דו"ח מעבדה- תרחיש מס' 2

# פרטים:

מגיש: שגיא סעדה

תאריך: 25/03/2018

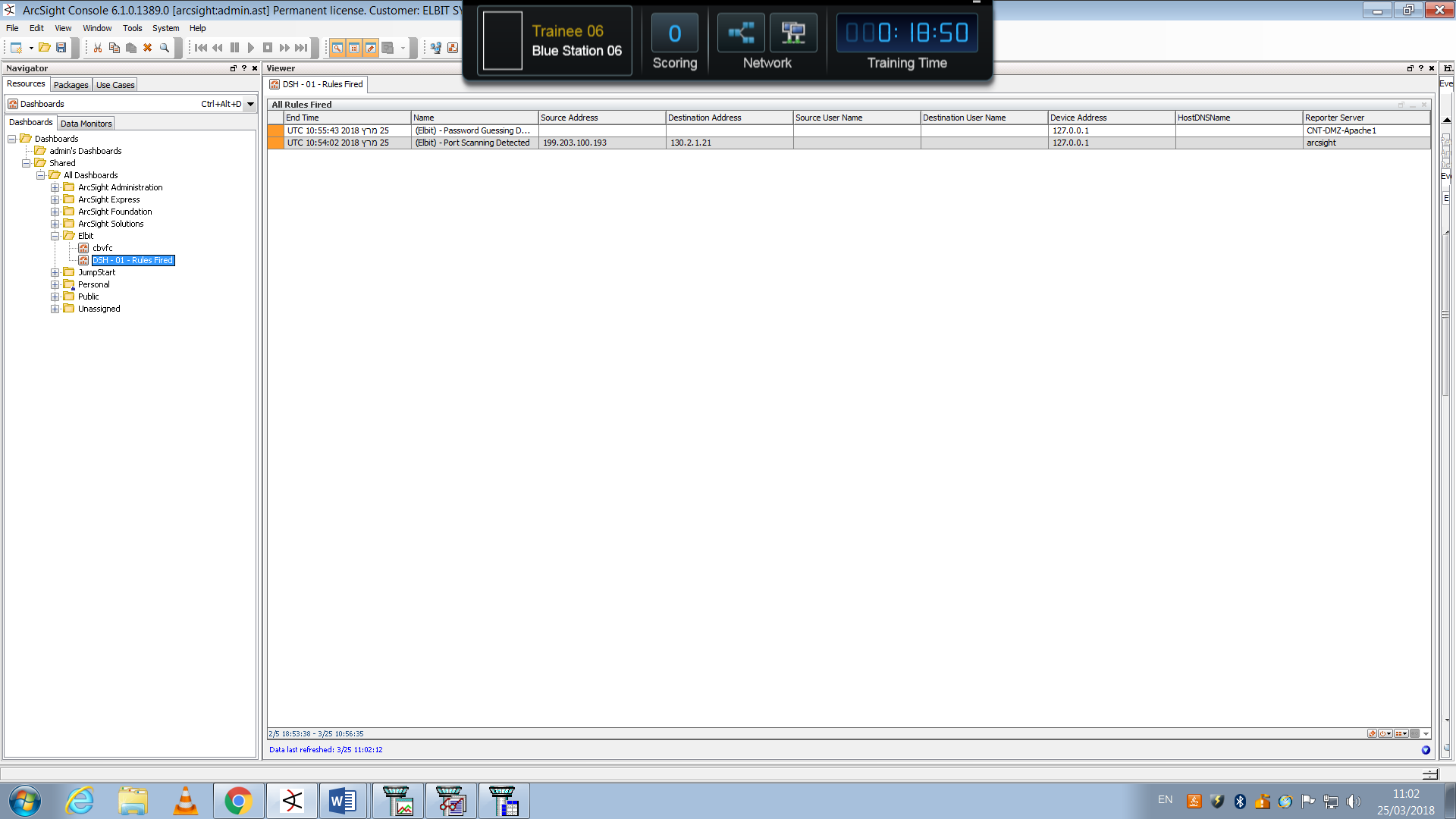
שם התרחיש: הפלת אתר הארגון באמצעות פריצה למתזמן המשימות של לינוקס והרצת סקריפט השולח מידע לתוקף.

# תהליך ההתקפה:

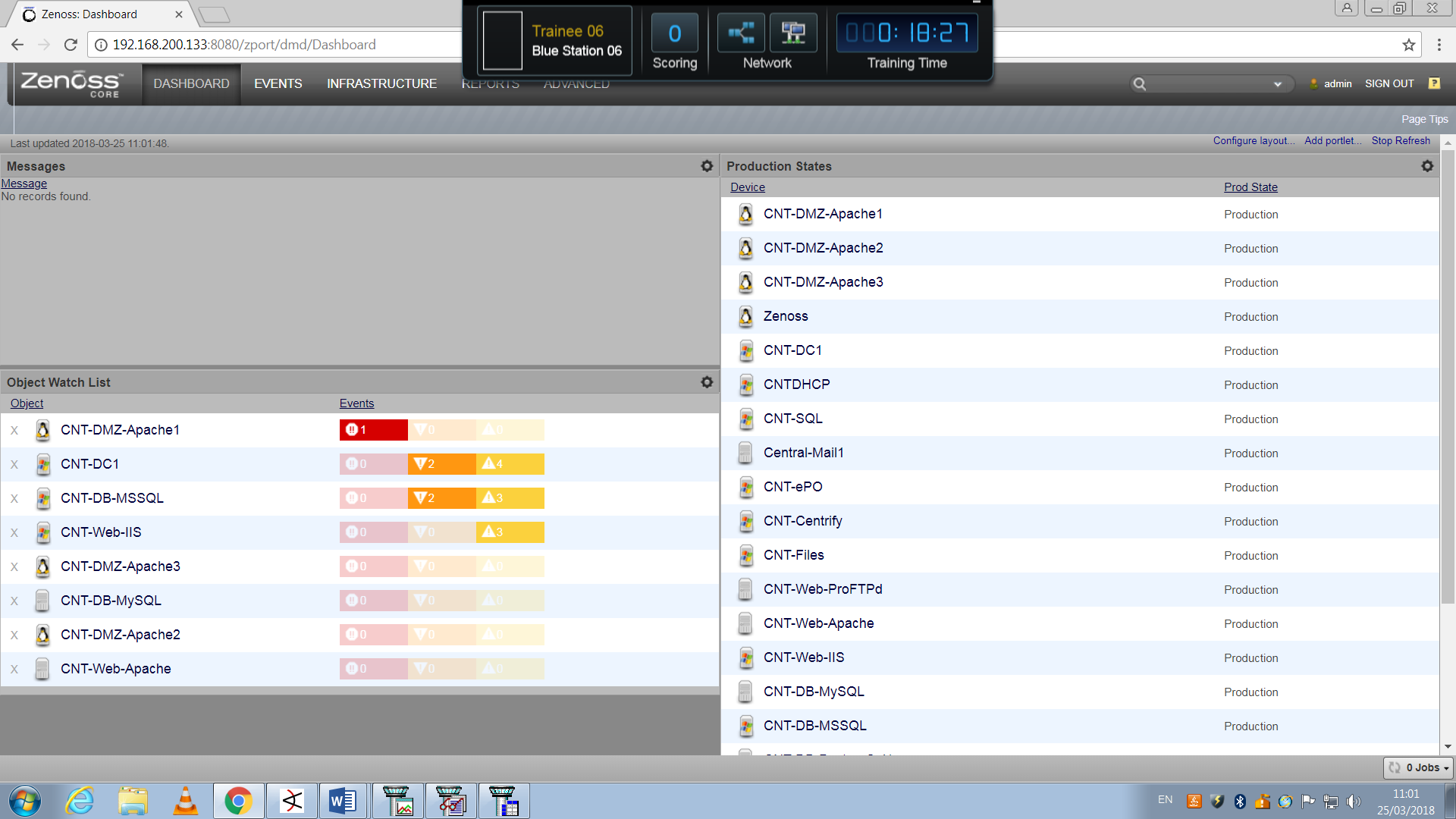
בתהליך ההתקפה זוהה על ידי הכלי ARCSIGHT – Portscanning על ידי האייפי – 199.203.100.193.

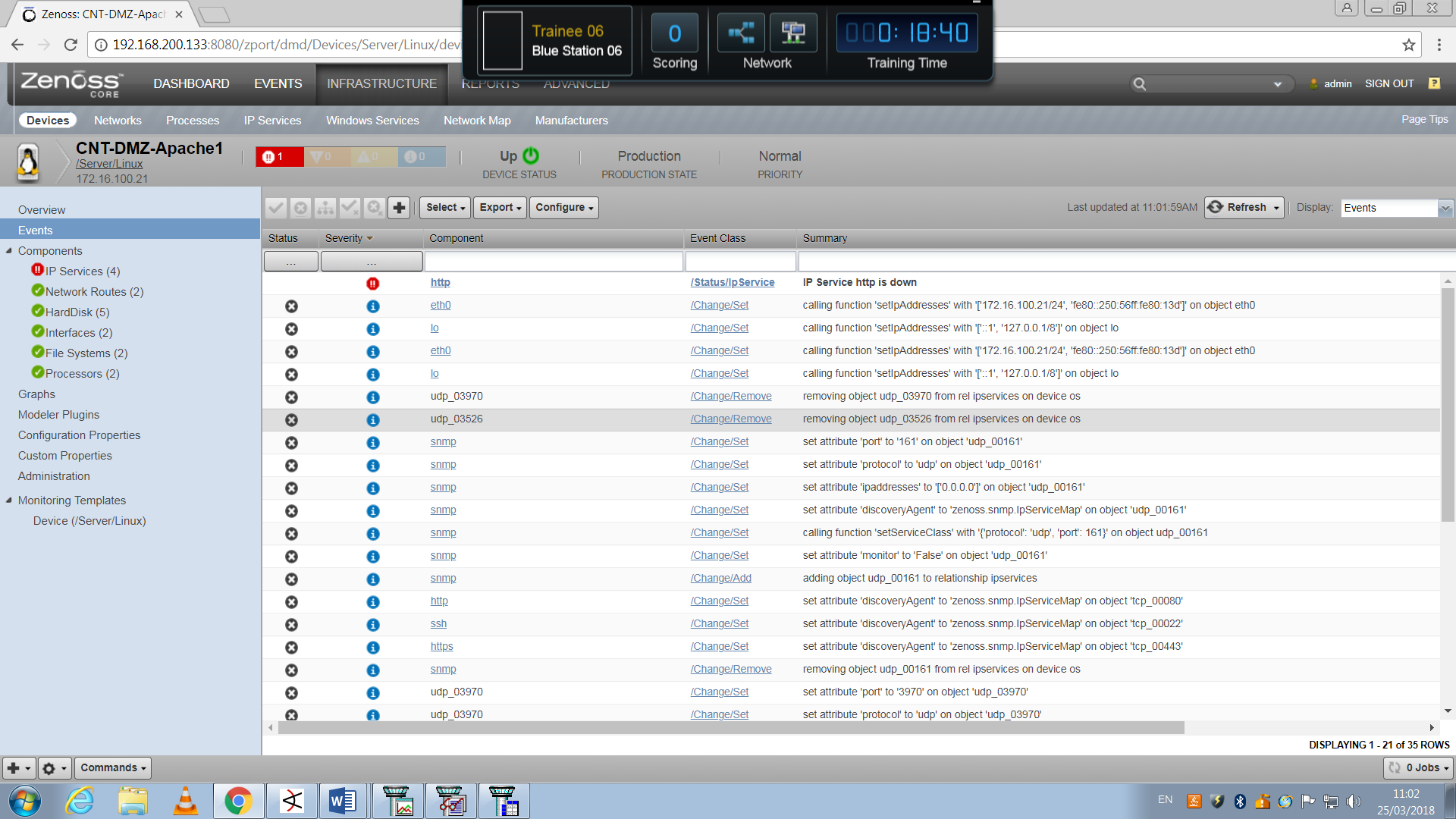
סריקת הפורטים התבצעה על אייפי בשרת ה- dmz – 130.2.1.21 – מקבלי ב- subnet לאייפי 172.16.100.21 – שרת זה אחראי על אתר foxnews.

