דו"ח מעבדה- תרחיש מס' _2_

מגיש: יגאל נאמן

תאריך: 18.11.19

שם התרחיש:WEB DEFACEMENT

הסבר קצר על אופן המתקפה:(**Web defacement)**

היא מתקפה אשר מוחקת\מחליפה דפי אינטרנט (בדיוק כמו שראינו במאמן סייבר בשיעור קודם)

WEB DEFACEMENTנחשב ל אחד האתגרים הגדולים ביותר לכל ארגון הפועל באינטרנט.

מחיקת דפי אינטרנט מתבצעת בדרך כלל על ידי האקרים בכדי לפרוץ לשרת אינטרנט ולהחליף את האתר מתארח עם אחד משלהם, תוך שימוש בטכניקות כגון התחזות, הזרקת קוד cross site scripting אתר ועוד.. ,להאקרים מטרות משותפות של השחתה הם אתרי דת, אתרי אינטרנט ממשלתיים, בנק אתרי אינטרנט ואתרי חברות.

ראה: #DARKNET irc.efnet.net

יש להתחבר דרך MIRCערוץ של קבוצת האקרים.

אם חיפשת אותי , תמצא אותי בכינוי: r2_K2

תהליך ההתקפה:

תהליך ההתקפה הינו מצב שבו התוקף קודם שולח בקשה דרך תעבורת הרשת לפורטPORT מסויים (תהליך זה נקרא PORN SCANING דיברו על זה כבר זה מצב שהתוקף קודם כל סורק אותך במטרה לאתר אצלך חולשות אבטחה VULNIRABILTYS

בד"כ הסריקה נתבצעת ממחשב זומבי,מחשב שנפרץ, או מפרוקסי VPN בכדי שיהיה יותר קשה להעלות על העקבותת, בנוסף הסריקות בד"כ מבוצעות דרך ה-NMAPנמצא בפקודת הSHELL

פקודת הסריקה אצל התוקף Nmap -v -St IP port

על מנת לאתר חולשות אבטחה....

(לאחר שהבנתם את דרכו של התוקף.....)

התוקף מבקש את אובייקט מהאתר כלומר עושה לו wget דרך ה shell

ובכך אוסף מידע על האתר עצמו ויכול לזהות חולשות שבהם הוא יכול לתקוף או איזורים שהם פגיעים ברגע שהתוקף מזהה כי ישנו איזור בו הוא יכול לתקוף לאחר איסוף המידע שבמקרה שלנו (שרת אפצי WEB DEFACEMENT)התוקף מזהה כי יוכל להזריק קוד זדוני+קובץ דרך איזור shell זוהי הפקודה אותה מריצים שבעזרתה ניתן לכתוב פקודות גם לשרת APACHEישירות.

