# Review 71: Deep Reinforcement Learning for Cyber System Defense under Dynamic Adversarial Uncertainties

**Paper: https://arxiv.org/abs/2302.01595v1**

פינת הסוקר:

המלצת קריאה מעדן ומייק: מומלץ לאנשים העוסקים בתחום ה-Cybersecurity או לאנשים שאוהבים Reinforcement learning

בהירות כתיבה: טובה

ידע מוקדם:

הבנה בסיסית בלמידה עמוקה מחיזוקים (DRL)

ידע בלמידה עמוקה

יישומים פרקטיים אפשריים: מערכת לזיהוי התקפות סייבר

פרטי מאמר:

פורסם בתאריך: 3.2.2023, בארקיב.

הוצג בכנס: -

תחומי מאמר:

למידה עמוקה מחיזוקים

אבטחת סייבר

כלים מתמטיים, מושגים וסימונים:

SDP

מבוא:

פיתוח אלגוריתמים האחראים על קבלת החלטות עבור מערכת הגנת סייבר איננה משימה פשוטה. הקושי העיקרי נובע מרמת דינמיות גבוהה שקיימת בסביבת תוכנה המכילות גם את כלי האבטחה (נקרא לו גם כלי הגנה בהמשך הסקירה) וגם את התוקף, המשפיעים יחד על הסביבה ומשנים את מאפייניה בכל נקודת זמן. בעיה נוספת הינה מגבלת המשאבים המוקצים לכלי המגן. משאבים אלו כוללים זמן מקסימלי לטיפול בפרצה, משאבי אנוש זמינים וכדומה. בכל פעולה של הכלי המגן הוא צריך להתחשב במגבלות אלו. למשל אם כל כוח העבודה שיכול לטפל בפרצה מסוימת תפוס, המודל לא יוכל להתריע על פרצה חדשה. לאחר שהכלי המגן מבצע פעולה, מספר המשאבים מתעדכן בהתאם, דבר שמשפיע על הסביבה.

בשנים האחרונות עם התקדמות הטכנולוגיה, חוקרים רבים ניסו לשלב למידה עמוקה מחיזוקים (DRL) בתוך כלים שונים תחום הסייבר. הפתרונות הללו ניסו למדל את הבעיה כ-Partially Observable Markov Decision Process (POMDP). הבעיה בפתרונות הללו הינן שהן התמקדו בסוג התקפות ספציפי, במרחב פעולות מצומצם מאוד של הכלי המגן יכול לבצע ובנוסף הניחו שקיים מודל לסביבה למרות שמודל הסביבה האמיתי מורכב מדי כדי למדל אותו בכלים מתמטיים פשוטים יחסית.

כדי לתת מענה לבעיות אלו, כותבי המאמר מציעים שיטה המתמקדת בפיתוח כלי מגן שיפעל כסוכן DRL שמטרתו להתמודד נגד תוקף מתוחכם ובעל יכולות מגוונות, כיוון שבעולם האמיתי אלו סוגי התוקפים אותם הוא (הסוכן) יפגוש. השיטה גם שכוללת פיתוח מסגרת (Framework) בה יהיה ניתן לבדוק ולהעריך את ביצועי הכלי המגן בצורה אמינה. מטרת הסוכן כפי שהגדירו אותה במאמר הינה ללמוד את אסטרטגיית ההגנה הטובה ביותר אשר תשפיע כמה שפחות על התהליכים הקיימים בסביבה שבה הוא מותקן (למשל להימנע ככל האפשר מפעולות כגון ניתוק מחשב מהרשת) ותדע להתחשב בכל שלבי ההתקפה האפשריים של התוקף.

לכן, החוקרים השתמשו ב-MITRE ATT&CK Framework כדי למדל את כלל הפעולות אותן יכול לבצע התוקף ובאותה נשימה לבדוק את ביצועי הכלי המגן. החוקרים מדגישים שהתרומות העיקריות במאמר זה הינן:

ההשוואה של ביצועי אלגוריתמי DRL שונים תחת תנאי סביבה אמיתיים.

פיתוחה של סביבת סימולציה חדשה המדמה את הסביבה האמיתית שאותה יראה המגן בחוץ באופן מדויק. הסביבה כוללת תוקף מתוחכם, המאופיינת בחוסר ודאות גבוה, תהליכים אמיתיים ואילוצים אמיתיים.

