מבדק חדירות

אתר חיל הים

צוות אבטחת מידע

מאי, 2015

תוכן עניינים

1. מאפייני מסמך

מחבר	אודי ברוך
מבקר	
מספר גרסה	1.0
סטטוס	
תאריך הוצאה	
שם קובץ אלקטרוני	

תשומות / הערות

שם/תפקיד	הערה (אופציונאלי)	תאריך	חתימה

היסטוריה

מ. גרסה	ת. הוצאה	מחבר	שינויים מרכזיים בגרסה
1.0	10.09.2015	אודי ברוך	דוח ראשון

הפצה

מ. גרסה	נמענים

2. כללי

הקדמה

מסמך זה מתאר את ממצאי בדיקת החדירות שבוצעה על אתר חיל הים במהלך חודש יוני 2015, שארכה כשלושה ימים.

הבדיקה בוצעה על ידי צוות אבטחת מידע של ממשל זמין, באמצעות בודקי חדירות מוסמכים, המיומנים בתקיפת יישומים ותשתיות.

תיאור המערכת

אתר חיל הים מספק לגולשים שירות מקוון ורחב אודות החייל, היחידות וכלי השייט השונים הקיימים בצה"ל, חדשות, כתבות ועוד. האתר מספק מגוון קישורים נוספים לאתרי צהל השונים. באתר קיים טופס ליצירת קשר.
http://www.navy.idf.il/894-he/Navy.aspx

סיכום ממצאים טכניים

במערכת זוהו חולשות אבטחת מידע המאפשרות לתוקף כלשהו מרשת האינטרנט לתקוף את המערכת ומשתמשיה:

1. גורם כלשהו תוקף את שרת האתר עקב גירסה ישנה ושאינה נתמכת עוד.
2. גורם כלשהו מצליח לחשוף מידע חיוני על המערכת.
3. גורם כלשהו עשוי לנצל פרצות אבטחה באתר עקב יישום ופיתוח לא מאובטח.

חשיפת המערכת לרשת האינטרנט במצבה הנוכחי, מהווה סיכון לפגיעה בתהליכים העסקיים של המערכת, במשתמשי המערכת ובמערכות המחשוב המקושרות אליה.

3. סיכום התוצאות

במהלך המבדק, סווגו הממצאים השונים על פי 4 רמות חומרה אשר נקבעו מראש. רמת חומרת הממצאים נקבעה על בסיס הסיכון הנשקף לארגון בעקבות מימוש החשיפה. להלן רמות החומרה:

קריטית – קיים איום מיידי לתהליכים עסקיים בארגון.

גבוהה – קיים איום ישיר לתהליכים עסקיים בארגון.

בינונית – קיים איום עקיף/חלקי לתהליכים עסקיים בארגון.

נמוכה – לא קיים איום ישיר, אך ניתן לנצל את הפגיעות כדי לבצע תקיפות נוספות.

4. ממצאים

להלן ריכוז כלל הממצאים, שזוהו במסגרת בדיקת החדירות:

מס'	תיאור הממצא	רמת חומרה
Error! Reference source not found.Error! Reference source not found.	גירסת שרת ישנה ואינה נתמכת עוד	***גבוהה***
Error! Reference source not found.Error! Reference source not found.	שימוש ברכיבים לא מעודכנים	**בינונית**
Error! Reference source not 4.2**found.**	לא קיימת הגנה מפני Click-jacking	**בינונית**
4.4	מיפוי תקיות וקבצים בשרת	***נמוכה***
4.5	מנגנון ה- View State בשרת אינו מוצפן	***נמוכה***
4.54.5	מיפוי תקיות וקבצים בשרת	***נמוכה***
Error! Reference source not found.	שרתים לא מוקשחים חושפים מידע פנימי אודות המערכת	***נמוכה***

