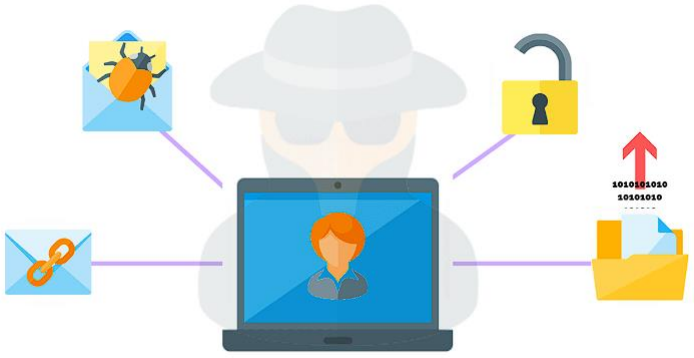
Phishing je metoda koja se javlja putem e-pošte i pokušava prevariti korisnika da oda osjetljive podatke ili otvori zlonamjernu datoteku koja može zaraziti njihov uređaj. *Vishing* je sličan *phishingu*, ali se javlja putem telefonskih poziva. Ovi telefonski pozivi pokušavaju prevariti korisnika da oda osjetljive informacije. *Smishing* je sličan *phishing-u*, ali se javlja putem SMS poruka. Ove tekstualne poruke imaju istu namjeru kao *phishing*.

Lažno predstavljanje je metoda kojom napadač pokušava prevariti osobu da povjeruje da je netko drugi. Na primjer, napadač bi mogao lažno predstavljati rukovoditelja s ciljem uvjeravanja zaposlenika da izvrše financijska plaćanja fiktivnim dobavljačima ili da dopuste pristup povjerljivim informacijama. Napad lažnim predstavljanjem također može ciljati na korisnika s ciljem dobivanja pristupa njegovom računu. To se može postići zahtjevom za ponovno postavljanje lozinke bez da administrator potvrdi njihov identitet.

Ronjenje u kontejner je metoda pri kojoj napadač prolazi ne samo kroz smeće, već i kroz druge predmete koji su mu vidljivi, kao što su ljepljive bilješke i kalendari, kako bi dobio korisne informacije o osobi ili organizaciji.

Ispuštanje USB-a je metoda koja koristi zlonamjerne USB-ove ispuštene u zajedničke prostore u radnom prostoru. USB-ovi obično sadrže softver koji, kada se priključi, instalira zlonamjerni softver koji može omogućiti ulaz u sustav ili prenijeti datoteke s uobičajenim ekstenzijama datoteka.

Ispitivač bi trebao raditi penetracijski test društvenog inženjeringa zbog toga jer se korisnici obično nazivaju "najslabijom karikom" kada je u pitanju sigurnost, ali ti korisnici ipak imaju više od potrebnih dopuštenja za obavljanje svojih poslova. Stoga bi imalo smisla testirati samo te korisnike. Ovi testovi mogu pokazati tko je unutar tvrtke osjetljiv na napade. Penetracijski testovi socijalnog inženjeringa obično se provode na hibridni način kombinirajući testove na licu mjesta i izvan njega. Na slici 6 prikazano je penetracijsko testiranje društvenog inženjeringa.



Slika 6: Prikaz penetracijskog testiranja društvenog inženjeringa

Izvor:[12]

Testovi na licu mjesta koriste se za testiranje fizičke sigurnosti zgrade i važećih pravila, poput politike čiste radne stanice. Tipične metode napada koje se koriste za testiranje na licu mjesta su: Lažno predstavljanje, ronjenje u kontejneru i ispuštanje usb-a.

Testovi izvan mjesta koriste se za testiranje svijesti korisnika o sigurnosti tijekom njihovog uobičajenog dana. Tijekom ove vrste testa, ispitivač će istražiti tvrtku

i koristiti informacije koje su javno dostupne za testiranje tvrtke. Ovi se testovi provode na daljinu i obično se sastoje od sljedećih napada: *Vishing*, *Phishing* i *Smishing*, [12]

2.5.6. Fizičko penetracijsko testiranje

Fizičko penetracijsko testiranje procjenjuje rizik za organizaciju od napadača koji fizički provali. Fizičko penetracijsko testiranje je procjena fizičkih sigurnosnih kontrola organizacije. Fizičke sigurnosne kontrole uključuju brave, ograde, zaštitare, kamere i drugo. Tijekom fizičkog penetracijskog testiranja, vješti inženjer pokušat će zaobići sigurnosne kontrole i dobiti fizički pristup ograničenim područjima, identificirati osjetljive informacije i steći uporište na mreži. Neke organizacije koje su imale fizičko penetracijsko testiranje uključuje:

- Pružatelje komunalnih usluga koji žele procijeniti rizik za trafostanice ili ICS/SCADA sustave itd.
- Zdravstvene pozivne centre koji žele procijeniti mogu li se dobiti informacije o zdravlju korisnika.
- Organizacije koje žele opravdati nadogradnju svoje fizičke sigurnosti ili procijeniti učinkovitost nedavnih nadogradnji.
- Trgovce koji žele procijeniti rizik od napadača na lokaciji trgovine ili poslovnice.

Tehnike koje ispitivač može koristiti na fizičkom penetracijskom testiranju uključuje:

Kloniranje identifikacije radio frekvencije (Radio-frequency identification - RFID-a) – koristeći RFID-kloner, ispitivač će se pokušati približiti bedževima zaposlenika dovoljno blizu da ih pročita i kopira. Nakon što se dobije važeća kartica, ispitivač će je upotrijebiti za pokušaj pristupa objektu.

Traženje – traženje znači korištenje društvenog inženjeringa kako bi se pokušalo natjerati zaposlenika da ispitivaču pridrži otvorena vrata.

Zaobilaženje kontrola pristupa – mnogo puta se koriste druge tehnike za dobivanje pristupa kao što je puzanje ispod ili preko ograde, korištenje metalne šipke za posezanje ispod vrata i povlačenje ručke, itd.

Otključavanje – većina modernih vrata ima zaštitu koja otežava otvaranje brave i pristup. Međutim, često usluge uništavanja to ne posjeduju, a dobivanje pristupa spremnicima za uništavanje može biti relativno lako i korisno za napadača, [13].

3. Istraživanje različitih tehnika napada i metode zaštite

Kibernetički napad je svaki namjerni pokušaj krađe, razotkrivanja, izmjene, onemogućavanja ili uništavanja podataka, aplikacija ili druge imovine putem neovlaštenog pristupa mreži, računalnom sustavu ili digitalnom uređaju. Akteri prijetnji pokreću kibernetičke napade iz raznih razloga, od sitnih krađa do ratnih djela. Koriste se različitim taktikama, kao što su napadi zlonamjernim softverom, prijevare društvenog inženjeringa i krađa lozinki, kako bi dobili neovlašteni pristup svojim ciljnim sustavima.

Kibernetički napadi mogu poremetiti, oštetiti ili čak uništiti poduzeća. Prosječna cijena povrede podataka je 4,35 milijuna američkih dolara. Ova cijena uključuje troškove otkrivanja i reagiranja na kršenje, vrijeme prekida rada i izgubljeni prihod te dugoročnu reputacijsku štetu tvrtki i njezinom brendu.

Ali neki kibernetički napadi mogu biti znatno skuplji od drugih. Napadi ucjenjivačkog softvera (engl. *ransomware*) doveli su do isplate otkupnine u iznosu od čak 40 milijuna američkih dolara. Prijevare s kompromitiranom poslovnom e-poštom ukrale su čak 47 milijuna dolara od žrtava u jednom napadu. Kibernetički napadi koji kompromitiraju osobne podatke korisnika (Personal Identifying Information - PII) mogu dovesti do gubitka povjerenja korisnika, regulatornih kazni, pa čak i pravnog postupka. Prema jednoj procjeni, kibernetički kriminal će koštati svjetsko gospodarstvo 10,5 trilijuna američkih dolara godišnje do 2025.

Kibernetički kriminalci koriste mnoge sofisticirane alate i tehnike za pokretanje kibernetičkih napada na IT sustave poduzeća, osobna računala i druge mete, [14].

3.1. Zlonamjerni softveri

Zlonamjerni softver je softver koji može onesposobiti zaražene sustave. Neprijateljski, nametljiv i namjerno neugodan, zlonamjerni softver nastoji upasti, oštetiti ili onemogućiti računala, računalne sustave, mreže, tablete i mobilne uređaje, često preuzimanjem djelomične kontrole nad operacijama uređaja. Poput ljudske gripe, ometa normalno funkcioniranje sustava.

Motivi zlonamjernog softvera su različiti. Zlonamjerni softver može biti usmjeren na krađu novca, sabotiranje korisnikove sposobnosti da obavi posao, sabotiranje političkih protivnika ili samo hvalisanje napadača. Iako zlonamjerni softver ne može oštetiti fizički hardver sustava, on može ukrasti, šifrirati ili izbrisati podatke, promijeniti ili oteti osnovne funkcije računala i špijunirati aktivnosti računala bez korisnikovog znanja ili dopuštenja.

3.1.1. Simptomi zaraze zlonamjernim softverom

Zlonamjerni softver može se otkriti s mnogo različitih nenormalnih ponašanja. Evo nekoliko znakova koji pokazuju da se zlonamjerni softver nalazi u sustavu:

Računalo radi manjom brzinom - jedna od nuspojava zlonamjernog softvera je smanjenje brzine operativnog sustava (Operating System - OS), bilo da se korisnik kreće internetom ili samo koristi svoje lokalne aplikacije, korištenje resursa sustava čini se neuobičajeno visokim.

Zaslon je preplavljen oglasima - neočekivani skočni oglasi tipičan su znak zaraze zlonamjernim softverom. Posebno su povezani s oblikom zlonamjernog softvera poznatog kao reklamni softver(engl. *Adware*).

Često rušenje sustava - to može doći kao zamrzavanje ili BSOD (plavi ekran smrti), potonji se pojavljuje na Windows sustavima nakon što naiđe na kobnu pogrešku.

Gubitak prostora na disku - to bi moglo biti zbog napuhanog zlonamjernog softvera koji se skriva u tvrdom disku uređaja

Povećanje internetske aktivnosti sustava - za primjer se može uzeti Trojanac. Jednom kada Trojanac sleti na ciljano računalo, sljedeće što radi je da dopre do napadačevog naredbenog i kontrolnog poslužitelja kako bi preuzeo sekundarnu infekciju, često ucjenjivački softver. Zbog toga dolazi do porasta internetske aktivnosti. Isto vrijedi i za *botnet* mreže, špijunske softvere i sve druge prijetnje koje zahtijevaju povratnu komunikaciju s naredbenog i kontrolnog poslužitelja.

Antivirusni proizvod prestaje raditi i korisnik ga ne može ponovno uključiti, što ga ostavlja nezaštićenim od zlonamjernog softvera koji ga je onemogućio

Gubi se pristup vlastitim datotekama ili cijelom računalu - ovo je simptom zaraze ucjenjivačkog softvera. Hakeri se oglašavaju ostavljajući poruku o otkupnini na vašoj radnoj površini ili mijenjajući samu pozadinu radne površine u poruku o otkupnini- U bilješci počinitelji obično obavještavaju da svoju žrtvu da su podaci šifrirani i zahtijevaju plaćanje otkupnine u zamjenu za dekriptiranje njihovih datoteka.

3.1.2. Vrste zlonamjernih softvera

Reklamni softver je neželjeni softver dizajniran za izbacivanje oglasa na žrtvin zaslon, najčešće unutar web preglednika. Obično se koristi prikrivenom metodom kako bi se ili maskirao kao legitiman ili povezao s drugim programom kako bi prevario žrtvu da ga instalira na svoje računalo, tablet ili mobilni uređaj.

Špijunski softver (engl. *Spyware*) je zlonamjerni softver koji potajno promatra aktivnosti korisnika računala bez dopuštenja i prijavljuje ih autoru softvera.

Virus je zlonamjerni softver koji se pričvršćuje na drugi program i, kada se pokrene - obično nenamjerno od strane korisnika - replicira se mijenjajući druge računalne programe i inficirajući ih vlastitim dijelovima koda.

Crvi su vrsta zlonamjernog softvera slična virusima. Poput virusa, crvi se sami umnožavaju. Velika je razlika u tome što se crvi mogu sami širiti po sustavima, dok je virusima potrebna neka vrsta akcije korisnika kako bi pokrenuli infekciju.

Trojanac ili trojanski konj jedna je od najopasnijih vrsta zlonamjernog softvera. Obično se predstavlja kao nešto korisno kako bi prevario žrtvu. Nakon što je u žrtvinom sustavu, napadači koji stoje iza Trojanca dobivaju neovlašteni pristup pogođenom računalu. Odatle se Trojanci mogu koristiti za krađu financijskih podataka ili instaliranje drugih oblika zlonamjernog softvera, često ucjenjivačkog softvera.

Ucjenjivački softver je oblik zlonamjernog softvera koji žrtvi ne dopušta pristup njenom uređaju i/ili šifrira njene datoteke, a zatim ju prisiljava da plati otkupninu kako bi ponovno dobila pristup. Ucjenjivački softver se naziva odabranim oružjem kibernetičkih kriminalaca jer zahtijeva brzo, isplativo plaćanje u kriptovaluti kojoj je teško ući u trag. Kôd koji stoji iza ucjenjivačkog softvera lako je nabaviti putem mrežnih kriminalnih tržišta, a obrana od njega je vrlo teška. Dok su napadi ucjenjivačkog softvera na pojedinačne korisnike u ovom trenutku smanjeni, napadi na tvrtke porasli su za 365 posto u 2019.

Rootkit je oblik zlonamjernog softvera koji napadaču daje administratorske ovlasti na zaraženom sustavu, poznate i kao "*root*" pristup. Obično je također dizajniran da ostane skriven od korisnika, drugog softvera u sustavu i samog operativnog sustava.

Keylogger je zlonamjerni softver koji bilježi sve korisničke pritiske tipki na tipkovnici, obično pohranjuje prikupljene informacije i šalje ih napadaču koji traži osjetljive podatke poput korisničkih imena, lozinki ili podataka o kreditnoj kartici.

Zlonamjerno kripto rudarenje sve je raširenija zlonamjerni softver kojeg obično instalira Trojanac. Omogućuje nekom drugom korištenje žrtvinog računala za rudarenje kriptovalute poput Bitcoina.

Eksploatacije su vrsta zlonamjernog softvera koji iskorištava pogreške i ranjivosti u sustavu kako bi napadaču dao pristup žrtvinom sustavu. Dok je žrtva u sustavu, napadač bi mogao ukrasti njene podatke ili ispustiti neki oblik zlonamjernog softvera. Iskorištavanje nultog dana odnosi se na ranjivost softvera za koju trenutno ne postoji dostupna zaštita ili popravak.

3.1.3. Mjere zaštite od zlonamjernih softvera

Potrebno je obratiti pozornost na domenu i biti oprezan ako stranica nije domena najviše razine, npr. com, mil, net, org, edu ili biz itd. Potrebno je koristiti jake lozinke s više faktorskom autentifikacijom. Upravitelj lozinki ovdje može biti od velike pomoći. Potrebno je izbjegavanje klikanja na skočne oglase dok se pregledava internet. Potrebno je izbjegavati otvaranje privitaka e-pošte nepoznatih pošiljatelja. Ne smije se klikati na čudne, neprovjerene poveznice u e-pošti, tekstovima i

porukama na društvenim mrežama. Ne smije se preuzimati softver s nepouzdanih web stranica ili *peer-to-peer* mreža za prijenos datoteka. Potrebno se držati službenih aplikacija s Google Play-a i Apple App Store-a na Androidu, OSX-u i iOS-u (ne preporučuje se „*jailbreakirati*“ svoj telefon). Korisnici računala trebaju provjeriti ocjene i recenzije prije instaliranja bilo kojeg softvera. Potrebno je provjeravati jesu li operativni sustav, preglednici i dodaci zakrpani i ažurni. Potrebno je redovno brisanja programa koje se više ne koriste. Preporučuje se redovno sigurnosno kopiranje vlastitih podataka. Ako se datoteke oštete, šifriraju ili na drugi način budu nedostupne, bit će dostupna sigurnosna kopija. Potrebno je imati instalirani program za kibernetičku sigurnost koji aktivno skenira i blokira prijetnje na korisničkom uređaju poput Malwarebytes, NordLayer, McAfee itd., [15].

3.2. Napadi društvenog inženjeringa

Napadi društvenim inženjeringom brzo se povećavaju na današnjim mrežama i slabe lanac kibernetičke sigurnosti. Cilj im je manipulirati pojedincima i poduzećima kako bi otkrili vrijedne i osjetljive podatke u interesu cyber kriminalaca. Društveni inženjering dovodi u pitanje sigurnost svih mreža bez obzira na robusnost njihovih vatrozidova, metoda kriptografije, sustava za otkrivanje upada i antivirusnih softverskih sustava.

Ljudi će vjerojatnije vjerovati prije drugim ljudima u usporedbi s računalima ili tehnologijama. Stoga su oni najslabija karika u sigurnosnom lancu. Zlonamjerne aktivnosti koje se ostvaruju kroz ljudske interakcije psihički utječu na osobu da otkrije povjerljive informacije ili da probije sigurnosne procedure. Zbog ovih ljudskih interakcija, napadi društvenim inženjeringom najsnažniji su napadi jer prijete svim sustavima i mrežama. Ne mogu se spriječiti korištenjem softverskih ili hardverskih rješenja sve dok ljudi nisu obučeni za sprječavanje ovih napada. Cyber kriminalci biraju ove napade kada ne postoji način za hakiranje sustava bez tehničkih ranjivosti, [16].

3.2.1. Tehnike napada društvenim inženjeringom

Napadi društvenim inženjeringom dolaze u mnogo različitih oblika i mogu se izvesti bilo gdje je uključena ljudska interakcija. Vrste mogućih napada društvenim inženjeringom su:

Mamljenje (engl. *Baiting*) - kao što naziv implicira, napadi mamljenja koriste lažno obećanje da potaknu žrtvinu pohlepu ili znatiželju. Oni namamljuju korisnike u zamku koja krade njihove osobne podatke ili zarazi njihove sustave zlonamjernim softverom. Na primjer, napadači ostavljaju mamac – obično *flash* diskove zaražene zlonamjernim softverom - na vidljivim mjestima gdje će ih potencijalne žrtve sigurno vidjeti (npr. kupaonice, dizala, parkiralište ciljane tvrtke). Žrtve pokupe mamac iz znatiželje i ubace ga u radno ili kućno računalo, što rezultira automatskom instalacijom zlonamjernog softvera na sustav. Mrežni oblici mamljenja sastoje se od

primamljivih oglasa koji vode do zlonamjernih stranica ili potiču korisnike na preuzimanje aplikacije zaražene zlonamjernim softverom.

Softver zastrašivanja (engl. *Scareware*) - uključuje bombardiranje žrtava lažnim uzbunama i fiktivnim prijetnjama. Korisnici su prevareni misleći da je njihov sustav zaražen zlonamjernim softverom, što ih navodi da instaliraju softver od kojeg nema stvarne koristi (osim za počinitelja) ili je sam zlonamjerni softver. Uobičajen primjer softvera zastrašivanja su skočni transparenti legitimnog izgleda koji se pojavljuju unutar preglednika dok korisnik surfa mrežom, prikazujući tekst kao što je, "Vaše računalo je možda zaraženo štetnim špijunskim programima." Ili nudi instalaciju alata (često zaraženog zlonamjernim softverom) ili će usmjeriti korisnika na zlonamjerno mjesto gdje se njegovo računalo zarazi.

Pretexting - napadač dobiva informacije putem niza vješto smišljenih laži. Prijevaru često pokreće počinitelj koji se pretvara da treba osjetljive informacije od žrtve kako bi izvršio kritični zadatak. Napadač obično počinje uspostavljanjem povjerenja sa svojom žrtvom lažno se predstavljajući kao suradnik, policajac, bankovni i porezni službenik ili druge osoba koje imaju pravo znati tu informaciju. Napadač *pretexting-a* postavlja pitanja koja su potrebna za potvrdu identiteta žrtve, putem kojih prikuplja važne osobne podatke. Pomoću ove prijevare prikupljaju se sve vrste relevantnih informacija i zapisa, kao što su brojevi socijalnog osiguranja, osobne adrese i telefonski brojevi, telefonski zapisi, datumi godišnjeg odmora osoblja, bankovni zapisi.

Phishing - popularan tip napada društvenim inženjeringom putem e-pošte i tekstualnih poruka. Cilj je stvoriti hitnost, znatiželju ili strah kod žrtava, potičući ih na otkrivanje osjetljivih informacija, klikanje na zlonamjerne poveznice ili otvaranje zlonamjernih privitaka. Primjer je e-poruka koja upozorava korisnike na kršenje pravila i traži hitnu akciju poput promjene lozinke. Sadrži poveznicu na lažnu web stranicu s ciljem krađe vjerodajnica. Identifikacija i blokiranje takvih poruka lakši su za poslužitelje e-pošte s pristupom platformama za dijeljenje prijetnji.

Krađa identiteta - Ovo je ciljanija verzija *phishing* prijevare u kojoj napadač odabire određene pojedince ili tvrtke. Zatim kroje svoje poruke na temelju karakteristika, radnih mjesta i kontakata koji pripadaju njihovim žrtvama kako bi njihov napad bio manje uočljiv. Krađa identiteta zahtijeva puno više truda u ime počinitelja i mogu trajati tjedni i mjeseci da se postigne. Puno ih je teže otkriti i imaju bolje stope uspjeha ako rade vješto. Scenarij krađe identiteta mogao bi uključivati napadača koji, lažno predstavljajući se kao IT konzultanta organizacije, šalje e-poštu jednom ili više zaposlenika. Formulirana je i potpisana točno onako kako konzultant inače radi, čime se primatelji varaju da misle da je to autentična poruka. Poruka traži od primatelja da promijene lozinku i pruža im poveznicu koja ih preusmjerava na zlonamjernu stranicu gdje napadač tada preuzima njihove vjerodajnice.

3.2.2. Mjere zaštite od društvenog inženjeringa

Društveni inženjeri manipuliraju ljudskim osjećajima, poput znatiželje ili straha, kako bi izveli planove i uvukli žrtve u svoje zamke. Stoga je korisnik potreban biti oprezan svaki put kad ga uznemiri e-pošta, privuče ga ponuda prikazana na web stranici ili kad naiđe na zalutale digitalne medije koji lažu. Biti oprezan može pomoći da se zaštiti od većine napada društvenog inženjeringa koji se odvijaju u digitalnom svijetu.

Potrebno je ne otvarati e-poštu i privitke iz sumnjivih izvora – ako korisnik ne zna dotičnog pošiljatelja, on ne mora odgovoriti na e-poštu. Čak i ako ga korisnik poznaje ali sumnja u njihovu poruku, tada je potrebno provjeriti i potvrditi vijesti iz drugih izvora, poput telefona ili izravno sa stranice davatelja usluga. Potrebno je imati na umu da se adrese e-pošte cijelo vrijeme lažiraju; čak i e-poštu koja navodno dolazi iz pouzdanog izvora koju je možda zapravo pokrenuo napadač.

