אבטחת מערכות ויישומים ברשת / תרגיל בית 1#

אריאל סטולרמן

1) המנגנון המספק גם Data confidentiality וגם Data integrity הוא מנגנון ה-Access Control (authorization), המאפשר שליטה בהרשאות הקריאה והכתיבה למשתמשים השונים. המנגנון יכול למנוע ממשתמש לא מורשה לשנות את המידע, וכך לשמר את ה-integrity שלו, וגם לקרוא את המידע, וכך לשמר את ה-confidentiality שלו. המנגנון מספק שתי תכונות אלו רק כאשר פועל לפי רשימת משתמשים והרשאותיהם, לפיהם הוא מספק או מונע גישה לקריאה או כתיבה של מידע. המנגנון מספק את הנ"ל עבור מידע נייח, כאשר עבור מידע העובר ברשת יש צורך בחתימות דיגיטליות לזיהוי משתמשים וב-encryption כדי לשמר את חיסיון המידע במעברו ברשת.

מנגנון אבטחת המידע הנוסף שהינו תנאי למנגנון ה-access control הוא כמובן מנגנון authentication לזיהוי המשתמשים, שכן אם לא ניתן לזהות באופן ודאי את המשתמשים המנסים לגשת למידע או לשנותו, לא ניתן לאכוף את חוקי ההרשאות וכך מנגנון ה-access control חסר משמעות.

2) התקפת DoS (Denial of Service) היא התקפה על זמינות מערכת, ע"י הצפתה בבקשות פיקטיביות על מנת למנוע מהמערכת לתת מענה ושירות לבקשות משתמשים לגיטימיים. ההתקפה גורמת לצריכה מוגברת של משאבי המערכת המותקפת: משאבי רשת (רוחב פס), משאבי מערכת (CPU, זיכרון וכו') ומשאבי תוכנה (כמו חיבורי DB). להלן מספר דוגמאות להתקפת DoS:

- DDoS – distributed denial of service – תוקף שותל תוכנה עויינת מורדמת במחשבים לא מאובטחים ברשת. כאשר התוקף רוצה לתקוף מחשב מסויים ברשת, הוא ייתן לכל המחשבים בהם שתולה התוכנה הוראה לבצע פקודה כלשהי ולגשת לכתובת המחשב המותקף. כל המחשבים (zombies) ייגשו לאותו מחשב בבת אחת ויגרמו לעומס גדול ואולי אף קריסה – שתמנע מתן שירות. אלו רשתות BotNet.
- Smurf attack: תוקף מבצע ping לשרתים רבים עם spoofed source IP – התחזות לשרת אותו רוצים לתקוף. השרתים שישלחו מענה לשרת המותקף יגרמו אצלו ל-traffic גבוה ועומס רב על הרשת.
- SYN attack: בהתקפה זו התוקף שולח לשרת המותקף חבילות SYN רבות הגורמות לו לפתיחת connections רבים, עד שלא נותרים לו משאבים לפתיחת חיבורים לגיטימיים. התקפה זו מיושנת ורוב השרתים כיום מחוסנים בפניה.

התקפה מסוג זה מהווה סיכון משמעותי בסביבת האינטרנט כיוון שהיא גם מאפשרת לתוקפים מיומנים לנצל מחשבים רבים אחרים הנמצאים על רשת האינטרנט לטובת ההתקפה (zombies), וגם זו הרשת הגדולה והנפוצה למתן שירותים שונים והתקפת שרתים המתקשרים עם לקוחותיהם על רשת זו בודאי עלולה לגרום לנזקים כלכליים כבדים (לדוגמא התקפה על אתרי הימורים, שכל דקה בה הם לא זמינים נמדדת בסכומי עתק).

3) עקרון ה-Positive Secure Logic הינו העקרון לפיו הכל אסור פרט למה שהוגדר ספציפית כמותר – deny-all policy. היתרון המשמעותי של שיטה זו היא האבטחה הגבוהה שהיא מספקת, שכן רק מי שהוגדר באופן פרטני כבעל הרשאות יוכל לגשת לרשת המוגנת לפי עקרון זה, וכל ניסיונות תקשורת אחרים, ביניהם התקפות, ייכשלו. החיסרון הגדול של שיטה זו לעומת allow-all policy היא התחזוקה הגבוהה שהיא דורשת והקונפיגורציה המסובכת יחסית לשיטה השניה (דורשת ידע מעמיק בניהול ניטור המידע, שלא תמיד קיים בקרב מנהלי הרשת).

