LAIR - מאורת המטמון של אליבאבא

מאת ליאור ברש וישי גרסטל

הקדמה

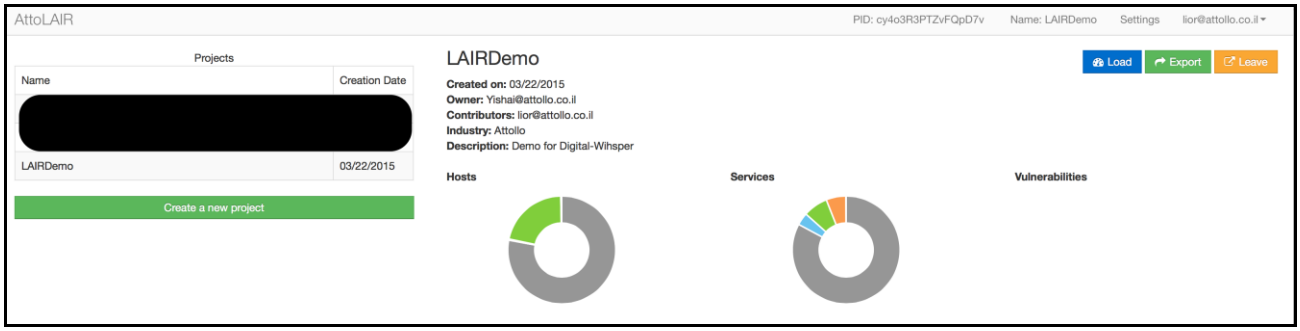
דמיינו בעיני רוחכם את צוות השודדים של אליבאבא, 40 שודדים זו אופרציה מורכבת, כבר לא חברה של איש אחד, זו דינאמיקה שדורשת שיתוף פעולה ושיתוף מידע, ניהול מרכזי ובקרה, על השוד וגם על האוצר.

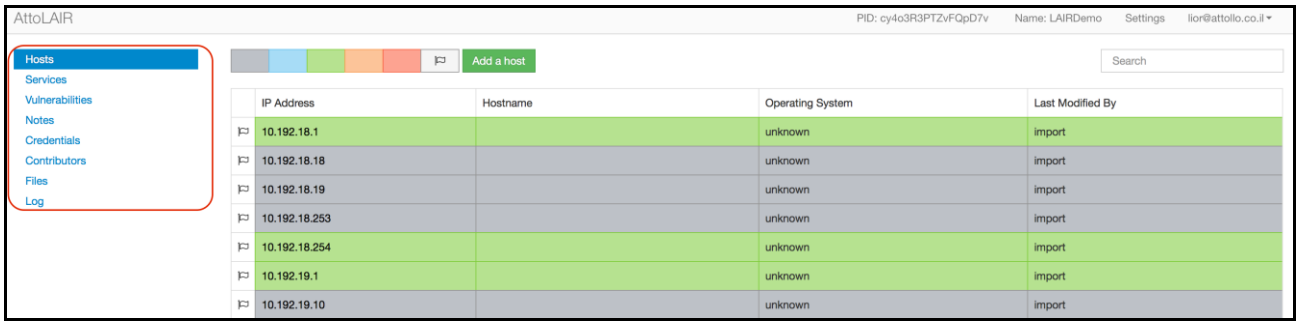
בצוות של אליבאבא יש 40 שודדים ויש את אליבאבא עצמו, הוא עצמו לא ממש מנהל את הצוות אלא יותר מפקח עליו ומתעניין בתוצרים שלו. את אליבאבא מעניין להבין לאיזה בית כבר פרצו ואם נכנסו מהחלון או הדלת, מעניין להבין אם מצאו כספת או כד עם זהב, מעניין מי פרץ לאיזה בית ואם הספיקו לבדוק את המרתף או שלשם עוד יחזרו.

אפקטיביות וסדר הן מילות המפתח, יש צורך במערכת ניהול מידע שמאפשר לנו להיות עם האצבע על הדופק ולדעת מי עושה מה, מה התוצרים, מה נשאר פתוח ומחכה לטיפול ולא פחות חשוב, מה בכלל לא ראינו.

