תקן WS – GOV.IL

ל-GATEWAY הממשלתי

נספח א' – הקשחת סכמות וסוגי נתונים

גרסה 1.6

מסמך זה כולל מידע השייך לממשל זמין, רשות התקשוב הממשלתי. כל חשיפה, שימוש או העתקה של מסמך זה או חלקים ממנו – ללא קבלת אישור בכתב ממנהל מערך סייבר ואבטחת מידע בממשל זמין – אסורה בהחלט. מסמך זה מיועד לעובדי ממשל זמין ולקוחותיו

מעקב גרסאות

מס"ד	תאריך	עודכן על ידי	תיאור השינויים
1.3	7.10.2015	ישי כהן	גרסא ראשונה לאחר המעבר לתבנית ממשל זמין החדשה. מסמך זה נערך על בסיס גירסה 1.2 אשר היתה בתבנית הישנה.
1.4	29.10.2015	יוגב מזרחי	עדכון תכנים והוספת הנחיות ל- WcfExtension
1.5	29.11.2015	יוני ארוך	שינוי לוגו והוספת נתוני גרסת המסמך
1.6	22.2.2016	אופיר יהב	תיקון המחרוזת ל-"*[0-9]" בעמוד 27

נתוני גרסת המסמך

גורם	תפקיד	שם מלא	תאריך	חתימה
נערכה ע"י	PMO	אופיר יהב	7.10.2015	(חתימה)
נבדקה ע"י	מוביל טכנולוגיות במערך סייבר ואבט"מ	אלעד פז	7.10.2015	(חתימה)
אושרה ע"י	מנהל מערך סייבר ואבט"מ	אברהם זרוק	7.10.2015	(חתימה)

תוכן עניינים

1. כללי 5
1.1 מטרה 5
1.2 ריכוז המלצות 5
1.2.1 אין להשתמש בטיפוסים הבאים : 5
1.2.2 דגשים כלליים 5
2. מסמכים ישימים 7
3. כתיבת סכמה מאובטחת 8
3.1 כללי 8
3.2 הקדמה 8
3.2.1 מבוא לפורמט XML 8
3.2.2 מבוא לסכמת XSD 9
3.2.3 תהליך יצירת הסכמה 10
4. דגשי אבטחה 14
4.1 הקדמה 14
4.2 טיפוס any 14
4.3 טיפוס object 15
4.4 שימוש ב-DataSet-ים 16
4.5 כמות הופעות של אובייקט 16
4.6 אורך מחרוזת 17
4.7 טווחי ערכים 18
4.7.1 הגבלת טווח הערכים למחרוזת 18
4.7.2 הגבלת טווח למספרים 20
4.8 כינויים 21
4.9 הגבלת גודל קובץ 22
4.10 תיעוד 22
4.10.1 תיעוד חובה 22
4.10.2 תיעוד רשות 22

4.10.3 תיעוד מומלץ 23
4.10.4 תיעוד נוסף בחשיבות נמוכה יותר 23
4.10.5 אופן כתיבת התיעוד 23
4.11 הפניות לא תקינות בקובץ WSDL 23
4.12 הגדרת אובייקטים ב-Namespace לא תקין 23
4.13 העברת שם משתמש בסיסמה בצורה גלויה..... 24
4.14 בדיקות תקינות המידע המוחזר ללקוח 24
5. אפיון שדות..... 25
6. תווי בקרה בסיסיים..... 27
7. מגבלות..... 29
8. דוגמאות..... 30
9. הנחיית מפתחים להטמעת רכיב GovIL WCF Extention..... 31
9.1 דרישות קדם 31
9.2 סדר פעולות הטמעה..... 31
9.3 תיאור מאפיינים 32
9.4 דוגמאות להקשחת השירות..... 34

1. כללי

1.1 מטרה

תחת תקן WS-gov.il קיימת דרישות במ"מ שונות כגון ביצוע אימות קלטיים וכן אימות של פירטי מידע XML המגיעים לשירותים לפני ביצוע שימוש בהם.

טכנולוגית Web Services מקבלת מידע דרך קבצי XML על גבי פרוטוקול SOAP, בטכנולוגיה זאת ניתן לבצע פעולות אימות קלט עוד לפני שהקלט מגיע לשירות ה-WS, שיטה זאת נקראת אימות סכמות XML ע"י כתיבת סכמות XSD מתאימות.

מסמך זה מספק המלצות לכתיבה מאובטחת של סכמות XSD עבור קבצי XML. כדי שסכמה תהיה מוקשחת ככל שניתן, יש צורך לנהוג לפי כלל least privilege, כדי לאפשר מעבר של תוכן תקין ומאושר בלבד לעבר שירותי ה-WS.

מסמך זה מכיל המלצות לכתיבה מאובטחת של סכמות כולל דוגמאות הממחישות את ההמלצות.

1.2 ריכוז המלצות

להלן ריכוז ההמלצות שבמסמך :

1.2.1 אין להשתמש בטיפוסים הבאים :

1.2.1.1 any - המאפשר הכנסה של ענף XML המכיל מידע כלשהו.
1.2.1.2 Object - המאפשר העברת מידע ע"י אובייקט כלשהו שאינו מוגדר.
1.2.1.3 DataSet-ים כלליים - צורה זו מאפשרת הכנסת DataSet המוגדר לסוג any.

1.2.2 דגשים כלליים

1.2.2.1 יש להגדיר את מספר המופעים המדויק (מקסימום ומומלץ גם מינימום) עבור כל אובייקט אשר עתיד להיכנס לשירות. חשוב לציין שבמידה ונעשה שימוש ב-DLL ההקשחות של ממשל זמין לצורך ביצוע ההקשחות, מכיוון

שעדיין אין תמיכה בהגבלת מספר מופעים של אובייקטים, נכון לעכשיו אין חובה להכניס הגבלה זאת לסכמה.

1.2.2.2 יש להגביל אורך המחרוזות כדי למנוע מצב בו האפליקציה המקבלת קובץ XML תקבל מחרוזת ארוכה מהמצופה. מצב זה יכול לגרום להתקפה פוטנציאלית.

חובה לבצע הגבלה זאת עבור קלט המגיע לשירות לרבות תשובות השירות ללקוח.

1.2.2.3 יש להגביל טווח הערכים עבור המחרוזות. קיימות מספר דרכים לעשות זאת הנפוצה בהן היא בעזרת ביטויים רגולריים. הסיבה להגבלה זו היא למנוע מאפליקציה לקבל קלט לא מצופה ולמנוע התקפה אפשרית.

חובה לבצע הגבלה זאת עבור קלט המגיע לשירות לרבות עבור תשובות השירות ללקוח.

1.2.2.4 מומלץ לעשות הגבלת טווח ערכים גם עבור מספרים, אך זה לא חובה.

1.2.2.5 אין להשתמש בהרחבת DTD עבור הגדרת כינויים (aliases). פתיחת שימוש בהרחבה זו פותח פתח להתקפת DoS אפשרית.

