stki.info
IT Knowledge Integrators
Moshav Bnei Zion P.O.Box 151, 60910 Israel Tel. 972-9-7907000 Fax. 972-97442444

סיכום מפגש שולחן-עגול

Secure Development

פיתוח מאובטח של מערכות מידע

מנחים:
סיגל רוסין
פיני כהן

stki.info
IT Knowledge Integrators
Moshav Bnei Zion P.O.Box 151, 60910 Israel Tel. 972-9-7907000 Fax. 972-97442444

לקוחות נכבדים שלום,

תודה על השתתפותכם במפגש שולחן עגול Round Table בנושא פיתוח מאובטח של מערכות מידע.

מצ"ב סיכום עקרי הדברים שעלו במהלך המפגש. במפגש עלו נושאים מהותיים שתומצתו בסיכום כפי שעלו. אין בסיכום זה המלצה גורפת ללקוחות אלא מתן פרספקטיבה והצגה של ההתלבטויות שעלו במפגש כלומר "מהשטח".

על פי הנאמר בדיון, תמונת המצב האידאלית בהקשר של פיתוח מערכות מידע בהקשר אבטחה הנה ברורה. בדיון השתתפו מנהלי אבטחת מידע, מנהלי סיסטם ומנהלי פיתוח. גופי אבטחת המידע צריכים להיות מעורבים בכל השלבים של הפרויקט – החל משלב הייזום כולל הערכת העלויות ואישור התקציב, בהמשך בשלבי הניתוח, הפיתוח הבדיקות (בעיקר הלא פונקציונליות) ,כולל סריקת קוד ממוכנת וידנית ולבסוף ביצוע PT. אולם בין הרצוי למצוי המרחק גדול- בייחוד בפרויקטים קטנים או שו"ש (כגון עידכון גרסת תוכנה, עדכון שרת קריטי בארגון ועוד).
אחד הארגונים מסר שהצליח לקיים סדנת התמודדות עם עולם הסייבר בהשתתפות בכירים בארגון עד לרמה של בעלים. לאחר שהמנהלים ראו מה המשמעות של מתקפת סייבר על ארגונם – הייתה הפנמה של הנושא ותקציבים מתאימים הופרשו בהתאם.

בברכה,

סיגל רוסין ופיני כהן

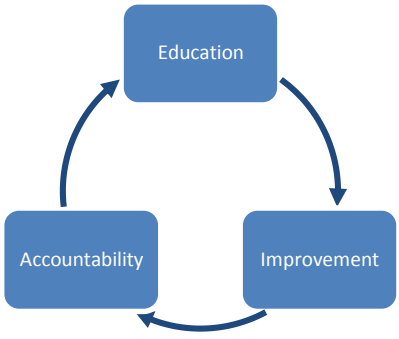
תוכן עניינים

הקדמה

חברת מיקרוסופט אימצה מתודולוגיה שלמה להטמעה, יישום ומימוש פיתוח מאובטח – Software Development Lifecycle (להלן; "SDL"). בהמשך נדבר על מתודולוגית מיקרוסופט תוך הוספת אלמנטים חשובים ממתודולוגיות אחרות ואתרים נוספים הנחשבים כ – Best Practices לנושא זה כגון OWASP וכדומה.

