digital whisper
Technologic Papers

לרמות במשחק של עצמך

מאת Dixe0

הקדמה

מגיל צעיר יחסית אהבתי לבנות אתרים. בערך בכיתה ה' בניתי את האתר הראשון שלי (בשנות ה-2000 המוקדמות), מאז התחום הזה תמיד סיקרן אותי, התחלתי עם HTML ו-CSS, עם הזמן למדתי על JavaScript שפתאום נתן חיים לאתר ופתח המון אפשרויות, אחרי תקופה למדתי גם PHP, עבודה עם מסדי נתונים (MySQL), בערך בשנת 2013 החלטתי שאני רוצה לפתח משחק דפדפן.

ההחלטה הזו קרתה מפני ששיחקתי כמה שנים במשחק דפדפן ישראלי שנסגר, חשבתי שאולי אני יכול לנסות לפתח משהו דומה, מדובר על משחק דפדפן שמתרחש בימיי הביניים, נרשמים למשחק ובעצם יוצרים דמות של לוחם, הדמות הזו יכולה לקנות חפצים ונשקים, לשדרג את יכולות הלחימה, לצבור ניסיון מקרבות למול שחקנים אחרים ועוד לא מעט אפשרויות שהתווספו במהלך הפיתוח. זה הצריך ממני למידה של שפות חדשות, טכנולוגיות חדשות והשקעה של מאות, באמת מאות שעות של פיתוח. בראייה לאחור אני חושב שרוב הידע שלי בתכנות אתרים הוא בזכות המשחק הזה, יש הרבה יותר מוטיבציה ופוקוס בלמידה כשאתה מעמיק מול צורך אמיתי (לדוגמה - איך אני מפתח עכשיו מערכת של חנות במשחק?) מאשר ללמוד ללא פרויקט בקצה.

לצד הפיתוח של מערכת הקרבות שהיא היתה פרויקט בפני עצמו, היה צריך לפתח עוד המון מערכות שונות, מערכת שוק כדי ששחקנים יוכלו לסחור בנשקים, מערכת חנות כדי לאפשר לשחקנים לקנות ציוד לחימה והגנה, מערכת בנק שתאפשר לשחקנים לשמור כסף ולהעביר כסף, מערכת שבטים (clans) כדי לאפשר לשחקנים להתאגד לכדי שבט אחד ולהלחם יחד למול שבטים אחרים, כל אחת מהמערכות היתה אתגר לא קטן בזמנו ולא מעט מערכות הסתמכו אחת על השניה, מעבר רק לתכנות היה צורך לתכנן כלכלה כמו של עיר קטנה, כמה כסף כל שחקן יקבל מפעולות שונות, מה יהיה המחיר של הנשקים והחפצים השונים, מתי להקשות יותר כדי לייצר עניין ומתי להקל בפעולות מסוימות כדי לא לשחוק את השחקנים. מאוד נהנתי מהחשיבה הזו ומהיכולת להשפיע על חווית המשחק של השחקנים.

אחרי כמעט שנה של פיתוח, המשחק יצא לאור, צבר לא מעט שחקנים בתקופה שהיה פתוח ולדעתי (שכנראה לא אובייקטיבית) הפך למשחק הדפדפן הישראלי מהמושקעים שהיו באותם שנים, מצרף כמה תמונות להמחשה:

לא מזמן העליתי את המשחק בחזרה (פרץ נוסטלגיה) והתחלתי קצת לשחק, אבל הפעם עשיתי את זה קצת שונה, פתחתי תוכנה שנקראת Burp (ארחיב עליה מטה) ואחרי בערך עשרים דקות שאני משחק, הסתכלתי על הבקשות הנשלחות לשרת המשחק, פתאום הבנתי שאני יכול לרמות במשחק של עצמי(!).

בזמנו לא היה לי ידע רחב בעולמות אבטחת מידע \ סייבר, למדתי לתכנת בצורה עצמאית ולרוב על פי צורך של אותם פיצ'רים שרציתי לפתח, לא תמיד הבנתי את המשמעות של לוודא בצד שרת פרמטרים מסוימים, מתי נכון להשתמש בבקשות POST או GET והתמקדתי בעיקר בנוחות שלי כדי לפתח כמה שיותר מהר.

אז החלטתי הפעם לנסות ולמצוא צ'יטים במשחק של עצמי, כדי לראות עד כמה שחקנים אחרים יכלו לנצל את המשחק לטובתם.