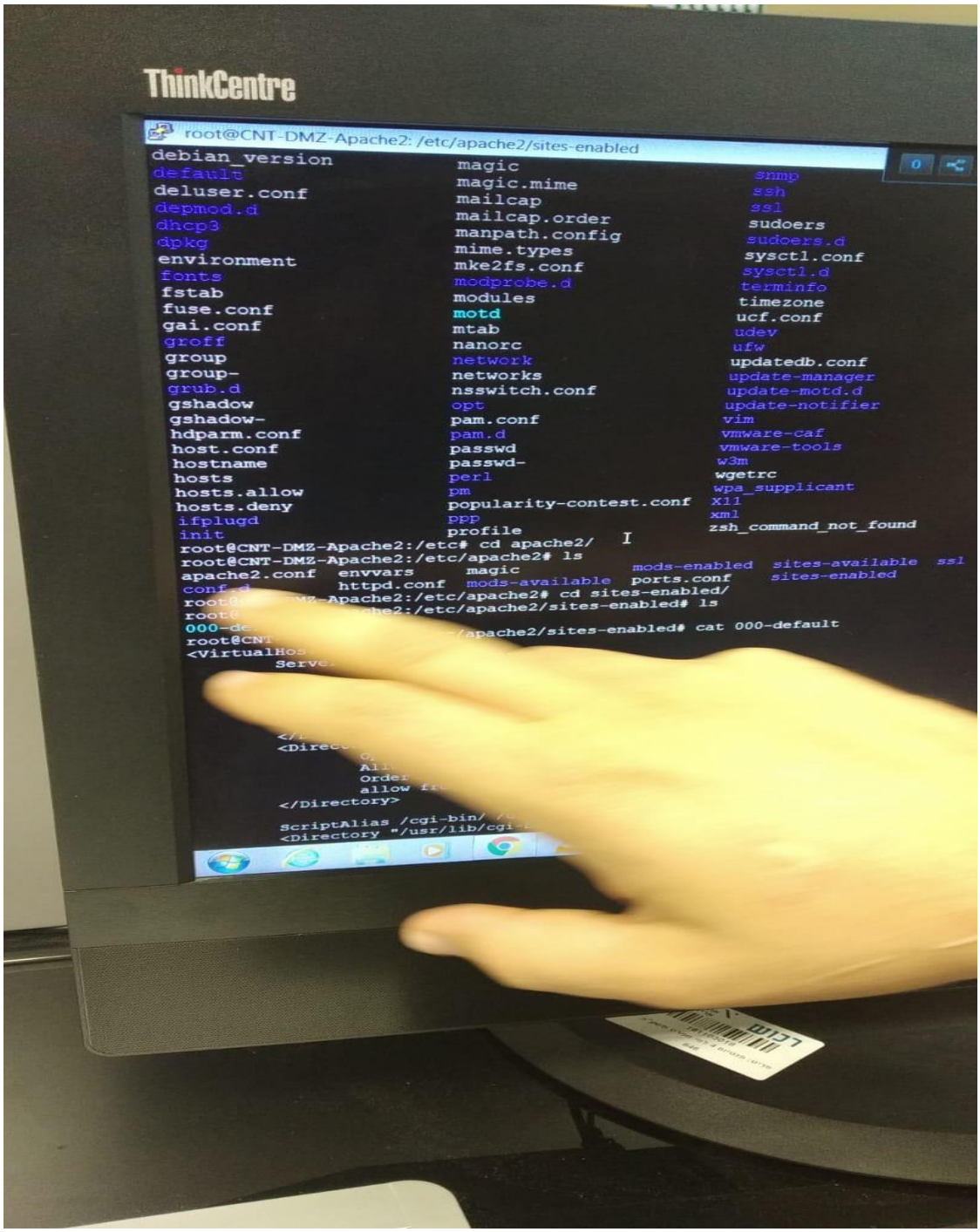
<u>ראה לאחר תום הסריקה אצל התוקף , כך נמצאו החולשות:(דוגמא להמחשה)</u>

```
`/ tokyoneon ~
  > nmap --script nmap-vulners,vulscan --script-args vulscandb=scipvuldb.csv -sV -p22 1[redacted]21

Starting Nmap 7.60 ( https://nmap.org )
Nmap scan report for 1[redacted]21
Host is up (0.54s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.3 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:4.3:
|       CVE-2006-5051          9.3            https://vulners.com/cve/CVE-2006-5051
|       CVE-2006-4924          7.8            https://vulners.com/cve/CVE-2006-4924
|       CVE-2007-4752          7.5            https://vulners.com/cve/CVE-2007-4752
|       CVE-2010-4478          7.5            https://vulners.com/cve/CVE-2010-4478
|       CVE-2014-1692          7.5            https://vulners.com/cve/CVE-2014-1692
|       CVE-2009-2904          6.9            https://vulners.com/cve/CVE-2009-2904
|       CVE-2008-4109          5.0            https://vulners.com/cve/CVE-2008-4109
|       CVE-2007-2243          5.0            https://vulners.com/cve/CVE-2007-2243
|       CVE-2017-15906         5.0            https://vulners.com/cve/CVE-2017-15906
|       CVE-2006-5052          5.0            https://vulners.com/cve/CVE-2006-5052
|       CVE-2010-5107          5.0            https://vulners.com/cve/CVE-2010-5107
|       CVE-2010-4755          4.0            https://vulners.com/cve/CVE-2010-4755
|       CVE-2012-0814          3.5            https://vulners.com/cve/CVE-2012-0814
|       CVE-2011-5000          3.5            https://vulners.com/cve/CVE-2011-5000
|       CVE-2011-4327          2.1            https://vulners.com/cve/CVE-2011-4327
|_      CVE-2008-3259          1.2            https://vulners.com/cve/CVE-2008-3259
| vulscan: scipvuldb.csv:
| [44077] OpenBSD OpenSSH up to 4.3 Signal denial of service
| [39331] OpenSSH 4.3p2 Audit Log linux_audit_record_event unknown vulnerability
| [32512] OpenBSD OpenSSH up to 4.3 unknown vulnerability
| [43307] OpenSSH 4.0 unknown vulnerability
| [41835] OpenSSH up to 4.8 unknown vulnerability
| [38743] OpenSSH up to 4.6 unknown vulnerability
| [36382] OpenBSD OpenSSH up to 4.6 information disclosure
| [32699] OpenBSD OpenSSH 4.1 denial of service
| [2667] OpenBSD OpenSSH 4.4 Separation Monitor Designfehler
| [2578] OpenBSD OpenSSH up to 4.4 Signal race condition
| [32532] OpenBSD OpenSSH 4.5 packet.c denial of service
| [1999] OpenBSD OpenSSH up to 4.2p1 scp system() Designfehler
| [1724] OpenBSD OpenSSH 4.0 GSSAPIDelegateCredentials Designfehler
| [1723] OpenBSD OpenSSH 4.0 Dynamic Port Forwarding Designfehler
| [26219] OpenBSD OpenSSH up to 4.1p1 information disclosure
| [16020] OpenBSD OpenSSH 4.5 Format String
```

