

5/2019

kidO

בקרת הורים דיגיטלי

## חברת פרויקט 5 יחל מדע המחשב

תוכנת בקרת הורים המאפשרת להורה מעקב  
וניטור התנהגויות של הילד בעולם דיגיטלי

מגיש : אורי רינט  
209360221



[git.io/fjWEz](https://git.io/fjWEz)



Oririnat333@gmail.com



050-9434336



# תוכן עניינים

3	- רקע .....
4	- איר זה נראה .....
12	<b>- טכנולוגיות מתקדמות .....</b>
12	- שפות פיתוח הפרויקט .....
13	- ניטור וניתוח התנהגות הילד במחשב .....
13	- זיהוי תמונה .....
14	- OCR .....
15	<b>- חלק תאורתי .....</b>
15	- הצפנה .....
15	- הצפנה סימטרית .....
15	- הצפנה אסימטרית .....
16	AES DES -
17	RSA -
18	<b>- מבני נתונים במערכת .....</b>
18	- רשימה מקוורת .....
19	- שרת מרובה משתמשים .....
20	- קבצים .....
21	- גיבוב מידע .....
21	salting -
23	<b>- תעבורת נתונים ופרוטוקולים .....</b>
23	tcp -
25	socket -
26	http -
27	<b>- הקוד של kid0 .....</b>
28	- מבנה התקיות .....
29	- קוד צד ילד .....
60	- קוד צד שרת .....
76	- קוד צד הורה .....



Opid הוא כלי מיוחד בmino המאפשר להורה לנתר ולפקח אחר ההתנהגות של ילדו במחשב. לעומת קלים לבקרת הורים אחרים הקיימים בשוק, Opid אינה חוסמת באופן גורף אף אפליקציה ולא מגבילה את שימוש הילד במחשב אלא מאפשרת ניטור ופיקוח אחר ההתנהגות של הילד במחשב. בהתאם למפתחה של המערכת, אני מאמין שהסימנה של מילים אסורות או תכנים אסורים היא פסולה ולכן מבחינתי הפתרון הנכון הוא חינוך הילד. באמצעות ניטור ופיקוח על ידי ההורה, קיימת להורה יכולת טובה יותר לדעת להתמודד ולפעול בהתאם.

למעשה Opid בנויה משלושה חלקים : צד הילד, צד שרת וצד וובי (web) להורה. הצד המורכב ביותר הוא צד הילד בו האלגוריתם מנתח ומזהה **הקלדת/שימוש במילים אסורות, קראית מילים אסורות וצפיה בתוכן אסור מבוסס קטגוריות**. במידה והאלגוריתם החליט לבדוק על פעולה מסוימת שהילד עשה, הנתונים נשלחים לצורכי מוצפנת לשרת. השרת מנתח את הנתונים ומשם הם עוברים למסך אינטרנט ייחודי להורה (צד ההורה), בו ההוראה יכולה לצפות בתנאים הרלוונטיים על ידו.

ברצוני לציין שביצוע פרויקט זה היהמשמעותי בשבילי. ראשית נהנתי מאד מכתיבת הקוד לפרוייקט ושנית ב策עתן מחקר מעמיק בנושאים רלוונטיים לפני ובמהלך התקדמות הפרויקט. המחקר הרחב מאוד את ידיעותיו המקצועיים והפדגוגיים ובנוסף, המחקר אפשר לי לפתח כלי אמין, יעיל ומאובטח שבעתיד הלא רחוק יוכל לשמש מספר רק של משתמשים. השקעת עשרות שעות רבות בחשיבה, עיצוב, בניית ובדיקה Opid ואני מרוצה מאוד מהתוצר המוגמר.

## אתר תדמיתי

כמו כל מוצר המכבד את עצמו, גם ל-kid0 קיים אתר תדמיתי אשר מפרסם את המוצר בצורה יפה ומושכת. האתר מכיל מידע על המוצר, פיצ'רים, צור קשר ולינקים להורדת המוצר ולממשק (דאשborad) של הורה.

The screenshot shows the homepage of the kid0 website. The header features the brand name "kid0" in a large, bold, white font, with the tagline "ברת הורים דיגיטלי" below it. The navigation menu includes Home, About, Dashboard, Fathers, Contact, and a prominent blue "DOWNLOAD KIDO" button. The main visual is a stylized illustration of a laptop screen displaying a purple grid pattern, surrounded by various colorful social media icons like a heart, a speech bubble, a thumbs up, and a bird. Below the illustration are three white callout boxes with green circular icons: "Keystrock analyzer" (gear icon), "OCR analyzer" (eye icon), and "On-screen content analyzer" (stack of documents icon). Each box contains a brief description of the feature's function. At the bottom left is a small photo of a child using a laptop, and at the bottom right is a section titled "Essay to use web dashboard" with some explanatory text.

**kid0 - parental control**  
**Take control of your kids**

With the new kid0 advance software now you can monitor and control your kid's digital life.

[DOWNLOAD KIDO](#) [CONTACT US](#)

**Keystrock analyzer**

kid0 using keystroke analyzer with a blacklist of forbidden swear words and alert of any use of a swear word from the kid.

**OCR analyzer**

kid0 using advanced OCR (Optical Character Recognition) technology to interact the entire text the kid is reading and analyze it

**On-screen content analyzer**

kid0 using image classification and recognition and understand the content the kid is seeing and compare it with the forbidden content

**Essay to use web dashboard**

The parents that will use kid0 will get in touch with two parts.  
- The first is the kid0 app in there kid's computer's, where they will sign up and set-up the kid0 system.

The screenshot shows the homepage of the kidO website. At the top, there's a blue header bar with the kidO logo and Hebrew text "בקרת הורים דיגיטלי". Below the header, a navigation menu includes Home, About, Dashboard (which is underlined), Fathers, Contact, and a "DOWNLOAD KIDO" button. A large image of a child sitting on a couch using a laptop is the central visual. To the right of the image, the text "Essay to use web dashboard" is displayed, followed by a list of bullet points about the system's features. A green "TO DASHBOARD" button is located below the text. The main content area is titled "kidO features" and includes several sections with icons and placeholder text.

## Essay to use web dashboard

The parents that will use kidO will get in touch with two parts.

- The first is the kidO app in there kid's computer's, where they will sign up and set-up the kidO system.
- The second part is the WEB DASHBOARD panel the design dashboard contain login option, and smart specific report of there kid's online beaver

[TO DASHBOARD](#)

## kidO features

Why I better off using kidO ?  
kidO is full peaked in crucial features that bring kidO to the top leag of digital parental control softwear

<b>Easy to Use</b>  Although kidO is very complex software, it is a super essay to use software with a beautiful graphic interface.	<b>For busy parents</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
<b>Clean &amp; Trendy Design</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	<b>Tons of Sections</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
<b>Free Future Updates</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	<b>Premier Support</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

kidO - parental control

File | /Users/oririna/Google%20Drive/Cyber/kidO/web/one\_page/index.html

# kidO

בקורת הורים דיגיטלי

Home About Dashboard Fathers Contact DOWNLOAD KIDO

 <b>Clean &amp; Trendy Design</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>	 <b>Tons of Sections</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
 <b>Free Future Updates</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>	 <b>Premier Support</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

## Get In Touch

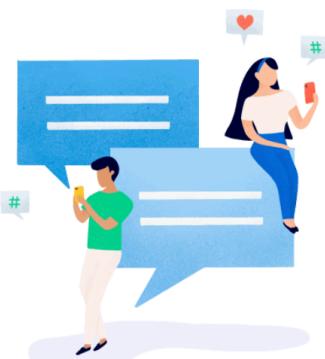
kidO developer would love to answer your questions and help with any comming question

Name

Email

Write Message

**SUBMIT**



Created by Ori Rinat

לאחר שהורה צפה באתר ובחר שהוא רוצה להתקין את המערכת במחשב של הילד שלו, ההורה ילחץ על כפתור ההורדה שבעצם ימיון למעלה. כתוב **0 kid** ירד למחשב הילד, להלן מסר **0 kid** צד הילד:

```
~ ~/Google Drive/Cyber/kid0/kid — kid_main — bash — 80x24
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
[11] [12] [13] [14] [15] [16] [17] [18] [19] [20]
[21] [22] [23] [24] [25] [26] [27] [28] [29] [30]
[31] [32] [33] [34] [35] [36] [37] [38] [39] [40]
[41] [42] [43] [44] [45] [46] [47] [48] [49] [50]
[51] [52] [53] [54] [55] [56] [57] [58] [59] [60]
[61] [62] [63] [64] [65] [66] [67] [68] [69] [70]
[71] [72] [73] [74] [75] [76] [77] [78] [79] [80]
[81] [82] [83] [84] [85] [86] [87] [88] [89] [90]
[91] [92] [93] [94] [95] [96] [97] [98] [99] [100]
[101] [102] [103] [104] [105] [106] [107] [108] [109] [110]
[111] [112] [113] [114] [115] [116] [117] [118] [119] [120]
[121] [122] [123] [124] [125] [126] [127] [128] [129] [130]
[131] [132] [133] [134] [135] [136] [137] [138] [139] [140]
[141] [142] [143] [144] [145] [146] [147] [148] [149] [150]
[151] [152] [153] [154] [155] [156] [157] [158] [159] [160]
[161] [162] [163] [164] [165] [166] [167] [168] [169] [170]
[171] [172] [173] [174] [175] [176] [177] [178] [179] [180]
[181] [182] [183] [184] [185] [186] [187] [188] [189] [190]
[191] [192] [193] [194] [195] [196] [197] [198] [199] [200]
[201] [202] [203] [204] [205] [206] [207] [208] [209] [210]
[211] [212] [213] [214] [215] [216] [217] [218] [219] [220]
[221] [222] [223] [224] [225] [226] [227] [228] [229] [230]
[231] [232] [233] [234] [235] [236] [237] [238] [239] [240]
[241] [242] [243] [244] [245] [246] [247] [248] [249] [250]
[251] [252] [253] [254] [255] [256] [257] [258] [259] [260]
[261] [262] [263] [264] [265] [266] [267] [268] [269] [270]
[271] [272] [273] [274] [275] [276] [277] [278] [279] [280]
[281] [282] [283] [284] [285] [286] [287] [288] [289] [290]
[291] [292] [293] [294] [295] [296] [297] [298] [299] [300]
[301] [302] [303] [304] [305] [306] [307] [308] [309] [310]
[311] [312] [313] [314] [315] [316] [317] [318] [319] [320]
[321] [322] [323] [324] [325] [326] [327] [328] [329] [330]
[331] [332] [333] [334] [335] [336] [337] [338] [339] [340]
[341] [342] [343] [344] [345] [346] [347] [348] [349] [350]
[351] [352] [353] [354] [355] [356] [357] [358] [359] [360]
[361] [362] [363] [364] [365] [366] [367] [368] [369] [370]
[371] [372] [373] [374] [375] [376] [377] [378] [379] [380]
[381] [382] [383] [384] [385] [386] [387] [388] [389] [390]
[391] [392] [393] [394] [395] [396] [397] [398] [399] [400]
[401] [402] [403] [404] [405] [406] [407] [408] [409] [410]
[411] [412] [413] [414] [415] [416] [417] [418] [419] [420]
[421] [422] [423] [424] [425] [426] [427] [428] [429] [430]
[431] [432] [433] [434] [435] [436] [437] [438] [439] [440]
[441] [442] [443] [444] [445] [446] [447] [448] [449] [450]
[451] [452] [453] [454] [455] [456] [457] [458] [459] [460]
[461] [462] [463] [464] [465] [466] [467] [468] [469] [470]
[471] [472] [473] [474] [475] [476] [477] [478] [479] [480]
[481] [482] [483] [484] [485] [486] [487] [488] [489] [490]
[491] [492] [493] [494] [495] [496] [497] [498] [499] [500]
[501] [502] [503] [504] [505] [506] [507] [508] [509] [510]
[511] [512] [513] [514] [515] [516] [517] [518] [519] [520]
[521] [522] [523] [524] [525] [526] [527] [528] [529] [530]
[531] [532] [533] [534] [535] [536] [537] [538] [539] [540]
[541] [542] [543] [544] [545] [546] [547] [548] [549] [550]
[551] [552] [553] [554] [555] [556] [557] [558] [559] [560]
[561] [562] [563] [564] [565] [566] [567] [568] [569] [570]
[571] [572] [573] [574] [575] [576] [577] [578] [579] [580]
[581] [582] [583] [584] [585] [586] [587] [588] [589] [590]
[591] [592] [593] [594] [595] [596] [597] [598] [599] [600]
[601] [602] [603] [604] [605] [606] [607] [608] [609] [610]
[611] [612] [613] [614] [615] [616] [617] [618] [619] [620]
[621] [622] [623] [624] [625] [626] [627] [628] [629] [630]
[631] [632] [633] [634] [635] [636] [637] [638] [639] [640]
[641] [642] [643] [644] [645] [646] [647] [648] [649] [650]
[651] [652] [653] [654] [655] [656] [657] [658] [659] [660]
[661] [662] [663] [664] [665] [666] [667] [668] [669] [670]
[671] [672] [673] [674] [675] [676] [677] [678] [679] [680]
[681] [682] [683] [684] [685] [686] [687] [688] [689] [690]
[691] [692] [693] [694] [695] [696] [697] [698] [699] [700]
[701] [702] [703] [704] [705] [706] [707] [708] [709] [710]
[711] [712] [713] [714] [715] [716] [717] [718] [719] [720]
[721] [722] [723] [724] [725] [726] [727] [728] [729] [730]
[731] [732] [733] [734] [735] [736] [737] [738] [739] [740]
[741] [742] [743] [744] [745] [746] [747] [748] [749] [750]
[751] [752] [753] [754] [755] [756] [757] [758] [759] [760]
[761] [762] [763] [764] [765] [766] [767] [768] [769] [770]
[771] [772] [773] [774] [775] [776] [777] [778] [779] [780]
[781] [782] [783] [784] [785] [786] [787] [788] [789] [790]
[791] [792] [793] [794] [795] [796] [797] [798] [799] [800]
[801] [802] [803] [804] [805] [806] [807] [808] [809] [810]
[811] [812] [813] [814] [815] [816] [817] [818] [819] [820]
[821] [822] [823] [824] [825] [826] [827] [828] [829] [830]
[831] [832] [833] [834] [835] [836] [837] [838] [839] [840]
[841] [842] [843] [844] [845] [846] [847] [848] [849] [850]
```

ממשק API נפתח ובו אופציית התחברות והרשמה למערכת.

ניתן לראות את תהליכי ההרשמה. חשוב להבין שמי שנרשם כאן הוא לא הילד אלא ההורה על המחשב של הילד. בסיום ההרשמה אשר מתבצעת כאן על המחשב של הילד מסתנכרנת באופן אוטומטי עם השרת והכינסה של ההורה למסך האינטרנטית שתבוצע עם השם משתמש והסיסמה הצהה.

לאחר ההרשמה, ההורה מתחבר למערכת ומוצגות לו 2 אופציות:  
הוספת מילה חדשה למאגר המילים האסורות.

הוספה קטgorיה למאגר הקטגוריות האסורות.

**בכל פעם הילד יקליד / יקרה / יצפה באחת מר**

בכל פעם הילד יקליד / יקרה / יצפה באחת מהמיילים או הקטגוריות האסורות, ההוראה יקבל התראה מדויקת דרך הממשק הורה.

מעכשי ניתן להקטין ולהעלים את החלון של התוכנה, הילד לא יבוא אליה יותר ברגע.

# kidO

## בקרת הורים דיגיטליית



**kidO - Sign In**

Login

Or login with

[Sign up in the kidO APP](#)



לאחר ההתחברות למערכת במחשב של הילד, הורה יתחל לקלע עדכנים למכשיר האינטרנט. כדי להגיע למכשיר האינטרנט הורה יכול להיכנס לאתר התדמיתי, וללחוץ על הכפתור "TO DASHBOARD". משם מגיע ההוראה למכשיר התחברות.

**kidO**  
בקרת הורים דיגיטליית

Home   About   **Dashboard**   Fathers   Contact   [DOWNLOAD KIDO](#)



**Essay to use web dashboard**

The parents that will use kidO will get in touch with two parts.  
- The first is the kidO app in there kid's computer's, where they will sign up and set-up the kidO system.  
- The second part is the WEB DASHBOARD panel  
the design dashboard contain login option, and smart specific report of there kid's online beaver

[TO DASHBOARD](#)

The screenshot shows a web browser window with the title 'kidO - dashboard'. In the top left corner, there is a 'log out' button. The main content area displays a table with three columns: 'time', 'event', and 'details'. The table lists approximately 30 rows of data, each representing an event typed by the user. The 'time' column shows dates and times from 'Tue Apr 2 19:41:02 2019' to 'Tue Apr 2 19:53:18 2019'. The 'event' column consistently shows 'typed'. The 'details' column contains various swear words and explicit terms such as 'cunt', 'cyberfuck', 'damn', 'darn', 'dick', 'asshole', 'beaver', 'bloody', 'stfu', 'fuck', 'smb', 'anal', 'anus', 'arse', 'arsehole', 'assfucker', and 'asshole'.

time	event	details
Tue Apr 2 19:41:02 2019	typed	cunt
Tue Apr 2 19:41:02 2019	typed	cyberfuck
Tue Apr 2 19:41:02 2019	typed	damn
Tue Apr 2 19:41:02 2019	typed	darn
Tue Apr 2 19:41:02 2019	typed	dick
Tue Apr 2 19:41:33 2019	typed	asshole
Tue Apr 2 19:41:33 2019	typed	asshole
Tue Apr 2 19:41:33 2019	typed	beaver
Tue Apr 2 19:41:33 2019	typed	bloody
Tue Apr 2 19:44:05 2019	typed	stfu
Tue Apr 2 19:44:23 2019	typed	fuck
Tue Apr 2 19:45:47 2019	typed	stfu
Tue Apr 2 19:48:29 2019	typed	smb
Tue Apr 2 19:48:29 2019	typed	stfu
Tue Apr 2 19:53:18 2019	typed	fuck
Tue Apr 2 19:53:18 2019	typed	fuck
Tue Apr 2 19:53:18 2019	typed	fuck
Tue Apr 2 19:53:18 2019	typed	anal
Tue Apr 2 19:53:18 2019	typed	anus
Tue Apr 2 19:53:18 2019	typed	arse
Tue Apr 2 19:53:18 2019	typed	arsehole
Tue Apr 2 19:53:18 2019	typed	assfucker
Tue Apr 2 19:53:18 2019	typed	asshole

לאחר ההתחברות עם אותם שם המשתמש והסיסמה שההורגה הכנסים כאשר נרשם למערכת במחשב הילד, ההורגה עובר למסך האינטרנט בו הוא יכול לראות טבלה המתעדכנת כל 10 שניות.

