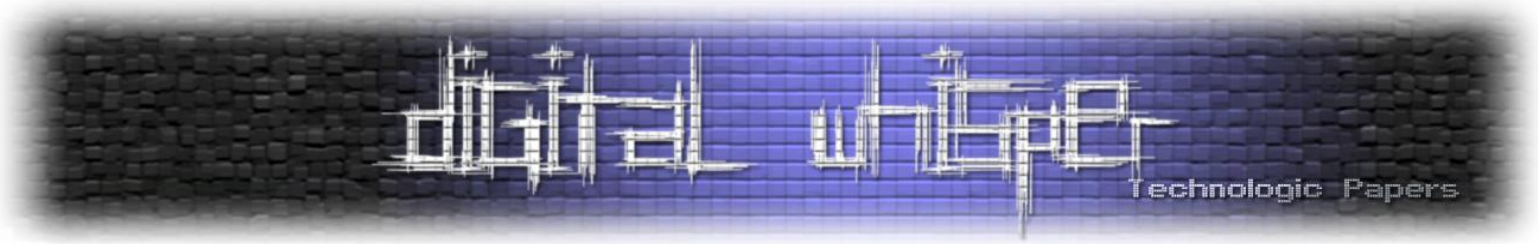


IDS- Intrusion Detection System

מאת נתנאל שיין

"If you spend more on coffee than on IT security, you will be hacked. What's more, you deserve to be hacked." - White House Cybersecurity Advisor, Richard Clarke

הקדמה

דמיינו לעצמכם עולם שבו כל המידע החיוני שלנו מאוחסן בצורה דיגטלית אך אין אף מערכת אחת שתגן עליו. כמובן שבמקרה כזה, כל המידע שלנו יהיה חשוף להתקפות חוזרות ונשנות של קראקרים , גנבות, פגיעות ברכוש ואף סכנה לבטחון הלאומי. עם העליה שחלה בשנים האחרונות בהעברת כל המידע הקיים ברשותנו לצורתו הדיגיטלית, עולה למודעות גם הצורך במערכות שונות אבטחה. על החשיבות האדירה שיש למערכות אבטחה בימינו, ניתן ללמוד מדוגמאות מוכרות כגון זו של אהוד טננבאום. טננבאום, שכונה האנלייזר, הוא קראקר ישראלי שהתפרסם בשנת 1998 כשנתפס על ידי ה-FBI, לאחר שפרץ למחשבים של נאס"א, הפנטגון, הכנסת והצבא האמריקאי, ובחלק מהם שתל תוכנות מסוג Sniffer וסוס טרויאני .

כעת, לאחר שהבנו את ה-"למה", נתחיל לדבר על ה-"מה". תחת הכותרת "מערכות אבטחת מידע" ישנם נושאים רבים ומגוונים, אחד מהם הנו ה-Intrusion Detection System- או בקיצור: "IDS" - מערכת לזיהוי חדירות. בתחום זיהוי החדירות ישנן מספר מערכות, על שתיים מהן אפרט במאמר זה:

- הראשונה: מערכת לזיהוי חדירות על בסיס המארח HIDS (קיצור של Host-based Intrusion Detection System).
- השניה: מערכת לזיהוי חדירות ברשת NDIS (קיצור שלNetwork Intrusion Detection System)

מאמר זה יעמוד על החלקים המשותפים בסוגי המערכות, על ההבדלים העיקריים ביניהן, ועל השימושים הקיימים לכל סוג של מערכת.

הצורך במערכות IDS

ראשית, חשוב להבהיר כי מטרתן של מערכות IDS אינה להחליף מערכות הגנה אחרות, כדוגמת מערכת Firewall וכדומה.
על מנת להסביר את תפקידן של מערכות ה-IDS אשתמש בדוגמה הבאה: נניח שתולעת נכנסת למחשב, אחד הדברים הראשונים שהיא עושה הוא כמובן לשכפל את עצמה ולהגיע לכמה שיותר קבצים, על מנת להסוות את עצמה ולהקשות את הסרתה. כמו כן, התולעת תוריד קבצים זדוניים למחשב אשר עלולים אף לגרום למחשב להיות שותף לעבירות הפורץ.