[redacted]גירסת שרת ישנה ואינה נתמכת עוד

רמת חומרה: גבוהה

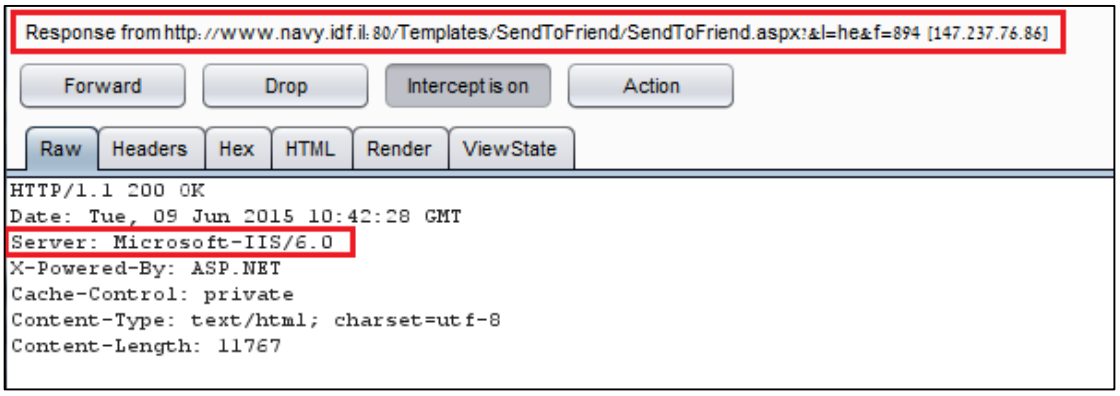
סיווג ממצא: Implementation

תיאור הבעיה

המערכת רצה על גבי שרת שגרסתו לא מעודכנת. כתוצאה מכך, המערכת חשופה למספר התקפות ידועות. ניצול התקפות אלו עלול לאפשר לתוקפים להריץ קוד דזוני על השרת, לבצע פעולות פוגעניות ומזיקות לשרת ועוד.

פרטים טכניים

שרת חיל הים אובחן כשרת IIS6, גרסה זו ידועה כחשופה למספר התקפות קיימות ובמקרה של התקפות חדשות לא ינתן מענה תמיכה למוצר שאינו נתמך יותר.



המלצות לתיקון

על מנת למנוע את החשיפה מומלץ לעדכן את גרסת הרכיב לגרסת היצרן היציבה האחרונה.
בנוסף יש לעדכן את השרת בהתקנות וחבילות ההתקנה על-פי עדכוני האבטחה האחרונים לגרסא האחרונה.

שימוש ברכיבים לא מעודכנים

רמת חומרה: ***בינונית***

סיווג ממצא: *Configuration*

תיאור הבעיה

במהלך המבדק נמצא כי האתר משתמש ברכיב jQuery בגרסה שאינה עדכנית ושקיימות בה בעיות אבטחה. שימוש בספריית jQuery לא מעודכנת חושף את האתר ומשתמשיו לבעיות אבטחה אשר התגלו באותה גרסה (גרסה 1.4.2). להלן קישור לפירצות אבטחה מסוג XSS (Cross Site Scripting) אשר התגלתה בגרסה הקיימת באתר:
http://seclists.org/fulldisclosure/2014/Sep/10

פרטים טכניים

כחלק מבדיקות המערכת נמצא כי נעשה שימוש באתר בספריית jQuery בגרסה ישנה, הגרסה בה נעשה שימוש הינה 1.4.2. היות וגרסה זו חשופה לבעיות אבטחה מסוג XSS, באפשרות גורם זדוני להכין עמוד פיקטיבי אשר גורם להרצה מרחוק של קובץ הג'אווה סקריפט של ספריית ה- jQuery.

הוכחת קיום ממצא:

קיום גרסה ישנה של jquey 1.4.2

המלצות לתיקון

- יש לבחון שדרוג של כל המודולים והתוספים באתר לגרסאות האחרונות בכדי להוריד את הסיכון לפגיעה במערכת. יש לוודא תחילה את תאימות האתר לגרסאות האחרונות ובמידה וקיימת גרסה שאינה תואמת לאתר, יש לעדכן לגרסה הכי עדכנית שניתן.