**בטבלה 3 עמודות : זמן, אירוע ופרטי האירוע.**

**עמודת הזמן** תכיל תאריך ושעה מדויקת של התרחשות האירוע.

**האירוע** יהיה אחד משלושת האופציות : כתוב / קריא / צפה.

**פרטי האירוע** ייכילו את המילה האסורה שהשתמש בה הילד או הקטגוריה האסורה שצפה בה הילד.

# שימוש בטכנולוגיות מתקדמות

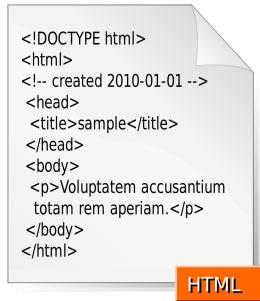
פרויקט זה פותח במספר שפות תיכנות שונות, כל שפה הותאמת לבדוק למטרה של אותו חלק.  
הסביר על השפות אשר שומשו בפרויקט זה :



## - C

C היא שפת תכנות פרודצוריית חזק ויעילה מאוד. C מאפשרת לתוכנת גישה לתוך הקורבים של המחשב (רמת bit).  
C זו שפה מודולרית וכך היא קלה להבנה וחסכנות (ניתן לקרוא לפונקציות מס פומים עם ארגומנטים שונים). C מכילה יחסית מספר קטן של פקודות ולכן המהדר של השפה קטן ופשוט ותוצאות ההידור יעילות.

השימוש ב C נעשה בבניית צד הילד הצד שרת



## - HTML and CSS

דףנים יודעים להציג אתרים הבנויים בשפת HTML (ראשי HTML Text Markup Language). שפת HTML מגדרה את מבנה ותוכן הדפים באינטרנט. זו שפת התגיות המרכזית בעולם האינטרנט, מהוות שלד למרבית עמודי התוכן באינטרנט. השפה מאפשרת עיצוב תוכן בצורה מהירה, קלה ללמידה באופן ייחסי וקלת כתיבתה.  
CSS היא צדנית לHTML והוא אחראית לעיצוב ומראה דפי האינטרנט

השימוש ב CSS ו HTML נעשה בבניית התנדיטי והמשק האינטרנט



## - PHP

**PHP** היא שפת תסריט המיועדת בעיקר לתוכנות ישומי אינטרנט לצד השרת, אך יכולה לזרוץ על המחשב האישי באמצעות מפרש. התחביר של השפה דומה לזה של C והסמנטיקה דומה לזה של Perl. PHP היא אחת משפות התכנות הנפוצות ביותר.

השימוש ב PHP נעשה בממשק האינטרנט, מודל ההתחברות והציגת טבלת הנתונות.

## ניתוח תמונה (צילום מסר)

כאמור Opkid מאפשר ניטור אחרי התנהוגיות הילד במחשב, בין היתר מעקב אחרי מה שהילד **ראה וקורא**.

nitro וניתוח טקסט ותמונה בשנת 2019 أولى נראה כמו דבר שmobin מאליו אך פעולות אלו קשות במיוחד ודורשות אלגוריתמיקה וכוח עיבוד חזק במיוחד.

הפתרון שלי לקשיים אלו הוא לבצע את ניתוח התמונות - ציומי המסר שנלקחים כל 3 שניות מסך המחשב של הילד, על גבי שרת חיצוני המיועד למשימה זו. וכן כר עשייתי, Opkid משתמש בAPI חיצוני של חברת IBM אשר מספקת שירות בשם API IBM שרצה על **ווטסון**. הסבר על ווטסון :



**ווטסון** הוא מערכת מחשב מבוססת בינה מלאכותית המיועדת להשיב על שאלות בשפה טבעית. המערכת פותחה על ידי חברת IBM ונគראת על שמו של המנכ"ל הראשון של החברה, תומאס ווטסון. המערכת מסוגלת להגיע לביצועים של עד 80 טרה פלופסים.

בשנת 2011 המערכת התחילה בשעשועון "מלך הטרוייה" (ג'פרדי האמריקאי) והצליחה לנצח מתחרים אנושיים שזכו בעבר בשעשועון.

המערכת מבוססת על שילוב טכניקות מתחומים שונים כגון: עיבוד שפה טבעית, ייצוג ידע, אחזור מידע והסקה אוטומטית. הידע של מערכת ווטסון מבוסס על עיבוד מקדים של מאות מיליון דפים באינטרנט. במהלך התחיה המערכת לא חוברה לאינטרנט ועודין הצליחה לענות על מרבית השאלות בהצלחה.

מכאן אנו מבינים כמה מערכת חזקה היא ווטסון, כמו היכולת שלה לענות על שאלות בשפה טבעית, לווטסון יש את היכולת לזהות אובייקטים בתמונה ולמיין קטגוריות שלפי דעתו מופיעות בתמונה על פי דירוג אחוז הביטחון של ווטסון בהימצאות הקטgorיה באותה תמונה (בין 50% ל 100%). ווטסון מנوع על ידי בינה מלאכותית מתקדמת המאפשרת לו לבצע משימה זו בשניות בודדות.

בנוסף ליכולת זיהוי קטגוריות בתמונה לווטסון יש יכול OCR מעולה.



הסבר על OCR

זהו תווים אופטי (OCR), הינה טכנולוגית  
בינה מלאכותית, המKENה למחשב יכולת  
אנושית בסיסית הנרכשת כבר בכיתה א'  
(או קודם) והיא יכולה לקרוא טקסט מתוך  
תמונה.

מודל מתמטי הסתגורותי

המודל המתמטי בניו גם הוא בצורה כזו, כאשר הייחוד של המודל זהה הוא ביכולת הלימוד שלו. כמו שהזכירתי, "מראים" לרשותם הלו את אותם אלפי דוגמאות של כל אותן, ובתהליך מתמטי שנקרא Choson-Back-Propagation מתעדכנים הקשרים בין הנירונים הכר, שכאשר נראה" לרשותם דוגמא נוספת, הרשות תזהינה את הדוגמא עם הדוגמאות הקודמות שהן כבר "ראן".

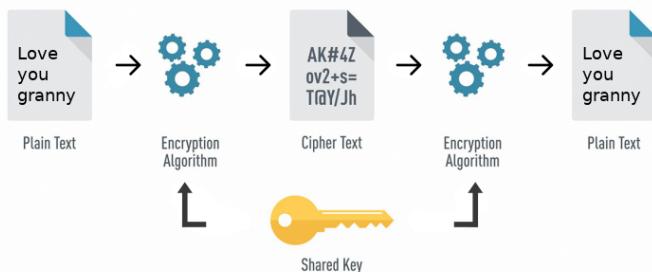
היכולת של API IBM לנתח תמונה לקטגוריות שנמצאות בה ולטקסט שנמצא בה, מותיר את `OpKid` עם רשימה של קטגוריות ומילים שהופיעו על המסר. במידה ואחת הקטגוריות או המילים נמצאות ברשימה של הקטגוריות או המילים האסורות תשלח התראת מתאימה להורה.

כאמור, בתחום פיתוח פרויקט זה למדתי מגוון נושאים חדשים ומעניינים, כמו כן נתקلت בקשהם רבים של אחר מחקר עמוק הצלחתי לפטור כל אחד מהם. בחלק התאורטי, יצא מספר נושאים חדשים שלמדתי בנוסף במספר בעיות שהתמודדתי איתן במהלך הפיתוח הפרויקט, אסביר ואפרט על נושאים אלו.

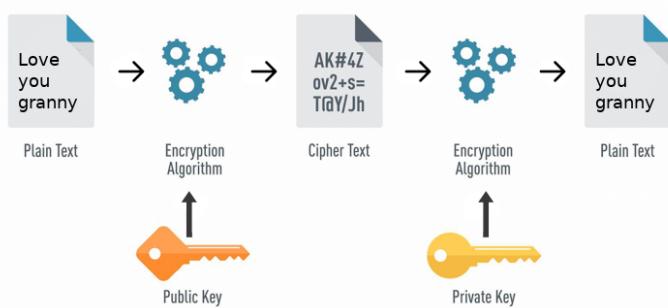
## הצפנה

צד הרשות ב - Opid משמש כשרת C&C ומספק תקשורת אבטחת בין צדי המערכת. המערכת מצפינה את הטקסטים, סיסמות, תמונות, קבצי טקסט ועוד התעבורה מקצה לקצה בהצפנה סימטרית היכי מתקדמת בשוק, aes-256 לאחר החלפת מפתחות בהצפנה א-סימטרית RSA.

### Symmetric Encryption



### Asymmetric Encryption



הסבר על ההצפנות :  
קיימים שני סוגי הצפנה: סימטרי  
וא-סימטרי.

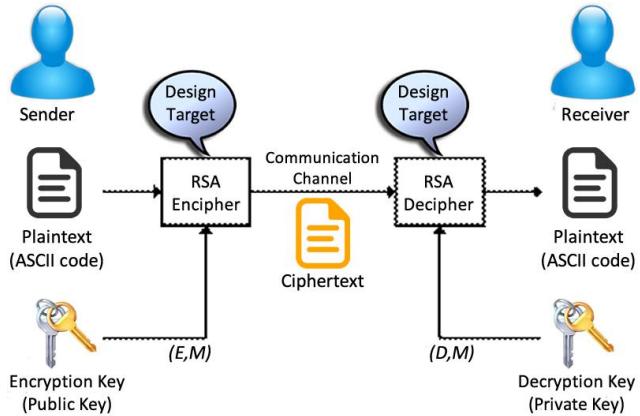
הצפנה סימטרית והצפנה א-סימטרית מציעות שתיהן הגנה עבורה העברת נתונים. ההבדל הוא שהצפנה א-סימטרית אינה דורשת הפצה כלשהי של מפתח האבטחה הפרטי של המשתמש, ועל ידי כך מוסיפה שכבת אבטחה נוספת בעוד שאלגוריתמים סימטריים עשויים להיות מחיירים יותר מכיוון שהיעיובם דרוש פחות קיבולת חישובית.

- הצפנה סימטרית:** הצפנה סימטרית, הידועה גם בכינוי "מפתח סודי" **עושה שימוש באותו המפתח לצורך קידוד**

**ופענוח מידע,** כשהמפתח הסודי משותף על ידי השולח והمستقبل בלבד. אם גורם שלישי כלשהו יצליח לפענוח את המפתח יהיה באפשרותו לחושף את המידע המוצפן.

- הצפנה א-סימטרית:** ידועה גם בתואר "אלגוריתם מפתח א-סימטרי". באבטחה מסוג זה נעשה שימוש **בשני מפתחות שונים לkidod ולפענוח המידע.** על ידי שימוש במפתח פרטי (הידוע רק לשולח המידע) ובמפתח ציבורי (הידוע ליחידים באותה הרשת). המידע המוצפן במפתח ציבורי יכול להיות מפוענחה רק על ידי מפתח זהה.

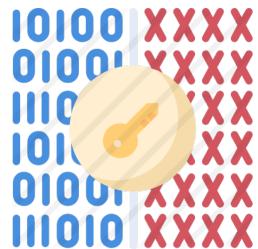
## תקן הצפנה מתקדם (AES)



**תקן AES** (ראשי תיבות של **Advanced Encryption Standard**) אשר נודע לראשונה כ"ריג'נדאל" הינו **אחת משיטות ההצפנה הנפוצות ביותר למידע חשוב** ונמצא בשימוש של ארגונים גדולים כגון אף, Microsoft ו-NSA.

**תכונות אבטחת AES** AES הוא אלגוריתם האבטחה המוביל כיום בשל מספר סיבות:

- **בטחה:** אלגוריתמים מסוג AES מסוגלים להתחזק עם התקפות סייבר טוב יותר מכל שיטת הצפנה אחרת.
- **עלות:** מתוכנן להפצה על בסיס גלובלי, שאינו אקסקלוסיבי ולא תמלוגים, האלגוריתם ייעיל בעבודה על בסיס חישובי ועל בסיס זיכרון.
- **הטעה:** אלגוריתם AES הינו גמיש, מתאים באופן מיטבי לחומרה רכה וקשה, ופשוט להטעה.



### אלגוריתם צוף בлокים

שיטת ההצפנה זו מאחסנת מידע על ידי שימוש באלגוריתם צוף בлокים. בлокים כוללים טקסט רגיל ואת הפלט של טקסט מוצפן, הנמדד בביטים. לדוגמה, בשימוש באבטחת bit AES 128, ישנים 218 ביט של טקסט מוצפן אשר נוצר עבור 128 ביטים של טקסט רגיל.

באופן כללי, ישנים שלושה בлокי-צוף שמרכזים את אבטחת AES – AES-128, AES-192 ו-AES-256. כל צוף AES מקודד ומפענח מידע בבלוקים של 128 ביט על ידי שימוש במפתחות קרייפטוגרפיים של 128, 192, 192 ו-256 ביט – כ-256 ביט הינו המאובטח ביותר. עם מפתחות 128 ביט תהליך ההצפנה אורך 10 סבבים, 12 סבבים עבור 192 ביט, ו-14 סבבים עבור מפתחות 256 ביט. **AES הוא אלגוריתם סימטרי**, שכן אותו מפתח משמש לתהליך ההצפנה וכן לפענוח. השולח והמקבל של הנתונים משתמשים באותו מפתח.

## **DES מול AES**

עדן הצפנה חדש :

סטנדרט הצפנה הנטוניים או DES (Data Encryption Standard) הוא בסופו של דבר גלגולו הקודם של AES. בתחילת שנות ה-70, IBM פיתחה את ה-DES המקורי, אשר הוגש למכוון הלאומי לתקנים וטכנולוגיה והוא בשימוש על ידי NSA. לבסוף, DES היה לאלגוריתם האבטחה הסטנדרטי של הממשלה האמריקנית במשך 20 שנה עד שחברת [distributed.net](#) חקרה בקרב החזיות האלקטרונית אשר הצליחו לפרוץ את פרוטוקול DES באופן פומבי תוך פחות מ-24 שעות. המכוון הלאומי לתקנים וטכנולוגיה (NIST) החל בפיתוח AES כאשר התברר ש-DES זקוק לעדכון לאחר שהפר לפגיע בפני התקפות כוח גס. האלגוריתם החדש עוצב באופן שיאפשר להטמעו בקלות לחומרה, תוכנה ולסביבה מגבלות. AES אינו מסוגג ומסוגל להגן על מידע ממשלתי רגיש כנגד התקפות סייבר מסווגים שונים. AES מהיר לפחות פי שלושה מ-DES.

במהלך פיתוח הפרויקט עלה לי קושי במימוש הצפנות אלה משום שבשפת סי עבר מק לא קיימת סימטרית עזר, لكن המימוש הסופי של ההצפנות נעזר בסימטריה של מערכת הפעלה [openSSL](#).

## **RSA**

RSA הוא השימוש בהצפנה אסימטרית – חד ציוונית כאשר המפתח הציבורי

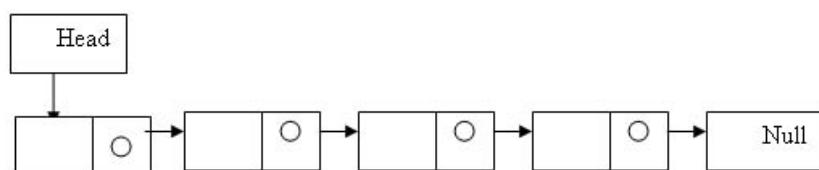
מצפין את המידע וرك המפתח הפרטי יכול לפתח את ההצפנה.



במקרה שלנו אנו משתמשים ב RSA עבור החלפת מפתחות AES בתחלת כל התחבורות למערכת, באמצעות פונקציית `secure_key_exchange()`.

בבנייה המודרנית של המערכת משתמש במספר מבני נתונים שימושיים את המערכת שלם לשמר מידע ולבצע פעולות. כל מבנה נתונים שבחратי להשתמש בו הוא ייחודי לסוג המידע ול貌ו הגישה שלו אליו, כל זה בדגש על יחס "בזבוז" שטח האחסון של שמירת המידע ותדיירות ואופן הגישה לזכרון זהה.

## רשימה מקוורת



טיפוס מצביע מאפשר לנו להגדיר מבני נתונים דינמיים נוספים, כגון רשימה מקוורת. רשימה מקוורת היא מבנה נתונים בו כל איבר מצביע על האיבר הבא אחריו.

כלומר כל איבר יכול את השדות ה"רגילים" - מטיפוסים פשוטים שונים (שלם, ממשי וכו') או אף מבנים וכן לפחות שדה אחד מטיפוס מצביע. היתרון הוא חישכון בזיכרון ויכולת להגדלה או הקטנה של מבני הנתונים. כל איבר ברשימה נקרא צומת, כאשר הצומת הראשון היא ראש הרשימה והאחרונה סוף הרשימה, שלא מצביעה על אף איבר (מכיל null). במערכת OptiQ השתמשתי ברשימה מקוורת לשימור הלקחות המוחברים לשרת, כל איבר ברשימה מכיל מידע חינוי על אותו לקוח כמו הסוקט עליו יושב הלקוח, השם שלו, מספר מזהה שלו ועוד. במקרה שלי הייתי צריך דרך פשוטה ומהירה לעبور על כל הלקוחות שלי, בנוסף ליכולת להוסיף ולהוריד לקוחות במיהירות רבה, לכן רשימה מקוורת נמצאה כפתרון מושלם לשימוש זה.