Potrebno je koristiti višefaktorsku autentifikaciju – jedna od najvrjednijih informacija koju traže napadači su korisničke vjerodajnice. Korištenje višefaktorske provjere autentičnosti pomaže u zaštiti korisničkog računa u slučaju kompromitacije sustava.

Potrebno je biti oprezan s primamljivim ponudama – ako ponuda zvuči previše primamljivo, potrebno je da korisnik dvaput razmisli prije nego što je prihvate kao činjenicu.

Potrebno je redovno ažuriranje svog antivirusnog softvera – korisnik bi trebao provjeriti jesu li uključena automatska ažuriranja ili bi korisniku trebala postati navika da svaki dan prvo preuzima najnovija ažuriranja. Potrebno je redovno provjeravati jesu li ažuriranja primijenjena i potrebno je skenirati svoj sustav radi mogućih infekcija, [17].

3.3. Distribuirani napadi uskraćivanja usluge (DDoS)

Distribuirani napad uskraćivanja usluge (DDoS) zlonamjeran je pokušaj prekida normalnog prometa ciljanog poslužitelja, usluge ili mreže preplavljivanjem cilja ili njegove okolne infrastrukture s poplavom internetskog prometa. DDoS napadi postižu učinkovitost korištenjem više kompromitiranih računalnih sustava kao izvora prometa napada. S visoke razine, DDoS napad je poput neočekivane prometne gužve koja začepljuje autocestu, sprječavajući redoviti promet da stigne na svoje odredište, [42].

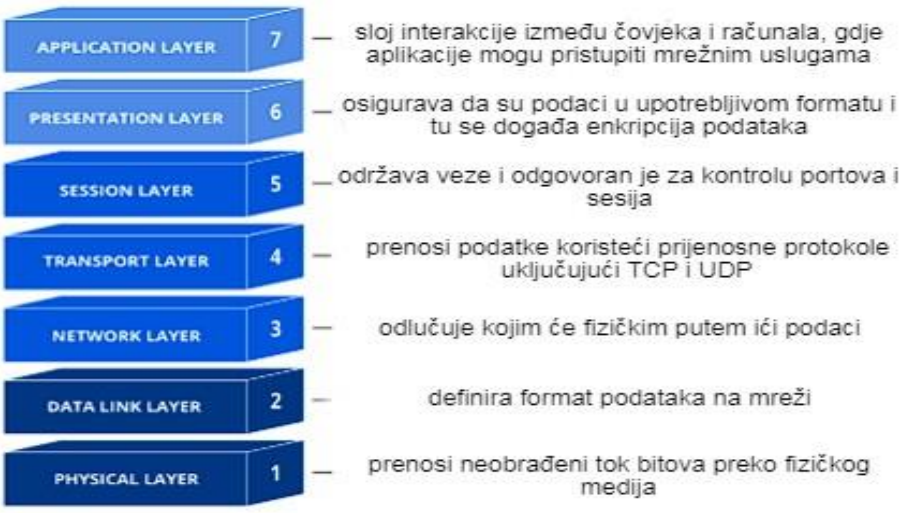
DDoS napadi se izvode s mrežama računala povezanih s internetom. Te se mreže sastoje od računala i drugih uređaja koji su zaraženi zlonamjernim softverom, što omogućuje da ih napadač kontrolira na daljinu. Ti pojedinačni uređaji nazivaju se *botovi* (ili zombiji), a skupina botova naziva se *botnet*. Nakon što je *botnet* uspostavljen, napadač može usmjeriti napad slanjem uputa svakom *botu*. Kada je žrtvin poslužitelj ili mreža na meti *botneta*, svaki *bot* šalje zahtjeve na ciljnu IP adresu,

potencijalno uzrokujući preopterećenje poslužitelja ili mreže, što rezultira uskraćivanjem usluge normalnom prometu. Budući da je svaki *bot* legitiman internetski uređaj, odvajanje napadačkog prometa od normalnog prometa može biti teško. Najočitiji simptom DDoS napada je iznenadna sporost ili nedostupnost stranice ili usluge. Alati za analizu prometa mogu uvelike pomoći da se uoči neke od ovih znakova DDoS napada:

- Sumnjive količine prometa koje potječu s jedne IP adrese ili IP raspona
- Poplava prometa od korisnika koji dijele jedan profil ponašanja, kao što je vrsta uređaja, geolokacija ili verzija web preglednika
- Neobjašnjivi porast zahtjeva prema jednoj stranici ili krajnjoj točki
- Čudni obrasci prometa kao što su skokovi u neparne sate dana ili obrasci koji se čine neprirodnim (npr. skokovi svakih 10 minuta)

3.3.1. Vrste DDoS napada

Različite vrste DDoS napada ciljaju različite komponente mrežne veze. Da bi se razumjeli kako funkcioniraju različiti DDoS napadi, potrebno je znati kako se uspostavlja mrežna veza. Mrežna veza na Internetu sastoji se od mnogo različitih komponenti ili "slojeva". OSI model, prikazan na slici 7, konceptualni je okvir koji se koristi za opisivanje mrežne povezanosti u 7 različitih slojeva.



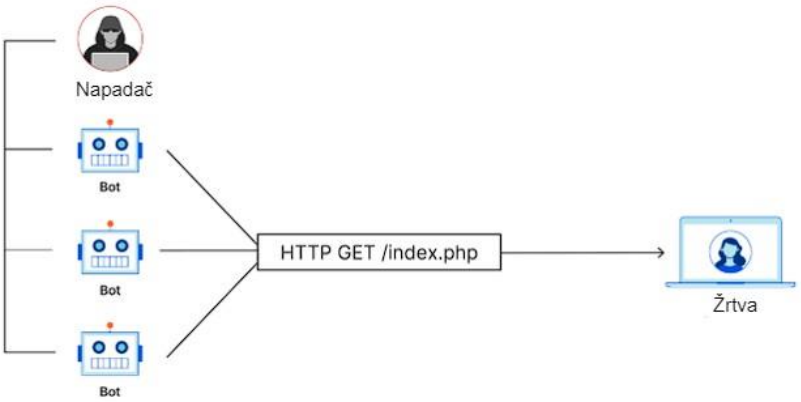
Slika 7: Prikaz OSI modela

Izvor:[19]

Iako gotovo svi DDoS napadi uključuju zatrpavanje ciljanog uređaja ili mreže prometom, napadi se mogu podijeliti u tri kategorije. Napadač može upotrijebiti jedan ili više različitih vektora napada ili ciklički vektore napada kao odgovor na protumjere koje poduzima meta.

3.3.1.1. Napad aplikacijskog sloja

Cilj ovih napada je iscrpiti resurse i uzrokovati uskraćivanje usluge. Napadi ciljaju sloj koji generira web stranice na poslužitelju kao odgovor na HTTP zahtjeve. Iako je izvršavanje HTTP zahtjeva na strani klijenta jeftino, odgovor poslužitelja može biti skup jer učitava datoteke i pokreće baze podataka. Obrana od napada na sloj 7 teška je jer je teško razlikovati zlonamjerni od legitimnog prometa. Primjer napada na aplikacijski sloj prikazan je na slici 8. Napad je sličan konstantnom osvježavanju u web pregledniku s više računala odjednom, stvarajući veliki broj HTTP zahtjeva i uzrokujući uskraćivanje usluge. Napadi se kreću od jednostavnih do složenih verzija. Jednostavniji napadi koriste isti URL i iste napadačke IP adrese, referere i korisničke agente. Složenije verzije koriste više napadačkih IP adresa i nasumično ciljaju različite URL-ove s različitim refererima i korisničkim agentima.



Slika 8: Prikaz primjera napada na aplikacijski sloj

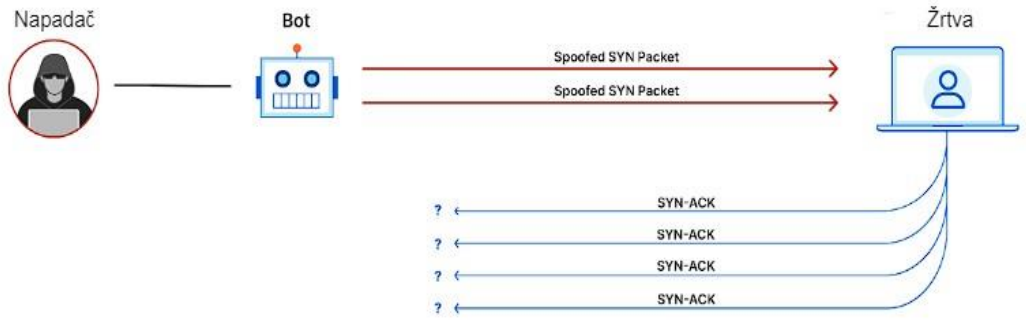
Izvor:[19]

3.3.1.2. Napadi na protokol

Napadi na protokol, također poznati kao napadi iscrpljivanja stanja, uzrokuju prekid usluge prekomjernom potrošnjom resursa poslužitelja i/ili resursa mrežne opreme poput vatrozida i uravnoteženja opterećenja. Napadi na protokol koriste slabosti u sloju 3 i sloju 4 skupa protokola kako bi metu učinili nedostupnom.

Na slici 9 je vidljiv prikaz primjera napada na protokol. SYN poplava je analogna radniku u sobi za opskrbu koji prima zahtjeve s prednje strane trgovine. Radnik prima zahtjev, odlazi po paket i čeka potvrdu prije nego što donese paket ispred. Radnik tada dobiva mnogo više zahtjeva za pakete bez potvrde sve dok više ne može nositi pakete, postane preopterećen i zahtjevi počnu ostajati bez odgovora. Ovaj napad iskorištava protokol kontrole prijenosa (Transmission Control Protocol - TCP rukovanje) — slijed komunikacije kojim dva računala započinju mrežnu vezu — šaljući meti veliki broj TCP “Initial Connection Request” SYN paketa s lažiranim izvornim IP adresama. Ciljni stroj odgovara na svaki zahtjev za povezivanjem i zatim

čeka posljednji korak u rukovanju, koji se nikada ne dogodi, iscrpljujući ciljne resurse u procesu.



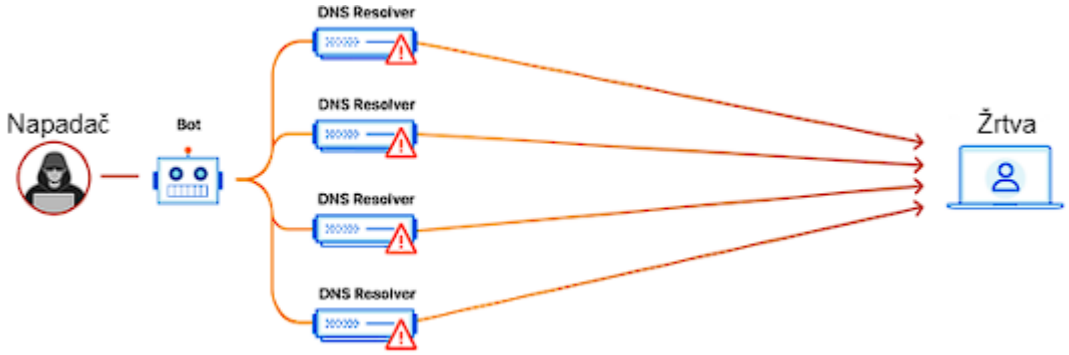
Slika 9: Prikaz primjera napada na protokol

Izvor:[19]

3.3.1.3. Volumetrijski napadi

Pokušava stvoriti zagušenje trošenjem cijele dostupne propusnosti između mete i interneta. Velike količine podataka šalju se meti korištenjem oblika pojačanja ili drugog načina stvaranja masovnog prometa, kao što su zahtjevi s *botnet* mreže.

Na slici 10 prikazan je primjer volumetrijskog napada. Pojačavanje DNS-a je kao kad bi netko nazvao restoran i rekao "Hoću po jedno od svega, molim vas nazovite me i ponovite moju cijelu narudžbu", pri čemu broj povratnog poziva zapravo pripada žrtvi. Uz vrlo malo truda, generira se dugačak odgovor i šalje žrtvi. Upućivanjem zahtjeva otvorenom DNS poslužitelju s lažiranom IP adresom (IP adresa žrtve), ciljana IP adresa tada prima odgovor od poslužitelja.



Slika 10: Prikaz primjera volumetrijskog napada

Izvor:[19]

3.3.2. Postupci za ublažavanje DDoS napada

Ključna briga u ublažavanju DDoS napada je razlika između napadačkog prometa i normalnog prometa. Na primjer, kad se pusti novi proizvod u promet tada je web stranica tvrtke preplavljena željnim kupcima, prekid tog cjelokupnog prometa bi bila pogreška. Ako ta tvrtka iznenada ima porast prometa od poznatih napadača, vjerojatno su potrebni napori za ublažavanje napada. Poteškoća leži u razlikovanju stvarnih kupaca od napadačkog prometa. DDoS napad s više vektora koristi višestruke putove napada kako bi nadvladao metu na različite načine, potencijalno ometajući napore ublažavanja na bilo kojoj putanji. Napad koji cilja na više slojeva skupa protokola u isto vrijeme, kao što je pojačanje DNS-a u kombinaciji s HTTP poplavom primjer je viševektorskog DDoS-a. Ublažavanje viševektorskog DDoS napada zahtijeva niz strategija za suzbijanje različitih putanja. Općenito govoreći, što je napad složeniji, veća je vjerojatnost da će promet napada biti teško odvojiti od normalnog prometa - cilj napadača je uklopiti se što je više moguće, čineći napore za ublažavanje što je moguće neučinkovitijima. Pokušaji ublažavanja koji uključuju ispuštanje ili ograničavanje prometa bez razlike mogu izbaciti dobar promet zajedno s lošim, a napad se također može modificirati i prilagoditi zaobilaženju protumjera.

3.3.2.1. Usmjeravanje crne rupe

Jedno rješenje koje je dostupno gotovo svim mrežnim administratorima je stvaranje rute crne rupe i usmjeravanje prometa na tu rutu. U svom najjednostavnijem obliku, kada se filtriranje crne rupe implementira bez posebnih kriterija ograničenja, i legitimni i zlonamjerni mrežni promet usmjeravaju se na nultu rutu ili crnu rupu i ispuštaju iz mreže. Ako neka internetska imovina doživi DDoS napad, davatelj internetske usluge (Internet Service Provider - ISP) od te imovine može poslati sav promet web stranice u crnu rupu kao obranu. Ovo nije idealno rješenje, jer učinkovito daje napadaču željeni cilj: čini mrežu nedostupnom.

3.3.2.2. Ograničenje brzine

Ograničavanje broja zahtjeva koje će poslužitelj prihvatiti tijekom određenog vremenskog okvira također je način ublažavanja napada uskraćivanjem usluge. Dok je ograničenje brzine korisno za usporavanje web strugača od krađe sadržaja i za ublažavanje pokušaja prijave korištenjem brutalne sile, ono samo po sebi vjerojatno neće biti dovoljno da se učinkovito nosi sa složenim DDoS napadom. Unatoč tome, ograničavanje brzine korisna je komponenta u učinkovitoj strategiji ublažavanja DDoS napada.

3.3.2.3. Vatrozid web aplikacije

Vatrozid web aplikacije (Web Application Firewall - WAF) alat je koji može pomoći u ublažavanju DDoS napada sloja 7. Postavljanjem WAF-a između Interneta i izvornog poslužitelja, WAF može djelovati kao obrnuti proxy, štiteći ciljani poslužitelj od određenih vrsta zlonamjernog prometa. Filtriranjem zahtjeva na temelju niza pravila koja se koriste za identifikaciju DDoS alata, napadi sloja 7 mogu se spriječiti.

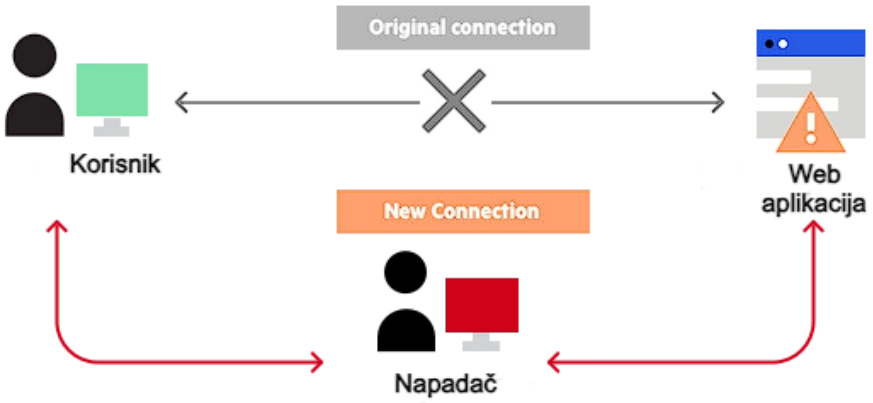
Jedna ključna vrijednost učinkovitog WAF-a je mogućnost brze implementacije prilagođenih pravila kao odgovor na napad.

3.3.2.4. Anycast mrežna difuzija

Ovaj pristup ublažavanja koristi Anycast mrežu za raspršivanje napadačkog prometa preko mreže distribuiranih poslužitelja do točke u kojoj mreža apsorbira promet. Poput usmjeravanja brze rijeke niz odvojene manje kanale, ovaj pristup širi utjecaj distribuiranog napadačkog prometa do točke u kojoj postaje upravljiv, raspršujući sve ometajuće sposobnosti. Pouzdanost Anycast mreže za ublažavanje DDoS napada ovisi o veličini napada te veličini i učinkovitosti mreže, [19].

3.4. Napad čovjek u sredini (MITM)

Napad čovjek u sredini opći je izraz za kada se počinitelj pozicionira u razgovoru između korisnika i aplikacije — ili da prisluškuje ili da se predstavlja kao jedna od strana, čineći se kao da je normalna razmjena informacija u tijeku. Cilj napada je ukrasti osobne podatke, kao što su vjerodajnice za prijavu, podaci o računu i brojevi kreditnih kartica. Ciljevi su obično korisnici financijskih aplikacija, softver kao usluga (Software as a service – SaaS) poslovanja, web stranica za e-trgovinu i drugih web stranica na kojima je potrebna prijava. Informacije dobivene tijekom napada mogu se koristiti u mnoge svrhe, uključujući krađu identiteta, neodobrene prijenose sredstava ili nedopuštenu promjenu lozinke. Na slici 11 prikazan je primjer napada čovjek u sredini.



Slika 11: Prikaz primjera napada čovjek u sredini

Izvor:[20]

Uspješno izvršenje napada čovjek u sredini ima dvije različite faze: presretanje i dešifriranje.

Prvi korak napadača jest da presretne korisnički promet kroz svoju mrežu prije nego što stigne do željenog odredišta. Najčešći (i najjednostavniji) način za to je pasivni napad u kojem napadač čini besplatne, zlonamjerne WiFi pristupne točke dostupnima javnosti. Obično su imenovani na način koji odgovara njihovoj lokaciji i

nisu zaštićeni lozinkom. Jednom kada se žrtva spoji na takvu žarišnu točku, napadač dobiva punu vidljivost bilo koje internetske razmjene podataka.

Napadači koji žele aktivnije pristupiti presretanju mogu pokrenuti jedan od sljedećih napada:

- IP *spoofing* - uključuje napadača koji se maskira u aplikaciju mijenjanjem zaglavlja paketa u IP adresi. Kao rezultat toga, korisnici koji pokušavaju pristupiti URL-u povezanom s aplikacijom šalju se na web stranicu napadača.
- Protokol za rješavanje adresa (Address Resolution Protocol – ARP) *spoofing* - proces povezivanja MAC adrese napadača s IP adresom legitimnog korisnika na lokalnoj mreži pomoću lažnih ARP poruka. Kao rezultat toga, podaci koje korisnik šalje na IP adresu glavnog računala umjesto toga prenose se napadaču.
- DNS *spoofing* - uključuje infiltraciju u DNS poslužitelj i mijenjanje zapisa adrese web stranice. Kao rezultat toga, korisnici koji pokušavaju pristupiti stranici poslani su od izmijenjenog DNS zapisa prema napadačevoj stranici.

Nakon presretanja, svaki dvosmjerni SSL promet treba dešifrirati bez upozorenja korisnika ili aplikacije. Postoji niz metoda da se to postigne:

- HTTPS spoofing - šalje lažni certifikat žrtvinom pregledniku nakon što se napravi početni zahtjev za povezivanje sa sigurnim mjestom. Sadrži digitalni otisak prsta povezan s ugroženom aplikacijom, koji preglednik provjerava prema postojećem popisu pouzdanih stranica. Napadač tada može pristupiti svim podacima koje je žrtva unijela prije nego što se proslijede aplikaciji.
- SSL BEAST (iskorištavanje preglednika protiv SSL/TLS-a) cilja na TLS verziju 1.0 ranjivost u SSL-u - ovdje je žrtvino računalo zaraženo zlonamjernim JavaScriptom koji presreće šifrirane kolačiće koje šalje web aplikacija. Tada je ulančavanje blokova šifre (CBC) aplikacije ugroženo kako bi se dešifrirali njezini kolačići i autentifikacijski tokeni.
- Do otmice SSL-a dolazi kada napadač proslijedi krivotvorene autentifikacijske ključeve i korisniku i aplikaciji tijekom TCP rukovanja. Time se uspostavlja ono što se čini sigurnom vezom dok, zapravo, čovjek u sredini kontrolira cijelu sesiju.
- SSL uklanjanje smanjuje HTTPS vezu na HTTP presretanjem TLS provjere autentičnosti koju aplikacija šalje korisniku. Napadač korisniku šalje ne kriptiranu verziju web-mjesta aplikacije dok održava zaštićenu sesiju s aplikacijom. U međuvremenu, cijela korisnikova sesija je vidljiva napadaču, [20].

3.5. XSS napadi

XSS napad je iskorištavanje (engl. *exploit*) gdje napadač prilaže kod na legitimnu web stranicu koji će se izvršiti kada žrtva učita web stranicu. Taj se zlonamjerni kod može umetnuti na nekoliko načina. Najčešće se dodaje na kraj URL-

a ili se postavlja izravno na stranicu koja prikazuje sadržaj koji su generirali korisnici. Na slici 12 prikazan je primjer XSS napada.

Slika 12: Prikaz primjera XSS napada

Izvor:[21]

Jedan primjer napada skriptiranjem između web-mjesta često se može vidjeti na web-mjestima koja imaju neprovjerene forume za komentare. U ovom slučaju, napadač će objaviti komentar koji se sastoji od izvršnog koda omotanog u oznake '<script></script>'. Ove oznake govore web-pregledniku da interpretira sve između oznaka kao JavaScript kôd. Nakon što se taj komentar nađe na stranici, kada bilo koji drugi korisnik učita tu web stranicu, zlonamjerni kod između oznaka skripte izvršit će njihov web preglednik, a oni će postati žrtva napada.

JavaScript XSS napadi su popularni jer JavaScript ima pristup nekim osjetljivim podacima koji se mogu koristiti za krađu identiteta i drugim zlonamjernim svrhama. Na primjer, JavaScript ima pristup kolačićima*, a napadač bi mogao upotrijebiti XSS napad kako bi ukrao korisnikove kolačiće i lažno ih predstavljao na mreži. JavaScript također može stvoriti HTTP zahtjeve koji se mogu koristiti za slanje podataka (kao što su ukradeni kolačići) natrag napadaču. Uz to, JavaScript na strani klijenta također može pomoći napadaču da dobije pristup API-jima koji sadrže geolokacijske koordinate, podatke web kamere i druge osjetljive informacije, [41].

