ИТ РЕВИЗИЈА АПЛИКАЦИЈА

АПЛИКАЦИЈА

- Рачунарски софтвер су у основи програми и процедуре намењени извршавању одређених задатака на систему.
- Системи рачунарског софтвера класификују се у три главне врсте и то системски софтвер, софтвер за програмирање и апликативни софтвер.
- **Апликативни софтвер (скраћено апликација) је програм или група програма намењених крајњим корисницима.**
- Апликативни софтвер или треба да буде инсталиран или може да се покрене на мрежи.

ТИПОВИ АПЛИКАЦИЈА

- Постоји неколико различитих врста апликација које су доступне за преузимање како предузећима тако и појединцима.
- Најчешће апликације које се данас користе:

- **Апликација заснована на вебу**

Апликација заснована на вебу је апликација која да би се користила захтева приступ Интернету. Ове врсте апликација су кодиране у JavaScript, HTML5 или CSS. Апликација заснована на web-у (веб) обично захтева много мање количине меморијског простора на корисничком уређају, јер се база података чува на Интернет серверу. Примери апликација заснованих на вебу укључују *Netflix*, *Google* документе и *Dropbox*.

- **Изворна апликација**

Апликације које су креиране за одређену мобилну платформу познате су као изворне или матичне апликације. На пример, апликација која је направљена за *Apple iPhone* биће употребљива само на Apple уређајима и неће радити на свим врстама мобилних телефона, као што је Android. Ове врсте апликација се првенствено користе за пружање највећих перформанси на одређеном мобилном оперативном систему. Пример изворне апликације је апликација калкулатор на iPhone-у.

- **Хибридна апликација**

Хибридна апликација је апликација која је направљена да подржава и матичне и веб технологије и комбинација је веб апликација и изворних апликација. Ове апликације су лакше и брже креирају и користе само једну базу кода која се може интегрисати на различите платформе. Међутим, важно је напоменути да хибридне апликације често имају нижу стопу перформанси изворних апликација или апликација на вебу.

ТИПОВИ АПЛИКАЦИЈА

- Апликациони софтвер се на основу употребе може поделити на:

- Услужни програми
- Генерички програми
- Интегрисани програми
- Специфични софтвер
- Софтвер по мери
- Софтвер за обраду текста
- Софтвер за стоно издаваштво
- Софтвер за прорачунске табеле
- Софтвер за базе података
- Презентацијски софтвер
- Интернет прегледачи
- Програми е-поште
- Графички програми (на основу пиксела)
- Графички програми (засновани на векторима)
- Комуникациони софтвер: Комуникација путем звука, видеа или путем ћаскања

ДЕСКТОП АПЛИКАЦИЈЕ

- Десктоп апликација представља сваку апликацију која се може инсталирати на један рачунар како би се помоћу ње обављали специфични задаци
- Једноставан пример су медиа плејери или MS Office Word, Excel, PowerPoint (пре Office365)
- Неке десктоп апликације се могу користити од стране више корисника у мрежном окружењу
- Карактеристике везане за десктоп апликације:
  - ✓ Морају се инсталирати на сваки рачунар
  - ✓ У већини случајева везани су за локацију где се налази рачунар на који су инсталирани и ту се могу користити
  - ✓ Већа контрола над безбедносним аспектима апликација услед мање изложености приступу са интернета
  - ✓ Нису везане за интернет конекцију или њену брзину да би се користиле
  - ✓ Након куповине десктоп апликације количина неопходног одржавања је мала до умерена у просеку (у поређењу са Web апликацијама)

ДЕСКТОП АПЛИКАЦИЈЕ - КАТЕГОРИЈЕ

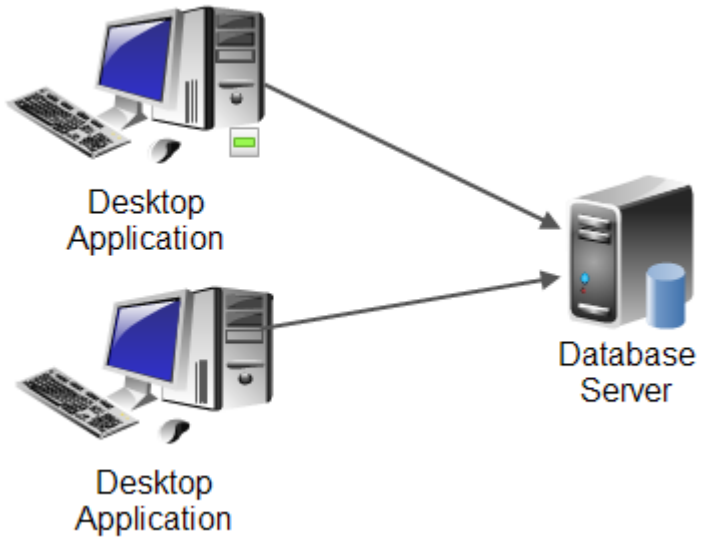
- **“*Standalone Business*” апликације:**

  Ова врста апликација су самосталне пословне апликације попут *Excel*, *Word*, *Outlook* и других апликација.

  Категорија укључује апликације које чине рад људи ефикас

- **“*Client-Server*” апликације:**

  Апликације које раде на локалном рачунару на којем неопходна је инсталација софтверског клијента, али приступ информацијама је на удаљеном серверу.



ОПЕРАТИВНИ СИСТЕМИ И ИНТЕГРИСАНА РЕШЕЊА СУ КАТЕГОРИЈА ЗА СЕБЕ

ДЕСКТОП АПЛИКАЦИЈЕ - КАТЕГОРИЈЕ

- **“*Utilities u Plug-ins за системе*”:**

  Услужни програми и додаци (plug-ins) који побољшавају ефикасност корисничког рада који спада у ову категорију.

- **“*Collaborative*” апликације:**

  Ове апликације служе обављању заједничких задатака за кориснике. Корисници могу да комуницирају преко ових апликација. Последњих година on-line колаборативне апликације преузимају примат над десктоп верзијама.

- **“*Multimedia*” апликације**:

  Ово су апликације које служе за репродукцију филмова, видео записа, звучних датотека и других датотека из корисниковог система.