Proxy

כדי לנסות לעקוף או לבצע מניפולציות על המשחק יהיה צורך לראות את הבקשות היוצאות לשרת המשחק, ולכן אשתמש ב-Proxy, המשמעות היא שכל תעבורת ה-HTTP מהדפדפן שלי תעבור דרך פורט ספציפי, זאת כדי שאני אוכל לראות את כל הבקשות והתגובות שאני מקבל מהשרת. כדי להמחיש - בקשות ה-HTTP יוצאות מהדפדפן שלי, אל תוכנה שבעזרתה אוכל לבצע פעולות ולראות את הבקשות והתגובות, התוכנה הזו תעביר את הבקשות דרכה אל שרת המשחק:

Burp Suite

את Burp suite חלקכם כנראה מכירים, אך למי מכם שאינו מכיר: זו תוכנה שבעזרתה אפשר לצפות בבקשות HTTP/S שיוצאות מהמחשב שלכם אל השרת שאליו אתם פונים, הכלי הזה נועד כדי לעזור לצוותי אבטחת מידע וחוקרים, Burp suite מספקת סט כלים מאוד משמעותי ונוח, כמו עריכה ושינוי בזמן אמת של בקשות, השוואה בין תשובות שחוזרות מהשרת (compare) ועוד המון אפשרויות, במאמר אני יוצא מנקודת הנחה שיש לכם היכרות מוקדמת עם הכלי, במידה ולא - יש המון הסברים מצויינים ברחבי הרשת, המדריך מ-Geeksforgeeks בהחלט יכול להסביר בצורה טובה לגבי הכלי והשימוש בו.

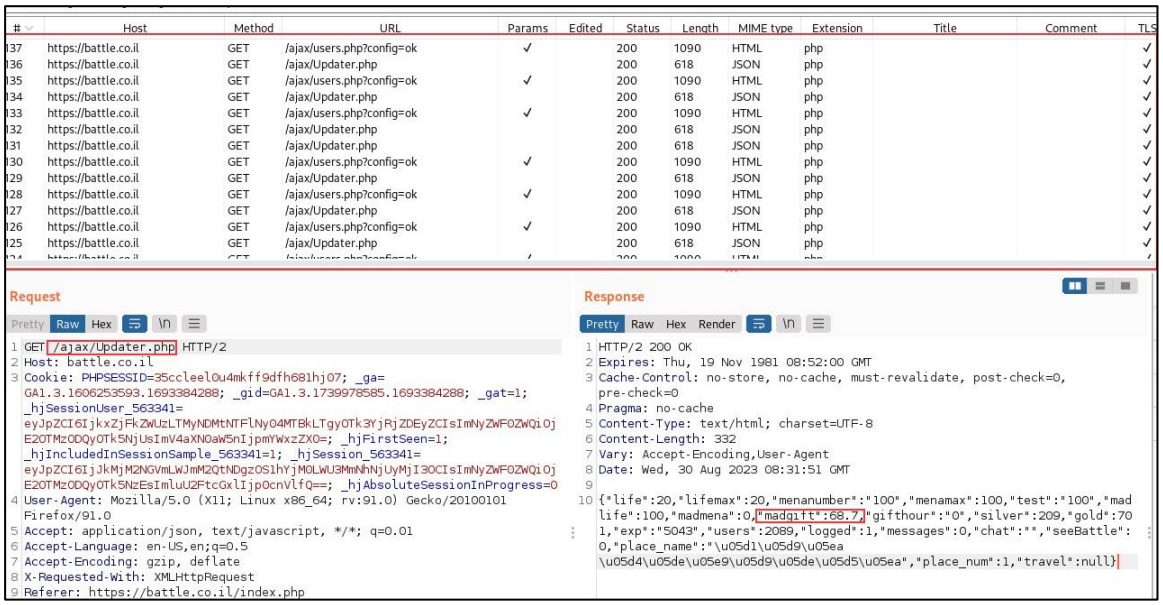
אז אחרי שפתחתי את Burp Suite והתחלתי לשוטט במשחק למשך כמה דקות, התחלתי להסתכל על בקשות ה-HTTPS שאני מוציא לשרת המשחק, נתקלתי בכמה פונקציונליות של המשחק שנראו לי די מוזרות, בשלב הזה החלטתי לנסות לא להציץ ב-Source Code ולגשת למשחק כאילו אני שחקן רגיל.