Click-Jacking

רמת חומרה: בינונית

סיווג ממצא: Insecure Deployment

תיאור הבעיה

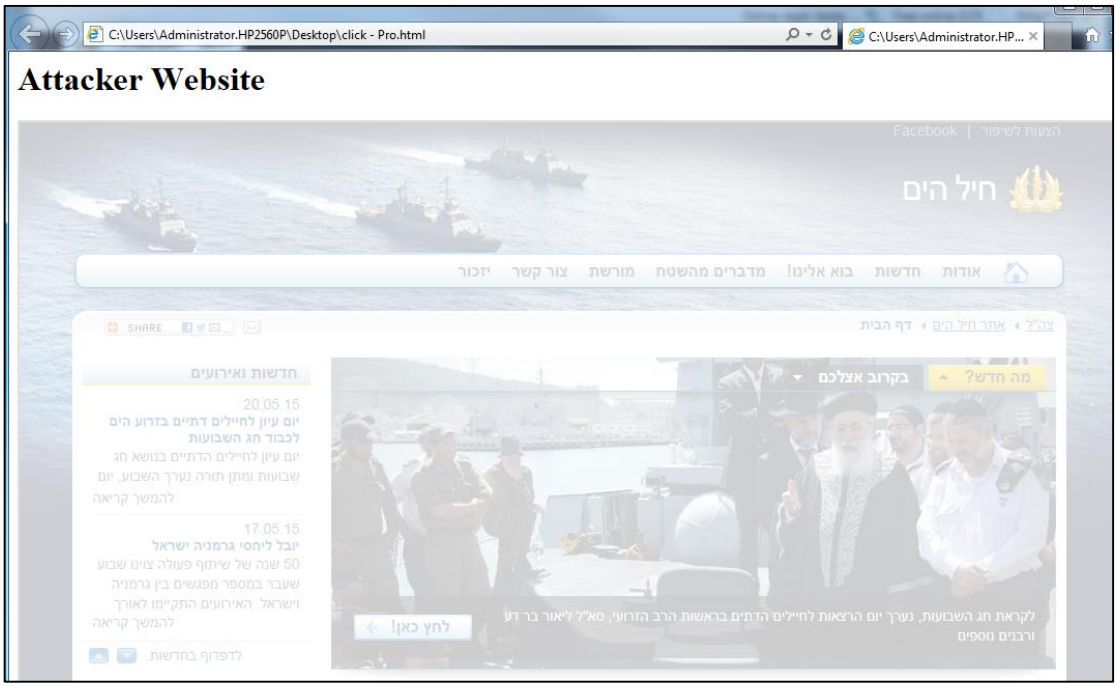
במהלך המבדק נמצא כי לא קיימת הגנה מפני הצגת תוכן באתר מרוחק (Iframe) מה שחושף את משתמשי האתר להתקפות מסוג Phishing ו – Clickjacking היות וניתן להציג תכנים של אתר חיל הים באתרים מרוחקים ללא כל חסימה מצד הדפדפן. יש לציין כי הגדרות למניעת התקפות מסוג זה מגיעות מהשרת והחסימה בפועל מבוצעת בדפדפן שבצד הלקוח.

פרטים טכניים

מתקפת ClickJacking הינה מתקפת הונאה שמטרתה "לגנוב" לחיצות עכבר של משתמשי קצה תמימים על מנת לבצע מגוון של פעולות זדוניות. התקפה זו מבוצעת דרך אתר צד שלישי זדוני הנמצא בשליטה מלאה או חלקית של תוקף, בזמן שמשתמשים לגיטימיים של המערכת גולשים בו.

"גניבת" לחיצות העכבר של משתמש הקצה יאפשרו לפורץ לבצע בשם המשתמש מגוון של פעולות במערכת, ובכלל זאת כל פעולה שניתן לבצע דרך טופס או קישור.

הוכחת הממצא:



המלצות לתיקון

ישנן מספר דרכים מומלצות להתמודדות עם חשיפה זו במערכת, ובמידת האפשר, מומלץ לממש את כולן.

שילוב של מספר טכניקות Anti-ClickJacking:

- מניעת קישור של דפים באתר לפקדי Frame ע"י קוד JavaScript ייעודי המבצע פעולה בשם Frame Busting. להלן דוגמת קוד ב-Javascript:

```
<script>
if (top!=self)
    top.location.href=self.location.href;
</script>
```

- שימוש ב- HTTP Response Header בשם X-FRAME-OPTIONS, אשר מונע מדפים להיות ממוקמים בתוך Frame (נתמך בעיקר בדפדפנים חדשים). להלן שני הערכים הניתנים להגדרה ב- Header:
    - DENY – מונע מהדף להיות מוצג בתוך Frame באופן גורף.
    - SAMEORIGIN – מונע מהדף להיות מוצג בתוך Frame במידה ורכיב ה-Frame המקושר אליו אינו מאותו Domain (מומלץ כברירת מחדל, במקום הפתרונות האחרים בסעיף 1, ובמיוחד בדפים שאמורים להיות מוצגים בתוך FRAME פנימי באתר). יש לציין כי פתרון זה נתמך על ידי דפדפני Internet Explorer מגרסא 8 ומעלה (אך בעתיד, עשוי להיתמך על ידי דפדפנים נוספים). להלן דוגמת קוד (מתאים לשפות C# ו-JAVA):

```
// sample code for completely preventing framing of this content
response.addHeader( "X-FRAME-OPTIONS", "DENY" );
// sample code for enabling content framing only from the same domain
response.addHeader( "X-FRAME-OPTIONS", "SAMEORIGIN" );
```

יישום מנגנון טוקניזציה:

ניתן להתמודד עם ההתקפה על ידי הטמעה של מנגנון הגנה מהתקפות CSRF כאשר המנגנון ימומש כך שיחלוש על כלל הדפים הפנימיים במערכת (מלבד דף ה-Login), וזאת בכדי לאפשר לו להתמודד גם עם התקפות ClickJacking.