התוקף:

מטרת התוקף אותו מימשו במאמר הינה לנוע מתחילת גרף (תזרים) ההתקפה (אותו ניתן לראות באיור 1) לסוף המסלול. כדי לעשות זאת מומש מודל מבוסס אסטרטגיה. לפי האסטרטגיה אותה מיישם התוקף מגדירה הצלחתו בשלב נתון (במאמר זה נקרא טקטיקה) במסלול אם הוא הצליח להריץ את אחת ההתקפות (נקרא טכניקה במאמר) באותו שלב. למשל, אם התוקף הצליח להריץ את ההתקפה (טכניקה) הנקראת Active Scanning, הוא הצליח להגיע לשלב הנקרא Reconnaissance. כדי לבצע התקפה התוקף מריץ תהליך התקפה (Procedure) אשר בפועל מממש את ההתקפה. כלומר התקפה מוגדרת בתור סדרת מהלכים שעל התוקף לבצע ותהליך יכול להיות קוד בשפת תכנות כלשהי אשר מאפשרת לתוקף להגשים את מטרתו. כדי לעשות זאת התוקף מנצל חולשות הקיימות במערכת ולא טופלו כהוגן בכל נקודת זמן, מיקום התוקף הינו בשלב האחרון בו התוקף הצליח לבצע בהצלחה התקפה כלשהיא. התוקף יפסיד במשחק אם פעולת התקיפה שלו כשלה בשל פעולה של המגן (כלומר הוא נחשף). ההנחה היא שהתוקף רשאי לשנות את אסטרטגיית התקיפה שלו בכל נקודת זמן בהתאם למצב המערכת ותוצאת ההתקפה הקודמת שלו. לכן פתרונות הגנה סטטיים לא יספיקו שכן התוקף יוכל למצוא את המסלול (הרצת התקפה) שאותו כלי הגנה לא יצליח למצוא (אם קיים אחד).

:MITRE ATT&CK

ה-ATT&CK הינו שם למודל שארגון MITRE יצר לצורך הפשטה של סוגים שונים של התקפות הסייבר. ההפשטה נועדה כדי לעזור לקהילת הסייבר לסווג התקפות שהם רואים בשטח בהתאם למסגרת המוגדרת מראש ידי ATT&CK. המודל מציג גם פתרונות עבור סוגי התקפה שונים ולכן יש ערך בניסיון מיפוי התקפות לפי מודל זה. בנוסף אנשים שירצו לנסות לסמלץ התקפות בעצמם יוכלו לעשות זאת בקלות רבה יותר שכן המודל מכיר תיאור של מגוון סוגי ההתקפות ומה הן מנסות להשיג. תהליך ההפשטה של המודל מכיל שלושה רמות שונות של אבסטרקציה:

טקטיקה - שם כללי להתקפה. השם מעיד על מטרת ההתקפה.

טכניקה - תיאור טכני על ההתקפה. התיאור נשאר ברמת אבסטרקציה גבוה.

תהליך - דרך המימוש של טכניקה מסוימת. כאן ממש נכתב קוד כדי לנסות לממש את הלוגיקה המוצגת בטכניקה.

שלושת מונחים אלו מהווים את עמודי התווך עליהם בנוי המודל של MITRE.

המגן:

מטרת המגן הינה למנוע מהתוקף להגיע לשלב האחרון המסלול התקיפה. כדי לבצע זאת, על המגן להבין איפה התוקף נמצא כעת ולחזות את הצעד הבא שלו. כדי לבצע זאת על המגן להתמודד עם אתגר מורכב הטומן בו חוסר ודאות הנובע מהשיקולים הבאים:

התוקף יכול לנוע במסלולים שונים על גרף התקיפה. למשל, לאחר ביצוע ההתקפה User Execution הוא יוכל לנסות שתי פעולות אפשריות שכל אחת שייכת לשלבים השונים של גרף התקיפה למשל: Modify System Process או Modify Registry. זאת יחד עם העובדה שהמגן לא מכיר את גרף התקיפה אותו מיישם התוקף מקשים על המגן לחזות את הצעד הבא (טכניקה) של התוקף בצורה טובה.

המגן לא יודע איזה תהליך (procedure) ביצע התוקף כדי לבצע טכניקה (התקפה) מסוימת בגרף התקיפה.