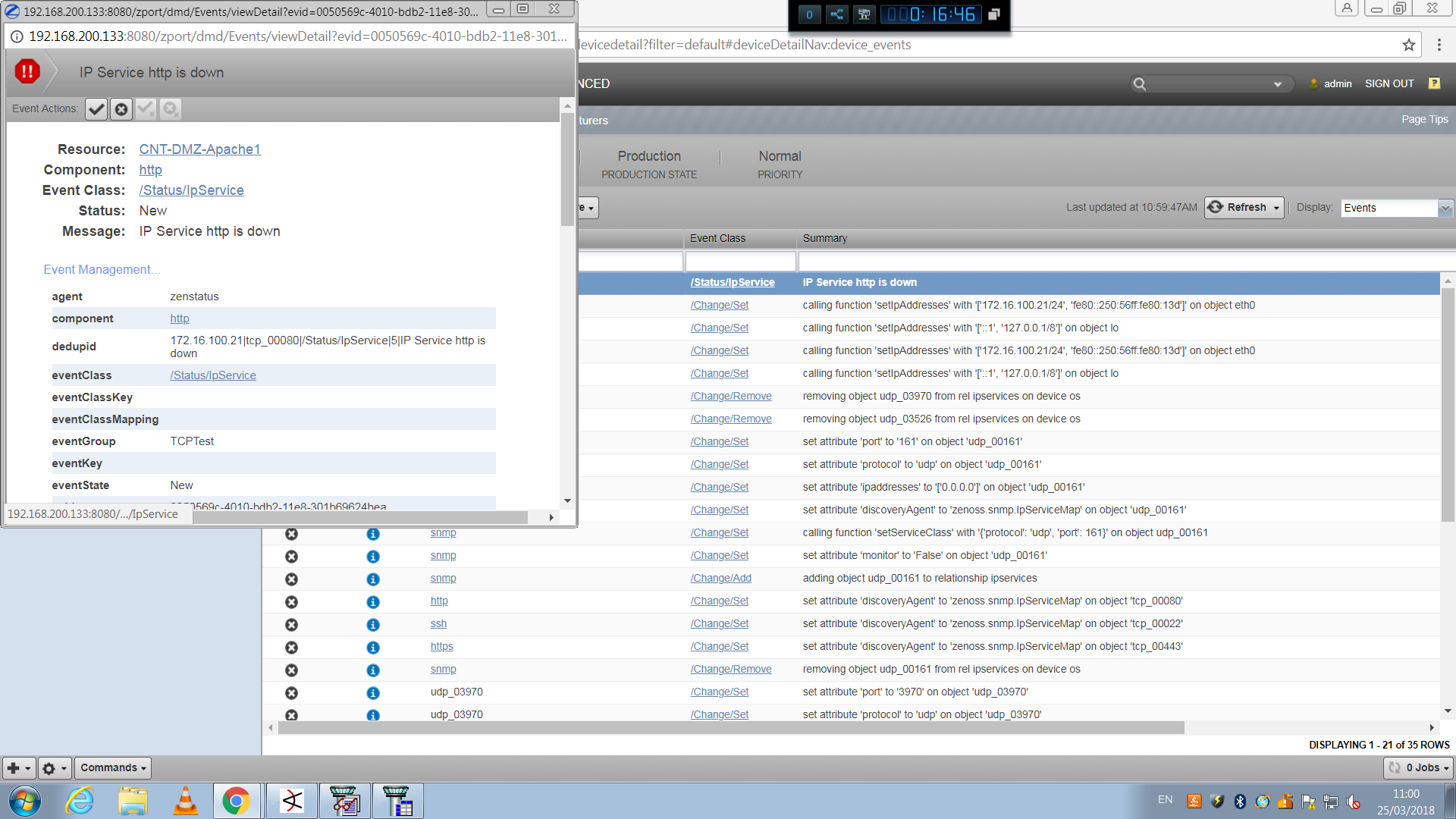
לאחר מכן (בשעה 10:55) ראינו Password Guessing כאשר השרת המדווח הוא – CNT-DMZ-APACHE1.

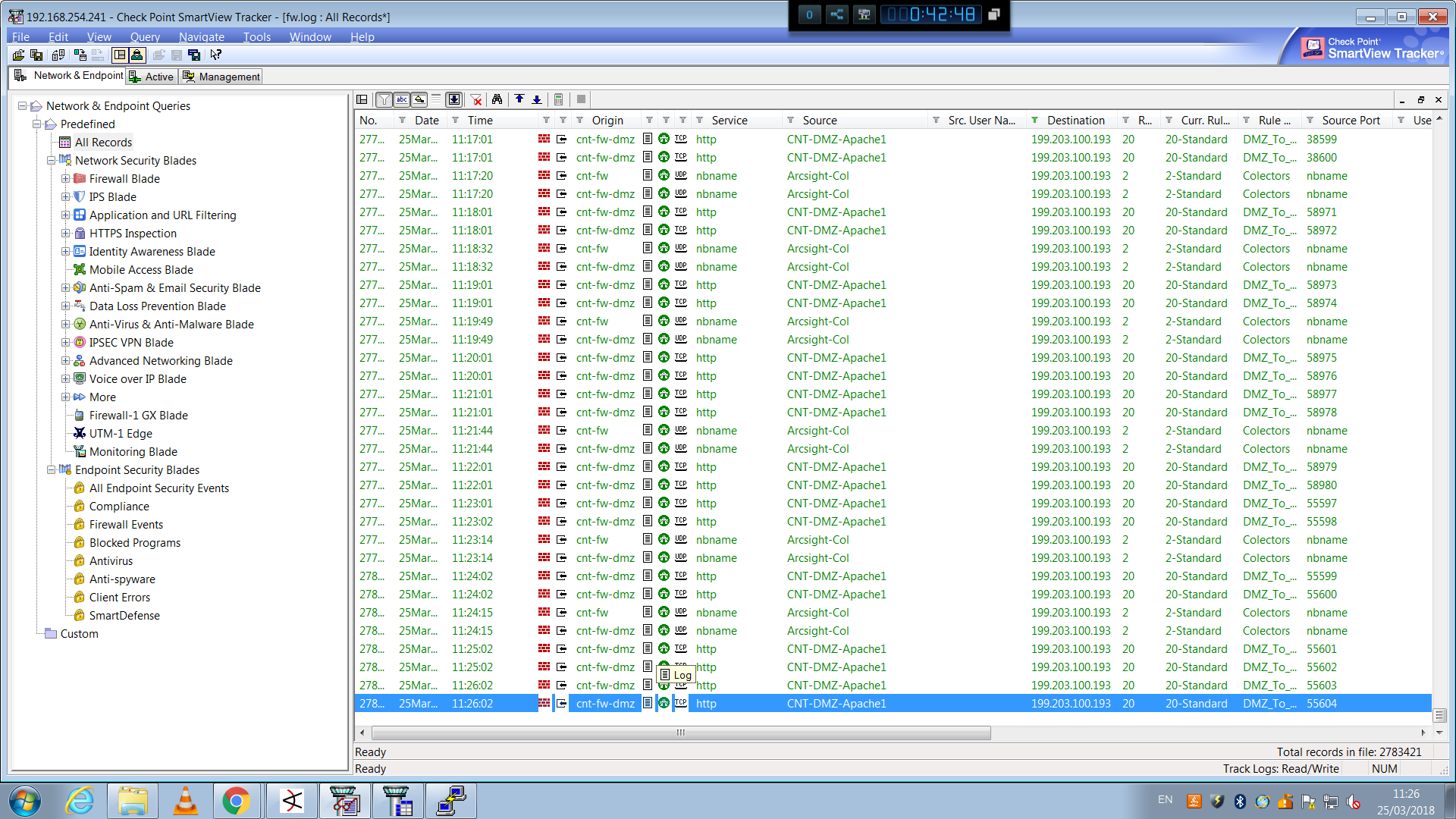


לאחר מכן (בשעה 10:59) ראינו התראה באתר ה- ZENOSS שלנו, שהאתר בשרת המסוים שאליו היו הפניות – נפל. (כלומר, האתר למטה ולא זמין להצגה בדפדפן).