РЕВИЗИЈА ДЕСКТОП АПЛИКАЦИЈА

- Специфичности десктоп апликација
- Преглед архитектуре десктоп апликација
- Преглед дефинисаних улога и матрице приступа
- Преглед корисника и администратора система
- Безбедносна подешавања и конфигурације
- Аутентификација
- Управљање сесијом
- Руковање улазним и излазним подацима и њихова заштита
- Преглед процеса ажурирања апликација
- Врсте напада и претњи

ИЗВОРИ ИНФОРМАЦИЈА

- Интерни извори
  - Ревизорски тим, претходни извештаји
  - Систем администратори
  - Администратори апликације
  - Процена критичности апликација
  - ИТ безбедност
  - Произвођачка документација о апликацији
  - Матрица приступа
  - Списак активних корисника апликације
  - Одобрена верзија апликације
  - Захтеви корисника за интервенцијом (тикети који се односе на апликацију, предмет ревизије)
  - Мрежни дијаграми
- Екстерни извори
  - Објаве произвођача апликације
  - Најбоља пракса
  - Сајтови са информацијама о познатим рањивостима апликација
  - Кориснички форуми у вези апликације, предмет ревизије
  - Извештаји екстерне ревизије

РАЊИВОСТИ ДЕСКТОП АПЛИКАЦИЈА

"*All time*" Листа десктоп апликација са потврђеним безбедносним рањивостима

Извор: https://www.cvedetails.com/

- Као што се може видети воде претраживачи и оперативни системи
- Flash и JAVA решења
- Овде треба потражити све апликације, од пословних до услужних (разни архивери, преводиоци и сл.)

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Debian Linux	Debian	OS	3067
2	Android	Google	OS	2563
3	Linux Kernel	Linux	OS	2357
4	Mac Os X	Apple	OS	2212
5	Ubuntu Linux	Canonical	OS	2007
6	Firefox	Mozilla	Application	1873
7	Chrome	Google	Application	1858
8	Iphone Os	Apple	OS	1655
9	Windows Server 2008	Microsoft	OS	1421
10	Windows 7	Microsoft	OS	1283
11	Acrobat Reader Dc	Adobe	Application	1182
12	Acrobat Dc	Adobe	Application	1182
13	Windows 10	Microsoft	OS	1111
14	Flash Player	Adobe	Application	1078
15	Windows Server 2012	Microsoft	OS	1050
16	Enterprise Linux Desktop	Redhat	OS	1039
17	Internet Explorer	Microsoft	Application	1030
18	Safari	Apple	Application	1029
19	Enterprise Linux Server	Redhat	OS	979
20	Windows 8.1	Microsoft	OS	978
21	Acrobat	Adobe	Application	949
22	Enterprise Linux Workstation	Redhat	OS	941
23	Thunderbird	Mozilla	Application	921

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА 1

- Лош апликативни модел – неодобрен или лоше анализиран пре пуштања у рад
- Апликативни систем за аутентикацију није ажуран
- Кориснички приступ није правилно дефинисан (принцип најмање потребне привилегије)
  - неопходно је имати у виду да се приступ апликацијама сада најчешће дефинише централизовано преко активног директоријума или другог система за управљање корисничким налозима
- *Default*-ни налози нису искључени
  - (Најбоља пракса налаже да се сви налози који се користе за успостављање апликације или иницијално пуштање у рад, а обезбеђени су од стране вендора измене)
- Иницијалне лозинке за приступ нису измењене
  - Апликација треба да тражи од корисника да промени лозинку након првог логовања
- Матрица приступа са дефинисаним ролама и привилегијама није добро дефинисана
- Редовна провера додељених права приступа се не врши

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА – МАТРИЦА ПРИСТУПА

Матрица приступа са ролама и привилегија ма мора бити добро дефинисана како би:

- се остварила правилна сегрегација дуж ности,
- осигурало да је принцип 2 пара очију (“4 eyes”) добро примењен,
- избегло преплитање привилегија, и
- испоштовао принцип најмањег потребн ог приступа.

Permission Groups - Security Roles	Application Administrator	Application Author	Application Deployment Manager	Asset Manager	Company Resource Access Manager <R2>	Compliance Setting Manager	Endpoint Protection Manager	Full Administrator	Infrastructure Administrator	Operating System Deployment Manager	Operations Administrator	Read-only Analyst	Remote Tools Operator	Security Administrator	Software Update Manager	Permission Groups - Permissions
Alert Subscription						X	X			X	X					Alert Subscription
Alerts	X	X	X		X	X	X	X	X	X	X			X		Alerts
Antimalware Policy							X	X			X	X				Antimalware Policy
Application	X	X	X					X		X	X	X				Application
Boot Image Package								X		X	X	X				Boot Image Package
Boundaries	X	X	X					X	X	X	X	X			X	Boundaries
Boundary Group	X	X	X					X	X	X	X	X			X	Boundary Group
Certificate Profile <R2>					X			X			X	X				Certificate Profile <R2>
Client Agent Setting	X		X	X				X	X		X	X			X	Client Agent Setting
Cloud Subscription								X	X		X	X				Cloud Subscription
Collection	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Collection
Communications Provisioning Profile <R2>					X			X			X	X				Communications Provisioning Profile <R2>
Computer Association								X		X	X	X				Computer Association
Configuration Item						X		X			X	X				Configuration Item
Configuration Policy					X	X	X	X			X	X				Configuration Policy
Deployment Templates	X		X					X			X	X			X	Deployment Templates
Device Drivers								X		X	X	X				Device Drivers
Distribution Point	X	X	X					X	X	X	X	X			X	Distribution Point
Distribution Point Group	X	X	X					X	X	X	X	X			X	Distribution Point Group
Driver Package								X		X	X	X				Driver Package
Firewall Settings							X	X			X	X				Firewall Settings
Global Condition	X	X	X					X			X	X				Global Condition
Inventory Reports				X				X	X		X	X				Inventory Reports
Migration Job								X	X		X	X				Migration Job
Migration Site-to-Site Mappings								X	X		X	X				Migration Site-to-Site Mappings
Mobile Device Enrollment Profiles	X		X	X				X	X		X	X				Mobile Device Enrollment Profiles
Operating System Image								X		X	X	X				Operating System Image
Operating System Installation Package								X		X	X	X				Operating System Installation Package

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА 2

- Када говоримо о приступу апликацији преко централизованих решења као што је активни директоријум (АД), најчешћи разлози за компромитовање апликације могу бити услед небезбедног начина управљања корисничким налозима:
- Компромитован налог = Компромитована апликација
- Небезбедно или лако за погодити давање корисничких имена
  - (корисничка имена треба да буду јединствена и додељена на начин да се не могу погодити од стране трећих лица)
- Администраторски налог није преименован
  - (налоге Admin, Administrator итд. потребно преименовати укључујући и локални администраторски налог на рачунарима)
- Коришћење исте иницијалне лозинке
  - (иницијалне лозинке морају бити насумичне, најбоља пракса је да иницијалну шифру генерише апликација и пошаље кориснику на *e-mail*)
- Не постоји обавеза измене лозинке током првог логовања
- Неадекватна дужина и комплексност лозинке
  - (нпр. Минимум осам карактера за стандардне кориснике, 16 карактера за администраторске налоге)
- Лоше управљање AD групама, чланство корисника у непотребним групама без постојећег мониторинга или одобрења