דוגמה אחת לכך היא מתקפות ה-DDoS בהן התולעת עשויה לבצע התקפות מהחשב אליו פרצה או כלפי המחשב עצמו. אחת הדרכים למזער את הנזקים הנגרמים מ התקפות מסוג זה, היא לשמור על המידע הרגיש, כלומר לגבות אותו על בסיס קבוע, הבעיה מתחילה כשעולה בפנינו האפשרות שגיבינו את המידע שלנו יחד עם התולעת המדוברת או מזיק מסוג אחר. כלומר, אין לנו דרך לדעת איזה קובץ נדבק בתולעת ועבר שינוי. כמו כן, הגנה על המידע באמצעות גיבויו תקפה רק עבור קבצים שנמצאים בתוך המחשבוכאן עולה השאלה כיצד ניתן להתמודד במקרה בו יש שינויים חשודים בתעבורה (כדוגמת DDoS) וה-Firewall לא מצליח להתמודד עם האיום (למשל 0-Day).

בדיוק בנקודות אלה נכנסות לתמונה מערכות IDS. מערכות IDS נועדו להוות את מערך ההגנה האחרון, אחרי שאזלו כל יתר האמצעים שברשותנו. מערכות מסוג זה לא נועדו להגן ישירות, אלא להתריע למנהל המערכת כאשר דבר-מה משתנה, ולעקוב אחר מדיניות האבטחה של המערכת. ישנן שני סוגי מערכות שהשימוש בהן נפוץ. הסוג הראשון נקרא HIDS- מערכת לזיהוי חדירות על בסיס המארח, וסוג שני הנקרא NIDS- מערכת לזיהוי חדירות ברשת.

מערכת HIDS - Host-based Intrusion Detection System

מטרתה העיקרית של מערכות מסוג זה היא להשגיח על קבצי מערכת חשובים, אך כמובן ניתן להגדיר לה מטרות נוספות. המערכת מנטרת ומחפשת את השינויים ע"פ מדיניות האבטחה שהוגדרה לה מראש. לדוגמה, קובץ מערכת מסוים השתנה, מדיניות האבטחה של כניסה באמצעות ה-SSH השתנו (מערכת ה-HIDS תודיע על כך לאחראי). עיקרון הפעולה של המערכת מתבסס על כך שפורצים לרב ישאירו אחריהם עקבות מסויימות לאחר הפריצה. למשל, סקריפט שיאפשר לפורץ גישה למחשב בכל עת בלי לפרוץ אותו מחדש.

בהתבסס על עקרון זה, אפילו אם הפורץ ישתמש בפריצה מסוג חדש (ZeroDay) שתנצל חולשה מסויימת במערכת, הוא ברב המקרים ישאיר עקבות בקבצים שעליהן תרצה להגן, וככה גם מערכת ה-HIDS תזהה את הפריצה.

עקרון הפעולה של מערכת ה-HIDS מתחלק לשלושה שלבים, אם כי בתוכנות שונות יכולים להיות שינויים קלים באחד או יותר מהשלבים:

שלב אתחול הנתונים - השלב החשוב ביותר:

המערכת תקרא את קובץ המדיניות המוגדר (אם לא מוגדר, היא תקרא את קובץ המדיניות במצב ברירת מחדל) ותיצור קובץ בסיס נתונים ראשוני. בסיס נתונים זה הוא "התמונה" של מצב כלל האובייקטים בתוך המערכת, ואליו המערכת תשווה את הבדיקות שלה.

הערה: מכיוון ששלב אתחול הנתונים יוצר בסיס נתונים ראשוני, אותו בסיס הוא החשוב יותר מכל, מכיוון שממנו יתבצעו כלל הבדיקות של המערכת. מומלץ בחום לג בותו, ולהשתמש בבסיס נתונים זה ממדיה קריאה בלבד (לדוגמה תקליטור) ורק ממנה לבצע את הבדיקות.

שלב בדיקת האמינות:

מערכת ה-HIDS סורקת את המערכת ומחפשת הפרה של המדיניות שהוגדרה. על פי מדיניות האבטחה, המערכת תשווה את מערכת המחשב במצבה הנוכחי, לאותו בסיס נתונים ראשוני.

יצירת דוח מצב:

לאחר פעולת ההשוואה של מערכת המחשב לבסיס הנתונים, המערכת תייצר דוח מפורט על כל הפרה שנעשתה מדיניות האבטחה.

מה שקורה מרגע יצירת דוח המצב תלוי בהגדרות שנקבעו. במידה וקיימת הפרת מדיניות, אותו דוח יוגש למנהל המערכת והמנהל יוכל לבחון את הבעיה ולטפל בה. לדוגמה: מדובר בבעייה שבה בקובץ מסויים נוספה שורה מיותרת שלא היתה שם קודם, המנהל יוכל להיכנס לקובץ, ולערוך את השורה בחזרה. כמו כן, הוא יוכל לראות את מקור השורה שנוספה ואת האופן שבו היא נוספה. אם מדובר בשינוי נחוץ בקובץ, המנהל יוכל גם לעדכן את בסיס הנתונים הראשוני כך שיכלול את השינוי הנחוץ, ובכך למנוע את הופעת ההתרעה שוב.