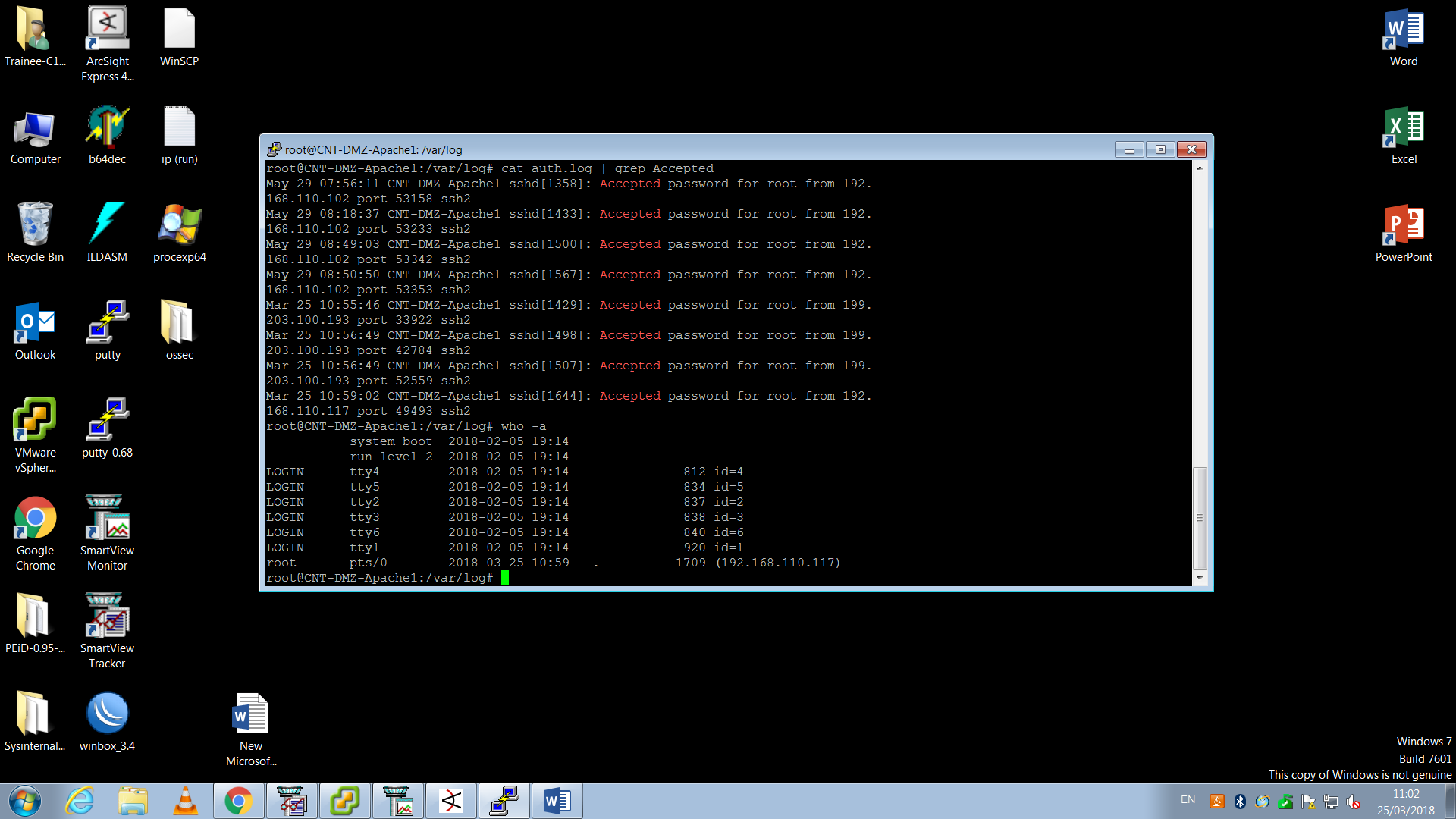


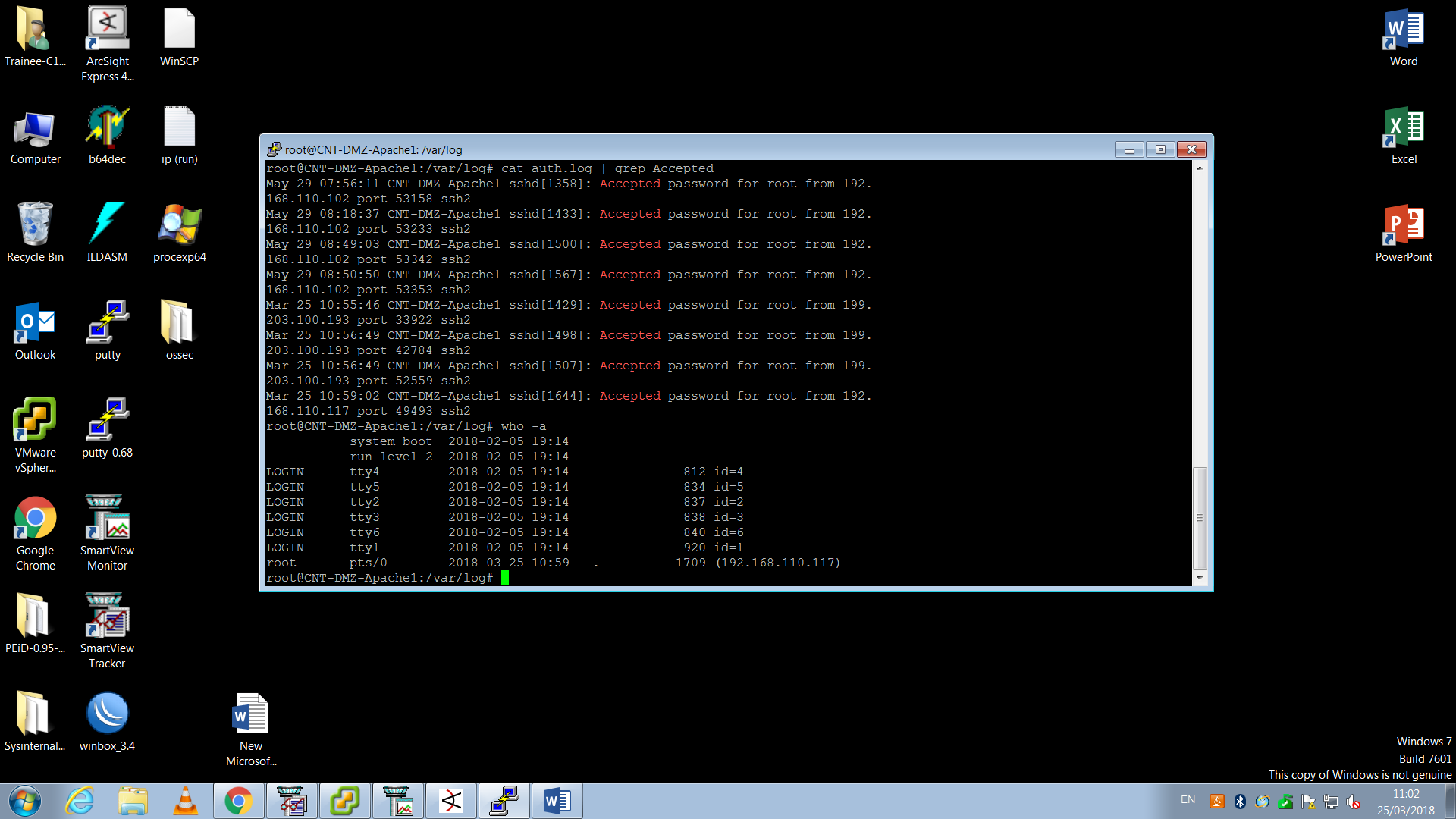


# תהליך הזיהוי:

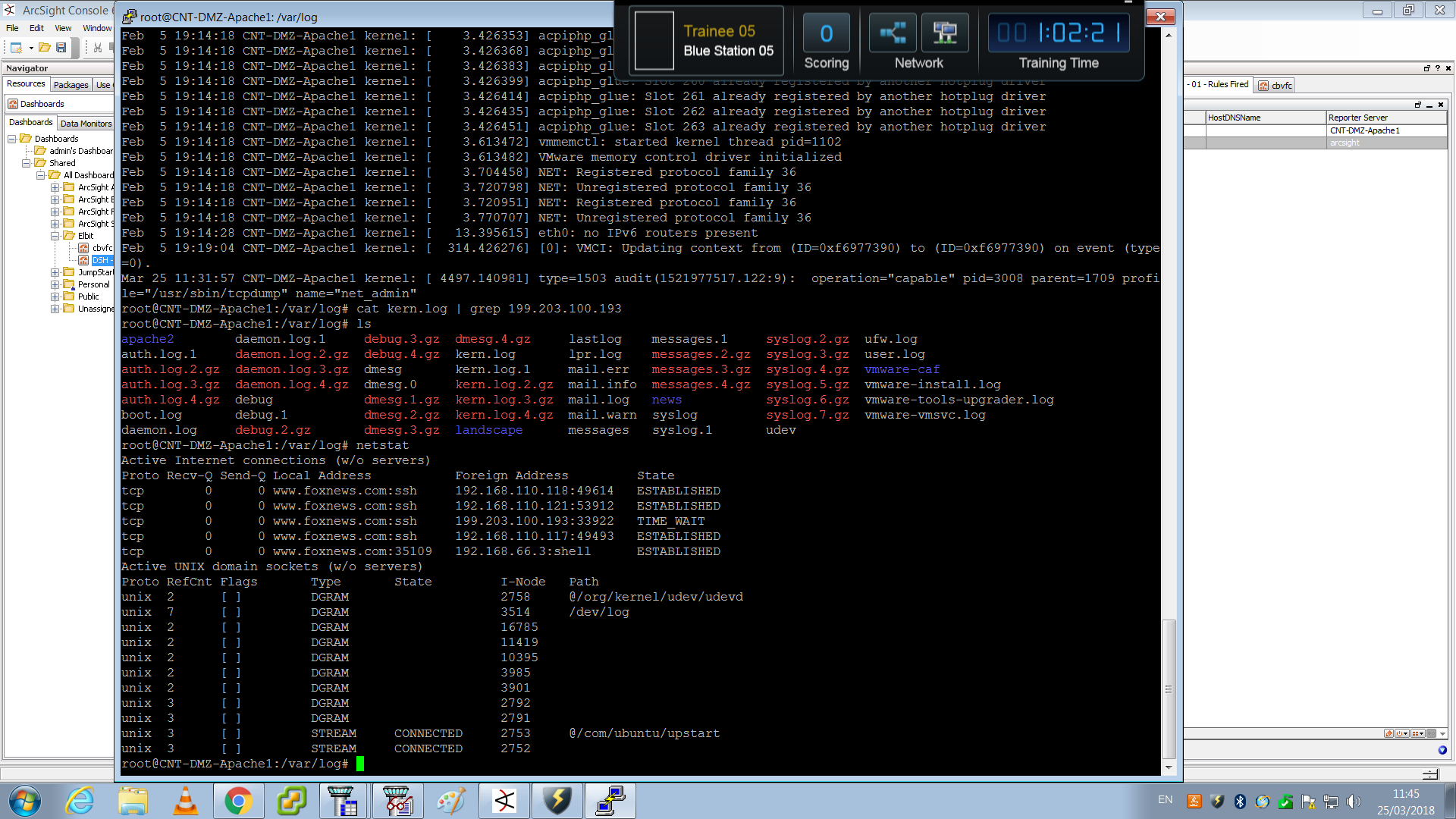
בתהליך הזיהוי ננקטו הצעדים הבאים על מנת להעמיק את הבדיקה של האירוע –

1 – התחברנו לשרת הנתקף באמצעות Putty על מנת לבדוק את הלוגים שלו, לראות התחברויות באמצעות netstat





לאחר מכן שמנו לב שהקובץ auth.log נמחק ואינו מצא בתיקייה var/log.



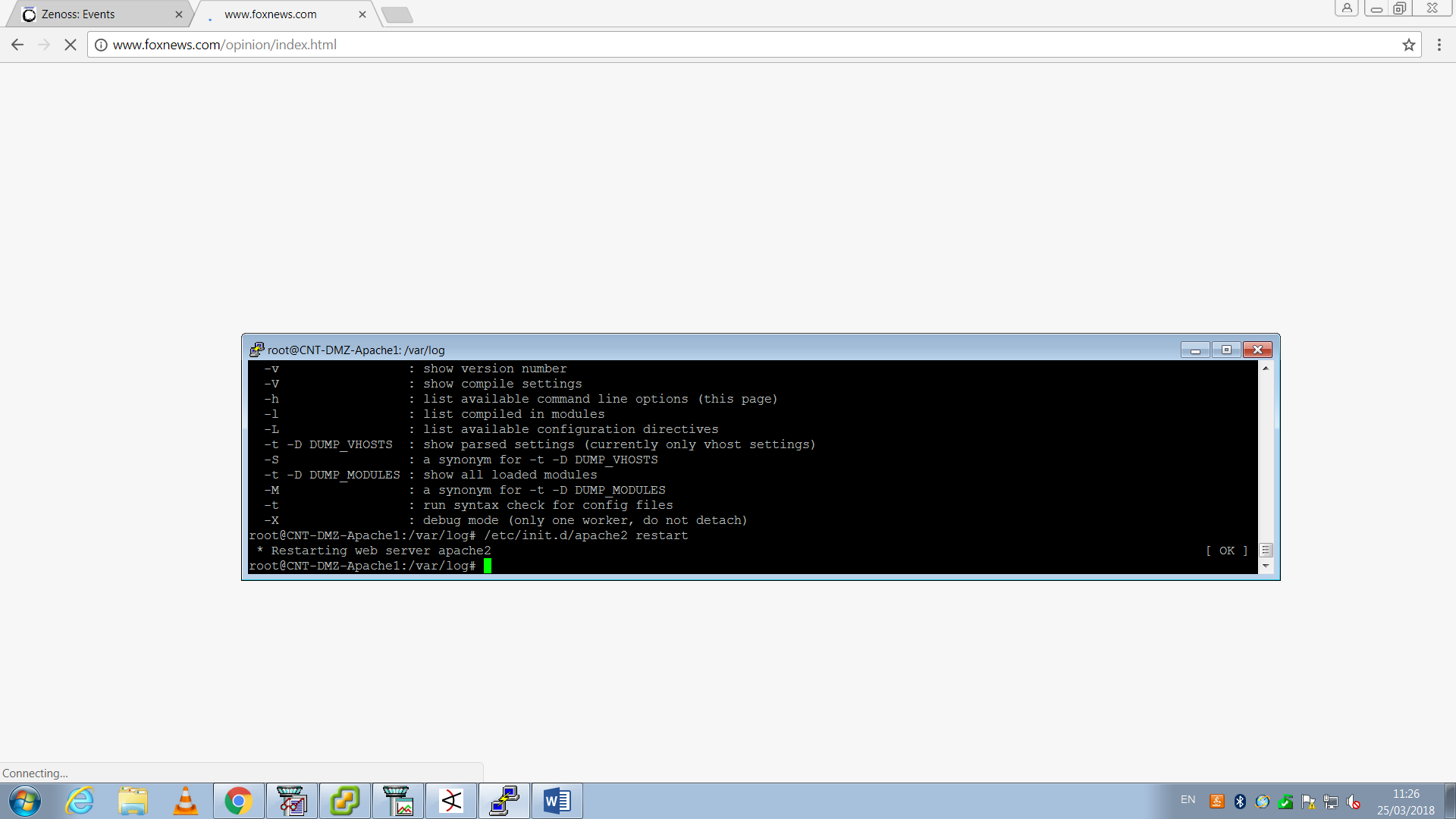
ניתן לראות בבירור כניסה של התוקף לשרת באמצעות שורת ה- Accepted.