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА 3

- Присуство активних налога који припадају бившим запосленима
- Уколико се користи Call центар – могућност социјалног инжењеринга и представљања као друго лице
  - (осмислити начин да се са сигурношћу утврди идентитет позиваоца)
- Више истовремених сесија са истим корисничким налогом је дозвољено
  - (Пракса налаже супротно)
- Време истека сесије није дефинисано
  - (Уобичајена временска ограничења сесија крећу се између 2-5 минута за апликације са високим ризиком и између 15-30 минута за апликације са малим ризиком)
- *CAPTCHA* се не користи
  - (*CAPTCHA* и верификација е-поште имају различите сврхе, али су обе подједнако важне. *CAPTCHA* осигурава да стварни људи предају обрасце, а не скрипте. Верификација е-поште осигурава да унесена адреса е-поште заиста постоји и да функционише.
- Неадекватна енкрипција података
  - У пракси се често користе: *Triple* DES, RSA, *Blowfish*, *Twofish*, *Advanced Encryption Standard* (AES)

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА 4

- Коришћење рањивих API-ја
  - API-ји су кључеви база података компаније, тако да је веома важно ограничити и надгледати ко има приступ њима.
- Безбедносне закрпе нису примењене једнако на свим инсталираним верзијама (укључујући и верзије десктоп клијента)
  - уколико не постоји централизовано управљање закрпама односно верзијама апликације неопходно је пронаћи најбољи начин да се изврши провера и праћење примене закрпа (сви корисници морају да користе исту верзију апликације)
- Изворни код није тестиран на рањивости
  - уколико је апликација интерно развијена, потребно је да се ради тестирање изворног кода
- Код критичних апликација развијених од стране трећих лица, обезбедити континуитет пословања
  - Уколико је могуће, уговором предвидети да у случају престанка пословања произвођача компанија добије изворни код
- Пенетрациони тестови се не спроводе редовно
- Апликације које приступају бази података нису изоловане у засебан мрежни сегмент
- Примедбе корисника апликације се не решавају системски
  - Потребно је анализирати захтеве корисника и решења имплементирати кроз унапређења апликације (нова верзија апликације)

WEB АПЛИКАЦИЈЕ

- Основна дефиниција веб (**web**) апликације је програм који се покреће у прегледачу (**web browser**).
- То није веб сајт, али линија између њих је нејасна. Да бисте разликовали веб апликацију од веб сајта, запамтите ове три формалне карактеристике.
- Веб апликација:
  - бави се одређеним проблемом, чак и ако је једноставно проналажење неких информација
  - је интерактивна попут апликација које се инсталирају на рачунар (десктоп апликација)
  - има Систем за управљање садржајем (*Content Management System*)

- Веб сајт се традиционално подразумева само као комбинација статичних страница. Али данас се већина веб сајтова састоји и од статичних и од динамичких страница, што чини готово све модерне веб странице - веб апликацијама.

РЕВИЗИЈА WEB АПЛИКАЦИЈА

- Специфичности веб апликација
- Преглед архитектуре веб апликација
- Безбедносна подешавања сервера и конфигурације
- Аутентификација
- Управљање сесијом
- Руковање улазним и излазним подацима и њихова заштита
- Врсте напада

ПРИПРЕМА ЗА ИТ РЕВИЗИЈУ

- Да ли постоји спсак веб апликација и њена процена критичности?
- Да ли је апликација интерно развијена или је набављена од vendora?
- Ко је одговоран за одржавање?
- Утврдити архитектура апликације
- Утврдити администраторе и листу корисника
- Утврдити матрицу приступа (профили права приступа додељени корисницима апликације) и када је последњи пут ажурирана
- Утврдити метод аутентификације
- Проверити да ли постоје познате и документоване слабости апликације
- Проверити да ли је и када рађен последње скенирање на рањивости и/или penetration тест
- Да ли је било претходних ревизија и који су били налази
- Припрема чек листа за проверу

ИЗВОРИ ИНФОРМАЦИЈА

- Интерни извори
  - Ревизорски тим, резултати претходне ревизије
  - Веб администратори
  - Систем администратори
  - Веб девелопер-и
  - ИТ безбедност
  - Произвођачка документација о апликацији
  - Матрица приступа
  - Списак активних корисника апликације
  - Одобрена верзија апликације
  - Захтеви корисника за интервенцијом (тикети који се односе на апликацију, предмет ревизије)
  - Мрежни дијаграми
- Екстерни извори
  - Објаве произвођача апликације
  - CEERT / CSIRT / CIAC
  - Списак рањивости веб сервера
  - Најбоље праксе
  - Извештаји екстерне ревизије

РАЊИВОСТИ WEB АПЛИКАЦИЈА

- Стална тема годинама уназад
- Тестирање веб апликација може имати озбиљне последице
- Многе веб апликације су свој животни циклус започеле као мали интерни пројекти
- Пенетрациони тест
- OWASP

ГДЕ СУ УЗРОЦИ ПРОБЛЕМА

- Потпуно другачији концепт, окружење и ризици
- Едукација и тренинг WEB програмера
- Учење на примерима
  - Показује како се нешто може реализовати без осврта на безбедносне импликације
- Илустрација проблема рањивих веб апликација
  - GHDB – https://exploit.db.com/google-hacking-database/