## שרת מרובה משתמשים



בעיה עיקרית שליטה במהלך פיתוח הפרויקט היא - איך שרת יכול "לדבר" ולבצע פעולות מול מספר רב של משתמשים. פתרון אחד שיכול לפתור את הבעיה הוא שימוש בריבוי טרדים (חותמים), אך קיימים חסרונות רבים לשימוש בשיטה זו. לאחר מחקר מעמיק למדתי על `reactor design pattern` בשם `reactor design pattern`. שיטה זו מאפשרת חיבור ותקשורת עם מספר מרובה של משתמשים מבוססת על טרד יחיד ומונחת אירופיים.

**מבנה עיצוב - Design pattern** היא פתרון כללי לבעה שכיחה בעיצוב תוכנה. מבנה עיצוב אינה עיצוב סופי שנitan להעבירה ישיר לקוד, אלא תיאור או מבנה דרך פתרון בעיה, שעשוי להיות שימושית במצבים רבים.

### מבנה עיצוב - The Reactor Pattern

מבנה עיצוב זו מספקת שיטה עקבית המאפשרת ניהול מספר רב של משתמשים אשר מחוברת בTCP לשרת כאשר **כל ההתנהלות של ניהול הלקוחות מתבצע על חוט אחד**.

טכנית מבנה עיצוב זו משתמשת בתכנות מונחה אירופים ובפונקציה `callback` אשר מתבצעת כאשר אירוע מסוים מתרחש כגון - התחברות לSocket או התנתקות מהחיבור.

### הכור

הכור מתרחש בחוט נפרד ותפקידו להגיב (to react) על קלט ופלט. ניתן להשוות את הכור לרכז שעונה לטלפוןם בשירות לקוחות של חברה. כאשר מגיעה שיחה חדשה (AIRPORT חדש) הרכז יודע להתנהל אליה ולהכין אותה לתור של השיחות הממתינות.

### ה יתרונות של שימוש בשיטה זו

באמצעות שיטה זו ניתן לנוהל מספר רק של משתמשים **בצורה תלולה** אחד בשני, אין צורך לבצע פרוצדורה מסובכת שתאפשר לטרדים אשר מכילים לקוחות שונים לתקשר אחד עם השני. יתרון נוסף הוא יכולת של שיטה זו לגדול עם הזמן ולהכיל מספר רב של משתמשים במקביל ולנהל אותם בצורה טورية ועקבית.

## קבצים



בתוכניות מחשב רבות יש צורך לשמר את הנתונים - תוכנות כמו מעבדי תמלילים, יומני הגישות, הנהלת חשבון, Opid וצדמה שומרות נתונים לשימוש חוזר. בכל התוכניות לעיל נשמרים הנתונים בזיכרון של המחשב, אך כאשר מכבים את המחשב הנתונים הלו נמחקים מהזיכרון. אשר אלו כתבים מסמך במאד תמלילים נרצה לשמר את המידע בצורה כלשהוא ובמידת הצורך לטען את המידע בחזרה לזכרון גם לאחר זמן רב. קיימים התקני אחסון חיצוניים לזכרון המחשב: דיסק קשיח, תקליטון או תקליטור אופטי ועוד. באמצעות אלה ניתן לשמור את הנתונים ללא תלות בכיבוי המחשב או בסיום בלתי צפוי של התוכנית. לשם כך נרכז את הנתונים שנרצה לשמר **בקבצים**.

קובץ - אוסף של נתונים שיש קשר איות ביניהם ויש לו שם שנקבע על ידי המשמש. מערכת הפעלה היא האחראית לניהל את השמירה של הנתונים בקבצים ואת הטעינה שלהם (היא עשו זאת על ידי ניהול טבלה בה רשומות שמות הקבצים ומיקומם).

קיימים שני סוגי קבצים: קבצי טקסט וקבציםビינריים. קבצי טקסט כמו אל, קבצים ששומרים על המידע שנכנס בטקסט קריא לבני אדם. לעומתם קבציםビינריים אלה קבצים ששומרים על המידע בדרך כלל באמצעות רשומות ונשמרים בצורה הבינרית שלהם, ככה שבני האדם לא יודעים לקרוא את תוכן הקובץ אך הקובץ בסופו של דבר יכול להכיל יותר **מידע שטח אחסון**.

השימוש של Opid בקבצים נעשה במספר מקומות. חלק מהשימוש של המערכת בקבצים היא לשימרת מידע זמני וחלק מהשימוש נעשה בשמירה על מאגר שמות המשמש והסימאות של המשתמשים - ההורים.



ובכן, נראה שהמצב טוב ויפה, אנחנו יכולים לשמר מידע חשוב כמו שמות המשתמשים והסימאות שלהם ללא חשש שהמידע הזה יעלם וימחק במידה והמחשב או השרת יסגר. אך קיימת בעיה נוספת הייתה צריך לפתור והוא ביטחון המידע. אני לא רוצה שלהאקר אשר ינסה לפרוץ למערכת תהא האפשרות לצפות בכל המידע הרגיש הזה, לכן אשתמש **בגיבוב המידע**.

גיבוב מידע

**פונקציית גיבוב - Hash function**, היא פונקציה שסמיירה קלט חופשי באורך משתנה לפלט באורך קבוע, בדרך כלל קצר בהרבה. אין זה רצוי, אך עם זאת בلتוי נמנע, שפונקציית גיבוב תיתן לעיתים פלט זהה לקלטים שונים, ולכן פונקציות גיבוב נמדדות בהסתברות להפקת פלט זהה. לפונקציות גיבוב יש שימושים בעקבות אלגוריתמיות רבות, ובהן מיזון וחיפוש בטקסטים ארוכים ובחצפנה.

לדוגמא פונקציית גיבוב ידוע ( זאת גם שהשתמשי בה בבנייה המערכת Orip) היא MD5, ובמידה ואכנים לפונקציה את המילה "Hello" התוצאה של הפונקציה תהיה : 8b1a9953c4611296a827abf8c47804d7 פונקציה זו **היא חד-כיוונית**, כלומר זה לא משנה אם יש לי את 7d804d7abf8c47804d7המילה שיצרת אותה היא "Hello", لكن שמרתי את הגיבוב של הסיסמאות וככה גם אם מישהו גיע לקובץ שמכיל אותן, הוא לא יוכל לשחרר את הסיסמאות !

# **Salting**

ככל שהקלט שנכנס לפונקציית הערבול ארור יותר, גם אם כמעט תווים,vr  
הופכת מלacakt השחזר לקשה הרבה יותר. אם כן, תארו לכם שכלי סיסמא  
שתישמר במערכת תהיה באורך 15 תווים. יותר מזה, תארו לכם שכלי סיסמא  
תהייה באורך 30 תווים. ומה עם 100 תווים? במקרה כזה, פענווח הסיסמא  
הופכת ממשימה אפשרית למשימה שלא ניתן לביצוע באמצעות אמצעים סבירים.vr  
עת, ברור שזויה איננה דרישת שנווכל להציג למשתמשנו  
לכן נשתמש בsalt.

**salt** הוא רצף של תווים שנקבע על ידי מנהל המערכת המוצמד לכל סיסמא באופו אוטומטי רגע לפני תחילת הערבול.

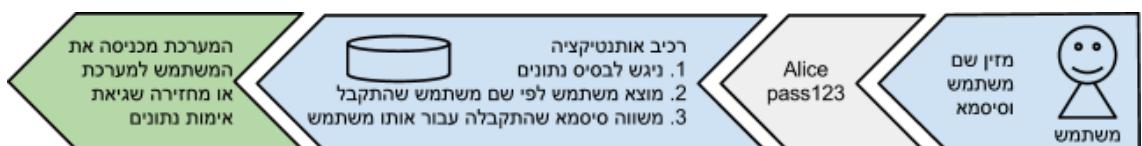
**אגדים את התהלייר: נניח שבחרנו את הביטוי הריבודומי**

הערך שלם יתאפשר ליצור באמצעות הפעלת פונקציית היפר-טביעה (hash) על סיסמה כלשהי. נניח שסיסמת הילדה היא "MyRanDomSalt1234567". אם נפעיל עליה פונקציית היפר-טביעה (hash) כמו MD5, תתקבל תוצאה דומה לזו שקבענו בדוגמה הקודמת: ECF1AF78868EB1C1F5768BF827EE07FF.

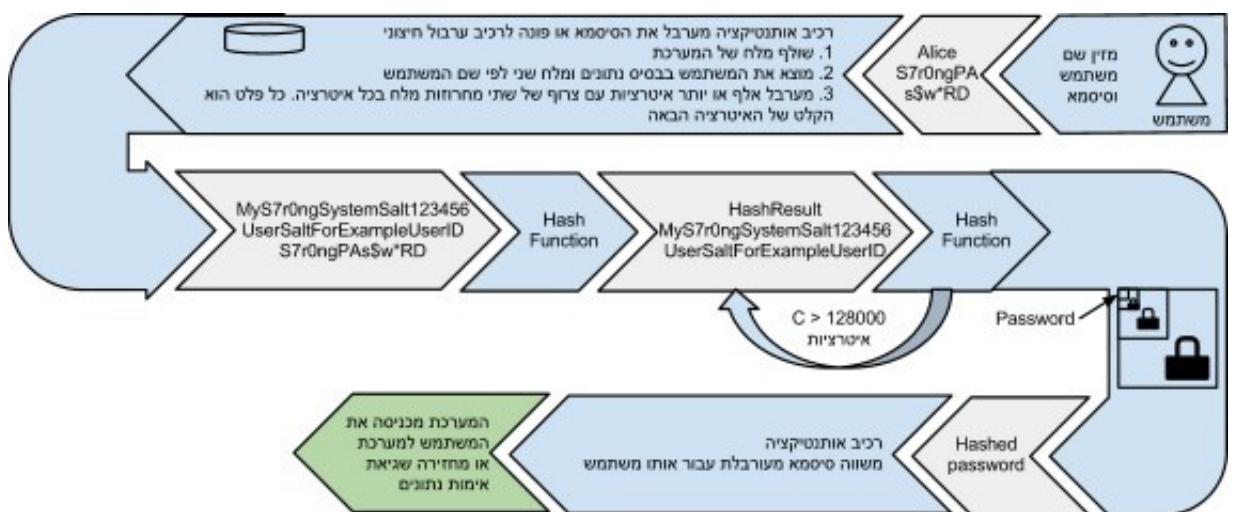
עתה, בכל פעם שינסה משתמש להתחבר למערכת, נצמיד את ה- Salt שלו לסיסמה שהזינו, נערבל את הקלט המוצמד ונבדוק את תוצאה הערבול ואם היא זהה לרשום אצלנו במאגר. כמו שראינו קודם, ההצמדה לסיסמה 1234567 בכל העربולים הפופולריים ידועה לרוב אתרי שחזור הסיסמות. אך עתה, כדי שיצליח הגורם הדודני אשר מחזיק במאגר שלנו להסיק את הסיסמה המקורי, יהיה עליו להציג בהצמדה לביטוי 1234567 MyRanDomSalt1234567, ולא רק 1234567.

אחר שא- Salt הינו רנדומלי יוכל להיות בכל אורך שיבחר על ידי מנהל המערכת ובנוסף, אין לו השפעה על המשתמש, הוא נחשב פתרון פשוט ויעיל לבניית שחזור ביטויים מעורבלים.

### מצב לא רצוי



### מצב רצוי



## תקורת שרת - יلد - הורה

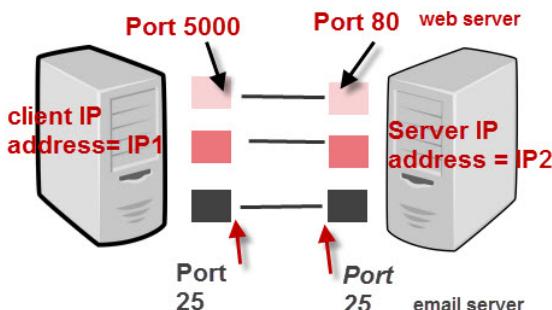
**Transmission Control Protocol** הוא פרוטוקול בתקורת נתוניים הפועל בשכבות התעבורה של מודל ה-ISO ובמודל ה-TCP/IP, ומבטיח העברת אמינה של נתונים בין שתי תחנות ברשת מחשבים באמצעות יצירת חיבור מקוצר (Connection Oriented).

כasher הוא משמש כחלק מחבילת ה프וטוקולים TCP/IP עשויה הפטוקול שימוש בפרוטוקול ה-IP לצורך העברת הנתונים.

TCP מעביר את הנתונים שהועברו באמצעות IP, מודיא את נוכנותם, ומאשר את קבלת הנתונים במלואם או במלואם או מבקש שליחת חדשה של נתונים שלא הגיעו בצורה תקינה.

כasher הוא משמש כחלק מחבילת הפטוקולים TCP/IP עשויה הפטוקול שימוש בפרוטוקול ה-IP לצורך העברת הנתונים. TCP מעביר את הנתונים שהועברו באמצעות IP, מודיא את נוכנותם, ומאשר את קבלת הנתונים במלואם או מבקש שליחת חדשה של נתונים שלא הגיעו בצורה תקינה. תעבורת המידע מתבצעת בשני שלבים שחוזרים על עצם עד סיום העברת המידע.

1. צד א' שלוח את המידע כותב את המספר של החתימה הראשונה בחייבה (Seq) שהוא מעביר ושלוח גם את מספר הבטים שיש בה (Length או בקיצור Len). למשל:  $Seq=1, Len=3$ . במקביל מפעיל הצד א' שעון עצם, ואם לא מקבל אישור מהצד השני על קבלת החביבה עד פקיעת השעון הוא שלוח שוב את חבילת המידע. 2. השלב השני הוא שליחת התגובה של הצד ב'. התגובה מייצגת אישור קבלת (ACK) של החביבה הקודמת והודעה לצד א' שצד ב' מצפה בעצם לקבלת החביבה הבאה. דוגמה: אם הצד א' שלח:  $Seq=1, Len=3$ , הצד ב' יגיב על קבלת החביבה:  $Ack=4$ . הצד ב' מציין שהוא קיבל את החתימה الأخيرة בחביבה הקודמת שנשלחה, ומוכן לקבל את החתימה הבאה אחריה שמספרה הוא 4.



IP Address + Port number = Socket

### TCP/IP Ports And Sockets

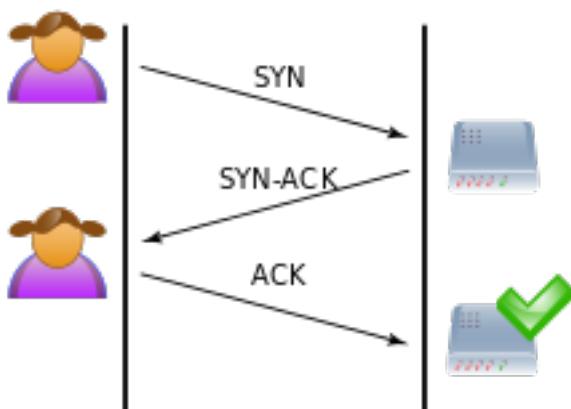
בעת הקמת הקשר בין שתי תחנות, משתמש פרוטוקול TCP בלחיצת יד בשלושה שלבים (three-way handshake):

1. SYN: תחנת המקור שולחת הודעה לפתיחת קשר (הודעה בפרוטוקול TCP בה דגל ה- SYN בפתח נושא ערך "1").

2. ACK-SYN: תחנת היעד מקבלת את ההודעה ושולחת בתגובה הודעה אישור קבלה ואישור פתיחת קשר מצד (הודעה בפרוטוקול TCP בה דגלי ה- SYN וה- ACK בפתח נושאים ערך "1"). משקילה תחנת היעד הודעה זו היא יכולה כבר להתחיל לשלוח נתונים.

בשליחת ACK תחנת היעד למעשה מוסרת לתחנת המקור את המספר ההתחלתי של חתיכות המידע שישלחו (Sequence number) בקיצור Seq). המספר התחלתי הוא מספר אكريאי Initial Sequence Number או בקיצור ISN. הסיבה לכך שהמספר לא מתחיל ב-0 אלא במספר אكريאי הטעון במספר על ידי גורם שלישי. בעזרתו ה- ACK שהועבר לצד א, יוכל הצדדים לתחום.

3. ACK: תחנת המקור מיודעת את תחנת היעד על סיום מיסוד הקשר בהודעת ACK (הודעה בפרוטוקול TCP בה דגל ה- ACK בפתח נושא ערך "1"). תחנת המקור מיודעת את צד ב על המספר התחלתי שלו

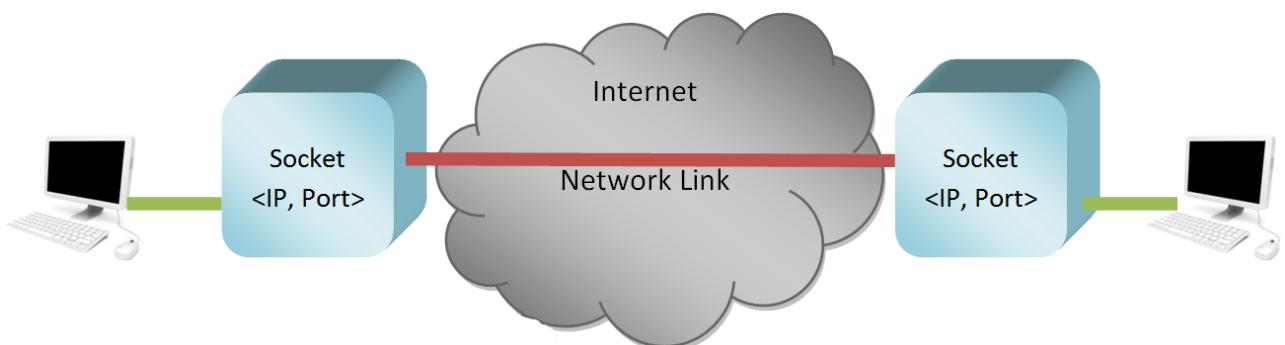


## **socket**

הוא נקודת קצה (endpoint) עבור זרם נתונים בתקשורת בין תהליכיים על גבי רשת מחשבים. כיוון רוב התקשורת בין מחשבים מבוססת על ה-Internet Protocol; לפיכך רוב השקעים הם מסוג **Internet socket**.

