Digital Whisper
Technologic Papers

חלק ג - Hacking Games For Fun And (mostly) Profit

מאת d4d

הקדמה

מטרת סדרת מאמרים זו הינה להציג את השלבים שעברנו בעת מחקר המשחק Worms World Party, במטרה לכתוב שרת פרטי למשחק זה. עד כה הצגנו את שלבי הקמת המעבדה לטובת ביצוע המחקר, ואף את שלבי המחקר המתקדמים:

- את ההצפנה בה מפתחי WWP השתמשו בכדי לבצע אימות משתמשים לשרת ה-IRC.
- את השלבים ואת תהליך הרברסינג למנגנון שבתוך המשחק בכדי לזייף את ה-Challenge Response.
- ניתחנו את שיטת ההצפנה שבה השתמשו לרשימת המשחקים והצגנו קוד שיודע להציג את המשחק ללא הצפנה.
- דיברנו על איך נראה מבנה המשחק ב-WWP בזיכרון.

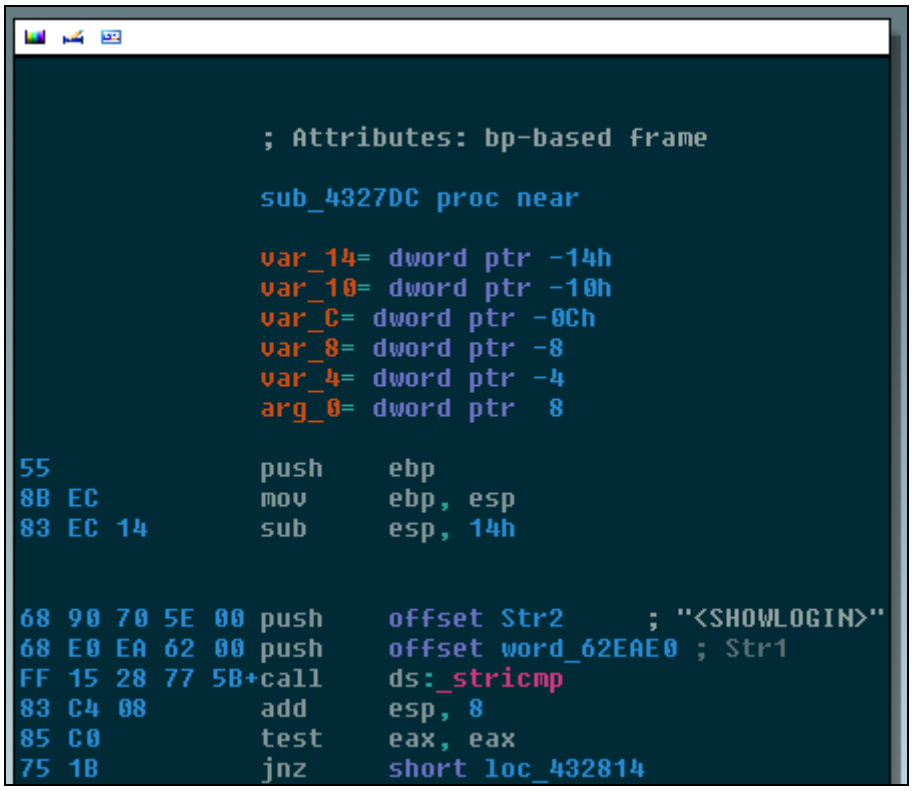
מאמר זה הינו החלק השלישי של סדרת מאמרים זו, מאמר זה מדבר על הנושאים הבאים:

- תיאור הפרוטוקול של WWP.
- יצירת אמולטור לשרת WormNET2.

תיאור הפרוטוקול

הפרוטוקול של WWP דומה ל"פקודות" HTML ליצירת דפי אינטרנט, אך הפקודות מעט שונות. כדי למצוא את כל הפקודות הקיימות ב-WWP הסתכלנו ב-IDA Pro על הפונקציה שבה בודקים את סוג הפקודה שהתקבלה.