Tipičan tok XSS napada na više web stranica je sljedeći:

1. Žrtva učitava web stranicu, a zlonamjerni kod kopira korisnikove kolačiće
2. Kod zatim šalje HTTP zahtjev web poslužitelju napadača s ukradenim kolačićima u tijelu zahtjeva.
3. Napadač zatim može upotrijebiti te kolačiće za lažno predstavljanje korisnika na toj web stranici u svrhu napada društvenog inženjeringa ili čak za pristup brojevima bankovnih računa ili drugim osjetljivim podacima.

3.5.1. Kod na strani klijenta

Kôd na strani klijenta (engl. *Client-side code*) je JavaScript kod koji se izvodi na korisnikovom računalu. Kod na strani klijenta obično je kod koji izvršava web preglednik nakon što preglednik učita web stranicu. To je u suprotnosti s kodom na strani poslužitelja koji se izvršava na web poslužitelju glavnog računala. Kod na strani klijenta vrlo je koristan s interaktivnim web stranicama; interaktivni sadržaj radi brže i pouzdanije budući da računalo korisnika ne mora komunicirati s web poslužiteljem svaki put kada dođe do interakcije. Igre temeljene na pregledniku jedna su od popularnih platformi za kod na strani klijenta, budući da kod na strani klijenta može osigurati nesmetan rad igre bez obzira na probleme s povezivanjem.

Kod koji pokreće klijentsku stranu vrlo je popularan u modernom web razvoju i koristi se na većini modernih web stranica. Budući da je kod na više web-mjesta glavna komponenta modernog weba, skriptiranje između web-mjesta postalo je jedna od najčešće prijavljenih ranjivosti u kibernetičkoj sigurnosti, a napadi skriptiranjem na više web-mjesta pogodili su glavne stranice kao što su YouTube, Facebook i Twitter.

3.5.2. Vrste XSS napada

Reflektirano skriptiranje između web-mjesta je najčešće korišten XSS napad. Uz reflektirani napad, zlonamjerni kod se dodaje na kraj URL-a web stranice; često će to biti legitimna, pouzdana web stranica. Kada žrtva učita ovu poveznicu u svoj web preglednik, preglednik će izvršiti kod umetnut u URL. Napadač obično koristi neki oblik društvenog inženjeringa kako bi prevario žrtvu da klikne na poveznicu.

Na primjer, korisnik može primiti legitimnu e-poštu koja tvrdi da dolazi od njegove banke. U e-poruci će se od njih tražiti da poduzmu neke radnje na web-mjestu banke i dati poveznicu. Veza bi izgledala otprilike ovako:

„http://legitamite-bank.com/index.php?user=<script>ovdje je loš kod!</script>“.

Trajno skriptiranje između web-mjesta Trajno skriptiranje između web-mjesta se događa na web-mjestima koja korisnicima omogućuju objavljivanje sadržaja koji će drugi korisnici vidjeti, kao što je, na primjer, forum za komentare ili web-mjesto društvenih medija. Ako web-mjesto ispravno ne provjerava valjanost unosa za sadržaj koji generiraju korisnici, napadač može umetnuti kod koji će preglednici drugih korisnika izvršiti kada se stranica učita. Na primjer, napadač može otići na internetsku stranicu za upoznavanje i staviti nešto poput ovoga u svoj profil:

"Bok! Moje ime je Marko, uživam u dugim šetnjama plažom i <script>zlonamjerni kod</script>"

Svaki korisnik koji pokuša pristupiti Markovom profilu postat će žrtva Markovog upornog napada skriptiranjem na više stranica.

3.5.3. Zaštita od XSS napada

Izbjegavanje HTML-a u unosima - jedan vrlo učinkovit način da se izbjegnu uporni napadi skriptiranjem na različitim mjestima jest spriječiti korisnike da objavljuju HTML u unosima obrasca. Postoje i druge opcije koje korisnicima omogućuju stvaranje bogatog sadržaja bez upotrebe HTML-a, kao što su oznake i ono što vidite je ono što dobivate (What You See Is What You Get - WYSIWYG) uređivači.

Validacijom unosa - validacija znači implementacija pravila koja sprječavaju korisnika da objavi podatke u obrazac koji ne zadovoljava određene kriterije. Na primjer, unos koji traži korisnikovo "Prezime" trebao bi imati pravila provjere valjanosti koja korisniku dopuštaju samo slanje podataka koji se sastoje od alfanumeričkih znakova. Pravila provjere također se mogu postaviti tako da odbiju sve oznake ili znakove koji se obično koriste u skriptiranju na više stranica, kao što su oznake “<script>”.

Sanitizacijom podataka - sanitizacija podataka je slična provjeri valjanosti, ali se događa nakon što su podaci već objavljeni na web poslužitelju, ali još prije nego što se prikažu drugom korisniku.

Poduzimanjem sigurnosnih mjera za kolačiće - web-aplikacije također mogu postaviti posebna pravila za rukovanje kolačićima koja mogu ublažiti krađu kolačića putem XSS napada. Kolačići se mogu vezati za određene IP adrese tako da im napadači skriptiranjem na različitim stranicama ne mogu pristupiti.

Postavljanjem WAF pravila - WAF se također može konfigurirati za provođenje pravila koja će spriječiti reflektirano skriptiranje između web-mjesta. Ova WAF pravila koriste strategije koje će blokirati čudne zahtjeve prema poslužitelju, uključujući napade skriptiranjem na različitim mjestima, [21].

3.6. Napadi SQL injekcijom

SQL injekcija (SQLi) sigurnosna je ranjivost na webu koja napadaču omogućuje miješanje u upite koje aplikacija postavlja svojoj bazi podataka. Općenito omogućuje napadaču pregled podataka koje inače ne može dohvatiti. To može uključivati podatke koji pripadaju drugim korisnicima ili bilo koje druge podatke kojima sama aplikacija može pristupiti. U mnogim slučajevima, napadač može modificirati ili izbrisati te podatke, uzrokujući stalne promjene sadržaja ili ponašanja aplikacije.

Uspješan napad SQL injekcijom može rezultirati neovlaštenim pristupom osjetljivim podacima, kao što su lozinke, podaci o kreditnoj kartici ili osobni podaci o korisniku. Mnoga kršenja podataka visokog profila posljednjih godina bila su rezultat napada SQL injekcijom, što je dovelo do oštećenja ugleda i regulatornih kazni. U nekim slučajevima, napadač može dobiti stalan stražnji ulaz (engl. *Backdoor*) u sustave organizacije, što dovodi do dugoročnog ugrožavanja koje može proći nezapaženo dulje vrijeme.

Većina ranjivosti SQL injekcije može se pronaći brzo i pouzdano pomoću alata poput „Burp Suite“ skenera za web ranjivosti. Napad SQL injekcije može se otkriti

ručno korištenjem sustavnog skupa testova protiv svake ulazne točke u aplikaciji. To obično uključuje:

- Slanje jednostrukog navodnika „ ' “ i traženje pogrešaka ili drugih anomalija.
- Podnošenje neke sintakse specifične za SQL koja procjenjuje osnovnu (izvornu) vrijednost ulazne točke i drugu vrijednost, te traženje sustavnih razlika u rezultirajućim odgovorima aplikacije.
- Podnošenje Booleovih uvjeta kao što su „OR 1=1“ i „OR 1=2“ i traženje razlika u odgovorima aplikacije.
- Podnošenje korisnih podataka osmišljenih za pokretanje vremenskih odgoda kada se izvršavaju unutar SQL upita i traženje razlika u vremenu potrebnom za odgovor.
- Podnošenje OAST korisnih podataka osmišljenih za pokretanje izvanpojasne mrežne interakcije kada se izvrši unutar SQL upita i praćenje svih rezultirajućih interakcija.

Većina ranjivosti SQL injekcije javlja se unutar „WHERE“ klauzule „SELECT“ upita. Ovu vrstu SQL injekcije općenito dobro razumiju iskusni testeri. Ali ranjivosti SQL injekcije se u načelu mogu pojaviti na bilo kojem mjestu unutar upita i unutar različitih vrsta upita. Najčešća druga mjesta na kojima dolazi do SQL injekcije su:

- U izjavama „UPDATE“, unutar ažuriranih vrijednosti ili klauzule „WHERE“.
- U izjavama „INSERT“, unutar umetnutih vrijednosti.
- U naredbama „SELECT“, unutar naziva tablice ili stupca.
- U naredbama „SELECT“, unutar klauzule „ORDER BY“.

3.6.1. Vrste SQL injekcije

Postoji širok izbor ranjivosti, napada i tehnika SQL ubacivanja koji se pojavljuju u različitim situacijama. Neki uobičajeni primjeri SQL ubacivanja uključuju: dohvaćanje skrivenih podataka, rušenje logike aplikacije, UNION napadi i slijepa SQL injekcija.

3.6.1.1. Dohvaćanje skrivenih podataka

Npr. Korisnik otvara aplikaciju za kupnju koja prikazuje proizvode u različitim kategorijama. Kada korisnik klikne na kategoriju „Darovi“, njegov preglednik zahtijeva URL: https://nesigurna-stranica.com/proizvodi?kategorija=Darovi.

To uzrokuje da aplikacija napravi SQL upit za dohvaćanje pojedinosti o relevantnim proizvodima iz baze podataka:

“ SELECT * FROM proizvodi WHERE kategorija = 'Darovi' AND pušten = 1“.

Ovaj SQL upit traži od baze podataka da vrati: sve detalje (*) iz tablice proizvoda gdje je kategorija „Darovi“ a „pušten“ je 1. Ograničenje „pušten = 1“ koristi se za skrivanje proizvoda koji nisu pušteni. Za neobjavljene proizvode, vjerojatno „pušten = 0“.

Aplikacija ne implementira nikakvu obranu od napada SQL injekcijom, tako da napadač može konstruirati napad poput: https://nesigurna-stranica.com/proizvodi?kategorija=Darovi'--.

To rezultira SQL upitom:

„SELECT * FROM proizvodi WHERE kategorija = 'Darovi'--' AND pušten = 1“.

Ključna stvar jest da je niz dvostrukih crtica -- indikator komentara u SQL-u i znači da se ostatak upita tumači kao komentar. Ovo učinkovito uklanja ostatak upita, tako da više ne uključuje „AND pušten = 1“. To znači da su svi proizvodi prikazani, uključujući proizvode koji nisu objavljeni.

Nadalje, napadač može uzrokovati da aplikacija prikaže sve proizvode u bilo kojoj kategoriji, uključujući kategorije za koje ne zna: „https://nesigurna-stranica.com/proizvodi?kategorija=Darovi +OR+1=1--„.

Ovo rezultira SQL upitom:

„SELECT * FROM proizvodi WHERE kategorija = 'Darovi'--' OR 1=1-- AND pušten = 1“.

Modificirani upit vratit će sve stavke gdje je ili kategorija Darovi ili je 1 jednako 1. Budući da je 1=1 uvijek istinito, upit će vratiti sve stavke.

3.6.1.2. Rušenje logike aplikacije

Razmatra se aplikacija koja korisnicima omogućuje prijavu s korisničkim imenom i lozinkom. Ako korisnik pošalje korisničko ime „marko95“ i lozinku fpz123, aplikacija provjerava vjerodajnice izvršavanjem sljedećeg SQL upita:

„SELECT * FROM korisnici WHERE korisnickoIme = 'marko95' AND lozinka = 'fpz123' “.

Ako upit vrati podatke o korisniku, tada je prijava uspješna. U protivnom se odbija. Ovdje se napadač može prijaviti kao bilo koji korisnik bez lozinke jednostavno pomoću niza SQL komentara -- kako bi uklonio provjeru lozinke iz WHERE klauzule upita. Na primjer, podnošenje korisničkog imena „administrator'—„ i prazne lozinke rezultira sljedećim upitom:

“SELECT * FROM korisnici WHERE korisnickoIme = 'administrator'--' AND password = ' ' “.

Ovaj upit vraća korisnika čije je korisničko ime administrator i uspješno prijavljuje napadača kao tog korisnika.

3.6.1.3. UNION napadi

U slučajevima kada se rezultati SQL upita vraćaju unutar odgovora aplikacije, napadač može iskoristiti ranjivost SQL ubacivanja kako bi dohvatio podatke iz drugih tablica unutar baze podataka. To se radi pomoću ključne riječi UNION, koja omogućuje izvršavanje dodatnog SELECT upita i dodavanje rezultata izvornom upitu. Na primjer, ako aplikacija izvrši sljedeći upit koji sadrži korisnički unos "Darovi": „SELECT ime, opis FROM proizvodi WHERE kategorija = 'Darovi' “.

Tada napadač može podnijeti unos:

„' UNION SELECT korisnickoIme, lozinka FROM korisnici--.“

To će uzrokovati da aplikacija vrati sva korisnička imena i lozinke zajedno s nazivima i opisima proizvoda.

3.6.1.4. Slijepa SQL injekcija

Mnoge instance SQL injekcije su slijepe ranjivosti. To znači da aplikacija ne vraća rezultate SQL upita ili pojedinosti o greškama baze podataka unutar svojih odgovora. Slijepe ranjivosti i dalje se mogu iskoristiti za pristup neovlaštenim podacima, ali uključene tehnike općenito su kompliciranije i teže ih je izvesti. Ovisno o ranjivosti i bazi podataka, sljedeće tehnike koriste se za slijepo SQL ubacivanje:

- Promjena upitne logike za pokretanje razlike u odgovoru, uključujući ubacivanje novih uvjeta ili uvjetno izazivanje pogreške (npr. dijeljenje s nulom).
- Uvjetno pokretanje vremenskog kašnjenja u obradi za zaključivanje istinitosti na temelju vremena odgovora aplikacije.
- Korištenje OAST tehnika za izvanpojasne mrežne interakcije, često omogućujući eks filtraciju podataka putem DNS pretraživanja za kontroliranu domenu.

3.6.2. Mjere zaštite od SQL injekcije

Većina slučajeva SQL injekcije se može spriječiti korištenjem parametriziranih upita (također poznatih kao pripremljene izjave) umjesto ulančavanja nizova unutar upita. Sljedeći navedeni kod je ranjiv na SQL ubacivanje jer je korisnički unos spojen izravno u upit:

„ String query = "SELECT * FROM proizvodi WHERE kategorija = '"+ input + " ' ";

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(query); „

Taj se kod može vrlo lako ispraviti na način koji sprječava da korisnički unos ometa strukturu upita:

„ PreparedStatement statement =connection.prepareStatement("SELECT * FROM proizvodi WHERE kategorija = ?");

Statement.setString(1, input);

ResultSet resultSet = statement.executeQuery();“

Parametrirani upiti mogu se koristiti za bilo koju situaciju u kojoj se nepouzdani unos pojavljuje kao podatak unutar upita, uključujući klauzulu WHERE i vrijednosti u INSERT ili UPDATE izjave. Ne mogu se koristiti za rukovanje nepouzdanim unosom u drugim dijelovima upita, kao što su nazivi tablica ili stupaca ili klauzula ORDER BY. Funkcionalnost aplikacije koja postavlja nepouzdane podatke u te dijelove upita morat će imati drugačiji pristup, kao što je stavljanje dopuštenih ulaznih vrijednosti na bijeli popis ili korištenje drugačije logike za isporuku traženog ponašanja.

Da bi parametrizirani upit bio učinkovit u sprječavanju ubacivanja SQL-a, niz koji se koristi u upitu mora uvijek biti tvrdo kodirana konstanta i nikada ne smije sadržavati nikakve varijabilne podatke iz bilo kojeg izvora. Ne smije se doći u iskušenje da se odlučuje od slučaja do slučaja je li neka stavka podataka pouzdana i potrebno je nastaviti koristiti ulančavanje nizova unutar upita za slučajeve koji se smatraju sigurnima. Previše je lako pogriješiti o mogućem podrijetlu podataka ili da promjene u drugom kodu krše pretpostavke o tome koji su podaci zaraženi, [22].

4. Primjena metodologije za penetracijsko testiranje u organizaciji

Uz rastući trend kibernetičkih napada u posljednjih nekoliko godina, bitno je da su organizacije svjesne ove prijetnje i da mogu identificirati ranjivosti u svojim sustavima. Kibernetičke prijetnje ne samo da postaju sve češće, već su i sve sofisticiranije. Najisplativiji način da se smanji rizik od kibernetičkih napada je penetracijsko testiranje.

Penetracijsko testiranje je proces testiranja računalnog sustava, mreže ili web aplikacije kako bi se pronašle ranjivosti koje bi napadač mogao iskoristiti. Smisao penetracijskog testa je identificirati potencijalne ranjivosti koje zlonamjerni korisnik može iskoristiti. Ideja je testirati slabosti koje bi zlonamjerni korisnik mogao iskoristiti, a ne administrator sustava. Penetracijsko testiranje nije jednokratna aktivnost. Umjesto toga, to je proces koji organizacija mora redovito provoditi. Učestalost testiranja ovisi o procjeni rizika i organizacijskoj strukturi organizacije.

Penetracijsko testiranje bitan je dio svake strategije kibernetičke sigurnosti. Penetracijsko testiranje pomaže u provjeri sigurnosti sustava, aplikacija i mreža organizacije. Koristi se za pronalaženje sigurnosnih slabosti prije kriminalaca. Ispitivači penetracije (ili "pentesteri") pokreću simulirane napade kako bi pronašli sigurnosne rupe. Taj proces pomaže organizaciji pronaći i popraviti nedostatke prije nego što ih kriminalac iskoristi. Penetracijsko testiranje pruža način testiranja učinkovitosti sigurnosnih kontrola sustava. Pomaže organizacijama da dizajniraju svoje sigurnosne procese i sigurnosne kontrole kako bi bile učinkovitije.

Postoje tri razloga zašto je penetracijsko testiranje bitno za organizacije a to su:

1. Sigurna infrastruktura - iznimno je važna za svaku organizaciju. Postoji mnogo načina testiranja sigurnosne infrastrukture, a jedan od najčešćih načina je penetracijsko testiranje. Penetracijsko testiranje pomaže u otkrivanju slabih točaka u aplikaciji ili mreži koje *cyber* kriminalci mogu lako iskoristiti.
2. Povjerenje kupaca i reputacija tvrtke - reputacija je sve. To je ono što pokreće svijet i to je glavni fokus većine organizacija. Reputacija organizacije može tu samu organizaciju uništiti ili doživjeti veliki uspjeh. Jednostavna vijest o curenju podataka iz tvrtke može uništiti svu reputaciju koju je ta tvrtka godinama gradila.
3. Učinkovite sigurnosne mjere i svijest o sigurnosti - sigurnost podataka organizacije je od najveće važnosti. Međutim, uvijek postoji opasnost od napada, bilo od strane zaposlenika koji prima mito za otkrivanje povjerljivih informacija ili od strane hakera, stoga je važno uvijek biti spreman. Penetracijsko testiranje je ne destruktivan način mapiranja potencijalnih sigurnosnih propusta prije nego što dođe do napada.

Povreda podataka može biti veliki problem za tvrtku, a posljedice mogu biti ogromne i utjecati na cijelu organizaciju. Tu su uključene financijske, pravne i reputacijske posljedice. Osim toga, izravne ekonomske posljedice također će proizaći iz troškova i implikacija povrede podataka. Financijske troškove povrede podataka neophodno je kvantificirati, ali oni su samo dio troška. Podmukliji učinak su izravni gubici koji nastaju zbog kršenja, poput smanjenog povjerenja potrošača, gubitka posla, regulatornih kazni, kazni, lažnih transakcija itd.

Mnogo je troškova povezanih s povredom podataka. Najizravniji od njih su troškovi povezani s istragom kršenja, obavještavanjem i sanacijom. To su troškovi koje često izravno snosi poduzeće. Kao što je pokazalo istraživanje IBM-a, ti troškovi nastavljaju rasti; Troškovi povrede podataka porasli su s 3,86 milijuna američkih dolara na 4,24 milijuna američkih dolara. Redoviti penetracijski testovi smanjuju šanse za upad u podatke održavajući aplikacije sigurnima.

Tvrtka bi trebala provoditi penetracijsko testiranje onoliko puta koliko ovisi o visini rizika. Organizacija koja nema osjetljive podatke na svojoj mreži može testirati jednom mjesečno, dok web-mjesto za e-trgovinu koja nosi visokorizičnu skupinu krađe informacija možda treba testirati na tjednoj ili dnevnoj bazi. Neki čak neprestano testiraju svoju sigurnost. Važno je pronaći ono što najbolje funkcionira za tu organizaciju. Ako vlasnici tvrtke nisu sigurni u razinu rizika s kojom se njihova tvrtka suočava, najbolje je konzultirati se sa stručnjakom za sigurnost.

Usklađenost s propisima jedna je od najvažnijih stvari koje se moraju uzeti u obzir pri pokretanju novog posla. Regulatorni aspekt jedna je od glavnih briga za uspjeh svake tvrtke. Svaka industrija ima svoj skup pravila i propisa. Penetracijsko testiranje je tehnika procjene sigurnosti koja je dizajnirana za prepoznavanje ranjivosti u odabranim aplikacijama. Tvrtke i organizacije često upotrebljavaju *Sarbanes-Oxley* (SOX), Zakon o prenosivosti i odgovornosti zdravstvenog osiguranja (HIPAA) i Savezni zakon o upravljanju sigurnošću informacija (FISMA) za usklađivanje s vladinim propisima.

Penetracijski testovi mogu se provoditi na različitim sustavima i uređajima, uključujući računala, prijenosna računala, web poslužitelje, vatrozidove i usmjerivače. Izvode ih neovisni izvođači i mogu ih koristiti organizacije za dokazivanje usklađenosti s industrijskim propisima. Penetracijski testovi sadržavat će izvješće o nalazima i često će preporučiti kako popraviti ili ublažiti identificirane ranjivosti.

Penetracijski testovi bitan su dio svake sigurnosne strategije. Oni uključuju tim stručnjaka koji simuliraju kibernetički napad u stvarnom svijetu na sustave i aplikacije tvrtke kako bi vidjeli ranjivosti njezine mreže. Penetracijski test se može podijeliti u 5 kategorija:

- Web aplikacija i API penetracijsko testiranje
- Test penetracije mobilne aplikacije
- Testiranje prodora u oblaku

- *Blockchain* i penetracijsko testiranje pametnih ugovora
- Penetracijsko testiranje mreže, [23].

4.1. Analiza različitih faza u procesu penetracijskog testiranja

Neke organizacije navode pet faza penetracije dok druge navode šest ili sedam. Osim toga, organizacije mogu imati različita imena za svaku fazu, unatoč tome što su procesi faze identični. Do razlike u broju faza testiranja dolazi zbog dvije faze koje se odvijaju prije testiranja i nakon što se završi, neke organizacije ga izostavljaju. Iako nisu tehnički dijelovi testa, pokazali su se ključnima za sigurnost.