אבני הדרך בהטמעת המתודולוגיה

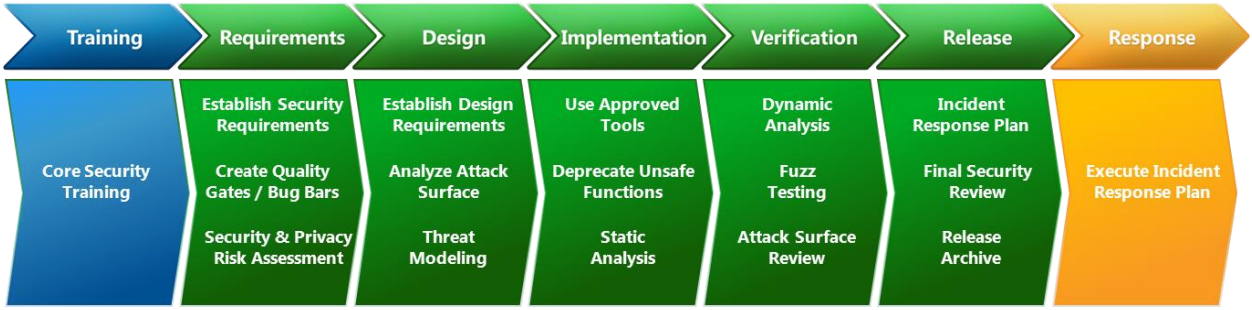
אבני הדרך של מתודולוגית SDL מושתתות על שלוש רגלים:



1. לימוד והעשרה (education)
2. תהליך השתפרות מתמשך (continuous process improvement)

3. נשיאה באחריות (accountability)

המתודולוגיה משתלבת היטב במחזור הפיתוח ומהווה השלמה טבעית לנושא אבטחת המידע. להלן תרשים הממחיש את מרכיב המתודולוגיה בהתאם למחזור הפיתוח.



המרכיבים כאמור חופפים למחזור הפיתוח, ובהטמעת המתודולוגיה מוגדרים שלבים רבים אשר יחד מאפשרים את יישום המתודולוגיה בארגון. בהתאם לדוקטרינה של חברת מיקרוסופט הוגדרו שישה עשר נושאים כנגד חמשת שלבי הפיתוח (Requirements, Design, Implementation, Verification, Release) להגדרת SDL והטמעתו בארגון. אנו נעביר את כל הנושאים המוגדרים לצוות הנבחר בחברת לקוח דוגמא.

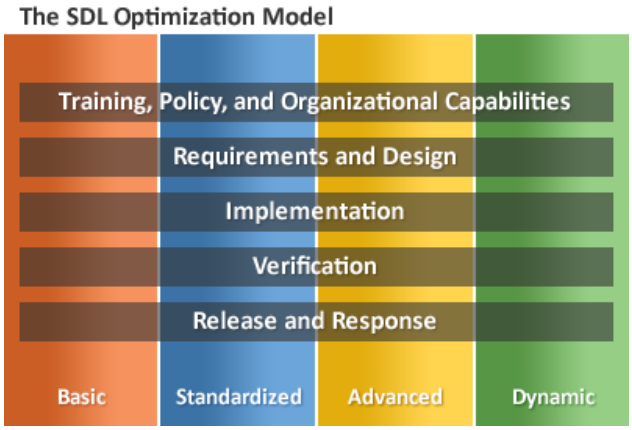
אופטימיזציה

כאמור, מתודולוגית SDL הנה תהליך מורכב וארוך. לשם כך הגדירה מיקרוסופט מודל אופטימיזציה ליישום ומימוש המתודולוגיה. השיטה הנהוגה הינה לעבור בכל פעם שלב נוסף ברמת הידע והיישום וזאת לאחר שהשלב הנוכחי מוטמע ומתקיים באופן שוטף בצוות הפיתוח. למתודולוגיה חמישה נושאים המשקפים יכולות צוותיות:

- Training, policy, and organizational capabilities
- Requirements and design
- Implementation
- Verification
- Release and response

הנושאים הללו מושתתים על ארבע רמות של הטמעה ארגונית. לכל רמה מוגדרים נושאים ללימוד ויישום בצוות הפיתוח.

על פי המתודולוגיה אנו נעבור עם לקוח דוגמא לאורך הרמות ונוודא מעבר ויישום בין רמה לרמה. לשם כך, אנו מציעים גם מרכיב של ליווי שבועי הכולל טיפול, ליווי והדרכה בפרויקטים נבחרים בהתאם לדרישת לקוח דוגמא. יש לציין שהיקף הפרויקט מתייחס לכל שלבים, מהבסיסי עד המתקדם.