כמובן שחלק מהאיתור של התוקף בשרת האפצי עצמו נצטרך לאתרה ע"י הפקודה Netstat -an ולאחר מכן לחפש אייפי זר שמגיע מהרשת החיצונית ושהיא מותרעת ב ZENOOS אשר תצביע על כתובת התוקף.

שתי פעולות שניתן לבצע עם Terminal shell

- Create a shell Job with a Task Step having a type of PowerShell or OS CmdExec
- Use Windows Task Scheduler and batch files



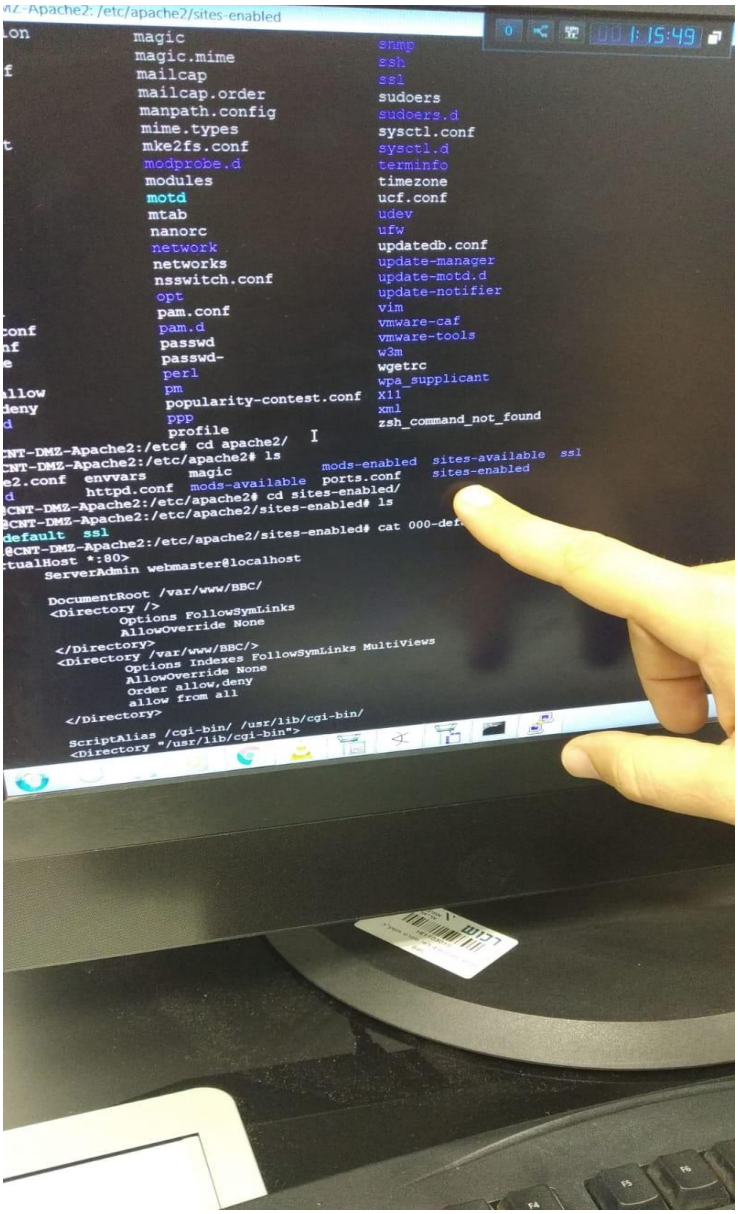
תיאור מצב:

התחברנו לשרת ה אפצי של נו , / cd

נכנסו ל ROOT

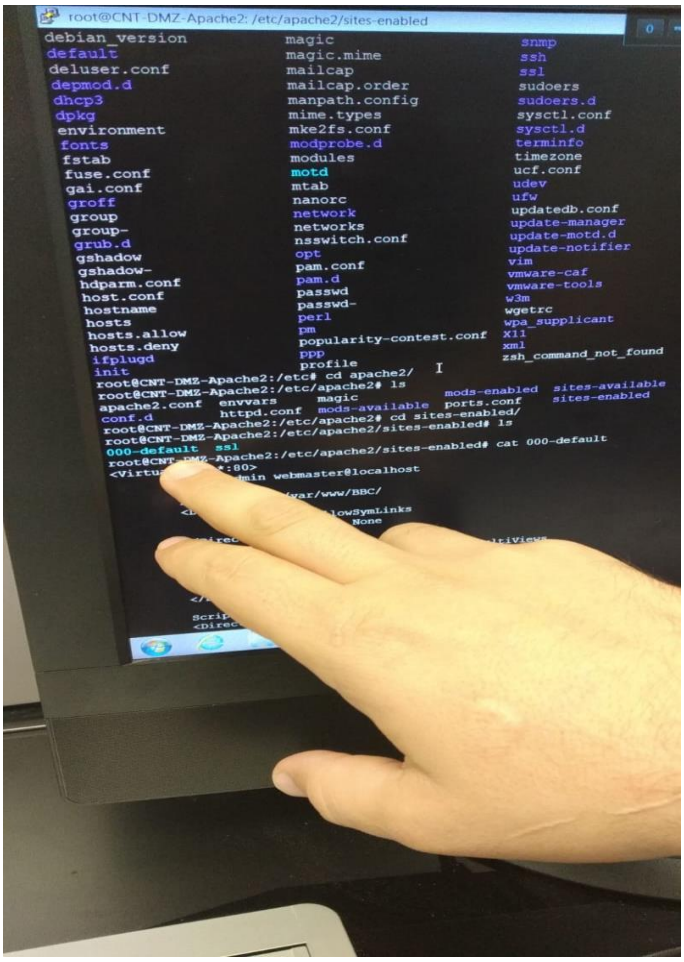
root@cnt-dmz-Apache2:/etc/apache2/

כתבנו ls בשביל לראות מה יש שם



CD root@cnt-dmz-Apache2:/etc/apache2/SITE-ENABLES