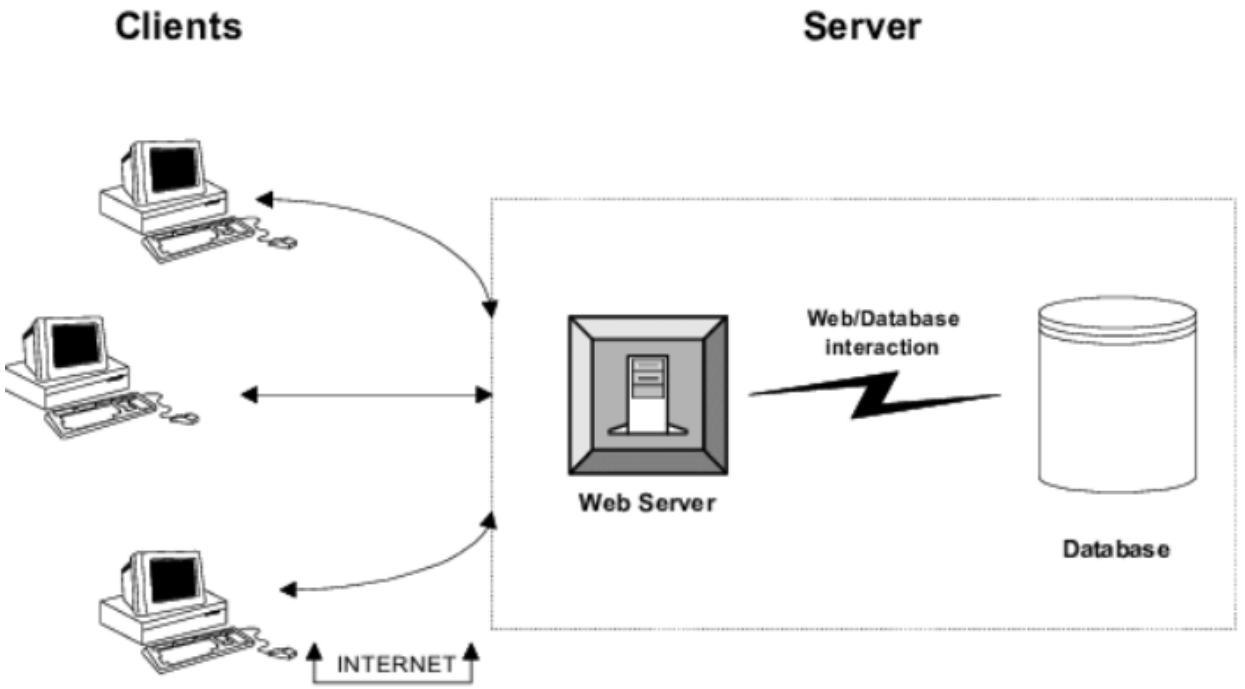
СВОЈСТВА ДОБРОГ ДИЗАЈНА

- Својства добро дизајниране веб апликације:
  - Валидација и санитизација улазних података
  - Детекција и корекција грешака
  - Аутентификација и управљање сесијом
  - Вишеслојна архитектура (*Multi tier architecture*)

- Најбоље је ако су сви слојеви одвојени:
  - Претраживач (*Browser*)
  - Веб сервер
  - Апликациони сервер
  - База

WEB АПЛИКАЦИЈЕ - АРХИТЕКТУРА

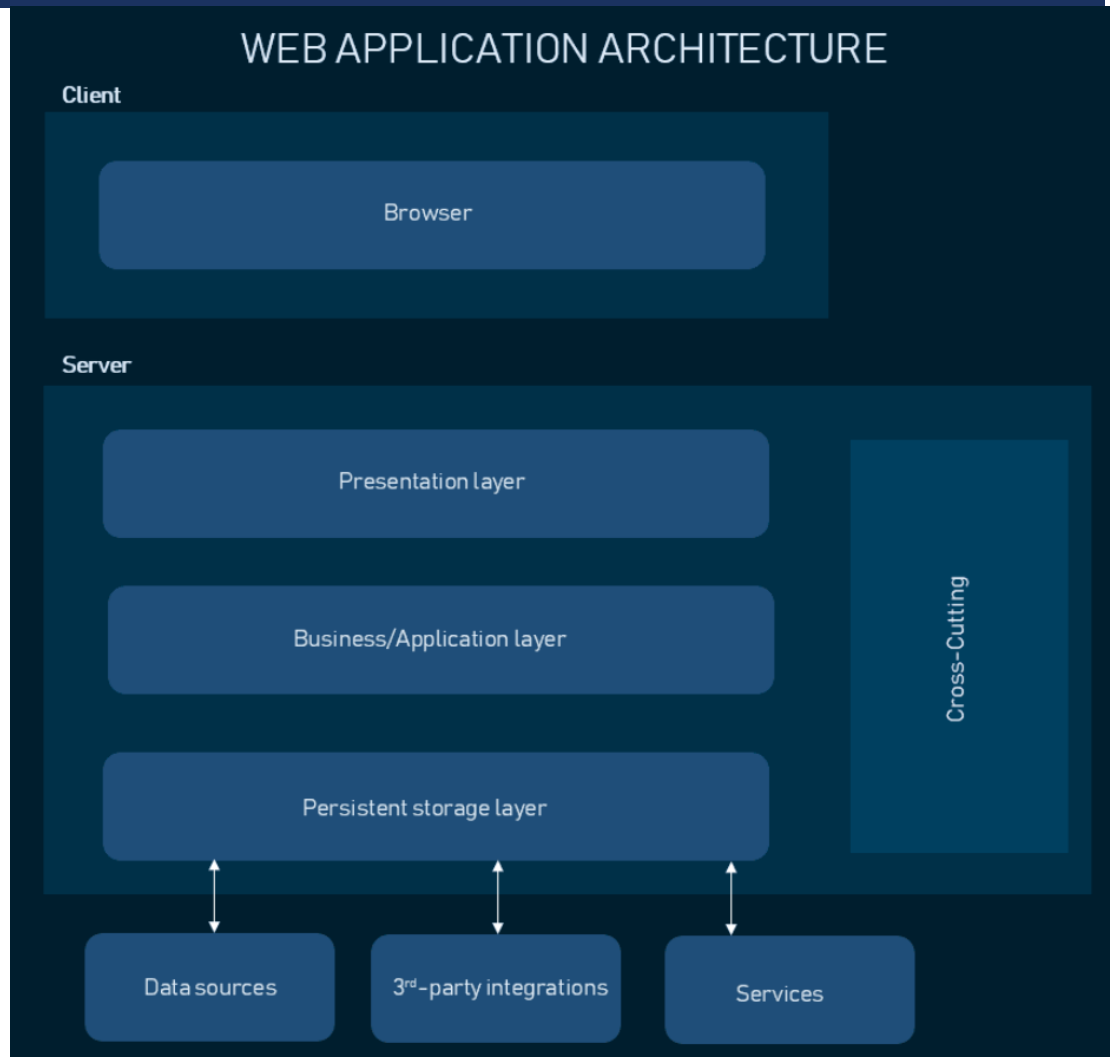
- Ваш рачунар, паметни телефон или било који други уређај са којим прегледавате назива се клијент.
- Друга половина веб једначине назива се сервером јер вам служи подацима које захтевате.
- Њихова комуникација назива се клијент-сервер модел, чија је главна брига примање вашег захтева и враћање одговора



WEB АПЛИКАЦИЈЕ – ТРОСЛОЈНА АРХИТЕКТУРА

- Компоненте архитектуре веб апликација и Трослојна архитектура:

Већина веб апликација развија се раздвајањем главних функција у слојеве. Ово вам омогућава да лако замените и надоградите сваки слој независно. Овај архитектонски образац назива се вишеслојна или трослојна архитектура



WEB АПЛИКАЦИЈЕ – СЛОЈЕВИ

- Презентацијски слој (*Presentation layer*)

Презентацијски слој је доступан корисницима путем прегледача и састоји се од компонената корисничког интерфејса и компонената корисничког интерфејса које подржавају интеракцију са системом. Развијен је користећи три основне технологије: HTML, CSS i JavaScript. Иако је HTML код који одређује шта ће садржати ваша веб локација, CSS контролише како ће изгледати. JavaScript и његови оквири чине вашу веб локацију интерактивном - реагујућом на радње корисника. Програмери користе JavaScript оквире као што су Angular и React како би садржај на страници учинили динамичним.