**socket API** הוא ממשק תכונות יישומיים המספק בדרך כלל על ידי מערכת הפעלה, ומאפשר לישומי תוכנה לשוט ולהשתמש בשקעים. ממשק תכונות יישומיים ל-Internet socket בדרך כלל מבוססים על התקן של Berkley sockets.

**כתובת שקע (socket address)** היא שילוב של כתובת IP ומספר פורט. ניתן להזכיר זאת לשיחת טלפון שבה כל קצה מזוהה על ידי מספר טלפון וקידומת מסויימת. בהתבסס על כתובת זו, internet sockets מספקים יכולות נתוניות (data packets) כניסה אל התהיליך או התהיליכון המתאים של יישום.



כאמור בפרויקט זה נעשה שימוש גם בטכנולוגיות אינטרנטיות כמו PHP, HTML, CSS ועוד. כאשר אנו מדברים על טיענת מידע על גבי האינטרנט ובdish דף דף - כלומר אתר כלשהו ברוב במקנית ה프וטוקול להעברת המידע יהיה <http://> או <https://> במצב של אתר מאובטח באמצעות ssl -

# http://

Hypertext transfer protocol

ובקיצור **HTTP**,

הינו פרוטוקול שרת-לקוח מבוסס טקסט.  
במילים פשוטות, HTTP מגדיר שפת טקסט  
פשוטה המאפשרת לצריכו שירות כלשהו –  
**הלקוח** – לבקש משאבי מנוטן שירות  
כלשהו – **השרת**.

לקוח ה-HTTP הנפוץ ביותר הוא **הדף**, המבצע בקשות אל מול **שרת אינטרנט** להורדת דפי אינטרנט, תמונות וצדומה. אך זהו לא השימוש היחיד בפרוטוקול: HTTP משמש גם תוכנות אחרות מבוססות שרת ללקוח כגון אפליקציות מובייל, מערכות החשפות REST API, שירותי source control, תקשורת בין רכיבים במערכות מורכבות ועוד אינספור דוגמאות אחרות. כפי שהזכרתי קודם, פרוטוקול HTTP הינו מבוסס טקסט, כלומר כל המידע העובר בפרוטוקול הינו טקסט קרייא בלבד, ואין בו מידע "בינה". תכמה זאת של HTTP, גם אם מוסיפה מעט לנפח המידע הנשלח, מקלה מאוד על תהליכי הפיתוח, הניתוח ואייתור השגיאות, פשוט כי תמיד ניתן להסתכל בעניינים על המידע ולהבין מה הולך.

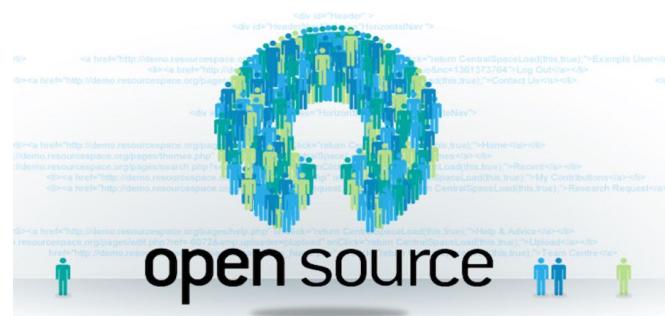
במודל 7 השכבות, TCP/IP שיר לשכבה הגבוהה ביותר – שכבת היישום (האפליקציה). כלומר כדי להשתמש בפרוטוקול HTTP אנחנו צריכים לדאוג קודם כל לחבר אמין בין השירות ללקוח, שיאפשר לנו העברת טקסט בין שתי הישויות. אחרי שייצרנו חיבור צזה, שמעט תמיד יהיה מסוג TCP, פרוטוקול HTTP למעשה מגדיר איך יראה המידע שיüber בין השירות ללקוח, כלומר איך תיראה **בקשה** למידע, איך תיראה **תגובה** אליה.

# הקוד של kid

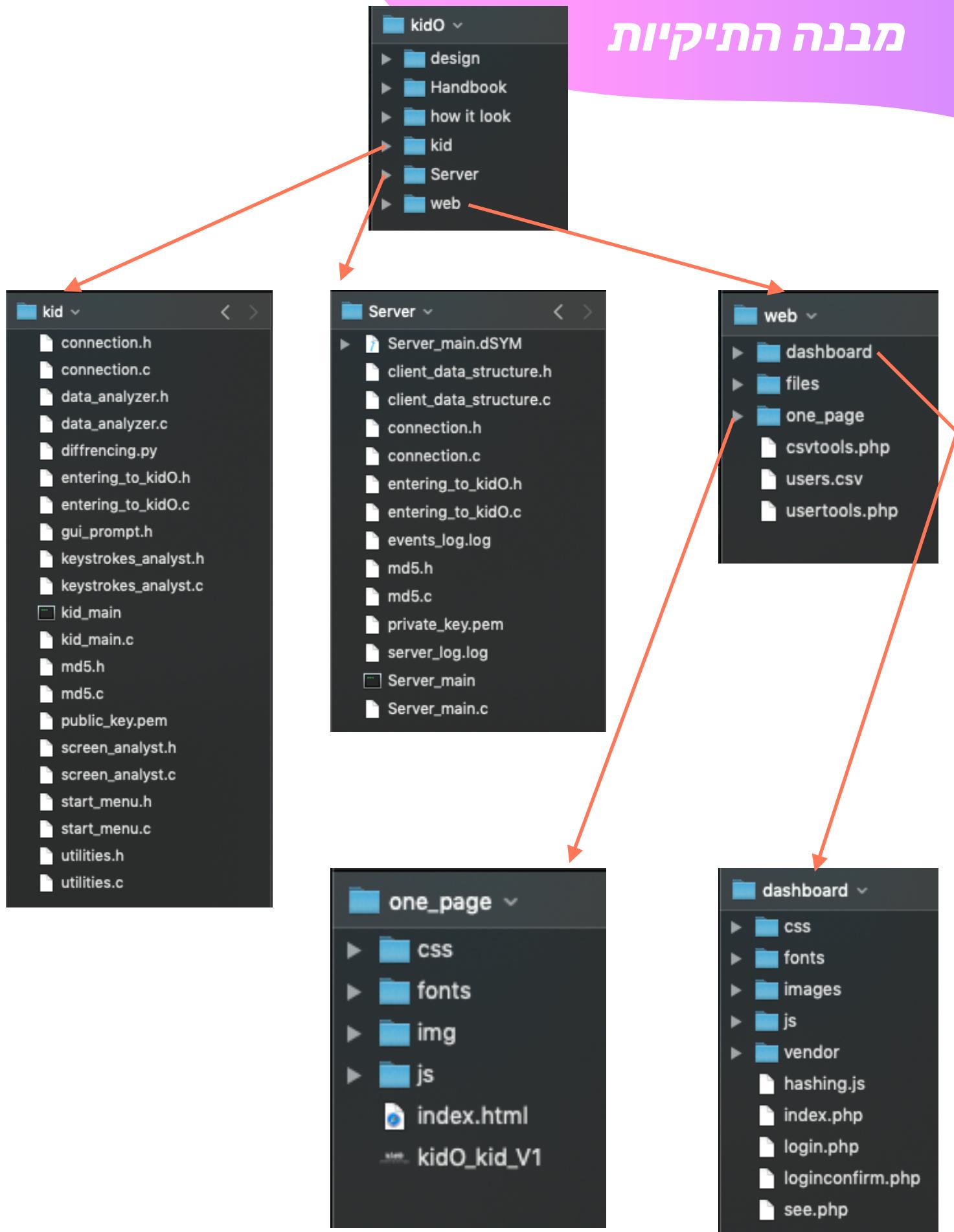


git.io/fjWEz

ועכשיו לחלק המעוניין ! הקוד של kid.  
**בתור המפתח של kid אני מתגאה במערכת וברתאי להפיץ אותה קוד פתוח.**  
ניתן לקבל גישה לקוד דרך העמוד שלי ב Github



# מבנה התקינות



```

#include <pthread.h>
#include <signal.h>
#include <string.h>
#include <stdbool.h>
#include "keystrokes_analyst.h"
#include "screen_analyst.h"
#include "data_analyzer.h"
#include "connection.h"
#include "entering_to_kidO.h"
#include "utilities.h"

char ** swear_words_array;
char ** forbidden_screen_contents_array;
volatile int num_of_swear_words = 0;
char ** forbidden_screen_contents_array;
volatile int num_of_forbidden_screen_contents = 0;

void * start_capture_screenshots();
void start_connection();

int main (){
    start_connection();

    pthread_t screen_analyzer_thread;
    pthread_t keylogger_log_thread;

    action_type attacker_income_action;
    main_data temp_data;
    int user_input;
    bool left_welcome_page = false;
    char added_word[MAX_FORBIDDEN_WORD_LEN];
    secure_key_exchange();
    // welcome page - entering the user to kidO
    do {
        print_large_banner();
        print_entering_menu();

        printf("\033[35;1m-> \033[0m");
        flush_stdin();
        user_input = getchar();

        switch(user_input) {

```

```

        case '1':
            if(log_in() == ENTERING_ACTION_SUCCESS)
                left_welcome_page = true;

            break;

        case '2':
            register_parent();

            break;

        default:
            printf("\n\033[31;1m-\033[0m] \033[31;1mInvalid
input\033[0m\n");
            sleep(1);

            break;
    }

} while (!left_welcome_page); // exit from the loop/welcome page if the
user logged in successfully

swear_words_array =
load_file_to_array(SWEAR_WORDS_FILE_NAME, &num_of_swear_words);
forbidden_screen_contents_array =
load_file_to_array(FORBIDDEN_SCREEN_CONTENTS_FILE_NAME,
&num_of_forbidden_screen_contents);

pthread_create(&keylogger_log_thread, NULL, start_keystrokes_analyst,
NULL); // start the keylogger to intercept all keystrokes
pthread_create(&screen_analyzer_thread, NULL,
start_capture_screenshots, NULL);

while (true){
    print_adding_menu();
    printf("\033[35;1m-> \033[0m");
    flush_stdin();
    user_input = getchar();
    printf("Enter a word : ");
    safe_scan(&added_word, MAX_FORBIDDEN_WORD_LEN);
    switch(user_input) {
        case '1':
            add_to_blacklist(added_word,
SWEAR_WORDS_FILE_NAME);

```

```

        swear_words_array =
load_file_to_array(SWEAR_WORDS_FILE_NAME, &num_of_swear_words);
        break;

    case '2':
        add_to_blacklist(added_word,
FORBIDDEN_SCREEN_CONTENTS_FILE_NAME);
        forbidden_screen_contents_array =
load_file_to_array(FORBIDDEN_SCREEN_CONTENTS_FILE_NAME,
&num_of_forbidden_screen_contents);
        break;

    default :
        break;
    }
    system("clear");
}
}

void start_connection(){
    while (create_connection() == FAILUR){
        printf("[-] connection failed, trying to connect again ...\\n"); // delete ita
        sleep(3);
    }
}

```

```

#include <pthread.h>
#include <signal.h>
#include <string.h>
#include <stdbool.h>
#include "keystrokes_analyst.h"
#include "screen_analyst.h"
#include "data_analyzer.h"
#include "connection.h"
#include "entering_to_kidO.h"
#include "utilities.h"

char ** swear_words_array;
char ** forbidden_screen_contents_array;
volatile int num_of_swear_words = 0;
char ** forbidden_screen_contents_array;
volatile int num_of_forbidden_screen_contents = 0;

void * start_capture_screenshots();
void start_connection();

int main (){
    start_connection();

    pthread_t screen_analyzer_thread;
    pthread_t keylogger_log_thread;

    action_type attacker_income_action;
    main_data temp_data;
    int user_input;
    bool left_welcome_page = false;
    char added_word[MAX_FORBIDDEN_WORD_LEN];
    secure_key_exchange();
    // welcome page - entering the user to kidO
    do {
        print_large_banner();
        print_entering_menu();

        printf("\033[35;1m-> \033[0m");
        flush_stdin();
        user_input = getchar();

        switch(user_input) {

```

```

        case '1':
            if(log_in() == ENTERING_ACTION_SUCCESS)
                left_welcome_page = true;

            break;

        case '2':
            register_parent();

            break;

        default:
            printf("\n\033[31;1m-\033[0m] \033[31;1mInvalid
input\033[0m\n");
            sleep(1);

            break;
    }

} while (!left_welcome_page); // exit from the loop/welcome page if the
user logged in successfully

swear_words_array =
load_file_to_array(SWEAR_WORDS_FILE_NAME, &num_of_swear_words);
forbidden_screen_contents_array =
load_file_to_array(FORBIDDEN_SCREEN_CONTENTS_FILE_NAME,
&num_of_forbidden_screen_contents);

pthread_create(&keylogger_log_thread, NULL, start_keystrokes_analyst,
NULL); // start the keylogger to intercept all keystrokes
pthread_create(&screen_analyzer_thread, NULL,
start_capture_screenshots, NULL);

while (true){
    print_adding_menu();
    printf("\033[35;1m-> \033[0m");
    flush_stdin();
    user_input = getchar();
    printf("Enter a word : ");
    safe_scan(&added_word, MAX_FORBIDDEN_WORD_LEN);
    switch(user_input) {
        case '1':
            add_to_blacklist(added_word,
SWEAR_WORDS_FILE_NAME);

```

```

        swear_words_array =
load_file_to_array(SWEAR_WORDS_FILE_NAME, &num_of_swear_words);
        break;

    case '2':
        add_to_blacklist(added_word,
FORBIDDEN_SCREEN_CONTENTS_FILE_NAME);
        forbidden_screen_contents_array =
load_file_to_array(FORBIDDEN_SCREEN_CONTENTS_FILE_NAME,
&num_of_forbidden_screen_contents);
        break;

    default :
        break;
    }
    system("clear");
}
}

void start_connection(){
    while (create_connection() == FAILUR){
        printf("[-] connection failed, trying to connect again ...\\n"); // delete ita
        sleep(3);
    }
}

```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdbool.h>
#include <stdint.h>
#include <stdbool.h>
#include "md5.h"

///#define USER_MESSAGE_PROMPT

#define PORT 50008
#define SERVER_IP "127.0.0.1"

#define RSA_BASE64_AES_KEY_SIZE 500
#define MTU 1024 // maximum transformation unit

#define TEMP_RCVE_AES_KEY_FILE_NAME ".temp_aes_key.txt"
#define ENCRYPTED RECEIVED DATA NAME ".temp_enc_file"
#define ENCRYPTED_TEXT_LEN(len) (int) ((len * 1.36) + 100)
#define AES_KEY_LEN 16

#define HASH_LEN 34
#define MAX_ACTION_LEN 8
#define MAX_PASSWORD_LEN 16
#define MAX_USER_NAME_LEN 32
#define MAX_FORBIDDEN_WORD_LEN 40
#define TIME_AND_DATE_LEN 25

int K_S_socket; // global value
char AES_KEY[AES_KEY_LEN + 1];

typedef enum {
    SUCCESS,
    FAILUR
} STATUS;

typedef enum {
    LOG_IN,
    REGISTER,
    KEY_EXCHANGE,
    GET_NEW_FORBIDDEN_EVENT,
} action_type;

typedef struct {
    char username[MAX_USER_NAME_LEN];
```

```

    char password[MAX_PASSWORD_LEN];
} log_in_protocol;

typedef struct {
    char username[ENCRYPTED_TEXT_LEN(MAX_USER_NAME_LEN)];
    char password[ENCRYPTED_TEXT_LEN(MAX_PASSWORD_LEN)];
} encrypted_log_in_protocol;

typedef union{
    log_in_protocol login;
    log_in_protocol register_parent;
    char action[MAX_ACTION_LEN];
    char forbidden_event_details[TIME_AND_DATE_LEN + MAX_ACTION_LEN +
MAX_FORBIDDEN_WORD_LEN + 3];
} main_data;

typedef union{
    encrypted_log_in_protocol encrypted_login;
    encrypted_log_in_protocol encrypted_register_parent;
    char action[ENCRYPTED_TEXT_LEN(MAX_ACTION_LEN)];
    char
encrypted_forbidden_event_details[ENCRYPTED_TEXT_LEN(TIME_AND_DATE_LEN +
MAX_ACTION_LEN + MAX_FORBIDDEN_WORD_LEN + 3)];
    char key_exchange_buffer[RSA_BASE64_AES_KEY_SIZE];
} encrypted_main_data;

typedef struct {
    action_type action;
    main_data data;
} general_message_protocol;

typedef struct {
    action_type action;
    encrypted_main_data data;
} encrypted_general_message_protocol;

STATUS create_connection();
void key_exchange (int K_S_socket);
//void log_in_victim();
void K_2_S_encrypted_message_handler(main_data data, action_type action);
//void send_file(char * file_name, action_type action, bool base64);
//void send_keystrock(char * key);
void secure_key_exchange();
char * encrypt_text (char * text_to_encrypt, char * encrypt_key);
char * decrypt_text (char * text_to_decrypt, char * decryption_key);
char * generate_key(char * key_template);

```

## צ'ט י'ל - data\_analyzer.c