Predobvezivanje je često zanemarena faza, no ključno je uskladiti penetracijske ispitivače i organizaciju. Postavljanjem opsega, logistike, pravila angažmana i vremenskog rasporeda testa s jasnim ciljevima prije početka, osigurava se uspješna komunikacija. Bez definiranja što će biti testirano i koja će vrsta testiranja biti provedena, rezultati mogu biti nepotpuni ili irelevantni. Ova faza je temelj planiranog ispitivanja, a ni organizacije ni ispitivači ne bi trebali krenuti bez nje. Osim toga, za detaljno testiranje, ispitivači trebaju provoditi radnje koje bi bile nezakonite bez odobrenja ili autorizacije. Stoga bi organizacije trebale postaviti jasna pravila u ugovorima s ispitivačima, koji bi također trebali sadržavati ključne informacije o testu i mjere opreza.

Izviđanje ili prikupljanje obavještajnih podataka otvorenog izvora (OSINT) je ključno za testere kako bi dobili relevantne informacije o sustavu. Prva faza, planiranje penetracijskog testiranja, omogućuje precizno prikupljanje podataka za strategiju napada. Izviđanje može biti aktivno, direktno s ciljnim sustavom, ili pasivno, koristeći javno dostupne informacije poput društvenih medija, web stranica i poreznih podataka. Alati kao što su Censys i Shodan koriste se za prikupljanje informacija o mreži putem indeksiranja odgovornih zaglavlja javnih IP adresa. OSINT Framework i kontrolni popisi koriste se za pronalaženje ulaznih točaka i ranjivosti unutar organizacije..

U fazi skeniranje ili otkrivanje ispitivači traže ulazne točke. U idealnom slučaju, nastoje identificirati što više otvorenih portova. U ovoj se fazi koristi nekoliko alata za prepoznavanje otvorenih portova i provjeru mrežnog prometa. Faza otkrivanja sastoji se od skeniranja i analize imovine pomoću alata kao što je Nmap, koji je mrežni skener koji se koristi za otkrivanje host-ova i servisa na računalnoj mreži slanjem paketa i analizom odgovora. U ovoj fazi ispitivač može dobiti informacije o dostupnim sredstvima i informacijama, kao što su operativni sustavi, otvoreni portovi i pokrenute usluge. Ako ispitivač pokrene test bijele kutije, organizacija im je možda već dostavila popis IP-ova za ciljanu imovinu i druge informacije o mreži. Međutim, ako izvode testove sive ili crne kutije, tada simuliraju stvarni napad i rade bez tih informacija. Stoga je ova faza ključna pri izvođenju testova sive i crne kutije. Osam najboljih alata za skeniranje ranjivosti su: Invicti, Nmap, OpenVAS, RapidFire VulScan, StackHawk, Tenable.io i Wiz.

Faza procjena ranjivosti (dobivanje pristupa) koristi podatke prikupljene tijekom prethodnih faza, ispitivač će započeti izgradnju modela prijetnji i procijeniti ranjivosti. Mete se identificiraju, a ispitivač mapira vektore napada. Ispitivač će mapirati i identificirati područja i imovinu visoke vrijednosti, kao što su: podaci o zaposlenicima, podaci o kupcima, podaci o partnerima i opskrbnom lancu, tehnički podaci, unutarnje i vanjske prijetnje od strane menadžmenta, dobavljači, luke, mreže, aplikacije i protokoli.

Ispitivači mogu koristiti resurse poput Nacionalne baze podataka o ranjivostima (NVD), repozitorija podataka o upravljanju ranjivostima koji analiziraju ranjivosti softvera objavljene u bazi podataka o uobičajenim ranjivostima i izloženostima (CVE). Iako se može izvršiti ručno skeniranje ranjivosti, ispitivači obično koriste alate kao što su Tenable, Rapid7, Qualys i Nmap. Neke sigurnosne organizacije ovu fazu nazivaju "dobivanjem pristupa". Testeri koriste napade web aplikacija, kao što su XSS, SQL injekcije i stražnja vrata, kako bi pronašli ranjivosti i iskoristili ih eskalacijom privilegija, krađom podataka, presretanjem prometa i drugim tehnikama.

U fazi iskorištavanje (održavanje pristupa) testeri dokazuju mogu li se identificirane ranjivosti iskoristiti. Poznato i kao održavanje pristupa, iskorištavanje je jedna od najkritičnijih faza jer ispitivač pokušava probiti i pristupiti ciljnom sustavu. U ovoj fazi penetracijskog testiranja, ispitivač pokušava pristupiti ciljnom sustavu i iskoristiti identificirane ranjivosti, obično koristeći alat kao što je Metasploit, koji simulira napade iz stvarnog svijeta. Ispitivači su odgovorni za imovinu organizacije i u ovoj fazi moraju osigurati da sustav nije ugrožen ili oštećen zbog njihovih simulacija.

Pravi kibernetički napadi mogu trajati od nekoliko minuta do nekoliko sati, tako da ranjivosti identificirane u prethodnim fazama moraju biti postojane kako bi ih loši akteri mogli iskoristiti. Općenito, testeri će tražiti *root* ili administratorske povlastice uređaja ili sustava. Metasploit se koristi zbog svojih pojednostavljenih mogućnosti procesa za pronalaženje i izvršavanje javno dostupnih eksploatacija za ranjivosti. Osim što osigurava da su ranjivosti stabilne, ova faza također mjeri posljedice proboja. Na primjer, može li ispitivač šifrirati ili eksfiltrirati podatke ili simulirati napade nultog dana ili hakiranja ucjenjivačkim softverom i u kojoj mjeri.

Predzadnja faza je posteksploatacija, izvješćivanje i analiza rizika. Iako većina organizacija navodi ovaj korak kao fazu izvješćivanja, ostale komponente nakon eksploatacije, poput aktivnosti čišćenja, moraju biti uključene u ovu fazu penetracijskog testa. Nakon završetka testiranja i prezentiranja izvješća i preporuka, ispitivač mora očistiti okoliš. To podrazumijeva ostavljanje sustava točno onakvim kakvog su ga pronašli, rekonfiguriranje pristupa korištenog za probijanje IT okruženja i vraćanje ostalih modifikacija koje su možda napravili. Aktivnosti čišćenja također otvaraju put sanaciji i završnoj fazi testiranja prodora. Tipične aktivnosti čišćenja su:

- Uklanjanje svih izvršnih datoteka, skripti i privremenih datoteka iz ugroženih sustava

- Ponovna konfiguracija postavki natrag na izvorne parametre prije penetracijskog testiranja
- Uklanjanje svih *rootkitova* instaliranih u okruženju
- Uklanjanje svih korisničkih računa kreiranih za povezivanje s ugroženim sustavom

Izvješće se smatra najključnijim dokumentom. To je završna prezentacija organizaciji koja je angažirala ispitivača. S izvješćem organizacije mogu poduzeti mjere, popraviti ranjivosti i ojačati svoje sustave i osoblje ako je potrebno. Izvješća moraju biti jasna i transparentna. Ispitivači moraju dokumentirati sve faze, ciljana sredstva, vrstu testa i tehnike te otkrivene ranjivosti. Osim toga, mogu se uključiti vodiči za popravak ili zakrpu ranjivosti. Izvješća o penetracijskog testiranju uključuju:

- Specifične ranjivosti koje su iskorištene
- Osjetljive podatke kojima je pristupljeno
- Količinu vremena tijekom kojeg je ispitivač mogao ostati u sustavu neotkriven

Uobičajena je praksa da organizacija od ispitivača zatraži dezinficirana izvješća prije nego unajmi njihove usluge. To im omogućuje pregled standarda i detalja koje koristi dobavljač. Dobra izvješća penetracijskog testiranja su uredno organizirana i prioritetne su po razini rizika.

Sanacija je završna faza penetracijskog testa i spada u odgovornost organizacije. Korištenjem izvješća i nalaza te informacija koje imaju iz interakcije s ispitivačem, posebno ako je napravljen penetracijski test bijele kutije, organizacije mogu početi unositi promjene u svoje sustave kako bi popravile otkrivene ranjivosti. Sanacija može biti vrlo izazovna za organizacije koje nemaju resurse. Stoga su izvješća koja uključuju vodiče za sanaciju najcjenjenija. Nakon popravka, faze će se često ponovno pokrenuti kako bi se testirale nadogradnje ili drugi sustavi ili pokrenule različite vrste penetracijskog testiranja, [24].

4.2. Metodologija i okviri za izvođenje penetracijskog testiranja

Penetracijsko testiranje je etička procjena kibernetičke sigurnosti koja pomaže organizacijama da poboljšaju svoje stanje kibernetičke sigurnosti. To je složen proces koji, ako se ne provodi pravilno, može propustiti značajne ranjivosti i ostaviti organizaciju ranjivom. Ispunjavanje penetracijskog testa u skladu sa strukturiranim okvirima i procedurama jamči postizanje specifičnih ciljeva i pokrivenost svih relevantnih područja. Međutim, budući da su svaka organizacija i okruženje jedinstveni, jedinstvena strategija za penetracijsko testiranje je neučinkovito.

Od ključne je važnosti razmotriti pruža li pristup penetracijskog testiranja odgovarajuću razinu procjene za tvrtku. To se postiže s upoznavanjem s raznim vrstama metodologija. Neke od njih su:

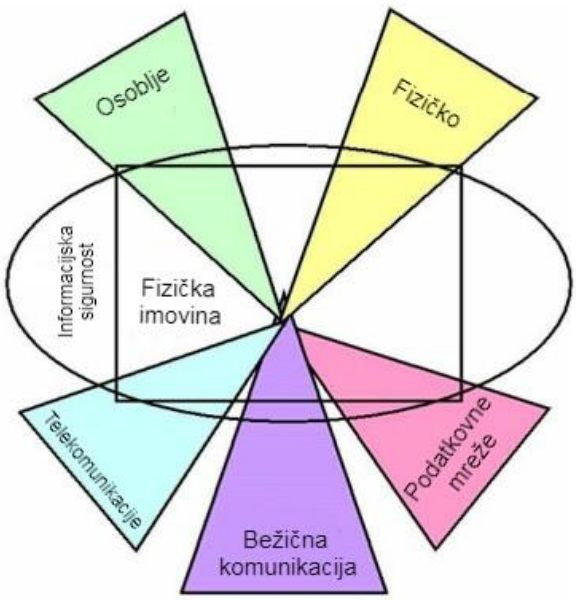
- Priručnik za metodologiju testiranja sigurnosti otvorenog koda (OSSTMM)
- Otvoren projekt sigurnosti web aplikacije (OWASP)
- Nacionalni institut za standarde i tehnologiju (NIST)
- Izvršni standard metodologije penetracijskog testiranja (PTES)
- Okvir za procjenu sigurnosti informacijskog sustava (ISSAF), [25].

4.2.1. OSSTMM

Priručnik za metodologiju testiranja sigurnosti otvorenog koda (OSSTMM) je standard za testere za sigurnost. Opisuje kompletnu metodologiju testiranja, nudeći prilično dobre alate za izvješćivanje o skupu rezultata. Glavni koncept je djelokrug, što je ukupno moguće operativno sigurnosno okruženje u kojem se odvija bilo kakva interakcija s bilo kojim sredstvom, što može uključivati i fizičke komponente sigurnosnih mjera. Djelokrug se sastoji od tri kanala:

- Sigurnosnog komunikacijskog kanala (COMSEC)
- Fizičkog sigurnosnog kanala (PHYSSEC).
- Sigurnosnog kanala spektra (SPECSEC).

Kanali su sredstva interakcije s imovinom. Imovina je ono što je vlasniku vrijedno. Djelokrug zahtijeva da se sve prijetnje moraju smatrati mogućima, čak i ako nisu vjerojatne. Na slici 13 prikazani su pet kanala OSSTMM metodologije.



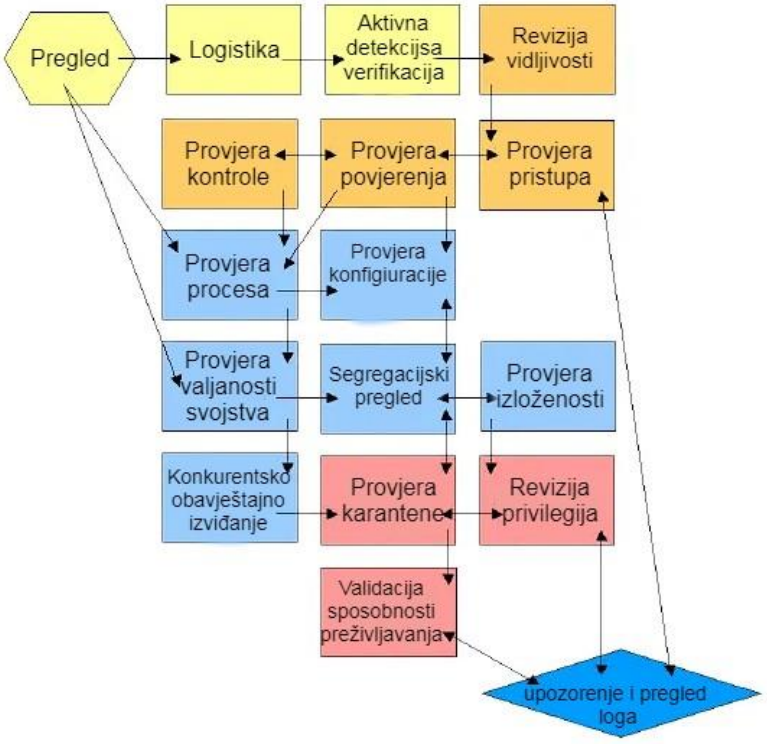
Slika 13: Prikaz pet kanala OSSTMM metodologije

Izvor:[26]

Tri glavna kanala podijeljena su u pet podkanala prije nego što ih počnu koristiti sami testeri.

- Osoblje - sadrži sve ljudske elemente komunikacije
- Fizičko - sastoji se od opipljivih elemenata sigurnosti gdje interakcija zahtijeva fizički napor ili izvor energije za manipuliranje imovinom.
- Bežična komunikacija - sadrži sve elektroničke komunikacije, signale i emanacije koje se odvijaju preko elektromagnetskog spektra.
- Podatkovne mreže - sadrži sve elektroničke sustave i podatkovne mreže gdje se interakcije odvijaju preko uspostavljenih kabela i žičnih mrežnih linija.
- Telekomunikacije - obuhvaća sve telekomunikacijske mreže, digitalne ili analogne, gdje se interakcija odvija preko uspostavljenih telefonskih ili telefonskih mrežnih linija.

OSSTMM opisuje sedamnaest modula za analizu svakog od pod-kanala (slika 14). Posljedično, ispitivač mora izvesti 17*5=85 analiza prije pisanja konačnog izvješća. Opisujući ogroman skup radnji OSSTMM je postao jedna od najpotpunijih metodologija ikada. Bila je to prva metodologija koja je uključila ljudski faktor u testove, uzimajući u obzir utvrđenu činjenicu da ljudi mogu biti vrlo opasni za sustav. Na slici 14 se opisuje područje djelovanja metodologije, vrlo je jasna i intuitivna te nudi lijep pregled onoga što OSSTMM radi ako se dobro implementira.



Slika 14: Prikaz 17 modula OSSTMM metodologije

Izvor:[26]

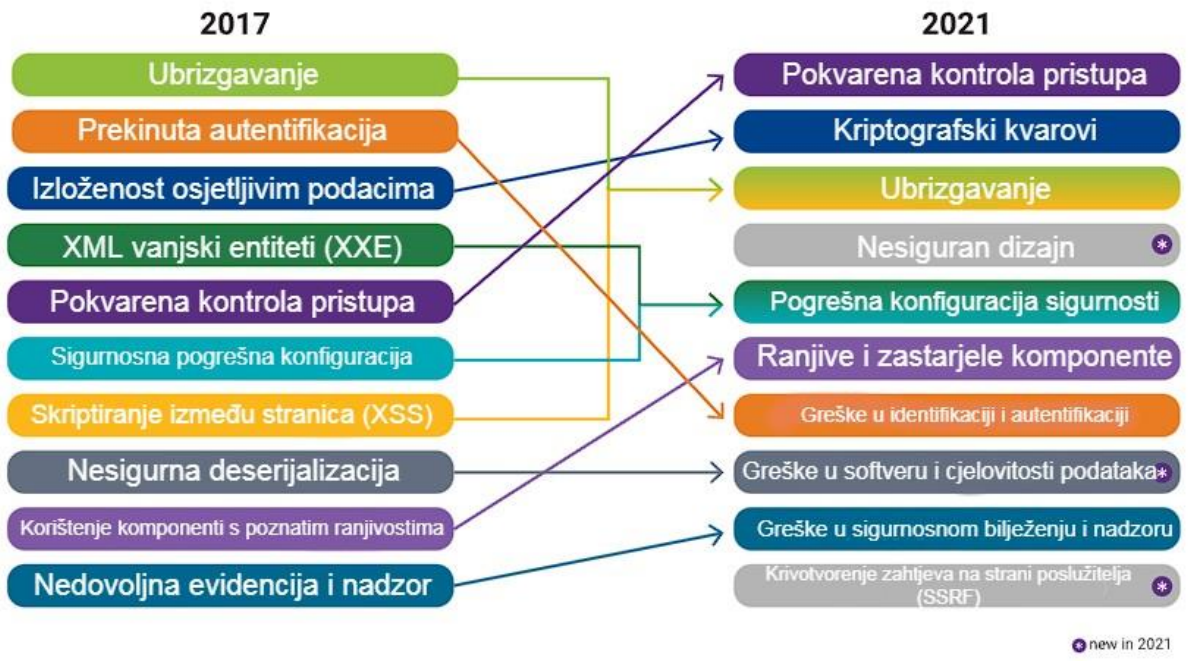
OSSTMM je vrlo pristran prema komunikacijskoj analizi, tjerajući testera da uloži mnogo truda u protok podataka, dok je kontrola protoka i analiza aplikacije, relativno, zanemarena. Inducirane hipoteze se ne razmatraju, izrezujući značajan skup mogućih putova otkrivanja ranjivosti, korisnih za testera. Drugi problem dolazi od dijagrama od sedamnaest modula (slika 15), koji nije intuitivan. Ispitivač treba jasan i ravan tijek, budući da je njegov posao pronalaženje ranjivosti, a ne tumačenje metodologija. OSSTMM ne pruža intuitivan proces, uglavnom zbog velikog broja stavki i mnogo neurednih petlji unutar tijeka. Način na koji OSSTMM predlaže pisanje izvješća također je problematičan. Iskustvo uči da je proces testiranja penetracije dugotrajan, stoga bi dobra praksa bila pisanje izvješća nakon svake akcije; inače mnogi detalji lako padaju u zaborav. Konačno, OSSTMM nudi predloške za popunjavanje izvješća, ali nažalost ti predlošci slijede striktno linearni proces čitanja. Iako ovaj holistički pristup može biti isplativ za najbolje razumijevanje, postoje čitatelji (na primjer: tehničke komisije ili softverski inženjeri) koji trebaju znati gdje su pojedina pitanja na prvi pogled, [26].

4.2.2. OWASP

Otvoren projekt sigurnosti web aplikacije (OWASP) neprofitna je zaklada posvećena poboljšanju sigurnosti softvera. Djeluje prema modelu "otvorene zajednice", što znači da svatko može sudjelovati i doprinijeti mrežnim razgovorima, projektima i još mnogo toga vezanim uz OWASP. Za sve, od mrežnih alata i videa do foruma i događaja, OWASP osigurava da njegove ponude ostanu besplatne i lako dostupne putem svoje web stranice.

OWASP Top 10 pruža rangiranje - i smjernice za sanaciju - 10 najkritičnijih sigurnosnih rizika web aplikacija. Korištenjem opsežnog znanja i iskustva suradnika OWASP- ove otvorene zajednice, izvješće se temelji na konsenzusu među sigurnosnim stručnjacima iz cijelog svijeta. Rizici su rangirani prema učestalosti otkrivenih sigurnosnih nedostataka, ozbiljnosti nepokrivenih ranjivosti i veličini njihovih potencijalnih utjecaja. Svrha izvješća je ponuditi razvojnim programerima i stručnjacima za sigurnost web aplikacija uvid u najraširenije sigurnosne rizike kako bi mogli uklopiti nalaze i preporuke izvješća u vlastite sigurnosne prakse, čime se smanjuje prisutnost poznatih rizika u njihovim aplikacijama.

OWASP održava svoju listu Top 10 od 2003., ažurirajući je svake dvije ili tri godine u skladu s napretkom i promjenama na AppSec (sigurnost aplikacije) tržištu. Važnost popisa leži u korisnim informacijama koje pruža služeći kao kontrolni popis i interni standard za razvoj web aplikacija za mnoge najveće svjetske organizacije. Na slici 16 je vidljiva usporedba popis za 2017. i 2021. godinu. OWASP je dodao tri nove kategorije, izvršio četiri promjene u imenovanju i djelokrugu te izvršio određenu konsolidaciju.



Slika 15: Prikaz usporedbe OWASP top 10 popisa između 2017. i 2021. godine

Izvor:[26]

4.2.2.1. Pokvarena kontrola pristupa

Prethodno broj 5 na popisu, slomljena kontrola pristupa - slabost koja napadaču omogućuje pristup korisničkim računima . premještena je na broj 1 za 2021. Napadač u ovom kontekstu može funkcionirati kao korisnik ili kao administrator u sustavu.

Primjer: aplikacija dopušta promjenu primarnog ključa, a kada se taj ključ promijeni u zapis drugog korisnika, račun tog korisnika može se pregledavati ili mijenjati.

Rješenje: rješenje za interaktivno testiranje sigurnosti aplikacija (IAST), kao što je Seeker, Invicti ili Contrast security može pomoći da se bez napora otkrije krivotvorenje zahtjeva između stranica ili nesigurna pohrana osjetljivih podataka..

4.2.2.2. Kriptografski kvarovi

Prethodno na poziciji broj 3 i ranije poznat kao izloženost osjetljivim podacima, ovaj unos je preimenovan u kriptografske greške kako bi se točno prikazao kao osnovni uzrok, a ne kao simptom. Kriptografski kvarovi nastaju kad su važni pohranjeni ili preneseni podaci (kao što je broj socijalnog osiguranja) ugroženi.

Primjer: financijska institucija ne uspijeva adekvatno zaštititi svoje osjetljive podatke i postaje laka meta za prijevare s kreditnim karticama i krađu identiteta.

Rješenje: potrebno je skenirati neadekvatnu snagu šifriranja i slabe ili tvrdo kodirane kriptografske ključeve, a zatim identificirati sve pokvarene ili rizične kriptografske algoritme.

4.2.2.3. Ubrizgavanje

Ubrizgavanje se pomiče s broja 1 na broj 3, a skriptiranje na različitim mjestima sada se smatra dijelom ove kategorije. U suštini, do ubrizgavanja koda dolazi kada napadač pošalje nevažeće podatke u web aplikaciju kako bi aplikaciju natjerao da učini nešto za što nije dizajnirana.

Primjer: aplikacija koristi nepouzdane podatke kada konstruira ranjivi SQL poziv.

Rješenje: uključivanje alata SAST i IAST u cjevovod kontinuirane integracije/kontinuirane isporuke (CI/CD) pomaže u prepoznavanju nedostataka ubrizgavanja i na statičkoj razini koda i dinamički tijekom testiranja vremena izvođenja aplikacije. Moderni alati za testiranje sigurnosti aplikacije (AST) kao što su Seeker, Invicti ili HCL AppScan mogu pomoći u zaštiti softverske aplikacije tijekom različitih faza testiranja i provjeriti postoje li različiti napadi ubrizgavanjem (uz SQL ubrizgavanje).