2015 והסביבה היא אחרת, אבל הבעיה והמורכבות עומדות בעינן, צוות של תוקפים שוקד על רשת של לקוח, מחפש דלתות וחלונות, מחפש סדקים בקירות ומפתחות בארון החשמל. יש 40 שודדים, לכל אחד יש כרטיס רשת שתומך בהזקרה, מקלדת מוארת ואצבעות שמקלידות לאט רק בקצת מקצב הופעת הרעיונות בראש.

עכשיו יש גם מערה, LAIR, שבה שומרים את המודיעין ואת התוצרים שמצאנו עד עכשיו, שם מסמנים איפה היינו ומה ראינו ולאן אנחנו הולכים מחר, שם הגביע הקדוש וכל מטבעות הזהב.





אז כשיושבים קבוצה של תוקפים על מערכת, לכל אחד יש את המסלול והמתודולוגיה שלו, לכל אחד יש את הכיוון והיצירתיות, את היכולות והתפיסה שמפרידה אותו מייתר הצוות. מצד אחד אסור לפגוע בזה אך מצד שני חייבם לנהל את זה. והנה זה קורה.

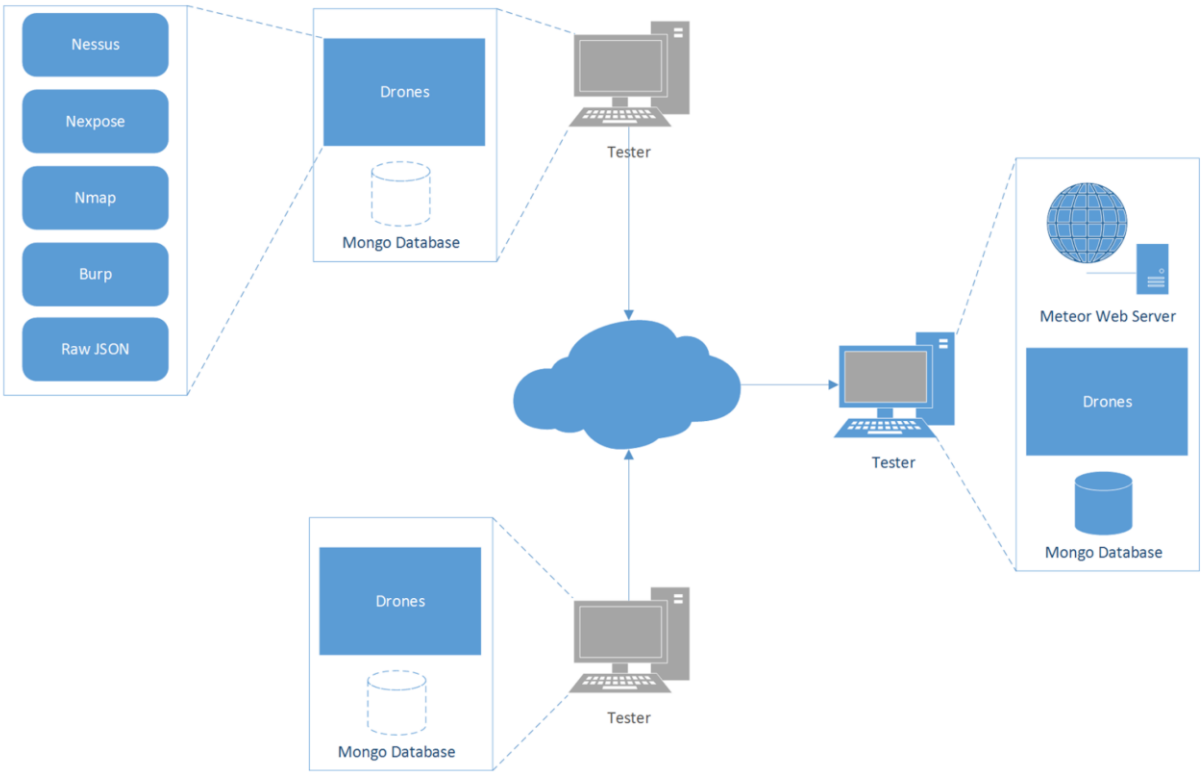
LAIR שנכתבה ע"י Tom Steele ו-Dan Cottman מייצרת לנו סביבת עבודה משותפת שחוסכת עבודה כפולה, מנגישה את התוצרים שיש בזמן אמת, ושומרת את המידע נגיש ככה שאם חזרת למבדק אחרי תקופה, נשארת בעניינים. את המידע אגב לא צריך להזין ידנית, המערכת יודעת לקבל יצא של תוצרים ממספר כלים.