1.2.2.6 אין להשתמש באובייקט מסוג DataSet (לא Dataset כללי ולא typed dataset), עקב יצירת אלמנטים מסוג Any וחוסר תמיכה של תשתיות תהילה בטיפוסי נתונים אלה (עם או בלי namespace מוגדר). במידה ויש צורך בכל זאת להשתמש ב-Typed Dataset, יש לכתוב באופן ידני סכמת XML מתאימה ל-Dataset.

1.2.2.7 יש להגביל את גודל הקבצים המועלים כדי למנוע התקפה אפשרית על האפליקציה תוך התחשבות בכך שגודל בקשה כוללת נכון להיום ב-Data Power תהילה מוגבל גם ככה למקסימום 4 מגה בייט. יש להגדיר את הגדרות השירות בהתאם גם ברמת הסכמה.

1.2.2.8 מומלץ לרשום תיעוד עבור חלקים מהותיים כגון אובייקט להעברת קובץ, אובייקט עם מבנה מורכב, אובייקט טופס וכו'.

1.2.2.9 יש להוריד ולתקן הפניות לא תקינות בקובץ WSDL בטרם העלייה לייצור (למשל להחליף הפניות ל-localhost לכתובות הנכונות).

1.2.2.10 יש לוודא הגדרה נכונה של Namespace עבור על האובייקטים. חשוב לציין שבמידה וקביעת Namespace יוצר בעיות עקב יצירת קבצים כפולים (wsdl ו-wsdl0 למשל) או במידה ונעשה שימוש ב-DLL ההקשחות של תהילה לצורך ביצוע ההקשחות (שם עדיין אין תמיכה ב-Namespace-ים), נכון לעכשיו אין חובה לקבוע Namespace-ים נכונים בסכמה.

1.2.2.11 אין להעביר שם משתמש וסיסמה בצורה גלויה, יש לוודא קיום הצפנה תשתיתית (כגון SSL או S-Box).

1.2.2.12 במקרה של שירותים קיימים שרצים כבר ואינם עומדים בנוהל, כל מקרה יבחן לגופו ויתכן וידרשו שינויים מצד מתכנתי השירות בכדי לגרום לו לעמוד בנוהל. במקרה ולא ניתן לעמוד באחד או יותר מסעיפי ההקשחה (כגון שימוש ב-DataSet-ים שיוצרים אלמנטים מסוג Any ולא מאפשרים ביצוע הקשחה כלל), כל מקרה יבחן לגופו כשלכל הפחות יחויב השירות בבדיקות חדירות בזמן ריצה על מנת להבטיח שאינו מכיל פרצות כלשהן עקב הורדת ההקשחה.

2. מסמכים ישימים

1. תקן WS-gov.il

3. כתיבת סכמה מאובטחת

3.1 כללי

בפרק זה נגדיר קווים מנחים לכתיבת סכמה באופן מאובטח. סכמה מאובטחת תאשר לעבור רק ל-XML במבנה לו מצפה האפליקציה\שירות הקצה, וכך לצמצם אפשרות להתקפה של האפליקציה ולהתקפה של שירותי קישוריות פנים עצמם.

בפרק זה נסקור אפשרויות ליצירת הסכימה ולאחר מכך נציין נקודות שיש לשים עליהם דגש בזמן יצירת הסכימה כדי שהיא תהיה מאובטחת.

מידע נוסף מעבר להקדמה הקצרה למטה ניתן למצוא ב:

1. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/
2. http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/
3. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

3.2 הקדמה

3.2.1 מבוא לפורמט XML

פורמט XML (Extensible Markup Language) הינו פורמט להעברת מידע אפליקטיבי. הפורמט נולד מתקן SGML. הפורמט בנוי מאוסף תגים מקוננים בצורה רקורסיבית ויכול להכיל כל סוג של מידע.

שימוש בפורמט זה מאפשר למפתח לאפיין אך ורק את התוכן והטיפוס של שדות העוברים בין אפליקציות ולא את הפורמט לשמירת הנתונים. בנוסף, פורמט XML הינו קריא מאוד, נוח להבנה, עריכה ויצירה.

פותחו מספר כלים לעבודה עם פורמט XML. תוכנות הנפוצות ביותר לעריכת תוכן בפורמט XML הם .NET Visual Studio, תוכנת XmlSpy. כמו כן, ניתן להציג את מבנה ה-XML בדפדפנים שונים כגון internet explorer של windows.

להלן דוגמא לקובץ XML פשוט. קובץ זה מכיל רשימה של אנשים, עבור כל בן אדם רשומה מכילה שם פרטי, שם משפחה ומספר אישי.

```
<?xml version="1.0" ?>
<persons>
```

```
    <person>
        <first-name>Eran</first-name>
        <last-name>Oliel</last-name>
        <id>1234567</id>
    </person>
    <person>
        <fist-name>Dudu</fist-name>
        <second-name>Bezalel</second-name>
        <id>7654321</id>
    </person>
</persons>
```

3.2.2 מבוא לסכמת XSD

כפי שהוצע בסעיף קודם, קובץ XML יכול להכיל מידע מגוון במבנים שונים. כדי לתאר מבנה של קובץ XML וטיפוסי השדות בו נוצר מנגנון של סכמה (schema).

כיום השיטה הנפוצה להגדרת סכמה היא XSD (Extensible Structure Declaration). בשיטה זו המבנה של XML מתואר ע"י קובץ במבנה XML אחר אשר מגדיר layout חוקי עבור המבנה.

הצורך לווידוא מבנה ה-XML מגיע כדרישה אבטחתית לווידוא המידע שמגיע לשירות (Web Service) טרם השימוש בפועל בשירות עצמו. יכולות אלו באות לידי ביטוי בפרוטוקולים להעברת מידע אפליקטיבי כגון: FTP, HTTP (כולל פרוטוקול SOAP) ו-SMTP.

קונפיגורציה זאת מתעדכנת בהתאם לשינויים שמתבצעים בשירותים השונים.

במסמך זה לא נפרט את כל היכולות הקיימות בפורמט XML וסכימות XSD, במקום זה אנו מפנים את הקורא למקורות מידע זמינים הקיימים באינטרנט (ניתן להתחיל מכתובת www.w3c.org). נציג בתור דוגמא מצומצמת את סכמה אשר מתארת את המבנה של קובץ XML המוצג בסעיף הקודם. סכמה זו לא נכתבה בצורה מאובטחת והיא כאן רק לשם הדוגמא, אנו נשפר את הסכימה בהמשך המסמך:

```
<?xml version="1.0" ?>
<xs:schema id="persons"
    targetNamespace="http://example.co.il/example1"
    xmlns="http://example.co.il/example1"
```

```
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    attributeFormDefault="qualified"
elementFormDefault="qualified">
    <xs:element name="persons">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="person">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="first-name" type="xs:string"
/>
                <xs:element name="last-name" type="xs:string" />
                <xs:element name="id" type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
</xs:schema>
```

3.2.3 תהליך יצירת הסכמה

ניתן ליצור סכמות בדרכים הבאות:

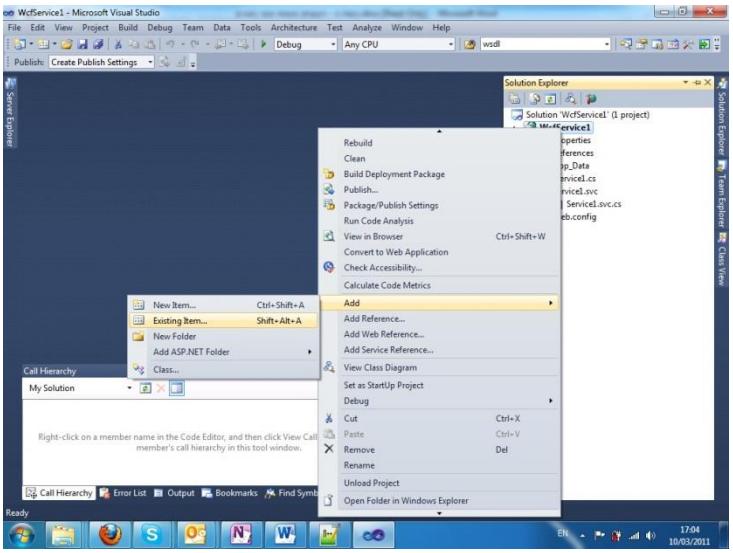
3.2.3.1 **ייצור ידני** - במקרה זה הסכמות מיוצרות בצורה ידנית או באמצעות כלים כגון Visual Studio, XMLSpy וכו' המספקים מעטפת גראפית.

3.2.3.2 **ייצור אוטומטי** - ניתן לייצר סכמה מקובץ XML מוכן באמצעות כלים כגון Visual Studio. כמו כן ניתן לייצר סכמת XSD מסכימה ישנה (למשל, בפורמט XDR) בצורה אוטומטית.

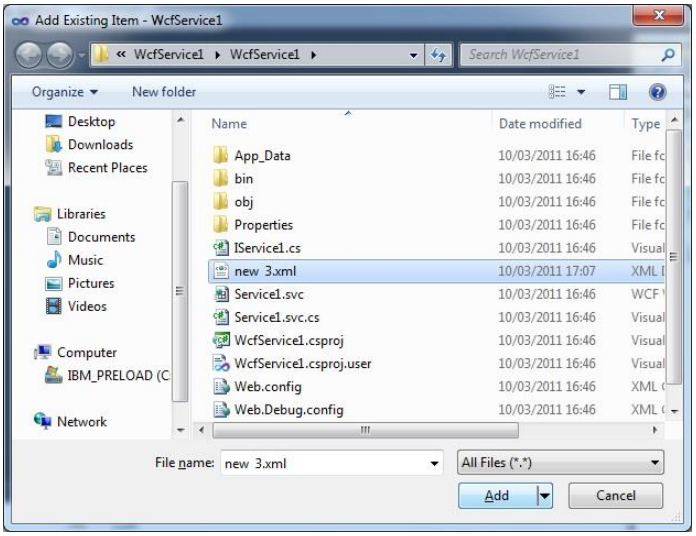
כמובן, יצירת סכמות בצורה ידנית ומבוקרת היא אופטימאלית מבחינה אבטחתית. במקרה זה קל לבנות סכמה מאובטחת מבוססת על דגשים המופיעים בהמשך הפרק. במקרה של ייצור אוטומטי, כברירת מחדל הכלי מייצר סכמה לא מספיק קשיחה (למשל ללא הגבלת אורך המחרוזות, הגבלת מספר מופעים וכד'), ולכן נדרש עדיין לעבור על הסכימה שנוצרה בעין ולהתאים את סכמה לדרישות האבטחה.

דוגמא לבניית סכמה דרך Visual Studio 2010 בהינתן XML קיים:

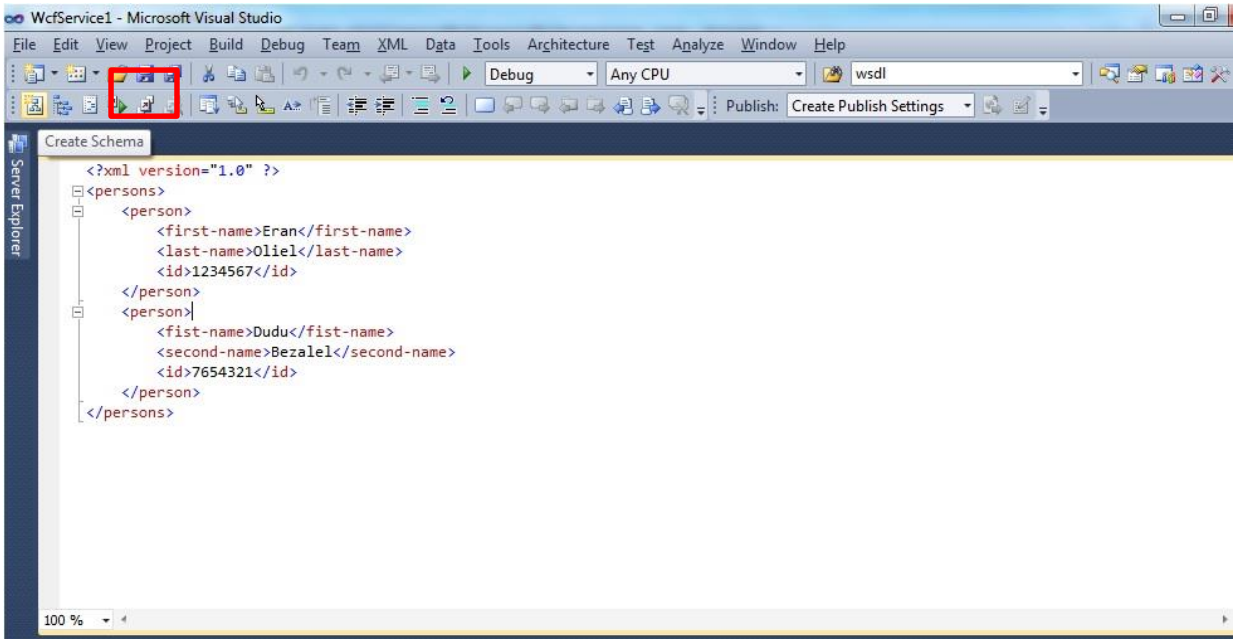
3.2.3.3 הכנסת ה-XML לסביבת העבודה ע"י לחיצה ימנית על שם הפרויקט ובחירת הכנסת פריט קיים