4.2.2.4. Nesiguran dizajn

Nesiguran dizajn nova je kategorija za 2021. koja se fokusira na rizike povezane s nedostacima u dizajnu. Dok se organizacije nastavljaju "pomicati ulijevo", modeliranje prijetnji, obrasci i principi sigurnog dizajna i referentne arhitekture nisu dovoljni.

Primjer: lanac kina koji omogućuje popuste na grupne rezervacije zahtijeva polog za grupe veće od 15 osoba. Napadači prijete ovom modelu protoka kako bi vidjeli mogu li rezervirati stotine sjedala u raznim kinima u lancu, uzrokujući tako tisuće dolara izgubljenog prihoda.

Rješenje: alati popust Seeker-a, Invicti ili Contrast security otkrivaju ranjivosti i otkrivaju sve ulazne i izlazne API-je, usluge i pozive funkcija u vrlo složenim aplikacijama koje se temelje na webu, oblaku i mikroservisima.

4.2.2.5. Pogrešne sigurnosne konfiguracije

Nekadašnja kategorija vanjskih subjekata sada je dio ove kategorije rizika, koja se pomaknula sa šestog mjesta na peto. Pogrešne sigurnosne konfiguracije su slabosti dizajna ili konfiguracije koje proizlaze iz konfiguracijske pogreške ili nedostatka.

Primjer: zadani račun i njegova izvorna zaporka još uvijek su omogućeni, što sustav čini ranjivim na iskorištavanje.

Rješenje: rješenja kao što je Coverity SAST uključuju alat za provjeru koji identificira izloženost informacija dostupnih putem poruke o pogrešci. Dinamički alati mogu otkriti otkrivanje informacija i neprikladne konfiguracije HTTP zaglavlja tijekom testiranja vremena izvođenja aplikacije.

4.2.2.6. Ranjive i zastarjele komponente

Ova kategorija prelazi s pozicije devet na poziciju šest i odnosi se na komponente koje predstavljaju i poznate i potencijalne sigurnosne rizike. Komponente s poznatim ranjivostima, kao što su uobičajene ranjivosti i izloženosti (CVE), treba identificirati i zakrpati, dok zastarjele ili zlonamjerne komponente treba procijeniti u pogledu održivosti i rizika koje mogu predstavljati.

Primjer: zbog količine komponenti koje se koriste u razvoju, razvojni tim možda neće znati ili razumjeti sve komponente koje se koriste u njihovoj aplikaciji, a neke od tih komponenti mogu biti zastarjele i stoga ranjive na napad.

Rješenje: alati za analizu sastava softvera (SCA) mogu se koristiti uz statičku analizu i IAST za prepoznavanje i otkrivanje zastarjelih i nesigurnih komponenti u aplikaciji. IAST i SCA dobro surađuju, dajući uvid u to koliko se ranjive ili zastarjele komponente zapravo koriste. IAST i SCA zajedno idu dalje od identificiranja ranjive komponente, otkrivajući detalje poput toga je li tu komponentu trenutno učitala aplikacija koja se testira.

4.2.2.7. Greške pri identifikaciji i autentifikaciji

Prethodno poznat kao prekinuta provjera autentičnosti, ova kategorija je pomaknuta s broja dva i sada uključuje uobičajena nabrajanja slabosti (CWE) povezane s neuspješnim identifikacijama. Točnije, funkcije povezane s autentifikacijom i upravljanjem sesijama, kada se neispravno implementiraju, omogućuju napadačima kompromitiranje lozinki, ključnih riječi i sesija, što može dovesti do krađe korisničkog identiteta i više.

Primjer: web aplikacija dopušta upotrebu slabih lozinki ili lozinki koje je lako pogoditi (tj. "password1").

Rješenje: višefaktorska provjera autentičnosti može pomoći u smanjenju rizika od kompromitiranih računa, a automatizirana statička analiza vrlo je korisna u pronalaženju takvih nedostataka, dok ručna statička analiza može dodati snagu pri procjeni prilagođenih shema provjere autentičnosti.

4.2.2.8. Pogreške u integritetu softvera i podataka

Ovo je nova kategorija za 2021. koja se fokusira na ažuriranje softvera, kritične podatke i CI/CD kanale koji se koriste bez provjere integriteta. Također sada uključena u ovaj unos, nesigurna deserijalizacija je greška deserijalizacije koja napadaču omogućuje daljinsko izvršavanje koda u sustavu.

Primjer: aplikacija deserijalizira neprijateljske objekte koje je dostavio napadač, otvarajući se ranjivosti.

Rješenje: alati za sigurnost aplikacije pomažu u otkrivanju nedostataka deserijalizacije, a penetracijsko testiranje može potvrditi problem.

4.2.2.9. Sigurnost prijavljivanja i kvarovi nadgledanja

Prethodno poznat kao nedovoljno prijavljivanje i praćenje, ovaj unos je pomaknut s broja 10 i proširen je kako bi uključio više vrsta kvarova. Prijavljivanje i praćenje su aktivnosti koje bi se na web-mjestu trebale provoditi često – ako se to ne radi, web-mjesto postaje ranjivo na ozbiljnije kompromitirajuće aktivnosti.

Primjer: događaji koji se mogu nadzirati, poput prijava, neuspjelih prijava i drugih važnih aktivnosti, ne bilježe se, što dovodi do ranjive aplikacije.

Rješenje: Nakon provođenja penetracijskog testiranja, programeri mogu proučiti testne zapisnike kako bi identificirali moguće nedostatke i ranjivosti. SAST i IAST alati mogu pomoći u identificiranju nezabilježenih sigurnosnih iznimaka.

4.2.2.10. Krivotvorenje zahtjeva na strani poslužitelja

Nova kategorija 2021. godine, krivotvorenje zahtjeva na strani poslužitelja (SSRF) može se dogoditi kada web aplikacija dohvati udaljeni resurs bez provjere URL-a koji je donio korisnik. To napadaču omogućuje da natjera aplikaciju da pošalje izrađeni zahtjev na neočekivano odredište, čak i kada je sustav zaštićen vatrozidom, VPN-om ili dodatnim popisom kontrole pristupa mreži. Ozbiljnost i učestalost SSRF napada raste zbog usluga u oblaku i povećane složenosti arhitektura.

Primjer: Ako je mrežna arhitektura ne segmentirana, napadači mogu koristiti rezultate veze ili proteklo vrijeme za povezivanje ili odbijanje SSRF korisnih veza kako bi mapirali interne mreže i odredili jesu li portovi otvoreni ili zatvoreni na internim poslužiteljima.

Rješenje: alati poput Seeker-a ili SSRFmap su jedni od modernih alata koji mogu pratiti, nadzirati i otkriti SSRF bez potrebe za dodatnim skeniranjem i trijažom, [27].

4.2.3. Nacionalni institut za standarde i tehnologiju (NIST)

Od pametne elektroenergetske mreže i elektroničkih zdravstvenih zapisa do atomskih satova, naprednih nano-materijala i računalnih čipova, bezbrojni proizvodi i usluge oslanjaju se na neki način na tehnologiju, mjerenja i standarde koje osigurava Nacionalni institut za standarde i tehnologiju. Osnovan 1901., NIST je neregulatorna savezna agencija unutar Ministarstva trgovine SAD-a. Misija NIST-a je promicanje inovacija i industrijske konkurentnosti SAD-a unaprjeđenjem znanosti o mjerenju, standardu i tehnologiji na načine koji povećavaju ekonomsku sigurnost i poboljšavaju kvalitetu života.

NIST svoju misiju ostvaruje kroz sljedeće programe:

- NIST laboratoriji - koji provode istraživanja svjetske razine, često u bliskoj suradnji s industrijom, koja unapređuje nacionalnu tehnološku infrastrukturu i pomaže američkim tvrtkama u stalnom poboljšanju proizvoda i usluga,
- Hollings partnerstvo za proširenje proizvodnje - nacionalna mreža lokalnih centara koji nude tehničku i poslovnu pomoć manjim proizvođačima kako bi im pomogli stvoriti i zadržati radna mjesta, povećati profit i uštedjeti vrijeme i novac,
- Baldrige program izvrsnosti performansi - promiče izvrsnost performansi među američkim proizvođačima, uslužnim tvrtkama, obrazovnim ustanovama, pružateljima zdravstvenih usluga i neprofitnim organizacijama; provodi programe informiranja, te upravlja godišnjom nacionalnom nagradom za kvalitetu Malcolma Baldrigea, koja prepoznaje izvrsnost u radu i postignuća u kvaliteti, [28].

4.2.4. Izvršni standard metodologije penetracijskog testiranja (PTES)

Izvršni standard metodologije penetracijskog testiranja (PTES) je metoda penetracijskog testiranja. Razvio ju je tim stručnjaka za informacijsku sigurnost s ciljem rješavanja potrebe za potpunim i ažuriranim standardom u penetracijskom testiranju. Osim usmjeravanja stručnjaka za sigurnost, također pokušava informirati tvrtke o tome što bi trebali očekivati od penetracijskog testa te ih voditi u određivanje djelokruga i pregovaranja o uspješnim projektima. PTES je opsežan vodič koji ocrtava standardiziranu metodologiju za provođenje penetracijskih testova. Uključuje najbolje prakse za svaku fazu procesa penetracijskog testiranja, od određivanja opsega i planiranja do generiranja izvješća. PTES opisuje penetracijski test u sedam glavnih odjeljaka: Predobvezivanje, prikupljanje obavještajnih podataka, modeliranje prijetnji, analiza ranjivosti, iskorištavanje, posteksploatacija i izvještavanje

PTES je sveobuhvatan popis stavki koje bi se trebale riješiti tijekom penetracijskog testa. Uključuje smjernice visoke razine o vrstama testova koje treba provesti, kao i specifične pojedinosti o svakom testu. PTES pruža dosljedan okvir koji testeri trebaju slijediti, što pomaže osigurati da su pokriveni svi aspekti penetracijskog testa. PTES je dizajniran da pomogne ispitivačima da odrede najučinkovitiji način provođenja penetracijskog testa, na temelju potreba njihove organizacije. Može se koristiti kao samostalni kontrolni popis ili kao dio veće metodologije testiranja. U svakom slučaju, pruža vrijednu polaznu točku za svakog ispitivača koji želi osigurati da pokriva sve svoje baze. PTES je besplatno dostupan na mreži i svatko mu može pristupiti. To ga čini izvrsnim resursom za svakoga tko želi započeti s penetracijskim testiranjem, [29].

4.2.5. Okvir za procjenu sigurnosti informacijskog sustava (ISSAF)

Okvir za procjenu sigurnosti informacijskog sustava (ISSAF) recenzirani je strukturirani okvir koji kategorizira procjenu sigurnosti informacijskog sustava u različite domene i detaljno opisuje posebne kriterije procjene ili testiranja za svaku od tih domena. Cilj mu je pružiti terenske podatke o procjeni sigurnosti koji odražavaju

scenarije iz stvarnog života. ISSAF bi se prvenstveno trebao koristiti za ispunjavanje zahtjeva organizacije za sigurnosnu procjenu, a dodatno se može koristiti kao referenca za ispunjavanje drugih potreba za informacijskom sigurnošću. ISSAF uključuje ključni aspekt sigurnosnih procesa i njihovu procjenu i ojačavanje kako bi se dobila potpuna slika ranjivosti koje bi mogle postojati.

Informacije u ISSAF-u organizirane su u dobro definiranom kriteriju ocjenjivanja, od kojih su svaki pregledali stručnjaci u tom području. Ti kriteriji ocjenjivanja uključuju: opis kriterija ocjenjivanja, njegove ciljeve, preduvjete za provođenje evaluacije, proces evaluacije, prikazuje očekivane rezultate, preporučene protumjere i reference na vanjske dokumente

Ciljevi ISSAF-a su:

- Služiti kao od kraja do kraja referentni dokument za procjenu sigurnosti
- Standardizirati proces procjene sigurnosti informacijskog sustava
- Postaviti minimalnu razinu prihvatljivog procesa
- Osigurati temelj na kojem se može (ili treba) izvršiti procjena
- Za procjenu zaštitnih mjera protiv neovlaštenog pristupa
- Djelovati kao referenca za implementaciju informacijske sigurnosti
- Ojačati postojeće sigurnosne procese i tehnologiju, [30].

4.3. Implementacija penetracijskog testiranja u organizacijama različitih veličina

Mnoge organizacije se razvijaju brže kao rezultat njihovog strateškog pristupa kojeg imaju. Također, tvrtke imaju tendenciju usvajanja promjena brže nego ikad i dodjeljuju točno određenim osobama da rade na određenim pitanjima samo kako bi bile sigurne da povećavaju vrijednost svojih usluga i proizvoda. S tim u vezi, kako se automatizirane usluge povećavaju, tako raste i broj aplikacija koje se koriste u njihovom radu. Stoga je pristup koji testira sve aplikacije koje su u procesu implementacije ili su operativne obavezan kako ne bi došlo do neuspjeha zbog bilo kakvog kršenja sigurnosti, [31].

4.3.1. Implementacija penetracijskog testiranja unutar korporacija u financijskom sektoru

Industrija financijskih usluga neprestano je na udaru brojnih i značajnih kibernetičkih napada i prijetnji. Postoji mnogo stvari koje tvrtka može učiniti, a penetracijski testovi jedni su od rješenja koji mogu pomoći u ublažavanju tih rizika.

Sigurnosni izazovi s kojima se financijska industrija suočava razlikuju se ovisno o nekoliko stvari: veličini organizacije, koliko su etablirani, proizvodima/uslugama koji su u ponudi i podacima koji se čuvaju o kupcima.

Mnoge financijske organizacije imaju aplikacije za klijente, što privlači većinu zlonamjernih napadača. Zbog količine podataka (osobnih i financijskih) koje aplikacije

drže i obrađuju o pojedincu ili organizaciji. Jedan primjer vrste ranjivosti web aplikacije koju bi financijske organizacije mogle uočiti je nesigurna izravna referenca objekta. Ranjivost obično uključuje prijavu hakera u aplikaciju i unošenje malih promjena u URL te dobivanje pristupa profilima drugih korisnika.

Za veće organizacije ključne ranjivosti često okružuju korištenje zastarjele tehnologije i migracije na oblak. Starije tehnologije nisu razvijene imajući na umu današnje prijetnje kibernetičkoj sigurnosti. Kao takva, zastarjela tehnologija može predstavljati povećani rizik – osobito ako se softverske zakrpe i ažuriranja ne objavljuju i ne primjenjuju redovito.

Rad na daljinu također predstavlja brojne sigurnosne rizike u cijelom financijskom sektoru, pri čemu zaposlenici pristupaju mrežama i sustavima izvan ureda, što znači da je ključno provoditi redovite procjene kako bi se identificirale slabosti. Na primjer, ako se podaci šalju u ne kriptiranom formatu, poput teksta plana, hakeri bi ih mogli presresti i ukrasti. Financijskim organizacijama stoga ne bi trebalo dopustiti pristup nepoznatim Wi-Fi mrežama osim ako ne koriste VPN vezu.

Odabir pravog sigurnosnog partnera ključan je za uspješan penetracijski test – važno je odabrati onoga koji može pokazati iskustvo specifično za sektor i razumijevanje najnovijih tehnika koje koriste hakeri, [32].

4.3.2. Implementacija penetracijskog testiranja u zdravstvenom sektoru

Zdravstvene organizacije zabrinute su o vlastitoj sigurnosti. Prema izvješću HIMSS-a iz 2018., samo oko 50 posto zdravstvenih organizacija izdvaja dio svog IT proračuna posebno za potrebe kibernetičke sigurnosti. Gledajući širu sliku, to znači da samo oko polovica tvrtki u zdravstvenom prostoru u potpunosti izdvaja resurse za zaštitu podataka pacijenata.

Razumijevanje najvećih prijetnji zdravstvenim organizacijama najbolje je mjesto za početak kada se radi o identificiranju najboljeg smjera djelovanja. Za zdravstvene organizacije sigurnosni incidenti najvišeg prioriteta vrte se oko pristupa klijentskim podacima, koji mogu biti ugroženi u slučaju kršenja ili incidenta s ucjenjivačkim softverom. To su dva možda najčešća incidenta s kojima su se susrele medicinske ordinacije i osiguravatelji u posljednjih nekoliko godina. Tehnike koje variraju od tehničkog do ljudskog elementa najčešće su odgovorne za izazivanje ozbiljnih sigurnosnih nesreća. Bilo da se radi o zaposleniku koji slučajno klikne vezu za krađu identiteta ili zlonamjerni akter koji zatvara organizaciju pomoću ucjenjivačkog softvera, postoje brojni znakovi upozorenja koje tvrtke moraju moći brzo prepoznati i sanirati.

Kako bi poboljšali sigurnost e-pošte, organizacija bi trebala primijeniti slojeviti pristup. To uključuje zaštitu kao što je skeniranje privitaka za e-poštu radi provjere zlonamjernog sadržaja, obuku za podizanje svijesti korisnika koja pomaže zaposlenicima da uoče potencijalne znakove pokušaja krađe identiteta i kako ih prijaviti, kao i provjeru autentičnosti s više faktora za zaštitu od pogađanja lozinke ili

brutalnih napada. Penetracijski test elektroničkog socijalnog inženjeringa može se koristiti kao potvrda da su te i druge sigurnosne kontrole pravilno implementirane i da se slijede unutar organizacije.

Kako bi spriječili napad ucjenjivačkog softvera, zdravstvene organizacije trebale bi testirati svoje sustave i osjetljivost na potencijalno iskorištavanje putem penetracijskog testa mreže. Penetracijski test mreže osmišljen je za testiranje različitih aspekata internih i eksternih mreža, simulirajući zlonamjerne aktere koji ciljaju različite dijelove arhitekture organizacije kako bi dobili veći interni pristup.

Najvažnije ranjivosti koje treba identificirati su *host*-ovi i servisi konfigurirani sa slabim ili zadanim lozinkama, kao i oni kojima nedostaju kritične zakrpe. Važno je da zdravstvene tvrtke prepoznaju prednosti koje penetracijsko testiranje može donijeti njihovoj organizaciji. Osim identificiranja potencijalnih sigurnosnih propusta gdje se može pojaviti zlonamjerna aktivnost, praksa također može testirati njihovu sposobnost otkrivanja i odgovora na napade s kojima se njihove organizacije danas suočavaju. Kako bi se osiguralo dosljedno i pouzdano sigurnosno stanje, zdravstvene organizacije moraju potvrditi da se postojeća politika i postupci slijede i da li su učinkoviti za sigurnosno okruženje koje se stalno mijenja, [33].

4.3.3. Implementacija penetracijskog testiranja u proizvodnoj industriji

Napadači se usavršavaju u napadima na industrijske kontrolne sustave zbog njihove uloge u kritičnoj infrastrukturi i mogućih posljedica napada. Električne mreže, obrada vode, skladištenje i transport nafte i plina te moderni proizvodni pogoni ovise o industrijskim sustavima upravljanja (ICS). Od sustava nadzorne kontrole i prikupljanja podataka (SCADA) koji se koriste za nadzor, do distribuiranih kontrolnih sustava (DCS) koji nadziru i reguliraju procese, industrijski kontrolni sustavi ključni su za sigurno odvijanje osjetljivih procesa.

Prekid ovih sustava može dovesti do golemog utjecaja na poslovanje i klijente. Dok su neki napadači financijski motivirani i znaju da će potreba za kontinuitetom operacija natjerati njihove mete da plate otkupninu, drugi su povezani s nacionalnim državama ili političkim skupinama i mogu ciljati na ICS kao način za postizanje svojih strateških ciljeva ili dobivanje utjecaja.

Napadači ciljaju ICS više nego ikada prije. Iako je ICS uvijek bio podoban za aktivnosti kao što su proizvodnja i održavanje kritične infrastrukture, operativna tehnologija nije uvijek bila online. Prelazak na industrijski internet stvari (IIoT) izvrstan je za učinkovitost i sigurnost jer nudi više mogućnosti za praćenje procesa i njihovo podešavanje za učinkovitost i sigurnost. Međutim, to također znači da su kritične operacije povezane na način na koji nikada prije nisu bile. Unošenje i obrada svih ovih podataka s IIoT uređaja zahtijeva mrežnu vezu, zahtijeva rubno računalstvo ili mogućnosti računalstva u oblaku. To znači da su također potrebne odgovarajuće sigurnosne kontrole kako bi se spriječile neovlaštene strane.

Penetracijski testovi najbolji su način za otkrivanje rupa u obrani organizacije, uključujući pogrešne konfiguracije uređaja, ne kriptirani promet, nepravilnu segmentaciju mreže, slab program zakrpa ili izložene ugrađene uređaje koji se ne mogu zakrpati. Samo testiranjem može se uvjeriti da sigurnosne kontrole stvarno rade ispravno i da li je ICS izoliran.

Bolje je i isplativije otkriti slabosti u penetracijskom testu i popraviti ih unaprijed nego se baviti učincima ako ih napadač prvi pronađe. Iskusni ICS penetracijski testeri koriste opreznu metodologiju svjesnu rizika. To znači da je zanemariv rizik od prekida usluge tijekom penetracijskog testa, a tester odlazi s ciljanim i djelotvornim preporukama kako poboljšati obranu od rastuće prijetnje ICS-a.

Procjene ranjivosti daju ideju o poznatim ranjivostima, ali ne zadubljuju se u to koliko ih je moguće iskoristiti. Iako je automatizirana identifikacija ranjivosti faza penetracijskog testiranja, pravi penetracijski test ide daleko dublje. Uključuje stručnjake za ljudsku sigurnost koji rade na otkrivanju koje se ranjivosti zapravo mogu iskoristiti i kakav pristup mogu dobiti iskorištavanjem tih ranjivosti. Ukratko, potrebno je penetracijsko testiranje prodora kako bi se saznalo ne samo što bi napadač mogao iskoristiti u ICS infrastrukturi, već i što zapravo može iskoristiti, te kako bi pomoglo organizaciji da donese informiranije i učinkovitije odluke o popravljanju.

Broj ICS ranjivosti prijavljenih u prvoj polovici 2021. porastao je za 41%, a 71% tih problema klasificirano je kao visoke ili kritične ozbiljnosti. Prema izvješću Clarotya iz 2021., 47% ispitanika identificiralo je napad na ICS ili operativnu tehnologiju u prošloj godini. Ukratko, napadači mogu ciljati na ICS probleme i oni ih aktivno pokušavaju napasti.

ICS su atraktivne mete iz tri glavna razloga:

1. Ucjenjivački softver - napadači znaju da organizacije koje koriste ICS često suočavaju s hitnom potrebom da ostanu u poslu. Jedan od napada za primjer bi bio napad Colonial Pipeline. Napadačima su platili otkupninu u iznosu od 4,4 milijuna dolara u bitcoinima, jer nisu znali razmjere napada i nisu znali koliko će vremena trebati da se u protivnom ponovno uspostavi rad.
2. Industrijska špijunaža - Napadači motivirani špijunažom žele dobiti detaljne informacije o tome kako se odvija poslovni proces. Iako su oni često manje uočljivi od financijski motiviranih napada, ipak imaju duboke dugoročne učinke zbog gubitka poslovnih tajni i konkurentske prednosti.
3. Akteri nacionalne države - Nacionalni državni akteri vide napade na pružatelje kritične infrastrukture kao učinkovit način za postizanje strateških ciljeva, što dovodi do nedavnog porasta ICS napada na takve organizacije. Trenutačni sukobi koji uključuju Rusiju, kako u kontekstu napada na Ukrajinu, tako i napetosti s NATO-om, bili su značajan pokretač.