המידע על מצב המערכת העומד לרשות המגן עלול להיות לא מלא. המגן נשען על כלי גילוי והתרעה שמטרתם לספק לו מידע על התרחשות חריגה במערכת. אותם כלים יתקשו לספק מידע על התקפה שבוצעה בשל ידע מוגבל על המערכת עליהן הן רצות או בשל העובדה שאין ברשותן מיפוי מפעילות חריגה להתקפה (טכניקה) ידועה.

המשימה של המגן נוסחה במאמר כבעיית (Sequential Decision Process (SDP. בבעיה זו, הפעולה הבאה של התוקף תלויה רק במצב (S) האחרון והפעולה (A) האחרונה אותה ביצע. כעת נגדיר את המאפיינים העיקריים של SDP.

מרחב המצבים (S): מכיל 17 מצבים שונים שכל אחד מהם הוא וקטור המייצג מיקום ייחודי של התוקף. המיקום מיוצג על ידי:

הטכניקה (סוג ההתקפה) בה השתמשו.

המצב ההתחלתי ממנו התחיל התוקף - Attack Initiated.

האם התוקף עבר למצב הסופי של התקפה זו- Attack Terminated.

מרחב הפעולות (A): למגן שלוש פעולות אפשריות אותן יכול לבצע:

Inactive: המגן לא עושה כלום

Reactive: המגן מסיר את כל התהליכים במערכת הקשורים להתקפה האחרונה אותה ביצע התוקף.

Proactive: המגן חוסם סט ספציפי של קריאות API של המערכת כדי למנוע התקדמות של התוקף. דוגמא: חסימה של קריאת API לשינוי מפתחות Registry.

פרס (Reward) - כותבי המאמר משתמשים בפונקציית הפרס הבאה:

כאשר:

שווה ל-1- אם המגן ניצח, אחרת 0.

- ההסתברות של התוקף להגיע לשלב האחרון במסלול / גרף ההתקפה.

- ההשפעה (=נזק) של התקפה מוצלחת על המערכת.

- המחיר של ביצוע פעולת הגנה. המחיר תלוי במידת ההפרעה של פעולת המגן על פעילותה התקינה של המערכת.

שילוב ו- מכמת את הסיכון (Risk) במצב s.

שילוב ו- מכמת את התמריץ של המגן לנצח את התוקף שכן אם הוא ינצח ( שווה ל-1-) הפרס יגדל.

נשים לב שבפעולות 2 ו-3 המגן עלול להפריע/לגרום נזק לפעילות התקינה של הרשת / מערכת שעליה הסוכן מגן.

הסביבה (Framework):

בתוך המסגרת אותה מציעים חוקרי המאמר מתבצע התהליך הבא:

בכל נקודת זמן, המגן (סוכן DRL) מבצע צעד הגנה ומקבל כמשוב על הפעולה את מיקום התוקף על גבי מסלול ההתקפה ופרס המתבסס על מיקום התוקף ועל ההשפעה על הסביבה.

כדי לאתר את מיקום התוקף, קיימות מערכות גילוי והתרעה אשר בוחנות קריאות API אשר מתבצעות במערכת ומתריאות על קריאות חריגות. בנוסף ברקע רצים תהליכים נוספים שגרתיים שבמקרה שהמגן משבש את פעילותם הסדירה עלולים לגרום לתקלות שונות במערכת. כפי שצוין בחלק הקודם, לא תמיד קיים מיפוי בין התראה לבין טכניקה (סוג ההתקפה) וגם אם קיים הוא לא בהכרח נכון. לכן כדי למדל את תופעה זו כותבי המאמר מניחים שקיים מיפוי בין התראה לבין טכניקה אך דיוקו אינו מושלם ויכול לעמוד על 85%, 75% ו-65%. (אלו הדיוקים שנבדק על ידי הכותבים) כלומר כאשר התראה ממופת לטכניקה מסוימת קיים סיכוי של 65% למשל שמיפוי זה הוא נכון.

כעת נתאר את החלקים הנוספים בסביבה שבה המגן רץ:

מגנון ה-Defense Success הינו חלק בסביבה האחראי לקבל את פעולת המגן ולהחליט האם היא עצרה את פעולת התוקף (ולכן הוא הפסיד).