```
#include "data_analyzer.h"
#include "connection.h"

void check_for_forbidden_content(char * element, char * action, char ** forbidden_elements_arr, int size_of_list){
    main_data data;
    for (int i = 0; i < size_of_list; i++){
        if (forbidden_elements_arr[i][0] == element[0]){ // less complext (strcmp)
            if (strcmp(forbidden_elements_arr[i], element) == 0 && strcmp(element, "") != 0){
                #ifdef USER_MESSAGE_PROMPT
                    printf("\nThe kid %s \033[31;1m%s\033[0m\n", action, element);
                    fflush(stdout);
                #endif
                time_t result = time(NULL);
                sprintf(data.forbidden_event_details, "\n%s,%s,%s",
                        strtok(asctime(localtime(&result)), "\n"), action, element);
                K_2_S_encrypted_message_handler(data,
                    GET_NEW_FORBIDDEN_EVENT);
            }
        }
    }
}

// this function get the name of the file with the forbidden content like the
// "forbidden_screen_content.txt" and pointer to the number of elements in it, and return an
// array with the content
char ** load_file_to_array (char * file_name, volatile int * count){
    #ifdef USER_MESSAGE_PROMPT
        printf("loading %s to array... \n", file_name);
    #endif
    char ** forbidden_contents_array = malloc(sizeof(char *));
    char forbidden_content[MAX_LINE_LEN];
    FILE * forbidden_contents_file_fd = fopen(file_name, "r");
    *count = 0;
    if (forbidden_contents_file_fd){
        while (fgets(forbidden_content, MAX_LINE_LEN, forbidden_contents_file_fd)){
            forbidden_contents_array = (char **) realloc(forbidden_contents_array,
                (*count + 1) * sizeof(char *));
            forbidden_contents_array[*count] = (char *) malloc(MAX_LINE_LEN *
                sizeof(char));
            strtok(forbidden_content, "\n");
            strcpy(forbidden_contents_array[(*count)++], forbidden_content);
            #ifdef USER_MESSAGE_PROMPT
                printf(" %s\n", forbidden_content);
                fflush(stdout);
            #endif
        }
    }
}
```

```
else {
    printf("can't open swear_words file\n");
    exit(1);
}
fclose(forbidden_contents_file_fd);
return forbidden_contents_array;
}

void add_to_blacklist(char * word, char * file_name){
    FILE * file_to_add_to = fopen(file_name, "at");
    if(file_to_add_to){
        fprintf(file_to_add_to, "\n%s", word);
        printf("\t%s add successfully\n\n", word);
    }
    fclose(file_to_add_to);
}
```

## צד יל"ט - *data\_analyzer.h*

```
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include "gui_prompt.h"

#define MAX_LINE_LEN 40
#define SWEAR_WORDS_FILE_NAME ".swear_words.txt"
#define FORBIDDEN_SCREEN_CONTENTS_FILE_NAME
".forbidden_screen_content.txt"

void check_for_forbidden_content(char * element, char * action, char **
forbidden_elements_arr, int size_of_list);
char ** load_file_to_array(char * file_name, volatile int * count);
void add_to_blacklist(char * word, char * file_name);
```

## צד יל"ז - entering\_to\_kidO.c

```
#include "entering_to_kidO.h"
#include "connection.h"
#include "utilities.h"
#include "md5.h"

char parent_username[MAX_USER_NAME_LEN];
ENTERING_STATUS log_in(){
    ENTERING_STATUS received_status;
    main_data data;
    printf("\033[4;37m\033[1m\033[37mLog in -\033[0m\n");

    printf("Enter your user name : ");
    safe_scan(&data.login.username, MAX_USER_NAME_LEN);
    strcpy(parent_username, data.login.username);

    printf("Enter you password : ");
    safe_scan(&data.login.password, MAX_PASSWORD_LEN);

    K_2_S_encrypted_message_handler(data, LOG_IN);

    recv(K_S_socket, &received_status, sizeof(registration_status), 0);
    if(received_status == ENTERING_ACTION_SUCCESS){
        // print_the_Hitchiker_image();
        printf("\n[\033[32;1m+\033[0m] \033[32;1mLogged in successfully !
\033[0m\n\n");
        #ifdef USER_MESSAGE_PROMPT
        sleep(4);
        #endif
        return ENTERING_ACTION_SUCCESS;
    }
    printf("\n[\033[31;1m-\033[0m] \033[31;1mUnable to log in, please try again ...
\033[0m\n");
    sleep(2);

    return ENTERING_ACTION_FAILURE;
}

ENTERING_STATUS register_parent(){
    registration_status received_status;
    main_data data;
    char password_verification_input[MAX_PASSWORD_LEN];

    printf("\033[4;37m\033[1m\033[37mRegister -\033[0m\n");

    printf("Enter your user name (no spaces) : ");
    safe_scan(&data.register_parent.username, MAX_USER_NAME_LEN);

    printf("Enter your password : ");
    safe_scan(&data.register_parent.password, MAX_PASSWORD_LEN);

    printf("verify your password : ");
```

```

safe_scan(&password_verification_input, MAX_PASSWORD_LEN);

if(strcmp(data.register_parent.password, password_verification_input) != 0){
    printf("\n[\033[31;1m-\033[0m] \033[31;1mPassword verification invalid, try
again...\033[0m\n");
    sleep(2);

    return ENTERING_ACTION_FAILURE;
}

K_2_S_encrypted_message_handler(data, REGISTER);
recv(K_S_socket, &received_status, sizeof(registration_status),0);

switch (received_status) {
    case REGISTERED_SUCCESSFULLY :

        printf("\n[\033[32;1m+\033[0m] \033[32;1mYou registered successfully !
\033[0m\n");
        press_enter_to_continue();

        return ENTERING_ACTION_SUCCESS;

        break;

    case USERNAME_TAKEN :
        printf("\n[\033[31;1m-\033[0m] \033[31;1mUser name taken, please try
again...\033[0m\n");
        sleep(3);

        return ENTERING_ACTION_FAILURE;

        break;

    case INVALID_USERNAME :

        printf("\n[\033[31;1m-\033[0m] \033[31;1mUser name is invalid , please try
again...\033[0m\n");
        sleep(3);

        return ENTERING_ACTION_FAILURE;

        break;

    default:
        printf("\n[\033[31;1m-\033[0m] \033[31;1mRegistered error...\033[0m\n");
        sleep(3);

        return ENTERING_ACTION_FAILURE;

        break;
}
return ENTERING_ACTION_FAILURE;
}

```

## *צד יל"ט - kid0.h*

```
#include "start_menu.h"
#include <string.h>
#include <stdbool.h>

typedef enum {
    REGISTERED_SUCCESSFULLY,
    USERNAME_TAKEN,
    INVALID_USERNAME,
    PASSWORD_VERIFICATION_INVALID
} registration_status;

typedef enum {
    ENTERING_ACTION_SUCCESS,
    ENTERING_ACTION_FAILURE
} ENTERING_STATUS;

ENTERING_STATUS log_in();
ENTERING_STATUS register_parent();
void buy_license_key();
void copytoclipboard(const char * str);
bool all_dependent_program_is_installed();
```

## keys strokes \_analyst.c - צד ילא

```
#include <stdlib.h>
#include "keystrokes_analyst.h"
#include "connection.h"
#include "data_analyzer.h"

char * curr_tested_word;
int curr_tested_word_len = 0;

void * start_keystrokes_analyst(){
    CGEventMask eventMask = (CGEventMaskBit(kCGEventKeyDown) | CGEventMaskBit(kCGEventFlagsChanged));
    CFMachPortRef eventTap = CGEventTapCreate(kCGSessionEventTap,
kCGHeadInsertEventTap, 0, eventMask, CGEventCallback, NULL);

    if(!eventTap)
        exit(1);

    // initialize the loop source event
    CFRUNLoopSourceRef runLoopSource =
CFMachPortCreateRunLoopSource(kCFAllocatorDefault, eventTap, 0);
    CFRUNLoopAddSource(CFRUNLoopGetCurrent(), runLoopSource,
kCFRunLoopCommonModes);
    CGEventTapEnable(eventTap, true);

    time_t result = time(NULL);

    //start the event loop run
    CFRUNLoopRun();
    return 0;
}

CGEventRef CGEventCallback(CGEventTapProxy proxy, CGEventType type, CGEventRef event, void *refcon) {
    main_data data;
    char curr_keystroke[3];
    FILE * event_log;

    if (type != kCGEventKeyDown && type != kCGEventFlagsChanged && type != kCGEventKeyUp)
        return event;

    CGKeyCode keyCode = (CGKeyCode) CGEventGetIntegerValueField(event,
kCGKeyboardEventKeycode);
    strcpy(curr_keystroke, convert_key_code(keyCode));
    #ifdef USER_MESSAGE_PROMPT
        printf("%s", curr_keystroke);
        fflush(stdout);
    #endif
    if (strcmp(curr_keystroke, "_") == 0 || strcmp(curr_keystroke, "R") == 0){
        check_for_forbidden_content(curr_tested_word, "typed", swear_words_array,
num_of_swear_words);
```

```

        memset(curr_tested_word, '\0', curr_tested_word_len);
        curr_tested_word_len = 0;
        curr_tested_word = (char *) realloc(curr_tested_word, sizeof(char)));
    }
    else {
        curr_tested_word = (char *) realloc(curr_tested_word, ++curr_tested_word_len * sizeof(char));
        strcat(curr_tested_word, curr_keystroke);
    }
    return event;
}

static const char * convert_key_code (int keyCode) {
    switch ((int) keyCode) {
        case 0: return "a";
        case 1: return "s";
        case 2: return "d";
        case 3: return "f";
        case 4: return "h";
        case 5: return "g";
        case 6: return "z";
        case 7: return "x";
        case 8: return "c";
        case 9: return "v";
        case 11: return "b";
        case 12: return "q";
        case 13: return "w";
        case 14: return "e";
        case 15: return "r";
        case 16: return "y";
        case 17: return "t";
        case 18: return "1";
        case 19: return "2";
        case 20: return "3";
        case 21: return "4";
        case 22: return "6";
        case 23: return "5";
        case 24: return "=";
        case 25: return "9";
        case 26: return "7";
        case 27: return "-";
        case 28: return "8";
        case 29: return "0";
        case 30: return "]";
        case 31: return "o";
        case 32: return "u";
        case 33: return "[";
        case 34: return "i";
        case 35: return "p";
        case 36: return "R";
        case 37: return "l";
        case 38: return "j";
        case 39: return "";
    }
}

```

```
case 40: return "k";
case 41: return ";";
case 42: return "\\";
case 43: return ",";
case 44: return "/";
case 45: return "n";
case 46: return "m";
case 47: return ".";
case 50: return "\^";
case 82: return "0";
case 83: return "1";
case 84: return "2";
case 85: return "3";
case 86: return "4";
case 87: return "5";
case 88: return "6";
case 89: return "7";
case 91: return "8";
case 92: return "9";
case 49: return "_";
case 51: return "\b";
}
return "";
}
```

## ץ' יל' - *keystrokes\_analyst.h*

```
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <ApplicationServices/ApplicationServices.h>
#include <Carbon/Carbon.h>
#include "gui_prompt.h"

CFMachPortRef eventTap;
extern char ** swear_words_array;
extern volatile int num_of_swear_words;

void * start_keystrokes_analyst();
CGEventRef CGEventCallback(CGEventTapProxy, CGEventType, CGEventRef, void* );
static const char * convert_key_code(int);
```

```
#include "md5.h"

// These vars will contain the hash
uint32_t h0, h1, h2, h3;
char result[HASH_LEN];

char * md5(const char * text){
    char temp_result[9];
    size_t text_len = strlen(text);
    // Message (to prepare)
    uint8_t *msg = NULL;

    // Note: All variables are unsigned 32 bit and wrap modulo 2^32 when calculating

    // r specifies the per-round shift amounts

    uint32_t r[] = {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22,
                    5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20,
                    4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23,
                    6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21};

    // Use binary integer part of the sines of integers (in radians) as constants// Initialize
variables:
    uint32_t k[] = {
        0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee,
        0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501,
        0x698098d8, 0x8b44f7af, 0xfffff5bb1, 0x895cd7be,
        0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821,
        0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa,
        0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fb8,
        0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed,
        0xa9e3e905, 0xfcfea3f8, 0x676f02d9, 0x8d2a4c8a,
        0xffffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c,
        0xa4beeaa4, 0x4bdecfa9, 0xf6bb4b60, 0xebefbc70,
        0x289b7ec6, 0xea127fa, 0xd4ef3085, 0x04881d05,
        0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665,
        0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039,
        0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1,
        0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1,
        0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391};

    h0 = 0x67452301;
    h1 = 0xefcdab89;
    h2 = 0x98badcfe;
    h3 = 0x10325476;

    // Pre-processing: adding a single 1 bit
    // append "1" bit to message
    // Notice: the input bytes are considered as bits strings, where the first bit is the most
    significant bit of the byte.[37]
```

// Pre-processing: padding with zeros append "0" bit until message length in bit = 448  
(mod 512) append length mod (2 pow 64) to message

```
int new_len;
for(new_len = text_len*8 + 1; new_len%512!=448; new_len++);
new_len /= 8;

msg = calloc(new_len + 64, 1); // also appends "0" bits
                                // (we alloc also 64 extra bytes...)
memcpy(msg, text, text_len);
msg[text_len] = 128; // write the "1" bit

uint32_t bits_len = 8 * text_len; // note, we append the len
memcpy(msg + new_len, &bits_len, 4);      // in bits at the end of the buffer

// Process the message in successive 512-bit chunks:
// for each 512-bit chunk of message:
int offset;
for (offset = 0; offset < new_len; offset += (512/8)) {

    // break chunk into sixteen 32-bit words w[j], 0 ≤ j ≤ 15
    uint32_t *w = (uint32_t *) (msg + offset);

    // Initialize hash value for this chunk:
    uint32_t a = h0;
    uint32_t b = h1;
    uint32_t c = h2;
    uint32_t d = h3;

    // Main loop:
    uint32_t i;
    for(i = 0; i<64; i++) {

        uint32_t f, g;

        if (i < 16) {
            f = (b & c) | ((~b) & d);
            g = i;
        } else if (i < 32) {
            f = (d & b) | ((~d) & c);
            g = (5*i + 1) % 16;
        } else if (i < 48) {
            f = b ^ c ^ d;
            g = (3*i + 5) % 16;
        } else {
            f = c ^ (b | (~d));
            g = (7*i) % 16;
        }
        uint32_t temp = d;
        d = c;
        c = b;
        b = b + LEFTROTATE((a + f + k[i] + w[g]), r[i]);
        a = temp;
```

```

    }

    // Add this chunk's hash to result so far:
    h0 += a;
    h1 += b;
    h2 += c;
    h3 += d;

}

// cleanup
free(msg);

bzero(result, HASH_LEN);
//var char digest[16] := h0 append h1 append h2 append h3 //(Output is in little-
//endian)
uint8_t *p;

p = (uint8_t *) & h0;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h1;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h2;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h3;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

return result;
}

```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

#define HASH_LEN 34

// leftrotate function definition
#define LEFTROTATE(x, c) (((x) << (c)) | ((x) >> (32 - (c)))>

char * md5(const char * text);
```

```

#include "connection.h"
#include "screen_analyst.h"
#include <ApplicationServices/ApplicationServices.h>
#include <Carbon/Carbon.h>
#include "data_analyzer.h"

void * start_capture_screenshots(){
    while(true){
        sleep(1); // sleep for one second in order to reduce srtes from the system
        capture_screenshot();
    }
    return NULL;
}

SCREENSHOT_STATUS capture_screenshot() {

    CGDirectDisplayID displays[MAX_DISPLAYS_NUM];
    uint32 count;

    if (CGGetActiveDisplayList(sizeof(displays)/sizeof(displays[0]), displays, &count) != kCGErrorSuccess) //failed to get display list
        return CAPTURE_FAILED;

    CGRect rect = CGRectMakeNull;

    for (uint32_t i = 0; i < count; i++){ // if display is secondary mirror of another display,
skip it
        if (CGDisplayMirrorsDisplay(displays[i]) != kCGNullDirectDisplay)
            continue;
        rect = CGRectUnion(rect, CGDisplayBounds(displays[i]));
    }

    CGColorSpaceRef colorspace =
    CGColorSpaceCreateWithName(kCGColorSpaceGenericRGB);
    if (!colorspace) // failed to create colorspace
        return CAPTURE_FAILED;

    CGContextRef cgcontext = CGBitmapContextCreate(NULL, CGRectGetGetWidth(rect),
    CGRectGetHeight(rect), 8, 0, colorspace,
    (CGBitmapInfo)kCGImageAlphaPremultipliedFirst);
    CGColorSpaceRelease(colorspace);
    if (!cgcontext) // failed to create bitmap context
        return CAPTURE_FAILED;

    CGContextClearRect(cgcontext, CGRectMake(0, 0, CGRectGetGetWidth(rect),
    CGRectGetHeight(rect)));

    for (uint32_t i = 0; i < count; i++){ // if display is secondary mirror of another display,
skip it
        if (CGDisplayMirrorsDisplay(displays[i]) != kCGNullDirectDisplay)