להלן צילום מסך של קטע מפונקציה זו:

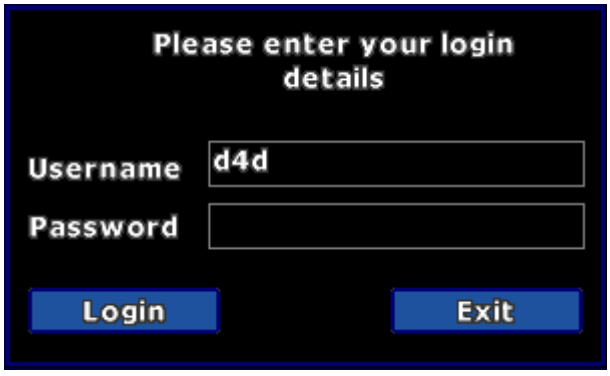


```
                ; Attributes: bp-based frame

                sub_4327DC proc near

                var_14= dword ptr -14h
                var_10= dword ptr -10h
                var_C= dword ptr -0Ch
                var_8= dword ptr -8
                var_4= dword ptr -4
                arg_0= dword ptr  8

55              push    ebp
8B EC           mov     ebp, esp
83 EC 14        sub     esp, 14h


68 90 70 5E 00 push    offset Str2     ; "<SHOWLOGIN>"
68 E0 EA 62 00 push    offset word_62EAE0 ; Str1
FF 15 28 77 5B+call    ds:_stricmp
83 C4 08        add     esp, 8
85 C0           test    eax, eax
75 1B           jnz     short loc_432814
```

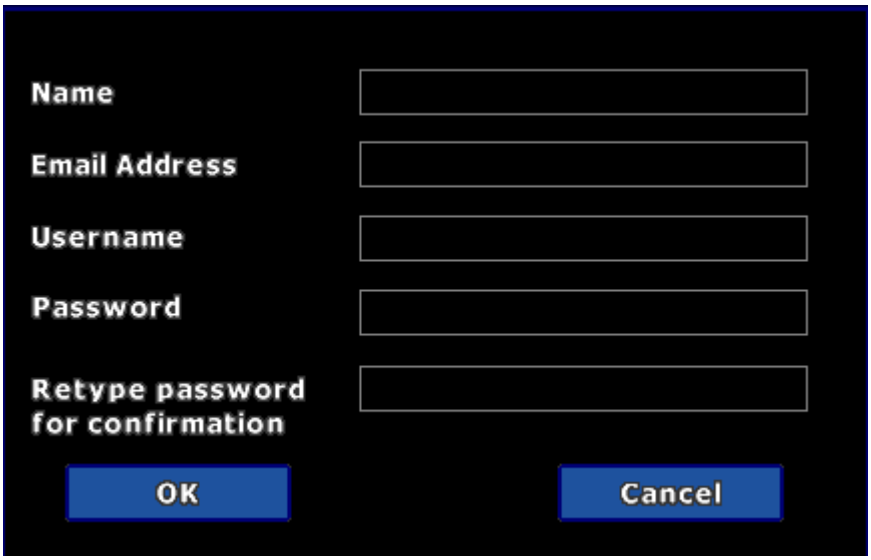
בדוגמא זו ניתן לראות השוואה לפקודה <SHOWLOGIN>, במידה וזו לא הפקודה שהתקבלה - הפונקציה תעבור לבצע השוואה עם הפקודה הבאה.

בכדי להגיע למסקנה מה כל פקודה עושה השתמשנו בסניפר (WireShark) כדי לבדוק מתי יש שימוש בפקודות החשובות. רשימת הפקודות החשובות הן:

- הפקודה <SHOWLOGIN> תציג במשחק את החלון הבא:



- הפקודה <SHOWNEWUSERENTRY> משמשת ליצירת משתמש חדש:



חלק מהשדות שמופיעים בתהליך ההרשמה אינם בשימוש באמולטור. נראה שמפתחי המשחק "התבלבלו" והם שולחים פרטים שלא מופיעים בהרשמה כלל.

כאשר נשלחת בקשת ה-GET לשרת, על מנת לבצע את ההרשמה נשלחים הארגומנטים הבאים:

```
[Username] => d4d
[Password] => 12345
[Email] => sdsdsd
[Surname] => abcddf
[Address] =>
```

- הפקודה לביצוע חיבור לשרת ה-IRC הינה הפקודה הבאה:

```
<CONNECT [IRCServer] IRCPORT=[port] IRCUSER=[user] IRCPASS=ELSILRACLIHP>
```

שדה	תיאור
IRCServer	כתובת לשרת IRC
IRCPORT	הפורט
IRCUSER	שם משתמש
IRCPASS	סיסמא שניתן לשנות בפקודה

- הפקודה לביצוע Challenge Response היא הפקודה: ANSWER אותה ראינו בחלק א' והרחבנו עליה בחלק ב':

```
<ANSWER CVttpTpl5cd7dP+0Ae1WIr71jqpXllw2jXE1qmAyQb0gEwFZ>
```

- הפקודות להצגה של רשימת המשחקים הם הפקודות הבאות:

```
<GAMELISTSTART>
<GAME … >
<GAMELISTEND>
```

פקודה	תיאור
GAMELISTSTART	פקודה המציגה את תחילת רשימת המשחקים
GAMELISTEND	פקודה המציגה את סוף רשימת המשחקים
GAME	כל משחק מופיע בפקודה חדשה והוא חייב להיות מוכל בין GAMELISTSTART ל GAMELISTEND

- הפקודה WEBADDRESS הינה הפקודה בה קובעים את התיקייה של השרת.