נכנסתי לשרת למיקום בתיקייה עשיתי ls

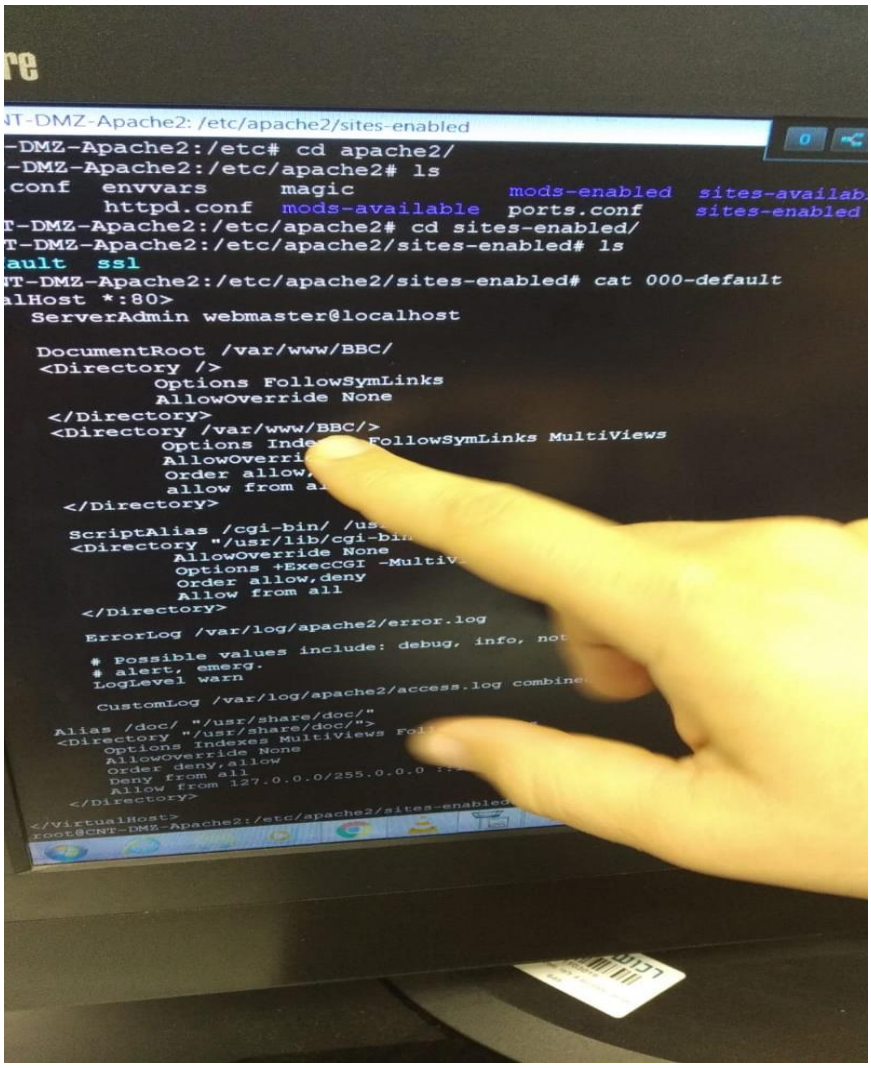


ערכנו את 000- DEFAULTעי קאט

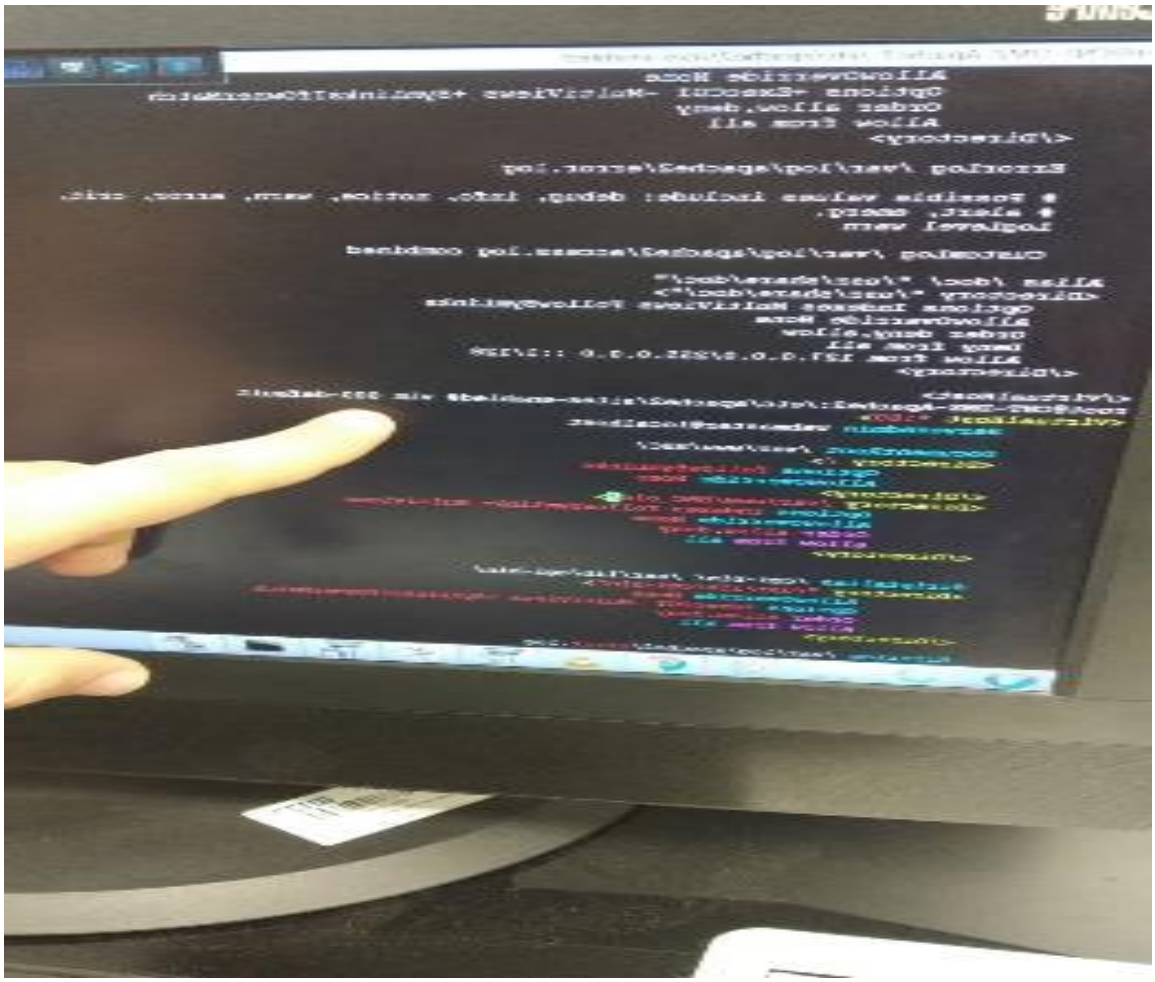
CAT 000-DEFAULT

ושם ראינו לאן בעצם כתובת האתר מופנת ב WWW BBC