בנוסף, חשוב לציין כי למערכת זו אפשרויות פעילות נוספות ושימושים רבים שלא הוזכרו במאמר.

מערכת NIDS - Network Intrusion Detection System

מערכת NIDS הינה מערכת לזיהוי חדירות ב סביבת הרשת. היא אינה פועלת כמו מערכת ה -HIDS ולא מחפשת אחר שינויים בקבצים במחשב, אלא מחפשת אחר שינויים חשודים ברשת (כגון התקפות DDoS המכוונות אל המחשב או נסיונות פריצה למחשב תוך כדי שימוש בכוח ברוטאלי וכדומה.)

אופן פעולת המערכת:

מערכת ה-NIDS מנטרת את כל התעבורה הנכנסת לרשת (רוב התוכנות העדכניות כיום מנטרות גם את התעבורה היוצאת מהשרת) ומחפשת תבנית שתתאים למדיניות האבטחה שלה. לדוגמה, אם ישנן יותר מדי ניסונות כניסה למערכת דרך ה-SSH מכתובות שונות שלא הוגדרו בקובץ המדיניות. כפי שכבר הוזכר, המערכת לרוב לא מנטרת רק את התעבורה הנכנסת, אלא גם את היוצאת, ולכן אם ישנה הפרת מדיניות שפועלת מתוך המחשב (כמו למשל התקפות DDoS שיוצאת מהמחשב עצמו, או עומס על תעבורת הרשת בשעה לא סבירה) המערכת תזהה את הבעיה.

לאחר מכן, המערכת תייצר דו"ח מצב מסודר, המבוסס על הניטור שנעשה בתעבורת הרשת, ותאפשר למנהל המערכת לבדוק מה קרה, מה מקור הבעיה וכיצד זאת נוצרה.

יתרונותיה של מערכת זו:

- המערכת יכולה להיות כמעט בלתי-נראית לפורץ.
- המערכת יכולה להיות משולבת עם מערכת Firewall כלשהי על מנת ליצור מערך דינאמי לזיהוי ובלימת חדירות.
- המערכת מספקת פירוט רחב על מצב הרשת.

מערכות IDS הינן נושא גדול ורחב תחת הכותרת של אבטחת מידע וניתנות לשילוב במגוון דרכים מערכות אחרות. לדוגמה, מערכת שמתריעה למנהל האבטחה בארגון כאשר משתמש מנסה להתחבר לחשבון שלו, תחת סיסמה שגויה מספר מסויים של פעמים, דוגמה נוספת, מערכת שמתריעה על כניסה לחשבון בשעה לא סבירה, ואפילו מערכת שלומדת את המשתמש.

רוב המערכות האלו משתמשות באחד מתוך שני מודולים נפוצים:

- Anomaly Detection
- Misuse Detection

דוגמאות לשימוש במודולים בבנק לאומי:

שימוש במערכת IDS שלומדת את המשתמש תוכל להיות למשל מאחורי מנגנון הזדהות מסויים, המערכת תרשום כל שעת כניסה ויציאה, וכך אם במשך תקופה מסויימת, המשתמש נכנס רק בין השעות 12:00 עד 16:00, המערכת לומדת את שעות הכניסה הקבועות שלו, ויוצרת מדיניות אבטחה מותאמת לו. על כן, אם לפתע תרשם במערכת כניסה בשעה 04:00, הדבר אינו יתאם את שעות הפעילות הקבועות של המשתמש ותהיה התראה על כך למשתמש.

דוגמה זו מבוססת על מודל שנקרא "Anomaly Dtection" כלומר, זיהוי חריגות. היא משתמשת במודלים שנבנו והוגדרו על פי נתוני הרגלי השימוש של המשתמש במערכת על מנת לתאר שימוש "רגיל" במערכת

המחשב (לדוגמה שימוש בין השעות x ל-y יוגדרו כשעות רגילות ומה שמחוץ לשעות אלו- יחשבו לחריגה במערכת).

