Technologic Papers

על קופסאות נופלות ושטחים-לא-נדל"ניים אחרים

נכתב ע"י יובל נתיב (tisf) ואפיק קסטיאל (cp77fk4r)

הקדמה

במאמר הבא נדבר על נושא שהפך להיות נחלת הכלל ועל רמות האבטחה שלו. כיום, כמעט כולנו משתמשים בשטחי אחסון חיצוניים כגון Dropbox, Google Drive, Amazon Storage, Ubuntu One ודומיהם. השירותים הללו מציעים לנו שטח דיסק מרוחק ואמצעי סנכרון לכמעט כל מכשיר קיים, החל ממכונות Windows, Linux, Mac, Android, iOS וכן הלאה.

הרעיון מאחורי שירותים אלו הוא שאנחנו, בתור לקוח פרטי, יכולים לאחסן את המידע שלנו בשרת באינטרנט, ואז כאשר נהיה מעוניינים לגשת אליו מהמשרד בעבודה, מביתו של הדוד, או סתם כאשר אנו בחופשה, נוכל לגשת, מבלי הצורך להיסחב עם כונן נייד, או Disk On Key. דבר נוסף ששירותים אלו מציעים הוא בקרה וגיבוי למידע. אם נשמור מידע על Disk On Key והוא ייגנב או ישכח באיזה מקום - המידע אבד, אך אם נשמור אותו בענן של אמאזון, ספק שהם יאבדו לנו אותו...

בתור לקוח עסקי או חברה, יש כאן פוטנציאל גדול יותר, נוכל לשתף מידע או להעביר מידע כבד לספקים, סניפים, לקוחות ועוד, מבלי הצורך בהתעסקות של הלוגיסטיקה, שלא נדבר על כך שאין צורך בבקרה על השרתים או הקמת התשתיות לכך. מספיק שיש לנו חיבור לאינטרנט - ונוכל לסנכרן את הקבצים שם.

הרעיון נשמע טוב מצד אחד, אך מצד שני, כל המידע האישי, הפרטי או העסקי שלנו - נמצאים להם אי שם על שרת באינטרנט, לא ב-Disk On Key בתיק שלי, מחובר לצרור המפתחות. זה קצת מלחיץ, לא?

במהלך מאמר זה נבצע סקירה של הסכנות הקיימות בעת השימוש בשירותים אלו, וכיצד אנו, המשתמשים הביתיים יכולים להתמודד מפניהם. ננסה לסקור את האיומים הקיימים בכל אחד משלבי השימוש / חלקי המערכת ונראה כיצד נוכל למנע מהם. חשוב לזכור כי המידע שיופיע במאמר תקף גם למערכות אחרות שמרכיביהן מקבילים למערכות שיוצגו במאמר זה.

מבנה המערכת

על מנת לעשות סדר במאמר, נתייחס לארכיטקטורת המערכת מבחינת הסיכונים והאיומים הקיימים עליה, ועל מנת לעשות זאת, נחלק את המאמר לשלושה חלקים עיקריים:

1. **מנגנון הזיהוי ואימות הזהות למערכת** (כניסה למערכת / זיהוי חוזר)
2. **פרוטוקול התקשורת של המערכת (**הדרך בה אנו מדברים עם המערכת)
3. **ממשק/לוגיקת המערכת** (פונקציונאליות המערכת)

במהלך המאמר נעבור על כל חלק וחלק במערכת, נציג אותו, את תפקידו במערכת ואת הסיכונים הקיימים המיוחסים אליו.

מנגנון הזיהוי ואימות הזהות למערכת

שלב האימות הוא השלב בו אנו "מדברים" בפעם הראשונה עם השרת ומודיעים לו מי אנחנו (בעזרת שם משתמש לדוגמא, או מזהה אחר במערכת). בתמורה, הוא מבקש מאתנו דבר נוסף על מנת לוודא שמי שאנחנו טוענים שאנחנו - זה באמת אנחנו (כמו סיסמה, Token או OTP למשל). את החלק הנ"ל, ניתן לחלק בדרך כלל לשני חלקים עיקריים: "שלב האימות הראשוני" ו"שלב האימות החוזר".

ב**שלב האימות הראשוני**, במצב ברירת מחדל, השרתים מבקשים מאתנו שני דברים: שם משתמש וסיסמה. זה ידוע כאימות חד שלבי. אימות חד שלבי הוא אימות המבקש לדעת מי אנחנו (שם משתמש) ומשהוא שאנו יודעים (במקרה הזה, סיסמה).

שלב האימות החוזר נגזר משלב האימות הראשוני, שלב האימות החוזר מתבצע במקביל לכל פעולה ופעולה שלנו בשרת. לאחר שביצענו בהצלחה את שלב האימות הראשוני, נשמר מזהה השיחה / חיבור שלנו עם המערכת (על מנת שלא נצטרך לבצע אימות בכל פעולה שנבצע במערכת). מזהה זה נקרא עוגייה (Cookie) או Session ID. מדובר בדרך כלל בערך ארוך וייחודי בעל תאריך תפוגה. תפקידו הוא לאמת כל פעולה שלנו (שאכן התבצע על ידינו) מבלי הצורך שנשלח את סיסמתנו בכל פעם. בפעם הבאה שאנו ניגש לשרת לביצוע פעולה מסוימת, השרת יבקש לראות מהו תוכן העוגייה שלנו ולאחר שנשלח אותה אליו, הוא יזהה לאיזה חשבון מקושרת העוגייה, ייבצע בדיקות כגון תאריך תפוגה וכו', ולאחר מכן, יספק לנו את השירות המבוקש. במידה ותוקף השיג גישה לתווך התקשורת דרכו אנו מתקשרים עם השרת ועל ידי כך השיג את העוגייה הנ"ל, בדרך כלל היא תספיק לו על מנת לבצע פעולות בשמנו - אך חשיפת תוכן העוגייה שלנו עדיף מחשיפת סיסמתנו.

בנוסף, ניתן לחולל את הערך הקיים בעוגייה בעזרת מספר משתנים, כאשר אחד מהם הוא כתובת ה-IP של המשתמש. במידה וערך זה ייפול, לתוקף לא תהיה אפשרות לבצע בו שימוש מפני שכתובת ה-IP שלנו אינה זהה לכתובת ה-IP של המשתמש. אך למרות כי דרך מימוש זה נחשבת כמאובטחת יותר, רב השירותים לא מבצעים בה שימוש מפני אי-הנוחות הקיים למשתמש (בכל פעם שמשתנה כתובת ה-IP של מחשב המשתמש, הוא יאלץ להקליד את סיסמתו שוב על מנת לחולל מזהה חדש).

פרוטוקול התקשורת של המערכת

פרוטוקול התקשורת של המערכת מגדיר את מכלול החוקים שבעזרתם הדפדפן, או כל תוכנת סנכרון קבצים ייעודית שבעזרתה אנו עובדים מול ספקי שירות הענן תתקשר עם השרת. המטרה מאחורי קביעת פרוטוקול אחיד לתקשורת היא לקבוע סדר בחבילות המידע ובתוכן שאותו הן מעבירות. ללא סדר קבוע, צד הלקוח וצד השרת לא יוכלו להבין זה את זה. בדיוק כמו שללא **שום** שפה משותפת בין שני אנשים, הם לא יוכלו לתקשר אחד עם השני.

