Digital Whisper
Technologic Papers

חלק א' - Hacking Games For Fun And (Mostly) Profit

מאת d4d

הקדמה

מאמר זה עוסק בניתוח הפרוטוקול וההצפנה של המשחק Worms World Party (משחק אסטרטגיה מרובה משתתפים שיצא בשנת 2001 ע"י חברת Team17) במהלך המאמר אציג את אופן הקמת סביבת המחקר, את שלבי ההכנה לקראת ניתוח הפרוטוקול ואת פרוטוקול התקשורת של המשחק. כאמור, במאמר אציג ניתוח של המשחק Worms World Party, אך חשוב להבין, כי ברוב המקרים, אין השלבים שאציג במאמר שונים מניתוח פרוטוקולים של משחקי מחשב אחרים, הפרוטוקול ברוב המקרים יהיה שונה, אך אופן המחקר ורוב שלביו יהיו זהים. מטרת המחקר הינה לחקור את המשחק ופרוטוקול ההזדהות שלו על מנת לכתוב שרת משחק פרטי, שיכלול פיצ'רים שאינם קיימים בגרסתו המקורית.

לפני שנתחיל, ישנן מספר הגדרות שנגדיר פה שיהיו בשימוש בהמשך המאמר:

- Worms world party - **WWP**
- Worms Armageddon - **WA**
- **Team17** - החברה שהוציאה את המשחק

זהו חלק א' במאמר, מאמר זה מדבר על הנושאים הבאים:

- הסיבה לביצוע ניתוח על המשחק WWP.
- היסטוריה מה שונה WWP מהמשחק הקודם WA.
- הקמת סביבת העבודה וכלים לביצוע המחקר.
- הרצה ראשונית של סביבת העבודה.

החלק הבא במאמר ידבר על הנושאים הבאים:

- סוג ההצפנה שבה נעשה השימוש בפרוטוקול.
- ניתוח איך עובד מנגנון האימות של השרת עם המשחק.

למה WWP?

WWP הוא משחק מאוד מפורסם שיצא בשנת 2001, מדובר בגרסא הרביעית של סדרת המשחקים Worms, ובגרסא השניה שבה היה ניתן לשחק עם שחקנים אחרים דרך האינטרנט. בגרסא זו Team17 החלו להשתמש בהצפנה מאוד מעניינת על מנת לאמת את המשתמשים בעת הכניסה לשרתי המשחק. שימוש נוסף אשר נעשה בהצפנה הוא הצפנת רשימת המשחקים הקיימים בשרת, כך שרק מי שיש לו את העותק של המשחק יוכל לראות את רשימת המשחקים.

בשנת 2001 היה נדיר מאוד לראות משחקי רשת שהשתמשו בהצפנה, משחק זה הוא בין המשחקים הראשונים שהשתמשו בהצפנה. משחק זה נבחר כי הוא מציג בצורה מאוד טובה את הדרכים שיש לעשות על מנת להבין פרוטוקול תקשורת העושה שימוש בהצפנה, את הפעולות לאיך ניתן לנתח את ההצפנה של הפרוטוקול ולקרוא את כל הפרוטוקול כאילו היה לא מוצפן על ידי ביצוע Reverse Engineering למשחק וכו'.

הייתה סיבה, אותה גילו חברת Team17 מהר מאוד, למה משחק מחשב צריך הצפנה משאר החברות אותה נפרט בחלק הבא.

היסטוריה

חברת Team17 הוציאה בסוף שנת 1998 את המשחק השלישי בסדרה - Worm Armageddon, משחק זה הצליח מאוד. במסגרת השקת המשחק, חברת Team17 פרסמו את סביבת WormNET, שרתי משחק שפתחו את האפשרות לשחק באינטרנט עם משחקים מבוססי ליגות, מערכת דרוג וניקוד לכל שחקן. ביום בהיר אחד (בסביבות יולי-אוגוסט בשנת 1999) חברת Team17 הודיעה על ביטול הדרגות והניקוד שהיו ב-WormNET.

מסתבר שבאותה תקופה, היו מספר פרצות אבטחה חמורות שנתנו לקבוצות האקרים לשנות את הנקודות בטבלאות. המשחק WA עבד ללא פרוטוקול הצפנה וכל המידע עבר באופן גלוי. נקודה זו הייתה אחד הדברים שהקל על התוקפים לשנות את המידע בפרוטוקול ובסכימת הנקודות והדירוג.