Pomisao na penetracijsko testiranje industrijskih sustava upravljanja često isprva izaziva strepnju. Vrijeme rada leži u središtu tih strahova: i za proizvođače i za pružatelje kritične infrastrukture, uređaji moraju ostati na mreži kako bi zadovoljili zahtjeve korisnika. Ako se vremenski okvir testiranja prekorači ili se uređaj pokvari zbog neočekivanog odgovora na skeniranje ili simuliranog napada, ugled i sposobnost ispunjavanja ugovora mogu biti ugroženi. Međutim, iskusni stručnjaci znaju kako sigurno testirati unutar ICS okruženja, štiteći poslovne procese dok istovremeno jačaju sigurnost za budućnost.

Penetracijsko testiranje industrijskih sustava upravljanja zahtijeva drugačiji pristup od ostalih vrsta penetracijskog testiranja. Alati poput skenera i *fuzzera* korisni su protiv sredstava IT-a ili web-aplikacija, ali mogu uzrokovati prekide usluge kada se koriste za testiranje ICS-a. Iskusni ICS penetracijski testeri znaju da ove sustave treba testirati na oprezan i metodičan način, te koristiti alate i metode posebno dizajnirane za otkrivanje problema koji se mogu iskoristiti, a istovremeno minimizirati rizike testiranja. Iskusni tester ne samo da zna koliko pažnje zahtijeva testiranje protiv ICS-a, već zna i uobičajene probleme s kojima se suočavaju tvrtke koje koriste ICS. Često tvrtke pokušavaju riješiti svoje ICS probleme odvajanjem uređaja. Ali pokušaji segmentacije obično nisu tako učinkoviti kao što tvrtke misle da jesu. Uobičajeni razlozi za nedostatke perimetra ili segmentacije uključuju pristup dobavljača, tehničare koji pokušavaju olakšati upravljanje stvarima ili slabosti u konfiguracijama vatrozida. Jedini način da se otkriju ti nedostaci jest testiranje, a iskusan ICS tester dobro je vješt u traženju ovih problema.

Drugi uobičajeni problemi identificirani u ICS testovima odnose se na smetnje putem tehnika koje tvrtke možda ne očekuju. Na primjer, većina tvrtki shvaća mogućnost da će napadač kompromitirati kontroler. Međutim, oni možda ne razmišljaju o činjenici da napadač može izazvati sličan prekid kompromitacijom mrežnog uređaja i selektivnim odgađanjem prometa povezanog s procesom u koji napadač želi ometati. Iskusni ICS testeri znaju kako identificirati i testirati ove sekundarne uređaje koji mogu imati neočigledne učinke na ICS sustave, [34].

5. Primjena penetracijskog testiranja u oblaku

Cloud je ključna komponenta svakog sustava zbog svojih prednosti kao što su računalna sposobnost, učinkovitost i upotrebljivost na internetu. Uz sve ove dobre karakteristike, također je jeftiniji za korištenje od strane organizacija zahvaljujući modelu plaćanja oblaka "plaćaj po hodu (PAYG)" bez plaćanja prije korištenja. Budući da se oblak intenzivno koristi, ključno je osigurati njegovu sigurnost. U oblaku je svaki dio sustava međusobno povezan internetom, stoga svaki mali sigurnosni propust može utjecati na veći i kritičniji dio infrastrukture sustava, [35].

Izgradnja poslovanja temeljenog na oblaku ili migracija informacijskih sredstava u oblak donosi operativnu učinkovitost i isplativost. Pružatelji usluga u oblaku prate sigurnosne propise i privatnost podataka. Penetracijsko testiranje u oblaku je proces otkrivanja sigurnosnih ranjivosti u infrastrukturi oblaka putem kontroliranog kibernetičkog napada. To se izvodi prema smjernicama pružatelja usluga poput Amazon mrežnih usluga (AWS) i Google platforme u oblaku (GCP) kako bi se otkrile i otklonile ranjivosti prije nego što ih iskoriste zlonamjerni napadači. Penetracijsko testiranje je ofenzivno testiranje sustava, usluga ili mreža kako bi se otkrile sigurnosne slabosti. U kontekstu oblaka, to je simulirani napad na oblak usluge radi testiranja njihove sigurnosti., [36].

5.1. Modeli i sigurnost usluga u oblaku

IaaS, PaaS i SaaS tri su najpopularnije vrste ponuda usluga u oblaku. Ponekad se nazivaju modeli usluga u oblaku ili modeli usluga računarstva u oblaku.

- IaaS, ili infrastruktura kao usluga, pristup je na zahtjev fizičkim i virtualnim poslužiteljima koji se nalaze u oblaku, pohrani i umrežavanju - pozadinskoj IT infrastrukturi za pokretanje aplikacija i radnih opterećenja u oblaku.
- PaaS ili platforma kao usluga je pristup na zahtjev cjelovitoj platformi koja se nalazi u oblaku i koja je spremna za korištenje za razvoj, pokretanje, održavanje i upravljanje aplikacijama.
- SaaS ili softver kao usluga je pristup na zahtjev spremnom za korištenje, aplikacijski softver smješten u oblaku.

IaaS, PaaS i SaaS se međusobno ne isključuju. Mnoga poduzeća srednje veličine koriste više od jednog, a većina velikih poduzeća koristi sva tri. 'Kao usluga' odnosi se na način na koji se IT sredstva troše u ovim ponudama - i na suštinsku razliku između računalstva u oblaku i tradicionalnog IT-a. U tradicionalnom IT-u, organizacija troši IT imovinu - hardver, sistemski softver, razvojne alate, aplikacije - kupujući ih, instalirajući ih, upravljajući njima i održavajući ih u vlastitom lokalnom podatkovnom centru. U računalstvu u oblaku, pružatelj usluga u oblaku posjeduje, upravlja i održava imovinu; kupac ih konzumira putem internetske veze, a plaća ih na temelju pretplate ili pay-as-you-go principa.

Dakle, glavna prednost IaaS-a, PaaS-a, SaaS-a ili bilo kojeg rješenja 'kao usluge' je ekonomska: korisnik može pristupiti i skalirati IT mogućnosti koje su mu potrebne za predvidljivu cijenu, bez troškova i režijskih troškova kupnje i održavanja toga svega da je posjedovao svoj vlastiti podatkovni centar.

5.1.1. IaaS, infrastruktura kao usluga

IaaS je pristup na zahtjev računalnoj infrastrukturi smještenoj u oblaku - poslužiteljima, kapacitetima za pohranu i mrežnim resursima - koje korisnici mogu osigurati, konfigurirati i koristiti na gotovo isti način kao što koriste hardver. Razlika je u tome što pružatelj usluga u oblaku ugošćuje, upravlja i održava hardver i računalne resurse u svojim vlastitim podatkovnim centrima. Korisnici IaaS-a koriste hardver putem internetske veze i tu upotrebu plaćaju na temelju pretplate ili tekućeg plaćanja.

Obično korisnici IaaS-a mogu birati između virtualnih strojeva smještenih na dijeljenom fizičkom hardveru (pružatelj usluga u oblaku upravlja virtualizacijom) ili golih poslužitelja na namjenskom (nedijeljenom) fizičkom hardveru. Korisnici mogu osigurati, konfigurirati i upravljati poslužiteljima i infrastrukturnim resursima putem grafičke nadzorne ploče ili programski putem sučelja za programiranje aplikacija (API). IaaS se može smatrati izvornom ponudom 'kao usluge': svaki veći pružatelj usluga u oblaku - Amazon Web Services, Google Cloud, IBM Cloud, Microsoft Azure - započeo je s ponudom nekog oblika IaaS-a.

IaaS korisnicima omogućuje izbjegavanje početnih troškova i režijskih troškova kupnje i održavanja vlastitog lokalnog podatkovnog centra. Također eliminira stalni kompromis između rasipanja kupnje viška lokalnog kapaciteta za prilagodbu šiljcima, naspram slabe izvedbe ili prekida koji mogu proizaći iz nedostatka dovoljnog kapaciteta za neočekivane udare ili rast prometa. Druge prednosti IaaS-a uključuju:

- Veća dostupnost - s IaaS-om tvrtka može lako stvoriti redundantne poslužitelje, pa čak i stvoriti ih u drugim zemljopisnim područjima kako bi osigurala dostupnost tijekom lokalnih nestanaka struje ili fizičkih katastrofa.
- Niža latencija, poboljšana izvedba - budući da IaaS pružatelji usluga obično upravljaju podatkovnim centrima na više geografskih područja, IaaS korisnici mogu locirati aplikacije i usluge bliže korisnicima kako bi smanjili latenciju i povećali performanse.
- Poboljšana brzina odziva - korisnici mogu osigurati resurse u roku od nekoliko minuta, brzo testirati nove ideje i brzo predstaviti nove ideje većem broju korisnika.
- Sveobuhvatna sigurnost - s visokom razinom sigurnosti na licu mjesta, u podatkovnim centrima i putem enkripcije, organizacije često mogu iskoristiti napredniju sigurnost i zaštitu koju bi mogle pružiti da u svojoj kući drže infrastrukturu oblaka.
- Brži pristup najboljoj tehnologiji - pružatelji usluga u oblaku međusobno se natječu pružajući najnovije tehnologije svojim korisnicima.

Uobičajene upotrebe IaaS-a uključuju:

- Oporavak od katastrofe - umjesto postavljanja redundantnih poslužitelja na više lokacija, IaaS može implementirati svoje rješenje za oporavak od katastrofe na postojeću geografski raspršenu infrastrukturu pružatelja usluga oblaka.
- E-trgovina - IaaS je izvrsna opcija za online trgovce koji često bilježe porast prometa. Sposobnost povećanja tijekom razdoblja velike potražnje i visokokvalitetna sigurnost ključni su u današnjoj maloprodajnoj industriji koja radi 24 sata dnevno.
- Internet stvari (IoT) obrada događaja, umjetna inteligencija (AI) - IaaS olakšava postavljanje i povećanje pohrane podataka i računalnih resursa za ove i druge aplikacije koje rade s ogromnim količinama podataka.
- *Start-up-i* - ne mogu priuštiti ulaganje kapitala u lokalnu IT infrastrukturu. IaaS im daje pristup mogućnostima podatkovnog centra poslovne klase bez prethodnog ulaganja u hardver i troškove upravljanja.
- Razvoj softvera - uz IaaS, infrastruktura za testiranje i razvojna okruženja može se postaviti mnogo brže nego lokalno. (Međutim, ovaj slučaj upotrebe je prikladniji za PaaS), [37].

Pružatelj usluga u oblaku (CSP) odgovoran je za osiguranje infrastrukture i sloja apstrakcije koji se koristi za pristup resursima. Sigurnosne obveze organizacije pokrivaju ostatak slojeva, koji uglavnom sadrže poslovne aplikacije. Da bi se bolje vizualizirao sigurnosni problem mreže u oblaku, potrebno je implementirati mrežnog paketnog posrednika (NPB) u IaaS okruženju. NPB šalje promet i podatke sustavu upravljanja performansama mreže i relevantnim sigurnosnim alatima. Osim toga, potrebno je uspostaviti bilježenje događaja koji se događaju na krajnjim točkama mreže. Implementacije IaaS oblaka zahtijevaju sljedeće dodatne sigurnosne značajke: segmentaciju mreže, sustav za otkrivanje upada i sustav za sprječavanje upada, virtualne vatrozide postavljene ispred web-aplikacija radi zaštite od zlonamjernog koda i na rubu mreže u oblaku i virtualne usmjerivače, [38].

5.1.2. PaaS, platforma kao usluga

PaaS pruža platformu temeljenu na oblaku za razvoj, pokretanje i upravljanje aplikacijama. Pružatelj usluga u oblaku ugošćuje, upravlja i održava sav hardver i softver uključen u platformu - poslužitelje (za razvoj, testiranje i implementaciju), softver operativnog sustava, pohranu, umrežavanje, baze podataka, okvire, razvojne alate - kao i povezane usluge za sigurnost, nadogradnju operativnog sustava i softvera, sigurnosne kopije i više.

Korisnici pristupaju PaaS-u putem grafičkog korisničkog sučelja (GUI), gdje razvojni ili *DevOps* timovi mogu surađivati na svom radu tijekom cijelog životnog ciklusa aplikacije, uključujući kodiranje, integraciju, testiranje, isporuku, implementaciju i povratne informacije. Primjeri PaaS rješenja uključuju AWS Elastic Beanstalk, Google App Engine, Microsoft Windows Azure i Red Hat OpenShift na IBM Cloudu.

Primarna prednost PaaS-a je u tome što korisnicima omogućuje izgradnju, testiranje, implementaciju, ažuriranje i skaliranje aplikacija brže i isplativije nego što bi to mogli da izgrade i upravljaju vlastitom platformom u prostoru. Ostale pogodnosti uključuju:

- Brže vrijeme za izlazak na tržište - PaaS omogućuje razvojnim timovima da pokrenu razvojna, testirajuća i proizvodna okruženja za nekoliko minuta, u odnosu na tjedne ili mjesece.
- Testiranje s niskim do nikakvim rizikom i usvajanje novih tehnologija - PaaS platforme obično uključuju pristup širokom rasponu najnovijih resursa gore i dolje na hrpi aplikacija. To omogućuje tvrtkama da testiraju nove operativne sustave, jezike i druge alate bez potrebe za značajnim ulaganjem u njih ili u infrastrukturu potrebnu za njihovo pokretanje.
- Pojednostavljena suradnja - kao usluga temeljena na oblaku, PaaS pruža zajedničko razvojno okruženje softvera, dajući razvojnim i operativnim timovima pristup svim alatima koji su im potrebni, s bilo kojeg mjesta s internetskom vezom.

PaaS može unaprijediti niz razvojnih i IT inicijativa uključujući:

- Razvoj i upravljanje API-jem - sa svojim ugrađenim okvirima, PaaS olakšava timovima razvoj, pokretanje, upravljanje i osiguranje API-ja za dijeljenje podataka i funkcionalnosti između aplikacija.
- Internet stvari (IoT) - PaaS podržava niz programskih jezika (Java, Python, Swift itd.), alate i aplikacijska okruženja koja se koriste za razvoj IoT aplikacija i obradu podataka s IoT uređaja u stvarnom vremenu.
- Izvorni razvoj u oblaku i hibridna strategija u oblaku - PaaS rješenja podržavaju razvojne tehnologije izvorne u oblaku - mikroservisi, spremnici, računalstvo bez poslužitelja - koje programerima omogućuju da izgrade samo jednom, zatim implementiraju i dosljedno upravljaju u privatnom oblaku, javnom oblaku i lokalnim okruženjima, [37].

PaaS platforme omogućuju organizacijama izradu aplikacija bez dodatnih troškova i složenosti povezanih s upravljanjem hardverom i pozadinskim softverom. U modelu PaaS, CSP štiti većinu okoliša. Međutim, tvrtka je i dalje odgovorna za sigurnost aplikacija koje razvija. Stoga je sigurnosna arhitektura PaaS slična modelu SaaS. Potrebno je imati CASP, bilježenje i upozoravanje, IP ograničenja i API pristupnik kako bi se osigurao siguran unutarnji i vanjski pristup API-jima tih aplikacija, [38].

5.1.3. SaaS, softver kao usluga

SaaS (ponekad zvane i aplikacijske usluge u oblaku) je aplikacijski softver koji se nalazi u oblaku i spreman je za korištenje. Korisnici plaćaju mjesečnu ili godišnju naknadu za korištenje cjelovite aplikacije unutar web preglednika, desktop klijenta ili mobilne aplikacije. Aplikaciju i svu infrastrukturu potrebnu za njezinu isporuku -

poslužitelje, pohranu, umrežavanje, posredni softver, aplikacijski softver, pohranu podataka - *hostira* i upravlja SaaS dobavljač.

Dobavljač upravlja nadogradnjama i zakrpama softvera, osiguravajući razinu dostupnosti, performansi i sigurnosti u sklopu SLA sporazuma. Kupci mogu proširiti uslugu uz dodatne troškove. Mobilni korisnici često koriste SaaS, primjerice e-poštu, društvene medije i rješenja za pohranu u oblaku poput Dropboxa ili Boxa. Poslovna SaaS rješenja uključuju Salesforce, HubSpot, Trello, Slack i Canva. Mnoge aplikacije sada dostupne kao SaaS, kao što je Adobe Creative Cloud.

Glavna prednost SaaS-a je ta što svu infrastrukturu i upravljanje aplikacijama prebacuje na SaaS dobavljača. Sve što korisnik treba učiniti je kreirati račun, platiti naknadu i početi koristiti aplikaciju. Dobavljač se brine za sve ostalo, od održavanja hardvera i softvera poslužitelja do upravljanja korisničkim pristupom i sigurnošću, pohranjivanja i upravljanja podacima, implementacije nadogradnji i zakrpa i više.

Druge prednosti SaaS-a uključuju:

- Minimalni rizik - mnogi SaaS proizvodi nude besplatno probno razdoblje ili niske mjesečne naknade koje korisnicima omogućuju isprobavanje softvera da vide hoće li zadovoljiti njihove potrebe, uz malo ili nimalo financijskog rizika.
- Produktivnost bilo kada/bilo gdje - korisnici mogu raditi sa SaaS aplikacijama na bilo kojem uređaju s preglednikom i internetskom vezom.
- Jednostavna skalabilnost - dodavanje korisnika jednostavno je poput registracije i plaćanja novih mjesta; korisnici mogu kupiti više prostora za pohranu podataka za nominalnu cijenu, [37].

SaaS usluge omogućuju pristup softverskim aplikacijama i podacima putem preglednika. Specifični uvjeti odgovornosti za sigurnost mogu se razlikovati od usluge do usluge. Sigurnosni posrednici za pristup oblaku (CASB) nude mogućnosti zapisivanja, revizije, kontrole pristupa i enkripcije koje mogu biti kritične kada se istražuju sigurnosni problemi u SaaS proizvodu. Osim toga, u SaaS okruženju je potrebno imati: Zapisivanje i upozoravanje, popise dopuštenih IP-a i/ili popise zabrane, pristupnike API-ja, u slučaju da se usluzi pristupa putem API-ja, [38].

5.2. Prednosti i nedostaci penetracijskog testiranja u oblaku

Penetracijsko testiranje u oblaku od strane trećih strana nije korisno samo za pružatelje usluga u oblaku, već i za organizacije koje pohranjuju svoje osjetljive podatke i aplikacije u oblaku. Penetracijsko testiranje u oblaku pomaže u održavanju modela dijeljene odgovornosti koji većina pružatelja usluga u oblaku postavlja između sebe i korisnika putem:

Pomaže u identifikaciji ranjivosti - identifikacija bilo koje ranjivosti provođenjem penetracijskih testova u oblak osigurava njihovo brzo popravljanje. Sveobuhvatni skeneri mogu otkriti i najsitnije ranjivosti. To je ključno jer pomaže u trenutnom saniranju prije nego što hakeri iskoriste ranjivost.

Poboljšava sigurnost oblaka i aplikacija - još jedna prednost penetracijskog testiranja u oblak je ta što pomaže u stalnom ažuriranju sigurnosnih mjera te pomaže poboljšati postojeće sigurnosne mjere ako se u njima pronađu bilo kakve sigurnosne rupe.

Povećava pouzdanost među pružateljima i korisnicima - provođenje povremenih penetracijskih testova u oblaku može pomoći u povećanju pouzdanosti i pouzdanosti koji se pripisuju pružateljima usluga u oblaku. To može dovesti više klijenata zahvaljujući prirodi pružatelja usluga oblaka koja vodi računa o sigurnosti, dok će postojeći klijenti biti zadovoljni razinom zaštite dostupnih podataka koje pohranjuju.

Pomaže u održavanju usklađenosti - provođenje penetracijskih testova u oblaku ne pomaže samo u pronalaženju ranjivosti, već i kod područja neusklađenosti s različitim regulatornim standardima. Stoga se ta područja koja su identificirana mogu ispraviti kako bi se ispunili zahtjevi usklađenosti i izbjegle novčane kazne za nepoštivanje.

Postoji dosta ranjivosti koje mogu dovesti do kompromitiranja računa u oblaku a neke od njih su:

Nesigurni API-ji - API-ji se naširoko koriste u uslugama u oblaku za dijeljenje informacija među različitim aplikacijama. Međutim, nesigurni API-ji također mogu dovesti do curenja podataka velikih razmjera kao što je viđeno u slučaju Venmo-a, Airtel-a, itd. Ponekad nepropisno korištenje HTTP metoda kao što su PUT, POST, DELETE u API-jima može dopustiti hakerima da učitaju zlonamjerni softver na korisnikov poslužitelj ili izbrisati podatke. Nepravilna kontrola pristupa i nedostatak dezinfekcije ulaza također su glavni uzroci ugrožavanja API-ja koji se mogu otkriti tijekom testiranja prodora u oblak.

Pogrešne konfiguracije poslužitelja - pogrešne konfiguracije usluga u oblaku danas su najčešća ranjivost u oblaku (osobito pogrešno konfigurirane S3 *bucketi*). Najpoznatiji slučaj je bio curenje podataka Capital One koji je doveo do kompromitacije podataka otprilike 100 milijuna Amerikanaca i 6 milijuna Kanađana. Najčešće pogrešne konfiguracije poslužitelja u oblaku su neodgovarajuće dozvole, ne kriptiranje podataka i razlika između privatnih i javnih podataka.

Slabe vjerodajnice - korištenje uobičajenih ili slabih zaporki može učiniti račune u oblaku ranjivima na napade grubom silom. Napadač može upotrijebiti automatizirane alate za nagađanje i na taj način ući u račun pomoću tih vjerodajnica. Rezultati bi mogli biti katastrofalni i dovesti do potpunog preuzimanja računa. Budući da ljudi imaju tendenciju ponovnog korištenja zaporki i lako pamtljivih zaporki, ovi su napadi prilično česti.

Zastarjeli softver - sadrži kritične sigurnosne propuste koji mogu ugroziti usluge u oblaku. Većina dobavljača softvera ne koristi pojednostavljeni postupak ažuriranja ili korisnici sami onemogućuju automatsko ažuriranje. Zbog toga su usluge

u oblaku zastarjele, a hakeri ih identificiraju pomoću automatiziranih skenera. Kao rezultat toga, veliki broj ugroženih usluga u oblaku koji koriste zastarjeli softver.

Nesigurne prakse kodiranja - većina tvrtki pokušava izgraditi svoju infrastrukturu u oblaku za što jeftiniju cijenu. Dakle, zbog loše prakse kodiranja, takav softver često sadrži pogreške poput SQLi, XSS, CSRF. One koje su među njima najčešće označene su kao OWASP top 10. Upravo su te ranjivosti glavni uzrok ugroženosti većine web usluga u oblaku, [36].