כאמור, פרוטוקול התקשורת אחראי על אחידות שפת התקשורת, וכמובן - העברת המידע עצמו, אך חוץ מתפקיד זה, הפרוטוקול בדרך כלל מעביר מידע נוסף ("Metadata"), מדובר מידע שהוא לא התוכן המבוקש עצמו, אלא מידע נלווה, שיכול לעזור בתפעול התקשורת. דוגמא קלאסית הינה פרוטוקול ה-HTTP, כאשר אנו גולשים לאתר אינטרנט ומבקשים תוכן של עמוד מסוים, צד השרת שולח לנו את תוכן העמוד, מלבד תוכן העמוד מתווסף מידע נוסף כגון סוג התשובה ("200" - הדף נמצא, "404" - הדף אינו קיים, "403" - הדף קיים, אך אין לנו הרשאות מתאימות לצפות בו, ועוד), כמה זמן לשמור את העמוד במנגנון ב-Cache, ועוד.

כאשר אנו מתקשרים אם שטח האחסון המרוחק שלנו, הנמצא על שרתי ספקית אותו השטח, ואנו מעבירים קבצים / או מעוניינים לבדוק כמה מקום פנוי נשאר לנו, לחיצות העכבר והקשות המקלדת שלנו, מתורגמות למידע המועבר דרך פרוטוקול התקשורת לצד השרת, צד השרת מסוגל להבין מה ביקשנו, לבצע את המשימה ולהחזיר את המידע הרלוונטי שביקשנו.

בעזרת פרוטוקול התקשורת, המידע לא רק עובר מהצד השולח אל הצד המקבל, אלא גם מקבל משמעות, מפני שבפרוטוקול התקשורת, למידע הממוקם באזור X יש משמעות אחת, אך מידע הממוקם באזור Y יכול להיות משמעות שונה לחלוטין.

ממשק/לוגיקת המערכת

ממשק המערכת הינו הרכיב המרכזי במערכת, כאשר אנחנו כותבים "ממשק" אנחנו לא מתכוונים לתפריטים, אלא לכלל לוגיקת המערכת ולכלל הפונקציות המרכיבות אותה. יכול להיות כי תהליך ההזדהות למערכת, מנגנון האימות החוזר, ופרוטוקול המערכת מאובטחים למדי, אך חשוב כי אלו הם רק המעטפת, אם ממשק המערכת עצמו נכתב בצורה בעייתית, יכול להיות שלמרות שכל המידע שלנו מוצפן, תוקף יוכל לנצל כשלים לוגיים / תכנותיים במערכת ולהשיג מידע מחשבוננו. זה שעמוד מילוי טופס הכנסת הסיסמה משתמש ב-SSL על מנת לאבטח את המידע מפני מתקפות MITM, אך המידע מאוחסן בצורה גלויה בטבלאות שניתן לשלוף מהן את המידע בעזרת מתקפות כגון SQL Injection, או שניתן לגשת לממשק הניהול עצמו ע"י שינוי ה-ID של המשתמש, זה נחמד, אבל לא ממש עוזר מעיד את רמת אבטחה גבוהה.

ממשק המערכת מורכב ממספר רכיבים, הממשק הקיים בצד הלקוח, הממשק המיוצא לצד הלקוח בצד השרת (כדוגמת ה-Web Services, או ה-API), לוגית צד הלקוח, לוגיקת צד השרת, וכמובן - מסד הנתונים שתפקידו לאחסן את כלל המידע בצורה שיהיה קל לשלוף ולבצע עליו חיתוכים שונים. בכל רכיב ורכיב הממשק בכל צורה שהיא, מנגנון אבטחה - יכול להיות כשל.

כשלים כלליים יכולים להיות מקרים בהם כלל לוגית האבטחה או זיהוי המשתמש מתבצעת בצד הלקוח, מה שמאפשר לתוקף לבצע מניפולציות על המידע הנשלח לשרת לאחר שזה יצא מתוכנת הלקוח (על ידי Data Tampering או על ידי Revers Engineering) וכך להשיג גישה לאזורים/מידע/פונקציות שלא היו נגישים אליו בצורה "טבעית".

כשלים מקומיים יכולים להיות מקרים בהם Web Service מסוים, אשר אחראי על פעולות ניהול קריטיות במערכת נגיש לכלל המשתמשים ללא הצורך בביצוע הזדהות. במקרה כזה, תוקף יוכל לגשת לאותו Web Service ולהפעילו "ידנית" וכך לחבל במערכת, או לגנוב ממנה מידע רגיש.

סוגי האיומים

לאחר שראינו והבנו מה הם מרכיבי המערכת, נוכל לעבור ולסקור את האיומים והסכנות הקיימות לנו בתור משתמשי קצה בעת השימוש במערכת. האיומים שנגע בהם הם:

1. **מנגנון הזיהוי ואימות הזהות למערכת** (כניסה למערכת / זיהוי חוזר)
   - מתקפות Key Logging / Key Sniffing.
   - מתקפות Man In The Browser.
   - ניחוש סיסמאות שיטתי.
   - מתקפות פישינג והונאות.
   - שליפת מידע רגיש השמור באופן לא מאובטח.
2. **פרוטוקול התקשורת של המערכת** (הדרך בה אנו מדברים עם המערכת)
   - מתקפות Man In The Middle ומתקפות Rogue Routing נוספות.
   - מתקפות Data Tampering וחולשות לוגיות בפרוטוקול המערכת.
3. **ממשק/לוגיקת המערכת** (פונקציונאליות המערכת)
   - מתקפות Server Side.

מנגנון הזיהוי ואימות והזדהות למערכת

כמו שראינו, הליך הזיהוי מתבצע בשני אופנים, בעת השלב הראשוני - שלב הכנסת פרטי ההזדהות למערכת, ושלב האימות החוזר - השלב בו אנו מזדהים לשרת עם כל פעולה שלנו בעזרת העוגייה וה-Session ID.

מתקפות Key Logging

שלב הכנסת פרטי ההזדהות למערכת הינו שלב מהיר, מדובר במספר שניות שבהן אנו מקלידים מחרוזת השמורה בראש שלנו לממשק ההזדהות של המערכת, אחד הסיכונים העיקריים שלנו כאן הוא מתקפות Key Sniffing / KeyLogging. מי מבצע אותן? תוכנות / חומרות קטנות בשם "Key Loggers", מדובר בתוכנות שמבצעים מספר שינויים בתהליכים הנמצאים בין המקלדת שלנו לבין עמוד האינטרנט, הן יכולות להיות בתצורה חומרית, ולהתלבש כמתאם קטן בין המקלדת למחבר ה-PS2 / USB מאחורי המחשב, הן יכולות להיות בתצורה וירטואלית כ-Driver, ולבצע מספר Hook-ים ברמת ה-Kernel על מנת להאזין

לתעבורת המידע המגיעה מהמקלדת. והן להיות בתצורה וירטואלית כתוכנת User-Land (מה שנפוץ בעיקר, בעיקר בגלל הפשטות) ולבצע Hook-ים ומניפולציות על המידע ברמת ה-User Mode.

כיום, נראה שכמעט כל וירוס, תולעת, בוט-נט וכל שאר המזיקים ניחנים ביכולות Key Logging כאלה ואחרות על מנת לגלות את סיסמאות המשתמשים במחשב אליהם הן הצליחו לחדור.

כיצד ניתן להתגונן?

בדרך כלל, יחסית קל להישמר מפני מתקפות אלו, למרות כל הקלות הבלתי נסבלת היום של הפריצה למחשב אישי. עם כמה שזה נשמע מצחיק - נוכל להימנע ממתקפות אלו על ידי מספר עקרונות פשוטים שיעזרו לנו להימנע מלהידבק או להשתמש במחשבים עם פוטנציאל גבוהה להדבקות בכל מני וירוסים או מזיקים שיכולים לגנוב לנו את הסיסמאות:

- דואגים להתקין תוכנת Anti-Virus רצינית על המחשב (אנחנו לא מדברים על כל התוכנות ה-Cleaners המצחיקות האלה, שלא באמת עושות משהו רציני ומשום מה משתמשים שונים מחשיבים אותן, אלא אחת מארבעת-חמשת תוכנות אנטי וירוס הרציניות היום בשוק). ודואגים לעדכן אותה!