היה עלינו לערוך את הקובץ ולהחזירו למצב BBC _OLD



מקום הנתקף!! /VAR/WWW/BBC



תיאור קצרצר מסוף התהליך, כאשר ערכנו את קובץ התמונה הזדוני שמחליף את ה- PAGE

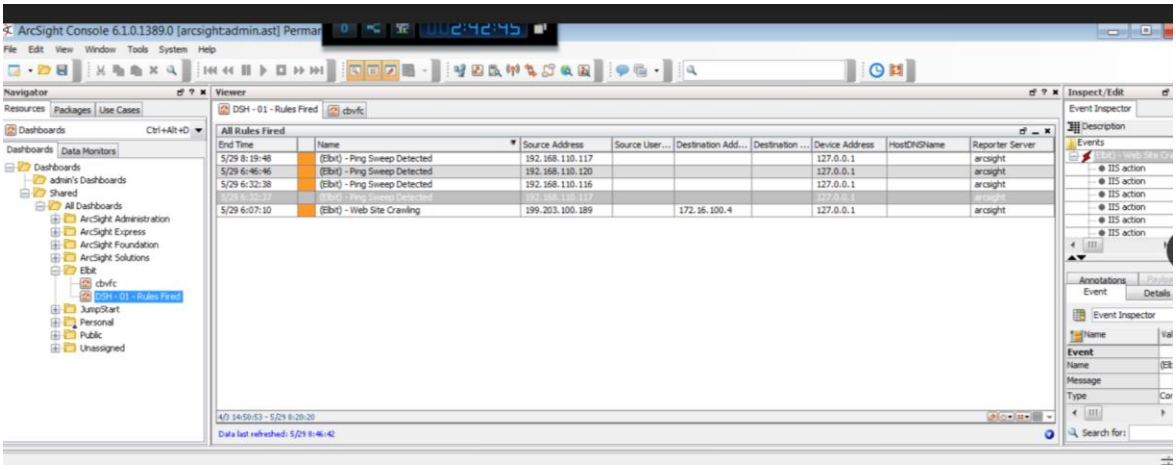
בשרת האפצי

תהליך הזיהוי:

בתהליך הזיהוי תחילה קיבלנו התראה ב arcsight שיש הרבה התקפות של SWAP PINGS

משמעות הדבר, כי התוקף מצליח להחליף כתובות IP כל כמה שניות ודבר זה מקשה לחסום אותו מן המערכת , PASSWORD GUSSINGגניבת סיסמאות (ציינתי זאת בדוח הקודם) נוסף על כך מה שהיה גם ניתן לעשות זה לחסום את CLASS שלו 1.2.*.*

התייעצתי עם אחראי המשמרת בעיניין זה ונאמר לי שמיותר לחסום את זה(מסיבת הזמן).



ניתן לראות שההתקפה מגיעה מכתובת ip חיצונית 192.168.110.117 אל שרת ה APACHE serverשבארגון שכתובתו 172.16.100.4

לאחר מכן נכנסו לשרת ה APACHE אשר נמצא בארגון שלנו ובדקנו בקבצי ה log איזהם חריגיות וזיהנו כי יש הרבה בקשות get מצד אותו כתובת Ip שזיהנו כתוקפת וניתן לראות זאת בתמונה שלהלן

הזרקת נתונים ל שרת האפצי ולאחר מכן ביצוע הפקודה cmdshell שאיתה ניתן ליצור שאילתות או לייצר איזשהו תהליך אשר קשור ל שרת אפצי או ל שרת הנתונים של האתר וזה פגע לנו בשרת בכך שהוא נחשף לגורם חיצוני וגם הציג דף "**YOU ARE HACKED**"

Tip: Don't be shy!!!....

Make sure to drop the attacker with logs from the list by define top any:any with the ip of the attacker! make sure to save a logs file! to talk about this with the isp of the attacker.

<u>תהליך הגנה מונעת + תהליך הגנה עצמו:</u>

תהליך ההגנה המיידי הוא לנסות לחסום כרגע כל תקשורת מכתובת ip שהותקפת

להגידר חוק שברגע שיש תנועה חריגה מאותו גורם יש להשהות את פעולתו לאלתר

תהליך הגנה לטווח הרחוק הוא לייעץ לארגון לבצע בקרת קלט כלומר שלא יוכלו להזין ערכים שאינם חוקיים!

אין פתרונות קסם! או תוכנה או חומרה שמבטיחה 100% בטיחות נגד החספוס באינטרנט, אבל יש שיטות עבודה מומלצות שיכולות למנוע ו\או להקטין את הבעיה של מחיקת\שינוי דפי האינטרנט.

1.ביקורות אבטחה ובדיקות חדירה (PEN TEST)

האקרים מנסים תמיד לנצל פגיעויות שאינן מטולאות כראוי(אין להם טלאי תיקון). ZERODAY זהו אחד הפגיעויות הידועות: שימוש ביציאות פתוחות כדי להתחבר לשרת מבלי להיכנס, לבצע קוד זדוני על גבי חיבור לגיטימי פתוח, באמצעות הצפת מאגר כדי לייבא קוד זדוני המבוצעת

בהקשר האבטחה של המערכת בשרת . בדיקות ביקורת וחדירה שוטפות מסייעות בהערכת האבטחה של תשתית IT (מערכות הפעלה, פגמים בשירותי יישומים, בתצורות לא תקינות או בהתנהגות מסוכנת של משתמשי קצה) והגנה טובה יותר על המערכת.

הנה כמה שיטות עבודה מומלצות המתייחסות לסוגיית מחיקת האינטרנט:

.2להגן על עצמך מפני התקפות הזרקתSQL

התקפות הזרקת SQL לערב את השימוש של משפטי SQL מוכנס לשדות הזנת נתונים על מנת להשפיע על ביצוע משפטי SQL מוגדרים מראש. עם הצהרות SQL שונה, התוקפים עשויים להיות מסוגלים להתעסק עם נתונים קיימים, להרוס נתונים במערכת, או אפילו לחלץ את מסד הנתונים כולו של המערכת.

