Digital Whisper Technologic Papers

בינה מלאכותית והגנת סייבר: הייפ ומציאות - חלק ב'

מאת ד"ר יעקב רימר

הקדמה

בשנים האחרונות נוצר הייפ גדול סביב השינוי מהקצה עד הקצה שיביאו לעולם הגנת הסייבר שיטות של בינה מלאכותית (**A**rtificial **I**ntelligence - AI) בכלל ולמידת מכונה (**M**achine **L**earning - ML) בפרט. בסדרת מאמרים זו אנסה לבחון לעומק את הפוטנציאל של שיטות מתקדמות בלמידת מכונה לקידום משמעותי של יישומים שונים בתחום הגנת הסייבר.

קיימות מספר שיטות לחלק את נושאי הגנת הסייבר. אני אנקוט בחלוקה אותה בחרו כותבי המאמר של CSET (קיצור של Center for Security and Emerging Technology) שאני סוקר ואחלק את נושאי הגנת הסייבר שאעסוק בהם לארבע קבוצות: פעולות מוכנות וחוסן, יישומי ניטור, פעולות תגובה והתאוששות, והגנה אקטיבית.

במאמר הראשון הגדרתי את מסגרת הדיון והצגתי דיון לדוגמא שעסק בתרומה האפשרית של למידת מכונה לבדיקות חדירות, רכיב חשוב בשלב המוכנות והחוסן של המערכות. במאמר הנוכחי אמשיך בשני דיונים פרטניים נוספים. נתחיל בדיון ביכולת של למידת מכונה לאתר פוגענים על ידי מוצרי AV ו-EDR כחלק מתהליך הניטור. לאחר מכן נדון ביישומים לטובת כתיבה של קוד מאובטח, נציגים נוספים של שלב המוכנות והחוסן.

למידת מכונה לטובת איתור פוגענים (Malwares)

במאמר הקודם ציינתי כי כותבי המאמר של CSET סוקרים את ההיסטוריה של אינטליגנציה מלאכותית עבור שלושה יישומים שונים של הגנת סייבר. נתחיל הפעם בסקירה של מוצרי AV (Anti-Virus). חברות AV הוקמו בסוף שנות ה-80 של המאה הקודמת. החל משנת 1996 חברת IBM החלה לחקור יישומי למידת מכונה לטובת איתור פוגענים. זאת מכיוון שכבר בשלב מוקדם זה מפתחי פוגענים הבינו שניתן לעקוף די בקלות את מנגנוני החתימות ששימשו את ה-AV בראשית דרכם.

כל מה שצריך לעשות הוא לשנות במעט את הקוד של הפוגען (מכונה פולימורפיזם או מטמורפיזם) ולייצר וריאנטים שונים שלו. די דומה לווריאנטים של וירוס הקורונה שנוצרים בתהליך האבולוציה שלו, עליהם אנחנו שומעים הרבה לאחרונה. יצור של וריאנטים שונים שיבש את יעילות מנגנוני החתימות הבסיסיים. זאת מכיוון שהם היו מסוגלים לחפש רק קבצים זהים לאלו שהם כבר זיהו בעבר.

לכאורה, למידת מכונה אמורה לשפר מאוד את מצב המגינים בתחום הזה כי שיטות שונות של למידת מכונה מתמחות בזיהוי של תבניות שנשארו קבועות בפוגענים, למרות השינוי שנעשה בקוד. המחקר הראשוני של חברת IBM ניסה להשתמש ברשתות נוירונים כדי לנסות לזהות בוטקיטים (Bootkits). במהלך העשור הראשון של שנות ה-2000 גורמים נוספים הצטרפו לניסיונות מחקר לזהות פוגענים בשימוש בשיטות למידה מכונה נוספות. ובעשור האחרון הצטרפו אליהם כמובן רבים אחרים בניסיונות להשתמש מ"מלכת הכיתה" הנוכחית של עולם הלמידה - למידה עמוקה.

דרך אגב, לפחות על סמך סקר הספרות של אנשי CSET, לא ניכר שבהקשר הזה יש יתרון משמעותי ללמידה עמוקה על פני שיטות למידת מכונה ותיקות יותר.