5.3. Najbolje prakse penetracijskog testiranja u oblaku

Alati za penetracijsko testiranje u oblaku također bi trebali nuditi kontinuirano i sveobuhvatno skeniranje ranjivosti kako bi se procijenile i pronašle sve ranjivosti unutar sustava u oblaku. Trebale bi se tražiti ranjivosti na temelju poznatih ranjivosti iz CVE-a, intel-a, OWASP Top 10 i SANS 25. Također bi trebalo biti u mogućnosti skenirati iza prijava i pronaći sve pogreške poslovne logike.

Redoviti penetracijski testovi ključni su za sigurnost okruženja u oblaku od strane kupaca i pružatelja kako bi se analizirale i iskoristile ranjivosti unutar sigurnosnog sustava. Rezultati takvog penetracijskog testa detaljno će opisati pronađene nedostatke zajedno s mjerama koje se mogu poduzeti da se poprave prije nego što ih zlonamjerni napadači iskoriste.

Vatrozid temeljen na oblaku netradicionalno je rješenje za održavanje sigurnosti podataka koji se pohranjuju i prenose s oblakom. Ti se vatrozidi nalaze u samom oblaku. Vatrozidi temeljeni na oblaku lako su skalabilni prema potrebama pružatelja usluga oblaka ili korisnika.

Osiguranje podataka koje korisnici u oblaku prenose i pohranjuju apsolutno je od kritične važnosti. Ovdje dolazi do izražaja enkripcija podataka. Šifriranjem podataka koji miruju i u prijenosu korištenjem sigurnosti transportnog sloja osigurava se da podatke ne mogu dešifrirati pogrešne strane, čime se održava povjerljivost.

Potrebno je osigurati da alat koji je odabran za procjenu sigurnosti oblaka ima odgovarajuće mjere za otkrivanje svih neovlaštenih aktivnosti i pružanje upozorenja u stvarnom vremenu za iste. Strojno učenje može pomoći sigurnosnim mjerama u oblaku da prepoznaju obrasce i time otkriju aktivnosti koje su izvan utvrđenih sigurnosnih obrazaca.

Penetracijski test u oblaku također osigurava da se kao klijent ili pružatelj usluga u oblaku poštuje usklađenost koja se mora održavati kao što su HIPAA, Standard sigurnosti podataka industrije platnih kartica (PCI-DSS), GDPR i drugi zakoni o zaštiti podataka.

Kontrolni popis za testiranje prodora u oblak:

1. Identificiranje pružatelja usluga u oblaku i usluge koje se koriste.
2. Razumijevanje obveza, pravila i propisa.

3. Razumijevanje sigurnosnih kontrola pružatelja usluga oblaka.
4. Utvrđivanje tko ima pristup okruženju oblaka i koju razinu pristupa ima.
5. Priprema plana za penetracijski test.
6. Pokretanje penetracijskog testa i kontinuirano praćenje njegovog napretka.
7. Potrebno je osigurati da postoje jake kontrole provjere autentičnosti i autorizacije.
8. Provođenje načela najmanje privilegija.
9. Izrada izvješća penetracijskog testa.
10. Održavanje okruženja u oblaku ažuriranim s najnovijim sigurnosnim zakrpama i ažuriranjima.

5.4. Izazovi penetracijskog testiranja u oblaku

Uvođenje tehnologija u oblaku donosi nove dinamike i izazove u svijetu kibernetičke sigurnosti. Penetracijsko testiranje u oblaku suočava se s kompleksnim okruženjima, dinamičkim resursima i virtualnim mrežama. Izazovi sa kojima se penetracijski testeri mogu susresti su:

Nedostatak transparentnosti - neke usluge u oblaku imaju podatkovne centre kojima upravljaju treće strane. Korisnici možda nisu svjesni gdje su podaci pohranjeni i koja se konfiguracija hardvera ili softvera koristi, čime se korisnički podaci izlažu sigurnosnim rizicima na usluzi u oblaku. Na primjer, pružatelj usluga u oblaku možda čuva osjetljive podatke bez znanja korisnika. Štoviše, poznato je da popularni CSP-ovi kao što su AWS, Azure, GCP itd. provode interne sigurnosne revizije, dok korisnici mogu koristiti tvrtke za sigurnost u oblaku. Nedostatak transparentnosti znači da te resurse ne može revidirati sigurnosni revizor po ispitivačevom izboru. Kao rezultat toga, ispitivač neće moći odgovoriti ako su ti temeljni resursi zaraženi.

Dijeljenje resursa - dobro je poznata činjenica da usluge u oblaku dijele resurse na više računa. Međutim, ovo dijeljenje resursa može se pokazati izazovnim tijekom penetracijskog testiranja u oblaku. Ponekad pružatelji usluga ne poduzimaju odgovarajuće korake za segmentaciju svih korisnika. U tim slučajevima, ako tvrtka mora biti usklađena s PCI DSS-om, standard kaže da svi drugi računi koji dijele resurs i pružatelja usluga u oblaku također trebaju biti usklađeni s PCI DSS-om. Ovakvi složeni scenariji prisutni su jer postoji više načina implementacije infrastrukture oblaka. Ova složenost otežava proces testiranja prodora u oblak.

Ograničenja politike - svaki pružatelj usluga u oblaku ima vlastitu politiku u vezi s provođenjem penetracijskog testiranja u oblaku. Ovo definira krajnje točke i vrste testova koji se mogu provesti. Uzet će se sada kratki pregled politike penetracijskog testiranja u oblaku 3 najpopularnija pružatelja usluga u oblaku:

1. AWS - Postoji 8 dopuštenih usluga za web usluge Amazona na kojima se penetracijsko testiranje u oblaku može izvršiti bez prethodne najave. One su navedene u dopuštenim uslugama politike. Međutim, ako tester želi provesti test mrežnog stresa, za to postoji posebna politika.

2. Azure dopušta penetracijsko testiranje u oblaku na osam Microsoftovih proizvoda koji se spominju u njihovoj politici. Sve izvan toga je izvan dosega.
3. Za Google Cloud Platform-u ne postoji posebna politika penetracijskog testiranja u oblaku kao takva, samo je potrebno slijediti njihova pravila prihvatljive upotrebe i uvjete usluge. Nije potrebno obavijestiti Google prije provođenja testova. Međutim, pravila prihvatljive upotrebe spominju nekoliko stvari koje se ne bi trebalo činiti: slanje neželjene pošte, kršenje prava drugih GCP korisnika ili provođenje penetracijskih testova na njima, kršenje ili pokušaj zaobilaženja uvjeta pružanja usluge, ometanje opreme koja podržava GCP.

Ostali faktori - zbog same veličine usluga u oblaku, jedan stroj može ugostiti više virtualnih mašina (VM), što povećava opseg penetracijskog testiranja u oblaku. Također, opseg takvih testova može varirati od korisničkog softvera do softvera pružatelja usluga. Oba ova čimbenika zajedno dodatno povećavaju složenost penetracijskog testiranja u oblaku. Kada se enkripcija doda na ovaj popis, to može dodatno pogoršati situaciju za revizore jer tvrtka koja se revidira možda neće biti voljna dijeliti ključeve enkripcije.

5.5. Procesi izvođenja penetracijskog testiranja u oblaku

Prije početka testiranja važno je formulirati plan testiranja temeljen na politici pružatelja usluga u oblaku. To je zato što svaki CSP ima vlastitu politiku u vezi s:

- Vrste penetracijskog testa oblaka koji se mogu izvesti.
- Krajnje točke koje se mogu testirati.
- Dopuštenja za izvođenje testova.
- Opseg testova.

Dakle, ako plan testiranja nije u skladu s tim, pružatelj usluga može kazniti ispitivače. Na primjer, ako ispitivač pokuša testirati svoj račun na DDOS, a CSP to ne dopušta, postoje automatski sustavi koji to mogu otkriti. Nakon toga, CSP može zaključati njegov račun na neko vrijeme i morat će obaviti mnogo objašnjenja prije nego što mu vrate njegov račun. Suština postupka dobivanja je upoznavanje s CSP politikom.

Drugi dio je izrada plana za izvođenje penetracijskog testiranja u oblaku. Ne postoji određena formula za izradu plana jer se razlikuje od revizora do revizora. No, neki od koraka koje ispitivač može poduzeti da bi formulirao plan su:

1. Mapiranje svih krajnjih točki kao što su korisničko sučelje, API-ji, podmreže itd. za koje treba provesti testiranje.
2. Potrebno je odlučiti koje će se krajnje točke isključiti na temelju ograničenja pravila, korisničkih dopuštenja itd.
3. Potrebno je izabrati rutu za izvođenje penetracijskog testa, tj. iz aplikacije ili baze podataka.

4. Potrebno je odrediti koliko dobro aplikacijski poslužitelj i VM mogu podnijeti opterećenje testova.
5. Potrebno je saznati koje sve zakone treba poštovati tijekom izvođenja testova.
6. Potrebno je odrediti koji će se alati koristiti i koje će se vrste testova izvoditi na kojim krajnjim točkama (automatski ili ručno).
7. Konačno, od klijenta ispitivač dobije odobrenje za svoj plan i obavijesti ga kada želi započeti.

Potom je potrebno pokreniti alate i promatrati odgovore na ranjivost. Iako su neki alati kao što su Nmap, Sqlmap, OpenVAS itd. dobro poznati, postoje i neki alati specifični za CSP koji se mogu koristiti:

- AWS Inspector: prilagođeno sigurnosno rješenje za AWS. Može se koristiti kao osnovni minimum ili alat za preliminarno testiranje.
- S3Scanner: alat otvorenog koda za skeniranje S3 spremnika u potrazi za pogrešnim konfiguracijama i ispisivanje njihovih podataka.
- MicroBurst: zbirka PowerShell skripti za skeniranje Azure usluga radi sigurnosnih problema. Dakle, da bi se koristio potrebno je imati instaliran PowerShell koji je prisutan prema zadanim postavkama na Windows OS-u.
- Azucar: još jedan popularan Azure alat za skeniranje izgrađen pomoću PowerShell-a baš kao i MicroBurst.
- Cloudsploit: popularan alat otvorenog koda koji može skenirati više vrsta pružatelja usluga u oblaku kao što su Azure, AWS, Google Cloud Platform itd.

Neki od automatiziranih alata mogu generirati lažne rezultate. Stoga je prije dodavanja u izvješće potrebno provjeriti može li se svaki od njih iskoristiti. Potrebno je ponoviti ovaj postupak za svaki sloj (mreža, baza podataka, aplikacija itd.) koji se testira. Slijedi, generiranje izvješća. Za penetracijske testere u oblaku važno je predstaviti ranjivosti klijentu na razumljiv način. Prezentacija je razlika između toga da klijent shvaća ranjivosti ozbiljno ili neozbiljno. Dakle, potrebno je provjeriti jesu li izvješća dobro organizirana i kategorizirana na temelju vrste i razine prijetnje.

Nakon što su ranjivosti pronađene, potrebno je stupiti u kontakt sa programerima kako bi se ranjivosti zakrpale. Neke se ranjivosti mogu popraviti uvođenjem manjih izmjena koda, dok neke mogu zahtijevati značajnu reviziju. Međutim, ako testovi nisu uspjeli otkriti nikakvu ranjivost, možda je potrebno promijeniti plan i provesti detaljnije sigurnosne testove, [36].

6. Automatizacija penetracijskog testiranja

Automatizirano penetracijskog testiranja je kada ispitivač koristi softver za automatizaciju nekih ili svih otkrivanja i iskorištavanja sigurnosnih propusta u mrežama, infrastrukturi oblaka, web stranica te web i mobilnih aplikacija. Automatizirano testiranje brzo identificira ranjivosti korištenjem strojnog učenja, algoritama i informacija o prijetnjama. Međutim, automatizirano testiranje ne može zamijeniti potrebu za ljudskom stručnošću u planiranju, analizi i tumačenju rezultata.

Dok skeniranje ranjivosti ispituje poznate ranjivosti i stvara izvješće koje se može koristiti za smanjenje rizika, automatizirano penetracijsko testiranje nije samo skeniranje ranjivosti, već ima za cilj pronaći i iskoristiti sigurnosni jaz. Penetracijsko testiranje pruža emulaciju površinskih slojeva napada i kontinuiranu provjeru rezultata, a skener ranjivosti skenira mreže, računala i sigurnosne ranjivosti i slabosti sustava. Ovaj se proces može u određenoj mjeri automatizirati kako bi se vidjelo koja bi se imovina mogla iskorištavati.

Provjere koje izvodi automatizirano testiranje penetracije su:

1. Ranjivosti:
    - Ranjivost na SQL injekcije
    - Ranjivost na XSS
    - Krivotvorenje zahtjeva na različitim mjestima
    - Otkrivanje informacija – osjetljive informacije u URL-u, zaglavlju HTTP preporuke, poruke o pogreškama
    - Slaba autentifikacija
    - Provjerava nedostaju li sigurnosna zaglavlja
    - Otkrivanje PII
    - Javno dostupne datoteke
    - Neovlašten pristup
2. Vrsta grešaka:
    - Uključivanje JavaScript izvorne datoteke među domenama
    - Odsutnost anti-CSRF tokena
    - Nedostaje SSL
    - Obrnuti tabnabbing
    - Nesigurni kolačići
    - Trovanje kolačićima
    - .htaccess curenje informacija
    - Proxy otkrivanje
    - Zastarjela verzija itd.

6.1. Usporedba ručnog i automatskog penetracijskog testiranja

Dva glavna nedostatka ručnog penetracijskog testiranja su cijena testa i vrijeme potrebno za izvođenje testa. Ovisno o opsegu penetracijskog testa, mogli bi proći tjedni da se dobiju valjani i upotrebljivi rezultati, što nije uvijek poželjno, posebno ako na ciljanim sustavima postoje kritične i visoke ranjivosti. Naslijeđeni alati za simulaciju sigurnosnih napada temeljenih na agentima uvode dodatne troškove i nedostatke u pokrivenosti koji ne ispunjavaju očekivanja. Penetracijsko testiranje nudi samo snimku trenutka trenutne izloženosti. Nedostatak je što će izloženost u većini slučajeva biti drugačija sljedećih dana, tjedana, godina, što može ograničiti širu vidljivost rizika.

Iako je ručno testiranje penetracije ključan proces, također je težak, skup i dugotrajan. Prednosti automatizacije su jeftiniji i lakši pristup penetracijskom testiranju. Redovito automatizirano penetracijsko testiranje omogućuje tvrtkama da procijene svoju cjelokupnu računalnu infrastrukturu, koja se može ažurirati češće od ručnog testiranja prodora, na primjer, tijekom ciklusa brzog izdavanja.

Automatizirane platforme za penetracijsko testiranje uvelike smanjuju utrošak vremena, kao i operativne troškove pružajući valjanije i brže rezultate u usporedbi s tradicionalnim rješenjima za penetracijsko testiranje kojima mogu proći tjedni da se dobiju ispravni rezultati. Automatizirani alati za testiranje prodora mogu se ponovno koristiti i pokrenuti više puta, što dugoročno može uštedjeti troškove.

Prednosti tradicionalnog penetracijskog testiranja dodatno su povećane automatiziranim sigurnosnim testiranjem, uključujući današnje platforme za simulaciju proboja i napada (Breach And Attack Simulation - BAS) izvrstan su primjer toga. Kako bi se pronašle vidljive i skrivene ranjivosti, BAS platforme će kontinuirano simulirati stvarne napade na ciljana okruženja, koristeći način razmišljanja i taktike napadača i u procesu potvrđujući učinkovitost obrambenih kontrola. Automatizirani alati za penetracijsko testiranje za određivanje ranjivosti i rizika nisu savršeni, a rješenje za taj izazov bila bi kontinuirana provjera sigurnosti. Tijekom kontinuirane sigurnosne validacije, važno je ubrzati ciklus validacije-popravke davanjem prioriteta sanaciji kritičnih i visokih ranjivosti čim se otkriju za koje postoji veći rizik od iskorištavanja.

Za pravilno upravljanje stvarnom izloženošću kritične imovine važno je promijeniti način razmišljanja o testiranju u određenom trenutku na kontinuiranu sigurnosnu provjeru. Otpornost kibernetičke sigurnosti treba kontinuirano poboljšavati. Upravljanje agentima kroz cijelu infrastrukturu kako bi imali veću vidljivost može biti izazovno zbog dugotrajnih procesa, kao i šire kompatibilnosti sustava. Najbolji pristup bio bi imati automatizirani alat za prodor koji će kontinuirano provjeravati rizike za vašu organizaciju. Drugi način na koji se može pristupiti kako bi se timovima za kibernetičku sigurnost pružio potpuni pregled površine napada i

ranjivosti organizacije jest korištenje pristupa bez agenta za trenutno otkrivanje i provjeru valjanosti.

S ciljem stvaranja djelotvornih podataka o napadima koji poboljšavaju izvedbu sigurnosnih proizvoda i odgovor na incidente, timovi za kibernetičku sigurnost mogu skalirati izvršenje scenarija napada od jedne do više meta probojnih točaka. Primarni fokus trebao bi biti na ranjivostima koje predstavljaju točke kršenja bilo koje kritične organizacijske imovine, [39].

Tablica 2. Prikaz usporedbe između automatiziranog i ručnog penetracijskog testiranja

Automatizirano penetracijsko testiranje	**Ručno penetracijsko testiranje**
Automatizirani proces otkrivanja ranjivosti koji se izvodi pomoću alata za testiranje penetracije.	Pedantna procjena sigurnosne infrastrukture koju provode kompetentni sigurnosni ispitivači.
Brzo se izvodi i štedi mnogo vremena.	Ručni penetracijski testovi mogu trajati danima.
To je laka i učinkovita metoda skeniranja mreža u potrazi za ranjivostima.	Zahtijeva odgovarajuće planiranje i pripremu za provođenje punog ručnog penetracijskog testa.
Ne pruža dublji uvid u ranjivosti.	Pruža detaljan i dublji uvid u ranjivosti.
Otkriva uobičajene sigurnosne propuste poput nedostatka ažuriranja, manjkavih pravila dopuštenja, konfiguracijskih nedostataka, uz nevjerojatnu učinkovitost.	Otkriva akutne nedostatke koje skener često propušta, poput pogrešaka poslovne logike, rupa u zakonu, greške u kodu, itd. Također uključuje iskorištavanje tih ranjivosti za procjenu utjecaja na sustav.
Može se raditi često bez puno priprema i planiranja.	Zahtijeva trud i vrijeme, stoga se ne može raditi često.

Izvor: [39]

6.2. Opis različitih alata za automatsko penetracijsko testiranje

Alati za automatsko testiranje penetracije ne automatiziraju cijeli plan testiranja. Umjesto toga, oni samo pružaju brze programske usluge koje štede mnogo vremena u svakoj istraživačkoj vježbi.

U potrazi za automatiziranim alatima za penetracijsko testiranje ispitivač se mora usredotočiti na vrste alata koji uklanjaju mnoge zadatke koji se ponavljaju. Dostupne su mnoge vrijedne usluge i mnoge od njih su besplatne.

Nakon što ispitivač pregleda tržište automatiziranih sustava za penetracijsko testiranje i potrebno je analizirati alate na temelju sljedećih kriterija:

1. Alat koji uklanja mnogo unosa podataka i ponavljajuće izvođenje naredbi.
2. Specijalizirana usluga koja može brzo napasti jedan aspekt sigurnosti sustava.
3. Uslužni program koji može izvesti i istraživanje i pokrenuti napad bez potrebe za prijenosom podataka.
4. Sustav koji može bilježiti sve aktivnosti za procjenu.
5. Alati koji nude nekoliko strategija napada.
6. Besplatan alat ili besplatna proba ili demonstracija za plaćeni alat.
7. Besplatan alat koji se isplati koristiti ili plaćeni alat koji je vrijedan cijene.

6.2.1. Intruder

Intruder je skener ranjivosti, koji je automatizirana usluga penetracijskog testiranja. Sustav se isporučuje iz oblaka i može se postaviti da radi kontinuirano za razvoj testiranja ili za nadzor sigurnosti operacija. Usluga se također može pokrenuti na zahtjev. Sustav provodi interne i eksterne provjere i ima bateriju od više od 11.000 testova. Glavne značajke:

- Integracija cjevovoda CI/CD
- SaaS paket
- Redovito ažuriran
- Identificira potrebne zakrpe

Platforma Intruder.io pruža tri razine usluge koje se sviđaju širokom rasponu poduzeća i proračuna. Zajednički element za sve planove je vanjski skener ranjivosti. Učestalost tih skeniranja ovisi o planu koji je odabran. Viši planovi također uključuju interno skeniranje. Skeniranje ranjivosti potrebno je često ponavljati jer promjene komponenti u sustavu ili u dodacima predstavljaju dodatne potencijalne sigurnosne ranjivosti, pojedinačno ili u kombinaciji s postojećim sustavima. Nove hakerske strategije pojavljuju se cijelo vrijeme i tako se sustavi koji su bili potvrđeni kao sigurni mogu iznenada otkriti kao ranjivi. Prednost *hostiranog* sustava je u tome što korisnik ne mora stalno ažurirati softver. SaaS paket uključuje sve usluge tima tehničara za održavanje, ostavljajući korisnicima sustava samo korištenje softvera.

Sve tvrtke trebale bi razmotriti korištenje skenera ranjivosti, a planovi Intruder.io čine ga prikladnim za bilo koju veličinu i vrstu poslovanja. Alat ne zahtijeva stručnost za rad jer je ovo automatizirani sustav. Međutim, ad hoc skener može se koristiti kao alat za penetracijsko testiranje.

Prednosti:

- Kontinuirano skeniranje i skeniranje na zahtjev
- Rutina od više od 11.000 provjera ranjivosti
- Radi i za razvojno testiranje i za sigurnosno praćenje operacija
- Popis rezultata označen bojama

Mana:

- Više je skener ranjivosti nego alat za penetracijsko testiranje

Cijene se temelje na broju meta koje je potrebno skenirati i svaka meta dobiva istu razinu pregleda. To znači da je sustav pristupačan i prilagodljiv. Može se iskusiti sustav uz 30-dnevno besplatno probno razdoblje.

6.2.2. Metasploit

Metasploit je dostupan u besplatnoj i plaćenoj verziji. Ovaj paket stvara platformu koja omogućuje istraživanje napada i njihovo pokretanje. Hakeri ga naširoko koriste. Besplatna verzija zove se Metasploit Framework, a verzija koja se plaća Metasploit Pro. Iako je Metasploit projekt otvorenog koda, tvrtka za kibernetičku sigurnost, Rapid7 podržava razvoj sustava, opskrbljujući sredstva i opremu za tim. Zauzvrat, Rapid7 dobiva pravo proizvoditi svoju verziju koja ima više značajki plus profesionalnu podršku. Glavne značajke:

- I automatizirani i ručni
- Istraživanje ranjivosti
- Implementacija napada

Metasploit je jedan od temeljnih alata za penetracijsko testiranje koji se uglavnom uči na većini tečajeva za penetracijsko testiranje. Dvije verzije paketa obraćaju se različitoj publici jer dok je Metasploit Framework besplatan za korištenje, Metasploit Pro je proizvod koji se plaća i skup je. Oba sustava mogu detektirati više od 1500 slabosti za korištenje u napadima. Pro verzija je namijenjena tvrtkama, ali Framework izdanje podložno je ručnim napadima. Iako Pro izdanje više naginje kao skener ranjivosti, ono ima nekoliko vrijednih automatiziranih usluga s kojima bi korisnici verzije Framework mogli raditi, poput automatiziranog brutalnog alata za razbijanje lozinki. Metasploit nudi besplatne i plaćene verzije s dovoljno strategija napada za temeljito testiranje sigurnosti bilo kojeg sustava. Alat omogućuje prikupljanje podataka o sustavu i uočavanje iskorištavanja. Zatim pomaže u trikovima kao što je probijanje lozinke kako bi pomogao da provaljivanje i ponašanje bude što sličnije pravom napadu.