יישומי אינטרנט רבים לוקחים קלט משתמש מטופס, והקלט של המשתמש מועבר ישירות להצהרת SQL בתוך יישום האינטרנט. להלן דוגמה שלילית לפיה התוקף יכול בקלות להיכנס "דוא"ל" שמוביל להתקפה הזרקת:SQL

<-mysqli $ = $ $שאילתה') בחר דוא"ל Userid, מחברים;' ('.$ email' = 'WHERE email

אתה יכול בקלות למנוע זאת באמצעות משתנים מאוגדים עם שיטת הצהרה מוכנה; רוב הספריות מאפשרות לך לקשור תשומות למשתנים בתוך משפט ,SQL כמו בדוגמה זו.PHP

<-mysqli $ stmt = $ הכנה") בחר דוא"ל Userid, מחברים;("=?; WHERE email

$ stmt-> bind_param ("s", $ email);

<-stmt $לבצע;()

סיבה נוספת

מומלץ להגן מפני התקפות Scripting בין אתרים(XSS)

קרוסס סייט סקריפטינג.... (שם המתקפה באתגלית CROSS SITE SCRIPTING

Scriptingבין אתרים הוא כאשר התוקף מנסה להעביר קוד script לתוך טופס אינטרנט כדי לנסות להפעיל קוד לא מורשה באתר. זה מאפשר לתוקפים להטביע קוד scripting בדף אינטרנט שיכול לבצע מגוון של פעולות לא מורשות כולל: שינוי המראה של דף האינטרנט, WEB DEFACEMENT

גניבת קבצי Cookie של משתמשים אחרים באתר או אפילו כאמצעי ליצירת התקפות XSS על אתרים אחרים.

כדי למנוע התקפות ,XSS עליך למנוע מהמשתמש להזריק קוד באמצעות טופסי אינטרנט.

אחד השיטות המומלצות למניעת התקפות Scripting בין אתרים הוא קידוד פלט תקין

קידוד פלטHTML

אם הנתונים הגיעו מתוך קלט משתמש, מסד נתונים או קובץ

לא 100% יעיל, אך מונע את רוב הפגיעויות

קידוד פלט בכתובת האתר אם מחזירים מחרוזות של כתובות אתרים

כמו כן, חשוב מאוד לאמת קלט, כמו עם מניעת הזרקות .SQL בנוסף, יש להיזהר עם תווים מיוחדים כגון "" <>ו- =. לעתים קרובות, רוב השדות זקוקים לתווים אלפאנומריים בלבד.

חלק ממטרת ההתקפה ב XSS הוא לגנוב עוגיות.(COOKIES) כי ליצור סקריפט זדוני שגורם לדפדפן לשלוח את כל קובצי ה cookie -לכל מבקר בדומיין האתר שלך ואז לתקוף גם אותו.

כי פשוט ככה זה עובד. (חבר מביא חבר)

בכדי להימנע ממצבים אלו, מומלץ להשתמש בחומת אש של יישום אינטרנט .(WAF) הם יכולים לבדוק ערכי קלט זדוני, שינוי של פרמטרים לקריאה בלבד, לסנן פלט זדוני ולחסום בקשות חשוד.

יש להשתמש בתהליך ניטור ככלי עזר לאיתור DEFACEMENT

אחד ההשפעות של התקפות מסוג זה באינטרנט גורמות לחברות לעזוב חברות של שירותי אינטרנט ISP עם הזמן הקצר ולכן רצוי להגיב ולבצע בקרת נזק לאחר תקרית\חקירה.

ולערב כמובן את ספקיות האינטרנט של התוקף וגם את המשטרה.

הפרצות באבטחת הארגון

הפרצה באבטחת הארגון הינה חור באבטחת מערכת ההפעלה , לכאורה על פניו זה עושה רושם של איזשהוא באג שלא תוקן במערכת ושמביא בסופו של דבר לחדירה ל מסד הנתונים שהיה חשוף לביצוע פרוצדורות דרך ה- SHELL , כמובן ששאילתות מסוג זה מתבצעות בעזרת הפקודה CLI

כלים שפיתחנו

לא פיתחתי שום דבר אין לי זמן לינשום חודש הבא יש לי מבחן חיצוני בשפת C

אופן עבודת הצוות

במהלך התרחיש חלקנו תפקידים כך שכל אחד עבד על עמדה מסוימת כאשר אחד ניהל את האירוע