ברם, מנגנוני החתימות מהווים גם **כיום** (2022) את עמוד השדרה של מוצרי ה-AV. ככל שהתוקפים השתפרו, גם מנגנוני החתימות נעשו מתוחכמים יותר. זאת במחיר של עליה משמעותית במורכבות של מנגנון החתימות ובמשאבי זמן הריצה והזיכרון שהוא דורש. על רגל אחת, במקום לחפש חתימה של הקובץ כולו, המנגנונים המודרניים מחפשים חתימות של חלקים של הקובץ, או חתימות של תבניות התנהגותיות שלו (מה הפוגען עושה). בין אם באופן סטטי, למשל על ידי ניתוח של קריאות ה-API שלו, או באופן דינאמי, למשל בשימוש ב-Sandbox.

מדוע אם כן, למרות שהמחקר של ML לטובת איתור פוגענים חגג כבר 25 אביבים, יש עדיין שימוש נרחב במנגנוני חתימות? זה המקום לנתח באופן מעמיק את הקשיים בשימוש במערכות למידת מכונה בעולם האמיתי. אחת הבעיות המרכזיות בעולם הגנת הסייבר בכלל ובעולם איתור הפוגענים בפרט היא בעיית התרעות השווא, או בשפה המקצועית - False Positive.

גילוי של פוגען במערכת ארגונית היא בעיה מהסוג של מציאת מחט בערמת שחת. אלגוריתמי למידת מכונה מתקשים מאוד לפתור בעיות של מחט בערמת שחת. המעוניינים בהסבר פשוט מדוע מוזמנים לקרוא בלינק הזה שפרסמתי בדה-מרקר.

כותבי המאמר של CSET ממחישים את הבעיה באמצעות אנקדוטה מפורסמת משנת 2019. חברת Cylance שיווקה בעבר מוצר AV מבוסס למידת מכונה אך קבוצת האקרים לבנים גילתה דרך פשוטה מאוד לעקוף אותו. כל מה שצריך היה להוסיף לפוגען קטע באורך של כ-5MB מתוך המשחק המפורסם Rocket League. במקרה כזה, ב-90% מהמקרים ה-AV יזכה את הפוגען. ההשערה היא שזה קרה מכיוון שמודל למידת המכונה שהחברה פיתחה זיהה בטעות מספר משחקי מחשב כפוגענים (=התרעות שווא).

כמענה לכך, Cylance הוסיפה מנגנון שני שבחן את מידת הדמיון של הקובץ הנבדק לרשימה לבנה (white-list) של קבצי משחקים.

החלטת המנגנון השני גברה על הפסיקה של המודל הראשון. לכן כל מה שצריך לעשות הוא ל"הדביק" על הפוגען חתיכה מתוך משחק מחשב, והנה הוא צח כשלג. ודרך אגב, הכותבים מציינים שאסור לשכוח שלאלגוריתמי ML יש גם פגיעויות מבניות משלהם (Adversarial AI) שמוסיפים משטחי תקיפה חדשים (על כך בפעם אחרת).

אני אוסיף שמעבר לכך שמדובר בסוגיית מחט בערמת שחת, זו טעות נפוצה להתייחס לפוגענים כאל ישות הומוגנית אחת. יש כידוע סוגים שונים של פוגענים, שלפעמים מתנהגים בצורה הפוכה לחלוטין. לדוגמה, סוסט"רים (Trojans) ישאפו להיות חשאים ככל הניתן, בעוד כופרות (Ransomwares) יכריזו על הגעתן בקול רעש גדול וצורם. יש כמובן גם דוגמאות אחרות.

כיוון שמדובר בקבוצה הטרוגנית תהליך הלמידה צריך לאתר קבוצות שונות (ולפעמים זרות) של התנהגויות, מה שאומר שלכל קבוצה יתווספו התרעות שווא משל עצמה. כלומר, חברות ה-AV פשוט אינן יכולות לאפשר לעצמן כמות כל כך גדולה של התרעות שווא ולגרום בפועל, במו ידיהן, ל-DoS ללקוחות שלהן. הלקוחות פשוט יסירו מיד את מוצרי ה-AV.