חברת Team17 ספגה התקפה שגרמה לה לשנות, במשחק החדש (שיצא מאוחר יותר בתחילת 2001) את ההגנה על הפרוטוקול, במטרה להקשות על אותן קבוצות האקרים לבצע התקפות בסגנון זה על ידי הוספת הצפנה לפרוטוקול.

ב-WWP אף פעם לא היו דרגות, אך לאחר הסתכלות בקוד (על ידי ביצוע Reverse Engineering) התגלה כי יש קטעי קוד שלמים מ-WA המופיעים במשחק, כך שעל ידי ניתוח מלא של הפרוטוקול והקמת שרת פרטי בעקבות הניתוח יהיה ניתן להכניס דרגות ל-WWP.

הקמת סביבת עבודה לניתוח

כלים שצריך להתקין לסביבת עבודה

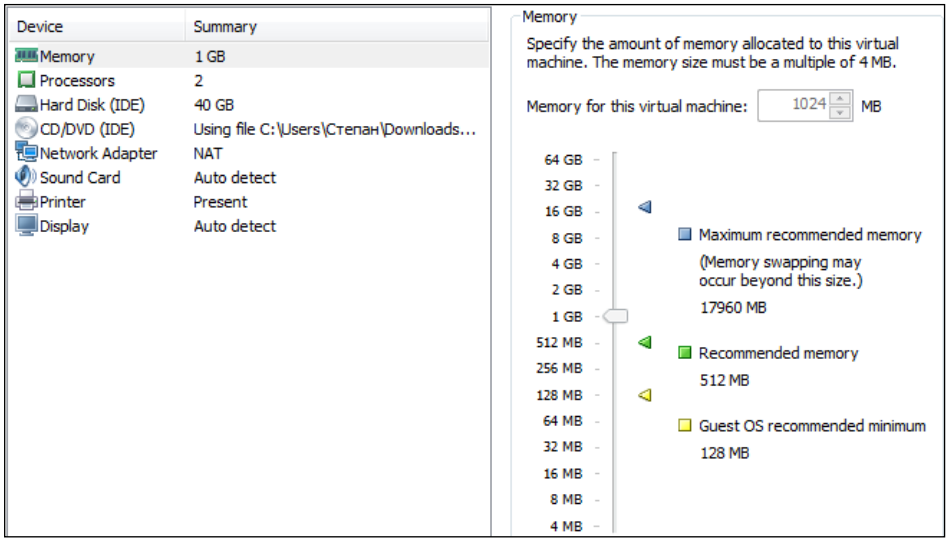
עד כאן היסטוריה, בואו נראה כיצד ניתן להקים את המעבדה הביתית שלנו לטובת מחקר הפרוטוקול של המשחק. על מנת לבצע ניתוח של פרוטוקול המשחק בפרט וניתוח תקשורת בכלל, אנו נדרשים להשיג את הכלים הבאים:

- **סביבה וירטואלית** - VMWare או VirtualBox או כל תוכנה אחרת שיכולה לדמות סביבה וירטואלית למערכות הפעלה שעליהן ניתן להריץ את המשחק (עדיפות ל-VMWare).
- **דיבאגר דינאמי** - Ollydbg או כל דיבאגר דינאמי שמאפשר בעזרתו לנתח **בצורה דינמית** את אופן פעולתו של המשחק (ובפרט - את פעולתו של אלגוריתם ההצפנה).
- **IDA PRO** - כלי סטטי שמאפשר לתעד את הפונקציה ולהמיר אותה לשפה גבוהה יותר, לטובת ביצוע סימולציה לקוד של השרת.
- **Packet Sniffer** - WireShark או כל כלי אחר לביצוע sniffing לתקשורת של הנתונים באינטרנט. למרות שהפרוטוקול מוצפן כלי זה יכול להראות לנו את המידע שעובר ומי מדבר עם מי (השרת אל הקליינט ולהיפך).
- **Visual Studio** - לטובת פיתוח הכלים שיעזרו לניתוח המשחק / ההצפנה.
- **WWP** – כמובן, יש צורך להשיג עותק של המשחק ב-CD / ISO ולהתקין את ה-Patch מהאתר של WWP על מנת שתהיה אפשרות להתחבר לאינטרנט.