מנגנון ה-Attack Success אחראי לקבוע האם פעולת התוקף הצליחה וההתקפה בוצעה בהצלחה. אם אכן כך, התוקף מקבל אינדיקציה שהוא יכול לעבור לשלב הבא, אחרת הוא נשאר במקום או מפסיד.

מנגנון ה-Benign Process Interrupter יודע לעקוב אחרי אילו תהליכים תמימים הופסקו כתוצאה מפעולת המגן. הוא שולח את הרשימה הזו למגנון ה-Reward Generation בשביל חישוב הפרס.

ניסויים:

בסביבת הניסוי שלהם, החוקרים לא השתמשו במלוא הטקטיקות והטכניקות המופיעות ב-MITRE אלא ב-7 מתוך 11 הטקטיקות. כתוצאה מכך, מספר הטכניקות נמוך אף הוא מהמקור. מספר פעולות המגן שמומשו הינן 23 כאשר 21 מתוכן הן מסוג Proactive (לא מוזכר מה הן אותן פעולות).

בכדי לאמן את המגן, החוקרים יוצרים קודם את כלל מסלולי ההתקפה האפשריים בגרף ההתקפה מנקודת ההתחלה לנקודת הסיום. מתוכם 80% מסך המסלולים שומשו לאימון ו-20% לטסט. בזמן האימון כל איטרציה כוללת ניסיון של התוקף לנוע על מסלול אחד מתוך גרף התקיפה מההתחלה עד הסוף. בכך נותנים למגן לראות המון סימולציות שונות ותנאי סביבה שונים.

החוקרים מציגים את הטענה שמידול התוקף מתבצע בשני מישורים: כישרון ועקשנות. כאשר כישורי התוקף גבוהים יותר כך גדל הסיכוי שההתקפה שהוא בוחר לבצע בשלב הבא תצליח. תוקף עקשן הוא תוקף שלא יוותר וימשיך לנסות לתקוף גם אם לא צלח בניסיון הראשון. פירוש אחר הוא שתוקף עיקש הוא אחד כזה שלא ניתן לזהות בקלות גם אם התקפה מסוימת שלו כשלה. במקרה זה עקשנות הינה מספר הפעמים שהתוקף יבצע התקפה שנכשלה לפני שיפסיד (כלומר יתגלה על ידי המגן). החוקרים ניסו רמות שונות של כישרון ועקשנות כדי לאמן את המגן תחת דרגות שונות של קושי. בתור המגן נוסו אלגוריתמי DRL שונים, ביניהם: DQN ו-A2C.

תוצאות:

בתרשימים הבאים ניתן לראות את ביצועי כל אחד מהמגנים נגד שלושה סוגים של פרופילים של תוקף כאשר ההבדל בין כל פרופיל הינה רמת הכישרון והעקשנות של התוקף. המדד בו השתמשו להשוואה הינו אחוז הפעמים שהמגן ניצח את התוקף מסה"כ המשחקים ששוחקו.

מהתרשימים ניתן לראות ש-DQN הציג את הביצועים הטובים ביותר נגד סוגי התוקף השונים. הוא לא רק הצליח לנצח ביותר משחקים מאשר A2C אלא הוא הצליח לעצור את התוקף בשלב מוקדם יותר כפי שניתן לראות לפי מספר האיטרציות בציר ה-X, שם מספר האיטרציות נמוך יותר בתרשימים של DQN.

קיים פוטנציאל גדול לסוג כזה של מגן וסביבה בעולם האמיתי. מגן אשר ראה מספר רב ומגוון של התקפות ושל פרופילים שונים של התקפות. ביצועי ה-DRL מראים כיוון מחקר עם ישומים אמיתיים ועזרה אמיתית לארגונים להתמודד עם האיומים השונים המחכים להם.

סיכום:

מאמר זה מציג את ההיתכנות של פריסת סוכנים אוטומטיים לצורך הגנה מפני התקפות סייבר. כותבי המאמר הראו אשר ניתן לאמן סוכנים כאלו באמצעות אלגוריתם DRL אשר יוכל להגן בהצלחה מפני התקפות סייבר שונות. בנוסף הסוכן מסוגל להתמודד עם סוגים שונים של תוקפים מתוחכמים.

שיתוף פעולה: הסקירה נכתבה בשיתוף פעולה עם עדן יבין