כדי להקטין את הסיכון מהתקפות הונאה, בנוסף למאפיינים הרגילים של מנגנון זה, במקרה וה - AntiCSRF Token אשר התקבל מצד המשתמש אינו תקין, יש לנתק את המשתמש מהמערכת באופן יזום.

מיפוי תקיות וקבצים בשרת

רמת חומרה: ***נמוכה***

סיווג ממצא: *Configuration*

תיאור הבעיה

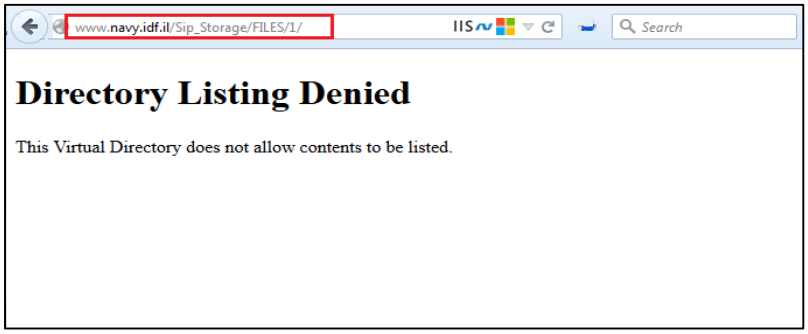
הבדלים בתגובות של אפליקציית חיל הים לבקשות שונות לקבצים וספריות מאפשר לתוקפים למפות ספריות וקבצים על השרת.
היכולת למפות קבצים וספריות בשרת מפחיתה את המורכבות של התקפות מילון ו- Brute Force, ומאפשרת לתוקפים גילוי של קבצים וספריות רגישות. מידע זה עשוי לעזור לתוקפים למנף התקפות נוספות.

פרטים טכניים

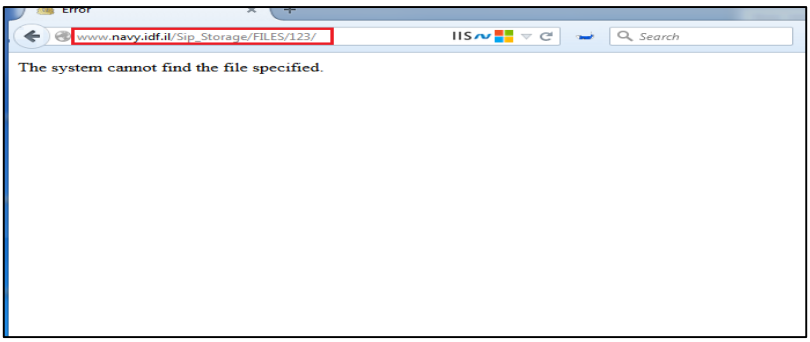
הבדלים בתגובות השרת חושפים את קיומם של תקיות וקבצים ומאפשר לתוקפים למפות אותם ביעילות רבה יותר.

הוכחת קיום ממצא:

ניתן לראות תגובות שונות בהתאם לנתיב קיים ולא קיים:
נתיב קיים:



נתיב לא קיים:



המלצות לתיקון

האפליקצייה צריכה להציג הודעת שגיאה כללית ללא תלות בנתיבים מסויימים.

[redacted] מנגנון ה- View State בשרת אינו מוצפן

רמת חומרה: ***נמוכה***

סיווג ממצא: *Data Exposure*

<u>תיאור הבעיה</u>

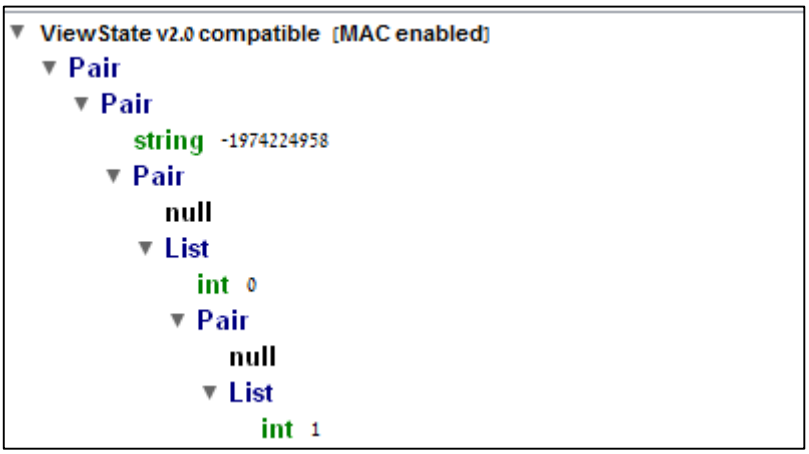
במהלך המבדק נבחנו הבקשות והתשובות השונות המועברות וחוזרות משרת המערכת ונמצא כי קיימת עבודה עם מנגנון ה- View State המכיל מידע בהתאם לבקשות השונות באתר. View State הינו מנגנון המאפשר לשמור נתונים בין הבקשות החוזרות והשונות באתר. לאחר ניתוח התעבורה נראה כי המידע המועבר בשרת במנגנון ה- View State הינו מקודד בלבד ולא מוצפן, מה שמאפשר לחשוף יותר מידע על אופי העבודה של המערכת בין הבקשות השונות באתר.