לדוגמא:

```
<WEBADDRESS /test/>
```

בביצוע פקודה זו השרת ייגש לקבצים הנמצאים בתיקיה test, בכתובת זו: http://myurl.com/test.

- הפקודה SCHEME מתבצעת כאשר אנו נכנסים לערוץ בשרת. הסבר מפורט על הפרמטרים שמקבלת הפקודה SCHEME ניתן למצוא באתר (http://worms2d.info/WormNET_(Worms_Armageddon משום שקיים מידע על מה קורה ב-WA זה חסך קצת עבודה לבדוק מה כל מוד בפקודה זו עושה.

ישנן פקודות נוספות למשחק, אך הן לא חשובות לאמולטור שלנו.

בחלק א' הזכרנו שהיו דרגות ב-WA ושחברת Team17 הורידו אותן, בתחילה תכננו לעשות שרת שיהיה בו מימוש גם לדרגות ב-WWP אך החלטנו לא להפיץ אותו מהסיבות שיפורטו בחלק הבא.

דרגות - אבטחת מידע

מנגנון הדרגות אינו מאובטח כלל, כל משתמש שיזייף פאקט ששולח ניצחון יקבל אותו, גם אם המשחק לא בוצע כלל. על מנת שהדרגות יעבדו בצורה שלא יהיה קל לכל ילד בן 10 לרמות צריך לשנות את המודול. הבעיה במודול של חברת team17 היא הבעיה הבאה:

team17 סומכים על המשתמש שישלח את התוצאה של מי ניצח לשרת. מי שולח את הבקשה? המשתמש שיצר את המשחק, הדבר דומה לקזינו שבו בעל הבית תמיד מנצח ולכן החלטנו לא לפרסם את המימוש לדרגות.

למה עשו אז הצפנה ל-WWP?

הסיבה שעשו את ההצפנה זה בכדי להקשות על משתמשים לא מורשים להיכנס לשרת משחק בלי עותק של המשחק ושלא יוכלו לקרוא את רשימת המשחקים. הסיבה שהשתמשו בהצפנה של RSA בכדי להצפין את המשחקים הייתה דרך טובה להקשות על הקמת שרת פרטי בקלות.

בכדי ליצור משחק אנו צריכים להחליף את המפתחות RSA של השרת על ידי מודול שנכתב שתפקידו לשנות את המפתח הפומבי שנמצא בקליינט (בשרת יהיה מפתח private שבעזרתו נחתום את המידע שמוצפן ב-Twofish)

המפתחות של ה-Twofish לא שונו, המפתחות זהות בהצפנה סימטרית אז אין טעם לשנות גם אותם.

יצירת אמולטור ל-WWP

לאחר שניתחנו את ההצפנה שהייתה בשימוש ב-WWP שנתנה לנו את היכולת להבין איך לפענח את רשימת המשחקים ולפתור את ה-Challenge Response לשרת IRC בלי עותק של המשחק הגענו לשלב המתבקש שהוא יצירת האמולטור.

כדי להבין את שאר החלקים החסרים ב-WWP היה שימוש ב-WireShark כדי לבדוק איפה להשתמש בכל פקודה.

דוגמא לחבילה ניתן לראות בתמונה הבאה:

```
GET http://wormnet2.team17.com:80/wwpweb/welcomeloginform.php
HTTP/1.0
User-Agent: T17Client/2.0
Pragma: No-Cache
UserServerIdent: 1
Counter: 1424867105

HTTP/1.1 200 OK
Date: Wed, 25 Feb 2015 12:19:54 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
Set-Cookie: PHPSESSID=5ja3uncu2dh7fk8kguht1gjtu3; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 627
Connection: close
Content-Type: text/html

<CHECK DD4nxeKM9Lwni4rd
+7VUIXwea8PyPIqgQW8Xg5S2FTPPiRdRl7kDVsDoLdFrsWk8L3hvbo1UXqdL1w0=>
<WEBADDRESS /wwpweb/>
<EXTENSION .php>
<FONT Size=1 Colour=0>                                    Welcome to
WormNet2<BR></FONT>
<br><br><FONT Size=0 Colour=0>                                          You
can use any username / password to log in<BR></FONT>
```