- כנ"ל על תוכנות Firewall.

- לא מתחברים לחשבונות שלנו באינטרנט ממחשבים לא מוכרים, כי, לכו תדעו מה לעזאזל יש על המחשב הזה. (לדוגמא, מחשבי אינטרנט-קפה, אינטרנט בספריה באוניברסיטה, לכו תדעו לאיפה גלשו מהם ומה עשות איתם - מחשבים כאלה, הם מדגרות קלאסיות לווירוסים).

- לא גולשים לכל מני אתרים מפוקפקים באינטרנט, נזהרים לא ללחוץ על כל מני קישורים הנראים חשודים (לאט לאט, צוברים את הניסיון לדעת מה נראה חשוד ומה לא).

- לא משתמשים בכונן USB לפני שסורקים אותו מווירוסים, עם עדיפות לשימוש בכונן USB אישי שאנחנו יודעים בדיוק מה יש עליו - ושנבדק ונסרק מדי פרק זמן קצר. וכמובן, לא מכניסים את הכונן הנ"ל למחשב נוסף בלי תוכנת אנטי-וירוס מעודכנת. תודות למיקרוסופט, ולפיצ'ר ה-AutoRun שלה, תולעים אוהבות מאוד להדביק התקני USB ניידים, מדובר בטרמפ חינם למחשב האישי של כל אחד ואחד מאתנו.

- לא מתקינים או מריצים כל מני תוכנות שאנו לא יודעים מה טבען או שמקורן לא אמין.

- ופשוט מאוד - משתמשים במחשב בצורה מודעת לאבטחת מידע.

מתקפות Man In The Browser

מתקפות MiTB הן מתקפות המזכירות את מתקפות ה-Key Sniffing מבחינת המקור מהן הן מגיעות (ולכן דרך ההימנעות ממתקפה זו דומה מאוד לדרך ההימנעות מהמתקפה הקודמת שהצגנו), השוני בין המתקפה הקודמת למתקפות MiTB הוא בדרך מימושן ובזה שמתקפות אלו הן ספציפיות מאוד.

מבחינת דרך המימוש - מדובר בתוספים זדוניים לדפדפן (או בצורה קלאסית על ידי התקנת הרחבה לדפדפן, או על ידי שינוי בקוד שלו וביצוע Hooking על התהליך עצמו בעת השימוש בו) הנמצאים בין המשתמש לבין התוכנה, מסוגלים לעשות כל העולה על רוחם - מפעולות באתר בשם המשתמש, ועד הצגת תוכן מסולף ושקרי.

מבחינת המיקוד - מדובר במתקפות שבדרך כלל ממוקדות למספר אתרים קטן, והן ממוקדות לפעולות ספציפיות. הספציפיות הנ"ל מצד אחד מקטינה את הסיכון מהן (מפני שלא לכולם יש חשבון PayPal - ואם נדבקנו בווירוס שמבצע MiTB על PayPal ואין לנו חשבון באתר זה, לא נרגיש בו כל כך), אך מצד שני היא מגבירה את יכולת הביצוע (ולכן את הנזק), מדובר במתקפות שמסוגלות לבצע פעולה ספציפית ולכסות אותה (לדוגמא, התוסף הזדוני מחכה שנתחבר לאתר של PayPal, מחכה שנבצע העברת כספים מסוימת, ובזמן אישור הפעולה - הוא משנה את יעד הפעולה כך שהכסף יגיע ליוצר התוסף. וכמובן - שינוי המידע המוצג למשתמש בסופו של דבר, המשתמש יראה כי הכסף אכן עבר לספק השירות אליו הוא התכוון להעביר את הכסף, אך כמובן שספק זה לא קיבל את הכסף מעולם, הוא הגיע לחשבונו של יוצר הפלאגין). ניתן לקרוא עוד על מתקפות אלו במאמר שנכתב על ידי הרצל לוי ו[פורסם בגליון ה-18 של Digital Whisper](#).

כיצד ניתן להתגונן?

כמו שכבר נכתב, ההתגוננות מפני מתקפות אלו הוא שימוש מודע לאבטחת מידע במחשב, כל מה שנבצע על מנת התגונן מפני מתקפות Key Logging נבצע גם כאן. עם הדגשים הבאים:

- **אין להתקין תוספי דפדפן ממקורות שאינם אמינים**, כאשר מתקינים תוספים לדפדפן, הם מסוגלים לקבל שליטה על המידע המתקבל ועל המידע הנשלח מהדפדפן. לא פעם ראינו מקרים בהם פורסמו תוספים זדוניים שמטרתם הינה לחבל בפעולתו התקינה של הדפדפן על מנת להזיק למשתמש.

ניחוש סיסמאות שיטתי

גם אם המערכת אליה אנו מתחברים היא מבצר, או בונקר, ורמת האבטחה בו גבוהה, ואנחנו משתמשים בהצפנות משוגעות על מנת לאבטח את התקשורת שלנו עם המערכת, כל עוד נשתמש בסיסמאות חלשות - אין לרמת האבטחה שום סיכוי לעצור את התוקפים מלהגיע לחשבון שלנו. אחת המתקפות הוותיקות בעולם ההאקינג היא מתקפת Brute-Force, ובעזרתה, כל סיסמה חלשה תיפול. בעת מתקפות אלו, תוקפים מנסים באופן שיטתי את כלל המחרוזות האפשריות על מנת לנסות ולפגוע בסיסמה הנכונה. מדובר במתקפה הנחשבת "טיפשה" ואיטית, אך למרות טיפשותה ואיטיותה, ניתן לבצעה במספר אופנים שיכולים להקל על התוקף ולקצר את המלאכה.

לדוגמא - ניחוש סיסמאות ממילון ("Dictionary Attack") שהוכן מראש, המכיל סיסמאות הנפוצות בשימוש, וכך לנסות לנחש סיסמאות נפוצות, כמות הפריצות לאתרים שבוצעו לאחרונה, ופרסום סיסמאות המשתמשים, מקנה לתוקפים יכולות הרכבת מילון עם סיסמאות שנעשו בהן שימוש במציאות. או ביצוע ניחוש של סיסמאות נפוצות באופן רוחבי (על כמות גדולה של משתמשי המערכת), שימוש בטבלאות Hash או טבלאות Rainbow (במידה והתוקף השיג את סיסמאות המשתמשים באופן מגובב, ועליו כעת רק לנסות "לשבור" אותן). בנוסף, גורמים עוינים בעלי אמצעים, יכולים לגייס רשתות בוט-נט על מנת לנצל כוח עיבוד גדול יותר, וכך לבצע חישובים באופן מהיר יותר על מנת לקצר את זמן התקיפה הנדרש לכיסוי מירב הסיסמאות. דרך נוספת להגביר את קצב החישוב היא שימוש בחומרה ייעודית (כגון מאיצים גרפיים) וחישובים מבוססי GPU.

בנוסף, מלבד ניחוש במתקפות אלו על מנת לנחש את סיסמא החשבון, תוקפים יכולים לנסות לפרוץ לחשבונות אחרים שלנו על מנת להשיג את סיסמאותינו באופן קל יותר (לדוגמא - ניתן להניח שהרבה יותר קל לפרוץ לחשבון שלנו באתר האוניברסיטה מאשר לחשבון ה-PayPal שלנו), ואז לנסות את הסיסמה שבחרנו בחשבון אחד על מנת להתחבר לחשבון היעד בהנחה שאנו משתמשים באותה הסיסמה למספר חשבונות. ממחקרים שארגוני אבטחה עושים, נראה כי רב האנשים משתמשים במספר קטן מאוד של סיסמאות לרב חשבונותיהם.

