תכנות בטוח תרגיל 4

חולשה:

החולשה שעליה נרחיב בתרגיל זה היא העלאת קבצים שרירותית לא מאומתת, המובילה לביצוע קוד מרחוק בדפדפנים מבוססי כרומיום.

מזהה CVE:

מזהה הCVE של חולשה זו הוא CVE-2021-30118.

הקדמה + תיאור:

תחילה נסביר את המושג "נקודת קצה" (endpoint). נקודת קצה היא כתובת אתר ספציפית או URL שנחשף על ידי יישום אינטרנט או API ומייצג מיקום או משאב ייחודי בתוך היישום או המערכת.

לדוגמה, ביישום אינטרנט, נקודות קצה שונות עשויות להתקיים עבור פונקציונליות כגון אימות משתמשים, אחזור נתונים, הגשת נתונים, העלאת קבצים או כל פעולות ספציפיות אחרות בהן האפליקציה תומכת. לכל אחת מנקודות הקצה הללו כתובת אתר ייחודית הקשורה אליו.

כאשר לקוח (כגון דפדפן אינטרנט או יישום אחר) רוצה לקיים אינטראקציה עם פונקציונליות ספציפית או לאחזר נתונים ממערכת, הוא שולח בקשת HTTP לכתובת האתר של נקודת הקצה המתאימה. השרת, עם קבלת הבקשה בנקודת הקצה שצוינה, מעבד את הבקשה ומחזיר תגובה מתאימה.

החולשה שבה אנו עוסקות, קיימת בנקודת הקצה SystemTab/uploader.aspx/ של

Kaseya VSA Unified Remote Monitoring & Management גרסה 9.5.4.2149.

החולשה מאפשרת למשתמשים לא מאומתים להעלות קבצים לשרת תוך שימוש באותה רמת גישה והרשאות כמו תהליך שרת האינטרנט עצמו, ללא בדיקות אימות מתאימות. על ידי שליחת בקשה לנקודת קצה זו, התוקף יכול לנצל את החולשה ולהעלות קבצים שרירותיים בכל מקום בו תהליך שרת האינטרנט יש גישה לכתיבה, ובכך עשוי להשיג גישה בלתי מורשית או ביצוע קוד שרירותי בשרת.

שלבים:

ניצול החולשה מתבצע על ידי מספר שלבים:

שלב ראשון - שליחת הבקשה:

נראה דוגמה לבקשת HTTP POST שנעשתה לנקודת הקצה SystemTab/uploader.aspx/ ונפרט את המרכיבים השונים של הבקשה.

נסתכל על הבקשה הבאה:

```
POST
/SystemTab/uploader.aspx?Filename=shellz.aspx&PathData=C%3A%5C
Kaseya%5CWebPages%5C&__RequestValidationToken=ac1906a5-d511-
47e3-8500-47cc4b0ec219&qqfile=shellz.aspx HTTP/1.1
Host: 192.168.1.194
Cookie:  sessionId=92812726;
%5F%5FRequestValidationToken=ac1906a5%2Dd511%2D47e3%2D8500%2D4
7cc4b0ec219
Content-Length: 12

<%@ Page Language="C#" Debug="true" validateRequest="false" %>
<%@ Import namespace="System.Web.UI.WebControls" %>
<%@ Import namespace="System.Diagnostics" %>
<%@ Import namespace="System.IO" %>
<%@ Import namespace="System" %>
<%@ Import namespace="System.Data" %>
<%@ Import namespace="System.Data.SqlClient" %>
<%@ Import namespace="System.Security.AccessControl" %>
<%@ Import namespace="System.Security.Principal" %>
<%@ Import namespace="System.Collections.Generic" %>
<%@ Import namespace="System.Collections" %>

<script runat="server">

private const string password = "pass"; // The password (pass)
private const string style = "dark"; // The style (light/dark)

protected void Page_Load(object sender, EventArgs e)
{
      //this.Remote(password);
      this.Login(password);
      this.Style();
      this.ServerInfo();


<snip>
```

שיטת הבקשה היא "POST", מה שמציין שהנתונים יישלחו לשרת.

כפי שהזכרנו קודם לכן, נקודת הקצה שניגשים אליה היא /SystemTab/uploader.aspx.

פרמטרי השאילתה הכלולים בכתובת האתר הם:

- "Filename=shellz.aspx": מציין את שם הקובץ הרצוי עבור הקובץ שהועלה, כ-"shellz.aspx".
- "PathData=C%3A%5CKaseya%5CWebPages%5C": מציין את נתיב היעד של הקובץ שהועלה בתור "C:\Kaseya\WebPages". ה-"%3A" ו-"%5C" הם ייצוגים מקודדים ב-URL של התווים ":" ו-"\" בהתאמה.
- "__RequestValidationToken": מייצג את אסימון אימות בקשה, שהוא אמצעי אבטחה למניעת סוגים מסוימים של התקפות (כמו סקריפטים בין אתרים).

- "qqfile=shellz.aspx": מתייחס לקובץ המועלה, שצוין גם כ-"shellz.aspx".
- Host: 192.168.1.194: מציין את שם המארח או כתובת ה-IP של השרת שאליו ניגשים. במקרה זה, "192.168.1.194".
- "Cookie: sessionId=92812726; %5F%5FRequestValidationToken=ac1906a5%2Dd511%2D47e3%2D8500%2D47cc4b0ec219" :
מכיל את העוגיות הנשלחות עם הבקשה. בדוגמה זו, יש שתי עוגיות:
  - "sessionId=92812726": מייצג את מזהה ההפעלה המשויך להפעלה של המשתמש.
  - "F%5FRequestValidationToken=ac1906a5%2Dd511%2D47e%53%2D8500%2D47cc4b0ec219": מייצג את אסימון אימות הבקשה שהוזכר קודם לכן.
- "Content-Length: 12 ": מציין את אורך גוף הבקשה בבתים. במקרה זה, גוף הבקשה הוא באורך 12 בתים.