עקרון ה-positive secure logic בא לידי ביטוי ב-firewalls המוגדרים לפי מדיניות זו, ברשימת חוקים שכל ניסיון תקשורת עובר דרכו ומסונן לפיו. רשימת החוקים למעשה מגדירה exceptions, כאשר אם תקשורת נכנסת עונה על אחד החוקים האלה היא מורשית מעבר. במצגת ישנה דוגמא לקונפיגורציה של firewall בה החוק האחרון מבטא את עקרון ה-deny-all policy, שהוא חוק שדוחה (מסנן) הכל, כלומר כל ניסיון תקשורת שלא ענה על אחד החוקים המתירים את מעברו יסונן ולא יתאפשר.

4)

- Stateless packet filtering: סוג הסינון הזה מסתכל על כל packet בפני עצמה, ללא התייחסות לשיוכה לזרם תקשורת קיים, כלומר לא נשמר מידע אודות מצב התקשורת. כל פקטה נבחנת ע"פ הנתונים המצויים בה בלבד. לרוב המידע בו נעשה שימוש לבחינת הפקטה: כתובת המקור והיעד, הפרוטוקול ומספר הפורט (באם מדובר ב-TCP או UDP; כיוון שרוב התקשורת באינטרנט נעשית מעל פרוטוקולים אלו, וישנה קונבנציה לגבי פורטים ספציפיים עבור אפליקציות שונות, לרוב ניתן לשלוט באמצעות stateless filtering בסוגים שונים של תקשורת נתונים).
- Stateful packet filtering: בניגוד לקודם, firewall העובד לפי סוג סינון זה מחזיק בכל רגע נתון את מצבי החיבורים העוברים דרכו. בנוסף למידע שבוחן ה-stateless packet filtering, נשמר גם השלב הנוכחי במחזור החיים של כל חיבור (אתחול session, handshake וכו'). כל

פקטה שמגיעה משוייכת לחיבור מסויים ונבדקת התאמתה למצב החיבור. באם לא נמצאה התאמה, היא נבחנת לפי קריטריונים של חיבורים חדשים. לעומת זאת אם נמצא שיוכה לחיבור מסויים, היא תורשה לעבור הלאה.

ה-stateful בטוח יותר, שכן מאפשר בחינה יותר מורכבת של כל פקטה לפי מצב התקשורת הקיים, ה-scope עליו מסתכלים רחב יותר ובאופן עקרוני הכללים החלים על הפקטות העוברים דרך stateless חלים גם ב-stateful (למעט מקרים בהם פקטות "משוחררות" מידית כאשר משוייכות ל-connection מוכר במצב המתאים), אך לא להיפך (מכאן יתכנו פקטות שיתפסו ע"י stateful אך לא ע"י stateless).

5) ב-port restricted cone המיפוי לכתובת IP ופורט חיצוניים נעשה על בסיס כתובת מקור ופורט מקור בלבד, כאשר שרתים חיצוניים יכולים לשלוח פקטות בעלות כתובת IP ופורט מסויים לשרת פנימי רק אם אותו שרת פנימי שלח בעבר פקטה לאותה כתובת IP ואותו פורט. בפרט, אם שרת פנימי בעל כתובת מסויימת שולח דרך פורט מסויים פקטות לשני שרתים שונים, שניהם יוכלו לחזור אליו דרך אותה כתובת IP חיצונית ופורט חיצוני. לעומת זאת, ב-symmetric המיפוי לכתובת IP ופורט נעשה על בסיס כתובת מקור, פורט מקור, כתובת יעד ופורט יעד, כאשר הגבלת השרתים החיצוניים זהה לקודם, אך השוני הוא שאם שרת פנימי ניגש מכתובת מסויימת ופורט מסויים לשני שרתים חיצוניים (בעלי כתובת ופורט שונים זה מזה), שניהם יראו את השרת הפנימי בכתובת ופורט שונים (יראה שונה "כלפי חוץ").