ובעזרת netstat רואים חיבור של התוקף גם כן. מסקנות – בשלב זה כבר היינו חוסמים את ה-IP של התוקף.

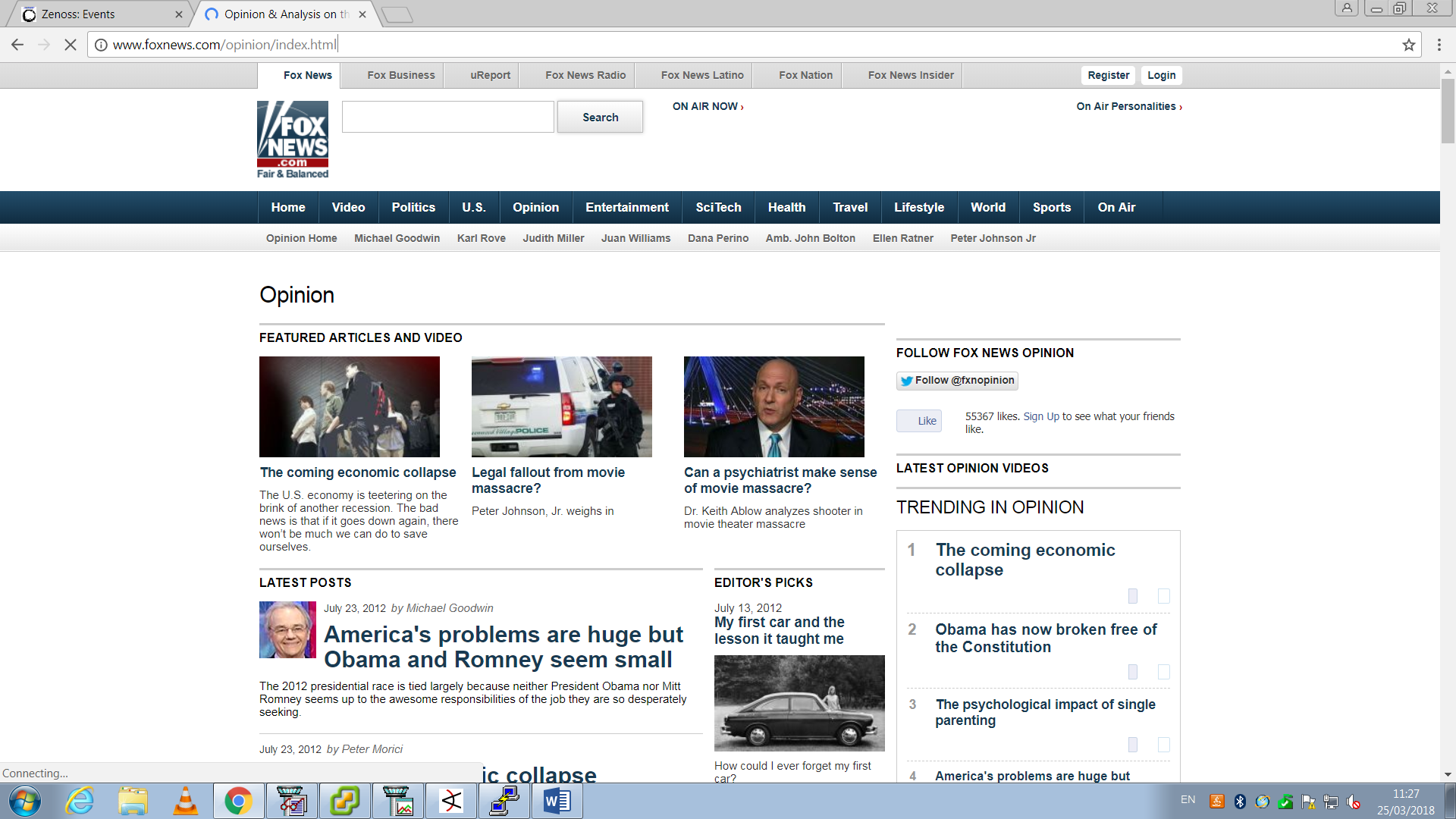
בעיקרון, מתרחיש קודם כבר היינו מבצעים חסימה לסריקת פורטים מ-IPים חיצוניים לחברה לפחות (פנימיים צריך לבדוק איך לעשות זאת בתבונה) וגם אפשרות כניסה מ- IPים חיצוניים אלה אם כן הם שמורים במערכת וכך יתבצע פילטור.

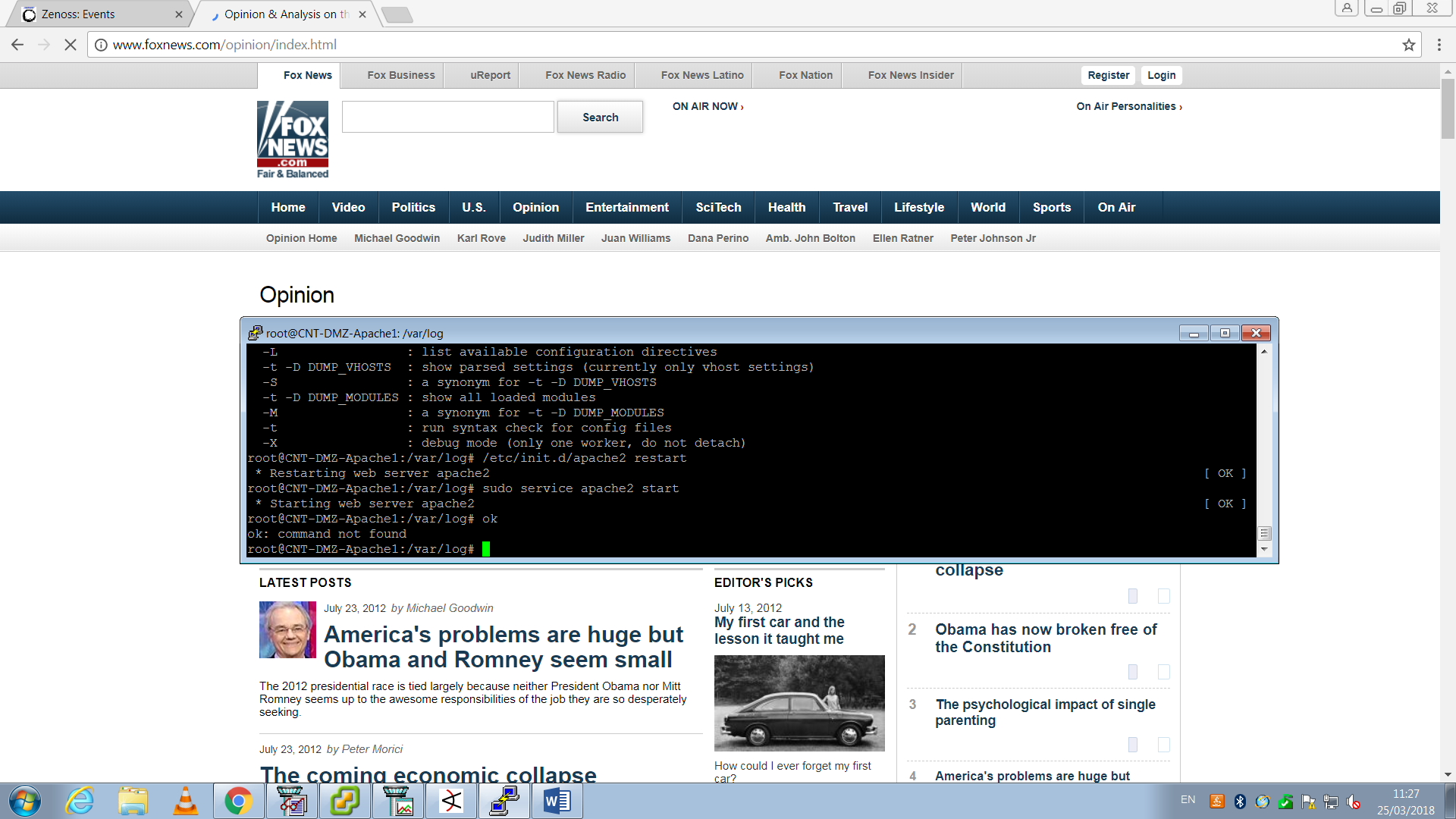
כעת, אנו מנסים להחזיר את האתר לפעילות מלאה ולכן השתמשנו בפקודה –

service apache2 restart או /etc/init.d/apache2 restart

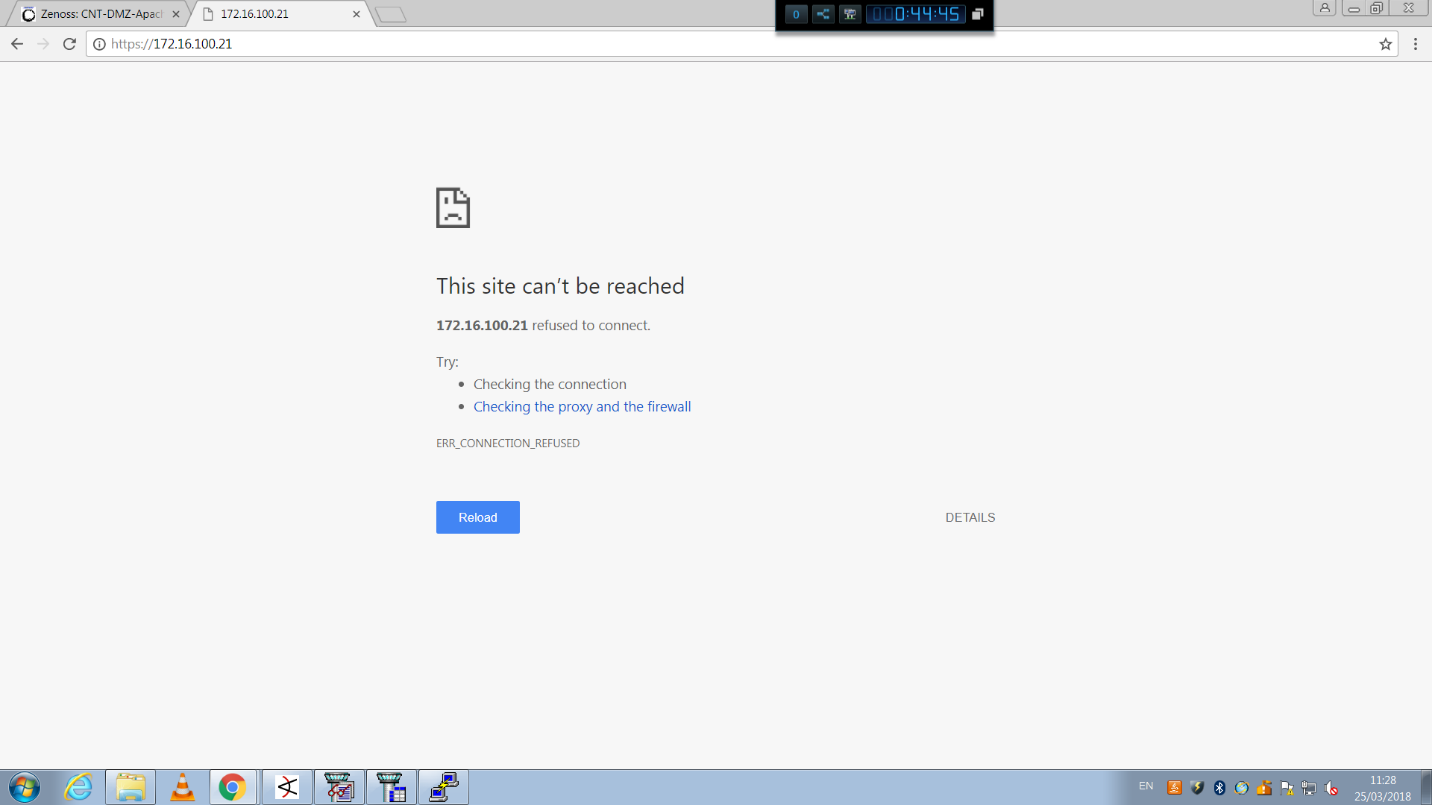


ובדקנו את האתר – ואכן הוא חזר לפעילות רגילה





לאחר כמה שניות האתר נפל שוב.