- Пословни слој (*Business layer*)

Овај слој, који се назива и пословна логика или логика домена или слој апликација, прихвата захтеве корисника из прегледача, обрађује их и одређује руте путем којих ће се приступити подацима. Токови посла којима подаци и захтеви путују кроз позадину кодирају се у пословном слоју. На пример, ако је ваша апликација веб локација за резервацију хотела, пословна логика ће бити одговорна за редослед догађаја кроз које ће путник проћи приликом резервације собе.

Иако пословна правила могу бити манифестација пословне логике, она нису иста. Понекад се пословна правила издвајају и њима се управља засебно.

- Слој постојаности (*Persistance Layer*)

Такође се назива слој за складиштење или приступ подацима, слој постојаности је централизована локација која прима све позиве података и омогућава приступ трајном складишту апликације. Слој постојаности је уско повезан са пословним слојем, тако да логика зна са којом базом података треба разговарати, а поступак преузимања података је оптимизованији.

ПРЕГЛЕД WEB ТЕХНОЛОГИЈА

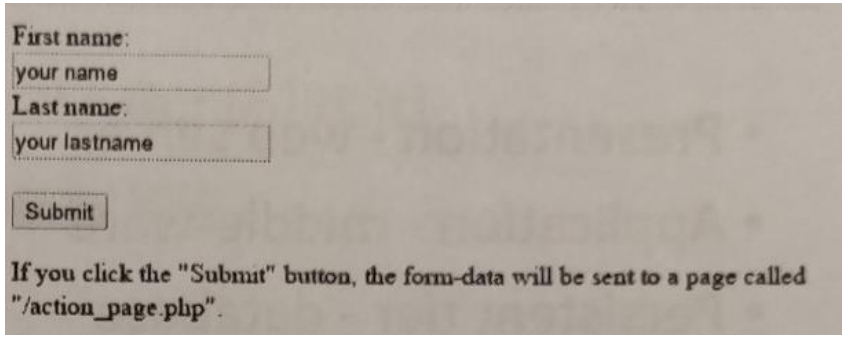
- HTML – *Hypertext Markup Language*
- HTTP – *Hypertext Transfer Protocol*
- HTTP FORM
- HTTP *methods* GET / POST / HEAD
- HTTP *cookie*
- *Javasript*
- AJAX – *Asynchronous JavaScript and* XML
- CSS – *Cascading Style Sheets*

HTML – ОСНОВНЕ НАПОМЕНЕ

```
<!DOCTYPE html>
<html>
<title>User form</title>
<body>
<img src= “w3schools.jpg“ alt= “w3schools.com“,,with= “104“ height= “142 “>
<form action= “/action_page.php“>First name:<br>
<input type= “text “ name=“First name“ value= “your name“>
<br>last name:<br>
<input type= “text “ name=“Last name“ value= “your lastname“>
<br><br>
<input type= “submit“value= “Submit“>
</form>
<p>If you click the “Submit “ button, the form-data will be sent to a page called “/action_page.php “.</p>
</body>
</html>
```

Почетак странице

Крај странице



HTML – OSNOVNE NAPOMENE

- HTML страница садржи тагове и текст
- *Case-insensitive*
- Наводници
  - Некадда нису обавезни ->‘value=“firefox-a” ‘> или ‘value=firefox-a’
  - Некада су обавезни value=“firefox-a”
- Постоје резервисани знакови, на пример & <>
- Резервисани знакови се у тексту замењују посебним кодом:
  - & &(amperstad, U+0026)
  - ⁢ <(less-than sign, U+003C)
  - > >(greater-than sign, U+003E)
  - " ” (quotation mark, U+0022)
  - '‘ (apostrophe, U+0027)
- Коментари се смештају између тагова <! ….. I ….. >

HTTP – ОСНОВНЕ НАПОМЕНЕ

- HTTP је механизам за трансфер HTML документа
- HTTP је текстуални протокол (clear text)
- Претраживач захтева HTML документ од сервера HTTP
- Одговор сервера садржи HTTP Header и HTML документ
- *Header* садржи поље са статусом одговора
- 1xx (info), 2xx (sucess), 3xx (redirection), 4xx (errors), 5xx (server errors)
- 200 OK, 404 Not Found, 502 Bad Gataway…
- HTTP методи: HEAD, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH
- Приликом логовања и преноса поверљивих информација, ОБАВЕЗНО је коришћење HTTPS (исто што и HTTP само криптован текст)

HTTP GET И POST МЕТОДЕ

- GET и POST методи се користе за слање захтева и података ка web серверу
- Примери
  - GET

https://www.w3schools.com/action_page.php?firstname=Petar&lastname=Petrovic

  - POST

http://192.168.1.1/index/login.cgi

- Post data:
  - Username[admin]
  - Password[&hdro,%djsber]

COOKIES

- *Cookie* (ili „колачић“) је текстуални податаку форми name=value
- Веб апликација креира *cookie* и придружује га HTTP одговору
- Претраживач претражује cookie сваком наредном захтеву који упућује тој истој апликацији
- *Cookie* поседује атрибуте:
  - *Secure*
  - *Http only*
  - *SameSite*
  - *Domain and Path*
  - *Expire and Max-age*
- Веб апликација помоћу атрибута даје претраживачу инструкције о животном циклусу *cookie*

РЕЗИМЕ КЉУЧНИХ ПОЈМОВА

- HTTP / HTML само текст
- HTTPS – енкриптовани HTTP
- POST смешта податке у body
- SSL обезбеђује енкрипцију, не и безбедност
- *Cookie* су „комадићи текста“ – атрибути
- Без изузетака, контрола свих параметара послатих серверу је на страни клијента

СКЕНИРАЊЕ НА РАЊИВОСТИ

- Утврђивање рањивости web апликација је важан и обавезан корак приликом одређивања безбедности
- Скенери:
  - Опште намене: *Nessus*, *Qualys* су корисни али не и једино довољни
  - WEB скенери:
    - Комерцијални: *Netsparker*, *Rapid 7 insightAppSec*, *Acunetix Web vulnerability scanner*, *Webinspekt*, *Wikto* …
    - Бесплатни (*open source*): *Grabber*, *Vega*, *Zed attack proxy*,
- Први корак: утврђује се идентитет сервера
  - *Fingerprint the OS*
  - *Fingerprint the web service*
  - *Fingerprint the add-ons*
- Други корак
  - Тестирају се само слабости релевантне за сервер и веб апликације