אין בדברים הללו לקבוע שאין עתיד למחקר ML עבור גילוי פוגענים. מזה כעשור, חברות ה-AV מעבירות את מרכז הכובד של השיטות שלהן מתחנות הקצה (כגון PC ומובייל) אל הענן שלהם. זה התחיל עם ההופעה בשוק של Cloud AV וצובר תאוצה בשנים האחרונות עם הבשורה של ה-EDR-ים (Endpoint Detection and Response). המעבר לענן מאפשר מחקר ML איכותי יותר, לא מדובר רק בהוספת כוח חישוב משמעותי. יש לכך מספר סיבות. ראשית, עצם העובדה שחברת ה-AV מקבלת בענן תמונה רחבה שמגיעה מלקוחות רבים מאפשרת להצליב מידע ולשפר מאוד את ביצועי השיטות.

בנוסף, ה-EDR agent יכול (בהנחיית השרת שלו) להעמיק את החקירה באופן שקט, בלי להטריד את המשתמש, עד לנקודה שבה הענן יקבל החלטה ודאית על איתור פוגען. בשלב הזה ניתן לעצור את הפוגען מיידית אצל כל לקוחות ה-EDR האחרים. ראו לדוגמא את הפרסום של חברת מיקרוסופט משנת 2017 אודות Detonating a bad rabbit.

אבל מצד שני צריך להודות ביושר שנכון לעכשיו הפוגענים עדיין איתנו והתקפות כופרה רק מתגברות ומתפשטות. יתכן שכמו שנאמר בחלק א', שיטות ML הכרחיות כדי שנוכל "לרוץ הכי מהר" ולפחות "להישאר במקום".

במאמר הקודם נגעתי גם בסוגיה של שיטות מדידת הצלחה עבור טכנולוגית למידת מכונה או מוצר הגנת סייבר ספציפיים. טענתי שזאת אינה סוגיה פשוטה כלל וכלל ופעמים רבות חברות שרוצות להגן על עצמן נאלצות לרכוש "אבקה נגד פילים". עוד ציינתי, כי אפילו עבור AV, שנחשב לאחד ממוצרי ההגנה הבסיסיים

ביותר, קיימים אתגרים משמעותיים לבניית בדיקה מקצועית. הגיע הזמן להסביר מדוע. כדי לערוך את הבדיקה נדרש מאגר של קבצים זדוניים. ראשית, נדרש לתכנן טוב את סוגי ה-Malware שנבדוק.

כאמור, התנהגות של תולעת, סוס טרויאני, כופרה וכו' שונה למדי. ומהיכן נביא אותם? מצד אחד, אם נאסוף קבצים זדוניים "טריים" שהועלו ל-Virus Total בימים האחרונים, יש סיכון שרבים מהם יהיו וריאנטים של אותו פוגען בדיוק.

הבדיקה שלנו לא תקיף מגוון מספיק רחב של איומים. מצד שני, קצב התפתחות של מרוץ החימוש בין התוקף למגן (אבולוציית הפוגענים) הוא מהיר מאוד. מאגר וירוסים שנבנה לפני כשנה כנראה לא יהיה מספיק רלוונטי. שוב נדרשים חשיבה ותכנון.

אסור לשכוח שצריך לאסוף גם מאגר של קבצים תמימים כדי לבדוק את כמות התרעות השווא (False Positive) של המוצר. כאמור לעיל, זה פרמטר לא פחות חשוב מאשר איכות הגילוי של פוגענים. ומהיכן נביא הרבה קבצים תמימים? טעות שכיחה של תכנון לקוי היא לאסוף דוגמאות "תמימות" מכל הבא ליד, ללא חשיבה על ההשלכות. אם למשל נאסוף קבצי הרצה מכמה מחשבים "בסביבה", רובם המוחלט יהיה שייך לדוגמא לחברת מייקרוסופט (קבצי מערכת ההפעלה Windows, תוכנות אופיס וכדומה).

ואם נחזור ל-VT ונאסוף קבצים שהוגדרו נקיים בתקופה האחרונה, אולי הם רק נראים "תמימים" בשל העובדה שמוצרי ה-AV ב-VT עדיין לא עלו על הזדוניות שלהם? הרי אנחנו מתכוונים לבחון איכות של AV אחד, על סמך ביצועים של AV-ים אחרים. יש עוד אתגרים נוספים, אבל נסתפק באלו לעת עתה.