<u>פרטים טכניים</u>

כברירת מחדל באמצעות מנגנון ה- View State ניתן להעביר מידע בצורה שאינה מוצפנת אלא המידע מקודד בקידוד מסוג base64 אשר ניתן להמירו לטקסט רגיל ללא צורך בפיענוח הצפנה, עם זאת ניתן להגדיר כי המידע המועבר ב- View State יועבר תמיד בצורה מוצפנת מה שיחשוף פחות מידע אודות מבנה המערכת לגורם זדוני. לאחר בדיקת המידע המועבר במנגנון זה באתר, ניתן להבין כי המידע מקודד בלבד ואינו מוצפן ולכן ניתן להמירו לטקסט ולצפות בו.

<u>הוכחת קיום ממצא:</u>

דוגמא 1: זיהוי מידע במנגנון ה- View State



<u>המלצות לתיקון</u>

- יש להגדיר בהגדרות הדף את הצפנת ה- View State באמצעות ההגדרה הבאה:

```
ViewStateEncryptionMode="Always"
```

וכך המידע המועבר שם יועבר תמיד בצורה מוצפנת.

חשיפת שרת האפליקציה

רמת חומרה: נמוכה

סיווג ממצא: Fingerprinting

תיאור הבעיה:

כברירת מחדל, שרת הרשת חושף את גרסתו וסוגו. מידע זה יעיל ביותר עבור תוקפים, אשר יכולים לחפש ברשת פגיעויות ידועות ומוכרות עבור אותו סוג שרת ולנצל אותם על מנת לפרוץ אל השרת.

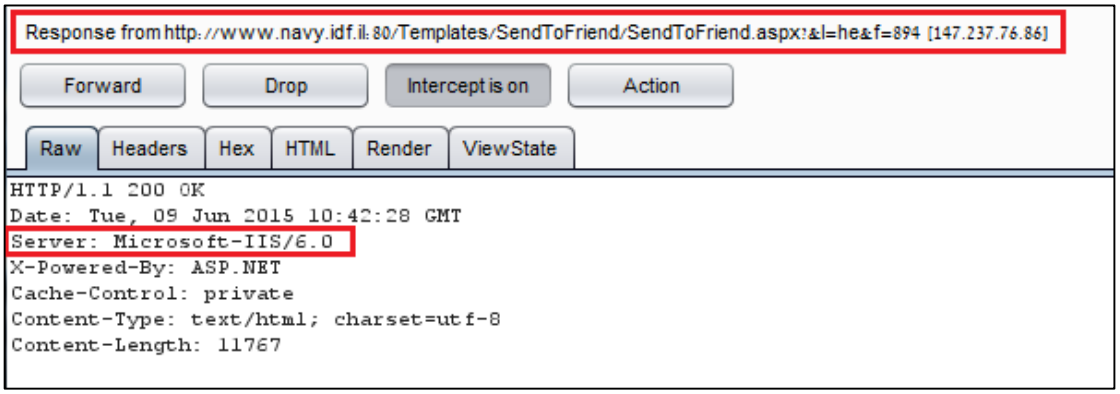
פרטים טכניים:

שרת האפליקציה חושף מידע רגיש, כגון סוג השרת.

הערה: פגיעות זאת נמצאה במספר מקומות באפליקציה ולכן צריך לטפל בה בכל המקומות בהן היא מופיעה, ולא רק בדוגמאות המופיעות בממצא זה.

הוכחת הממצא:

ניתן לראות כי סוג שרת הרשת וגרסתו נחשפים ב-headers המוחזרים משרת האפליקציה.



המלצות לתיקון:

יש להסיר את סוג וגרסת השרת מה-HTTP Headers.