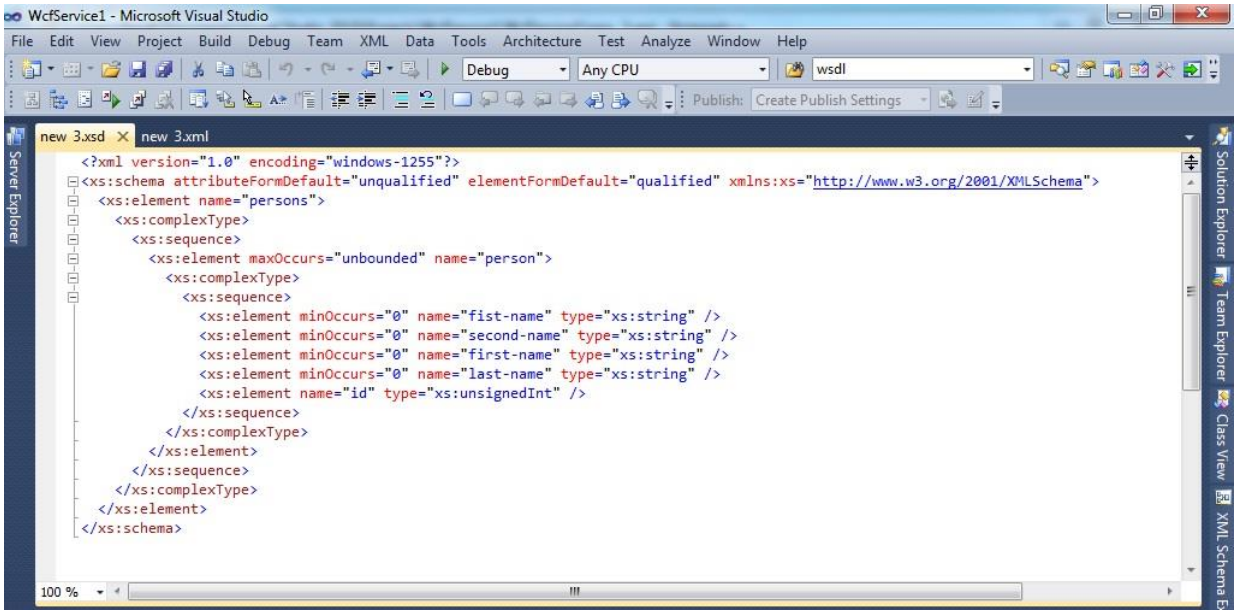
בחירת קובץ ה-XML המתאים (במקרה זה new 3.xml)



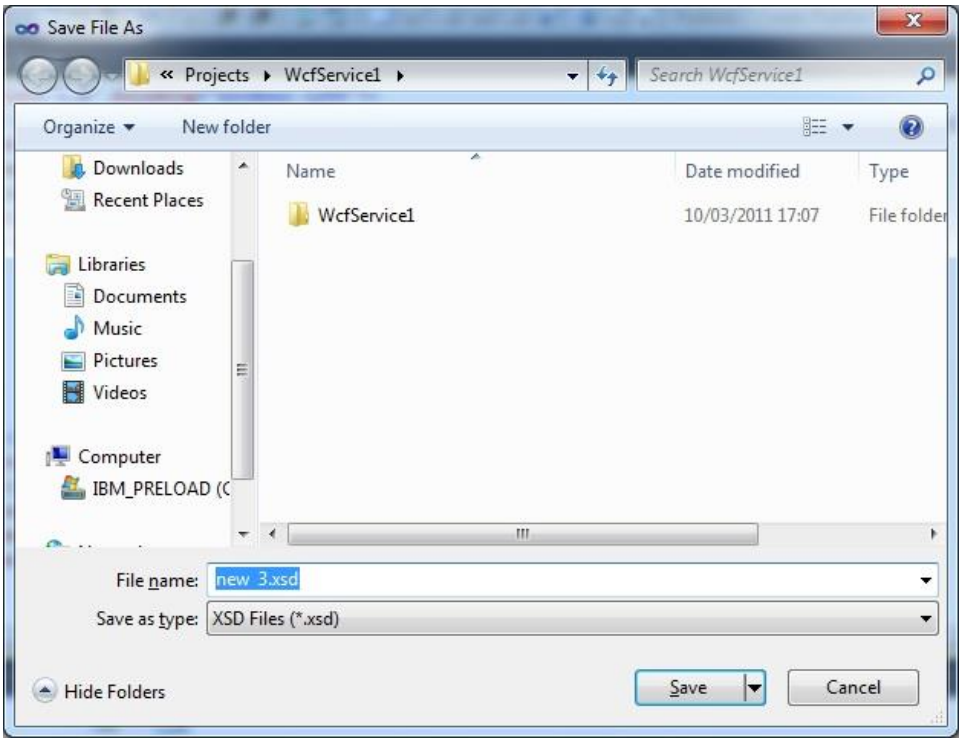
לחיצה כפולה על קובץ ה-XML והוא יפתח על המסך הראשי. עכשיו ניתן לראות כי סרגל הפעולות מעל לחלון הראשי התעדכן וכרגע קיימות אפשרויות לעבודה עם משפחת ה-XML-ים. הכפתור הימני התחתון הינו הכפתור ליצירת הסכמה



לחיצה על כפתור זה תיצור לנו את הסכמה המתאימה והיא תפתח בחלון חדש



כעת, כל מה שנותר לבצע הוא לשמור את הסכמה ע"י לחיצה על כפתור ה-Save


Save File As
« Projects ▸ WcfService1 ▸
Search WcfService1
Organize
New folder
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
IBM_PRELOAD (C
Name
Date modified
Type
WcfService1
10/03/2011 17:07
File folder
File name:
new_3.xsd
Save as type:
XSD Files (*.xsd)
Hide Folders
Save
Cancel

4. דגשי אבטחה

4.1 הקדמה

ממשל זמין פותח שירות חדש של פרסום שירותים (אפליקציות) על ידי המשרדים השונים. שירות זה הוא יצירת Gateway ממשלתי. ה-Gateway יאפשר למשרדים לפרסם שירותים משרדיים בין המשרדים לבין עצמם ובין הציבור (internet) למשרדים.

הפרסום מבוצע ע"י קישור השירות במשרד הרלוונטי ל-Gateway (מוצר בשם "DataPower"). על מנת שה-Gateway יוכל לאבטח את השירות בצורה יעילה, אחד הפרטים אותם נדרש להעביר לצוות SOAממשל זמין את סכמה המגדירה את השירות. בצורה זו ה-DataPower מוודא את המידע הנשלח לשירות (בפועל המידע מעובד טרם הגעתו לשירות) ובמידת הצורך מונע העברת מידע לא תקין.

הסעיפים הבאים מציגים את דגשי האבטחה אשר כל שירות הרוצה להתפרסם דרך ה-Gateway הממשלתי נדרש לעמוד בהם.

במקרה של שירותים קיימים שרצים כבר ואינם עומדים בנוהל, כל מקרה יבחן לגופו ויתכן וידרשו שינויים מצד מתכנתי השירות בכדי לגרום לו לעמוד בנוהל. במקרה ולא ניתן לעמוד באחד או יותר מסעיפי ההקשחה (כגון שימוש ב-DataSet-ים שיוצרים אלמנטים מסוג Any ולא מאפשרים ביצוע הקשחה כלל), כל מקרה יבחן לגופו כשלכל הפחות יחויב השירות בבדיקות חדירות בזמן ריצה על מנת להבטיח שאינו מכיל פרצות כלשהן עקב הורדת ההקשחה.