דוגמה נוספת היא מערכת להעברת כספים, אם המשתמש במערכת ינסה להעביר סכום כסף גדול במשך כמה ימים ברצף (למשל 6000 ש"ח, הסכום המירבי שנקבע על ידי בנק ישראל ליום פעילות כיום) בשעה 08:00 בבוקר במשך כמה ימים סביר להניח שהמערכת תתריע . בדוגמה זו, המערכת מבוססת על מודל שנקרא Misuse Detection- כלומר, זיהוי שימוש לרעה. היא משתמשת במודלים שמנטרים את המחשב (HIDS) או את הרשת (NIDS) ומשווה אותם לתבניות או חתימות ידועות שמתעדכנות כל הזמן ומזהות התנהגות זדונית (זוהי פעולה דומה מאוד לתוכנת אנטי וירוס).

אתם בטח שואלים את עצמכם- איך אתה יודע שמודולים אלה מיושמים בבנק לאומי? שאלה מצויינת. את המידע קיבלתי משלמה פרגמנט- סגן ראש מערך תפעול ומנהלה בבנק לאומי, בראיון שנתן לאתר ידיעות אחרונות תחת הנושא: "איך הבנק מגן עליכם".

אני חייב לציין פה, למרות שהמאמר לא נועד לדבר על בנק לאומי , שחשיפת מידע על מאפייני האבטחה של ארגון שצריך לשים את האבטחה למראשותיו , מעמידה את הארגון במצב פגיע ואיננה מעשה רצוי לארגון שכזה. מידע נוסף על מערכות האבטחה בבנק לאומי ניתן לקבל בלינק שאספק בסוף המאמר.

חסרונות בסוגים ובמודולים של מערכות IDS

מודל Misuse Detection:

- חסרונו של מודול זיהוי שימוש לרעה מגיע מתפקודו- שימוש במודולים המשווים נתונים לתבניות או חתימות ידועות. כל התבניות והחתימות נבנו מראש, ולכן תווצר בעיה כאשר המערכת תותקף על ידי גורם זדוני חדש ולא מוכר. ולכן, חשוב מאוד שהמודל הזה יהיה עדכני.

- גלאים במודל זה לרוב נכתבים מחתימות ספציפיות ומותאמות אישית, דבר שמונע מהם לזהות התקפות בסיסיות שנכתבו מחדש.

מודל Anomaly Detection:

- מכיוון שמודל זיהוי חריגות פועל על ידי שימוש במודולים שנבנו על מנת להגדיר שימוש"רגיל" במערכת, מערכות שמתמשות במודול זה סובלות לרוב ממספר גדול של אזעקות-שווא, היות ולא כל המשתמשים מתנהגים באופן זהה ולא כל תעבורות הרשת זהות, לכן ישנן הרבה התראות על שימוש שאין בו שום פעילות זדונית.

HIDS:

- יישום מערכת מסוג זה בארגון גדול יכול לגרום לסיבוכים רציניים בארגון שבו כמות המחשבים עצומה, איסוף וניטור לוגים של כל מחשב בפני עצמו תהיה למשימה קשה מנשוא (מה שווה מערכת שכזו אם אף אחד לא יקרא את הדוחות שהיא מייצרת?).

- אם מערכת ה-IDS נפרצת ואיסוף הלוגים נפסק, המערכת מושבתת ואם הלוגים המשיכו לפעול, אי אפשר להסתמך עליהם כמקור מהימן.

NIDS:

- שימוש ב- NIDS יתן כיסוי גדול על תעבורת הרשת. מערכת שלא מוגדרת כהלכה יכולה לגרום למספר גבוה של אזעקות שווא. בדיוק כמו בסיפור "זאב-זאב", כאשר חברה תראה שהיא מתעסקת יותר באזעקות שווא מאשר באזעקות אמיתיות, המערכת תוחלף, או גרוע מכך, איש לא יתייחס להתראות שהיא מייצרת.

דוגמה למערכת לזיהוי חדירות על בסיס המארח Open Source Tripwire - OST:

מערכת OST מבוססת על מערכת Tripwire Inc (הגרסא המסחרית לתוכנה) הינה תוכנה לזיהוי חדירות על בסיס המארח (HIDS) ומטרתה היא בדיקת אמינות קבצים למערכות יוניקס. בעברית אגב, השם שלה הוא חוט הפעלה (למלכודת כלשהי). התוכנה מנטרת את הקבצים על פי בסיס הנתונים הראשוני שהוגדר ומתריעה כאשר יש שינוי בקובץ שהוגדר להשגיח עליו.