להלן טבלה המשקפת את הפעילויות השונות ואשר מיושמות בכל רמה ורמה.

	Basic	Standardized (Activities are expert led)	Advanced (Activities are led by central security team)	Dynamic (Activities are co-led by product teams and central security team)
Training, Policy, and Organizational Capabilities	• Setting baseline and goals for SDL	• Executive support: Tacit • Enterprise coverage: Few SDL pilot projects • Training: Basic concepts • Basic security bug tracking	• Executive support: Explicit • Enterprise coverage: New, high-risk projects • Training: Common baseline • Central security team exists	• Executive support: SDL mandated and enforced • Enterprise coverage: All projects with meaningful risk • Training: Custom training • SDL is adapted to the development methodology
Requirements and Design	• Undefined or inconsistent	• Risk assessment • Threat models: Piloted for high-risk modules	• Functional requirements for security and privacy • Standard security solutions • Threat models: Created with expert assistance	• Threat models: Independently created by product groups
Implementation	• Undefined or inconsistent	• Compiler defenses • Banned functions • Cross-site scripting and SQL injection defenses	• Static analysis tools	• In-house, product-specific security tools development and customization
Verification	• Undefined or inconsistent	• File fuzzing • Basic Web application scanning • Penetration testing by third parties as appropriate	• Comprehensive fuzzing and Web application scanning • Threat model validated	• In-house development and customization of tools to: ◦ Detect vulnerabilities ◦ Audit compliance with SDL
Release and Response	• Undefined or inconsistent	• Final Security Review: Verify internal and external compliance • Project archiving • Response: Basic	• Response plan in place, incident tracking • Basic root-cause analysis	• Real-time incident tracking • Advanced root-cause analysis and formal feedback into policy

- חשוב לציין כי המתודולוגיה לקוחה מחברת ייעוץ בשם GRSEE המתמחה בתהליכי פיתוח מאובטח בארגונים. איש קשר: בן בן אדרת, מנכ"ל 052.3866591

גבולות גזרה בפרויקט

בדיון עלתה סוגיה חדשה יחסית והיא "גבולות הגזרה" של תחום אבטחת המידע. באופן מסורתי תחום אבטחת המידע אחראי בין היתר על נושאים של מניעת פריצות לארגון לצורך ביצוע גניבת כסף או מידע, מניעת פריצות לצורך ונדליזם ("מחיקת פרטי לקוחות", השחתת אתר), מניעת מתן שירות (DDOS) ועוד.

אולם לאחרונה שומעים קולות המכללים תחת התחום של "אבטחת מידע" נושאים של אמינות מידע ולפיכך טוענים שאמינותו של תהליך עסקי והנתונים הקשורים אליו גם הם קשורים לאבטחת מידע. לדוגמה לקוח מקליד כתובת "הנורית 7" אבל בגלל טעות בתכנות נשמר במסד הנתונים "הנורית 8". ארגונים (מעטים) אשר נוקטים בגישה זו מחייבים שכל תחום הבדיקות יהיה כפוף וינהג לפי דרישות וכללים של ארגון אבטחת המידע.
אולם רוב הארגונים נוקטים בגישה שנושאי אבטחת המידע בהקשר של פיתוח קשורים רק לבדיקות הלא פונקציונליות ולכן לא מטפלים בסוגיה כפי שהוצגה למעלה.
ארגונים מתקדמים מבצעים גם סקר קוד בהקשר אבטחת מידע ומנחים את המפתחים בהתאם לכתיבה ובדיקה של הקוד. אם ישנו שדה קלט ארוך הדבר בעייתי מבחינת אבטחת מידע כי ניתן בשדה כזה להכניס קלט בעייתי שהוא פתח לפגע כגון sql injection . אולם אם אנשי אבטחת המידע מגלים שבשדה תעודת זהות יש 2 תווים מיותרים, למרות שהדבר בעייתי מבחינה פונקציונלית (יהיה קושי להשוות לתעודת זהות במערכות אחרות) אין הדבר בעייתי מבחינת אבטחת מידע ולכן לא תתבצע פניה למפתחים.