מי לא אוהב מתנות בחינם?

אחד האתגרים הגדולים בפיתוח משחקים הוא איך לשמור שחקנים במשחק למשך כמה שיותר זמן, אז מעבר לאפשרויות של המשחק, אחד הרעיונות שעלו לי הוא לעשות מערכת שבכל דקה ששחקן מחובר הוא יצבור נקודה, ברגע שהוא יהיה במשחק שלוש שעות - הוא יקבל תגמול מסוים, כסף או חפץ במשחק,

המטרה הייתה באמת לגרום לשחקנים להישאר מחוברים וזה עבד לא רע בזמנו, שחקנים שנשאר להם עוד 15-20 דקות כדי לקבל את הפרס המתינו מחוברים וככה שמרנו על כמות מחוברים לא רעה בעיקר בערב, ניתן לראות את המדד המודגש באדום:

כשצפיתי בבקשות HTTPS שנשלחות לשרת המשחק, שמתי לב שישנה פניה שחוזרת על עצמה כל דקה ל-2 עמודים, אחת הפניות נעשת לדף בשם Updater.php, לא זכרתי מה תפקידו אבל שראיתי את התשובה שחזרה מהשרת (Response) מיד נזכרתי שהוא אמור לשקף את נתוני השחקן ושיש פונקציות במשחק שעושות שימוש בנתונים שלו. שמתי לב שבכל פעם שאני שולח בקשה, הפרמטר שנקרא madgift (שאפשר בקלות להבין שזה המדד של המתנות) עולה ב-1, אז כאמור כל דקה מתבצעת אליו פניה אוטומטית, אבל מה יקרה אם נפנה אליו יותר מפעם אחת בדקה? האם הדבר הזה ישפיע על מד המתנות? אז כמובן שהחלטתי לבדוק בעצמי, ניתן לראות בתמונה הבאה את הבקשה לדף Updater.php ואז תגובת השרת (Response):



#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS
137	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
136	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
135	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
134	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
133	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
132	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
131	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
130	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
129	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
128	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
127	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓
126	https://battle.co.il	GET	/ajax/users.php?config=ok	✓		200	1090	HTML	php			✓
125	https://battle.co.il	GET	/ajax/Updater.php			200	618	JSON	php			✓

Request

Pretty Raw Hex

```
1 GET /ajax/Updater.php HTTP/2
2 Host: battle.co.il
3 Cookie: PHPSESSID=35ccleel0u4mkff9dfh681hj07; _ga=
GA1.3.1606253593.1693384288; _gid=GA1.3.1739978585.1693384288; _gat=1;
_hjSessionUser_563341=
eyJpZCI6IjkxZjFkZWUzLTMyNDMtNTFlNy04MTBkLTgyOTk3YjRjZDEyZCIsImNyZWF0ZWQiOj
E2OTMzODQyOTk5NjUsImV4aXN0aW5nIjpmYWxzZX0=; _hjFirstSeen=1;
_hjIncludedInSessionSample_563341=1; _hjSession_563341=
eyJpZCI6IjJkMjM2NGVmLWJmM2QtNDgzOS1hYjM0LWU3MmNhNjUyMjI3OCIsImNyZWF0ZWQiOj
E2OTMzODQyOTk5NzEsImluU2FtcGxlIjp0cnVlfQ==; _hjAbsoluteSessionInProgress=0
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101
Firefox/91.0
5 Accept: application/json, text/javascript, */*; q=0.01
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 X-Requested-With: XMLHttpRequest
9 Referer: https://battle.co.il/index.php
```