כיצד ניתן להתגונן?

- מורכבות הסיסמה (אורך, סוג תווים, שכיחות וכו') הוא המפתח כאן. חשוב מאוד להשתמש בסיסמאות מורכבות - ככל שהסיסמה תהיה מורכבת יותר, כך לתוקפים יידרש זמן רב יותר למצוא את הסיסמה הנכונה. יש עוד מספר אלמנטים (כגון מנגנוני Lock-Out או מנגנוני זיהוי מתקפות מסגנון זה) שניתן לממש בעת יצירת המערכת, אך אנו, בתור משתמשי קצה מחוייבים לעשות את

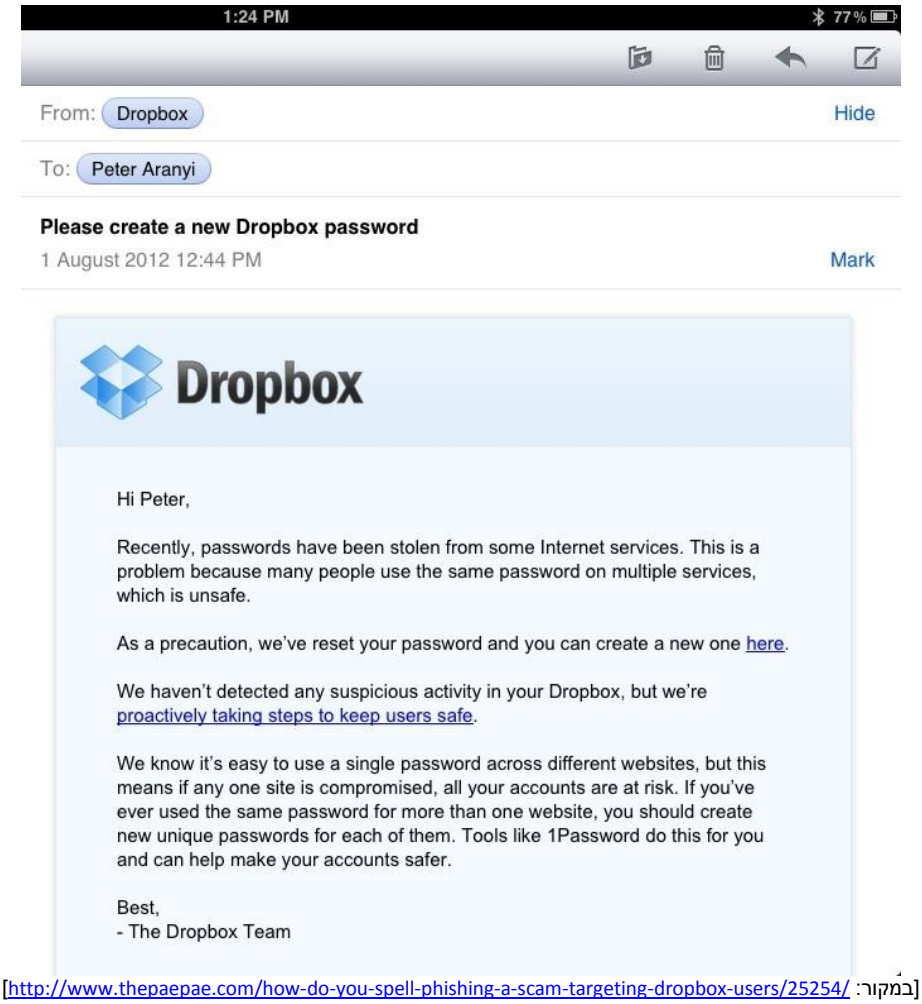
המקסימום הנדרש מאתנו על מנת לשמור על חשבוננו מאובטח. סיסמה חזקה, כיום, נחשבת כסיסמה עם מעל 15 תווים, ולפחות שלושה סוגי תווים (אותיות גדולות, אותיות קטנות ומספרים, או - אותיות קטנות, מספרים וסימנים מיוחדים). עם זאת, עדיף תמיד לבצע שימוש בסיסמאות מורכבות יותר. ניתן להשתמש במשפט קצר בתור סיסמה, בדרך זו, גם נקשה על התוקף לנחש את הסיסמה וגם נקל עלינו בעת זכירתה (הרבה יותר קל לזכור משפט עם היגיון מאשר רצף תווים אקראי), למידע נוסף ניתן להיכנס לקישור הבא:

http://strongpasswordgenerator.com/

- אין להשתמש באותה הסיסמה למספר חשבונות, כבר ראינו מקרים בהם פורסמו מאגרי מידע עצומים על כל פרטיהם, ולאחר מכן - נפרצו חשבונות נוספים באתרים מאובטחים לגמרי על בסיס השימוש באותה הסיסמה שפורסמה לחשבונות שונים של אותו המשתמש.

מתקפות פישינג והונאות

מתקפות פישינג אינן נושא חדש. ועקב הפשטות בביצוען גם לא נראה כאילו הן עומדות להעלם. מדובר בארגונים, או אנשים פרטיים המקימים אתרים הזהים בחיצוניותם לאתרים אשר את משתמשיהם הם מנסים לתקוף, לאחר הקמת אתר כזה, כל שעל יוצר האתר לעשות הוא לגרום לאנשים להגיע אליו, בדרך כלל מדובר בשליחת מייל שנראה אותנטי הדורש מהמשתמשים להיכנס לאתר המתחזה על מנת לבצע פעולה מסוימת (לדוגמא - לעדכן פרטים שנמחקו עקב תקלה במערכת, או לענות להודעה שהתקבלה ממש אחר וכו). מתקפות הפישינג לא דילגו על עולם האחסון בענן. דוגמא קלאסית:



[במקור: http://www.thepaepae.com/how-do-you-spell-phishing-a-scam-targeting-dropbox-users/25254/]

מי שיכנס לקישור ויעבור על הפוסט שפורסם, יגלה כי בעצם, לא מדובר במתקפת פישינג, אלא בבקשה לגיטימית של Dropbox. וזאת דוגמא מצוינת - כמעט ואין סיכוי לזהות הודעות פישינג מבלי לדעת איך הנושא עובד. על פניו, הודעות אלו נראות זהות לחלוטין להודעות הלגיטימיות של ספק השירות.

כיצד ניתן להתגונן?

קל מאוד לפול לפישינג, מפני שאם המשתמש הזדוני מאחורי האתר ביצע עבודה מקצועית - לא נזהה שום דבר חריג. כמעט. יש מספר נקודות קטנות שאם נשים לב אליהן, נוכל להקטין משמעותית את הסיכוי לפול למתקפת פישינג:

- לפני שלוחצים על קישור, גם אם זה בתיבת המייל, וגם אם זה בפורום או באתר חדשות לגיטימי - שווה להעיף מבט לחלק התחתון של הדפדפן, בכל הדפדפנים הסטנדרטיים, ברגע שנעביר את העכבר על הקישור (מבלי ללחוץ!), נוכל לראות להיכן אנו מובלים. עם זאת, קיימות מתקפות ישנות ופשוטות לביצוע על מנת לזייף את מה שמציג מנגנון זה בכמעט כל דפדפן בשוק. לדוגמא.