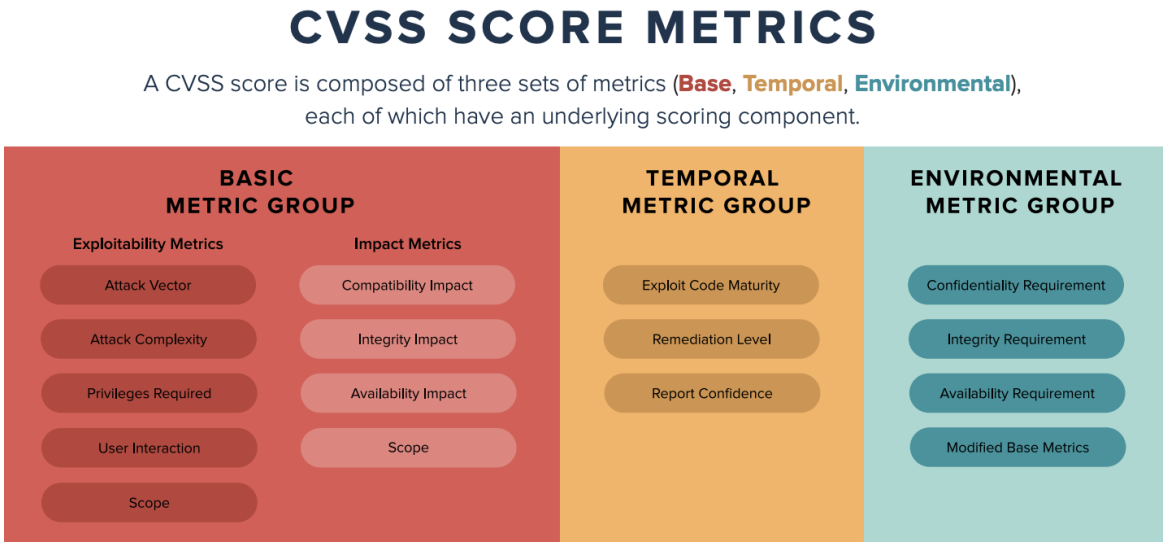
למידת מכונה לטובת כתיבת קוד מאובטח

השלב הראשון בהגנת סייבר אפקטיבית הוא מוכנות וחוסן. תמיד רצוי לנסות למנוע או לצמצם אפשרות לתקיפה, מאשר לרדוף אחריה. יש מספר היבטים למוכנות וחוסן ואחד מהם הוא כתיבת קוד מאובטח ודיבוג שלו. כמעט כל קוד מכיל באגים וחולשות (לא הקוד שלי כמובן ☺). בהמשך אעסוק באוטומציה לגילוי חולשות, כעת אדון בנושא פשוט יותר, סיווג רמת החומרה של באגים. כל מפתח בחברה רצינית מכיר את הישיבות המייגעות בהם דנים עם אנשי ה-QA והמוצר בבאגים שנמצאו ומנסים להחליט את מי מהם חובה לתקן עכשיו ומי יישארו פתוחים ויתוקנו "לגרסה הבאה" (כלומר, אף פעם).

כותבי המאמר של CSET דנים ביכולת לנתח באופן אוטומטי דיווחי שגיאות (bug reports) ולהחליט על רמת החומרה של הבאג והאם הוא עלול להוות חולשת אבטחה שניתנת לניצול. קיימים מחקרים אקדמיים בתחום כבר כ-15 שנה, רובם ללא תוצאות שמאפשרות שימוש מעשי. כפי שציינתי כבר במאמר הקודם, לצערנו זה דבר שכיח במאמרים אקדמיים של למידת מכונה בעולם הגנת הסייבר. החוקרים מגיעים לתוצאות טובות מאוד על מאגרי ניסוי אקדמיים קטנים, המאמרים זוכים בפרסי הצטיינות בכנסים, אבל מימוש מעשי שלהם גורם למפח נפש (לאלו שמנסים).