עם זאת על פי ניסיונם של ארגונים רבים, המקומות שבהם יש יותר באגים בתוכנה הם גם היותר חשופים לפגעי אבטחת מידע ולכן ישנם ארגונים אשר צוות אבטחת המידע מקבל את תוצאות הבדיקות הלא פונקציונליות וגם הפונקציונליות שמתקבלות מצוות הבדיקות. לקוחות ציינו כי עולם הפיתוח מאובטח התחזק בעקבות פיתוח אפליקציות במובייל ושימוש בשפות תוכנה מתקדמות.

עמידה בתקנים ורגולציה

לקוחות ציינו שכחלק מעמידה בתקנים בינלאומיים כמו ISO 270001 ישנו שינוי ושיפור בתהליכי הפיתוח בהקשר של אבטחת מידע. בהמשך לאותו התקן נדרשים מפתחי הארגון לכתיבה מאובטחת, נהלים מסודרים בנושא ואף מתודולוגיה.

פה קבור הכלב

ישנה שונות גדולה בין הארגונים השונים בהקשר לתהליכים ונהלים של פיתוח מאובטח אולם בדיון הייתה תמימות דעים כי רובם המכריע של הפרויקטים הגדולים מטופלים בצורה סבירה בהיבט של פיתוח מאובטח. אולם הבעיה בעיקר היא בפרויקטים הקטנים ובמיוחד במשימות

אחזקה קטנות מה שמוגדר במקרים רבים כ-שו"ש. בארגוני enterprise בישראל יש מאות משימות קטנות ברבעון, כאלה שאינן עוברות תהליך פיתוח מלא כי הרי מדובר ב"שינוי קטן" והתוצאה בהקשר של פיתוח מאובטח עלולה להיות הרת אסון. דוגמאות כגון, עדכון קושחה, עדכון תוכנה לשרת קריטי, ממשק ארגוני היוצא החוצה, ועוד.
בתור דוגמה תאר לקוח מצב שבו בארגון המשתמש ב- IE8 בתור דפדפן ארגוני סטנדרטי עלתה יוזמה באחד מהפרויקטים הפנימיים להשתמש בפרויקט זה ב- chrome בכדי ליישם פונקציונליות מתקדמת של java script. המפתחים (והמנהלים) בפרויקט התייחסו לשימוש ב-chrome בתור דבר שולי ורצו להעלות את הפרויקט לאוויר. אולם הסתבר שאבטחת מידע לא אישרה שימוש ב- chrome (עקב חוסר היכולת כיום לנהל את הדפדפן באופן מרוכז). כלומר מדובר על נושא שמבחינת המפתחים נראה פעוט אבל מבחינת אבטחת מידע אינו טריוויאלי בכלל ולכן נוצר עיקוב בפרויקט.
חשוב לציין כי הכל תלוי בניהול סיכונים שנעשה עוד בתחילת פרויקט. כוח האדם בתחום האבטחה מצומצם וקשה להקצות לכל פרויקט אדם שיישב וינחה את נושא האבטחה. לכן יש לייעל תהליכים ולפחות בפרויקטים מורכבים ומשמעותיים יש לעבור דרך אבטחת המידע בארגון.

בכדי להתמודד עם הסוגיה זיהוי הפרויקטים והשינויים המחייבים עבודה צמודה מול אבטחת המידע יצר אחד הארגונים template של שאלות למפתח לגבי מהות השינוי (לדוגמא, האם השינוי כולל שינוי בהרשאות או משתמשים, האם השינוי כולל עדכון הדרך שבה נגשים לנתונים וכד') ולפי המענה על השאלות מחליטים האם יש לערב את אבטחת המידע ובאיזה שלבים בפיתוח. שאלון זה אמור להיות חלק בסיסי מייזום כל פרויקט\שינוי.