- לפני שמקלידים את פרטי ההזדהות, או בעצם, בעת הכניסה לכל אתר - שווה להעיף מבט בשורת הכתובת ולבדוק שאנו אכן נמצאים באתר בו אנו אמורים להיות. את רב מתקפות הפישינג יהיה ניתן בקלות לגלות ברגע שנסתכל על שורת הכתובת, אך מתקפות פישינג מתוחכמות בדרך כלל יאוחסנו על אתרים עם כתובות זהות ויזואלית לכתובות האתרים אותם הם מנסים לחכות. לדוגמא, פישינג על הכתובת Gmail.com יהיה ניתן למצוא בכתובת כגון Grnail.com (m מתחלפת עם r ו-n שביחד נראות יחסית זהות).

- בנוסף, חשוב לזכור כי רב האתרים הגדולים כיום לא יבקשו מאתנו לשלוח אליהם את סיסמאותינו, או בקשות דומות אחרות. וגם אם כן - עדיף יהיה להקליד ידנית את כתובת האתר המדובר מאשר ללחוץ על קישור, קיימות מתקפות מסוג "URL Spoofing" שמנצלות חולשות במנגנונים שונים בדפדפן על מנת להציג לנו כתובת שונה מהכתובת בה אנו נמצאים, דוגמאות:
  - http://lcamtuf.blogspot.co.il/2010/06/yeah-about-that-address-bar-thing.html
  - http://lcamtuf.blogspot.co.il/2010/04/address-bar-and-sea-of-darkness.html
  - http://seclists.org/fulldisclosure/2011/Jul/282
  - http://www.idownloadblog.com/2012/03/22/safari-exploit-in-ios-5-1/
  - http://www.yourdailymac.net/2012/03/adressbar-spoofing-vulnerability-found-in-mobile-safari-webkit/
  - https://bugzilla.mozilla.org/show_bug.cgi?id=514232

  אלו דוגמאות, ולכן סביר להניח כי כולן כבר נסגרו, אך פגיעויות כאלה מתגלות כל הזמן.

- להיות ערני, מתקפות מסוג Tab Nabbing הן מתקפות המאפשרות לתוקף להפוך אתר שנראה לתמים לאתר אחר, כאשר אנו לא נמצאים על הטאב הספציפי בפוקוס, בתקווה שנחשוב כי פתחתנו את האתר ההוא ונקליד שם את פרטי ההזדהות לאותו אתר.

לדוגמא: אנו גולשים באתר חדשות ונכנסים לקישור המפנה לאתר (שנראה) תמים, ומשאירים את הטאב פתוח. עובדים לטאב חדש וממשיכים לגלוש, לאחר מספר שניות, אותו הטאב מזהה שאנו לא נמצאים עליו בפוקוס ומחליף את פניו כך שיראה כאילו מדובר בעמוד ההזדהות לחשבון ה- Amazone Storage שלנו (שפתוח בטאב אחר), כאשר נחזור לאותו הטאב, יש סיכוי שנקליד את פרטי ההזדהות ממחשבה כי נותקנו מחשבוננו באמאזון. דוגמא למתקפה זו ניתן לראות בקישור הבא:
http://www.azarask.in/blog/post/a-new-type-of-phishing-attack

- שימוש באימות דו שלבי. עד כה דיברנו על התהליך אימות חד-שלבי שהוא האימות שכולנו מכירים. השלב הבא הוא אימות דו-שלבי. אימות דו-שלבי הוא אימות המכיל שלב נוסף (ואיתו אלמנט זיהוי נוסף). אימות חד שלבי יכלול בדרך כלל "משהוא שאני יודע" (כגון סיסמא), אך אימות דו-שלתי יכלול אלמנט נוסף, כגון: "משהוא שיש לי". את הראשון אנו מכירים. לשלב השני יש הרבה גרסאות שונות. את חלקם אנחנו מכירים בתור אימות ביומטרי - טביעת אצבע, סריקת רשתית, זיהוי פנים ועוד. חלקם מופיעים ברכיב Keyfob המפורסם של חברת RSA שמחולל כל 30 שניות קוד בן 6 ספרות. בעולם האינטרנט שתי השיטות הנפוצות ביותר לאנשים הפרטיים הן אימות בעזרת מכשיר סלולרי או Google Authenticator.

האפשרות הראשונה אומרת שאוודא את מספר הסלולר שלי ואקשר אותו עם החשבון, כך שכאשר אבצע אימות, השירות ישלח אליי הודעה עם קוד חד פעמי אותו אצטרך להזין וכך יוכל לוודא השירות בעוד דרך, שלא רק שאני יודע את הסיסמא לחשבון, אלא אני גם מחזיק את מכשיר הסלולר שמחובר עם אותו החשבון.

האפשרות השנייה מעניינת יותר. Google השיקה מוצר בשם PAM (קיצור של Pluggable Authentication Module) אשר מייצר מנגנון אימות נוסף בעזרת אפליקציה קטנה ונוחה. היתרון המשמעותי של שירות זה, הוא אופן התכנון שלו. שירות ה-PAM הוא פתוח, חינמי ואינו מערב את Google אלא רק פותח על ידה. כיום, ניתן להוריד, להגדיר, לשנות ככל העולה על רוחכם ולהפעיל את הרכיב עם כל שירות שאתם מחזיקים. Dropbox הכניסו את החבילה הזאת לתוכנית שלהם, ועוד רבים אחרים. כיום אתם יכולים להגיד את ה-PAM של גוגל עם שרת ה-SSH שלכם בבית, לדוגמא. בקישור הבא תוכלו לראות כיצד לממש זאת:
https://www.macworld.com/article/1168299/how_to_configure_dropboxs_two_step_authentication.html

שליפת מידע רגיש השמור באופן לא מאובטח

כאשר אנו מזדהים למערכת מסוימת בעזרת קליינט מסוים, אותו הקליינט שומר מידע באופן מקומי על המחשב, אם זה מידע בנוגע להגדרות המועדפות עלינו, אם מדובר בהיסטוריית פעולות, והם מדובר בפרטי הזיהוי שלנו. בייחוד כאשר אנו מסמנים ב-V את האופציה "זכור אותי". במידה ופרטים אלו ישמרו באופן שאינו מאובטח, תוקף בעל גישה מלאה למחשב (פיזית, או מרוחקת בעזרת סוס טרויאני המאפשר לו גישה זו), יוכל לגנוב את המידע. אם בעבר מערכות היו שומרות את פרטי ההזדהות באופן לא מוצפן, אז כיום המודעות לכך עלתה וכבר קשה לאתר מקרים כאלו. עם זאת, מתגלים מקרים אחרים, כדוגמת החולשה שהתגלתה באפריל שנה שעברה ב-Dropbox על ידי חוקר האבטחה Derek Newton.