התקנת ה-VM

יש עדיפות למערכת הפעלה של XP PRO בתור VM מכיוון שהמשחק די ישן, אפשר לגרום לו לרוץ גם על מערכות הפעלה אחרות אך צריך להוסיף מספר קבצי DLL וכו' על מנת שהמשחק ירוץ.

אלו הגדרות המכונה שאני אשתמש בהן בעת כתיבת המאמר:



[כרטיס הרשת מוגדר על NAT כי כך נוח יותר לסדר את האינטרנט במכונה מבלי התעסקות בהגדרות נוספות]

כלים נוספים לצורך הניתוח

ישנם כלים נוספים אותם צריכים לכתוב תוך כדי ביצוע ניתוח (ולאו דווקא בעת ניתוח משחקי מחשב) על מנת להאיץ את העבודה. משחק המחשב בדרך כלל רץ ב-Full Screen Mode. על מנת לנתח אותו בצורה דינמית בצורה נוחה, יש צורך להעביר אותו ל-Window Mode. לצורך זה נשתמש בכלי שנכתב לטובת ביצוע מספר הוקים (Hooks) לפונקציות של ה-DirectX.

> Hooking - זו דרך בה ניתן לשנות / להחליף פונקציות בכדי לגרום לה לבצע דברים אחרים ממה שהן היו צריכים לעשות. למידע נוסף על נושא זה ניתן לקרוא את המאמר User-Land Hooking שנכתב על ידי Zerith ופורסם בגיליון העשירי של Digital Whisper.

לגרום ל-WWP לרוץ כ-Window Mode זו לא פעולה פשוטה אך יש DLL בשם WndMode (המבוסס על d3dWindower) שנועד למטרה זו (באופן גנרי) ובו בוצעו שינויים (על ידי Kawoosh ו-StepS) בכדי שיהיה אפשר לגרום ל-WWP לעבוד איתו כמו שצריך. (הקוד של d3dWindower סגור ואף יש דיון על כך בעמוד הפרויקט של dxWnd) Kawoosh כתב DLL Proxy ל-dsound.dll שדרכו ניתן להזריק כל קובץ DLL למשחק WWP, עלינו פשוט ליצור DLL, ולקבוע כי שמו יתחיל ב-wk ולמקם אותו בתיקיה של המשחק.

> DLL Proxy - הינו DLL המבצע את אותן הפעולות כמו ה-DLL המקורי אך עם דברים נוספים המתבצעים ברקע בדומה ל-hooks.

על מנת להאיץ את העבודה בניתוח ההצפנה נכתב קובץ DLL נוסף שאליו ניתן לשלוח את הערכים אותם אנו נרצה לפענח (או להצפין!) והקובץ DLL יחזיר את הנתונים לאחר ההצפנה / פיענוח. התצלום הבא מתוך IDA PRO נותן היבט קצר על הפונקציה שבעזרתה ניתן לקבל את התשובה שצריך להחזיר על מנת להתחבר למשחק:

```
text:0043D55F
text:0043D55F                         loc_43D55F:
text:0043D55F 8B 4D 08                    mov     ecx, [ebp+arg_0]
text:0043D562 89 4D F0                    mov     [ebp+var_10], ecx
text:0043D565 83 7D F0 00                 cmp     [ebp+var_10], 0
text:0043D569 74 57                       jz      short loc_43D5C2
text:0043D56B 8D 4D EC                    lea     ecx, [ebp+var_14]
text:0043D56E E8 41 7F 16 00              call    CString::CString(void)
text:0043D573 C7 45 FC 00 00 00+          mov     [ebp+var_4], 0
text:0043D57A
text:0043D57A
text:0043D57A 8B 55 0C                    mov     edx, [ebp+buffer]
text:0043D57D 52                          push    edx
text:0043D57E B9 EC 83 79 00              mov     ecx, offset keysOffsets
text:0043D583 E8 84 4F 06 00              call    getHashForPong
text:0043D588
text:0043D588
```

buffer הינו הפרמטר שבו מוגדר המידע שצריך להצפין / לפענח. לכל חלק בקוד יש מפתח אחר על מנת לפענח את המידע. המשתנה keysOffset מכיל את המידע של איזה מפתח יש לקחת לטובת אימות המשתמש עם השרת בדוגמא זו. הפונקציה מחזירה את התשובה שצריך להחזיר כדי לאמת את הקליינט עם השרת.