ЗА И ПРОТИВ АУТОМАТСКОГ СКЕНИРАЊА

- Добре стране:
  - Брзо и лако
  - Извештаји прилагођени различитим нивоима корисника
- Лоше стране:
  - Могу да се пронађу само познате врсте слабости
  - Користе познате технике и обрасце
  - Имају проблема са апликацијама које су „приватне“

Најбоља пракса је да се поред скенирања на рањивости ради и *penetration* тест (све чешћи захтев разних безбедносних стандарда)

БЕЗБЕДНОСТ АПЛИКАЦИЈА - OWASP

- OWASP (*Open Web Application Security Project*) је непрофитна фондација која ради на побољшању сигурности софтвера.
- Кроз пројекат отвореног кода које воде десетине хиљада чланова и водеће конференције о образовању и обуци, Фондација OWASP извор је програмерима и технолозима да осигурају мрежу.
  - Алати и ресурси
  - Заједница и умрежавање
  - Образовање и обука
- Произвођач апликације или локални програмери би морали бити упознати са препорукама и захтевима овог програма и примењивати их у пракси
- https://owasp.org/

БЕЗБЕДНОСТ АПЛИКАЦИЈА - OWASP

- Листа OWASP првих десет претњи представља широки консензус о томе које су најкритичније грешке у безбедности веб апликација. Чланови пројекта укључују разне стручњаке за безбедност из целог света који су поделили своју стручност за израду ове листе.
- Последња верзија је из 2017 године: https://owasp.org/www-project-top-ten/2017/
- OWASP топ 10 питања
  - 1.1. Injection
  - 1.2. Broken authentication and session management
  - 1.3. Sensitive Data Exposure
  - 1.4. XML External Entities
  - 1.5. Broken Access Control
  - 1.6. Security misconfiguration
  - 1.7. Cross-Site Scripting XSS
  - 1.8. Insecure Deserialization
  - 1.9. Using Components with Known Vulnerabilities
  - 1.10. Insufficient Logging & Monitoring

ОБЛАСТИ ОД ЗНАЧАЈА ЗА РЕВИЗИЈУ

- Конфигурација и безбедност сервера
  - OS, *Web server defaults*
  - Верзије
  - Статус инсталираних закрпа (*patch*)
  - *Directories, scripts, robots*
- Аутентификација
  - HTTP *основа*
  - *Form-based*
  - *Client сертификати*
- Управљање сесијом
  - *URL rewriting*
  - *cookies*
  - *Hidden field*
- Руковање улазним и излазним подацима
  - Филтер улазних података
  - Контрола излазних података
  - *Anticaching*

КОНФИГУРАЦИЈА И ИТ БЕЗБЕДНОСТ СЕРВЕРА

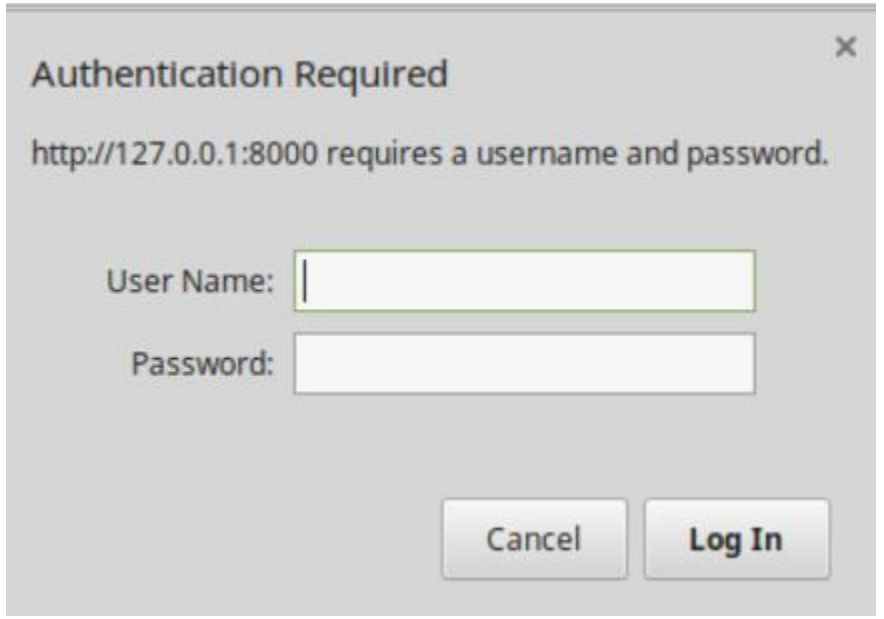
- Грешке у конфигурацији Оперативног система и веб сервера
  - Превише администратора
  - Конфигурациони фајл садржи креденцијале за конекцију са базом у clear text формату
  - Закрпе инсталиране на веб серверу а не и на оперативном систему, и обрнуто
  - *Directory indexing*
  - Аутентификација имплементирана на страни клијента
  - Линк странице за login садржи user name и password
  - Страница за логовање је HTTP уместо HTTPS
  - Након измене веб апликације стари фајл није обрисан већ је преименован
  - Default Web интерфејс за администрацију није уклоњен
  - Додатне информације у HTTP Header-у

АУТЕНТИФИКАЦИЈА

- Циљ: одредити да ли нам је механизам аутентификације безбедан
- Врсте контрола
  - Одговарајућа имплементација
  - Енкрипција
  - Јака средства аутентификације
- Методи аутентификације
  - HTTP *basic* аутентификација
  - *Form based* autentifikacija
  - *Client based certificates*
  - NTLM аутентификација

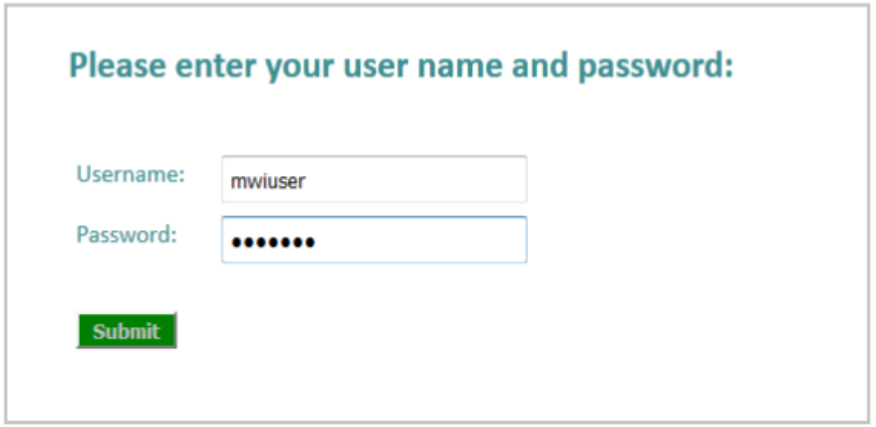
МЕТОДА HTTP АУТЕНТИФИКАЦИЈЕ

- Предности
  - Део је **HTTP** протокола
  - Лако се имплементира
- Слабости
  - Не постоји **log out** функција / потребно је затворити *browser*
  - Нема **enkripcije** (саобраћај који садржи лозинку је *clear text*)
  - Подложна *brute force* нападима
  - Креденцијали се шаљу серверу уз сваки захтев
  - Креденцијали имају улогу идентификатора сесија
- Препорука:
  - ОБАВЕЗНО применити енкрипцију комплетног саобраћаја