4.2 טיפוס any

בסכמה טיפוס any יראה בצורה הבאה:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <xs:schema targetNamespace="http://tempuri.org//any"
  xmlns="http://tempuri.org//any"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="xml-body" type="xs:anyType"/>
  </xs:schema>
```

ניתן להכניס לתג xml-body (במקרה זה תג ראשי) כל מבנה, בפרט ניתן ליצור XML הבא:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body targetNamespace="http://tempuri.org//any"
xmlns="http://tempuri.org//any"
xmlns="http://www.w3.org/2001/XMLSchema">
```

```
    <all posible="data"/>
    <more>data</more>
    <maybe possible="attack"/>
</xml-body>
```

ברור שאין להשתמש בטיפוס זה מחשש שהמערכת תקבל קלט שהמערכת אינה מצפה לו ותיפגע בביצועים או תיפול.

בנוסף, כתיבת שירות המקבל כפרמטר XmlDocument מייצר בסכמת האתר משתנה מסוג any. לאור הדברים שהוצגו למעלה, נדרש לערוך את סכמת השירות כדי שניתן יהיה לוודא את תוכן הפרמטר.

מכך נובע שקיים איסור בפלטפורמות NET. על שימוש ב-DataSet-ים מסוג כלשהו בסכמה (רגילים או Typed Dataset) עקב יצירת אלמנט Any שאינו מאפשר ביצוע הגבלות קלט תקינות ואינו נתמך בתשתיות תהילה (Data Power), כפי שכתוב גם בסעיף 4.4.

4.3 טיפוס object

שימוש באובייקט מסוג object כפרמטר לשירות יוצר בסכמה את השורות הבאות:

```
<s:element name="Object_Example">
  <s:complexType>
    <s:sequence>
      <s:element name="o">
   </s:sequence>
  </s:complexType>
</s:sequence>
```

שימוש במשתנה מסוג שמוצג למעלה נותן אפשרות להכניס לשירות מידע מכל סוג מבלי שניתן לבדוק את תוכנו (היות ומוגדר שקיים משתנה בשם "o" אך לא מוגדר טיפוסו). ברור שאין להשתמש בהגדרה מסוג זה.

4.4 שימוש ב-DataSet-ים

הגדרת משתנה מסוג DataSet כפרמטר בשירות מייצרת סכמה המגדירה טיפוס מסוג any.
במצב זה לא ניתן לבצע סינון לנתונים המתקבלים.
נראה כיצד נראית הסכמה במצב המתואר:

```
<s:element name="dsIn">
  <s:complexType>
    <s:sequence>
      <s:element ref="s:schema"/>
      <s:any/>
    </s:sequence>
  </s:complexType>
</s:element>
```

מכך נובע שקיים איסור בפלטפורמות NET. על שימוש ב-DataSet-ים מסוג כלשהו בסכמה (רגילים או Typed Dataset) עקב יצירת אלמנט Any שאינו מאפשר ביצוע הגבלות קלט תקינות ואינו נתמך בתשתיות תהילה (Data Power).

חשוב להדגיש שגם שימוש ב-Typed Dataset אשר יוצר אלמנט Any עם Namespace שאמור להכיל בדיקות קלט כלשהן אינו נתמך בתשתיות תהילה כלל ולכן אין להשתמש בו.
במידה ויש צורך בכל זאת בשימוש באובייקטים מסוג Typed DataSet, נדרש לייצר באופן ידני לכל אובייקט שכזה סכמה מתאימה ולקשר אותה (לבצע import namespace) בסכמות הרלוונטיות.
במקום שימוש ב-Dataset מומלץ להשתמש באובייקטים ומערכי אובייקטים.
במקרה של שירותים קיימים שכבר נעשה בהם שימוש ב-Dataset-ים היוצרים אלמנטים מסוג Any, כל מקרה יבחן לגופו בצוות אבטחת מידע בתהילה כשלכל הפחות יחויב השירות בבדיקות חדירות בזמן ריצה על מנת להבטיח שאינו מכיל פרצות כלשהן עקב הורדת ההקשחה.

4.5 כמות הופעות של אובייקט

נדרש להגדיר בסכמה את מספר ההופעות המקסימלי עבור כל אובייקט אשר נדרש להיכנס לשירות, ומומלץ גם להגדיר מספר מופעים מינימלי אם כי אין חובה לעשות זאת. הגדרת כמות ההופעות תבוצע ע"י הוספת המאפיינים minOccurs ו-maxOccurs.
שים לב! אין להשתמש בפרמטר Unbounded במאפיין ה-MaxOccurs.

בדוגמא הבאה נראה סכמה המגדירה אובייקט בשם persons אשר יכול להכיל מופע של person **לפחות** פעם אחת אך **לא יותר** מ-5 פעמים.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
  <xs:element name="persons">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="person" minOcuurs="1" maxOccurs="5">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="full_name" type="xs:string"/>
              <xs:element name="child_name" type="xs:string"
              minOccurs="0" maxOccurs="5"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

חשוב לציין שמכיוון והגבלות מספר המופעים של האלמנטים אינם נתמכים בשלב זה ב-DLL ההקשחות של תהילה, במידה ונעשה שימוש בכלי זה לשם ביצוע ההקשחות ל- Web Service ניתן לוותר נכון להיום על הקשחה זאת בכדי למנוע מצב של הקשחה גם ברמת הקוד בעזרת ה-DLL וגם באופן ידני.

4.6 אורך מחרוזת

כאשר לא מגבילים אורך המחרוזת אזי XML עלול להכיל מחרוזת ארוכה מאוד. הדבר עלול להפיל את המערכת בצד השני וגם להוות ערוץ סמוי רחב מאוד להעברת מידע.

נתמקד בדוגמא לסכמה הבאה.

```
<?xml version="1.0" ?>
<xs:schema id="xml-body"
     targetNamespace="http://tempuri.org//any"
     xmlns="http://tempuri.org//any"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     attributeFormDefault="qualified"
elementFormDefault="qualified">
    <xs:element name="xml-body">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" minOccurs="0"
maxOccurs="1">
```

```
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="50"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="value" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

סכמה זו מגדירה מבנה הבא:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body>
    <name>Possible Attack with very long string</name>
    <value>More long string</value>
</xml-body>
```

כפי שניתן לראות, אורך של תג name חסום, מחרוזת ארוכה מ-50 תווים לא תעבור אימות מול הסכמה. לאומת זאת, עבור תג value הגבלה דומה לא מוגדרת, שעושה תג זה לפתח להתקפה או דליפת מידע.

מומלץ להגדיר חסמים עבור כל מחרוזות בהן משתמשים בסכמה ע"י חסם הגיוני עבור האפליקציה. בפרמטרים המגיעים לשירות ה-WS חובה להגדיר זאת לרבות בפרמטרים המוחזרים על ידי ה-WS .

4.7 טווחי ערכים

4.7.1 הגבלת טווח הערכים למחרוזת

בדומה להגבלת אורך, ניתן להגדיר סט של ערכים אפשריים עבור מחרוזות. למשל:

```
<?xml version="1.0" ?>
<xs:schema id="xml-body"
targetNamespace="http://tempuri.org//any"
xmlns="http://tempuri.org//any"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="qualified" elementFormDefault="qualified">
    <xs:element name="xml-body">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" minOccurs="0"
maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
```

```
                                <xs:enumeration value="Frodo" />
                                <xs:enumeration value="Sam" />
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="value" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
</xs:schema>
```

סכמה זו מגדירה ערכים חוקיים עבור תג name להיות מחרוזות Frodo ו-Sam.