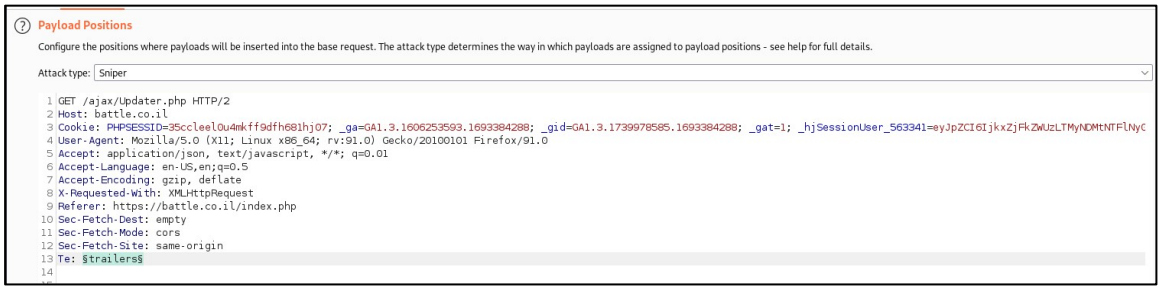
Response

Pretty Raw Hex Render

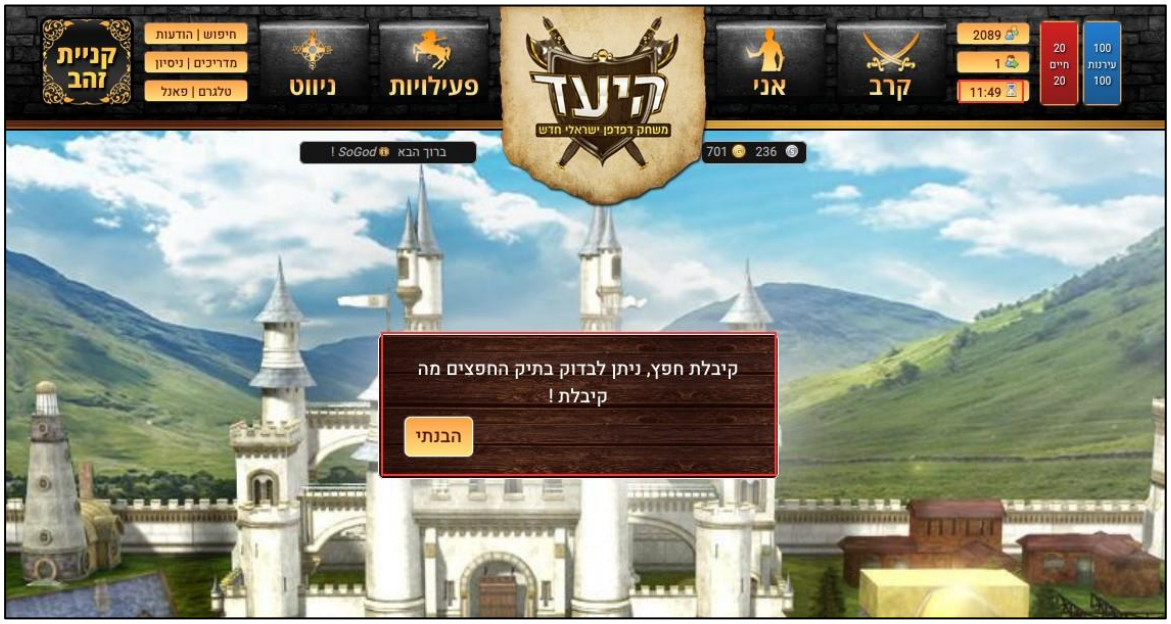
```
1 HTTP/2 200 OK
2 Expires: Thu, 19 Nov 1981 08:52:00 GMT
3 Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
4 Pragma: no-cache
5 Content-Type: text/html; charset=UTF-8
6 Content-Length: 332
7 Vary: Accept-Encoding,User-Agent
8 Date: Wed, 30 Aug 2023 08:31:51 GMT
9
10 {"life":20,"lifemax":20,"menanumber":"100","menamax":100,"test":"100","mad
life":100,"madmena":0,"madgift":68.7,"gifthour":"0","silver":209,"gold":70
1,"exp":"5043","users":2089,"logged":1,"messages":0,"chat":"","seeBattle":
0,"place_name":"\u05d1\u05d9\u05ea
\u05d4\u05de\u05e9\u05d9\u05de\u05d5\u05ea","place_num":1,"travel":null}
```

כדי לבדוק איך שליחת כמות גדולה של בקשות לדף זה תשפיע, קודם כל ווידאתי שמד המתנות שלי מאופסת, ניתן לראות שהשעה היא 11:46:

כדי לשלוח בערך 100 בקשות בדקה השתמשתי כמובן ב-Burp suite בפי'צר שנקרא Intruder. זה בעצם Fuzzer שיכול לבצע שינוי של ערכים בבקשה ולשלוח בכל פעם את הבקשה שנרצה עם השינוים, היתרון שלו הוא שאפילו בגרסה החינמית של Burp Suit שנקראת Community Edition יש אפשרות לשלוח מספר לא מועט של בקשות בשניה, שלחתי את אותה הבקשה שיצאה לעמוד Updater.php וככה היא נראית ב-Intruder:



המטרה היתה לנסות לשלוח כ-100 בקשות בדקה, שיגרתי את הבקשות לשרת המשחק וחיכיתי, בזמן הזה ראיתי המון בקשות יוצאות לשרת המשחק ואחרי בערך 200 בקשות השרת התחיל קצת "לגמגם" שמתי לב שלוקח לו בערך 5-7 שניות להחזיר תגובה (Response), כנראה שהקצב הזה היה מעט גבוה בשבילו, אבל לאחר שכל הבקשות נשלחו, חזרתי למשחק עצמו כדי לבדוק האם באמת השרת התייחס לכל הבקשות ששלחתי לו, והעלה לי את מד המתנות בכל בקשה שהתקבלה אצלו בהצלחה. בשעה 11:49 (כשלוש דקות לאחר תחילת שליחת הבקשות) מד המתנות שלי הגיע למקסימום (רמה 3). כמובן שרק לאחר ההגעה לרמה המקסימלית ניתן בעצם לקבל תגמול, מה שאומר שב-3 דקות הצלחתי לקבל תגמול על פעילות שהיתה אמורה להמשך כ-3 שעות:



מה המשמעות של דבר כזה?

שחקנים יכולים תוך דקות לקבל חפצים ומטבעות כסף מה שיפגע במוטיבציה שלהם להיות מחוברים לאורך זמן, גרוע מכך הדבר הזה יכול לגרום לאינפלציה ושחקנים כבר לא יצטרכו למצוא דרכים שקיימות במשחק כדי להרוויח כסף, מה כשלעצמו יכול להוריד את הזמן שבו שחקנים צריכים להיות מחוברים ולא יהיה להם כבר עניין במשחק, אבל בנוסף אם אותו משחק מאפשר רכישה של חפצים או מטבעות במשחק בכסף אמיתי (לדוגמה שלם 5$ וקבל 150 מטבעות כסף במשחק). דבר כזה יכול להוריד דרסטית את ההכנסות של המשחק, מה שבמקרים קיצוניים יכול לגרום אפילו לסגירה של המשחק בגלל חוסר יכולת לכסות את הוצאות השרת, התחזוקה והפיתוח!

מה המיטגציה?

כדי למנוע מהמקרה להיות מנוצל לרעה על ידי שחקנים ישנים כמה פעולות שנוכל לעשות:

- לכתוב לקובץ נפרד שמטפל בניקוד ועליו יתבצע cron-job (משימה מתוזמנת) פעם בדקה, ככה בעצם שם הקובץ לא ייחשף בצד לקוח ויהיה קשה יותר להגיע אליו.
- להגדיר לאותו קטע קוד שיבצע Execute אך ורק אם הוא מגיע מכתובת IP מסוימת, כדי קודם כל לוודא שהבקשה מגיעה מהשרת שלי - ולא מכתובת IP לא רצויה.
- הצמדת Header עם ערך מסוים שהשרת מצפה לו, ככה גם אם יגיעו בקשות לאותו הדף אבל ללא הפרמטרים המוזכרים מעלה - הוא לא יבצע Execution לאותו קוד PHP שאחראי להעלאת הניקוד.

מימוש בפועל (Patch):

בהנחה וכתבנו את הקוד שאחראי על הניקוד בדף אחר, אנחנו יודעים שמי שאמור לפנות לאותו דף הוא אך ורק שרת המשחק עצמו, שבו מתבצעת פניה לדף בכל דקה באמצעות Cron Job (משימה מתוזמנת), לכן נרצה לגביל את הפניה לדף שלנו ל-IP הציבורי של שרת המשחק, לאחר שאנחנו יודעים מה היא אותה כתובת IP, נוכל בעזרת PHP לבדוק מאיזו כתובת IP נשלחה אל הקוד שלנו בקשת ה-HTTP, אפשר לעשות זאת באמצעות:

```
$_SERVER['REMOTE_ADDR']
```

מה שנוכל לעשות זה להכניס תנאי לפני הפעולה של העלאת הנקודות, התנאי ייבדוק מאיזה IP הגיעה בקשת ה-HTTP, ורק אם הכתובת ה-IP תואמת למה שהגדרנו אז הוא יכנס לתוך בלוק הקוד הרלוונטי ויבצע אותו, בפועל בקוד עצמו זה נראה ככה:

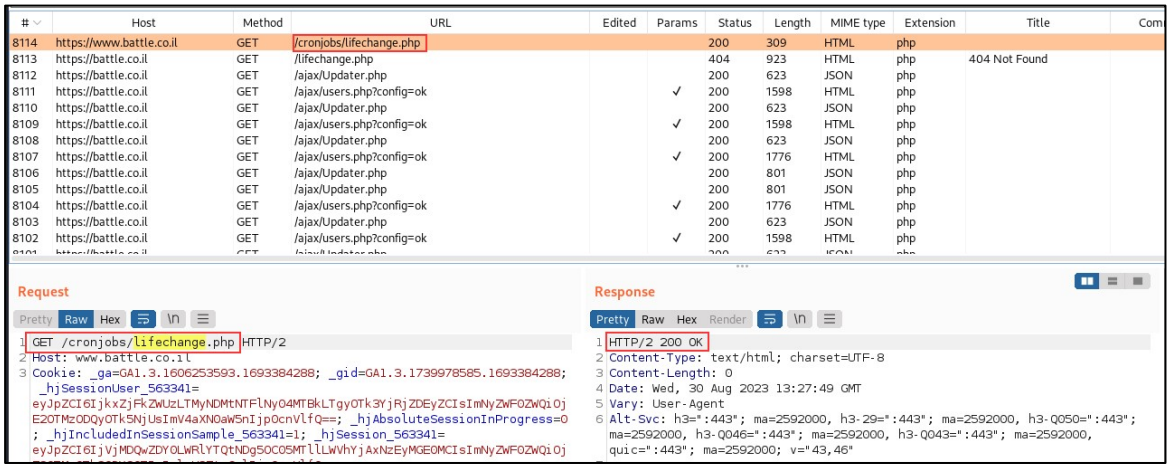
```
<?php
    if ($_SERVER['REMOTE_ADDR'] == '<OUR_SERVER_IP_ADDRESS>') {
        function username($id) {
```

לאחר התוספת הזו לקוד ביצענו מעין Patch לניצול שראינו למעלה, בכל פניה לדף שבו יש את הקוד שאחראי על מד המתנות, קודם כל תתבצע בדיקה האם הבקשה הגיע מה-IP של שרת המשחק, במידה ולא היא לא תמשיך לבצע את הפעולות הקשורות למד המתנות. האם זה מספיק? כנראה שלא, הפעולה שעשינו למעלה אכן תוריד משמעותית את הסיכון שניצול כזה יחזור על עצמו, אבל אנחנו יכולים להקשות עוד יותר על המימוש הזה, בהמשך הכתבה נבצע Patch נוסף ושם נסגור בצורה הרמטית עוד יותר.

נקודות בחינם ולכולם

אחד מנתוני השחקן החשובים ביותר הוא מספר נקודות החיים, במהלך קרב שחקן סופג פגיעות מהיריב והדבר גורם לירידה בנקודות החיים, גם במידה ולשחקן ירדו כל נקודות החיים יש תהליך לאחר הקרב שמעלה לשחקן 2% מסך נקודות החיים שלו בכל דקה, זאת אומרת שנניח ויש לשחקן 100 נקודות חיים סך הכל והוא יצא עם 0 נקודות חיים מהקרב, לאחר 50 דקות נקודות החיים שלו יתמלאו חזרה ל-100 נקודות.

בזמן שקניתי נשק ראיתי כמה בקשות יוצאות לשרת, שהסתכלתי עליהן ב-Burp Suite שמתי לב לפניה לדף PHP שאני זוכר בדיוק מה הוא עשה (כי השקעתי עליו לא מעט שעות תכנות), אותו הדף נקרא lifechange.php, וכך נראית הבקשה לדף:



אומנם אין תגובה מפורטת מהשרת ואפשר לראות רק שהבקשה נענתה בהצלחה, אבל השם של הדף די אינטואיטיבי ואפשר להבין שהוא קשור לשינוי נקודות החיים, כדי לבדוק את השפעת הדף הזה נכנסתי לקרב שבו הפסדתי בצורה מכוונת, יצאתי ממנו עם 0 נקודות חיים ושלחתי כמה בקשות לדף בצורה ידנית (דרך הדפדפן), פתאום שמתי לב שנקודות החיים של ושכל מי שלא נמצא בקרב פעיל - עולות!.

