VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Bakalauro darbas

Saugus pažeidžiamumų skaitytuvas sistemų auditavimui

Atliko:
Jonas Gavėnavičius parašas

Vadovas:
Lektorius Virgilijus Krinickij

Vilnius
2019

Turinys

Sutartinis terminų žodynas

FTP - *File Transfer protocol*, protokolas leidžiantis .

Buffer overflow - tai viena iš potencialių rizikų, kai dėl per didelio kiekio duomenų, informacija yra perrašoma gretimuose atminties blokuose.

SSH - *Secure Shell*, tai tinklo protokolas kuris leidžia vartotojui saugiai pasiekti sistemą per nesaugu tinklą.

Santrauka

Šiais laikais kai pasaulis tampa vis labiau ir labiau skaitmenizuotas, iškyla pati opiausia problema, tai yra kibernetine sauga. Vis daugiau pavyždžių matome kaip įsibrauna į sistemas kurias galima potencialiai išnaudoti dėl finansinių ar kitų priežasčiu. Taip pat dėl to nukenčia ir tų sistemų vartotojai - finansiškai ar morališkai, jų privati informacija būna pavogiama ir paviešinama. Niekas nėra saugus nuo tokių situacijų. Todėl valstybės, įmonės ar korporacijos vis daugiau investuoja į kibernetinę apsaugą, kibernetinė sauga tampa vis dažnesnė diskusija visuomenėja, vis daugiau dėmėsio ir resursų skiriama butent jai, stengemasi užkirsti kelią minėtoms situacijoms. Kuriami įrankiai kurie analizuoja sistemas ir randa jų spragas, pritaikomos įvairios technologijos ar metodai kurie perspėja apie galima ar vykstančią ataką prieš sistemą, stengiamasi rasti sistemoje spragas ir užlopyti jas kol jų nerado asmenys kurie išnaudotų jas.

Summary

Darbo pavadinimas kita kalba

This is a summary in English...

Įvadas

Šiame darbe kuriamas saugus įrankis, kuris neišduotų savo serverio vietos ir vartotojai pasinaudoję juo galėtų gauti išsamią informaciją apie jų internetinėje svetainėje egzistuojančias spragas bei pažeidžiamumus ar modifikuotus failus. Darbo tikslas – sukurti saugų pažeidžiamumų skenavimo įrankį, kuris pasileistų iš tam tikros VPN technologija slepiamos vietos, skenuotų internetinę svetainę bei jos failus, aptiktų pažeidžiamumus ar modifikuotus failus, ir pateiktų klientui informaciją apie skenavimo rezultatus. Siekiant, kad darbas būtų paklausus ir veiksmingas, keliami šie darbo uždaviniai:

1. Apžvelgti egzistuojančius panašius įrankius;
2. Išanalizuoti dažniausiai pasitaikančius internetinių svetainių pažeidžiamumus;
3. Išanalizuoti dažniausiai pasitaikančių pažeidžiamumų egzistuojančius atviro kodo įrankius;
4. Išanalizuoti dažniausiai pasitaikančius internetinių svetainių skenavimo metodus.

Kibernetinis saugumas yra svarbi kiekvienos sistemos dalis. Kadangi kiekvieną sistemą kuria žmogus, ir į kiekvieną sistemą įeina žmogiškasis faktorius, dėl kurio sistemos beveik visada turi pažeidžiamumų, kuriuos gali išnaudoti puolėjas siekiantis tam tikros naudos. Dėl šios priežasties toks įrankis būtų ypač naudingas bet kokiai sistemai. Aptikus pažeidžiamumą, galima įtarti, kad tą pažeidžiamumą gali aptikti ir nuostolio siekiantys asmenys, arba jis jau buvo aptiktas, ir tam tikri failai buvo modifikuoti įdedant tam tikrą kodą, kuris leistų atakuojančiui asmeniui patekti į sistemą nepastebėtam. Dėl šių priežasčių saugus ir automatizuotas sistemų pažeidžiamumo skenavimo įrankis yra ypač aktualus. Tačiau šio įrankio kūrimą apsunkina šios priežastys:

- Įrankio kūrimas reikalauja didelio bagažo žinių;
- Daugumos sistemų, kurios turi pažeidžiamas vietas, išeities kodas nėra atviras, todėl kai kurios spragos negali būti aptiktos automatizuotu įrankiu. Taip pat negalima atlikti kai kurių sistemos kodo analizių, kas taip pat sumažina tokio įrankio efektyvumą;
- Sistemoje gali būti tiek daug skirtingų potencialių pažeidžiamų, jog jų aptikimo automatizavimas tampa problematiškas.