כלים ל- static code analysis בהקשר אבטחת מידע

בדיון עלה הצורך בכלים שמבצעים static code analysis כגון checkmarks, c-care ורבים אחרים (חלקם קוד פתוח). לדעת לקוחות הכלים מסייעים בהעלאת רמת האבטחה בהקשר של פיתוח אולם הדעה הרווחת היא שכלים אלו לא מחליפים את הצורך בביצוע PT.
אחד הארגונים תאר מצב שבו משתמשים ב- CCARE כאשר הפלט שמתקבל הנו טוב וברור, ממוין לפי קריטיות וכולל המלצות.

אולם כלים מתחום זה אינם מכסים קוד LEGACY (MF AS400 ומערכות שנות) כמו גם חבילות עליהם מבוססים רובם המכריע של ארגוני ה- enterprise ולכן המקום הפוטנציאלי שבו אפשר להשתמש בכלים אלו אינו גדול.
בנוסף, ישנם כלים ותיקים בתחום כמו HP FORTIFY ויבמ APPSCAN. מתחרה נוסף שעלה בשנים האחרונות הוא SEEKER התומך גם בבדיקות דינמיות.

ישנם לקוחות שציינו כי הם נעזרים בשירותי חברות ייעוץ הבודקות את הקוד ולא במוצר. חברות ייעוץ שעלו בנושא הם קומסק, אבנט, Beyond security.

תקציב אבטחת מידע (בהקשר פיתוח)

בנושא של תקציב אבטחת המידע ו"גילגולו" על הפרויקטים התגלתה שונות רבה בין הארגונים. על פי הנראה, רובם המכריע של הארגונים אינם מקצים מראש פר פרויקט לנושא אבטחת מידע ורבים גם אינם מעמיסים הוצאות בתחום זה (בדרך כלל בדיקות PT – סדר גודל עלות של K20 ₪ לבדיקה, אך ביצוע PT לפרויקטיי WEB המתבססים על טכנולוגיות REST עלול לעלות יותר) על הפרויקט.
ארגונים שכן מקצים תקציב פר פרויקט דיברו על סדר גודל של 5% מתקציב הפרויקט.
בנוסף, כמה לקוחות ציינו כי לתחום הפיתוח מאובטח אין נהלים או מתודולוגיה. הארגון נתקל במצב בו אבטחת המידע מעכבת פרויקט היות והיא לא נכללה עוד בשלב הייזום. כתוצאה מכך, תקציב הפרויקט גדל.

הכשרות מפתחים ובודקים בנוגע לאבטחת מידע

בדיון עלה הנושא של הכשרות מפתחים בהקשר אבטחת מידע. מסתבר שבהחלט ישנה מודעות לחשיבות לימוד ותזכור הנושא בקרב המפתחים. ישנם ארגונים אשר מכשירים מפתחים כחלק מכניסתם לתפקיד. ישנם ארגונים אשר כחלק מעמידה בתנאי PCI מחויבים לתזכר את נושא הפיתוח המאובטח למפתחים בתדירות של לפחות שעת הדרכה בשנה אולם לדברי ארגונים מדובר בכמות שלא תמיד מספיקה. כנ"ל לגבי בודקי תוכנה בחלק מהארגונים. בארגונים אלו בודקי התוכנה מבצעים בדיקה בסיסית בהקשר של אבטחת מידע (אבל לא מדובר על החלפת תהליך של ביצוע PT חיצוני).