בשלב זה הנחנו שקיים תהליך אוטומטי או שהתוקף עדיין נמצא בשרת ומפיל את האתר באופן קבוע.

ראינו פעולה קבועה שמתבצעת כל דקה אחת בדיוק דרך ה- Zenoss – נשלחת הודעה לאייפי של התוקף מהשרת שלנו (מהשרת הנתקף).

ולכן נובע מכאן שהפעולה היא אוטומטית – זאת אומרת, קיים תהליך (PROCESS) או סקריפט שהוכנס בזדון לשרת שלנו או שהתוקף נמצא עדיין בשרת וכל דקה –

1 – שולח הודעה לתוקף מהשרת המותקף.

2 – מפיל את האתר שלנו.

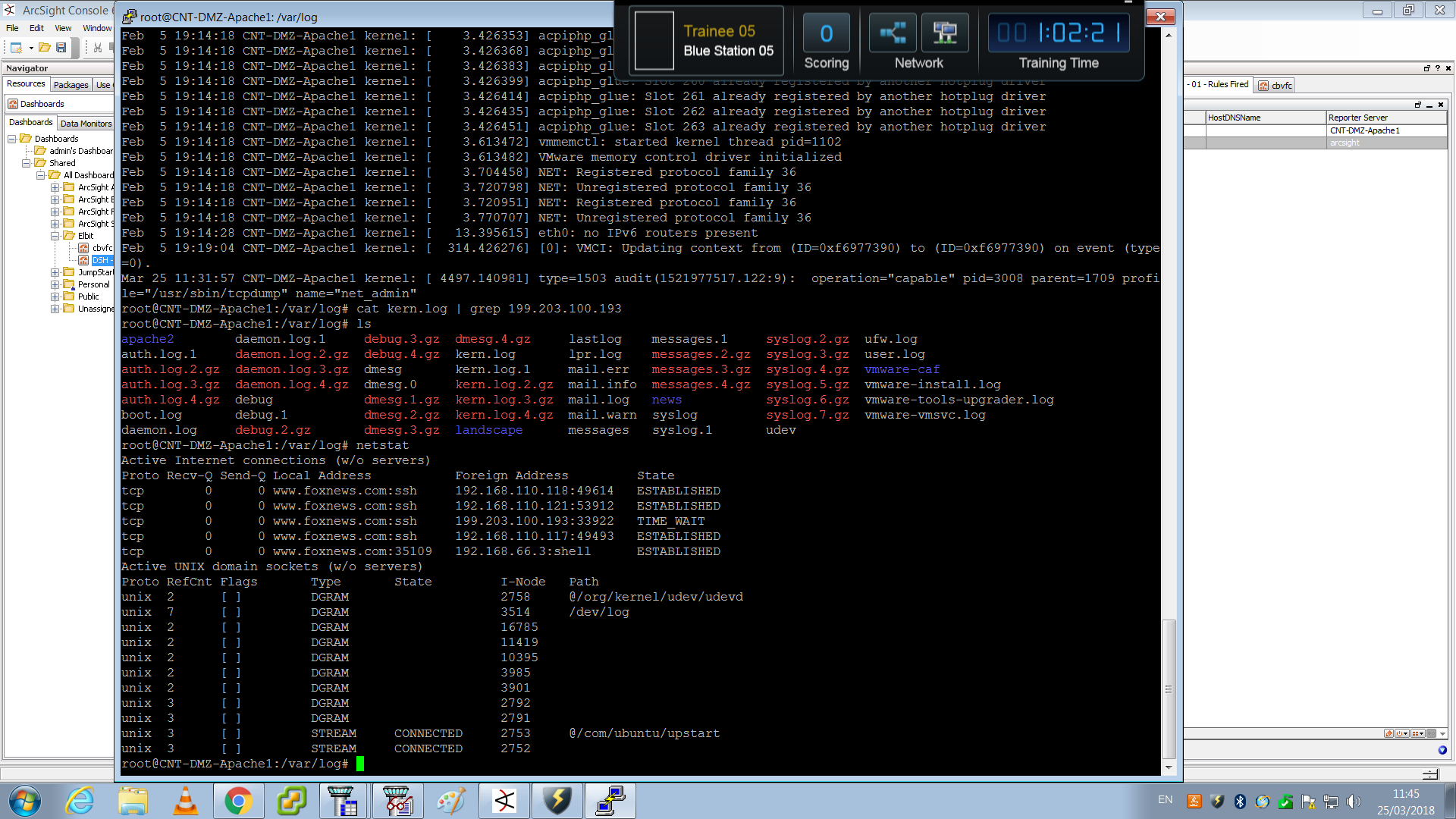
(בתהליך ההגנה אציג איך "הוצאנו" את התוקף מהשרת ועדיין נראה את התהליך האוטומטי).

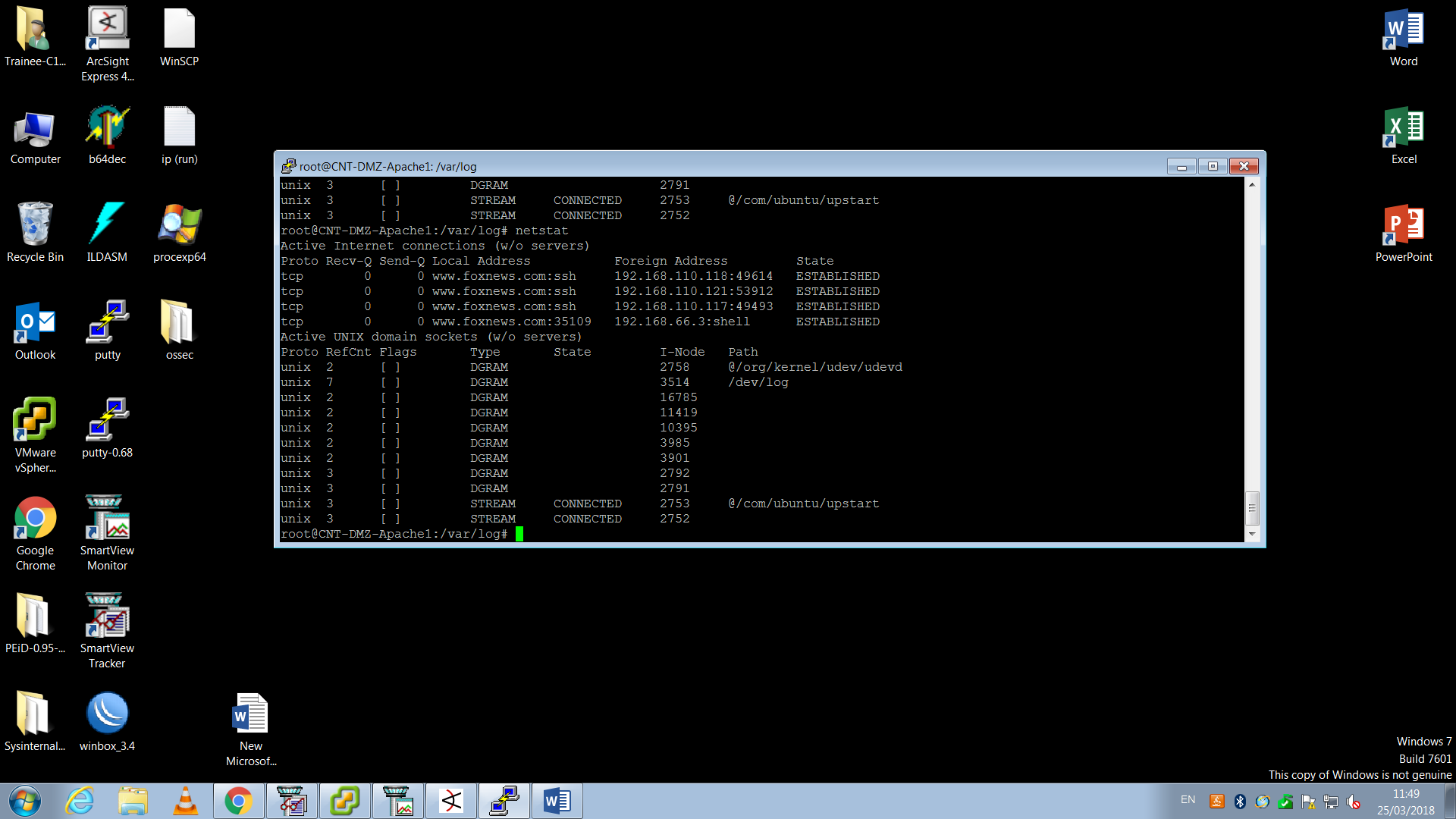
# תהליך הגנה:

בתהליך ההגנה לאחר שבדקנו מי מחובר לשרת ב – sshd – ראינו 4 חיבורים כאשר רק 3 מחברי הצוות מחוברים.

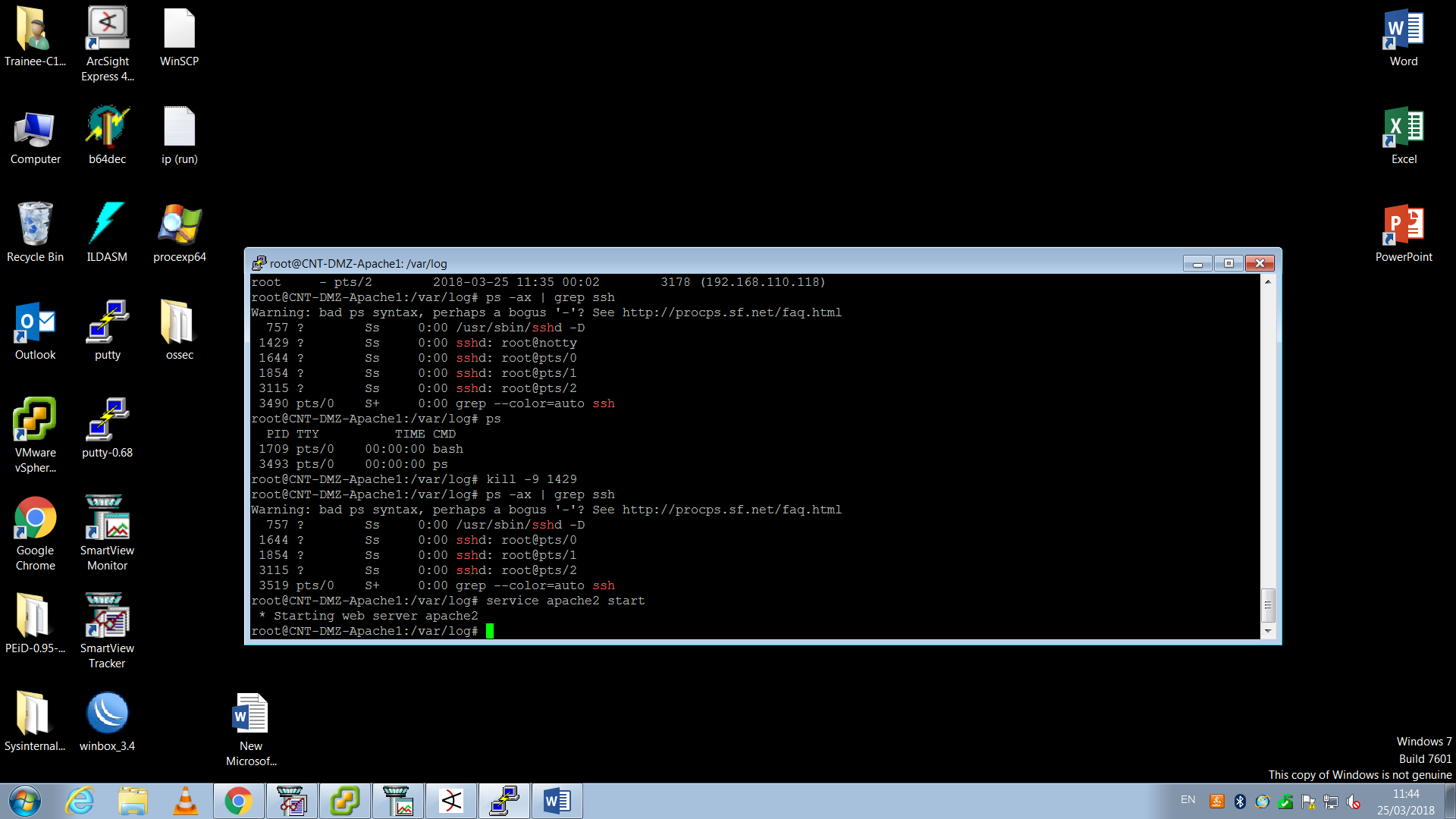
חיבור אחד היה יוצא דופן שהשם של היוזר היה root/notty. בעזרת הפקודה kill -9 PID "הרגנו" את החיבור הזה.