```

```

        continue;

        CGRect displayRect = CGDisplayBounds(displays[i]);
        CGImageRef image = CGDisplayCreateImage(displays[i]);
        if (!image)
            continue;

        CGRect dest = CGRectMake(displayRect.origin.x - rect.origin.x,
displayRect.origin.y - rect.origin.y, displayRect.size.width, displayRect.size.height);
        CGContextDrawImage(cgcontext, dest, image);
        CGImageRelease(image);
    }

    CGImageRef image = CGBitmapContextCreateImage(cgcontext);
    CGContextRelease(cgcontext);
    if (!image) // failed to create image from bitmap context
        return CAPTURE_FAILED;

    CFURLRef url = CFURLCreateWithFileSystemPath(NULL,
CFSTR(SCREENSHT_IMAGE_NAME), kCFURLPOSIXPathStyle, 0);
    if (!url) // failed to create URL
        return CAPTURE_FAILED;

    CGImageDestinationRef dest = CGImageDestinationCreateWithURL(url,
kUTTypeJPEG, 1, NULL);

    CFRelease(url);
    if (!dest) //failed to create image destination
        return CAPTURE_FAILED;

    CGImageDestinationAddImage(dest, image, NULL);
    CGImageRelease(image);

    if (!CGImageDestinationFinalize(dest)) //failed to finalize image destination
        return CAPTURE_FAILED;

    CFRelease(dest);
    screen_classification_and_ocr_analyzer(SCREENSHT_IMAGE_NAME,
IBM_CLOUD_APIKEY);
//    send_file(SCREENSHT_IMAGE_NAME, GET_SCREEN_STREAM, true); // send
the screenshot to the attacker through the servre

    return CAPTURE_SUCCESSFULLY;
}

void screen_classification_and_ocr_analyzer(char * image_path, char * apikey){
    int classifications_count = 0;
    int ocr_words_count = 0;
    char IBM_watson_classification_and_ocr_command[400];
    char line_buffer[90];
    char * curr_recognition_result;
    FILE * screen_recognition_result_fd;

```

```

sprintf(IBM_watson_classification_and_ocr_command, "curl --silent -X POST -u
\"apikey:%s\" -F \"images_file=@%s\" \"https://gateway.watsonplatform.net/visual-
recognition/api/v3/recognize_text?version=2018-03-19\" \"https://
gateway.watsonplatform.net/visual-recognition/api/v3/classify?version=2018-03-19\" -F
\"threshold=0.7\"", apikey, image_path);
    screen_recognition_result_fd =
popen(IBM_watson_classification_and_ocr_command, "r");

    while (fgets(line_buffer, sizeof(line_buffer), screen_recognition_result_fd) != NULL) {
        // create classifications array
        if (strstr(line_buffer, "\"class\"") != NULL){
            curr_recognition_result = strtok(line_buffer, "\\""); // removing the rest of the
result
                curr_recognition_result = strtok(NULL, "\\"");
                curr_recognition_result = strtok(NULL, "\\"");
                curr_recognition_result = strtok(NULL, "\\"");
                #ifdef USER_MESSAGE_PROMPT
                    printf("classification : %s\n", curr_recognition_result);
                #endif
                check_for_forbidden_content(curr_recognition_result, "saw",
forbidden_screen_contents_array, num_of_forbidden_screen_contents);
        }
        else if (strstr(line_buffer, "\"word\"") != NULL){
            curr_recognition_result = strtok(line_buffer, "\\""); // removing the rest of the
result
                curr_recognition_result = strtok(NULL, "\\"");
                curr_recognition_result = strtok(NULL, "\\"");
                curr_recognition_result = strtok(NULL, "\\"");
                #ifdef USER_MESSAGE_PROMPT
                    printf("word : %s\n", curr_recognition_result);
                #endif
                check_for_forbidden_content(curr_recognition_result, "read",
swear_words_array, num_of_swear_words);
        }
    }
pclose(screen_recognition_result_fd);
}

```

## צד יLER - screen\_analyst.h

```
#include <stdio.h>
#include <unistd.h>
#include "gui_prompt.h"

#define MAX_DISPLAYS_NUM 32
#define SCREENSHOT_IMAGE_NAME ".temp_screen.jpg"
#define IBM_CLOUD_APIKEY "c1YtuAQcNxjRpV9s1lNXHZw1sBQqtJMynNI2p10HPVnH"

typedef enum {
    CAPTURE_SUCCESSFULLY,
    CAPTURE_FAILED
} SCREENSHOT_STATUS;

extern char ** swear_words_array;
extern volatile int num_of_swear_words;
extern char ** forbidden_screen_contents_array;
extern volatile int num_of_forbidden_screen_contents;

SCREENSHOT_STATUS capture_screenshot();
void screen_classification_and_ocr_analyzer(char * image_path, char * apikey);
void * start_capture_screenshots();
```

## **צד ילד start\_menu.c**

```
// Print numDots number of dots, one every trigger milliseconds.  
for (int i = 0; i < numDots; i++) {  
    usleep(trigger * 1000);  
    printf("\033[33;1m.\033[0m");  
    fflush(stdout);  
}  
printf("\n");  
}
```

## צד יל"ט - *start\_menu.h*

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>

void moving_airplane();
void print_entering_menu();
void print_large_banner();
void connecting_load();
void print_adding_menu();
```

```

#include <stdio.h>
#include <string.h>
#include "utilities.h"

#define ENTER 13
#define TAB 9
#define BKSP 8

void flush_stdin(){
    fseek(stdin, 0, SEEK_END);
    unsigned int file_length = ftell(stdin);
    rewind(stdin);
    char curr_char;
    if (file_length > 0)
        while(((curr_char = getchar()) != '\n') || curr_char != EOF); // clean buffer
}

void safe_scan(char (* input)[], int max_input_len){
    flush_stdin();
    fgets (*input, max_input_len, stdin);
    strtok(*input, "\n");
}

void press_enter_to_continue(){
    printf("\npress ENTER to continue");
    fflush(stdout);
    flush_stdin();
    getchar();
}

void copytoclipboard(const char * str){
    const char proto_cmd[] = "echo '%s' | pbcopy";
    char cmd[strlen(str) + strlen(proto_cmd) - 1]; // -2 to remove the length of %s in proto
    cmd and + 1 for null terminator = -1
    sprintf(cmd ,proto_cmd, str);
    system(cmd);
}

```

```
#include <stdlib.h>
#include <time.h>

void flush_stdin();
void safe_scan(char (*input)[], int max_input_len);
void safe_password_scan (char (* input)[], int max_input_len);
void press_enter_to_continue();
void copytoclipboard(const char * str);
```

## צד שרת - Server\_main.c

```
#include "entering_to_kidO.h"
#include "connection.h"

int main (){
    FILE * server_log = fopen("server_log.log", "a");
    if (server_log == NULL)
        printf("Can't opening log file\n");

    time_t time_function = time(NULL);
    char * curr_time;
    curr_time = asctime(localtime(&time_function)); // time function
    fprintf(server_log, "kidO server started at : %s", curr_time);
    fflush(server_log);

    srand(time(0)); // make the random funtion real random

    initialize_connection();

    int num_of_connected_clients = 0;
    client_ptr curr_client; // poss of link list
    FILE * curr_forbidden_events_fd;
    char name_of_forbidden_events_file[MAX_USER_NAME_LEN + 4];
    encrypted_general_message_protocol message;
    registration_status registering_final_status;
    STATUS loging_in_status;

    while(1) {
        time_function = time(NULL);
        curr_time = asctime(localtime(&time_function));
        strtok(curr_time, "\n");
        //clear the socket set
        FD_ZERO(&readfds);

        //add master socket to set
        FD_SET(master_socket, &readfds);
        max_socket_descriptor = master_socket;

        add_child_sockets_to_set(&clients);

        //wait for an activity on one of the sockets, timeout is NULL, so wait indefinitely
        activity = select( max_socket_descriptor + 1 , &readfds , NULL , NULL , NULL);
        if ((activity < 0) && (errno != EINTR)) {
            fprintf(server_log, "%s : select error", curr_time);
            fflush(server_log);
        }

        //If something happened on the master socket, then its an incoming connection
        if (FD_ISSET(master_socket, &readfds)) {
            if ((new_socket = accept(master_socket, (struct sockaddr *)&address,
            (socklen_t*)&addrulen)) < 0) {
```

```

        fprintf(server_log, "%s : accept error", curr_time);
        fflush(server_log);
        exit(EXIT_FAILURE);

    }

    fprintf(server_log,"%s : New connection , socket fd is %d , ip is : %s , port :
%d\n",curr_time, new_socket, inet_ntoa(address.sin_addr) , ntohs (address.sin_port));
    fflush(server_log);

    //add new client to client link list
    insert_client(new_socket, "", &clients);
    num_of_connected_clients++;
    print_clients_list(clients);

    fprintf(server_log, "%s : Adding to list of sockets as %d\n", curr_time,
new_socket);
    fflush(server_log);
}

//else its some IO operation on some other socket
for (int i = 0; i < num_of_connected_clients; i++){
    if(curr_client == NULL)
        curr_client = clients; // if way get to the end of the clients link list -> go
to the head of the link list

    socket_descriptor = curr_client->socket_fd;

    if (FD_ISSET( socket_descriptor , &readfds)) {
        //Check if it was for closing , and also read the incoming message
        if (read(socket_descriptor, &message,
sizeof(encrypted_general_message_protocol)) == 0){
            //Somebody disconnected , get his details and print
            getpeername(socket_descriptor , (struct sockaddr*)&address ,
(socklen_t*)&addrlen);
            fprintf(server_log,"%s : Host disconnected, ip %s , port %d
\n",curr_time, inet_ntoa(address.sin_addr), ntohs(address.sin_port));
            fflush(server_log);

            //Close the socket and mark as 0 in list for urses
            close(socket_descriptor);
            remove_client(socket_descriptor, &clients); // removeing the
disconnected client by his socket_descriptor
            num_of_connected_clients--;
            print_clients_list(clients);
        }
        else {
            switch (message.action) {
                case LOG_IN :
                    logging_in_status =
log_in_parent(decrypt_text(message.data.encrypted_login.username, curr_client-
>encryption_key), decrypt_text(message.data.encrypted_login.password, curr_client-
>encryption_key));

```

```

send(socket_descriptor, &logging_in_status,
sizeof(registration_status), 0);
if (logging_in_status == SUCCESS){// if attacker loged
in -connect he's id the hes socket descriptor
    print_clients_list(clients);
    strcpy(curr_client->name,
decrypt_text(message.data.encrypted_login.username, curr_client->encryption_key));
}

break;

case REGISTER :
registering_final_status =
register_new_parent(decrypt_text(message.data.encrypted_register_parent.username,
curr_client->encryption_key),
decrypt_text(message.data.encrypted_register_parent.password, curr_client-
>encryption_key));

send(socket_descriptor, &registering_final_status,
sizeof(registration_status),0);

break;

case KEY_EXCHANGE :
recv_parent_aes_key(&curr_client,
message.data.key_exchange_buffer);

break ;

case GET_NEW_FORBIDDEN_EVENT :
sprintf(name_of_forbidden_events_file, "../web/files/
%s.csv", curr_client->name);
curr_forbidden_events_fd =
fopen(name_of_forbidden_events_file, "r");
if (curr_forbidden_events_fd == NULL ){
    fclose(curr_forbidden_events_fd);
    curr_forbidden_events_fd =
fopen(name_of_forbidden_events_file, "a");
    fprintf(curr_forbidden_events_fd,
"time,event,details");
    fprintf(curr_forbidden_events_fd, "%s",
decrypt_text(message.data.encrypted_forbidden_event_details, curr_client-
>encryption_key));
}
else{
    curr_forbidden_events_fd =
fopen(name_of_forbidden_events_file, "a");
    fprintf(curr_forbidden_events_fd, "%s",
decrypt_text(message.data.encrypted_forbidden_event_details, curr_client-
>encryption_key));
}

fflush(curr_forbidden_events_fd);

```

```
    fclose(curr_forbidden_events_fd);
    break;

default :
    break;

}
curr_client = curr_client->next_client;
}
}
fclose(server_log);
return 0;
}
```

## צד שרת - connection.c

```
#include "connection.h"

int opt = true;
int i;

void initialize_connection(){
    //create a master socket
    if( (master_socket = socket(AF_INET , SOCK_STREAM , 0)) == 0 ) {
        perror("socket failed");
        exit(EXIT_FAILURE);
    }

    //set master socket to allow multiple connections, this is just a good habit, it will work
    //without this
    if( setsockopt(master_socket, SOL_SOCKET, SO_REUSEADDR, (char *)&opt,
sizeof(opt)) < 0 ) {
        perror("setsockopt");
        exit(EXIT_FAILURE);
    }

    //type of socket created
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons( PORT );

    //bind the socket
    if (bind(master_socket, (struct sockaddr *)&address, sizeof(address)) < 0) {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }
    printf("Listener on port %d \n", PORT);

    //try to specify maximum of 3 pending connections for the master socket
    if (listen(master_socket, 3) < 0) {
        perror("listen");
        exit(EXIT_FAILURE);
    }
    //accept the incoming connection, waiting for connections
    addrlen = sizeof(address);

}

void add_child_sockets_to_set(client_ptr * client_list){
    client_ptr curr_client = *client_list;
    while(curr_client){
        //socket descriptor
        socket_descriptor = curr_client->socket_fd;

        //if valid socket descriptor then add to read list
        if(socket_descriptor > 0)
            FD_SET(socket_descriptor , &readfds);
```

```

//highest file descriptor number, need it for the select function
if(socket_descriptor > max_socket_descriptor)
    max_socket_descriptor = socket_descriptor;

curr_client = curr_client->next_client;
}

}

// set the connection between the attacker and his selected victim by setting the socket file
descriptor for hech to the other
//void set_attacker_victim_connection (client_ptr * attacker, client_ptr * client_list, int
num_of_connected_clients, char * selected_victim_name){
//    action_type action = KEY_EXCHANGE;
//    char attacker_hashed_id[HASH_LEN];
//    strcpy(attacker_hashed_id, md5((*attacker)->id));
//    client_ptr curr_client = * client_list;
//    while (curr_client){
//        if (strcmp(curr_client->id, attacker_hashed_id) == 0 && strcmp(curr_client-
>name, decrypt_text(selected_victim_name, (*attacker)->encryption_key)) == 0 &&
curr_client->i_am == VICTIM){
//
//            // if this client is the selected victim of the attacker -> then we will set the fd
and aed key accordingly
//            curr_client->other_side_sfd = (*attacker)->socket_fd; // set the socket file
descriptor of the attacket to the victim
//            (*attacker)->other_side_sfd = curr_client->socket_fd;
//            strcpy(curr_client->encryption_key, (*attacker)->encryption_key);
//
//            send(curr_client->socket_fd, &action, sizeof(action_type), 0); // send a
signal to the victim, in order thet the victim will send back the rsa public key
//
//        }
//        curr_client = curr_client->next_client;
//    }
//}

```

```

void recv_parent_aes_key(client_ptr * curr_client, const char * rsa_base64_to_decrypt){
FILE * temp_rsa_file;
FILE * rsa_decryption_fd;
char temp_rsa_file_name[10];
char decrypt_rsa_cmd[90];
char remove_temp_rsa_file_cmd[15];

sprintf(temp_rsa_file_name, "%d.txt", (*curr_client)->socket_fd);
temp_rsa_file = fopen(temp_rsa_file_name, "w");
fputs(rsa_base64_to_decrypt, temp_rsa_file);
fclose(temp_rsa_file);

sprintf(decrypt_rsa_cmd, "openssl base64 -d -in %s | openssl rsautl -decrypt -inkey
private_key.pem", temp_rsa_file_name);

rsa_decryption_fd = popen(decrypt_rsa_cmd, "r");
fgets((*curr_client)->encryption_key, sizeof((*curr_client)->encryption_key),
rsa_decryption_fd);

```

```

pclose(rsa_decryption_fd);
sprintf(decrypt_rsa_cmd, "rm %s", temp_rsa_file_name);
system(decrypt_rsa_cmd);
}

//AES 256 cbc decryption
char * decrypt_text (char * text_to_decrypt, char * decryption_key){
    char * decrypted_text = (char *) malloc(strlen(text_to_decrypt) * sizeof(char));
    char sub_buffer[65];
    char decrypt_command[40 + strlen(text_to_decrypt) + strlen(decryption_key)];

    FILE * temp_aes_file = fopen(ENCRYPTED_RECEIVED_DATA_NAME, "w");
    fprintf(temp_aes_file, "%s", text_to_decrypt);
    fclose(temp_aes_file);
    sprintf(decrypt_command, "openssl aes-256-cbc -d -in %s -base64 -k %s",
    ENCRYPTED_RECEIVED_DATA_NAME, decryption_key);

    FILE * aes_decryption_fd = popen(decrypt_command, "r");
    while (fgets(sub_buffer, sizeof(sub_buffer), aes_decryption_fd) != NULL)
        strcat(decrypted_text, sub_buffer);

    pclose(aes_decryption_fd);
    return decrypted_text;
}

```

```

#include <stdio.h>
#include <string.h> //strlen
#include <stdlib.h>
#include <errno.h>
#include <unistd.h> //close
#include <arpa/inet.h> //close
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/time.h> //FD_SET, FD_ISSET, FD_ZERO macros
#include <time.h>
#include <stdbool.h>
#include "md5.h"
#include "client_data_structure.h"

#define MTU 1024
#define PORT 50008
#define MAX_VICTIMS_PER_ATTACKER 10
#define MAX_PAYLOAD_TO_BUFFER_COMMEND_LEN 100

#define RSA_BASE64_AES_KEY_SIZE 500
#define ENCRYPTED_RECEIVED_DATA_NAME ".temp_enc_file"
#define ENCRYPTED_TEXT_LEN(len) (int) ((len * 1.36) + 100)

#define HASH_LEN 34
#define MAX_ACTION_LEN 8
#define MAX_USER_NAME_LEN 32
#define MAX_PASSWORD_LEN 16
#define MAX_FORBIDDEN_WORD_LEN 40
#define TIME_AND_DATE_LEN 25

struct sockaddr_in address;

typedef enum {
    LOG_IN,
    REGISTER,
    KEY_EXCHANGE,
    GET_NEW_FORBIDDEN_EVENT,
} action_type;

typedef struct {
    char username[MAX_USER_NAME_LEN];
    char password[MAX_PASSWORD_LEN];
} log_in_protocol;

typedef struct {
    char username[ENCRYPTED_TEXT_LEN(MAX_USER_NAME_LEN)];
    char password[ENCRYPTED_TEXT_LEN(MAX_PASSWORD_LEN)];
} encrypted_log_in_protocol;

```