גוף הבקשה מכיל קוד ASPX שיועלה כתוכן הקובץ. הוא כולל קוד עבור דף ASP.NET מבוסס C# המגדיר משתנים, מייבא מרחבי שמות ומיישם פונקציות שונות.

<u>שלב שני - עקיפת אימות</u>:

כאשר משתמש מקיים אינטראקציה כלשהי עם מערכת Kaseya VSA Unified Remote Monitoring & Management, נוצר session כדי לשמור על המצב המאומת שלו.
הsession הזה מזוהה בדרך כלל על ידי session id, שמאוחסן בקובץ cookie בדפדפן של המשתמש.

בדרך כלל, כאשר מתבצעת בקשה ליישום, היא צפויה לכלול את אותו session id, השרת בודק את ה-session Id כנגד נתוני ההפעלה המאוחסנים שלו כדי לאמת את זהות המשתמש ולהבטיח שהבקשה לגיטימית.

עם זאת, במקרה הזה, התגלה שהשרת לא מאמת כראוי את ה-session id, ולכן תהליך האימות פגום וכתוצאה מכך כל ערך מספרי יכול לשמש כ-session Id חוקי, ללא קשר אם הוא מתאים ל-session מאומת או לא.

המשמעות היא שהתוקף יכול לעקוף את דרישת האימות, ועל כן הוא יכול להצטייר כמשתמש מאומת מבלי שיש לו אישורים מתאימים. זה מאפשר לו לגשת למשאבים מוגבלים, לבצע פעולות לא מורשת ולנצל פגיעויות אחרות במערכת.

<u>שלב שלישי - העלאת הקובץ הזדוני</u>:

עם הבקשה המעוצבת, התוקף מעלה קובץ המכיל קוד זדוני (למשל קוד ASPX) למיקום נגיש על ידי תהליך שרת האינטרנט. זה יכול לכלול את ה-webroot, (ספריית ה-root של שרת האינטרנט) או ספריות אחרות שבהן לתהליך יש הרשאות כתיבה.

<u>שלב רביעי - ביצוע הקוד הזדוני</u>:

לאחר העלאת הקובץ הזדוני, מכיוון שלתהליך שרת האינטרנט יש גישת כתיבה למיקום הקובץ, הקוד הזדוני יורץ, מה שעלול להוביל לפגיעה מלאה של המערכת.

<u>השפעה</u>:

ההשפעה של חולשה זו היא חמורה ועלולה לגרום להשלכות הבאות:

1. העלאת קבצים שרירותית:

התוקף יכול להעלות קבצים לפי בחירתו לכל מקום בשרת שבו לתהליך שרת האינטרנט יש גישת כתיבה. זה כולל ספריות רגישות כגון webroot, ספריות תצורה או ספריות מערכת קריטיות אחרות.

2. ביצוע קוד מרחוק (RCE):

על ידי העלאת קבצים זדוניים המכילים קוד, התוקף יכול לבצע פקודות או סקריפטים שרירותיים. זה מאפשר לו לסכן את השלמות, הסודיות או הזמינות של המערכת, ולקבל גישה לא מורשית או לגנוב אישורים.

3. פגיעה במערכת:

אם התוקף יריץ בהצלחה קוד זדוני, הוא עלול לקבל שליטה מלאה על המערכת שנפרצה. זה יכול להוביל להתקפות נוספות, חילוץ נתונים, גישה לא מורשית למערכות אחרות או הפרעה לשירותים.

<u>הימנעות</u>:

כדי לצמצם את הסיכונים הקשורים לחולשה זו, מומלץ לבצע את הפעולות הבאות:

1. תיקון ועדכון:

החלת עדכוני האבטחה והתיקונים האחרונים שסופקו על ידי Kaseya עבור

Management & VSA Unified Remote Monitoring.

עדכונים אלה עשויים לכלול תיקונים עבור הפגיעות.

2. אימות:

הטמעת מנגנוני אימות נאותים עבור נקודות קצה רגישות כדי להבטיח שרק משתמשים מורשים ומאומתים יכולים לגשת אליהם ולתקשר איתם.

3. סינון קלט:

הטמעת מנגנוני סינון קלט קפדניים כדי למנוע העלאה של קבצים זדוניים או למנוע גישה לא מורשית לקבצים.

4. עקרון ההרשאות הקטנות ביותר:

לפעול לפי עיקרון ההרשאות הקטנות ביותר עבור תהליך שרת האינטרנט והגבלה של גישת הכתיבה שלו לספריות הדרושות בלבד. זה מבטיח כי כל רכיב, כמו תהליך שרת האינטרנט, מקבל רק את הרשאות שהוא זקוק לו, מצמצם את משטח ההתקפה ומקל את הנזק הפוטנציאלי שיכול להיגרם על ידי תוקף או פעילות זדונית.

5. בדיקות אבטחה:

עריכת הערכות אבטחה ובדיקות חדירה קבועות כדי לזהות ולטפל בכל נקודות התורפה האפשריות באפליקציה ובתשתית התומכת שלה.

6. מודעות לאבטחה:

ללמד את מנהלי המערכת והמפתחים לגבי נוהלי קידוד מאובטח, נקודות תורפה אפשריות והחשיבות של עדכונים ותיקונים קבועים.

נציין ש Kaseya סגרו פגיעות זו ע"י שחרור גרסה עדכנית ומתוקנת.

קישור לCVE של החולשה:

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-30118