לאחר מכן – הרצנו את הפקודה netstat וראינו שסטטוס החיבור עבור האייפי של התוקף שונה ל- TIME\_WAIT ולאחר כמה דקות נעלם לגמרי.

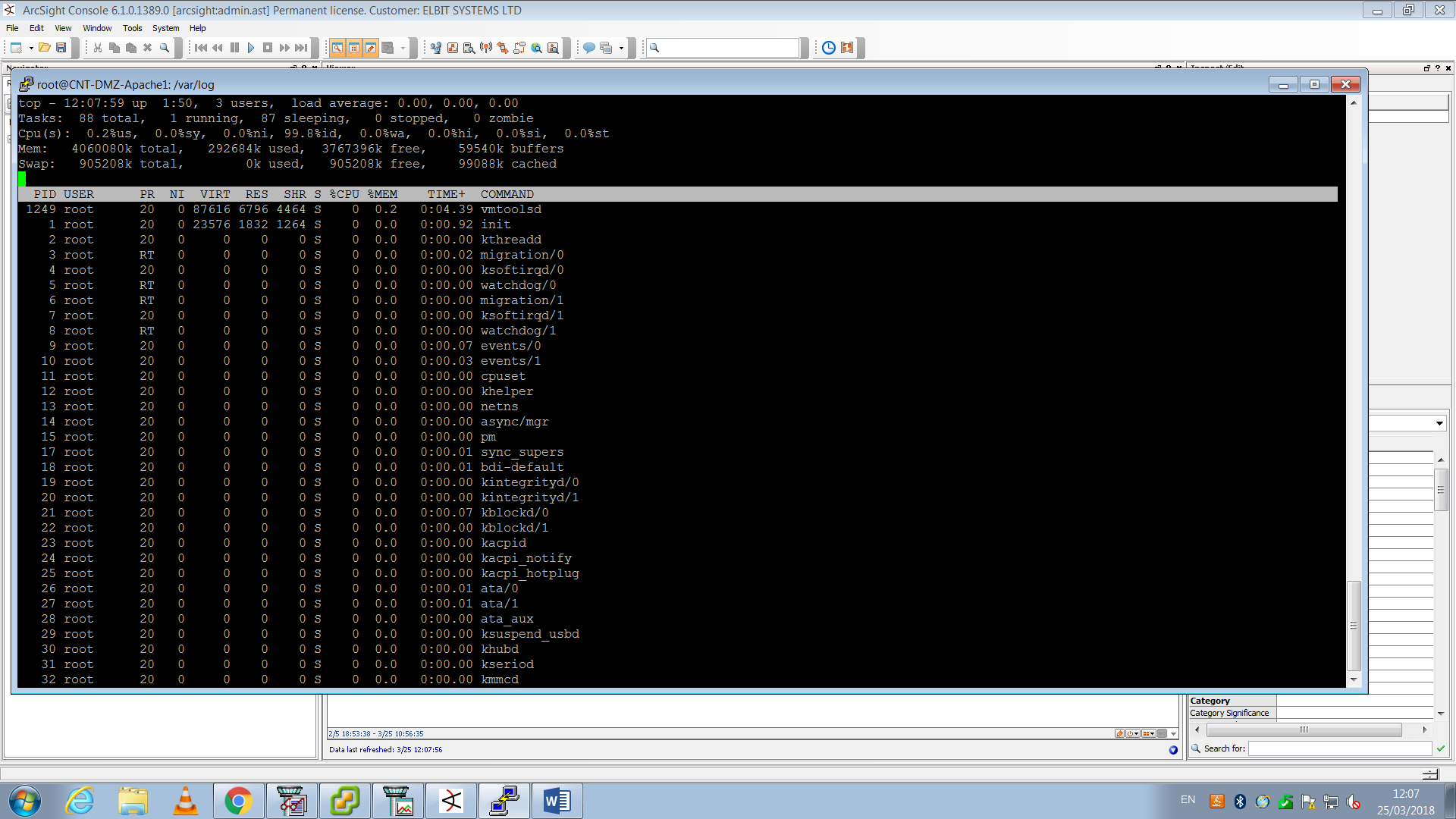




העלינו את האתר מחדש, אך שוב הוא נפל.



כעת, המשכנו לחקור את כיוון התהליך (PROCESS) או סקריפט שאולי פועל ברקע ובאופן אוטומטי כל דקה מבצע את הפעולות שצייתי קודם.

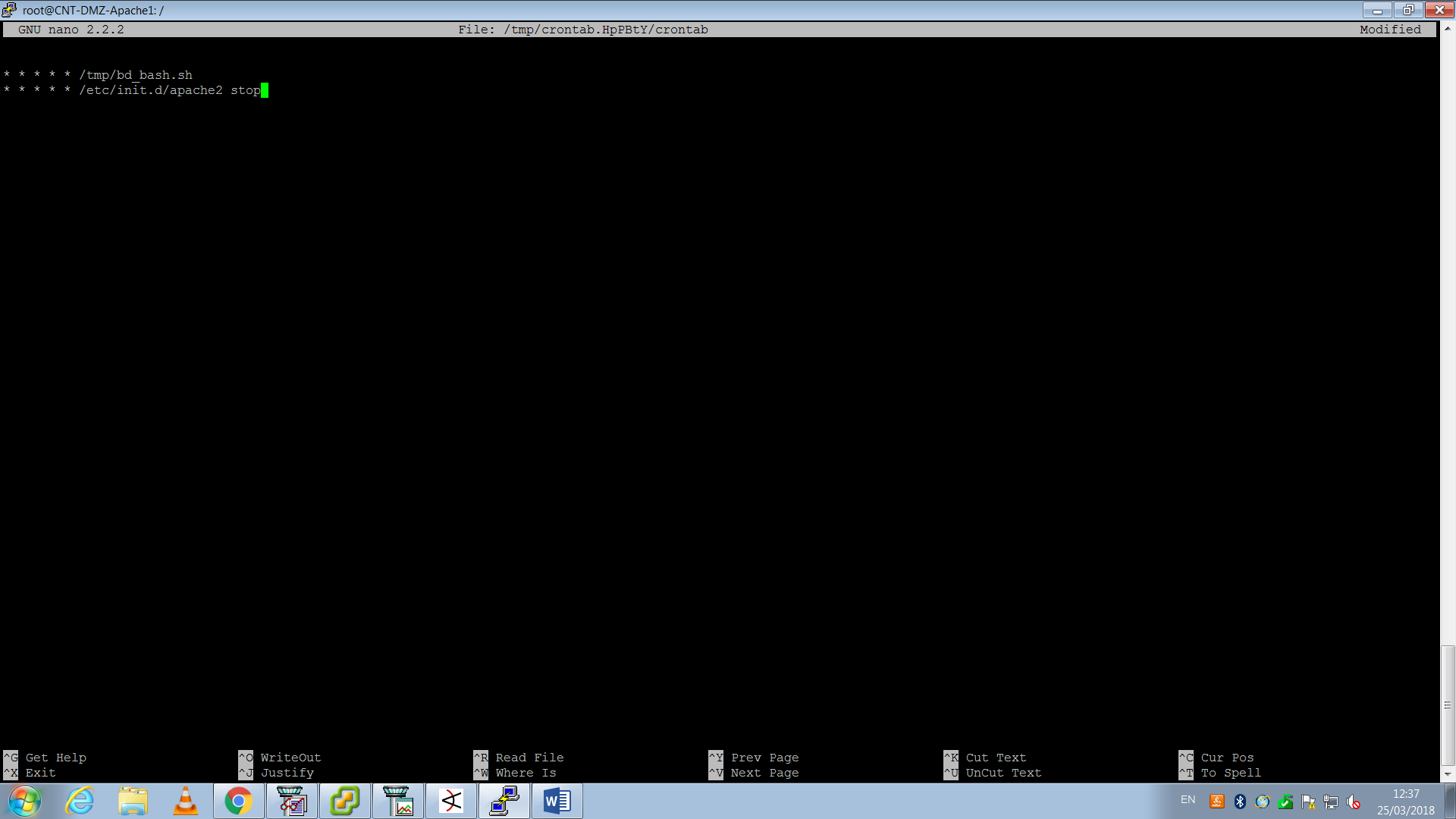


לאחר זמן ממושך התחלנו לחשוב ולנסות לפענח האם יש דרך כלשהי בה מורץ תהליך או סקריפט כלשהו אשר לא מופיע ברשימת התהליכים, זאת אומרת – מה יכול להפעיל את הפעולה הזדונית שרצה כל דקה.

התחלנו להקביל את מערכת לינוקס למערכת ווינדוס והסקנו שיש מערכת שנקראת "מתזמן משימות" – בלינוקס נקרא crontab.

בעזרת הפקודה cron -e נכנסו לעריכת הרשימה של ה- cronים.

וראינו את המשימות הבאות –



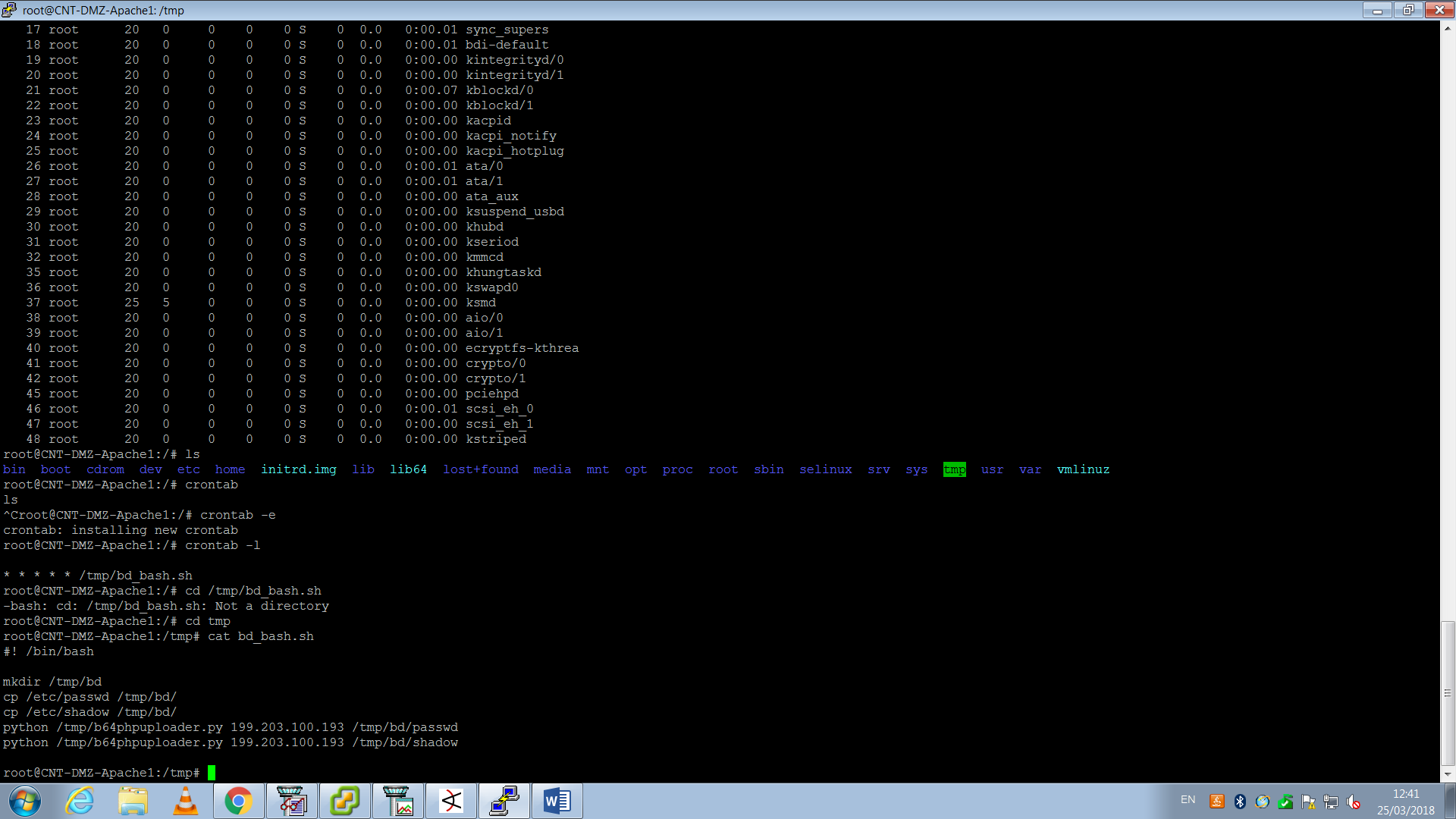
ניתן לראות שיש 2 משימות –

המשימה הראשונה - /tmp/bd/bash.sh מריצה כל דקה את הקובץ bash.sh. (תכף נגיע אליה).