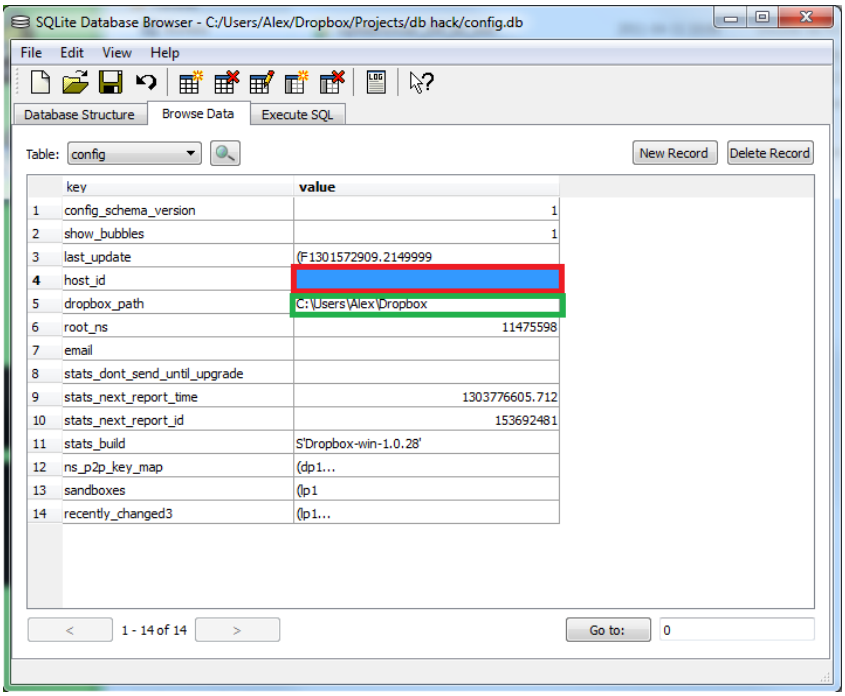
כפי שהוא פרסם, הקליינט של Dropbox שומר מידע אודות החיבור בקובץ SQLite על המחשב עליו הוא מותקן, במיקום:

```
%APPDATA%\Dropbox
```

המידע נשמר בקובץ בשם config.db, לאחר פתיחה של הקובץ וביצוע מחקר קצר, גילה דרק, כי שמורים בקובץ מספר ערכים, לדוגמא:

- Email
- Dropbox_path
- Host_id

הקובץ נראה כך:



[במקור: http://kittybomber.com/images/configdb.png]

המזהה הראשון (**Email**) שומר את כתובת האימייל של המשתמש, ממחקר שביצע דרק, נראה כי כתובת זו אינה נמצאת בשימוש בשום שלב של ההזדהות. המזהה השני (**Dropbox_path**) שומר את תיקיית הסנכרון שנבקעה במהלך ההתקנה של תוכנת הסנכרון על המחשב המקומי. ובסוף, המזהה השלישי, מזהה ה-**Host_id**, לפי המחקר שביצע דרק, הוא גילה כי מזהה זה הינו המזהה היחידי המקשר בין תוכנת הסנכרון לבין חשבון ה-Dropbox, על פי מזהה זה, תוכנת הסנכרון יודעת לקשר בין התיקייה בענן לבין תיקיית הסנכרון על המחשב המקומי.

מבדיקות שעשה דרק, הוא גילה כי קובץ ה-config.db (הקובץ השומר בתוכו מזהים אלו) אינו מקושר בשום מקרה ובשום צורה למחשב עליו הוא יושב, מה שאומר, **שבמידה וקובץ זה נגנב, ניתן להעתיקו לכל מחשב אחר ולגשת בעזרתו לחשבון ה-Dropbox של המחשב שממנו הוא נגנב.** מדובר בחולשה קריטית במערכת של Dropbox המאפשרת לכל מי שמשיג את הקובץ, או את המזהה השמור בקובץ להשיג גישה מלאה לכלל הקבצים בחשבון ה-Dropbox מבלי שהמשתמש ידע על כך (באותו הזמן, לא הייתה שום אינדיקציה לכך שמכשיר נוסף התנכרו עם החשבון שנפרץ.

ואפילו יותר מזה, מזהה זה אינו קשור אף לסיסמא המשתמש, כך שגם אם המשתמש זיהה או חשד שמישהו פרץ לו לחשבון ועל מנת להתגונן הוא שינה את סיסמתו, התוקף, בעל הקובץ הגנוב עדיין יכול לעשות בו שימוש ולהסתנכרן עם אותו החשבון. לא עבר זמן רב וכבר זוהו תולעים ווירוסים שחלק מתפקידם היה לגנוב קובץ זה.

באג דומה התגלה לפני שנים רבות בתוכנת הלקוח הרשמית של תוכנת המסרים המיידים ICQ. בתחילת שנות ה-2000 התגלה כי אחד מקבצי ה-IDX הנוצר בעת התקנת תוכנת הלקוח אשר אחראי על שמירת רב הנתונים בנוגע לתהליך ההתחברות לשרת, שומר את סיסמת המשתמשים באופן שניתן לשחזר אותה. מה שאומר שבמידה והתחברתם לחשבון ה-ICQ שלכם מהבית של אחד החברים שלכם – הם בקלות יתרה יוכלו להשיג את סיסמתכם.

כיצד ניתן להתגונן?

הבאג הספציפי הנ"ל כבר תוקן, אך אי אפשר לדעת מתי הבאג הבא יתגלה. נוכל להימנע מכך על ידי זה שלא ניתן גישה לכל מני וירוסים ותולעים למחשבינו, ולכן ננקוט באותם הצעדים שראינו עד כה. עם הדגשים הבאים:

- אין להתחבר לחשבונות האינטרנט שלנו ממחשבים שאנו לא מכירים (אינטרנט קפה, מחשב בקבלה של המלון וכו), כי, שוב, לכו תדעו מה עבר עליהם.

- לפראנואידים: ניתן לשמור קבצים מסגנון זה בכוננים מוצפנים, כגון כונני TrueCrypt וכדומה, ורק כאשר נרצה לסנכרן את התיקייה שלנו, נבצע Mount לכונן המוצפן, נשלוף את הקובץ, נשתמש בו, נחזיר לכונן המוצפן, ונבצע Unmount.

פרוטוקול התקשורת של המערכת

כמו שראינו קודם לכן, פרוטוקול התקשורת של המערכת אחראי על "השפה" בה מדברים תוכנת הלקוח אשר מסנכרנת את תיקיית הקבצים המקומית עם המידע הקיים בשרת, בדרך כלל, לשם הנוחות והיעילות חברות האחסון מפתחות פרוטוקול ייעודי לצורך פעולה זו, אך עם זאת, במקרים רבים אפשר לראות כי פרוטוקול זה הינו פרוטוקול פשוט יחסית ללא מאפיינים מיוחדים (כגון אבטחה, הצפנה וכו') והוא מסתמך בדרך כלל על שכבות נוספות על מנת למלא חוסרים אלו (כגון השימוש ב-SSL על מנת להבטיח את בטיחות תווך התקשורת וכו'). תוקפים יכולים לנצל את סוג החיבור, או חולשות שהתגלו בפרוטוקול עצמו על מנת לפגוע בנו ולקבל הרשאות שאינן אמורות להיות להם לחשבון שלנו בענן.

מתקפות Man In The Middle ומתקפות Rogue Routing נוספות.

מתקפות Man In The Middle ניתן לבצע בשלל דרכים, למען האמת, "Man In The Middle" אינה מתקפה מסויימת, אלא מצב הוא משיג התוקף לאחר ביצוע מתקפה אחרת המשפיעה על תצורת ניתוב המידע (Routing) של מספר רכיבים במערכת. על מנת לבצע Man In The Middle על התוקף לנתב את תעבורת המידע אל עבר נקודת ביניים הנמצאת בשליטתו (כדוגמת שרת שלישי, או המחשב ממנו הוא תוקף). קיימות מספר רב של מתקפות המאפשרות שליטה בתצורת ניתוב המידע, מתקפות כגון Arp Spoofing, מתקפות כגון DNS Spoofing / DNS Cache Poisoning, מתקפות Session Hijacking, מתקפות Evil twin ועוד.

לאחר שהתוקף ביצע בצורה מוצלחת מתקפה אשר מאפשרת לו שליטה על הליך ניתוב המידע בין תוכנת הסנכרון לבין שרתי הענן של חברת ה-Storage שלנו, הוא יוכל לבצע את רב העולה על רוחו (אם זה גניבת סיסמת ההתחברות שלנו לחשבון, או צפייה בתוכן המסתנכרן בין התיקיות וכו'), על מנת להתמודד עם צרות אלו, ספקי השירות הוסיפו מספר מנגנוני אבטחה שתפקידם למנוע ממקרים כאלו להתרחש, כדוגמת הוספת מנגנוני אבטחה שתפקידם לזהות מתקפות MITM ולהתריע על כך, או מנגנוני הצפנה מורכבים שתפקידם למנוע שינוי, זיוף או שליפת המידע באת התוועדות לתוכן המידע המועבר.