הרצה ראשונית של הסביבה במעבדה

לאחר שהגדרנו את המכונה הווירטואלית והמשחק הותקן בהצלחה, נסתכל עם הסניפר שלנו על החבילות שנשלחות מהשרת למחשב שלנו ומהמחשב שלנו בחזרה לשרת, נבצע זאת על מנת לאמת את ההתחברות לשרת של WWP. הסברים מפורטים על מה עושה כל חלק בפרוטוקול הם מעבר לחומר במאמר של חלק זה, אך יוסברו בחלק הבא באופן מלא, בחלק זה נראה סקירה כללית על הפרוטוקול של WWP.

ראשית, נשלח פינג לכתובת שרת המשחק, לטובת זאת נכתוב ב-Cmd את הפקודה הבאה:

```
Ping wormnet2.team17.com
```

הכתובת IP שקיבלנו היא: 212.110.191.17.

כעת, נשים פילטר על ה-IP שאליו ניגש WWP על מנת להציג רק את המידע הרלוונטי שאנחנו צריכים.

הפילטר שלנו בסניפר יהיה החוק הבא:

```
ip.addr == 212.110.191.17
```

לאחר מכן נריץ את המשחק ב-VM בצורה רגילה (ללא Debugger וללא שום כלי מיוחד):

כעת, נעבור לסניפר שלנו, על מנת שנראה את המידע שעבר מהמכונה הוירטואלית שלנו לשרתים של Team17.

בתמונה הבאה ניתן לראות כי ראשית נשלחת בקשת GET (בצבע אדום) מהמכונה הוירטואלית לשרת המשחק, ולאחריה (בצבע סגול) ניתן לראות את המידע שחזר מהשרת (ה-Response) - הקוד שמתבצע על מנת להציג את הדף במשחק כפי שמוצג בתמונה הבאה:

```
GET http://wormnet2.team17.com:80/wwpweb/welcomeloginform.php HTTP/1.0
User-Agent: T17Client/2.0
Pragma: No-Cache
FileResult: 1
UserServerIdent: 1
Counter: 1398193358

HTTP/1.1 200 OK
Date: Tue, 22 Apr 2014 19:03:21 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
Set-Cookie: PHPSESSID=0nr5c6bcjnn3qpbl4jknlonmh3; 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 5B6vNhd9g0um9oMF/yqimhvI5jeTXRPBgyACxsaVbaqR/9NhyekDoL+cGFUJBiyOiv6MPRQazXIx1DU=>
<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>
<FONT Size=3 Colour=3>                                    Your IP address is detected as:      <BR></FONT>
<br>
<a href="/wwpweb/SelectServer.php"><FONT Size=0>                                                Login<BR></FONT>
<br></a>
```

מהסתכלות ראשונית מאוד, ניתן לראות כי מדובר בעצם בקוד HTML המכיל את מסך ההתחברות המוצג למשתמש בעת הכניסה למשחק. מסך ה-login לא מעניין אותנו בשביל לבצע את האימות אך בכל זאת - אנו חייבים לעבור דרכו:

לאחר מכן, נגיע למסך ההתחברות לשרתי המשחק, בו נבחר את שם המשתמש שלנו ואת הסיסמא לחשבון:

מהתבוננות בסניפר, ניתן לראות כי לאחר שמבצעים login, נשלחו הנתונים הבאים לשרת (בצבע אדום), ולאחר מכן התבצע חיבור לשרת IRC בפורט 6677:

```
GET http://wormnet2.team17.com:80/wwpweb/Login.php?
UserName=test&Password=test&IPAddress=192.168.197.131 HTTP/1.0
User-Agent: T17Client/2.0
Pragma: No-Cache
Cookie: PHPSESSID=0nr5c6bcjnn3qpbl4jknlonmh3
FileResult: 77
FileData:
151F1F1F16133434344D4D4D4D4D4D4D4D4D4D4D4D4D4D4D0000000000000000000000000000000000000000
0000000000000000000000000000000000003942424242424242424242424242424242424242424242424242
4242424242424242424242
UserServerIdent: 1
Counter: 1398193507

HTTP/1.1 200 OK
Date: Tue, 22 Apr 2014 19:05:51 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
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
SetUserId: 1
SetPlayerType: 0
PlayerLevel: (0
SetPlayerScore: (0
Vary: Accept-Encoding
Content-Length: 64
Connection: close
Content-Type: text/html

<CONNECT 212.110.191.17 IRCPORT=6677 IRCUSER=Username IRCPASS=>
```