ההבדל בין שתי השיטות בא לידי ביטוי כאשר רוצים לחבר שני שרתים (פנימי וחיצוני) ברמת האפליקציה. למשל בהעברת מדיה, ברמה האפליקטיבית כתובת ה-IP נשארת הכתובת הפנימית, וה-NAT לא נוגע ב-packet ברמת ה-DATA (ולא יודע איך). כך, אפליקציה בשרת החיצוני אליו ניגשים חושב שה-IP שלו הוא הפנימי. כמו כן אפליקציה לא יכולה לראות את הכתובת החיצונית שלה. קיים שירות המאפשר לאפליקציה לראות את הכתובת שלה כלפי חוץ וכך לאפשר ברמת האפליקציה לראות את הכתובת החיצונית של המחשב הפנימי, וכאן בא לידי ביטוי ההבדל – בשיטת symmetric השירות לא יעבוד מן הסתם, שכן מול כל שרת חיצוני שונה, כתובת ה-IP והפורט שונים. לסיכום, אפליקציות המתייחסות לכתובת המחשב לא יוכלו לעבוד כאשר המיפוי הוא symmetric, כיוון שכל שרת חיצוני רואה את השרת הפנימי בכתובת אחרת.

6) באופן פוטנציאלי אכן ניתן להגיע באמצעות מימוש DMZ ע"י single firewall לאותה רמת הגנה כמו שניתן במימוש DMZ ע"י dual firewall, ע"י החלת שני סטים של כללים באופן המדמה שימוש בשני firewalls – סט כללים עבור גישה לשרתי ה-DMZ וסט כללים עבור גישה לשרתים הפנימיים.

עם זאת, מימוש DMZ באמצעות שני firewalls מאפשר כתיבת חוקים שונים בין firewall אחד לשני באופן מופרד היוצר אי תלות בין firewall אחד לשני, וכך יישום קונפיגורציות שונות בין שתי חומות האש מקטין את הסיכוי שטעויות בקונפיגורציה יחלחלו מ-firewall אחד לשני. לעומת זאת שימוש ב-single firewall מגדילה את הסיכוי לחשיפה להתקפה שכן אם פורץ מצליח לעקוף את ה-firewall הוא יכול לגשת לשרתים הפנימיים, לעומת היתקלות ב-firewall נוסף בשיטה השניה. גיוון ב-vendors של שני ה-firewalls מקטין גם הוא את הסיכוי להתקפה בכך שפגיעות ב-firewall של vendor אחד בסבירות גבוהה לא תהיה בשני (מ-vendor אחר), וזאת כמובן מתאפשר רק במימוש DMZ באמצעות שני firewalls.

7) ה-Firewalls אינם מספקים הגנה בפני web application vulnerabilities ע"י משתמשים עויינים כיוון ש ה-firewall מגדיר חוקים עד רמת הפורט, ולכן אם פורט כלשהו פתוח, ישנה פתיחות להתקפה – אין ל-firewall חוקים לבדוק האם למשל HTTP request המתקבלת היא חוקית או לא, ה-firewall יעביר בין כה וכה. אמנם כיום ישנם deep-packet inspection firewalls, המתייחסים ל-data בצורה מוגבלת. אבל, כדי להסתכל על ה-DATA ולחפש request לא חוקי, יש צורך לחבר Packets לכדי request message מלאה, וגם תחת הנחה שה-firewalls יודעים לבצע זאת, ה-firewalls מעבירים הרבה פרוטוקולים, וכאשר התעבורה מוצפנת ה-data סגור לחלוטין מול ה-firewall ומגביל אף את המתקדמים באיתור ניסונות פריצה ברמה האפליקטיבית. לסיכום, העובדה שה-firewalls פותחים פורטים רבים ושרוב התעבורה מועברת באופן מוצפן (SSL) אזי ברמה האפליקטיבית ה-firewall לא מגן מפני ניצול פגיעויות ברמה זו.