ל-OST ישנן חלופות רבות מהקוד הפתוח לדוגמה: Samhain, OSSEC ו-AIDE, חלקן מציעות תוספות וכלים שונים. לדוגמה: Samhain, תוכנה מומלצת לכל הדעות שמספקת יכולות כגון: הסרת Rootkits במערכת, ניטור פורטים ועוד. בנוסף יש לה ממשק web אשר נועד לשליטה קלה ונוחה בתוכנה שנקרא: Beltane.

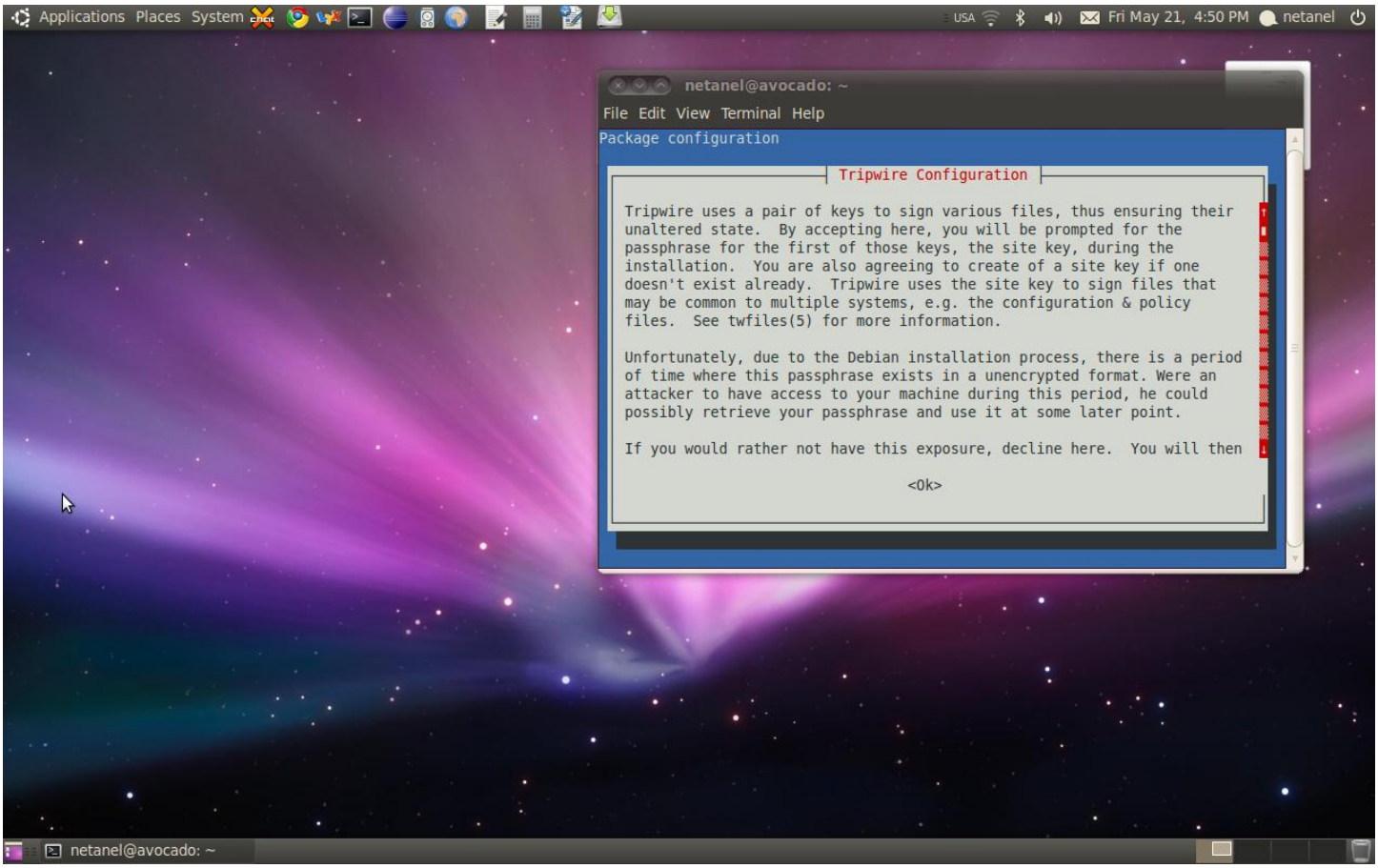
בחלק הזה אתמקד בתכלס- איך מקימים את מערכת ה-OST.

במערכת ההפעלה אובונטו, על מנת להתקין את המערכת נצטרך לרשום במעטפת הפקודה הבאה:

```
sudo apt-get install tripwire
```

בשאר המערכות יש להיכנס לאתר שלהם, להוריד את הקובץ ולקרוא את ה-README, שיסייע לכם בעת הידור והרצת התוכנית.