אז כשהיו לי 39 נקודות חיים חשבתי לשלוח שוב באמצעות Intruder מאות בקשות לדף lifechange.php, כדי לראות האם אפשר להאיץ את תהליך העלאת נקודות החיים ולהגיע למקסימום בדקות בודדות במקום לחכות כמעט כשעה. בתמונה הבאה ניתן לראות את מספר נקודות החיים שהיו לי (39) ושהשעה היתה 16:27:

לקחתי את אותה בקשה שמופיעה למעלה והשתמשתי ב-Intruder שוב כדי לבצע פעולה של שליחת כמה מאות בקשות כמו בדוגמה הקודמת, התחלתי לצפות בבקשות יוצאות ומקבלות תגובה תקינה משרת המשחק:

אפשר לראות שבשעה 16:31 כבר היו לי 72 מתוך 72 נקודות חיים - זאת אומרת שהגעתי למקסימום נקודות החיים שאמורות להיות לי בתוך 4 דקות בלבד, ולא בתוך 50 דקות כמו שאמור לקרות בפועל על פי הלוגיקה של המשחק:

ולא רק אני קיבלתי את "בוסט" העלאת נקודות החיים הזה, אלא כל מי שלא היה בזמן קרב פעיל!, הקוד שנמצא בדף lifechange.php אחראי לתהליך העלאת חיים לכל מי שנמצא בשרת!

מה המשמעות של דבר כזה?

כל שחקן יכול לעלות לעצמו ולכל השחקנים את נקודות החיים שלהם, מה שלא מצריך מהם לחכות זמן כדי לעלות חזרה את נקודות החיים האלה, במקום להמתין שעה עד הקרב הבא - הם יוכלו לעשות כמה עשרות קרבות בשעה (!) כי אין צורך להמתין, מה שיגרום כמובן לעליית רמות בצורה מוגברת ויצריך מהם הרבה פחות השקעה של זמן, דבר כזה גם פוגע מאוד במודל הכלכלי של משחק שמוכר לדוגמה חפצים להעלאת נקודות חיים, בלא מעט מקרים ובעיקר במשחקים שהם "חינמיים" - זאת אומרת לא נדרש לשלם כדי להתחיל לשחק במשחק, המודל הכלכלי עובד על רכישות בתוך המשחק, במידה והמשחק מוכר לדוגמה שיקוי העלאת חיים מיידי ב-3$, למה שמישהו ישלם כמה דולרים שאפשר פשוט לשלוח בקשות HTTP?

מה המיטגציה?

כדי למנוע משחקנים לעלות נקודות חיים לכולם ישנים כמה פעולות שנוכל לעשות:

- להסיר את הפניה לדף הזה מתוך טריגר שקורה ששחקן מבצע פעולה מסוימת (פעולה זו קרתה כאשר ניסיתי לקנות חפץ), בכך הדף הזה לא ייקבל בקשות משחקנים, אל הדף הזה אמור לפנות אך ורק שרת המשחק דרך Cron job (משימה מתוזמנת).
- בדומה למקרה הקודם - הייתי מוודא שרק שרת המשחק יוכל לפנות לאותו הדף כדי להפעיל את הפונקציונליות שלו גם ברמת ה-IP.
- הוספת Token מסוים ב-HEADER כדי לוודא שזה אכן הגורם שרשאי לבצע את הרצת הקוד.

מימוש בפועל (Patch):

ב-Patch הקודם שביצענו למעלה ראינו איך אפשר לוודא מאיזו כתובת IP מגיעה בקשת ה-HTTPS, נכון שזה מוריד משמעותית את האפשרות לממש את הניצול שראינו עכשיו אבל זה עדיין לא הרמטי מספיק, אז הפעם בנוסף לזה - לא רק שנוסיף ולידציה על ה-IP, אנחנו גם נרצה לוודא שיש Header ספציפי עם ערך שאותו הגדרנו מראש, ורק אם נקבל את כל הנתונים האלו - הקוד יבצע את הפעולה הנדרשת.