כאשר ננסה ליצור משחק חדש יקרה הדבר הבא:

```
GET http://wormnet2.team17.com:80/wwpweb/Game.php?
Cmd=Create&Name=d4d&HostIP=          !&Nick=d4d&Chan=AnythingGoes&Loc=25&Type=0 HTTP/1.0
User-Agent: T17Client/2.0
Pragma: No-Cache
Cookie: PHPSESSID=5ja3uncu2dh7fk8kguht1gjtu3
FileResult: 77
FileData:
151F1F1F16133434344D4D4D4D4D4D4D4D4D4D4D4D0000000000000000000000000000000000000000
0000000000000000000000000000000003942424242424242424242424242424242424242424242424242
4242424242424242424242
UserLevel: (0
UserServerIdent: 1
Counter: 1424867473

HTTP/1.1 200 OK
Date: Wed, 25 Feb 2015 12:26:01 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
SetGameId: 3758392
Vary: Accept-Encoding
Content-Length: 10
Connection: close
Content-Type: text/html

<NOTHING>
```

ניתן לראות שהשרת מחזיר למשתמש הידר בשם SetGameId בשביל ליצור משחק, אם ה-Header הנ"ל לא יופיע, המשחק לא ייווצר.

כאשר כותבים אמולטור לשרת פרטי, צריך לנסות לשחזר את מה שמקבל השרת המקורי, ובמידה ומשהו לא עובד צריך להשוות את הקוד שלנו עם השרת המקורי או לקרוא בקוד איך המשחק ניגש למידע מסויים.

לדוגמא כאשר אנחנו מנסים להיכנס לערוץ, אם יהיה חסר לנו הדף RequestChannelScheme.php המשחק ייתקע, אנו חייבים להחזיר לו תשובה, אז במידה ולא הוגדר סוג של ערוץ אנו נקבע את הערך הדיפולטיבי:

```
<SCHEME=Pf,Be>
```

כדי לדעת איזה ארגומנטים הקליינט שולח לשרת כתבנו סקריפט ב-PHP שידפיס לתוך קובץ בשם logs.txt את כל מה שנשלח ב-GET על ידי קטע הקוד הבא:

```
$fp = fopen('logs.txt', 'a');
fwrite($fp,"Login.php\r\n");
fwrite($fp, print_r($_GET, TRUE));
fclose($fp);
```

סקיצה בסיסית של טבלאות מה-Database

כעת אנו צריכים ליצור Database שישמור את כלל הנתונים. ה-Database צריך להכיל את הדברים הבאים:

- נתונים על המשתמשים בשרת.
- רשימת המשחקים בכל הערוצים הקיימים בשרת.
- רשימת כל הערוצים שיש בשרת וסוג הערוץ.
- במידה ויהיו דרגות הצטרכו להוסיף את הנתונים מי ניצח והפסיד.

הטבלה USERS - טבלה זו שומרת פרטים על המשתמש, כגון ניקוד ובדיקה אם המשתמש חסום או לא לפני כניסה לשרת.

שם שדה	תיאור
UserID	מפתח ראשי לזיהוי השם משתמש
PlayerType	סוג מד החיים אשר יופיע למשתמש בזמן המשחק. אם הערך יהיה 2 למשתמש תיהיה אש כחולה. אם הערך הוא 1 תופיע למשתמש אש אדומה.
PlayerScore	נקודות שיש לכל משתמש, אם אין דרגות ניתן לוותר על השדה.
PlayerLevel	הדרגה שתהיה למשתמש בערוץ. לדוגמא, הערך: 5 יציג את הדרגה: 
Banned	במידה ונרצה לחסום את המשתמש. ערך זה יכיל: 1, אחרת: 0.
UserName	הכינוי של המשתמש (שדה זה מוגדר כ-UNIQUE).
Password	הסיסמא של המשתמש
IPAddress	כתובת ה-IP של המשתמש
Email	המייל של המשתמש (שדה זה מוגדר כ-UNIQUE)
Access	1 במידה ולמשתמש יש גישה לפאנל ניהול של המערכת

הטבלה HostGames - בטבלה זו נשמרים כל המשחקים שנוצרו בשרת:

שם שדה	תיאור
GameID	מפתח ראשי לזיהוי השם משתמש
UserID	ה-ID של המשתמש
GuestID	ID שנקבע במשחק עם דירוג, ניתן לוותר על השדה הזה אצלנו
Nick	השם של יוצר המשחק
HostIP	כתובת ה-IP של יוצר המשחק
Name	השם של המשחק

Loc	המדינה של המשתמש
Type	המספר יכיל 0, במוד של דרגות, מכיל מספר 1-4
ServerUsage	1 אם ניתן ללחוץ על המשחק, 0 אם אי אפשר.
Chan	הערוץ שבו נוצר המשחק

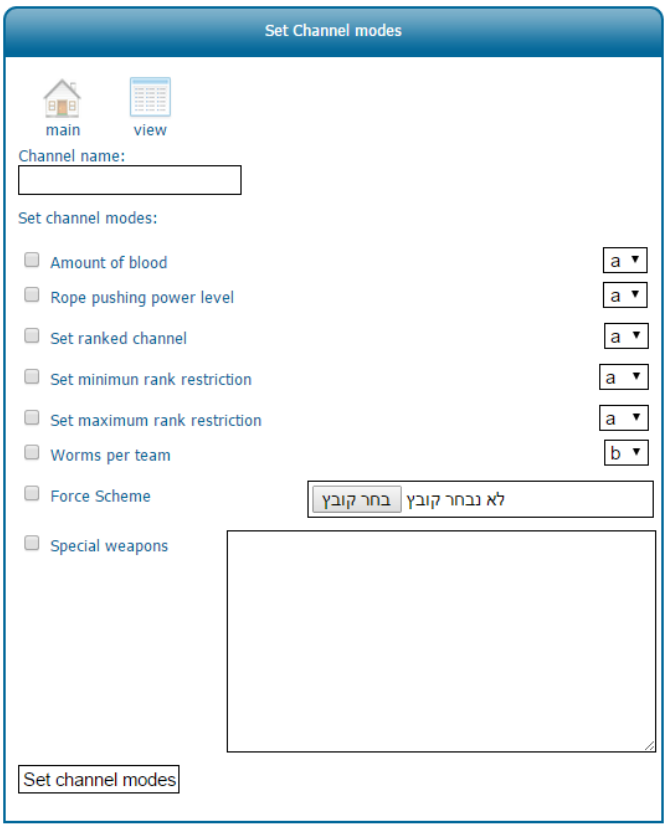
הטבלה EncryptedHosts - בטבלה זו נשמרים כל המשחקים שנוצרו ואחרי 2 דקות הם נמחקים מהרשימה ולא ניתן להיכנס למשחק.

שם שדה	תיאור
Id	מפתח ראשי לזיהוי המשתמש המוצפן
EncryptedGame	המידע של המשחק עם הצפנה ומקודד ב-base64 כפי שהוסבר בחלק ב'
GameID	ה-ID של המשחק
Channel	הערוץ בו המשחק נוצר
Time	זמן שבו נוצר המשחק

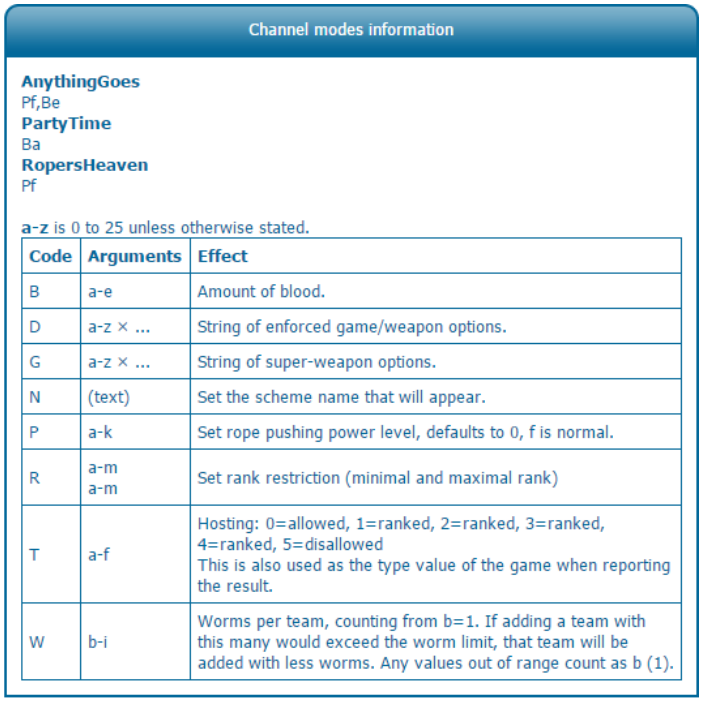
טבלה SCHEMES - בטבלה זו נשמרים המודים לכל ערוץ שקובעים את סגנון המשחק.