LAIR - ארכיטקטורה של מערה שיתופית.

LAIR מבוססת קוד פתוח וזמינה דרך Github. מבנה המערכת נשען על מודל Back & Front End כאשר ה-Front מעבר להצגת המידע מאפשר לשודדים לבצע פעילויות ב-Client side וזה אומר שהכל מאוד מהיר. מאוד!

ה-Front בנוי על Meteor, שזו בפני עצמה פלטפורמה קוד פתוח שמאפשרת לבנות ממשקי עבודה ל-Web- ול-Mobile על בסיס Javascipt והמהירות בגדול נובעת מכך שהעדכונים מתבצעים קודם כל בצד המשתמש ורק לאחר מכן נשלחים לעדכון בשרת, כל זה קורה מהר מאוד אבל יש למודל את הסיכונים הידועים של צד משתמש. מהצד השני יש את האפליקציה שנשענת על MongoDB שהוא עצמו בסיס נתונים מבוסס noSQL זריז ומוכן ל-Scale.

הלכה למעשה, ככה זה נראה:



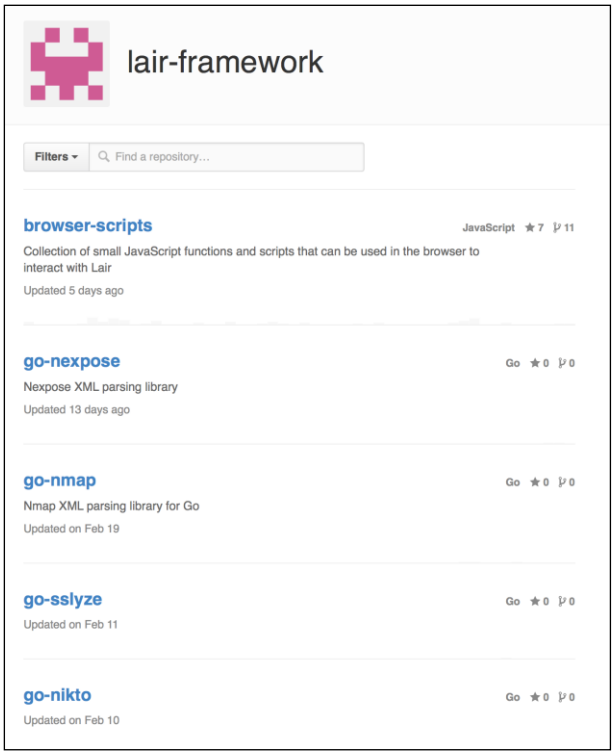
החלק שעדיין חסר בסכימה הוא ה-Drone. זה רכיב שממיר את המידע שמוזן אל ה-LAIR ממגוון כלים או סטנדרטים ומזין את המידע לשרת המרכזי. אגב, ההתקנה למי שתוהה, פשוטה מאוד. מורדים את ה-7z. ומחלצים, ממש next next...

עדיין חשוב לוודא שההזדהות מול ה-DB תקינה ושהתקשורת מה-Drone למערכת המרכזית מאובטחת כדי שלא נשלח את הזהב שלנו למערה לא מכוסה.

עכשיו כשהמערכת באוויר התהליך די פשוט:

- פותחים פרויקט חדש
- קובעים לו משתתפים
- מתחילים להזין מידע
- צובעים, רק בתוך הקווים

מה שמעניין בעיקר אלו האפשרויות שיש לעשות עם המערכת ברגע שהמידע כבר בפנים.