מיד בתהליך הראשוני לאחר ביצוע הפקודה, OST תודיע לכם, שהיא צריכה זוג מפתחות על מנת לחתום קבצים שונים ולשנות דברים במערכת (תהליך זה יבטיח לכם שהקבצים ישארו ללא שינוי). כשלב ראשון, תתבקשו להכניס את המפתח הראשון שהוא - local key, ואת המפתח השני שהוא ה-site key בתהליך ההתקנה (הסבר על המפתחות תמצאו בהמשך).



רשימת קבצי מערכת ה-OST ותהליך הגיבוי שלה (מיקום הקבצים שאתן פה, הוא מקום הברירת מחדל שלהם, כל דבר ניתן לשינוי):

קובץ ההגדרות: /etc/tripwire/tw.cfg

קובץ ההגדרות מאחסן את כלל המידע הספציפי למערכת, לדוגמה: מיקום של קבצי המידע שלה. בנוסף, הקובץ מכיל את ההגדרות של התראת הדוא"ל. תצורת ההגדרות נוצרת תוך כדי תהליך ההתקנה תחת השם tw.cfg בתיקייה /etc/tripwire ובנוסף ישמר עותק טקסט באותה התיקייה בשם twcfg.txt. קובץ ההגדרות ניתן לשינוי על ידי שימוש בפקודה:

```
twadmin --create-cfgfile
```

הפקודה נותנת למשתמש את האפשרות להשתמש בקובץ הטקסט המוכן כקובץ ההגדרות הנוכחי ובאותו הזמן כמובן גם מצפינה אותו באמצעות שימוש בחתימה.

רכיבי קובץ ההגדרות:

מבנה קובץ ההגדרות בנוי כרשימה של זוגות של "מילת מפתח - ערך" וניתן גם להוסיף הערות ומשתנים להגדרות. כל שורה עם "#" בטור הראשון, מיוחסת כהערה.

דוגמה למבנה:

```
: ROOT = /usr/tripwire
```

החלפת משתנה בצד הימני מותרת בשימוש בתבנית הבאה לדוגמה:

```
: DBFILE = $(ROOT)/db/$(HOSTNAME).twd
```

בדוגמה, המשתנה הנוסף הוא: $(HOSTNAME). ישנם 2 משתנים שמוגדרים בקובץ ההגדרות ולא ניתן לשנות אותם: HOSTNAME ו-TIME, הראשון הוא שם המארח הלא רשמי ש-OST מורצת דרכו והשני הוא התאריך שהוא מחרוזת שמייצגת את התאריך ואת הזמן.

למערכת ישנם גם משתנים דרושים, שחייבים להיות מוגדרים על מנת שהמערכת תפעל.
הערכים המוגדרים בהם מושמים תוך כדי ההתקנה:

```
POLFILE Default = /etc/tripwire/tw.pol
DBFILE Default = /var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE Default = /var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE Default = /etc/tripwire/site.key
LOCALKEYFILE Default = /etc/tripwire/$(HOSTNAME)-local.key
```

קיימים עוד משתנים שלא נדרשים על מנת ש-Tripwire תרוץ, (דוגמה טובה תוכל להיות משתני התראת הדוא"ל) אבל חלק מהיעילות של התוכנה תיעלם בלעדיהם ולכן מומלץ ביתר חום לקרוא עליהם בלינק ל-manual שלה שמצורף בסוף המאמר.

קובץ המדיניות: /etc/tripwire/tw.pol

קובץ מדיניות מכיל סדרה של חוקים המגדירים את אובייקטי המערכת ש-Tripwire צריכה לנטר, והמידע של כל אובייקט שצריך להיאסף ולהישמר בקובץ בסיס-הנתונים. כל אובייקט בקובץ המדיניות מקושר עם מסיכת ערך שמגדירה עבור אילו שינויים בקובץ או בתיקייה Tripwire צריכה לנטר, ובאילו מהשינויים היא יכולה להתעלם. על ידי עריכת אספקטים שונים בקובץ המדיניות, מנהל המערכת יוכל להיות בעל שליטה מלאה על איך Tripwire בודקת את אמינות המערכת.

קובץ המדיניות (tw.pol) מוגדר בהתקנה כקובץ מוצפן וחתום תחת תיקיית /etc/tripwire, בנוסף ישמר באותה התיקייה גם קובץ בשם: twcfg.txt ועוד קובץ טקסט בשם: policyguide.txt שמדגים בתוכו את כל המאפיינים של קובץ המדיניות (אותם קבצי הטקסט העמוסים בהערות ותיאורים, מומלץ להשתמש בהם כמדריכי עזרה).