החיבור ל-IRC מתבצע בצורה הבאה:

```
NICK test
USER username hostname servername: 0 18
AUTHPING wormnet.team17.com&1 93BDA0375C0798135837F326B296B34A30221E14
```

לאחר מכן מתבצעת בקשת GET עם הנתונים הבאים (צבועים באדום), בתמונה הבאה ניתן לראות את המחרוזת המוצפנת שנשלחת לשרת, השרת יקבל אותה, יבצע חישוב מסויים ויחזיר למשחק את התשובה, (בסגול) ניתן לראות את התשובה שקיבלנו:

```
GET http://wormnet2.team17.com:80/wwpweb/RequestAuth.php?Secret=wormnet.team17.com
$1&Challenge=93BDA0375C0798135837F326B296B34A30221E14 HTTP/1.0
User-Agent: T17Client/2.0
Pragma: No-Cache
Cookie: PHPSESSID=0nr5c6bcjnn3qpbl4jknlonmh3
FileResult: 77
FileData:
151F1F1F1613343434344D4D4D4D4D4D4D4D4D4D4D4D4D4D4D000000000000000000000000000000000000000000000
0000000000000000000000000000000000000394242424242424242424242424242424242424242424242424242424242
42424242424242424242
UserLevel: (0
UserServerIdent: 1
Counter: 1398193510

HTTP/1.1 200 OK
Date: Tue, 22 Apr 2014 19:05:54 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
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: 58
Connection: close
Content-Type: text/html

<ANSWER WZX9LqwLb8KBUvx2nFV5XChT2+4Jocchv4Rd3YBE6CizXMy+>
```

ניתן לראות את כי בתשובה שהשרת החזיר, קיימת מחרוזת תגובה למחרוזת שהתקבלה מהקליינט:

```
<ANSWER WZX9LqwLb8KBUvx2nFV5XChT2+4Jocchv4Rd3YBE6CizXMy+>
```

התשובה הנ"ל תגיע לקליינט המשחק, והוא בתורו יבצע חישוב נוסף, ואת ה-answer יחזיר כ-AUTHPONG לשרת ה-IRC באופן הבא:

```
AUTHPONG wormnet.team17.com$1 492A15320FE9DEA7C2D1CC3558C56349FCCC8AAF
```

שימו לב שאת כל המידע עד כה הצלחנו להשיג ע"י הסתכלות בתווך התקשורת בין קליינט המשחק לבין השרת, מדובר במידע שעובר בין המחשב שלנו (בתור שחקנים) לבין השרת, ל-Team17 אין הרבה מה לעשות - זה מידע שיוצא מהמחשב שלנו, ולכן בכמעט רוב המקרים, נוכל להגיע אליו. Team17 הכניסו את מנגנוני ההזדהות (עם המחרוזות המורכבות) על מנת למנוע מאיתנו להצליח לזייף התחברות לשרת גם במידה ואנו מסוגלים לצפות בתקשורת (כמו שראינו עכשיו).

על מנת להבין את השלבים לעומק - ועל מנת להצליח להתחקות אחריהם (ואף לזייפם), עלינו לבצעReverse Engineering למשחק, על כך נרחיב בחלק הבא של המאמר.

סיכום חלק א'

בחלק זה ראינו מה מיוחד ב-WWP מכל שאר משחקי המחשב האחרים שהיו באותה תקופה, הצגנו את הבעיה שאנו רוצים לפתור על ידי ביצוע Reverse Engineering לפרוטוקול על מנת לכתוב שרת משחק פרטי שיהיו בו דרגות.

בחלק הבא אציג פירוט על איך בוצע תהליך ה-Reverse Engineering המעמיק על פרוטוקול המשחק, שיאפשר להתחבר לשרת של המשחק בלי המשחק עצמו.

מי שמעניין אותו הנושא, ורוצה להרחיב את הידע לקראת החלק הבא במאמר, אני ממליץ לו מאוד לעיין בנושאים הבאים:

- HTTP overview
- IRC overview
- Introduction to Server Side Emulation
- Proxy DLL
- Proxy-dll for start DirectDraw of games in a Window Mode

על מחבר המאמר (d4d)

מחבר המאמר עוסק בתחום ה-Reverse Engineering ואוהב לחקור משחקי מחשב והגנות, לכל שאלה שיש או ייעוץ ניתן לפנות אלי בשרת ה-IRC של Nix, בערוץ:

#reversing

או בכתובת האימייל: llcashall@gmail.com.