МЕТОДА FORM BASED АУТЕНТИФИКАЦИЈА

- Предности
  - Успоставља равнотежу између безбедносних захтева и корисничког искуства
  - Лако се имплементира
- Слабости
  - Могућ је *brute force* напад
  - Конфигурационе грешке могу довести до откривања креденцијала
  - Не примењује се енкрипција (*by default*)
- Препорука:
  - Обавезна употреба HTTP POST
  - Обавезна употреба енкрипције (TLS)
  - Поље за унос лозинке означити (type=PASSWORD)
  - Размотрити употребу дво-факторске аутентификације



МЕТОДА CLIENT SIDE СЕРТИФИКАТИ

- Предности
  - Висока безбедност
  - Непорецивост (*non-repudiation*)
  - Поверљивост
  - Узајамна аутентификација
- Слабости
  - Ограничена покретљивост клијента
  - Сложена администрација (нпр. Реиздавања и повлачење сертификата)
- Препорука:
  - Погодан за апликације где се захтева висок ниво безбедности
  - Ради повећања мобилности клијента могу се користити *smart* картице

МЕТОДА NTLM АУТЕНТИФИКАЦИЈЕ

- Предности
  - SSO решење (single sign on) за IIS i ASP:NET апликације
  - Корисник не мора да уноси креденцијале, користе се они са којима се улоговао у рачунар
  - Лака контрола активних корисника
- Слабости
  - Сервер мора бити IIS
  - Могућност злоупотребе уколико корисник оставља незакључан – не излогује се са рачунара
  - NTLM није стандард, безбедност осигурана само уз употребу TLS
- Препорука:
  - ДОбро решење за апликације у оквиру компаније (SSO са AD)

ПРЕТЊА: USER NAME HARVESTING

- Нападач користи функционалности страница за аутентификацију ради формирања листе корисничких имана
- Напад се заснива на информацијама које веб апликација емитује приликом:
  - Неуспешног покушаја пријављивања
  - Креирања налога за новог корисника
  - Подношења захтева за ресет лозинке
- Препорука:
  - Апликација треба да емитује једну поруку без обзира на то где је извор грешке

ПРЕТЊА: *BRUTE FORCE*

- Апликација закључава налог одређеног броја неуспешних покушаја пријављивања
  - Закључавање налога се може користи за откривање корисничких налога
- Велики број корисничких налога може бити закључан истовремено помоћу одговарајућих алата
- Рањиве су **HTTP basic form** и **form-based** аутентификације
- Препорука:
  - Користити технику успоравања одзива (**speed bump lockouts**) између неуспелих покушаја логовања и краткорочног закључавања налога (30 сек, 1 мин, 2 мин, 5 мин)
  - Алати за тестирање: *Brutus, Hydra, John the Ripper password cracker*

АУТЕНТИФИКАЦИЈА – ПРИМЕРИ ДОБРОГ ПРАВИЛА I

- Током аутентификације веб апликација мора обезбедити да се сваки корисник јединствено може идентификовати помоћу безбедног оквира.
- Дозвољени методи аутентификације су:
  - корисничко име и лозинка,
  - одговор на захтев (упит),
  - једнократне лозинке,
  - методи засновани на сертификату.
- Одговарајући методи аутентификације морају се евалуирати на основу процене ризика, узимајући у обзир регулаторне захтеве. Аутентификацију мора обезбедити сама веб апликација или преко интерфејса до централног система аутентификације и одобрења (нпр. Провајдер идентитета/IdP). Веб апликација мора бити у могућности да обезбеди методе аутентификације који одговарају безбедносним захтевима које ће дефинисати провајдер.
- У складу с тим, морају се евидентирати (логовати) сви успешни и неуспешни покушаји аутентификације

АУТЕНТИФИКАЦИЈА – ПРИМЕРИ ДОБРОГ ПРАВИЛА 2

- Без обзира на протокол комуникације који се користи, примењују се следећи захтеви за избора и чување лозинке:
- Тајна лозинке мора бити обезбеђена одговарајућим методама током процеса преноса и чувања лозинке (нпр. коришћењем криптографских функција као што је енкрипција током преноса или хешинг уз додавање додатних параметара заштите - енг, salting).
- Лозинке се морају чувати у "data layer"-у.
- Мора се спречити да се лозинке погоде помоћу бруте форце метода користећи одговарајуће процедуре (нпр. закључавање након 3 безуспешна покушаја, „ *tarpit*").
- Веб апликација мора бити у могућности да примора спровођење политике лозинки компаније
- Обавезно сетујте претраживач ДА НЕ ПАМТИ корисничке налоге и лозинке (иако је то корисницима лакше)

УПРАВЉАЊЕ СЕСИЈОМ

- HTTP је „*stateless*“ протокол
- Протокол дизајниран тако да се сваки захтев обрађује независно од претходног. Узастопни захтеви истог корисника нису међусобно повезани
- Сесија је секванца HTTP трансакција размењених између неког корисника и апликације
- *Session tracking* је механизам који омогућава апликацији / серверу да утврди да ли нови HTTP захтев припада сесији неког корисника
- *Session identifier* (SID) је средство помоћу којег је сесија једнозначно идентификована
- Сервер додељује SID након првог корисниковог HTTP захтева (на почетку сесије) и прослеђује га у оквиту HTTP одговора
- Сваки наредни HTTP захтев садржи SID
- Напомена: Сесија може бити успостављена и без претходне аутентификације

SESSION TRACKING И SESSION ID

- SID је јединствени идентификатор (SID=528546815)
- Апликација користи SID да би одредила стање (резултат претходних трансакција)
- Апликација управља животним циклусом SID
- *Session tracking* механизам обезбеђује трансфер SID између корисника и сервера
- URL *rewriting*
- *Cookies*
  - Веб апликације смештају SID у *cookie*
  - *Cookie* је описан текстуалним податком и атрибутима
  - Атрибути су средство које веб апликација користи за давање директива претраживачу
- *Hidden Fields*
  - Веб апликација смешта SID у скривена поља у коду HTML странице
  - Ова техника се не среће често