המשימה השנייה - /etc/init.d/apache2 stop מפסיקה את ה- apache2 – זאת אומרת, האתר נופל כל דקה (הכוכביות מסמנות את הזמן).

בהמשך למשימה הראשונה –

ניתן לראות בתמונה הבאה את התוכן של הקובץ –



הקובץ הוא סקריפט כלשהו שמבצע את הפעולות הבאות –

1 – פותח תיקייה בשם db בתוך tmp

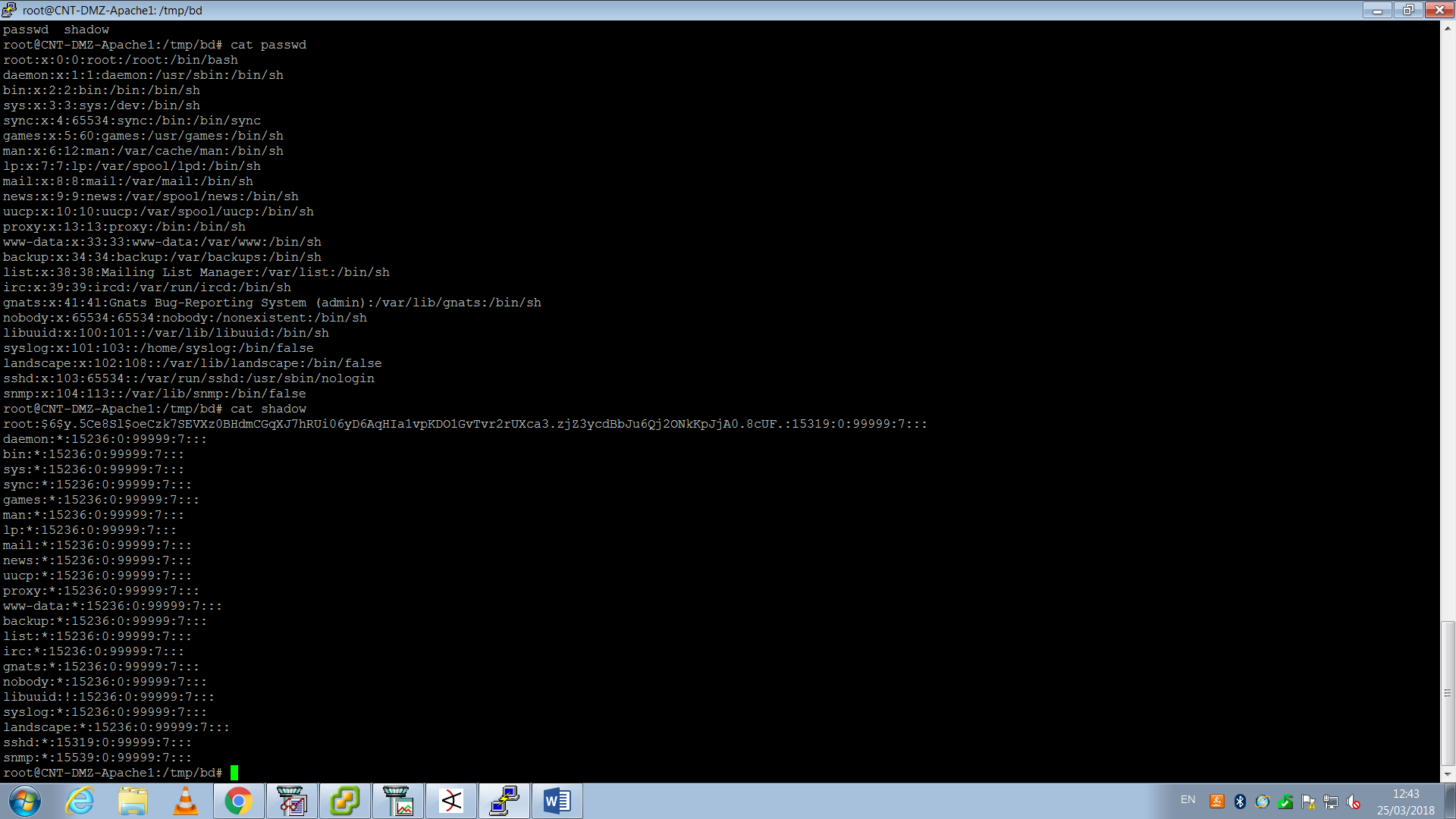
2 – מעתיק קובץ passwd מתוך התיקייה etc ומעתיק קובץ shadow מתוך התיקייה etc

את שני הקבצים מעתיק לתיקייה שפתח בפעולה 1 – db.

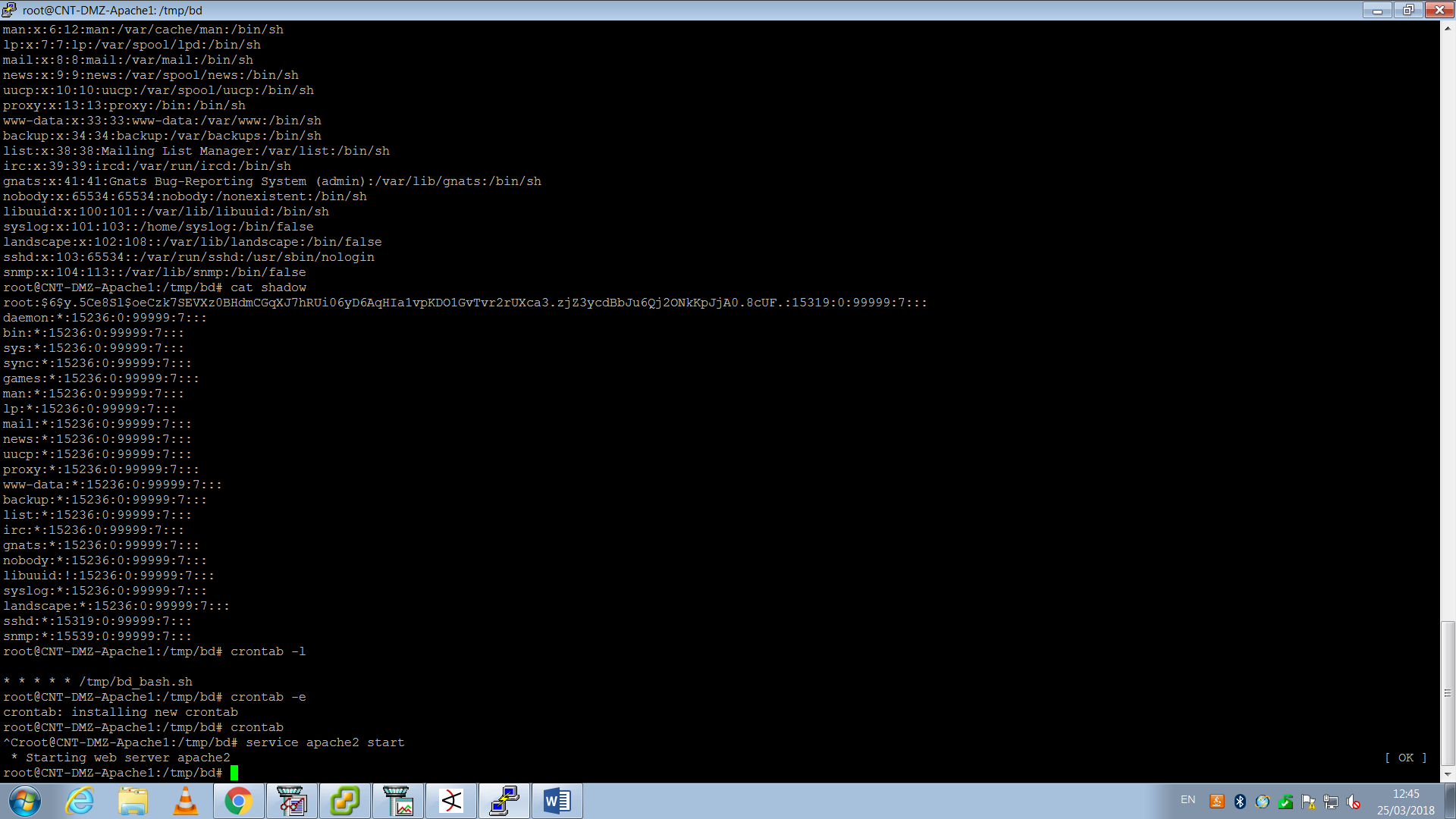
3 – מריץ קוד python ושולח לתוקף את קובץ ה- passwd שהעתיק.

4 – מריץ קובץ python ושולח לתוקף את הקובץ shadow שהעתיק.

ניתן לראות כאן את קובץ ה- passwd ואת קובץ ה- shadow.