קובץ מדיניות חדש נוצר בעזרת שימוש בפקודה:

```
twadmin --create-polfile
```

הפקודה מאפשרת למשתמש להשתמש בקובץ הטקסט המוכן כקובץ המדיניות הנוכחי. באמצעות שימוש בזוג המפתחות שנקבע, קובץ ההגדרות החדש מוצפן, חתום ושמור.
כאשר קובץ המדיניות ההתחלתי נוצר נוכל לבצע בו כל שינוי שנרצה באמצעות הפקודה:

```
tripwire --update-policy
```

חשוב לזכור: כאשר קובץ מדיניות חדש נוצר, קובץ בסיס הנתונים של Tripwire יהיה חייב להיות מאותחל מחדש. אם פורץ יעשה שינוי בקבצים מאז בדיקת האמינות האחרונה , השינויים לא יאותרו ויכללו כחלק מהבסיס של קובץ בסיס הנתונים החדש.

הרכיבים בקובץ הם הערות, חוקים, הנחיות ומשתנים. כל חלק מרכיבים אלו מתואר בהרחבה בקובץ ה-manual של התכנה שמצורף בסוף המאמר.

קובץ בסיס הנתונים: /var/lib/$(HOSTNAME).twd

קובץ בסיס הנתונים משמש כבסיס לבדיקות האמינות של המערכת. מיד אחרי ההתקנה, Tripwire יוצרת את בסיס נתונים הראשוני (בהתאם למיקום שניתן למשתנה DBFILE) שהוזכר קודם לכן. אותו הקובץ הוא בעצם תמונת מצב של מערכת הקבצים במצב תקין. כאשר יהיה צורך בבדיקת אמינות קבצים, Tripwire תשווה כל אובייקט במערכת, כפי שמתואר בקובץ המדיניות, כנגד ערך מקביל בבסיס הנתונים. יווצר דוח, ואם אובייקט מסויים השתנה מחוץ לכללים שהוגדרו בקובץ המדיניות, ההפרה מדווחת בדוח .

קבצי הדיווח: /var/lib/tripwire/report/$(HOSTNAME)-$(DATE).tw

לאחר ששלושת הקבצים שהזכרנו (קובץ הגדרות, קובץ המדיניות וקובץ בסיס הנתונים) נוצרו בהצלחה, Tripwire תוכל להריץ בדיקת אמינות ולחפש הבדלים בין המצב הנוכחי של המערכת, לבין המידע השמור בקובץ הנתונים הראשוני שנוצר . התוצאה שיוצאת הופכת להיות דוחות , אותם דוחות מהווים אוסף של "הפרות מדיניות" שנתגלו תוך כדי בדיקות האמינות.
כאשר משתמשים בהגדרות המתאימות, הדוחות יכולים להישלח בדוא"ל.

מצב הבדיקה מאפשר לבדוק את מערך ההתראה בדוא"ל של Tripwire. כשמפעילים את האפשרות הזו, תשתמש Tripwire בהגדרות ההתראה בדוא"ל שמפורטות בתוך קובץ ההגדרות ע"מ לשלוח את ההודעה (על מנת לברר עוד על המשתנים האפשריים כדי לשלוח מייל , מומלץ לקרוא ב-manual על מנת לקבל יותר מידע).

אגב, על מנת להשתמש במצב הבדיקה, יש להשתמש בפקודה:

```
-m t, --test
```

מה שהפקודה עושה הוא לאפשר את בחירת המצב TEST ולכן מכאן אפשר להשתמש במשתנה e שמסמל email (user@domain.com), משתנה זה מאפשר להשתמש בכתובת מייל ספציפית וחייב להינתן כאשר משתמשים במצב הבדיקה (חשוב לציין כי רק כתובת אחת מותרת פה).

קבצי המפתחות: /etc/tripwire/site.key ו-/etc/tripwire/$(HOSTNAME)-local.key

אחד הדברים החשובים הוא שקבצי המערכת של Tripwire יהיו מוגנים מפני משתמשים בעלי גישה לא מאושרת (דוגמה לכך תוכל להיות פורץ, אם תהיה לו גישה לקבצי המערכת הוא יוכל להשבית את כולה). בדיוק מסיבה זו כל הקבצים שתיארתי קודם לכן חתומים על ידי מפתח מוצפן, בכדי למנוע שינויים על ידי גישה לא מאושרת. ישנם שני מפתחות מופרדים שנועדו להגן על קבצי המידע הקריטיים של המערכת , אחד או הזוג כולו, נחוצים על מנת לבצע כמעט כל פעולה במערכת Tripwire.