לצורך הדוגמה החלטתי שה-Header שנרצה לוודא שמגיע אלינו נקרא Access-SuperKey והערך שלו צריך להיות: Secret. גם במקרה הזה PHP עוזר לנו עם פונקציה שאיתה נוכל לקחת את כל ה-Headers הנשלחים אלינו בבקשה, אפשר לבצע את זה באמצעות getallheaders(), בשלב הראשון ניצור משתנה שיכיל בתוכו את כל ה-Headers שמגיעים מהבקשה:

```
// Get all the HTTP request headers
$headers = getallheaders();
```

כעת נרצה לשלב גם את בדיקת ה-IP שביצענו למעלה, אבל גם שמתקבל ה-Header שהגדרנו שהוא Access-SuperKey והערך שלו הוא Secret, כך ניתן לממש את זה בקוד:

```
<?php
    if ($_SERVER['REMOTE_ADDR'] == '<OUR_SERVER_IP_ADDRESS>') {
    // Get all the HTTP request headers
    $headers = getallheaders();
    if (isset($headers['Access-SuperKey']) && $headers['Access-SuperKey'] == 'secret') {
        function username($id) {
```

אז בעצם אם נתרגם את הקוד הזה ללוגיקה מה שקורה בפועל, זה שאנחנו מוודאים מאיזו כתובת IP הגיעה אלינו הבקשה, במקרה הזה רק אם היא הגיע מ-IP של השרת שלנו, הקוד ימשיך הלאה. לאחר מכן מתבצעת הבדיקה השניה שלנו לגבי ה-Header, אנחנו נקבל את כל ה-Headers בבקשה, ובמידה ויש בבקשה Header בשם Access-SuperKey שהערך שלו הוא secret, רק אז הקוד ימשיך הלאה לביצוע הפעולות שהוא אמור לעשות.

האם שחקנים ניצלו את האפשרויות האלו בעבר?

המשחק היה מושבת במשך 3 שנים עד שהעלתי אותו מחדש ככה שאין לי את הלוגים מהזמן שהמשחק היה פעיל, אז כנראה שלא אדע לעולם, אבל לא נדרש ידע טכני מתקדם כדי לבצע את מה שהצגתי למעלה ולכן אני חושב שזה משהו שכנראה קרה. האמת? אני יכול להבין אותם, כולנו רוצים להתקדם מהר ולהיות טובים יותר במשחק כזה או אחר.

בנוסף למה שהצגתי למעלה היו לא מעט שחקנים שהיו משתמשים ב-Auto Clicker כדי לעשות קרבות אוטומטית, השחקנים היו מקליטים את הקליקים שלהם במהלך קרב ואז פשוט חוזרים על אותם הקלקות ב-Loop, ברגע שראיתי שיש שחקנים שנמצאים במשך המון שעות ברצף בקרבות שמתי לב שמשהו לא תקין, לא הגיוני ששחקן יושב 15 שעות ביום כדי לעשות קרבות.

אז כדי להבין מי באמת מאוד משקיע (כי היו גם שחקנים כאלה) ומי משתמש ב-Auto Clicker הטמעתי סקריפט של חברה בשם Hot jar, המוצר שלה במקור הוא על מנת לזהות סיבה לנטישת משתמשים באתר וביצוע אנליזות. השירות של Hot jar מקליט את המסך מהרגע שגולש נכנס לאתר (רק את דף האתר בזמן גלישה, לא את מסך המחשב של המשתמש כמובן) ועוקב אחרי תנועות העכבר וההקלקות שלו, ממש מלווה אותו לכל אורך השהות שלו באתר, מציג באיזה דפים הוא שהה וכמה זמן. במקביל הוא גם מייצר "מפת חום" של הקליקים שביצע אותו גולש.

פעם ביום הייתי עובר על ההקלטות האלו וראיתי שיש שחקנים שהעכבר שלהם זז בצורה מדויקת ברמה לא אנושית, לדוגמה מקליק בדיוק אבל בדיוק באותו המיקום במשך כמה שעות, ואפשר לזהות די בבירור שזה לא משהו שנעשה בצורה ידנית, לצערי היו לא מעט כאלו, אז אני מעריך שהצליחו "לרמות" את המערכת גם בעוד שיטות.

סיכום

היה לי קצת מוזר אבל גם כיף למצוא אפשרויות לעקוף ולתמרן את הקוד שאני עצמי כתבתי לפני כמה שנים. אמנם לפני 10 שנים בערך בתחילת הדרך של פיתוח המשחק עוד לא הבנתי עד כמה פשוט להגיע לאותם דפים, אני חושב שגם אפשר לראות עד כמה פעולות כמו שראינו למעלה אומנם לא מהוות "סיכון אבטחתי" פר-אקסלנס, אבל גם ההשפעות שלהן על המשחק ועל המודל הכלכלי שלו לא פחות חמורות. הייתי מפתח במיינד-סט שונה לגמרי אם הייתי צריך לכתוב מחדש את המשחק כיום.