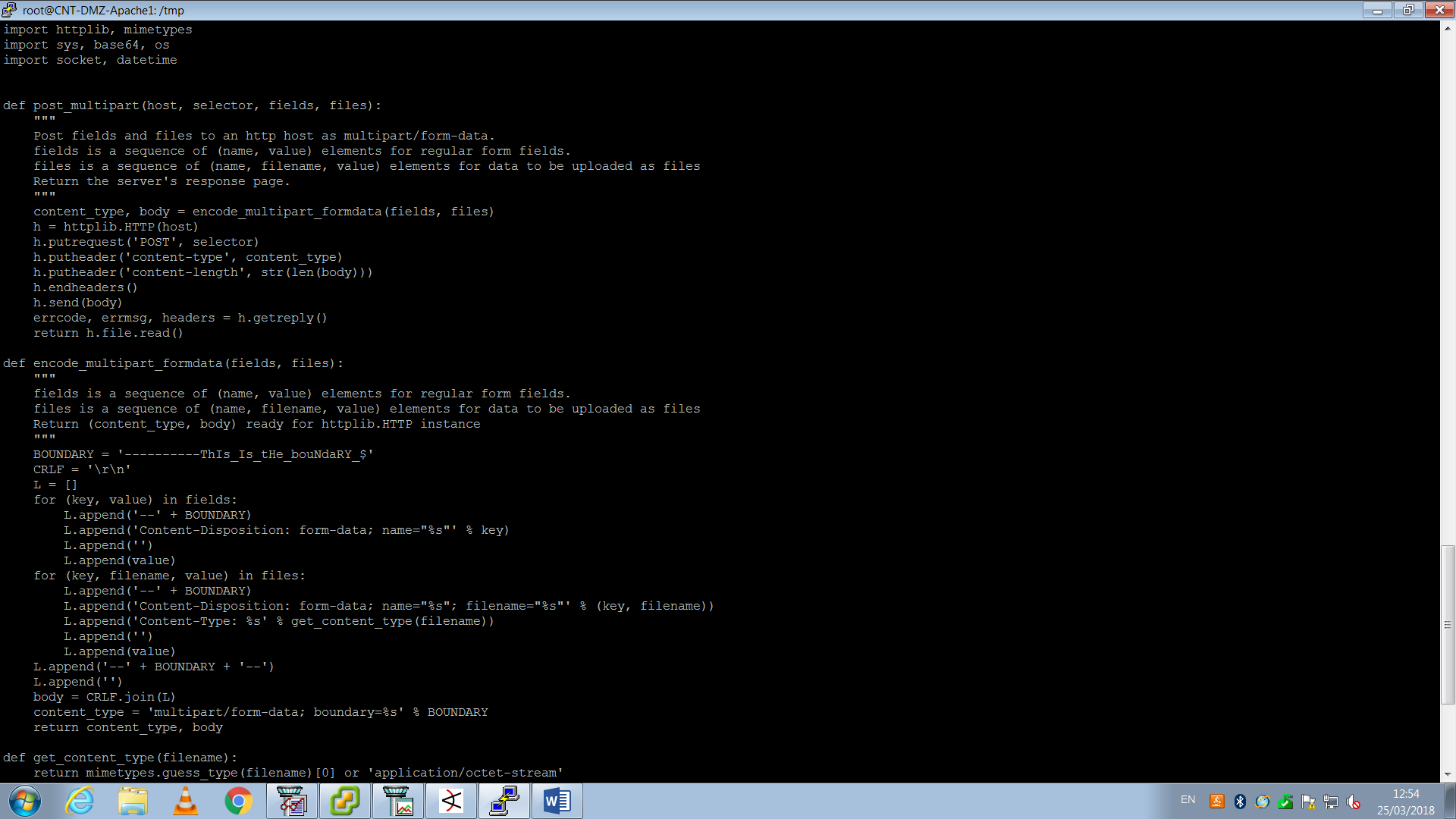


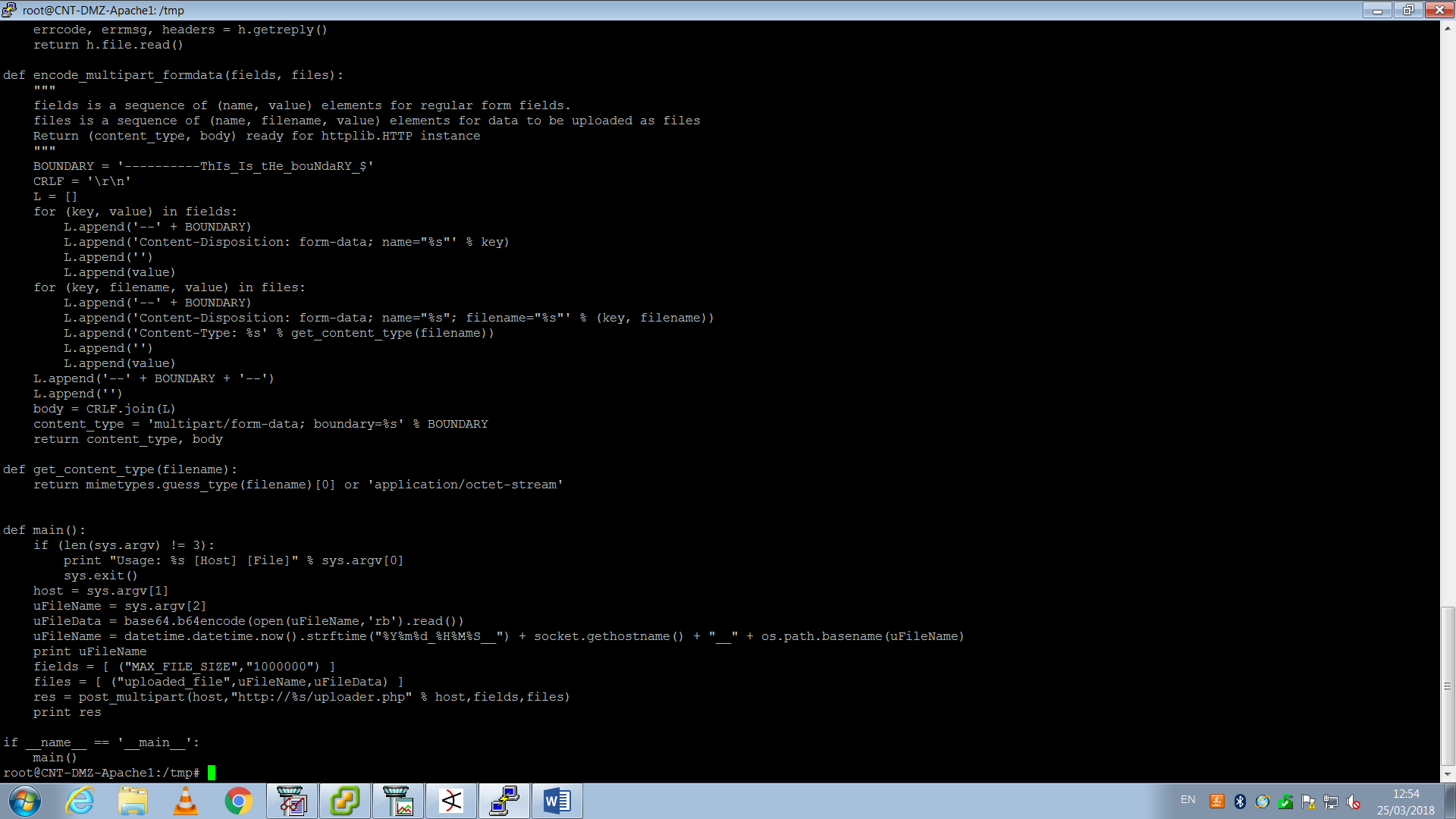
מחקנו את השורות ממתזמן המשימות – (וכמובן את הקבצים הזדוניים).



והפעלנו מחדש את האתר, וכעת הוא פעיל וזמין.

את קוד ה- python ניתן לראות כאן –





* על הקובץ etc/passwd – מקור ויקיפדיה.

הקובץ /etc/passwd הוא מסד נתונים טקסטואלי המכיל מידע על משתמשים הרשאים להתחבר למערכת או ששייכים אליהם תהליכים.

הקובץ משמש לאימות סיסמאות.

לרוב, לקובץ הרשאות מערכת קבצים המאפשרת לכל משתמש לקרוא אותו, אך רק למשתמש-על (ROOT) לשנותו.

הקובץ מכיל רשומה אחת בכל שורה, אשר מייצגת משתמש יחיד.

כל רשומה מכילה שבעה שדות המופרדים בנקודותיים ( : ).

באופן כללי, השדות, בסדר משמאל לימין, הם:

1 – שם משתמש

2 – מידע המשתמש לאימות סיסמא (ברוב מערכות ההפעלה המודרניות השדה מכיל x בלבד, והמידע נשמר בקובץ נפרד, כאשר השדה מכיל \*, זוהי דרך נפוצה לבטל התחברות למשתמש זה).

3 – מזה משתמש (מספרי)

4 – מזהה קבוצה ראשית (מספרי)

5 – תיאור

6 – מיקום (תיקיית בית)

7 – מעטפת ברירת מחדל

* על הקובץ etc/shadow – מקור ויקיפדיה.

הקובץ /etc/shadow נועד להגביר את רמת האבטחה על ידי הפרדת המידע הרגיש על הסיסמאות לקובץ נפרד, אשר רק משתמש-על (ROOT) יכול לקרוא.

השדות ברשומה בקובץ זה, הם:

1 – שם משתמש

2 – המידע על הסיסמא בפורמט CRYPT, אשר מכיל את מספר פונקציית ההאש, salt, ותוצאת הפונקציה.

הסימן ! או \* מייצגים משתמש נעול.

שדה ריק מייצג התחברות ללא סיסמא.

3 – זמן שינוי הסיסמא האחרון

4 – מספר הימים עד שיהיה ניתן לשנות שנית את הסיסמא

5 – מספר הימים עד שחובה יהיה לשנות את הסיסמא

6 – מספר הימים עד להתראה על שינוי סיסמא קרב

7 – מספר הימים עד שהמשתמש יחדל לפעול כאשר נדרשת החלפת סיסמא

8 – זמן תפוגת המשתמש

ניתן להסיק מהקוד שמצאנו (python) שתוכן הקבצים הללו נשלח לכתובת האייפי של התוקף וכך הוא יכול להתחבר לארגון עם המידע שברשותו או להתחזות לאחד מהשרתים המקומיים של הארגון ו-"לתקוף מבפנים".

בנוסף, ניתן להסיק זאת גם על ידי הפעולות שראינו ב- Zenoss שציינתי קודם.

לבסוף, לאחר מחיקת השורות ממתזמן המשימות בלינוקס והרצת השרת הכל פעל כשורה.

# תהליך הגנה מונעת:

בתהליך זה יש כמה דברים שכדאי לעשות בארגון כדי למנוע תקיפה כזו –

1 – האפשרות להתחבר לשרתי הארגון באמצעות SSH שפתוח לכניסה חיצונית (מחוץ לארגון – מחוץ לרשת הפנימית) יכולה להוות בעיה גדולה להגנה על הארגון.

ככל הנראה האפשרות הזאת פתוחה על מנת שטכנאים יתחברו מרחוק (טכנאים שלא נמצאים ברשת הפנימית) ובארגון שלנו האפשרות הזאת פתוחה וכך התוקף הצליח להתחבר בקלות.

פתרון לדבר זה – נקצה ב- FIREWALL אפשרות כניסה ל-IP ספציפיים של הטכנאים או כל אדם שקשור בצורה חיצונית לארגון שנרצה שתהיה לו גישה דרך ה- SSH לשרתי הארגון שלנו.

בתרחיש זה התוקף הצליח להתחבר חיצוני בהתחלה ולאחר מכן שלח לעצמו באופן אוטומטי באמצעות סקריפט סיסמאות לשרתים אחרים בארגון (שרתים מקושרים לשרת הנתקף) ולכן חסימה זו תמנע פגיעה גדולה יותר בארגון בתהליך ההתקפה.

2 – כתיבת חוק שברגע שמישהו מנסה לבצע סריקת פורטים יותר מ- X פעמים – ייחסם, כנ"ל לגבי ניסיונות באמצעות סיסמאות שונות.

3 – כדאי לחזק את הסיסמא, סיסמאות פשוטות קלות לפענוח (בנוסף, החלפת סיסמאות בארגון כל X זמן).

4 – כדאי שתהיה אפשרות לשחזר את קבצי השרת בכך שנוכל לעשות REST לכל הקבצים ולמצב שנחזור לזמן לפני התקיפה (כמובן שדבר זה ידרוש גיבוי כל X זמן ויגרום ככל הנראה לאיבוד מידע כלשהו).

# הפרצות באבטחת הארגון

בתהליך ההגנה עבור תרחיש זה ניתן לראות כמה פרצות אבטחה בארגון כגון: (יש כאן כמה דומים לתרחיש 1 ודברים שכתבתי בסעיף הקודם).

1 – חוסר בחוק המונע סריקת פורטים ובנוסף חוק שמונע מספר ניסיונות התחברות ל- SSH באמצעות סיסמאות שונות.

2 – סיסמא פשוטה מדי.

3 – אפשרות כניסה ל- SSH כגורם חיצוני (אייפי לא מוכר לארגון – הרחבה בסעיף הקודם).

# כלים שפיתחנו

אין ברשותנו כרגע את הידע לפתח כלים.

# אופן עבודת הצוות

בתרחיש זה חילקנו את עבודת הצוות בכך שכל אחד יחקור נושא אחר תוך כדי תהליך ההגנה.

למשל, יצחק היה בפיקוח על ה- Zenoss ועדכן אותנו בפעילויות חשודות.

מאמין שבתרחישים גדולים יותר תהיה לנו חלוקת עבודה גדולה יותר בכך שכל אחד יחקור כיוון אחר – כרגע 2 התרחישים האחרונים היו יחסית קטנים וממוקדים ולא היה צורך בחלוקה עבודה גדולה.

# חוסרים/קשיים

-