שם שדה	תיאור
Id	מפתח ראשי לזיהוי הערוץ
Channel	השם של הערוץ (שדה זה מוגדר כ-UNIQUE)
Modes	המודים שקובעים את סגנון המשחק

על מנת לנהל את השרת עצמו, כתבנו פאנל ניהול, הוא נראה כך:



בחלק זה אנו קובעים את המודים בהם הערוץ שלנו יתמוך, המוד לדרגות גם מופיע אך נתונים של מי ניצח לא יישמרו, כי ה-database לא טורח לשמור אותם. בתמונה הבאה מופיעים סוג ותיאור המודים:



Code	Arguments	Effect
B	a-e	Amount of blood.
D	a-z × ...	String of enforced game/weapon options.
G	a-z × ...	String of super-weapon options.
N	(text)	Set the scheme name that will appear.
P	a-k	Set rope pushing power level, defaults to 0, f is normal.
R	a-m a-m	Set rank restriction (minimal and maximal rank)
T	a-f	Hosting: 0=allowed, 1=ranked, 2=ranked, 3=ranked, 4=ranked, 5=disallowed This is also used as the type value of the game when reporting the result.
W	b-i	Worms per team, counting from b=1. If adding a team with this many would exceed the worm limit, that team will be added with less worms. Any values out of range count as b (1).

נתונים אלו ידועים מ-WA ולכן עליהם לא בוצע רברסינג כפי שהוסבר קודם בסוגי הפקודות.

בוצע רברסינג על 2 מודים בהם אין יותר מדי מידע (הם מוד D ומוד G). מוד D קובע את סוג המשחק שישחקו בערוץ ואותו אי אפשר לשנות. נכתב בפאנל ניהול קוד ב-PHP שאליו מעלים את ה-scheme ותפקידו להמיר את הקובץ למידע שאותו WWP מבין.

דוגמא ניתן לראות בתמונה הבאה:

AnythingGoes	<SCHEME=Be,Pf,We,Daddabbaaaadddabbbfaabcb cbaaaaaaaaabakcaabcbbfcabkcabdaabbcbbcca bacabkcaakcaakcaaacacccabbcfbbchbccabkca akcaabcaakcaaccabdcabdcabdcabcbabcaabcaa bcaabbcabbccbbccbacacacacacacacacacacaca cacacacacaaaaaaaaaaaaaaaaaaaaaa,NAnyth ingGoes,Ghakafsck>		

בתמונה זו רואים את התווים a-z במוד D שאחראים על סוגי הנשק, כמה שניות יהיה בכל תור למשחק וכו'. מוד G קובע נשקים מיוחדים אותם אי אפשר לקבוע ישירות במשחק, אין אפשרות כזו. על מוד G בוצע רברסינג כדי לדעת איזה מוד לבקש בשביל לקבל נשק מיוחד שאי אפשר לקבוע. הפונקציה עליה צריך להסתכל ב-IDA PRO הינה הפונקציה הבאה:

```
0049EF0D                    handleGMode:                            ;
0049EF0D                                                            ;
0049EF0D C7 85 FC FE FF FF+     mov     [ebp+specialWeaponOffset], 0
0049EF17 C7 85 04 FF FF FF+     mov     [ebp+var_FC], 0
0049EF21 C7 85 00 FF FF FF+     mov     [ebp+offset], 1
0049EF2B EB 0F                  jmp     short loc_49EF3C
0049EF2D                    ; ---------------------------------------
```

הרשימה המלאה נמצאת כאן:

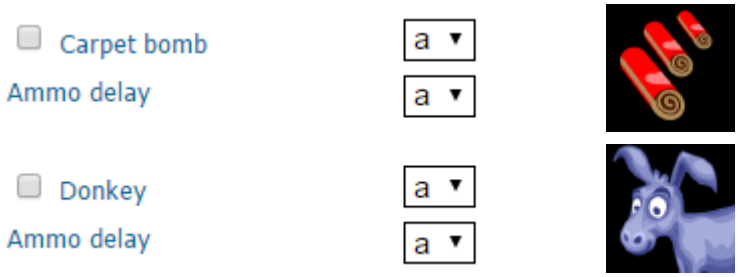
Special weapons information

a-z is 0 to 25 unless otherwise stated.
k in Amount for unlimit weapons
k in delay for disable weapon unless otherwise stated