מפתח ה-Site Key נחוץ לנו על מנת להגן על קבצים שאפשר להשתמש בהם גם על גבי מערכות שונות (דוגמה: קבצים הגדרות והמדיניות).

מפתח ה-Local Key נחוץ לנו בכדי להגן על קבצים שנועדו למערכת הנוכחית (כמו קובץ הנתונים). באותו מפתח אפשר להשתמש גם על מנת לחתום על דוחות של בדיקת אמינות

תהליך גיבוי הקבצים:

על מנת למנוע טעויות של מחיקת מידע , Tripwire יוצרת באופן אוטומטי קבצי גיבוי בכל פעם שקובץ Tripwire נכתב מחדש. אותו קובץ "ישן" יוסיף לשמו סיומת bak. והגירסא החדשה של הקובץ תיקח את מקומו. דבר חשוב שיש להזכיר הוא כי קיימת אפשרות לגבות רק עותק אחד על כל שם קובץ. אם עותק של הקובץ כבר קיים , הגיבוי הישן ימחק ויוחלף בחדש יותר . גיבוי קבצים הינו חלק אינגטרלי ממערכת Tripwire ולא ניתן להסירו או לשנות את אופן פעולתו

twadmin - כלי הניהול של OST.

מעבר לכל מה שרשמתי מקודם ישנו כלי מיוחד לניהול המערכת אותו הכלי נקרא twadmin והוא מאפשר לבצע פעולות שקשורות בניהול המערכת הכלי מספק למנהל המערכת את האפשרויות הבאות:

יצירת קובץ הגדרות:

```
--create-cfgfile
```

הדפסת קובץ ההגדרות:

```
--print-cfgfile
```

החלפת קובץ מדיניות:

```
--create-polfile
```

הדפסת קובץ המדיניות:

```
--print-polfile
```

הסרת הצפנה מקובץ:

```
--remove-encryption
```

הצפנת קובץ:

```
--encrypt
```

בחינת מצב החתימה של קובץ:

```
--examine
```

ייצור מפתחות:

```
--generate-keys
```

סיכום

בימים אלה, בהם סכנות אורבות לנו ולמידע שלנו מכל עבר ומאימות לפלוש למקום הפרטי ביותר שלנו- המחשב שלנו, שמאחסן את המידע הרג יש ביותר, עלינו להיות מודעים לאפשרויות שיש בידינו על מנת להגן עליו. כל עוד המחשב יהיה מחובר לאינטרנט לא יהיה ניתן להגן על המידע שלנו בכל מאת האחוזים, תמיד יהיו פרצות אבטחה שיהיו נגישות לפורצים ויאפשרו להם להזיק למידע שלנו.

עם זאת, כשכל יתר המגננות שיש ביד ינו, בכדי להגן על המידע שלנו, נופלות ונכשלות במטרתן, עומדת לרשותנו מערכת ה-IDS שמטרתה אינה למנוע את הפריצה למערכת אלא לתת לנו פיתרון כשכבת הגנה נוספת ולזהות מתקפות שעקפו את שאר מעגלי ההגנה.

מערכות ה-IDS, שהחלו להיות נפוצות בזמן האחרון, מערכות ההתראה בשילוב מערכות ומודולים מסוגים שונים, יכולות להעלות את רמת האבטחה שלנו לרמה גבוהה ביותר ולתת לנו הגנה מיטבית למידע שלנו. עלינו להעלות את המודעות שלנו לגבי האפשרויות העומדות בידינו להגנה מירבית למידע שלנו ולנצל ידע זה בתבונה.

על הכותב

נתנאל שיין עוסק בפיתוח ובאבטחת מידע בפרט, מעורב בפרוייקטים שונים בנושא הקוד הפתוח בעיקר בהתנדבות, חבר בעמותת המקור, כיום עובד בהייטק וסטודנט למדעי המחשב באוניברסיטה הפתוחה.

קישורים חיצוניים:

http://idstutorial.com
http://netshine.wordpress.com
http://www.la-samhna.de
http://sourceforge.net/apps/wordpress/tripwire
http://www.ynet.co.il/articles/0,7340,L-3856254,00.html
http://linux.die.net/man/4/twconfig
http://linux.die.net/man/4/twpolicy
http://linux.die.net/man/8/twadmin