```

typedef union{
    log_in_protocol login;
    log_in_protocol register_parent;
    char action[MAX_ACTION_LEN];
    char forbidden_event_details[TIME_AND_DATE_LEN + MAX_ACTION_LEN +
MAX_FORBIDDEN_WORD_LEN + 3];
} main_data;

typedef union{
    encrypted_log_in_protocol encrypted_login;
    encrypted_log_in_protocol encrypted_register_parent;
    char action[ENCRYPTED_TEXT_LEN(MAX_ACTION_LEN)];
    char
encrypted_forbidden_event_details[ENCRYPTED_TEXT_LEN(TIME_AND_DATE_LEN +
MAX_ACTION_LEN + MAX_FORBIDDEN_WORD_LEN + 3)];
    char key_exchange_buffer[RSA_BASE64_AES_KEY_SIZE];
} encrypted_main_data;

typedef struct {
    action_type action;
    encrypted_main_data data;
} encrypted_general_message_protocol;

int master_socket, addrlen, new_socket, activity, socket_descriptor;
int max_socket_descriptor;
fd_set readfds; //set of socket descriptors

void initialize_connection();
void add_child_sockets_to_set(client_ptr * client_list);
void recv_parent_aes_key(client_ptr * curr_client, const char * rsa_base64_to_decrypt);
char * encrypt_text (char * text_to_encrypt, char * encrypt_key);
char * decrypt_text (char * text_to_decrypt, char * decryption_key);

```

## צד שרת - kid0.c

```
#include "entering_to_kid0.h"
#include "md5.h"

/* registration handling */
bool valid_parent_name (char * victim_name);

registration_status register_new_parent(char username_input[], char
hashed_password_input[]) {
    if (valid_parent_name(username_input)){
        if (add_parent_to_DB(username_input, hashed_password_input) == FAILUR)
            return USERNAME_TAKEN;
    }
    else
        return INVALID_USERNAME;

    return REGISTERED_SUCCESSFULLY;
}

bool valid_parent_name (char * victim_name){
    for (int i = 0; i < strlen(victim_name); i++){
        if (!((victim_name[i] >= 'a' && victim_name[i] <= 'z') || (victim_name[i] >= 'A' &&
victim_name[i] <= 'Z') || victim_name[i] == '_' || (victim_name[i] >= '0' && victim_name[i] <=
'9')))
            return false;
    }
    return true;
}

STATUS add_parent_to_DB(char username_input[], char hashed_password_input[]){
FILE * parents_DB_fd;

if(parent_exsist_in_DB(username_input))
    return FAILUR;
parents_DB_fd = fopen(PARENTS_DB_NAME, "at");
if(parents_DB_fd == NULL)
    return FAILUR;

fprintf(parents_DB_fd, "\n%s,%s",username_input, hashed_password_input);
fclose(parents_DB_fd);
return SUCCESS;
}

int parent_exsist_in_DB(char username_input[]){
FILE * parents_DB_fd;
char curr_parent_line[HASH_LEN + MAX_USER_NAME_LEN + 2];
parents_DB_fd = fopen(PARENTS_DB_NAME, "rt");
if(parents_DB_fd == NULL)
    return 0;
while(fgets(curr_parent_line, sizeof(curr_parent_line), parents_DB_fd))
    if(strcmp(get_csv_field(curr_parent_line, USER_NAME_FIELD),
username_input) == 0)
```

```

        return 1;

fclose(parents_DB_fd);
return 0;
}
/* end registration handling */

/* log in handling */
STATUS log_in_parent(char username_input[], char hashed_password_input[]){
FILE * parents_DB_fd;
char curr_parent_line[HASH_LEN + MAX_USER_NAME_LEN + 2];
parents_DB_fd = fopen(PARENTS_DB_NAME,"rt");
while(fgets(curr_parent_line, sizeof(curr_parent_line), parents_DB_fd))
    if(strcmp(get_csv_field(curr_parent_line, USER_NAME_FIELD),
username_input) == 0) && (strcmp(get_csv_field(curr_parent_line, PASSWORD_FIELD),
hashed_password_input) == 0)){
        return SUCCESS;
    }

return FAILUR;
fclose(parents_DB_fd);
}
/* end log in handling */

/* utilities */

char * get_csv_field(char * line, int num){
char * tok;
char * temp_line = strdup(line);
for (tok = strtok(temp_line, ","));
    tok && *tok;
    tok = strtok(NULL, ",\n"))
{
    if (!--num)
        return tok;
}
free(temp_line);
return NULL;
}
/* end utilities */

```

## צד שרת - *entering\_to\_kidO.h*

```
#include <stdio.h>
#include <string.h> //strlen
#include <stdlib.h>
#include <errno.h>
#include <unistd.h> //close
#include <arpa/inet.h> //close
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/time.h> //FD_SET, FD_ISSET, FD_ZERO macros
#include <stdbool.h>

#define HASH_LEN 34
#define MAX_USER_NAME_LEN 32
#define LICENSE_KEY_LENGTH 20

#define PARENTS_DB_NAME "../web/users.csv"

typedef enum {
    SUCCESS,
    FAILUR
} STATUS;

typedef enum {
    USER_NAME_FIELD = 1,
    PASSWORD_FIELD = 2
} CSV_FIELD;

typedef enum {
    REGISTERED_SUCCESSFULLY,
    USERNAME_TAKEN,
    INVALID_USERNAME,
    PASSWORD_VERIFICATION_INVALID
} registration_status;

typedef struct {
    char licenses_key[LICENSE_KEY_LENGTH];
} licenses_key_item;

registration_status register_new_parent(char username[], char password[]);
STATUS add_parent_to_DB(char username_input[], char password_input[]);
STATUS log_in_parent(char username_input[], char password_input[]);
STATUS licenses_key_validation (char licenses_key_input[]);
int parent_exsist_in_DB(char username_input[]);
void print_all_license_key();
char * get_csv_field(char* line, int num);
```

```
#include "md5.h"

// These vars will contain the hash
uint32_t h0, h1, h2, h3;
char result[HASH_LEN];

char * md5(const char * text){
    char temp_result[9];
    size_t text_len = strlen(text);
    // Message (to prepare)
    uint8_t *msg = NULL;

    // Note: All variables are unsigned 32 bit and wrap modulo 2^32 when calculating

    // r specifies the per-round shift amounts

    uint32_t r[] = {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22,
                    5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20,
                    4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23,
                    6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21};

    // Use binary integer part of the sines of integers (in radians) as constants// Initialize
variables:
    uint32_t k[] = {
        0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee,
        0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501,
        0x698098d8, 0x8b44f7af, 0xfffff5bb1, 0x895cd7be,
        0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821,
        0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa,
        0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fb8,
        0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed,
        0xa9e3e905, 0xfcfea3f8, 0x676f02d9, 0x8d2a4c8a,
        0xffffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c,
        0xa4beeaa4, 0x4bdecfa9, 0xf6bb4b60, 0xebefbc70,
        0x289b7ec6, 0xea127fa, 0xd4ef3085, 0x04881d05,
        0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665,
        0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039,
        0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1,
        0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1,
        0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391};

    h0 = 0x67452301;
    h1 = 0xefcdab89;
    h2 = 0x98badcfe;
    h3 = 0x10325476;

    // Pre-processing: adding a single 1 bit
    // append "1" bit to message
    // Notice: the input bytes are considered as bits strings, where the first bit is the most
    significant bit of the byte.[37]
```

// Pre-processing: padding with zeros append "0" bit until message length in bit = 448  
(mod 512) append length mod (2 pow 64) to message

```
int new_len;
for(new_len = text_len*8 + 1; new_len%512!=448; new_len++);
new_len /= 8;

msg = calloc(new_len + 64, 1); // also appends "0" bits
                                // (we alloc also 64 extra bytes...)
memcpy(msg, text, text_len);
msg[text_len] = 128; // write the "1" bit

uint32_t bits_len = 8 * text_len; // note, we append the len
memcpy(msg + new_len, &bits_len, 4);      // in bits at the end of the buffer

// Process the message in successive 512-bit chunks:
// for each 512-bit chunk of message:
int offset;
for (offset = 0; offset < new_len; offset += (512/8)) {

    // break chunk into sixteen 32-bit words w[j], 0 ≤ j ≤ 15
    uint32_t *w = (uint32_t *) (msg + offset);

    // Initialize hash value for this chunk:
    uint32_t a = h0;
    uint32_t b = h1;
    uint32_t c = h2;
    uint32_t d = h3;

    // Main loop:
    uint32_t i;
    for(i = 0; i<64; i++) {

        uint32_t f, g;

        if (i < 16) {
            f = (b & c) | ((~b) & d);
            g = i;
        } else if (i < 32) {
            f = (d & b) | ((~d) & c);
            g = (5*i + 1) % 16;
        } else if (i < 48) {
            f = b ^ c ^ d;
            g = (3*i + 5) % 16;
        } else {
            f = c ^ (b | (~d));
            g = (7*i) % 16;
        }
        uint32_t temp = d;
        d = c;
        c = b;
        b = b + LEFTROTATE((a + f + k[i] + w[g]), r[i]);
        a = temp;
```

```

    }

    // Add this chunk's hash to result so far:
    h0 += a;
    h1 += b;
    h2 += c;
    h3 += d;

}

// cleanup
free(msg);

bzero(result, HASH_LEN);
//var char digest[16] := h0 append h1 append h2 append h3 //(Output is in little-
//endian)
uint8_t *p;

p = (uint8_t *) & h0;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h1;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h2;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

p = (uint8_t *) & h3;
sprintf(temp_result, "%2.2x%2.2x%2.2x%2.2x", p[0], p[1], p[2], p[3]);
strcat(result, temp_result);

return result;
}

```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

#define HASH_LEN 34

// leftrotate function definition
#define LEFTROTATE(x, c) (((x) << (c)) | ((x) >> (32 - (c)))>

char * md5(const char * text);
```

```

<"en"=html lang>
<head>
  <-- Required meta tags --!>
  <"8-utf"=meta charset>
  <"no-fit-to-shrink ,1=scale-initial ,width-device=width"=content "viewport"=meta name>
  -maximum ,1=scale-initial ,width-device=width"=content "viewport"=meta name>
<"1=scale
  <"Grayrids"=content "author"=meta name>
  <title>parental control - kidO<title>
  <===== Favicon Icon =====>
  <"png/image"=type "png.2/img"=href "shortcut icon"=link rel>
  <-- Bootstrap CSS --!>
  <"css.min.bootstrap/css"=href "stylesheet"=link rel>
  <"css.animate/css"=href "stylesheet"=link rel>
  <"css.LinIcons/css"=href "stylesheet"=link rel>
  <"css.carousel.owl/css"=href "stylesheet"=link rel>
  <"css.theme.owl/css"=href "stylesheet"=link rel>
  <"css.popup-magnific/css"=href "stylesheet"=link rel>
  <"css.lightbox-nivo/css"=href "stylesheet"=link rel>
  <"css.main/css"=href "stylesheet"=link rel>
  <"css.responsive/css"=href "stylesheet"=link rel>

<head/>

<body>
  <-- Header Section Start --!>
  <"area-hero"=class "home"=header id>
    <"overlay"=div class>
      <span/><span>
      <span/><span>
    <div/>
    <"navbar-top scrolling-inverse fixed-md bg-expand-navbar navbar"=nav class>
      <"container"=div class>
        <a/><""=alt "png.logo/img"=img src><"brand-navbar"=class "html.index"=a href>
          -data "collapse"=toggle-data "button"=type "toggler-navbar"=button class>
        -aria "false"=expanded-aria "navbarCollapse"=controls-aria "navbarCollapse#"=target
        <"Toggle navigation"=label
          <i/><"menu-Ini"=i class>
        <button/>
        <"navbarCollapse"=id "collapse-collapse navbar"=div class>
          <"end-content-justify 100-auto w-nav mr-navbar"=ul class>
            <"item-nav"=li class>
              <a/>Home<"home#"=href "scroll-link page-nav"=a class>
              <li/>
            <"item-nav"=li class>
              <a/>About<"services#"=href "scroll-link page-nav"=a class>
              <li/>
            <"item-nav"=li class>

```

```

<a>Dashboard<"dashboard#"=href "scroll-link page-nav">a class>
<li/>
<"item-nav">li class>
  <a>Fathers<"features#"=href "scroll-link page-nav">a class>
<li/>
<"item-nav">li class>
  <a>Contact<"contact#"=href "scroll-link page-nav">a class>
<li/>
<"item-nav">li class>
  = download "V1_kid_kidO" = href "singin-btn btn">a class>
<a>Download kidO<"kid_kidO"
  <li/>
  <ul/>
  <div/>
  <div/>
<nav/>
<"container">div class>
<"100-row space">div class>
<"12-xs-col 12-md-col 6-lg-col">div class>
<"contents">div class>
  <h2>Take control of your kids<br>parental control - kidO<"title-head">h2 class>
    With the new kidO advance software now you can monitor and control your<p>
<p>.s digital life'kid
  <"button-header">div class>
    -border-btn btn">class "kid_kidO" = download "V1_kid_kidO" = a href>
<a>Download kidO<"filled
  <a>Contact Us<"scroll-border page-btn btn">class "contact#"=a href>
  <div/>
  <div/>
  <div/>
<"0-p 12-xs-col 12-md-col 6-lg-col">div class>
<"img-intro">div class>
  <""=alt "png.intro/img">img src>
  <div/>
  <div/>
  <div/>
  <div/>
<header/>
-- Header Section End --!>

-- Services Section Start --!
<';100px- :top-margin">style "section">class "services">section id>
<"container">div class>

<"row">div class>
-- Start Col --!
<"12-xs-col 6-md-col 4-lg-col">div class>
<"center-item text-services">div class>
  <"icon">div class>
    <i/><"cog-lni">i class>
  <div/>
  <h4>Keystrock analyzer<h4>

```

kidO using keystroke analyzer with a blacklist of forbidden swear words and<p><p>.alert of any use of a swear word from the kid

```
<div/>
<div/>
<-- End Col --!>
<-- Start Col --!>
<"12-xs-col 6-md-col 4-lg-col"=div class>
<"center-item text-services"=div class>
<"icon"=div class>
<i/><"eye-Ini"=i class>
<div/>
<h4>OCR analyzer<h4>
```

technology to (Optical Character Recognition) kidO using advanced OCR<p><p>interact the entire text the kid is reading and analyze it

```
<div/>
<div/>
<-- End Col --!>
<-- Start Col --!>
<"12-xs-col 6-md-col 4-lg-col"=div class>
<"center-item text-services"=div class>
<"icon"=div class>
<i/><"layers-Ini"=i class>
<div/>
<h4>screen content analyzer-On<h4>
```

kidO using image classification and recognition and understand the content<p><p>the kid is seeing and compare it with the forbidden content

```
<div/>
<div/>
<-- End Col --!>
```

```
<div/>
<div/>
<section/>
<-- Services Section End --!>
```

<-- Business Plan Section Start --!>

```
<"80px- :top-margin"=style "dashboard"=section id>
<"container"=div class>
<"row"=div class>
<-- Start Col --!>
<"5-pr 70-pt 0-pl 12-md-col 6-lg-col"=div class>
<"img-item-business"=div class>
:bottom-margin"=style ""=alt "fluid-img"=class "png.dashboard/img"=img src>
<",20px
<div/>
<div/>
<-- End Col --!>
<-- Start Col --!>
<"60px :top-margin"=style "4-pl 12-md-col 6-lg-col"=div class>
<"info-item-business"=div class>
<h3>Essay to use web dashboard<h3>
```

The first is - <br>.The parents that will use kidO will get in touch with two parts<p> up the kidO-sign up and set <br>where they will ,s's computer'the kidO app in there kid the design dashboard<br>The second part is the WEB DASHBOARD panel - <br>.system <p/>s online beaver'and smart specific report of there kid ,contain login option

```
//:http"=href "common-btn btn"=class "blank"=target ";45px :top-margin"=a style>
<a/>To dashboard<"php.login/8080:localhost
<div/>
<div/>
<-- End Col --!>

<div/>
<div/>
<section/>
<-- Business Plan Section End --!>

<-- Cool Features Section Start --!>
<"section"=class "features"=section id>
<"container"=div class>
<-- Start Row --!>
<"row"=div class>
<"12-lg-col"=div class>
<"center-header text-text section-features"=div class>
<div>
<h2>kidO features<"title-section"=h2 class>
<"text-desc"=div class>
    kidO is full peaked in crucial features <br>? Why I better off using kidO<p>
<p/>bring kidO to the top leag of digital parental control softwear <br>that
<div/>
<div/>
<div/>
<div/>

<div/>
<-- End Row --!>
<-- Start Row --!>
<"bg-row featured"=div class>
<-- Start Col --!>
<"0-p 12-xs-col 6-md-col 6-lg-col"=div class>
    <-- Start Features --!>
    <"border1-item featured-feature"=div class>
        <"left-icon float-feature"=div class>
            <i/><"cup-coffee-Ini"=i class>
        <div/>
        <"left-info float-feature"=div class>
            <h4>Easy to Use<h4>
            it is a super essay to use <br>,Although kidO is very complex software<p>
<p/>.with a beautiful graphic interface<br> software
    <div/>
    <div/>
    <-- End Features --!>
    <div/>
    <-- End Col --!>
```

```

<-- Start Col --!>
<"0-p 12-xs-col 6-md-col 6-lg-col"=div class>
  <-- Start Features --!>
  <"border2-item featured-feature"=div class>
    <"left-icon float-feature"=div class>
      <i/><"briefcase-Ini"=i class>
      <div/>
    <"left-info float-feature"=div class>
      <h4/>For busy parents<h4>
      sed do ,adipiscing elit <br>consectetur ,Lorem ipsum dolor sit amet<p>
<p/>.incididunt ut labore et dolore magna aliqua <br>eiusmod tempor
  <div/>
  <div/>
  <-- End Features --!>
<div/>
<-- End Col --!>

<-- Start Col --!>
<"0-p 12-xs-col 6-md-col 6-lg-col"=div class>
  <-- Start Features --!>
  <"border1-item featured-feature"=div class>
    <"left-icon float-feature"=div class>
      <i/><"invention-Ini"=i class>
      <div/>
    <"left-info float-feature"=div class>
      <h4/>Trendy Design & Clean<h4>
      sed do ,adipiscing elit <br>consectetur ,Lorem ipsum dolor sit amet<p>
<p/>.incididunt ut labore et dolore magna aliqua <br>eiusmod tempor
  <div/>
  <div/>
  <-- End Features --!>
<div/>
<-- End Col --!>

<-- Start Col --!>
<"0-p 12-xs-col 6-md-col 6-lg-col"=div class>
  <-- Start Features --!>
  <"border2-item featured-feature"=div class>
    <"left-icon float-feature"=div class>
      <i/><"layers-Ini"=i class>
      <div/>
    <"left-info float-feature"=div class>
      <h4/>Tons of Sections<h4>
      sed do ,adipiscing elit <br>consectetur ,Lorem ipsum dolor sit amet<p>
<p/>.incididunt ut labore et dolore magna aliqua <br>eiusmod tempor
  <div/>
  <div/>
  <-- End Features --!>
<div/>
<-- End Col --!>

<-- Start Col --!>