עם זאת, במקרים מסוימים, מתגלות חולשות במנגנונים אלו, כדוגמת החולשה שהתגלתה בקליינט של השירות Ubuntu One בתחילת חודש מרץ השנה. כחלק ממנגנוני האבטחה של הפרוטוקול, השירות Ubuntu One משתמש בפרוטוקול SSL המסתמך על תעודות המונפקות על ידי גורם שלישי (CA) המוכר והמוסכם על ידי שני הצדדים (תוכנת הסנכרון ושרתי הענן). החולשה התגלתה בתהליך אימות תעודת ה-SSL וניצולה אפשר לזייף תעודות SSL שיחשבו "כאמינות" על ידי תוכנת הסנכרון. למעשה, אם תוקף הצליח לבצע מתקפה ולהגיע למצב שבו הוא משחק כ-MITM, הוא יכל לנצל חולשה זו ולעקוף את מנגנוני האבטחה בפרוטוקול ה-SSL כפי שהוא מומש בתוכנת הסנכרון ולגרום לה לשלוח את פרטי ההתחברות לשרתים הנמצאים בשליטתו, או לסנכרן את תיקיית הסנכרון עם קבצים המכילים קוד זדוני ועל ידי כך להשתלט על עמדת הקצה.

כיצד ניתן להתגונן?

- כל השירותים שהוצגו עד כה מאפשרים שימוש ב-SSL, יש להשתמש בהם באופן קבוע. ישנן תוספים לדפדפנים כגון HTTPS Everywhere אשר יוודאו כי במידה ולא מתבצע שימוש בשירות באופן מאובטח (SSL) והשירות אכן מאפשר זאת, הפעולה תפסק ותבצע שימוש בתווך המאובטח.

מתקפות Data Tampering וחולשות לוגיות בפרוטוקול המערכת.

כאמור, ספקי השירות מפתחים פרוטוקולים יעודיים לשם נוחות הפיתוח ועל מנת לספק את הפונקציונאליות הנדרשת לשירות אותו הם מספקים, במקרים כאלו, יכולות להתגלות חולשות בפרוטוקול הייעודי, דוגמא מצוינת הינה החולשה שהתגלתה בפרוטוקול הסנכרון המקומי של שירות ה- Cloud Storage של DropBox. אחד השירותים המעניינים המציע פרוטוקול התקשורת של השירות הנ"ל הוא היכולת לבצע סנכרון בתוך רשת תקשורת מקומית (LAN) בין שתי תחנות קצה. זאת אומרת שבמידה ויש לי שני מכשירים שמחוברים לאותה רשת עם אותו החשבון, שניהם יעבירו את המידע ביניהם קודם כל, ורק לאחר מכן, יעלו אותו לשרת. מנגנון זה יעיל מאוד, אך נמצאה בו חולשה המנצלת את אותה העברת הקבצים בתוך הרשת. כחלק מהמתקפה מתבצעת התחזות לתחנה המתבקשת לסנכרן את המידע ועל ידי כך לגרום לאותו החשבון לשלוח אלינו את הקבצים המסונכרנים.

מתקפות Data Tampering הן מתקפות בהן התוקף משנה פרמטרים ומבצע מניפולציות במידע העובר על הפרוטוקול עצמו או במבצע הפרוטוקול על מנת לשנות את תגובת המערכת ועל ידי כך להשיג גישה מעבר לגישה שניתנה לו באופן טבעי.

כיצד ניתן להתגונן?

ברוב המקרים, כמו גם בזה, אין לנו יכולת לשנות את הפרוטוקול. כאשר תוכנה מתפרסמת בקוד פתוח, אנו יכולים לשנות את הקוד ולרב, להוסיף לו שכבות הגנה, וגם אם אנו לא נעשה את זה, קיים סיכוי כי אדם אחר יעשה זאת בשבילנו. במקרה הזה, הפרוטוקולים הם פרוטוקולים סגורים (Proprietary) מה שאומר, שאנו, המשתמשים, לא יכולים לראות את הקוד או כיצד באמת הפרוטוקול עובד. לא, זה לא אומר שאי אפשר לפרוץ אותו, זה רק אומר שצריך לעשות עבודת מחקר יותר מעמיקה. ניתוח של תעבורת הרשת ו/או תהליך Reverse Engineering יוכל להראות לנו את ההוראות הלוגיות שהתוכנה מצבעת ואנו נוכל להבין איך היא עובדת. אז זה בהחלט אומר, במידה ונמצא בעיה בקוד או חולשה אבטחתית, יהיה לנו מאוד קשה עד בלתי אפשרי לתקן אותה.

בשלב הזה ניתן רק לנסות לדאוג שה-LAN עליו אנחנו גולשים יהיה כמה שיותר נקי. מקומות כמו ארומה הם מקומות נהדרים לבדוק תוך כמה זמן הילד הן 14 שיושב שם ושותה את השוקו שלו תופס את כל הקבצים שלכם. מומלץ לגלוש במקום העבודה שלכם והבית בלבד, בתנאי שהם מספקים אבטחה נאותה לרשת. תמיד יש אפשרויות של שיפור מנגנוני האבטחה על ידי גלישה דרך VPN ויצירת טאנלים מוצפנים, אבל זה כבר למאמר אחר.

כיצד ניתן להתגונן?

ובכל זאת, מה שאנו יכולים לעשות הוא תמיד לשמור על ערנות ולהתעדכן באתר החברה ובאתרי החדשות בנושא, ולדאוג כי במידה ויוצאים עדכוני אבטחה לחבילות התוכנה, יש לבצע עדכון בהקדם האפשרי.

מתקפות Server Side

במקרים רבים מתגלות חולשות בשירות, שלנו, כמשתמשי קצה אין שום קשר אליהם ויכולת להתגונן או להימנע מהן. מקרים בהם שרתי החברה נפרצים, מקרים בהם נמצאות חולשות המאפשרות לעקוף את מנגנוני ההזדהות או לבצע שליפות מידע באופן ישיר ממסד הנתונים. במקרים כאלה, אין לנו בתור משתמשי קצה שום יכולת להתגונן, וברוב המקרים, מדובר בחולשות פאטאליות, חולשות אשר ניצולן מקנה לתוקפים גישה לאזורים קריטיים במערכת.

עם זאת, יש מקרים שלמרות כי בהם החולשות נמצאות על צד השרת המרכיב את המערכת, ולנו כמשתמשים אין שליטה על כך, התוקף נדרש לבצע מול המשתמש הנתקף אינטראקציה מסויימת (כניסה לקישור מסוים על מנת לגרום לו לבצע שינויי הגדרה בחשבון וכו').

כיצד ניתן להתגונן?

במקרים בהם כלל המתקפה מתרחשת כנגד שרתי המערכת עצמה - אין לנו, בתור משתמשים תמימים מה לעשות (מלבד לאתרה לפני התוקפים ולדווח עליה לחברה על מנת שתתקן אותה), אך אם על מנת לבצע את המתקפה בהצלחה על התוקף להגיש אינטראקציה עם בעל החשבון - ביכולתנו להתמודד עם הסיכון על ידי כך שנהיה זהירים, בדרך כלל, על המשתמש להיכנס לקישור שנוצר על ידי התוקף, או שעלינו להגדיר את חשבוננו באופן מסוים הרגיש לפעולה מסויימת שעל התוקף לנצל, מודעות לסיכונים אלו יכולה להציל אותנו מנפילה במתקפות כאלה.