במקרה זה xml הבא ייחשב לחוקי באימות מול סכמה.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body>
    <name>Frodo</name>
    <value>More long string</value>
</xml-body>
```

לעומת זאת, מבנה xml הבא לא יהיה חוקי.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body>
    <name>Baggins</name>
    <value>More long string</value>
</xml-body>
```

בנוסף להגדרת טווח ערכים ניתן להגדיר תבנית למחרוזות חוקיות בעזרת ביטויים רגולריים שזאת הדרך הנפוצה ביותר להגדרת טווחי ערכים. לדוגמא סכמה הבאה מגדירה מחרוזות חוקיות עבור תג name להיות אוסף של מחרוזות באורך 3 בדיוק שמכילות תווים a, b, c בלבד.

```
<?xml version="1.0" ?>
<xs:schema id="xml-body"
targetNamespace="http://tempuri.org//any"
xmlns="http://tempuri.org//any"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="qualified" elementFormDefault="qualified">
    <xs:element name="xml-body">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" minOccurs="0"
maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:pattern value="[a-c]{3,3}"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
```

```
                <xs:element name="value" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

קובץ xml הבא יהיה חוקי.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body>
    <name>abc</name>
    <value>More long string</value>
</xml-body>
```

לעומת זאת, קובץ הבא לא חוקי.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body>
    <name>abcd</name>
    <value>More long string</value>
</xml-body>
```

חשובה להדגיש שאין לכלול בביטוי הרגולרי את ה-Syntax המסמן תחילת ביטוי רגולרי על ידי הוספת "^" בהתחלה, וסוף ביטוי רגולרי על ידי הוספת "$" בסוף. הגדרות אלה אינן נתמכות בתשתיות תהילה.

חובה לבצע הגבלה זאת עבור קלט המגיע לשירות לרבות עבור תשובות השירות ללקוח.

מידע נוסף על כתיבת ביטויים רגולריים בסכמה ניתן לקרוא בקישורים הבאים:

תחת סעיף F Regular Expressions:

http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html

http://en.wikipedia.org/wiki/Regular_expression

4.7.2 הגבלת טווח למספרים

בדומה להגבלת תחום ערכים חוקיים עבור מחרוזות, מומלץ להגביל ערכים של מספרים המופיעים בקבצי XML, עם כי אין חובה לעשות זאת. לדוגמא, נגביל ערכים של תג להיות מספרים שלמים בטווח עגול מ-0 עד 1024.

```
<?xml version="1.0" ?>
<xs:schema id="xml-body"
targetNamespace="http://tempuri.org//any"
    xmlns="http://tempuri.org//any"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    attributeFormDefault="qualified"
elementFormDefault="qualified">
```

```
    <xs:element name="xml-body">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="intvalue" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="1024"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

במקרה זה XML הבא יתקבל ויעבור ב"ממשק מוגן".

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body targetNamespace="http://tempuri.org//any"
xmlns="http://tempuri.org//any">
    <intvalue>5</intvalue>
</xml-body>
```

כמו כן, נציג XML שלא יעבור.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-body targetNamespace="http://tempuri.org//any"
xmlns="http://tempuri.org//any">
    <intvalue>5000</intvalue>
</xml-body>
```

4.8 כינויים

לסטנדרט XML קיימת הרחבה הנקראת DTD (Document Type Definition). הרחבה זו (הקיימת גם בפורמט HTML) מאפשרת להגדיר סוג של מאקרו (או כינויים, aliases) וגם להגדיר layout עבור מבנה XML. הרחבה זו מאפשרת ליצור קבצי XML אשר פוגעים קשות בביצועים של parser-ים סטנדרטיים. כיום הרחבה DTD כמעט ולא נמצאת בשימוש.

אין להשתמש בהרחבת DTD עקב החשש כי השירות ייחשף למתקפות DoS, ועקב חוסר תמיכה של מערכות תהילה בהרחבה זאת.

4.9 הגבלת גודל קובץ

בדומה לאי הגבלת אורך מחרוזת גם אי הגבלה לגודל קובץ אשר מועלה לשירות יכול להוות פתח למתקפות בצד השרת (דוגמא פשוטה ביותר של DoS יכול להיות העלאת קובץ בגודל של מאות ג'יגות ובכך לסתום את הזיכרון, דיסק קשיח וכו' בצד השרת.

מה שנדרש לבצע זה להוסיף הגבלה לגודל הקובץ לתוך הסכמה (הדוגמא הבאה מציגה מקרה בו ההגבלה הינה ל-4MB):

```
<xs:element name="name" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:base64Binary">
      <xs:maxLength value="4194304"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

בהגבלת גודל הקובץ יש להתחשב בכך שגודל בקשה כוללת נכון להיום ב-Data Power של תהילה מוגבל גם ככה למקסימום 4 מגה בייט, כך שיש להתחשב בהגבלה זאת גם ברמת הסכמה.

4.10 תיעוד

סכמות שירותים הינם קבצים גדולים ועמוסים במידע שלפעמים קשה להתמצא בהם. בזמן כתיבת הסכמה מומלץ לבצע תיעוד לחלקים העיקריים. הנחיית התיעוד מתחלקת לשני חלקים:

4.10.1 תיעוד חובה

4.10.2 תיעוד רשות

אין זה משנה אם הסכמה מיוצרת בצורה אוטומטית או בצורה ידנית, מומלץ בחום לבצע תיעוד של הסכמה.

יש לשים לב שבתוך תיעוד הקבצים לא ימצא קוד או חלקי XML שאינם נדרשים לצורך הפעלת השירות.

4.10.3 תיעוד מומלץ

מומלץ בחום לבצע תיעוד עבור חלקים מהותיים בשירות כגון, הגדרת משתנים המעבירים קבצים, הגדרת אובייקטים מורכבים, העברת טפסים וכו'.

4.10.4 תיעוד נוסף בחשיבות נמוכה יותר

עבור חלקים פחות מהותיים בסכמה לא נדרש לבצע תיעוד, אך חשוב לזכור כי תיעוד של חלקים אלו יכול לעזור בעתיד לעבוד בצורה טובה יותר עם הסכמה.
חלקים שלא חובה לתעד יכולים להיות שדות כללים, הסבר על מבנה הסכמה וכו'.

4.10.5 אופן כתיבת התיעוד

לצורך ביצוע תיעוד בסכמה נשתמש בתגיות xs:annotation ו-xs:documentation. כאשר תגית annotation מציינת התחלת אזור תיעוד ותגית documentation מציינת את שורות התיעוד. לדוגמא:

```
<xs:annotation>
  <xs:documentation xml:lang="en">
  This Schema defines a W3Schools note!
  </xs:documentation>
</xs:annotation>
```

4.11 הפניות לא תקינות בקובץ WSDL

יש לוודא כי ההפניות השונות בקובץ ה-WSDL אינן מכילות נתונים שגויים או נתונים מקומיים כגון:

```
<soap:address location="http://localhost:2966/PageName.asmx" />
```

נדרש לשנות הפניות מקומיות לשם השרת/ נתיב תקין. למשל:

```
<soap:address location="http://ServerName:2966/PageName.asmx" />
```

4.12 הגדרת אובייקטים ב-Namespace לא תקין

יש לוודא שלא קיימים אובייקטים המוגדרים ב-Namespace – **http://tempuri.org**, כגון:

```
<soap12:operation soapAction="http://tempuri.org/temp"
style="document" />
```

נדרש להעביר קישורים אלה ל-Namespace של השירות. את הקישור ניתן לבצע ע"י הוספת המאפיין Namespace. היות ולא כל השירותים מוגדרים ב-DNS, יש להגדיר את ה-Namesapce ע"י כתובת IP. לדוגמא:

```
[WebService(Namespace="http://192.168.0.2/webservicename/")]
public class MyWebService {
    // implementation
```

חשוב לציין שבמידה וקביעת Namespace יוצר בעיות עקב יצירת קבצים כפולים (wsdl ו-wsdl0 למשל) או במידה ונעשה שימוש ב-DLL ההקשחות של תהילה לצורך ביצוע ההקשחות (שם עדיין אין תמיכה ב-Namespace-ים), נכון לעכשיו אין חובה לקבוע Namespace-ים נכונים בסכמה.

4.13 העברת שם משתמש בסיסמה בצורה גלויה

יש לוודא שסיסמאות משתמש לא יעברו בצורה גלויה על גבי פרוטוקול ה-http. ניתן לבדוק מצב זה ע"י סריקת סכמת השירות. שם משתמש וסיסמא גלויים יכולים להראות בסכמה לפי הדוגמא הבאה:

```
<s:element minOccurs="0" maxOccurs="1" name="_user"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="_pass"
type="s:string" />
```

מומלץ להעביר את שם המשתמש והסיסמה לתווך מוצפן מסוג HTTPS או שימוש בהצפנת תעבורה על גבי תשתית SBox. לצורך מימוש דרישה זו יש להתייעץ עם צוות אבטחת מידע.

4.14 בדיקות תקינות המידע המוחזר ללקוח

בנוסף להנחיות להקשחת הקלט המועבר לשירות קיימת חשיבות רבה למידע המוחזר חזרה ללקוח. בדיקת המידע החוזר מהשירות ללקוח הינה שכבת הגנה נוספת הבאה להקטין את הסיכון לזליגת מידע והתקפות שונות כגון הזרקת קוד זדוני. קיימים מצבים בהם עלול להיגרם כשל בשירות ו\או בדרך כך שהמידע שיוחזר שונה ממה שנועד לחזור ולכן שכבה זו תהווה הגנה נוספת, לדוגמה במידה וקיימת באפליקציה פגיעות מסוג SQL Injection, תוקף זדוני עלול לנצלה לרעה בין השאר על ידי חשיפת מידע שלא היה אמור להיחשף אליו ולכן הפלט שחוזר מהשירות אינו הפלט שתוכנן לחזור (לדוגמה חשיפת רשימת משתמשי המערכת ופרטיהם במקום חשיפת פרטים של מבקש הבקשה בלבד). על ידי יישום בדיקות המידע החוזר מהשירות ללקוח ניתן למנוע זליגת מידע של השירות והתקפות שונות.

5. אפיון שדות

בכדי להגדיר ביטויים רגולריים בצורה נכונה ותקינה יש לאפיין את השירות על ידי מסמך אפיון מוסדר. במסמך האפיון יש לפרט ככל שניתן את כל שדות הקלט\פלט של השירות לרבות: שם השדה, תיאור קצר, סוג השדה וכו'. לאחר שלב אפיון השדות ניתן להבין ולהגדיר בהתאם את הביטוי הרגולרי הנדרש, לדוגמה אם קיים לנו שדה בשם "תעודת זהות", אנו יודעים מראש את הנתונים הבאים:

- המחרוזת צריכה להיות מורכבת לא פחות ולא יותר מ -9 תווים.
- המחרוזת צריכה להיות מורכבת ממספרים בלבד.

לכן על בסיס מידע זה נוכל להגדיר את הביטוי הרגולרי הבא:

[0-9]{9}

הסבר הביטוי:

כל תו במחרוזת הינו מ-0 עד-9, הגדרה זו חוזרת על עצמה בדיוק {9} פעמים (אורך המחרוזת). מדוגמה זו ניתן להבין מדוע בשדה מהסוג שצוין ביטוי רגולרי אשר יכיל אותיות ו\או סימנים אינו יהיה הביטוי הנכון להקשחה. חשוב לציין כי יש להגביל את אורך המחרוזת (מינימום אורך ומקסימום אורך) לפי אפיון השדה, ניתן להגביל את אורך המחרוזת עי ידי הביטוי הרגולרי עצמו, ו\או על ידי הגבלות האובייקט ב-XML עצמו.

יש לציין כי קיימות רשימות (כגון רשימת ערים\ארצות) שמהן ניתן להבין מה התווים המותרים ועל סמך זה להגדיר את הביטוי הרצוי והנכון.

6. תווי בקרה בסיסיים

סימונים בסיסיים לתווי הדפוס של ביטויים רגולריים הם

ביטוי	הסבר
.	נקודה. משמש לחפש כל תו שהוא.
^	משמש לסמן את תחילתה של המחרוזת, באם הסימן לא יופיע החיפוש יתבצע בכל מקום עד תחילתה של המחרוזת. *אין לשים תו זה בהקשחות ה-WS היות וביטוי זה מתווסף בצורה אוטומטית על ידי ה- DP.
$	משמש בכדי לסמן את סופה של המחרוזת, באם סימן זה לא יופיע החיפוש יתבצע בכל מקום במחרוזת עד סופה. *אין לשים תו זה בהקשחות ה-WS היות וביטוי זה מתווסף בצורה אוטומטית על ידי ה- DP.
[]	מתאים לכל מה שמוגדר בסוגריים לתו אחד בלבד
[a-z]	משמש לכל תו נמוך באלפבית האנגלי.
[א-ת]	משמש לכל תו באלפבית העברי.
[0-9]	משמש לכל הספרות, לדוגמה הביטוי [5-8] מציין כי התוכנה תחפש ספרות בין 5 ל8.
{}	משמש לסמן את מספר המופעים של תווי הביטוי שמבוקשים. לדוגמה [A-Z]{2} מציין שהתוכנה תחפש שתי אותיות גדולות. א{2,4} מציין שהתוכנה תחפש מחרוזות שמורכבות מהאות אלף בלבד בגדלים של שניים עד ארבע (אא, אאא, אאא).
*	משמש לציין מספר תווים לא מוגבל. למשל הביטוי הבא "*[0-9]" יבדוק האם כל המחרוזת היא ספרתית, מתחילתה ועד סופה.
-	מפריד בכדי לסמן כמה ביטויים. (משמש כמו האופרטור OR) למשל אב|אגג|הדד יגרום לחפש את אחד משלושת המחרוזות הללו.
\	משמש להתייחס לתו בקרה כאילו היה תו רגיל. אם למשל רוצים לחפש את התו $ יש לכתוב \$

לחלק מהתווים יש ביטויים קבועים:

\d	מציין תו ספרתי מ0-9 = [0-9]
\w	מציין תו של אות או ספרה =[a-zA-Z0-9]
\s	מציין רווח כל שהוא = [\r\t\n\f]

אותיות גדולות מציינות את ההיפך.

\D	מציין כל תו שאינו מספרי = [^0-9]
\W	מציין כל תו שאינו אות או ספרה = [^a-zA-Z0-9]
\S	מציין כל תו שאינו רווח = [^\r\t\n\f]

ניתן להגדיר את מספר החזרות של תו מסוים באמצעות כמה דרכים:

*	מציין 0 או יותר הופעות של התו שלפניו
+	מציין 1 או יותר הופעות של התו שלפניו
?	מציין 0 או 1 הופעות של התו שלפניו
{2,4}	מציין טווח של הופעות, כשהמספר הראשון מציין את הכמות המינימאלית והמספר האחרון מציין את הכמות המקסימאלית. ניתן גם להחסיר את אחד המספרים ולציין רק מינימום או מקסימום.

הגבלות אורכים באמצעות הקשחות מחוץ לביטוי הרגולרי על ידי שימוש ב- max\min Length

לדוגמה:

```
<xs:restriction base="xs:string">
    <xs:maxLength value="50" />
</xs:restriction>
```

מידע נוסף ניתן למצוא באינטרנט ובקישור הבא:

https://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx

7. מגבלות

בביטוי רגולרי קיימים תווי בקרה המורים על התחלה וסוף הביטוי, תווים אלה מתווספים אוטומטית על ידי ה- Data Power ולכן אין לשים תווים אלה כחלק מהביטויים הרגולריים בהקשחות השירות. להלן התווים האסורים לשימוש כהתחלה וסוף לביטוי:

^$

8. דוגמאות

שם השדה	סימנים נדרשים	מינימום תווים	מקסימום תווים	Regex
שם פרטי (עברית)	' " - ( ) ורווח	2	25	[א-ת'"\-\(\)\s]{2,25}
שם פרטי (לועזית)	' " - ( ) ורווח	2	25	[a-zA-Z'"\-\(\)\s]{2,25}
שם משפחה (עברית)	' " - ( ) ורווח	2	25	[א-ת'"\-\(\)\s]{2,25}
שם משפחה (לועזית)	' " - ( ) ורווח	2	25	[a-zA-Z'"\-\(\)\s]{2,25}
מספר זהות		9	9	[0-9]{9}
מספר ח.פ (חברה פרטית)		9	10	[0-9]{10}
תאריך לידה		10	10	[0-9/]{10}

*חשוב לציין כי הדוגמאות הינן למטרת עזרה בלבד ואינן המלצות מחייבות.

9. הנחיית מפתחים להטמעת רכיב GovIL WCF Extention

הקשחת הסכמה ניתנת לביצוע בצורה ידנית, אך פתרון זה מצריך עדכון ההקשחה לאחר כל עדכון של סכימת השרות.

GovILWcfExtention הוא רכיב קוד הנותן מענה למרבית דרישות ההקשחה של אוביקטים והוא עובד על עקרון של כתיבת Attributes על DataMemebers של אוביקטים ודרכם "להזריק" את ההקשחה לWSDL בזמן ריצה.

9.1 דרישות קדם

פיתוח בסביבת .Net Framework 3.5 ומעלה.

9.2 סדר פעולות הטמעה

- הוספה של reference בצד של שירות ה-WCF
- הגדרה של Attribute על ה-interface.
- הוספה של ה-Restriction המתאים מבין האפשרויות הבאות:
  - GovIL.WCF.Extensions.DataMemberLength
  - GovIL.WCF.Extensions.DataMemberMaxExclusive
  - GovIL.WCF.Extensions.DataMemberMaxInclusive
  - GovIL.WCF.Extensions.DataMemberMaxLength
  - GovIL.WCF.Extensions.DataMemberMinExclusive
  - GovIL.WCF.Extensions.DataMemberMinInclusive
  - GovIL.WCF.Extensions.DataMemberMinLength
  - GovIL.WCF.Extensions.DataMemberRestrictionPattern
  - GovIL.WCF.Extensions.DataMemberListMaxLength
- חשוב לזכור כי בעזרת Attribute ניתן לבצע Restrictions על Simple Type בלבד כמו string, integer וכו' **ואינם** עובדים על מחלקות Complex Types.
- מחלקה שעליה הופעל Restriction **חייבת** להיות בתוך חתימה של מתודה.

9.3 תיאור מאפיינים

Attribute Name	WSDL Restriction	Description
DataMemberLength	length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
DataMemberMaxExclusive	maxExclusive	Specifies the upper bounds for numeric values (the value must be less (than this value
DataMemberMaxInclusive	maxInclusive	Specifies the upper bounds for numeric values (the value must be less (than or equal to this value
	maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero
DataMemberMaxLength	minExclusive	Specifies the lower bounds for numeric values (the value must be greater (than this value
DataMemberMinExclusive	minInclusive	Specifies the lower bounds for numeric values

		(the value must be greater (than or equal to this value
DataMemberMinInclusive	minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
DataMemberMinLength	Pattern	Defines the exact sequence of characters that are acceptable
DataMemberRestrictionPattern	ListMaxLength	Specifies the maximum number of objects in list items allowed. Must be equal to or greater than zero

9.4 דוגמאות להקשחת השירות

1. הגדרה של Attribute על ה-interface

```
[ServiceContract]
[GovIL.WCF.Extensions.GovILDataContractRestrictionOperationBehavior]

public interface ICustomerService
{
  .
  .
  .
}
```

2. הגדרת ה- Restrictions כ- Attributes על ה- DataMembers

```
[DataContract]
public class ProductWraperListClass
{
    [DataMember]
    [GovIL.WCF.Extensions.DataMemberListMaxLength(30)]

    public List<ProductWraper> ProductWraperList { get; set; }
}

[DataContract]
public class ProductWraper
{
    [DataMember]
    [GovIL.WCF.Extensions.DataMemberMinLength(0)]
    [GovIL.WCF.Extensions.DataMemberMaxLength(200)]
    [GovIL.WCF.Extensions.DataMemberRestrictionPattern("[a-zA-Z0-9א-ת]",
IsNillabel = true)]

    public string Product { get; set; }
}



[DataContract]
public class ProductID
{

    [DataMember]
    public int ProductID { get; set; }
}
```

3. עבודה עם מערך של אוביקטים במקרים שיש צורך לעבוד עם מערכים של simple type נדרש לעטוף אותו ע"י מחלקה. לדוגמא:

9.5 בדיקת תקינות של ספריות

Checksum type	Checksum Value	File Version
MD5	F8825361F035F3B95B7BB6A67844C8CE	3.1.0.0
SHA1	27203014ECE0A26C17719BDEC220B9529C1466A7	
SHA-256	326EE1D75FE4DB1D2A219CA533805ABB8CB293B0E67ADFE1260BE13220324EDF	