Metasploit Framework nedvojbeno je namijenjen ispitivačima penetracijskog testiranja. Dok ispitivači pokušavaju modelirati način razmišljanja hakera, ispitivači penetracije obično koriste besplatne alate i zbog toga Metasploit Framework čini toliko popularnim. Metasploit Pro paket ima vrlo različitu skupinu korporativnih korisnika. Metasploit Framework se mnogo više koristi od Metasploit Pro.

Prednosti:

- Izbor besplatnih i plaćenih verzija
- Mogućnost potpune stručne podrške od Rapid7
- Alati za istraživanje sustava i prepoznavanje 1500 eksploatacija
- Veze od alata za istraživanje do napadačkih sustava

- Mnogo automatiziranih alata u sustavu

Mana:

- Svako izdanje ima neke dobre alate, ali nijedno nema kompletan set

6.2.3. Wireshark

Wireshark je njuškalo paketa. Omogućuje snimanje prometa i ispitivanje zaglavlja paketa za informacije. Ako se uspiju uhvatiti neki ne kriptirani korisnički podatci, čak je i moguće ukrasti i neke lozinke. Wireshark je bitan dio svakog hakerskog alata i, za razliku od većine hakerskih alata, ima vrlo ugodno GUI sučelje. Instalater će također ostaviti verziju naredbenog retka pod nazivom TShark. Glavne značajke:

- Filtri za snimanje
- Prikaz paketa označen bojama
- Prati razgovore

Wireshark je vodeći analizator paketa. Sustav koristi sustav libpcap/WinPcap za prikupljanje paketa i zatim ih prikazuje u Wireshark konzoli. Hvatanje paketa može vrlo brzo generirati vrlo velike datoteke, ali sustav Wireshark omogućuje primjenu filtera za hvatanje koji će smanjiti taj problem. Mogućnosti Wiresharka proširuju se na bežične mreže, kao što su WiFi i Bluetooth. Ispitivač koristi Wireshark za prikupljanje paketa, a zatim ih sortira i filtrira u pregledniku paketa. Zatim traži informacije koje može koristiti u simuliranim napadima.

Svi penetracijski testeri naučeni su kako koristiti Wireshark. Alat je besplatan za korištenje i nepobjediv u polju analize paketa zbog svog ugrađenog jezika za filtriranje i upita. Taj je jezik vrlo opsežan i dugo se uči, tako da ovo nije alat za povremenu upotrebu.

Prednosti:

- Korisno za istraživanje
- Izvlači promet iz lokalne mreže (Local Area Network – LAN) i bežičnih mreža
- Snimljeni promet može se izvesti u datoteku

Mane:

- Nema integrirane alate za napad

6.2.4. Burp suite

Burp Suite je alat koji se plaća, ali je dostupan i besplatno u obliku izdanja zajednice (engl. *Community Edition*). Plaćena verzija naziva se profesionalno izdanje (engl. *Professional Edition*). Postoji mnogo automatiziranog procesa u Burp Suite izdanju zajednice a još više i u profesionalnom izdanju. Taj plaćeni alat u sebi ima odjeljke za skeniranje ranjivosti, koje se mogu vidjeti u izdanju zajednice, ali se ne

mogu koristiti. Cijeli početni zaslon sučelja Burp Suitea ne radi za one koji ne plaćaju. Glavne značajke:

- Radi na sustavima Windows, macOS i Linux
- ARP trovanje
- Proxy format

Burp Suite je vrlo sličan Metaspolitu samo što se ne koristi toliko često. U besplatnoj verziji ovog alata najviše se radi u alatu za penetracijsko testiranje dok u verziji koja se plaća najviše se koristi ako se radi u konzultantskoj tvrtki za kibernetičku sigurnost. Plaćena verzija više je skener ranjivosti. Sjajna stvar kod sustava Burp Suite je da sučelje povezuje faze istraživanja i napada testova. Te postoji usluga „Presretanje“ koja se koristi za prikupljanje informacija i automatski ih čini dostupnima na zaslonu uljeza za postavljanje napada.

Preporuča se korištenje besplatne verzije izdanje zajednice koju koristi većina penetracijskih testera. Plaćene verzije su vrlo skupe i pružaju usluge skeniranja ranjivosti umjesto alata za penetracijsko testiranje.

Prednosti:

- I besplatna i plaćena verzija koriste isto GUI sučelje
- Istraživački alati unose podatke izravno u sustave napada
- Ima odličan alat za razbijanje lozinki

Mana:

- Plaćeno izdanje moglo bi navesti ispitivača na korištenje skenera ranjivosti

6.2.5. Aircrack-ng

Aircrack-ng prikuplja podatke o bežičnim mrežama i pruža mogućnost hvatanja paketa u prijenosu. Ovaj alat omogućuje ispitivaču da vidi različite kanale koji se trenutno koriste i sve bežične pristupne točke u dometu. Glavne značajke:

- Bežični skener
- Razbijač lozinki
- Može reproducirati promet

Aircrack-ng je bežični mrežni skener. Neobična značajka ovog alata je da može umetnuti promet natrag u bežičnu mrežu, a ne samo uhvatiti pakete. Alat za razbijanje lozinki može raditi na WEP, WPA1 i WPA2 sustavima. Usluga prikazuje naziv svake pristupne točke (Access Point – AP) i predstavlja alate koji mogu pomoći u napadima. Vrsta napada koji se mogu implementirati s ovim alatom naredbenog retka uključuje injekciju smrti, napade čovjek u sredini i napade preopterećenja kapaciteta reprodukcije. Također ima neke sposobnosti za probijanje lozinki. Aircrack-ng je besplatan za korištenje i instalira se na Linux-u.

Preporuča se ako se želi provesti penetracijsko testiranje na bežičnim mrežama. Jedan od najjačih alata na tržištu za to. Sustav nije baš jednostavan za korištenje, ali nakon što se svlada njegov jezik naredbi, alat može biti vrlo moćan. Stručnjaci za bežičnu sigurnost često će koristiti ovaj alat.

Prednosti:

- Poznato je da ga hakeri često koriste
- Korisno za postavljanje lažnih WiFi žarišnih točaka
- Može uvesti napadača u cijelu mrežu putem bežičnog segmenta

Mana:

- Nema grafičkog korisničkog sučelja

6.2.6. Zenmap

Zenmap je grafička verzija Nmapa. Kada se instalira Zenmap, dobit će se i Nmap za korištenje preko grafičkog sučelja ili naredbenog retka. Imati obje mogućnosti nudi veću fleksibilnost. Oba dva će uhvatiti pakete i prikazati njihova zaglavlja i sadržaj. Zenmap zaslon zatim tumači te informacije u cijeli sustav otkrivanja mreže. Glavne značajke:

- Stvara mapu mreže
- Alat za istraživanje
- Na temelju Nmapa

Zenmap i njegovu komponentu naredbenog retka Nmap vrlo često koriste penetracijski testeri i hakeri. Na jednoj razini, ovaj alat može dati stvarnu kartu mreže. Međutim, alat može pružiti mnogo dublje informacije o svakom uređaju na mreži i svakom sučelju na svakom uređaju. Sustav može dokumentirati cijelu mrežu, pružajući korisne informacije o imenima računala i adresama povezanih uređaja. Međutim, tester mora provaliti u krajnju točku na mreži prije nego što se ovaj alat može koristiti. Zenmap i Nmap su korisni za istraživanje, a ne za pokretanje napada. Oba su alata besplatna i oba će raditi na Windows-ima, Linux-u, BSD Unixu i macOS-u.

Korištenje Nmapa uključeno je u svaki tečaj penetracijskog testiranja. Zenmap će se svidjeti povremenim korisnicima jer je malo lakši za korištenje, zahvaljujući GUI-u. Prednosti:

- Brzo dokumentira mrežu
- Predstavlja korisne informacije o svim krajnjim točkama
- Hvata mrežne pakete

Mana:

- Ovaj alat će raditi samo unutar mreže

7. Zaključak

Penetracijski test prodora je ovlašteni simulirani napad koji se izvodi na računalni sustav kako bi se procijenila njegova sigurnost. Penetracijski ispitivači koriste iste alate, tehnike i procese kao i napadači kako bi pronašli i demonstrirali poslovne učinke slabosti u sustavu. Penetracijski testovi obično simuliraju razne napade koji bi mogli ugroziti tvrtku. Oni mogu ispitati je li sustav dovoljno robustan da izdrži napade s ovlaštenih i neovlaštenih pozicija, kao i niz uloga u sustavu. S pravim opsegom, penetracijski test može uroniti u bilo koji aspekt sustava.

Organizacije mogu smanjiti kibernetičke napade implementacijom sustava i strategija kibernetičke sigurnosti. Kibernetička sigurnost je praksa zaštite kritičnih sustava i osjetljivih informacija od digitalnih napada pomoću kombinacije tehnologije, ljudi i procesa.

Organizacijske IT infrastrukture postale su sve raznovrsnije i sofisticiranije. Iako se čini da određena imovina nikada ne dolazi u interakciju s drugima, to ne znači da su u potpunosti odvojena jedna od druge - i dalje su dio infrastrukture u cjelini. A kada je riječ o povredama podataka, pristup bilo gdje može značiti pristup svugdje. Nakon što se postigne početni ulazak, znatno je lakše koristiti tehnike nakon eksploatacije za usmjeravanje na drugo mjesto u infrastrukturi. Uključivanjem penetracijskog testiranja u manje uobičajena okruženja, možete zatvoriti sigurnosne rupe, proaktivno braneći svoje kritične podatke prije nego što dođe do pokušaja proboja.

Primarna razlika između tradicionalnog i penetracijsko testiranja u oblaku je okruženje u kojem se izvode; penetracijsko testiranje u oblaku isto je kao tradicionalno penetracijsko testiranje, ali se izvodi na uslugama u oblaku. Osim toga, okruženja u oblaku dolaze od pružatelja usluga u oblaku, kao što su AWS i GCP. Ovi pružatelji usluga u oblaku imaju stroge smjernice o tome kako bi trebalo provoditi penetracijsko testiranje. Kombinacija sigurnosnih aktivnosti pružatelja usluga oblaka i vlastitog penetracijskog testiranja čine potpuniji sigurnosni stav. U tradicionalnim okruženjima (u prostorijama), samo smo mi odgovorni za obavljanje sigurnosnih aktivnosti.

Kibernetička sigurnost je kontinuirani napor. Automatizirano penetracijsko testiranje pomaže nam da ostanemo oprezni protiv ranjivosti koje prijete funkcionalnosti našeg sustava. Budući da kibernetički kriminalci koriste sve vrste naprednih alata za svoje napade, ne možemo si priuštiti borbu protiv njihovih napada samo ručnim tehnikama kibernetičke sigurnosti. Puno smo sigurniji korištenjem automatiziranih alata jer su učinkovitiji, brži i uglavnom lakši za implementaciju.

Literatura

1. Purplesec. What Are The Different Types Of Penetration Testing?. Preuzeto sa: https://purplesec.us/types-penetration-testing/#WhatIsPenetrationTest [Pristupljeno: srpanj 2023.]
2. Criticalfault. International Journal of Advanced Computer Science and Applications. Preuzeto sa: https://criticalfault.com/blog/what-is-penetration-testing-why-is-it-important/ [Pristupljeno: srpanj 2023.]
3. Mohd. Ehmer Khan, Dr. Kohei Arai. (ur.) A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. The Science and Information (SAI) Organization. 6.izdanje; 2012. pp. 12-15. Preuzeto sa: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ab0ed151c010e03e2d34284ae5f89756372e7690#page=22 [Pristupljeno: srpanj 2023.]
4. Purplesec. White Box Penetration Testing: When Do You Need One?. Preuzeto sa: https://purplesec.us/learn/white-box-penetration-testing/ [Pristupljeno: srpanj 2023.]
5. Javatpoint. Advantages and Disadvantages of White Box Testing. Preuzeto sa: https://www.javatpoint.com/advantages-and-disadvantages-of-white-box-testing [Pristupljeno: srpanj 2023.]
6. Purplesec. Black Box Penetration Testing: When Do You Need One?. Preuzeto sa: https://purplesec.us/learn/black-box-penetration-testing/ [Pristupljeno: srpanj 2023.]
7. Javatpoint. Advantages and Disadvantages of Black-box Testing. Preuzeto sa: https://www.javatpoint.com/advantages-and-dsadvantages-of-black-box-testing [Pristupljeno: srpanj 2023.]
8. Artoftesting. Grey Box Testing. Preuzeto sa: https://artoftesting.com/grey-box-testing [Pristupljeno: srpanj 2023.]
9. Synopsys. Network Penetration Testing. Preuzeto sa: https://www.synopsys.com/glossary/what-is-network-penetration-testing.html [Pristupljeno: srpanj 2023.]
10. Synopsys. Web Application Penetration Testing. Preuzeto sa: https://www.synopsys.com/glossary/what-is-web-application-penetration-testing.html [Pristupljeno: srpanj 2023.]
11. Purplesec. How To Perform A Wireless Penetration Test. Preuzeto sa: https://purplesec.us/perform-wireless-penetration-test/ [Pristupljeno: srpanj 2023.]
12. Purplesec. Social Engineering Penetration Testing: Attacks, Methods, & Steps. Preuzeto sa: https://purplesec.us/social-engineering-penetration-testing/#Attacks [Pristupljeno: srpanj 2023.]
13. Triaxiomsecurity. What Is A Physical Penetration Test?. Preuzeto sa: https://www.triaxiomsecurity.com/what-is-a-physical-penetration-test/ [Pristupljeno: srpanj 2023.]
14. IBM. What are the common types of cyberattacks?. Preuzeto sa: https://www.ibm.com/topics/cyber-attack [Pristupljeno: srpanj 2023.]
15. Malwarebytes. What is malware?. Preuzeto sa: https://www.malwarebytes.com/malware [Pristupljeno: kolovoz 2023.]
16. MDPI. Social Engineering Attacks: A Survey. Preuzeto sa: https://www.mdpi.com/1999-5903/11/4/89 [Pristupljeno: kolovoz 2023.]
17. Imperva. Social Engineering. Preuzeto sa: https://www.imperva.com/learn/application-security/social-engineering-attack/ [Pristupljeno: kolovoz 2023.]
18. European Union Agency for Cybersecurity. ENISA threat landscape 2022. Preuzeto sa: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022 [Pristupljeno: kolovoz 2023.]
19. Cloudflare. What is a DDoS attack?. Preuzeto sa: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/ [Pristupljeno: kolovoz 2023.]

20. Imperva. Man in the middle (MITM) attack. Preuzeto sa: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/ [Pristupljeno: kolovoz 2023.]
20. Imperva. Man in the middle (MITM) attack. Preuzeto sa: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/ [Pristupljeno: kolovoz 2023.]
21. Cloudflare. What is cross-site scripting?. Preuzeto sa: https://www.cloudflare.com/learning/security/threats/cross-site-scripting/ [Pristupljeno: kolovoz 2023.]
22. Portswigger. SQL injection. Preuzeto sa: https://portswigger.net/web-security/sql-injection#what-is-sql-injection-sqli [Pristupljeno: kolovoz 2023.]
23. Getastra. Learn Why Penetration Testing is Important. Preuzeto sa: https://www.getastra.com/blog/security-audit/why-penetration-testing-is-important/ [Pristupljeno: kolovoz 2023.]
24. Esecurityplanet. Penetration Testing Phases & Steps Explained. Preuzeto sa: https://www.esecurityplanet.com/networks/penetration-testing-phases/ [Pristupljeno: kolovoz 2023.]
25. Linkedin. Top 5 Methodologies for Penetration Testing. Preuzeto sa: https://www.linkedin.com/pulse/top-5-methodologies-penetration-testing-dan-duran [Pristupljeno: kolovoz 2023.]
26. Marco Prandini, Marco Ramilli. Towards a practical and effective security testing methodology. Riccione, Italy: IEEE; 2010 Preuzeto sa: https://ieeexplore.ieee.org/document/5546813 [Pristupljeno: kolovoz 2023.]
27. Synopsys. OWASP Top 10 2021. Preuzeto sa: https://www.synopsys.com/glossary/what-is-owasp-top-10.html#J [Pristupljeno: kolovoz 2023.]
28. NIST. NIST General Information. Preuzeto sa: https://www.nist.gov/director/pao/nist-general-information [Pristupljeno: kolovoz 2023.]
29. Pentest standard. High Level Organization of the Standard. Preuzeto sa: http://www.pentest-standard.org/index.php/Main_Page [Pristupljeno: kolovoz 2023.]
30. Open Information Systems Security Group. ISAF. Preuzeto sa: http://cs.uccs.edu/~cs591/penetrateTest/issaf0.1.pdf [Pristupljeno: kolovoz 2023.]
31. Pecb. Penetration Testing on Organizations. Preuzeto sa: https://pecb.com/article/penetration-testing-on-organizations [Pristupljeno: kolovoz 2023.]
32. Stripeolt. Why penetration testing is essential for the financial services industry. Preuzeto sa: https://stripeolt.com/insights/cyber-security/penetration-testing-is-essential-for-the-financial-services-industry/ [Pristupljeno: kolovoz 2023.]
33. Chiefhealthcareexecutive. Pen Testing 101: How Healthcare Organizations Can Strengthen Their Security Posture. Preuzeto sa: https://www.chiefhealthcareexecutive.com/view/pen-testing-101-how-healthcare-organizations-can-s trengthen-their-security-posture [Pristupljeno: kolovoz 2023.]
34. Kroll. Penetration Testing Industrial Control Systems: What to Know. Preuzeto sa: https://www.kroll.com/en/insights/publications/cyber/pentesting-industrial-control-systems [Pristupljeno: kolovoz 2023.]
35. Ilke Yurtseven, Selami Bagriyanik. A Review of Penetration Testing and Vulnerability Assessment in Cloud Environment. İstanbul, Turkey: IEEE; 2020 Preuzeto sa: https://ieeexplore.ieee.org/document/5546813 [Pristupljeno: kolovoz 2023.]

36. Astra. Cloud Penetration Testing: A Complete Guide. Preuzeto sa: https://www.getastra.com/blog/security-audit/cloud-penetration-testing/ [Pristupljeno: kolovoz 2023.]
37. IBM. What are Iaas, Paas and Saas?. Preuzeto sa: https://www.ibm.com/topics/iaas-paas-saas [Pristupljeno: kolovoz 2023.]
38. Bluexp. Cloud Security Architecture for IaaS, PaaS and SaaS. Preuzeto sa: https://bluexp.netapp.com/blog/blg-cloud-security-architecture-for-iaas-paas-and-saas [Pristupljeno: kolovoz 2023.]
39. Purplesec. Why Automation Is The Future Of Penetration Testing. Preuzeto sa: https://purplesec.us/learn/automating-penetration-testing/ [Pristupljeno: kolovoz 2023.]
40. Comparitech. The Best Automated Penetration Testing Tools. Preuzeto sa: https://www.comparitech.com/net-admin/best-automated-penetration-testing-tools/ [Pristupljeno: kolovoz 2023.]
41. Cvitić, I., Peraković, D., Periša, M. et al. Defining Cross-Site Scripting Attack Resilience Guidelines Based on BeEF Framework Simulation. Mobile Netw Appl (2022). Preuzeto sa: https://doi.org/10.1007/s11036-022-02052-z [Pristupljeno: kolovoz 2023.]
42. I. Cvitić, D. Peraković, M. Periša and M. Musa, "Network parameters applicable in detection of infrastructure level DDoS attacks," 2017 25th Telecommunication Forum (TELFOR), Belgrade, Serbia, 2017, pp. 1-4, https://doi.org/10.1109/TELFOR.2017.8249347. [Pristupljeno: kolovoz 2023.]
43. Yadav, U.S., Gupta, B.B., Peraković, D., Peñalvo, F.J.G., Cvitić, I. (2022). Security and Privacy of Cloud-Based Online Online Social Media: A Survey. In: Knapcikova, L., Peraković, D., Perisa, M., Balog, M. (eds) Sustainable Management of Manufacturing Systems in Industry 4.0. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-90462-3_14 [Pristupljeno: kolovoz 2023.]

Popis slika

Popis kratica

AP	Access Point
APT	Advanced Package Tool
ARP	Address Resolution Protocol
AST	Application Security Testing
AWS	Amazon Web Service
BAS	Breach And Attack Simulation
BSOD	Blue Screen Of Death
CORS	Cross-Origin Resource Sharing
CASB	Cloud Access Security Broker
CI/CD	Continuous Integration / Continuous Delivery
CSP	Cloud Service Provider
CSRF	Cross-Site Request Forgery
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
DCS	Distributed Control System
DDoS	Distributed Denial-of-Service Attack
DNS	Domain Name System
FISMA	Federal Information Security Management Act
GCP	Google Cloud Platform
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
HTML	HyperText Markup Language
IAST	Interactive Application Security Testing
ICS	Industrial control systems
IoT	Internet Of Things
IIoT	Industrial Internet of Things
IT	Information technology
ISSAF	Information System Security Assessment Framework
ISP	Internet Service Provider
LAN	Local Area Network

MAC	Media Access Control
MITM	Man-In-The-Middle
NIST	National Institute of Standards and Technology
NPB	Network Packet Broker
NUC	Next Unit of Computing
NVD	National Vulnerability Database
OAST	Out-of-Band Application Security Testing
OSS	Open-source software
OSSTMM	Open Source Security Testing Methodology Manual
OWASP	Open Worldwide Application Security Project
PAYG	Pay As You Go
PCI-DSS	Payment Card Industry Data Security Standard
PII	Personal Identifying Information
PTES	Penetration Testing Execution Standard
QA	Quality Assurance
RFID	Radio-frequency identification
SaaS	Software as a service
SANS	SysAdmin, Audit, Network, Security
SAST	Static application security testing
SCA	Software composition analysis
SCADA	Supervisory Control and Data Acquisition
SDN	Software Define Networking
SDLC	Software Development Life Cycle
SLA	Service Level Agreement
SSL	Secure Sockets Layer
Sql	Structured query language
SQLi	SQL injekcija
SSRF	Server-Side Request Forgery
Usb	Universal Serial Bus
UML	Unified Modeling language
URL	Uniform Resource Locator

WAF	Web Application Firewall
WPA	Wi-Fi Protected Access
XSS	Cross-Site Scripting

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je _____diplomski rad_____
(*vrsta rada*)
isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada pod naslovom Analiza i izvedba penetracijskog testiranja za povećanje sigurnosti informacijsko-komunikacijskog sustava, u Nacionalni repozitorij završnih i diplomskih radova ZIR.

Student/ica:

U Zagrebu, ____7.9.2023._____ __Nikola Knőgel,_____
(ime i prezime, *potpis*)