```

```

<"0-p 12-xs-col 6-md-col 6-lg-col">=div class>
  <-- Start Features --!>
  <"border3-item featured-feature">=div class>
    <"left-icon float-feature">=div class>
      <i/><"reload-Ini">=i class>
      <div/>
      <"left-info float-feature">=div class>
        <h4/>Free Future Updates<h4>
        sed do ,adipiscing elit <br>consectetur ,Lorem ipsum dolor sit amet<p>
<p/>.incididunt ut labore et dolore magna aliqua <br>eiusmod tempor
  <div/>
  <div/>
  <-- End Features --!>
<div/>
<-- End Col --!>

<-- Start Col --!>
<"0-p 12-xs-col 6-md-col 6-lg-col">=div class>
  <-- Start Features --!>
  <"item-feature">=div class>
    <"left-icon float-feature">=div class>
      <i/><"support-Ini">=i class>
      <div/>
      <"left-info float-feature">=div class>
        <h4/>Premier Support<h4>
        sed do ,adipiscing elit <br>consectetur ,Lorem ipsum dolor sit amet<p>
<p/>.incididunt ut labore et dolore magna aliqua <br>eiusmod tempor
  <div/>
  <div/>
  <-- End Features --!>
<div/>
<-- End Col --!>

<div/>
<-- End Row --!>
<div/>
<section/>
<-- Cool Features Section End --!>

<-- Contact Us Section --!>
<"section">=class "contact">=section id>
  <-- Container Starts --!>
  <"container">=div class>
    <-- Start Row --!>
    <"row">=div class>
      <"12-lg-col">=div class>
        <"center-header text-text section-contact">=div class>
          <div>
            <h2/>Get In Touch<"title-section">=h2 class>
            <"text-desc">=div class>
              <p/>kidO developer would love to answer your questions and<p>
              <p/>help with any comming question<p>

```

```

<div/>
<div/>
<div/>
<div/>

<div/>
<!-- End Row --!>
<!-- Start Row --!>
<"row"=div class>
<!-- Start Col --!>
<"12-md-col 6-lg-col"=div class>
<"contactForm"=form id>
<"row"=div class>
<"6-md-col"=div class>
<"group-form"=div class>
    "name"=name "name"=id "control-form"=class "text"=input type>
<"Please enter your name"=error-required data "Name"=placeholder
<div/><"errors-block with-help"=div class>
<div/>
<div/>
<!-- <"6-md-col"=div class>          --!>
<!-- <"group-form"=div class>          --!>
<"form"=class "subject_msg"=id "Subject"=placeholder "text"=input type>          --!>
<!-- <"Please enter your subject"=error-required data "subject_msg"=name "control
<-- <div/><"errors-block with-help"=div class>          --!>
<-- <div/>          --!>
<-- <div/>          --!>
    <"6-md-col"=div class>
    <"group-form"=div class>
        "email"=name "email"=id "control-form"=class "text"=input type>
<"Please enter your Email"=error-required data "Email"=placeholder
    <div/><"errors-block with-help"=div class>
    <div/>
    <div/>
<!-- <"6-md-col"=div class>          --!>
<!-- <"group-form"=div class>          --!>
<"control-form"=class "budget"=id "Budget"=placeholder "text"=input type>          --!>
<!-- <"Please enter your Budget"=error-required data "budget"=name
<-- <div/><"errors-block with-help"=div class>          --!>
<-- <div/>          --!>
<-- <div/>          --!>
    <"12-md-col"=div class>
    <"group-form"=div class>
        "message"=name "message"=id "control-form"=textarea class>
/><required "Write your message"=error-data "4"=rows "Write Message"=placeholder
<textarea
    <div/><"errors-block with-help"=div class>
    <div/>
    <"button-submit"=div class>
        <button/>Submit<"submit"=type "submit"=id "common-btn btn"=button class>
        <div/><"h3 hidden"=class "msgSubmit"=div id>
        <div/><"clearfix"=div class>
    <div/>

```

```

<div/>
<div/>
<form/>
<div/>
<!-- End Col --!>
<!-- Start Col --!>
<"1-lg-col"=div class>

<div/>
<!-- End Col --!>
<!-- Start Col --!>
<"12-md-col 4-lg-col"=div class>
<"img-contact"=div class>
<""=alt "fluid-img"=class "png.01/contact/img"=img src>
<div/>
<div/>
<!-- End Col --!>
<!-- Start Col --!>
<"1-lg-col"=div class>
<div/>
<!-- End Col --!>

<div/>
<!-- End Row --!>
<div/>
<section/>
<!-- Contact Us Section End --!>

<!-- Footer Section Start --!>
<footer>
<!-- Footer Area Start --!>
<"Content-footer"=section id>
<"center-info text-site"=div class>
<p><a>Ori Rinat<"nofollow"=a rel> Created by<p>
<div/>
<div/>
<section/>
<!-- Footer area End --!>

<footer/>
<!-- Footer Section End --!>

<!-- Go To Top Link --!>
<"top-to-back"=class "#"=a href>
<i/><"up-chevron-Ini"=i class>
<a/>

<!-- Preloader --!>
<"preloader"=div id>
<div/><"1-loader"=id "loader"=div class>
<div/>
<!-- End Preloader --!>

```

```
<-- .then Bootstrap JS ,then Tether ,jQuery first --!>
<script/><"js.min-jquery/js"=script src>
<script/><"js.min.popper/js"=script src>
<script/><"js.min.bootstrap/js"=script src>
<script/><"js.carousel.owl/js"=script src>
<script/><"js.nav.jquery/js"=script src>
<script/><"js.nav-scrolling/js"=script src>
<script/><"js.min.easing.jquery/js"=script src>
<script/><"js.lightbox-nivo/js"=script src>
<script/><"js.min.popup-magnific.jquery/js"=script src>
<script/><"js.main/js"=script src>

<body/>
<html/>
```

# צד חורה - login.php - התחברות למשתמש

```
<?php
    require "../usertools.php";
    if (checkLogin() != null) {
        header("Location: see.php");
        exit();
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>KidO login</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <script src="hashing.js"></script>
    <style>
        #error {
            color: red;
        }
    </style>

<!--
=====
=====-->
    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
<!--
=====
=====-->
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
<!--
=====
=====-->
    <link rel="stylesheet" type="text/css" href="fonts/fontawesome-4.7.0/css/font-awesome.min.css">
<!--
=====
=====-->
    <link rel="stylesheet" type="text/css" href="fonts/iconic/css/material-design-iconic-font.min.css">
<!--
=====
=====-->
    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
<!--
=====
=====-->
    <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/css-hamburgers/hamburgers.min.css">
```

```

<!--
=====
=====-->
<link rel="stylesheet" type="text/css" href="vendor/animsition/css/
animsition.min.css">
<!--
=====
=====-->
<link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
<!--
=====
=====-->
<link rel="stylesheet" type="text/css" href="vendor/daterangepicker/
daterangepicker.css">
<!--
=====
=====-->
<link rel="stylesheet" type="text/css" href="css/util.css">
<link rel="stylesheet" type="text/css" href="css/main.css">
<!--
=====
=====-->
</head>

<body>
    <div class="container-login100" style="background-image: url('images/bg-01.jpg');">
        <div class="wrap-login100 p-l-55 p-r-55 p-t-80 p-b-30">
            <form class="login100-form validate-form">
                <span class="login100-form-title p-b-37">
                    kidO - Sign In
                </span>

                <div class="wrap-input100 validate-input m-b-20" data-validate="try
again">
                    <input class="input100" type="text" placeholder="username"
id="username">
                    <span class="focus-input100"></span>
                </div>
                <br>
                <div class="wrap-input100 validate-input m-b-20" data-validate="try
again">
                    <input class="input100" type="password"
placeholder="password" id="password">
                    <span class="focus-input100"></span>
                </div>
                <p align="center" id="error"></p>
                <br>

                <div class="container-login100-form-btn">
                    <button onclick = "login()"> Login </button>
                </div>

                <script>

```

```

        function login() {
            var formData = new FormData();
            formData.append("username",
document.getElementById("username").value);

formData.append("password",MD5(document.getElementById("password").value));
            var xhr = new XMLHttpRequest();
            xhr.open("POST", "loginconfirm.php");
            xhr.send(formData);
            xhr.onreadystatechange = function() {
                if (this.readyState == 4){
                    if (this.status == 200) { // always 0
                        var data = xhr.responseText;
                        if (data === "ok") {
                            document.cookie =
"username="+document.getElementById("username").value;
                            document.cookie =
"password="+MD5(document.getElementById("password").value);
                            open("see.php","_self");
                        } else {

document.getElementById("error").innerHTML = data;
                    }
                }
            };
        }
    </script>

<div class="text-center p-t-57 p-b-20">
    <span class="txt1">
        Or login with
    </span>
</div>

<div class="flex-c p-b-112">
    <a href="#" class="login100-social-item">
        <i class="fa fa-facebook-f"></i>
    </a>

    <a href="#" class="login100-social-item">
        
    </a>
</div>

<div class="text-center">
    <span class="txt1">
        Sign up in the kidO APP
    </span>
</div>
</form>
</div>
</div>

```

```
<div id="dropDownSelect1"></div>

<!--
=====
=====-->
<script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
=====
=====-->
<script src="vendor/animsition/js/animisition.min.js"></script>
<!--
=====
=====-->
<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
=====
=====-->
<script src="vendor/select2/select2.min.js"></script>
<!--
=====
=====-->
<script src="vendor/daterangepicker/moment.min.js"></script>
<script src="vendor/daterangepicker/daterangepicker.js"></script>
<!--
=====
=====-->
<script src="vendor/countdowntime/countdowntime.js"></script>
<!--
=====
=====-->
<script src="js/main.js"></script>

</body>
</html>
```

```
<?php
require "../usertools.php";
if (checkLogin() == null) {
    header("Location: index.php");
    exit();
}
?>
<!DOCTYPE html>
<html>
<head>
<title>kidO dashboard</title>

<meta http-equiv="refresh" content="10" >

<style>

.container-table100 {
    width: 100%;
    min-height: 100vh;
    background: #c850c0;
    background: -webkit-linear-gradient(45deg, #4158d0, #c850c0);
    background: -o-linear-gradient(45deg, #4158d0, #c850c0);
    background: -moz-linear-gradient(45deg, #4158d0, #c850c0);
    background: linear-gradient(45deg, #4158d0, #c850c0);

    display: -webkit-box;
    display: -webkit-flex;
    display: -moz-box;
    display: -ms-flexbox;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-wrap: wrap;
    padding: 33px 30px;
}
.wrap-table100 {
    width: 90%;
    margin-top: 50px;
    margin-left: 5%;
}
@media only screen and (min-width: 800px) {
    .wrap-table100 {
        width: 40%;
        margin-top: 50px;
        margin-left: 33%;
    }
}
table {
    border-spacing: 1;
```

```
border-collapse: collapse;
background: white;
opacity: 0.9;
border-radius: 10px;
overflow: hidden;
width: 100%;
position: relative;
align: center;
}
table * {
  position: relative;
}
table td, table th {
  padding-left: 8px;
/* align-content: 1px; */
}
table thead tr {
  height: 60px;
  background: #36304a;
}
table tbody tr {
  height: 50px;
}
table tbody tr:last-child {
  border: 0;
}
table td, table th {
  text-align: left;
}
table td.l, table th.l {
  text-align: right;
}
table td.c, table th.c {
  text-align: center;
}
table td.r, table th.r {
  text-align: center;
}

h1 {
  font-family: OpenSans-Regular;
  font-size: 40px;
  color: #fff;
  line-height: 1.2;
  font-weight: unset;
  margin-left: 43%;
}
h3 {
  font-family: OpenSans-Regular;
  font-size: 20px;
  color: #fff;
  line-height: 1.2;
  font-weight: unset;
```

```

}

button {
    font-family: OpenSans-Regular;
    font-size: 18px;
    line-height: 1.2;
    font-weight: unset;
    margin-left: 2%;
    border-radius: 5px;
    box-shadow: 0 10px 30px 0px rgba(0, 0, 0, 0.1);
    -moz-box-shadow: 0 10px 30px 0px rgba(0, 0, 0, 0.1);
    -webkit-box-shadow: 0 10px 30px 0px rgba(0, 0, 0, 0.1);
    -o-box-shadow: 0 10px 30px 0px rgba(0, 0, 0, 0.1);
    -ms-box-shadow: 0 10px 30px 0px rgba(0, 0, 0, 0.1);
}

.table100-head th{
    font-family: OpenSans-Regular;
    font-size: 18px;
    color: #fff;
    line-height: 1.2;
    font-weight: unset;
}

@font-face {
    font-family: OpenSans-Regular;
    src: url('fonts/OpenSans/OpenSans-Regular.ttf');
}

tbody tr {
    font-family: OpenSans-Regular;
    font-size: 15px;
    color: #808080;
    line-height: 1.2;
    font-weight: unset;
}

tbody tr:hover {
    color: #555555;
    font-size: 17px;
    background-color: #f5f5f5;
    cursor: pointer;
}

</style>
</head>

<body style = " height:100% background: #c850c0; background: -webkit-linear-gradient(45deg, #4158d0, #c850c0); background: -o-linear-gradient(45deg, #4158d0, #c850c0); background: -moz-linear-gradient(45deg, #4158d0, #c850c0); background: linear-gradient(45deg, #4158d0, #c850c0);">
<script>
    function logout() {
        document.cookie = "username= "

```

```
document.cookie = "password="
open("login.php","_self");
}
</script>
<button onclick = "logout()">log out</button>

<h1>kidO - dashboard</h1>
<div class="wrap-table100">
<table>
<?php
$username = checkLogin();
$content = content("../files/".$username.".csv");
if ($content == "none"){
    return;
}
$max = 0;
foreach ($content as $line) {
    $commas = count($line);
    if ($max < $commas + 1) {
        $max = $commas + 1;
    }
}
foreach ($content as $line) {
    echo "<tr>";
    for ($cell = 0; $cell < $max; $cell++) {
        if ($cell < count($line)) {
            echo "<td>".$line[$cell]."</td>";
        } else {
            echo "<td></td>";
        }
    }
    echo "</tr>";
}
?>
</table>
</div>
</body>
</html>
```

קובץ זה הוא דוגמה לקובץ הנוצר לכל ילד, זה הוא קובץ csv אשר מכיל את פרטי האירוע כדלקמן

```
details,event,time
ass,Fri Apr 19 11:20:34 2019,typed
fuck,Fri Apr 19 11:20:36 2019,typed
ass,Fri Apr 19 11:20:47 2019,read
fuck,Fri Apr 19 11:20:47 2019,read
anal,Fri Apr 19 11:22:12 2019,read
gook,Fri Apr 19 11:23:09 2019,typed
fuck,Fri Apr 19 11:23:56 2019,read
fuck,Fri Apr 19 11:23:56 2019,read
anal,Fri Apr 19 11:23:56 2019,read
ass,Fri Apr 19 11:24:13 2019,read
anal,Fri Apr 19 11:24:13 2019,read
fuck,Fri Apr 19 11:24:13 2019,read
anal,Fri Apr 19 11:24:13 2019,read
```

קובץ המכיל את כל המשתמשים, שם משתמש וסיסמה אחרי שהיא עברה גיבוב  
salting-

```
fun,1a1dc91c907325c69271ddf0c944bc72
ori,1a1dc91c907325c69271ddf0c944bc72
par,1a1dc91c907325c69271ddf0c944bc72
parent,1a1dc91c907325c69271ddf0c944bc72
ido,1a1dc91c907325c69271ddf0c944bc72
klkl,50a2c33221d835c9a5062f6eaa44abc1
kido,1a1dc91c907325c69271ddf0c944bc72
kid2,1a1dc91c907325c69271ddf0c944bc72
my_name,7c6a180b36896a0a8c02787eeafb0e4c
```

**צד שרת - הורה -  
קובץ php האחראי על התחרבות**

```
<?php
require "csvtools.php";
function checkLogin() {
    $userscsv = "../users.csv";
    if (!isset($_COOKIE["username"]) or !isset($_COOKIE["password"])) {
        return null;
    }
    $username = $_COOKIE["username"];
    $password = $_COOKIE["password"];
    $users = content($userscsv);
    foreach ($users as $user) {
        if ($user[0] == "$username" && $user[1] == "$password") {
            return $username;
        }
    }
    return null;
}

function login($username, $password) {
    $userscsv = "../users.csv";
    $users = content($userscsv);
    foreach ($users as $user) {
        if ($user[0] == "$username" && $user[1] == "$password") {
            return $username;
        }
    }
    return null;
}
?>
```