Šias problemas siūloma spręsti apskaičiuojant tam tikrą įrankio veiksmingumą procentaliai. Didžiausias tokio sprendimo privalumas yra tas, kad vartotojas supras, jog tokio įrankio rezultatų analizavimas ir pritaikymas neužtikrina visų pažeidžiamumų aptikimo ir kibernetinio saugumo užtikrinimo. Šio įrankio kūrimo metu, siekiant sukurti saugų pažeidžiamumų aptikimo įrankį, siekiama pagerinti sistemos saugumą, bet neužtikrinti jo. Įrankis bus skirtas tik internetinėms svetainėms ir nebus stengiamasi jo pritaikyti ir kitoms sistemoms.

1. Susijusių darbų analizė

1.1. Internetinių svetainių spragų skaitytuvai

1.1.1. Pentest-Tools skaitytuvas

Pateikiamas trečio lygio poskyrio tekstas.

2. Sistemų auditavimas

2.1. Elgsenos analizė

2.1.1. Example

Pateikiamas trečio lygio poskyrio tekstas.

2.2. Statinė kodo analizė

2.2.1. Panaudojimas

Statinė analizė suteikia galimybę gauti informacijos apie galimą programos elgesį programos vykdymo metu, nevykdant programos. Statinės analizės metodai tiria programos elgesį su visomis įmanomomis įvestimis ir visomis įmanomomis programos būsenomis, kurias programa gali pasiekti.

Atlikus teisingai statinę analizę, galima aptikti akivaizdžias klaidas kurių programuotojas galėjo nepastebėti, tai sutaupo laiko bei sumažina spragų kiekį, taip pat galimai aptinkami nenumatyti scenarijai. Kaikurios programavimo aplinkos (Visual Studio, IntelliJ...) atlieka pastovią statine analizę tam, kad programuotojai pamatytų potencialias klaidas prieš sistemos startą.

Statinė analizė padeda aptikti:

- Neicijuotus kintamuosius;
- Potencialias klaidas sistemos išeities kode;
- Buffer overflow spragas;

2.2.2. Panaudojimas darbe

Vartotojui davus FTP serverio adresą ir iš jo atsisiuntus sistemos išeities kodą bus atliekama statinė analizė ir taip bandoma aptikti scenarijus, kuriuose sistemą būtų galima sugadinti, ar į ją isilaužti.

2.3. Dinaminė kodo analizė

2.3.1. Panaudojimas

Dinaminė analizė vykdoma kai programa jau yra vykdomo stadijoje. Dinaminės analizės paskirtis yra aptikti atminties nutekėjimus, null rodykles. Dynaminės analizės metu bandoma igyvendinti visus įmanomus scenarijus ir išbandyti visas imanomas variacijas į programos ivestį.

Veikimo metu programa gali neatlaisvinti atminties atgal į operacinę sistemą, to pasekoje serveris kuriame programa veikia, pritruks atminties ir pradės veikti lečiau kol galiausiai sustos. Nuo to padėtų apsaugoti dinaminė analizė, atlikus ją teisingai, galima aptikti didžiają dalį spragų kurios potencialiai labiausiai įtakos sistemą. Jas ištaisius, sistema veikimo stabilumas padidėja, nenumatytų atvejų skaičius pamažėja.

Dinaminė analizė padeda aptikti:

- Atminties nutekėjimus;
- Netikėtus scenarijus;

- Opiausias spragas;

2.3.2. Panaudojimas darbe

Analizės metu, bus bandoma į visas sistemos įvestis pateikti nenumatytu reikšmių taip išbandant kuo daugiau nenumatytų scenarijų.