Code	Amount	Delay	Effect
ha	a-k	a-b	Crate shower, for unlimit Crate shower turns it would be set to k.
fs	a-b	a-k	Crate spy, how much turns Crate spy would be set, k for always.
fq	a-k	a-k	Invisibility.
eq	a-k	a-k	Armageddon.
cq	a-k	a-k	Girder pack.
fb	a-k	a-k	Carpet bomb.
dy	a-k	a-k	Donkey.
ba	a-k	a-k	Earth quake.
ga	a-k	a-k	Freeze.
gb	a-k	a-k	Magic bullet.
dq	a-k	a-k	MB bomb.
da	a-k	a-k	Mine strike.
dt	a-k	a-k	Ming vase.
db	a-k	a-k	Mole Squadron.
dz	a-k	a-k	Nuclear Test.
bt	a-k	a-k	Mail strike.
cz	a-k	a-k	Salvation Army.
eb	b-k	a-k	Scales Of Justice.
et	a-k	a-k	Select worm.
fa	a-k	a-k	Sheep strike.
bj	a-k	a-k	Suicide Bomber.
ec	a-k	a-k	Super banana bomb.

כל 2 אותיות מרכיבות נשק מיוחד אותו לא ניתן לקבוע דרך המשחק. שתי אותיות האלה מחשבות את האינדקסים במבנה שקובע את הנשקים במשחק, במקום לנחש אינדקסים נכתב קוד בפייתון שימצא לנו את הצירוף אותיות הנכון שיתן לנו את הנשק המבוקש. הקוד יצורף בסוף המאמר.

בפאנל ניהול נכתב כלי שיודע להמיר לקודים הדרושים את הנשקים שאנו מבקשים וידע להביא את הנשקים. דוגמא לבחירת חלק מהנשקים בפאנל:



בשביל שהשרת יעבוד כמו שצריך היינו צריכים להקים שרת IRC אליו הוא יוכל להתחבר, בגדול כל שרת IRC מתאים אך רצינו להכניס לשרת שלנו את ה-Challenge Respons כמו שעשו בחברת team17. אנו השתמשנו ב-hybrid-ircd בתור השרת IRC שלנו והוספנו לשם מודול ושינינו כמה core files.

אחד המפתחים הראשיים של הפרויקט הסכים לתת לנו תמיכה ולומר איזה קבצי core כדאי לשנות.

הסיבה שלא השתמשנו ב-unrealIRCd זה בגלל שאין תמיכה למי שמעוניין לכתוב מודולים, ניסינו לפנות אליהם במשך חודשיים ואף אחד לא היה זמין.

השרת IRC לא צריך Challenge Response, אך אם כבר מבינים איך המנגנון עובד אז אין טעם לא להוסיף את האופציה הנ"ל. זה רק משפר את האבטחה וכמובן לנסות להיות כמה שיותר קרוב למקור. אנו ערכנו את קבצי ה-core הבאים ב-hybrid-ircd:

- client.c
- user.c
- client.h

ב-client.h נוספו הדגלים הבאים, בהם יש שימוש למימוש המודול שלנו m_authpong.c

```
#define SetAuthPing(x)          ((x)->flags |= FLAGS_AUTHPING_SENT)
#define HasAuthPing(x)          ((x)->flags & FLAGS_AUTHPING_SENT)
#define ClearAuthPingSent(x)    ((x)->flags &= ~FLAGS_AUTHPING_SENT)
```

ב-client.h הוספנו עוד שדות למבנה Client בשם:

```
char *authping;
char *url;
```

ב-client.c הוספנו את השדות שיש לשחרר להם את הזיכרון בפונקציה free_client():

```
MyFree(client_p->authping); // added authping to free
MyFree(client_p->url); // added
```