על מנת לשלוח את המידע למערכת מפעילים סקריפט שרלוונטי לכלי שממנו רוצים להעביר את המידע, הסקריפט צריך לדעת מאיפה הוא לוקח את המידע ולאן. נכון להיום התהליך הזה הוא ידני וזה לא הכי נוח או יעיל.

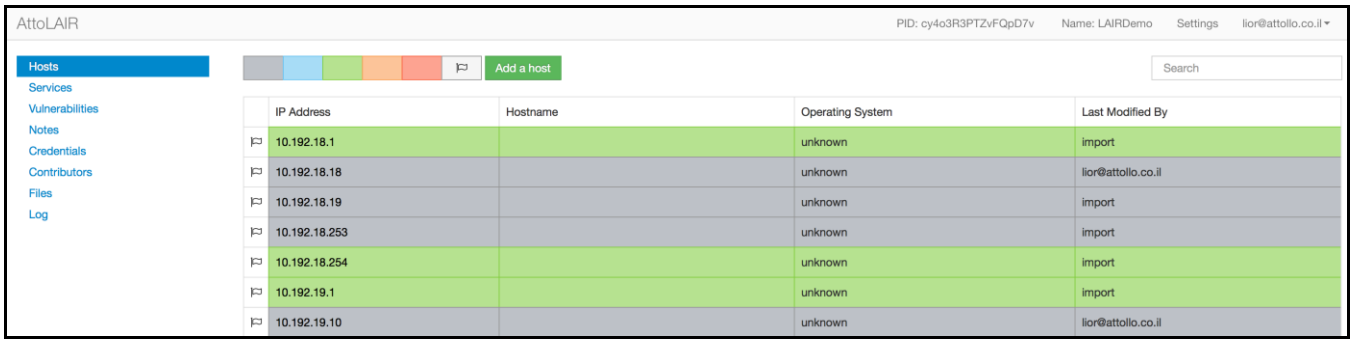
תהליך צביעת המידע גם מובצע ידנית וע"י שימוש בסקריפטים, מוכנים מראש או כאלו שתוכלו אתם לכתוב. ויש שני כיוונים עיקריים שמערכת מאפשרת לעבוד איתם. סקריפטים מבוססי Python או GO כשרוצים להעביר מידע מהמשתמש למערכת. וסקריפטים מבוססי JS כשרוצים לבצע תהליכים על המידע שכבר הועבר.

באופן כללי יש תנועה של מפתחי המערכת לעבור ל-GO.

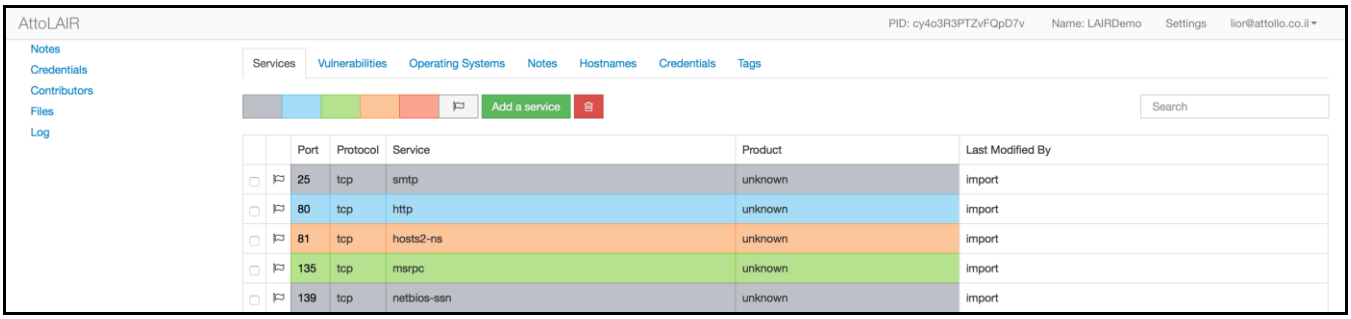
כאשר המידע עולה, הוא מסודר יפה אבל לא הכי נוח לקריאה:

	192.168.81.5		unknown	nmap
	192.168.81.6		unknown	nmap
	192.168.81.7		unknown	nmap
	192.168.81.8		unknown	nmap
	192.168.81.9		unknown	nmap
	192.168.81.10		unknown	nmap
	192.168.81.11		unknown	nmap
	192.168.81.12		unknown	nmap
	192.168.81.13		unknown	nmap
	192.168.81.14		unknown	nmap

וזה מוביל לשלב שבו רוצים לנוע יותר בחופשיות ובקלות בכמויות המידע שנצברות דיי מהר במערכת, כאן נכנסים ה-Browser scripts לתמונה ומשנים את הנראות באופן משמעותי. את שינוי ועדכון הפרטים ניתן כמובן לעשות גם ידנית:

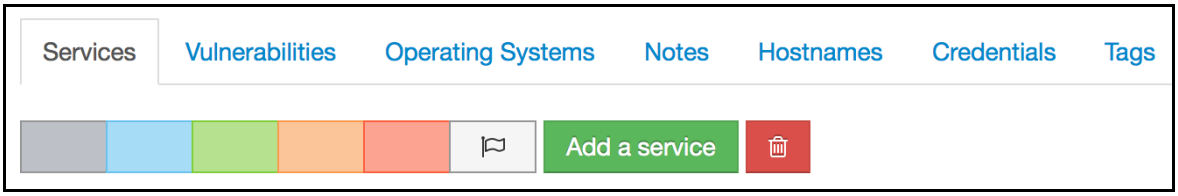


המערכת יודעת גם לוודא שלא תזינו מידע כפול כך שגם אם חזרנו ללקוח אחרי תקופה נוכל לקבל את התוספות באופן חלק לתוך מאגר המידע שלנו:



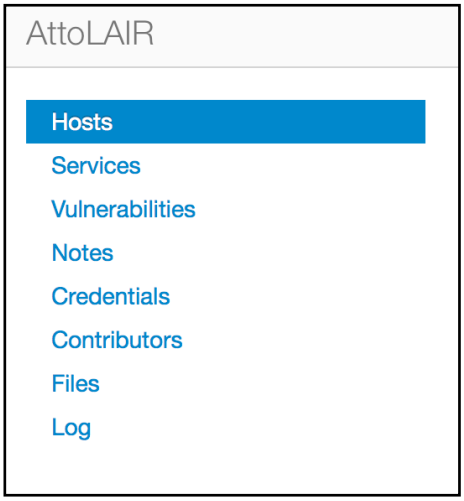
את המידע הרלוונטי לכל כתובת שנסרקה אפשר לקבל בלחיצה פשוטה וגם שם המידע נצבע בנוחות, בין אם כדי לסמל רמת סיכון, סטאטוס או הערה וגם כאן, המערכת מעדכנת דלתאות, אם התווספו פורטים על מערכת שסרקנו, פשוט נקבל את תוספת המידע החדש לתוך האובייקט.

בתוך כל אובייקט יש אפשרות לבצע מחקר עומק על המידע הרלוונטי לאותו האוביוקט בדיוק כפי שניתן לבצע את החיתוכים האלו על המערכת כולה, כך שניתן לפלח משתמשים וסיסמאות, חולשות, שירותים הערות, קבצים ועוד..



לסיכום

המערכת לכשעצמה עדיין לא בשלה מספיק כדי להיות כלי עבודה פונקציונאלי מלא לקבוצות התקפה אך זו התחלה מאוד מרשימה. היו חסרים לנו למשל אפשרויות ניהול לקוחות מסודר יותר ותהליכי הזדהות משמעותיים יותר, בכל זאת מדובר בתיבת המטמון!



אבל ככה זה, צריך להתחיל איפשהו, והמערה הזו היא התחלה מצויינת.

על המחברים

- ישי, שד טזמני וחובב קוד פתוח, מחבר וחוקר מערכות בעיקר כדי לפרק אותן.
- ליאור, חופר 24/7 ומעניש על זה את המקלדת.