# יותר בטוח להיות פתוח
## על מערכות אבטחת נתונים פתוחות

**גדי רפפורט ואבי ויסמן** (see-security.com)

סביבת תשתיות פתוחה (Open Source) היא סביבה עתירה באמצעי אבטחת מידע יעילים ומקצועיים. מיטב מומחי לוחמת המידע ואבטחת המידע בעולם סומכים עליהם את ידם. למרבה ההפתעה, חלק מכלי האבטחה שנבנו בסביבה פתוחה ויועדו במקורם לסביבה פתוחה, משמשים כיום לאבטחתן של סביבות "סגורות" - קנייניות - כמו למשל, בסביבת Microsoft.

ואולי זה דווקא לא כל כך מפתיע. סקרי סיכונים הנערכים חדשות לבקרים בארץ ובעולם, מגלים "תגליות" שגרתיות למדי: מרבית הפרצות באבטחה נגרמות עקב העדר ידע מספיק בקרב כוח-האדם המקצועי, או עקב העדר כוח-אדם מספיק, ולא בעטיים של כשלים בתוכנת אבטחה מסויימת. ניתן איפוא לומר, כי כלי האבטחה הנהוגים כיום – בין אם בסביבה פתוחה, ובין אם בסביבה קניינית של יצרנים ידועים – מספקים את שהם מתיימרים לספק. נכון לומר, שטכנולוגית האבטחה נמצאת בתחילת דרכה וכי צורכי האבטחה רבים לאין שיעור מהיצע הפתרונות. אולם, הכלים הקיימים עומדים ביעדים עבורם תוכננו.

אם כך, מדוע אנו רוכשים כלים יקרים של יצרנים ידועים, במקום מערכות אבטחה המבוססות על קוד פתוח?

מערכות קוד פתוח הינן תוכנות אשר רשיון השימוש שלהן מאפשר למשתמש חופש מוחלט לגבי השימוש בהן. יתרה מכך, בידיו החופש לשנותן בהתאם לרצונו, ולהפיצן מחדש תחת שם שונה. מערכות תוכנה מורכבות מאלפי שורות קוד, ושנות אדם רבות מושקעות בפיתוח מוצר התוכנה. היצרן מצפה להחזר נאות על ההשקעה הזו. לכן מוצרי תוכנה מוגנים ברשיון שימוש מגביל, והגישה אל קוד התוכנה חסומה.

אז מדוע בכל זאת בוחרות חברות תוכנה רבות לבנות את מוצריהן במודל הקוד הפתוח?

קיימות מספר תשובות:

1. דינמיקה קהילתית. מפתחים, בודקים, ומשתמשים רבים תורמים מזמנם ומכשרונם לפיתוח המוצר, בעיקר משום תחושת השיתוף, ה"יחד".
2. ה"טרנדיות" של הקוד הפתוח. התבססות על קוד פתוח מאפשרת חדירה מהירה יחסית לשוק לצד חשיפה רבה.

3. קל באופן יחסי למכור מוצר "חינמי", ולהישען על שירותי תמיכה והטמעה כמייצרים את הרווח עיקרי.

מנקודת מבטם של משתמשים, מנהלי רשתות, מנהלי מערכות מידע, ואפילו יצרני תוכנה, מאפשרות מערכות קוד פתוח תיכנון מורכב ומותאם במדויק לדרישות המקוריות של האירגון, שמשתחרר בכך מתלות ב"יכולות" מוגבלות המוגדרות מראש על ידי היצרן. מערכות אלו משמשות בסיס ליצירת מערכות מידע ייעודיות המקיימות דרישות וצרכים נקודתיים של ארגון / משתמש.

האם יתרונות אלו משתקפים גם במערכות אבטחה מבוססות קוד פתוח?

מערכות קוד פתוח, כשמן כן הן, הן פתוחות. האם נכון לסמוך על מערכת אבטחה שקוד המקור שלה זמין לכל החפץ באיתור חורי האבטחה שבו?

איזו מערכת מאובטחת יותר: מערכת אשר איש פרט ליצרנה אינו יודע את המבנה הפנימי שלה ולא יכול לבדוק את אטימותה; או מערכת שרבים מספור בודקים את אופן פעולתה, מגלים באגים ונקודות תורפה, מפרסמים את ממצאיהם, ובכך חושפים נקודות תורפה? האם שקיפות המערכת מחזקת או מחלישה את כושר האבטחה שלה?

כמה זמן חולף מרגע מציאת באג או פרצה, ועד לרגע פרסום הטלאי (Patch) המיועד לחסום אותה? האם התגובה של יצרני התוכנה הקניינית מהירה יותר מתגובת הקהילה התומכת במוצר פתוח?

סביר להניח כי דיונים אלו יימשכו עוד זמן רב. סכומי כסף מרשימים עומדים על הפרק, ועובדה זו מבטיחה כי תמיד יהיו בעלי אינטרסים שינסו לטרפד מגמות העלולות להיטיב עם הלקוח, אך לפגוע ברווחיהם.

מה תהיה בחירה נכונה עבורנו, אם נתבקש לתכנן מערכת אבטחה אירגונית?

בראש ובראשונה, יש להכיר את האפשרויות המסחריות השונות, ואת הפתרונות החלופיים שמציעה קהילת הקוד הפתוח. לאחר איסוף מידע על דרישות המערכת מחד, ועל הפתרונות הקיימים מאידך, ניתן להתחיל להשוות בין הפתרונות המסחריים לפתרונות הפתוחים, לפי קריטריונים המוכתבים על ידי דרישות האירגון.