אחד הלקוחות שיתף כי פעם בשנה נערך קורס פיתוח מאובטח למפתחים החדשים. כל רבעון ישנה הרצאה כללית על פיתוח מאובטח. חשוב להעלות את המודעות בנושא כמה שיותר. ישנו

קשר ישיר לתרבות הארגונית- ארגון מסורתי לעומת ארגון אינטרנט. רוב הארגונים לא עושים CHARGEBACK לאבטחה. הביזנס לא מוכן לשלם על אבטחה, ולכן מנהל IT מקדם זאת כך שנושא האבטחה אחרון. בעולם האבטחה כל הזמן נמדדים ויש להוכיח ROI על מוצרים שקונים. לכן עולם ניהול הסיכונים וGOVERNANCE משתלב חזק כאן.

קוד פתוח בהקשר של אבטחת מידע

בדיון עלתה סוגיה שאינה קשורה באופן ישיר לתחום הפיתוח והיא השימוש בקוד פתוח שאינו מסחרי בארגונים. השימוש בקוד פתוח רווח יותר ויותר בעיקר בחברות אינטרנט ובסטארטאפים וכעת מתחילים גם ארגוני enterprise ליישם קוד פתוח שאינו מסחרי. מצד אחד מכיוון שמדובר על קוד פתוח יותר עיניים רואות את הקוד ולכן במידה ויש חלקי קוד בעייתיים מבחינת אבטחת מידע הסיכוי שיתגלו יותר גבוה. אולם מצד שני ישנם יותר גורמים אשר עלולים להזין קוד בעייתי – לעומת חברות מסחריות שברמה הבסיסית מעוניינות לוודא שהקוד מאובטח (אולם ישנו כמובן סיטואציה שבה גורמים "רעים" חדרו לתוך החברה המסחרית).

ואכן, אחד הארגונים תאר מצב שבו הוא משתמש בקוד פתוח שאינו מסחרי אולם עקב עובדה זו יצרו בארגון מעטפות אבטחה נוספות לוודא שקוד זה לא יזיק.

נספח מיוחד התייחסות ספקים ויצרנים לנאמר במפגש

התייחסות חברת HP

איש קשר: עמי סוויסה, מכירות 0524265310 Ami.Suissa@hp.com

אנו רואים בחשיבות רבה את כלי העזר לפיתוח ומאובטח וגאים במוצר Fortify מבית HP המוביל בתחום.

Fortify כמו גם כלים אחרים בשוק משתלבים בתוך תהליך ה פיתוח המאובטח בארגון SDLC.

כפי שהוזכר לרוב משתמשים ב static code analysis שמאפשר מציאת פגיעויות ברמה הכי גבוהה, אך גם Dynamic code analysis בשימוש נרחב אצל לקוחותינו.

- HP גם מספקת שירותי הדרכה למפתחים בנושא פיתוח מאובטח שהוזכרו בשולחן העגול כך שכאשר מגיע עובד חדש, הוא מקבל לינק ומייד לומד את הפיתוח המאובטח.

יש לציין ש HP מספקת גם שירותי **בדיקת קוד בענן** Fortify on demand - שירות שצובר תאוצה בעולם ומספק תוצרים תוך יום עבודה ולא מצריך ידע מצד הלקוח.
מעבר לכך, חשיבות נוספת היא לגוף המחקר שעומד מאחורי החברה, כיוון שהאיומים גוברים כל יום ויש לדאוג שתוכנת הסריקה תדע להתמודד עימם, ל HP מעל 2000 אנשי מחקר.

העתיד נמצא גם בפרואקטיביות וחסימה של חורי אבטחה אפליקטיבים באופן יזום, מוצר ה Application Defender מאפשר להגן ברמה אפליקטיבית על אפליקציות שרצות בסביבת Production וגם אם נמצא חור אבטחה, אין צורך להחזיר את הקוד למפתחים באופן מיידי, Application Defender יודעת לחסום את הפגיעויות וכמובן לדווח על כך.

נשמח לעמוד לרשותכם באם יש שאלות נוספות