ב-user.c הוספנו בפונקציה register_local_user את הקטע קוד הבא:

```
if(!HasAuthPing(source_p))
{
                  char url[5][100] = {"wormnet.team17.com$1",
"wormnet.team17.com$2", "wormnet.team17.com$3", "wormnet.team17.com$4",
"wormnet.team17.com$5"};
                  uint8_t key[] =
"\xD5\x2D\x31\x08\xFB\x0F\x54\x9E\x6D\x7A\x0F"
"\xFD\xEE\xDC\x21\x9A\xD4\xA6\x84\x24\x6D\x61\xFA\x8A\xAE\x98\x96\xA1\xF
1\x63\x1A\x8D";
      uint8_t text[100];
      uint8_t iv[16];
      uint8_t hash[20];
            // choose the seed for the user
      memset(iv, '\0', 16);
      srand(time(NULL));
      for(int i = 0;i < 16;i++)
            iv[i] = rand() % 0xFF;
            // choose the secret value for the challenge response
 srand(time(NULL));
  int pickUrl = rand()%5;
      source_p->authping = MyCalloc(100);
      source_p->url = MyCalloc(100);
      // produce the encryption and hash it
      memcpy(text, iv, 0x10);
      memcpy(text + 0x10,url[pickUrl], strlen(url[pickUrl]));
      MCRYPT td = mcrypt_module_open("twofish", NULL, "cfb", NULL);
      mcrypt_generic_init(td, key, 0x20, iv);
      mcrypt_generic(td, text, 36);
      mcrypt generic deinit (td);
      mcrypt_module_close(td);
      computeRIPEMD160(text, 36, hash);
      memset(text,'\0',100);
      for(int i = 0;i < 20;i++)
            sprintf((char*)text + i*2, "%02X", hash[i]);
      memcpy(source_p->url, url[pickUrl], strlen(url[pickUrl]));
      memcpy(source_p->authping, text, strlen(text));
sendto_one(source_p, "AUTHPING %s %s ", source_p->url, source_p-
>authping);
      return;
  }
```

ב-WWP יש אפשרות ליצור ערוצים עם סיסמאות ככה שרק אנשים ספציפיים עם סיסמא יוכלו להיכנס, זה פיצר אותו אין ב-WA (אין אפשרות להכניס סיסמא בכלל בקליינט). במידה ולא קיבלנו Response מהמשתמש על ה-Challenge הוא מתנתק מהשרת אחרי חצי דקה.

בנוסף, ערכנו גם את ה-topic שלא יציג את המודים בכותרת בגלל שב-WWP/WA המספרים בין 00 ל-06 קובעים את האייקון של הערוץ במשחק, במידה ולא יוצגו בהתחלה מספרים אלה, ייקבע כברירת מחדל האייקון 06. הקוד ל-m_authpong.c יצורף בסוף המאמר.

סיכום

ביצוע מחקר לטובת הקמת שרת פרטי למשחק מחשב רציני זו בהחלט לא עבודה קלה. ראינו כי ראשית יש צורך לבצע ניתוח להצפנה שיש בפרוטוקול איתו מדברים השרת והלקוח. רק לאחר שמבינים את ההצפנה ויודעים איך לפענח את התקשורת ניתן לגשת לשלבים האחרים שהם ניתוח הפרוטוקול עצמו.

לפני המימוש של השרת, ישנו הצורך לכתוב Packet Logger שידע לקחת את כל חבילות המידע שהמשחק שולח לשרת ואז לנסות לסווג אותם לפי סוג ותפקיד, בכל משחק זה שונה.

במאמר זה הראינו את השלבים הבסיסיים אותם צריך בשביל לממש שרת פרטי, קודם הקמנו את סביבת העבודה וכתבנו קבצי DLL שיאיצו את העבודה, ניתחנו את ההצפנה והבנו איך היא פועלת, לאחר מכן בדקנו עם Sniffer את שאר הפקודות בשרת מה שלא היה ידוע מהגירסה הקודמת WA. ולבסוף כתבנו את הקוד של השרת וביצענו טסטים לראות שהקליינט מגיב למה שהשרת שלנו שולח.

אני מקווה שנהנתם מסדרת מאמרים זו ושלמדתם ממנה רבות. הקוד המלא לשרת יפורסם ב-github וב-bitbucket בהמשך, כרגע סגור מכיוון שאנו מעוניינים להוסיף עוד מספר פקודות. את הקוד שהוצג בחלק זה ניתן להוריד מהקישור הבא:

http://www.digitalwhisper.co.il/files/Zines/0x3B/WWP3.rar

על המחבר

d4d עוסק ב-Reverse Engineering ואוהב לחקור משחקי מחשב והגנות. לכל שאלה או יעוץ ניתן לפנות אליו בשרת ה-IRC של NIX, בערוץ: #Reversing. בכתובת האימייל: llcashall@gmail.com. או דרך האתר: http://www.cheats4gamer.com