2.4. Išorinių spragų skenavimas

2.4.1. Panaudojimas

Išorinis pažeidžiamumų skenavimas atliekamas iš sistemos tinklo išorės, o pagrindinis jo tikslas yra aptikti perimetro gynybos spragas, pvz., atvirus atvirus tinklo užkardos prievadus ar specializuotą žiniatinklio programų užkardą. Išorinis pažeidžiamumų skenavimas gali padėti organizacijoms išspręsti saugumo problemas, kurios įsilaužėliams galėtų suteikti prieigą prie organizacijos tinklo.
Išorinis pažeidžiamumų skenavimas aptiks:

- Didžiausios tiesioginės grėsmės sistemoje;
- Programinę įrangą kuriai reikia atnaujinimų bei priežiuros;
- Atidaryti prievadus ir protokolus - įėjimo taškus į sistemos tinklą;

2.4.2. Panaudojimas darbe

Testavimo įrankyje bus implementuota prievadų skanavimo funkcija, taip pat bus bandoma išgauti kuo daugiau informacijos apie pačią sistemą.

2.5. Vidinių pažeidžiamumų skenavimas

2.5.1. Panaudojimas

Vidinis pažeidžiamumo patikrinimas atliekamas iš organizacijos perimetro gynybos. Jos tikslas yra aptikti pažeidžiamumus, kuriuos galėtų išnaudoti įsilaužėliai arba nepatenkinti darbuotojai, sėkmingai įsiskverbiantys į perimetro gynybą, arba turintys teisėtą prieigą prie organizacijos tinklo.
Vidinių pažeidžiamumų skenavimas aptiks:

- Sistemos komponentus kurie galimai gali sukelti gresmę;
- Pasenusi programinė įranga, kuriai reikia atnaujinimų;
- ...;

2.5.2. Panaudojimas darbe

Įrankio vartotojas savo noru gali pateikti prisijungimus prie sistemos, taip pat SSH adresą. Pasinaudojus šia informaciją įrankis prisijungs prie sistemos, ir bandys išgauti kuo daugiau informacijos, bei aptikti joje spragas.

2.6. Aplinkos spragų skenavimas

2.6.1. Example

Pateikiamas trečio lygio poskyrio tekstas.

3. Gerosios praktikos

3.1. Atviros kodo programinė įranga

3.1.1. Privalumai

Pateikiamas trečio lygio poskyrio tekstas.

3.1.2. Trūkumai

Pateikiamas trečio lygio poskyrio tekstas.

3.1.3. Apibendrinimas

Pateikiamas trečio lygio poskyrio tekstas.

3.2. Uždaro kodo programinė įranga

3.2.1. Privalumai

Pateikiamas trečio lygio poskyrio tekstas.

3.2.2. Trūkumai

Pateikiamas trečio lygio poskyrio tekstas.

3.2.3. Apibendrinimas

Pateikiamas trečio lygio poskyrio tekstas.

4. Sistemų auditavimo įrankis

4.1. Įrankio aprašas

Pateikiamas 4.5 poskyrio tekstas. Vienas iš šaltinių [?]. Visas turinys priklauso 4 skyriui.

4.2. Aptiktini pažeidžiamumai

Pateikiamas 4.5 poskyrio tekstas. Vienas iš šaltinių [?]. Visas turinys priklauso 4 skyriui.

4.3. Tikslumas

Pateikiamas 4.5 poskyrio tekstas. Vienas iš šaltinių [?]. Visas turinys priklauso 4 skyriui.

4.4. Įgyvendinti lukeščiai

Pateikiamas 4.5 poskyrio tekstas. Vienas iš šaltinių [?]. Visas turinys priklauso 4 skyriui.

4.5. Trūkumai

Pateikiamas 4.5 poskyrio tekstas. Vienas iš šaltinių [?]. Visas turinys priklauso 4 skyriui.

Išvados ir rekomendacijos

Išvados bei rekomendacijos.

Ateities tyrimų planas

Pristatomi ateities darbai ir/ar jų planas, gairės tolimesniems darbams....

Literatūros šaltiniai

[1] G Balakrishnan, T Reps, D Melski, and T Teitelbaum. *WYSINWYX: What You See Is Not What You eXecute*, pages 202–213. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[2] Thoms Ball. The Concept of Dynamic Analysis. *SIGSOFT Softw. Eng. Notes*, 24(6):216–234, oct 1999.

Priedai

Dokumentą sudaro du priedai: A priede ....

A. Pirmojo priedo pavadinimas

Pirmojo priedo tekstas ...

B. Antrojo priedo pavadinimas

Antrojo priedo tekstas ...