8) השילוב הנכון בין הצפנה סימטרית ואסימטרית, כמו גם אופן ההצפנה בפרוטוקול ה-SSL, הוא שימוש בהצפנה אסימטרית כדי להחליף מידע אודות מפתח סימטרי, כך ששאר התקשורת תהיה מאובטחת תחת הצפנה סימטרית עם אותו מפתח עליו הוסכם. ההצפנה האסימטרית מאפשרת לשני צדדים להחליף מידע ראשוני מבלי הצורך להפגש פיסית לצורך הסכמה על מפתח ובאופן מאובטח, ואילו הצפנת התקשורת לאחר מכן תחת מפתח משותף (הצפנה סימטרית) יעילה יותר חישובית ומהירה יותר מאשר המשך התקשורת תחת הצפנה אסימטרית.

השימוש הנוסף של public key technology (הצפנה פומבית – אסימטרית) בפרוטוקול ה-SSL היא לצורך אותנטיקציה של השרת, כאשר השרת מעביר הודעה המוצפנת באמצעות המפתח הפרטי שלו (הידוע רק לו) והלקוח מפענח את ההודעה הזו באמצעות המפתח הפומבי המוצהר של אותו שרת. כך ניתן לודא שהשרת מולו מתבצעת התקשורת הוא אכן השרת הנכון ולא ניסיון התחזות לשרת זה.

9) הבדיקות שצריך לבצע SSL client על ה-certificate המתקבלת מה-server הן:

- זהוי trusted CA מתוך היררכית האותנטיקציה של השרת: על הלקוח למצוא CA עליו "ניתן לסמוך" מתוך רשימת ה-CAs ההיררכית, וכאשר ימצא אחד כזה למעשה הוא יתן גושפנקא שתחלחל מטה בהיררכיה עד לשרת עצמו. הדפדפן מכיל בהגדרותיו רשימה של trusted CAs שנבדקו ואושרו שלמולם מתבצעת הבדיקה. משמעות הבדיקה היא וידוא שה-issuer, או שרשרת ה-issuers שבסופו של דבר מגיעים למנפיק התעודה לשרת, הם בעצמם גופים שניתן לבטוח בהם.
- תיקוף תעודת השרת ע"י בדיקה ב-CRL: כל CA מפרסם רשימת רבוקציה של תעודות, כאשר תעודות הנמצאות ברשימה זו אינן תקפות. על כן על הלקוח לוודא שתעודת השרת מולו אינה נמצאת ברשימה זו (של ה-CA issuing של אותו שרת).
- ולידציה של תאריך תוקף התעודה: כל תעודה מונפקת לפרק זמן מוגבל, ועל כן על הלקוח לבדוק שהתעודה עדיין בתוקף.
- בדיקת שם השרת (שקול לכתובתו) המזדהה אל מול זה המופיע התעודה: בדיקת ההתאמה חיונית כדי לזהות ניסיונות התחזות.
- וידוא החתימה בתעודה: פענוח באמצעות המפתח הפומבי של ה-CA / היררכיית ה-CAs ווידוא ה-hash של החתימה, וזאת בכדי לבצע אותנטיקציה של השרת ולודא שהאישור תקף ונכון.

10) ה-SSL record layer מספקת integrity and data-origin authentication ו-confidentiality, ומכאן שמספקת מענה לאיומי ציטוט, שינוי מידע והתחזות, כאשר ה-confidentiality מבטיחה שהמידע יהיה נגיש לקריאה ופענוח אך ורק לגורמים המורשים לכך, ה-integrity מבטיחה שהמידע היוצא מנקודה A הוא המידע המתקבל בנקודה B ללא שינוי וה-data-origin authentication מבטיח את מקור המידע (שכל צד שולח הוא אכן מי שהוא טוען להיות).

- Confidentiality: שימוש באלגוריתמי הצפנה סימטריים (כמו IDEA, DES וכו').
- Integrity and Data-origin authentication: שימוש ב-MAC (המתקבל ע"י פונקצית hash מסוג MD-5 או SHA-1) לצורכי בדיקת אמינות המידע ואימות מקור המידע.

המידע מחולק תחילה לרשומות בעלות גודל מקסימלי, כל רשומה עוברת דחיסה, לאחר מכן מחושב ה-digest לכל אחת, ולבסוף כל רשומה יחד עם ה-digest שלה עוברים הצפנה. כך מבוצעים כל התהליכים הנדרשים לשמירת התכונות הנ"ל.