בנוסף ביכולתנו להוסיף שכבת הגנה אישית על המידע המאוחסן בענן. אם נוכל להגיע למצב שבו רק המחשב שלנו מסוגל לקרוא את המידע (בשל היותו מוצפן, לדוגמא) - לא משנה אם תוקפים ישיגו גישה מלאה לשרתי האחסון של החברה, המידע שהם ימצאו בחשבונותינו יהיה חסר ערך עבורם, מפני שהוא יהיה מוצפן.

נציג כאן דוגמא למערכת הפעלה מבוססת לינוקס (אבחר באובונטו לשם כך). החבר'ה מאובונטו היו נחמדים מספיק בשביל להכניס להפצה שלהם תוכנה שיודעת להצפין מערכות קבצים, היא נקראת EncFS. הדבר לא דומה להצפין קובץ ספציפי כגון RAR, הרעיון כאן, הוא שכל מערכת הקבצים מוצפנת ובלי המפתח אין יכולת לפענח אותה. בעת התקנת מערכת ההפעלה, אובונטו מציעה לכם להצפין את כל תיקייה הבית שלכם בעזרתה, כך שגם אם המחשב ייגנב, המידע עליו עדיין יהיה מוגן ובלי הסיסמא לא ניתן יהיה לראות אותו.

במידה ואין לכם את התוכנה מותקנת אצלכם על המכונה תוכלו להוריד אותה בעזרת הפקודה:

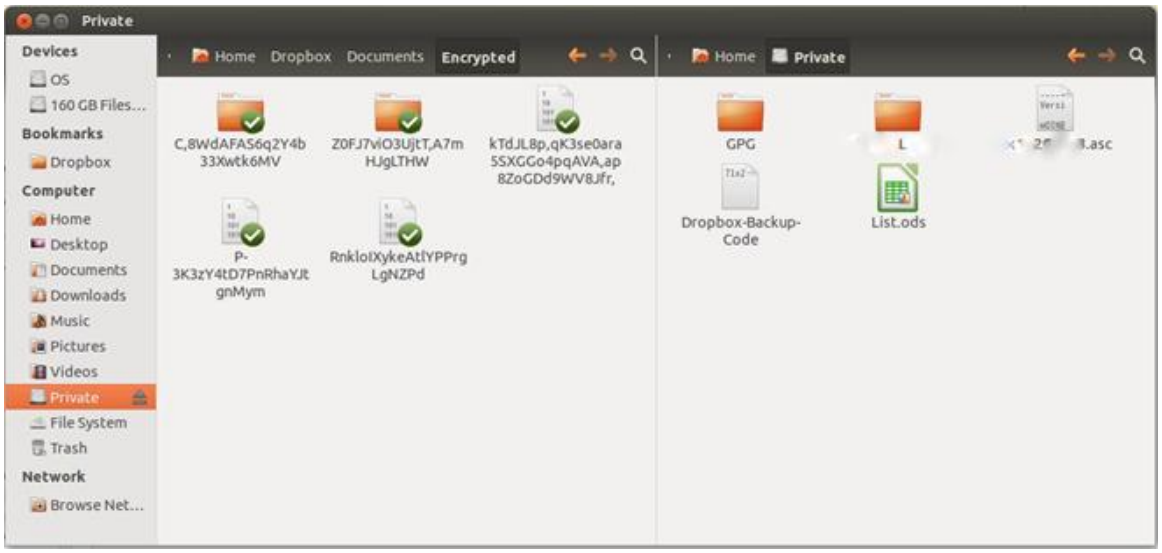
```
sudo apt-get install encfs
```

לאחר מכן, עלינו ליצור כונן חדש, נעשה זאת בעזרת encfs באופן הבא:

```
encfs ~/Dropbox/EncryptedDrive ~/EncMapped
```

תפקיד הפקודה הזאת היא להריץ את התוכנה encfs ולקחת נתיב שנמצא תחת תיקייה הבית (במקרה הזה בתוך התיקייה "Dropbox") ולמפות אותו לנתיב אחר.

לאחר הקלדת הפקודה הראשונה נצטרך להגדיר אוסף של אפשרויות על איך יוצפן הכונן שלנו, מה הסיסמא שנשתמש בה, האם להצפין את שמות הקבצים ועוד. לאחר מכן, התוכנה תמפה את הכונן המוצפן החדש עם המידע השמור בתוכו. **שימו לב שאין לאבד את הסיסמא מכיוון שאין אמצעי שחזור.** הדבר צריך להראות כך:



טיפ נוסף, לשם הנוחות:

לאחר הפעלה מחדש של המחשב הכונן יעלם ונאלץ למפות אותו מחדש. לצורך זה, אמליץ לכם להשתמש בטריק שמבצע שימוש ב-bashrc., הנקודה בתחילת שם הקובץ אומרת שהקובץ הוא קובץ נסתר, אך תוכלו לערוך אותו ישירות על ידי הקלדת הפקודה:

```
gedit . bashrc
```

תוכן קובץ זה הוא אוסף של פקודות אשר עולות ברגע שאנו מעלים את הטרמינל שלנו (לא ברגע שמערכת ההפעלה עולה, אלא ברגע שאנו מדליקים את הטרמינל). אחת הפקודות הנוחות והשימושיות בלינוקס הן פקודות alias ואנחנו ממליצים לכם להוסיף כאלה לפי ראות עיניכם אשר אחת מהן שמאוד נוח להוסיף תראה כך:

```
alias encmap='encfs ~/Dropbox/Encrypted ~/Private'
```

אחרי שתוסיפו את השורה הנ"ל, שמרו את הקובץ, סגרו את החלון ואת הטרמינל והעלו אותו שוב. כאשר תקלידו את הפקודה הזאת, הוא יבקש מכם את הסיסמא של מערכת הקבצים המוצפנת וימפה אותה בקלות רבה. כאן זה המקום להציג יכולות נוספות שניתן להוסיף לקובץ הנ"ל, יובל ריכז מספר רב מהן בקובץ הבא, לנוחיותכם:

http://pastebin.com/4qJYnLdw

לסיכום

כיווני התקיפה של מערכות ה-Cloud Storage הן רבות ומגוונות. קיימים תקדימים לכולם וגם לשרתי חברות שנפרצו ואף לפרוטוקולים לא תקינים בתוך החברה עצמה. מומלץ להימנע משמירת מידע רגיש במקומות אלה וכן להקפיד על שמירתם בדיסק מקומי. במידה ואתם נדרשים לשמור את אותם הקבצים על חשבון כזה או אחר, מומלץ לוודא את בריאות מערכת (אנטי וירוס מעודכן, Firewall ועוד) קודם לכן, סיסמא חזקה המורכבת לפחות מ-15 תווים (אותיות קטנות, גדולות, מספרים ותווים מיוחדים), עבודה עם אימות דו שלבי (הקיימת כמעט בכל הגופים האלה). ובמידה והמידע באמת רגיש לכם, תמיד יש את אפשרות הצפנת מערכת הקבצים עצמה בעזרת TrueCrypt או EncFS על מנת לוודא שגם אם החשבון יועמד בסיכון ויידלוף, המידע עדיין יהיה בלתי נגיש.

קידום עצמי חסר בושה

יובל נתיב, בן 23, מדריך בקורס Hacking Defined Experts, בודק חדירה וחוקר אבטחת מידע ב-See-Security ובעל בלוג אבטחת מידע ישראלי בשם "אבטחה":

https://avtacha.wordpress.com