לעומת מאמרים אלו, חברת מיקרוסופט פרסמה לפני כשנתיים מחקר שהראה שלמידת מכונה מסוגלת לסווג בהצלחה אלו באגים רלוונטיים לבעיית אבטחה ואלו לא. המודל שלהם אומן על מעל מיליון (!) דיווחי באגים אמיתיים של חברת מיקרוסופט. כלומר, המחקר הזה מעיד שחברת תוכנה עם גישה לכמות גדולה מאוד של דיווחי באגים עשויה להצליח לסווג אותן באופן אוטומטי. מצד שני, ציניקנים יאמרו שרק לחברה כמו מיקרוסופט יש מעל מיליון באגים. כל השאר ימשיכו לריב בפגישות באגים, בדיוק כמו היום. כנראה שנצטרך להמתין לפרסום המוצלח הבא כדי להחליט.

גם כאשר באג כבר מסווג כאיום אבטחה (חולשה) אפשרי, עדיין צריך להגדיר את חומרת החולשה ואת מידת הסבירות לנצל אותה. המנגנון הסטנדרטי המקובל כיום נקרא ה-CVSS (קיצור של Common Vulnerability Scoring System). במנגנון הזה מומחים מנתחים וקובעים ציון לחומרת החולשה. גם בתחום הזה יש מחקרים אקדמיים שמראים שבאמצעות למידת מכונה וניתוח של תיאור החולשה באמצעות עיבוד שפה טבעית (NLP) ניתן לקבוע באופן אוטומטי את ציון ה-CVSS.



[מקור: https://www.balbix.com/insights/understanding-cvss-scores]

חוקרים אחרים השתמשו בנתונים שנאספו אודות תקיפות אמיתיות כדי לבנות מודל חיזוי של הסיכוי שהחולשה תנוצל ע"י תוקפים.

לאור התפתחות מהירה של תחום ניתוח הטקסטים בשנים האחרונות, המחקרים האלו מצביעים על כיוון אופטימי ויתכן שנראה בעתיד שימוש אפקטיבי של בינה מלאכותית בתחום הזה. אבל למרות הנימה האופטימית, כותבי המאמר מציינים כי מדובר במשימה בעלת חשיבות בינונית להגנת סייבר והסיכוי שלמידת מכונה תביא לקפיצת דרך משמעותית על פני התהליך הקיים היא בינונית-נמוכה. אני אוסיף, שגם אם הם טועים בהערכות שלהם, פריצת דרך בתחום זה כשלעצמו לא תביא למהפכה משמעותית בהגנת הסייבר.

סיכום

המאמר הנוכחי התמקד בשני נושאים בעולם הגנת הסייבר. הראשון הוא היכולת לזהות פוגענים באמצעות AV או EDR, תחום משמעותי מאוד ונציג ראשון בסדרה של יכולות הניטור. התחום השני הוא היכולת לסייע בשני תהליכי משנה של עולם כתיבת הקוד המאובטח. זהו נציג נוסף של שלב המוכנות והחוסן, בהמשך לעיסוק שלי במאמר הקודם בבדיקות חדירות. במאמר הבא אעמיק (בלי נדר) בתחום האוטומציה לגילוי חולשות ובתחומים נוספים של שלב הניטור.

על הכותב

ד"ר יעקב רימר הוא יועץ בכיר ומרצה בנושאי סייבר, בינה מלאכותית וביולוגיה. יש לו תואר שני בלמידת מכונה ודוקטורט באימונולוגיה, שניהם ממכון ויצמן למדע. הוא עוסק במחקר מדעי באקדמיה במקביל ליעוץ במשרדי ממשלה ולחברות היי-טק. בעבר שימש בתפקידים בכירים בהיי-טק ובמשרד ראש הממשלה. בנוסף, הוא מלמד באוניברסיטת תל-אביב את הקורסים "בינה מלאכותית ויישומיה לביטחון" ו"בינה מלאכותית בעידן הסייבר" במסגרת תוכניות לתואר שני.

קישור ללינקדאין. מייל לתגובות: MrBigDataThemarker@gmail.com

מקורות לקריאה נוספת

Machine Learning and Cybersecurity - Hype and Reality. Micah Musser and Ashton Garriott. June 2021. https://cset.georgetown.edu/publication/machine-learning-and-cybersecurity/