УПРАВЉАЊЕ СЕСИЈОМ

- Циљ:
  - Утврдити да ли је могуће искористити *session tracking механизам ради остваривања неауторизованог приступа*
- Претња: *Session Hijacking*
  - SID је средство којим се корисник идентификује
  - Нападач који дође у посед SID може да га искористи за неауторизован приступ
- Препорука
  - Кокристити робустан SID
  - Заштитити *session tracking* механизам

ЛИСТА ПРОВЕРА ЗА ИДЕНТИФИКАТОР СЕСИЈЕ

- Циљ:
  - Потврдити да сви генерисани идентификатори сесије задовољавају следеће:
    - SID је насумичан (генерисан помоћу генератора случајних бројева)
    - SID није у корелацији са било којом информацијом о кориснику нити је генерисан на основу тих информација
    - SID је довољно дугачак да спречи *brute force* напад
    - SID је временски ограничен и не може бити употребљен поново након прекида сесије
    - SID се транспортује само путем безбедног комуникационог канала
    - Пожељно је да SID подржава превенцију и детекцију покушаја манипулације (tampering)
- Алати
  - *Websrab* и слични *proxy* алати, *Tamper add-on* за *Firefox*

СЕСИЈЕ – ПРИМЕРИ ДОБРОГ ПРАВИЛА 1

- ID сесије мора се доделити кориснику да би се омогућила недвосмислена додела варијабли сесије кориснику. Даље улазне вредности корисника (нпр. вредности варијабли достављене уз веб апликацију) треба да се похрањују заједно са ID сесије на серверу, при чему се корисник дате сесије недвосмислено може идентификовати као извор података.
- Веб апликација мора обезбедити функцију која омогућава кориснику да прекине текућу сесију.
- Веб апликација мора генерисати идентификаторе сесије уз довољно високу ентропију и насумичност, тако да су током текуће сесије практично немогући успешни brute force напади користећи алате доступне на тржишту (најмање 128 бита насумичних података). Приликом сваке поновне аутентификације мора се генерисати нови идентификатор сесија.
- Веб апликација мора обезбедити да се сесија прекида након одређеног времена неактивности корисника тако што се означава да је сесија истекла. Време трајања сесије треба бити конфигурабилно.
- Након одјављивања које је покренуо корисник, варијабле сесије морају бити промењене (преписане) од стране веб апликације.
- Морају се енкриптовати информације о аутентификацији у варијаблама сесије. Не смеју бити укључене у ID сесије.

СЕСИЈЕ – ПРИМЕРИ ДОБРОГ ПРАВИЛА 2

- Уколико је корисник ауторизован путем одређене сесије, ID сесије мора бити пренесен само преко енкриптованог канала.
- Активност корисника не треба да се "кешују" када се користе осетљиве информације.
- Идентификатори сесија не смеју се преносити у URL (нпр. не прослеђујте ID сесије као GET параметре) и никада не треба да садрже личне податке.
- За строго критичне послове у веб апликацијама, треба узети у обзир следеће додатне контроле:
  - Допуњавање стандардног управљања сесијама коришћењем снажних насумичних токена или параметара по захтеву (не по сесији) да би се спречили CSFR (*Cross-site request forgery*) напади,
  - Не дозвољава истовремене пријаве користећи исти кориснички ID.
- Примењују се следећи захтеви када се у веб апликацији користи постојаност сесија:
  - Колачићи или објекти за складиштење HTML5 не смеју да садрже или да се користе за добијање осетљивих информација
  - Корисници морају бити у могућности да избришу колачиће или објекте за похрањивање и придружено стање у било којем тренутку (нпр. тастер за одјаву).
  - Информација у колачићу/објекту за похрањивање не смеју се дистрибуирати трећим лицима без сагласности корисника.
  - Савремене мере најбољих пракси морају се предузимати за обезбеђивање управљања колачићима/објектима за похрањивање HTML5 (нпр. атрибути за колачиће „Безбедно (енг. *Secure*)", „ HTTPOnly", „ Domen" и „Путања (eng. *Path*)".

РУКОВАЊЕ УЛАЗНИМ И ИЗЛАЗНИМ ПОДАЦИМА

- Осетљиви подаци које корисник упућује апликацији попуњавањем HTML форме морају бити заштићени током транспорта (TLS)
- Све HTML форме у апликацији треба да користе POST методу
- Улазни подаци корисника морају проћи валидацију која проверава податке у односу на садржај, веродостојност, валидни опсег вредности, дозвољене карактере, дужину и тип.
- Морају се применити филтери, кодирање или методи који не обрађују карактере контроле када се улазне вредности користе за приступ крајњим системима или датотекама, за извршавање команди оперативног система и за приступ другим интерфејсима. Мора се спречити да уноси тумача компромитују поверљивост, интегритет или доступност.
- Валидација улазних вредности корисника мора водити рачуна о кодирању које омогућава да се веб апликацији доставе вектори напада.
- Валидација улазних података мора се спроводити користећи уобичајену библиотеку валидација поља (нпр. унос ограничења, одбијање познатих лоших улазних вредности, санирање улазних вредности). Уколико уобичајена библиотека не садржи рутине за решавање следећих параметара улазних вредности, онда њих посебно треба решити:
  - Нула бајтова (%00)
  - Нови линијски карактери (%0d, %0a, \r, \n)
  - Карактери измене путање “"Dot-dot-slash” (../, ..\, %c0%ae%c0%ae/)
- Валидација података се мора радити на серверу.
- Уколико веб апликација захтева потенцијално опасне карактере као улазне параметре, морају се имплементирати додатне контроле као што су кодирање, безбедни API који се односе на задатак и коришћење података током апликације (примери уобичајених опасних карактера су: < > " ' % ( ) & + \ \' \").

ТРАНСФЕР ПОДАТАКА (*UPLOAD*)

- Морају се евидентирати датотеке које учитају корисници (најмање датум, време и IP адреса лица које ју је учитало).
- Трансферовање (upload) датотеке морају се подударати са списком дозвољених типова датотека (уколико је то могуће) и мора се урадити њихова валидација у односу на очекивани тип датотеке провером хедера датотеке.
- Садржај сваке учитане датотеке (укључујући злонамерни софтвер) мора се скенирати пре него што се даље похрани или обради.
- Дозвољена величина датотеке коју корисник трансферује (upload-ује) треба да се ограничи да би се спречило да злонамерни корисници попуне простор диска.
- Веб апликација не сме проћи директоријум или путање датотека у оквиру своје логике, већ користити вредности индекса мапиране у складу са унапред дефинисаним списком путања.

ХВАЛА