## Firewall

מערכת Firewall היא הצינור דרכו נכנס ויוצא מידע מהאירגון. מערכות אלו סורקות את כל תעבורת הרשת היוצאת והנכנסת ומבצעות החלטות לניתוב או חסימה שלו, בהתבסס על אוסף חוקים מוגדר מראש. החוקים מתבססים על מבנה Packet של פרוטוקול TCP/IP, ומנוסחים על-ידי שימוש במיספרי Port (סוג השירות אליו מופנית הבקשה) וכתובות יעד/מוצא, ועל-ידי הוראה לנתב או לחסום את ניסיון

ההתקשרות. כל Packet נבנה בצורה זהה, כך שזיהוי כתובת היעד/מוצא ומספר ה-Port מתאפשר לגבי כל ניסיון התקשרות. מערכות Firewall עדכניות מאפשרות גם קביעת חוקי תעבורה על בסיס תוכן ה-Packet, ולא רק על בסיס המאפיינים. כך ניתן למנוע שימוש לרעה בשירותים ציבוריים מורשים (שרתי דאר, אתרים, פורומים וכד'), ולהגן מפני התקפות ברמת היישום.

באמצעות Firewall, יכולים ארגונים (או משתמשים פרטיים) לנסח מדיניות אבטחה ברורה לגבי חשיפתם לרשת האינטרנט ותעבורת המידע היוצא והנכנס מהאירגון. מערכות אלו נפוצות עד כדי כך שהפכו ל-Buzz-Word, בתחום אבטחת המידע. היצרן המוביל בתחום הוא חברת Check Point, אולם קיימות עשרות אלטרנטיבות מבוססות חומרה (Cisco PIX) או תוכנה ( Symantec Enterprise Firewall, Microsoft ISA, SonicWall PRO, ועוד). בחירת פתרון Firewall הולם היא נושא רחב ביותר, ומצריכה מחקר מעמיק של הדרישות והפתרונות הקיימים.

פתרון אפשרי על בסיס קוד פתוח היא Linux IPChains / IPTables. תוספת חשובה ליכולות האבטחה של לינוקס היא היכולת לבצע Packet Filtering: סינון תעבורת רשת והפעלת שירותי NAT (תרגום כתובות IP). שירות זה מוכר כ-IPTables או IPChains, בשל היכולת להגדיר טבלאות (או שרשראות) חוקים, שמשמשים בתורם בסיס להחלטות הניתוב.

תכונות עיקריות של IPChains/IPTables:

1. חוקי סינון מופעלים על כל Packet נכנס / יוצא.
2. זיהוי תהליכים מתמשכים. עבודה מול שרת אינטרנט מצריכה פניות תכופות מצד הלקוח ותגובות תכופות מן השרת; יכולתו של ה-Firewall לזהות התקשרויות מתמשכות, ולהעביר אותן ללא בדיקה מעמיקה נוספת, היא שיפור יחסי רב בביצועי ה-Firewall, ובזמן התגובה של שרת האינטרנט ללקוחות. נתמך רק על ידי IPTables.
3. אזור מפורז (DMZ – Demilitarized Zone). ניתן לתחום אזורי רשת מפורזים, על ידי שימוש בשתי מערכות Firewall, או על ידי חיבור של שלושה כרטיסי רשת שונים במתכונת של Three-Pronged Firewall.
4. מיסוך כתובות. אפשר לנתב בקשות שירות אל שרתים הנמצאים בתחומי ה-Firewall לפי כתובת חיצונית אחת, מבלי לחשוף את הכתובת האמיתית של כל שרת NAT, Port (Redirection).
5. תיעוד. מערכת התיעוד עוקבת אחרי פעילות ה-Firewall ומאפשרת תחקיר לאחר אירוע.

6. יכולות הרחבה:

א. Application Proxies – שרתים הבודקים את תכני בקשות השירות, למציאת קוד זדוני או בקשה לא קבילה. טיפול בבקשות שירות HTTP, SMTP, DNS, FTP וכד'. מוצרים אפשריים: TIS FWTK, Squid, Apache, FreeBSD.

ב. VPN – יצירת ערוצי תקשורת מוצפנים ברשת קיימת, בין שרת לשרת או בין שרת ללקוח.

ג. Event Management Policy – ניתוח קבצי היומן (Log Files) של ה-Firewall, ליצירת תגובות אוטומטיות לפי דפוסים חוזרים.

## עלויות שוטפות

ניתן למצוא הפצות שונות של לינוקס. מרביתן פטורות מתשלום או מתאפיינות בעלות חד-פעמית סמלית. בדרך-כלל, גם אין צורך לרכוש רשיונות לכל מחשב. ברובן קיים יישום של IPChains/IPTables שניתן לשנות ולהתאים לצורכי האירגון. זאת מבלי להשקיע במוצרים תומכים / נלווים.

## מערכות IDS

מערכות IDS (Intrusion Detection) הן כלבי השמירה האלקטרוניים של רשתות המידע הדיגיטליות. הן מסייעות להפוך את תהליך המעקב אחר מערכות מחשב או רשתות מחשב לאוטומטי, על ידי ניתוח של כל הפעולות המתרחשות תחת פיקוחן. מערכות אלו הפכו בשנים האחרונות למרכיב חיוני בתשתית האבטחה האירגונית. הן משתלבות במערכות קיימות כגון אנטיוירוס, Firewal, שירותי דאר ו-Web. הן מותקנות כדי להתריע על פעילות החשודה כפעילות עויינת. להבדיל ממערכות Firewall שמסננות מידע על פי תכונות ה-Packet ותכולתו, מערכת SNORT מייצרת דיווח המוגש למנהל הרשת.

בליבן של מערכות IDS נמצא רכיב Sniffer, המסוגל לקלוט ולקרוא את התעבורה ברשת, גם כאשר זו לא יועדה לו, ולחפש תבניות מוכרות של פריצה. תבניות אלו נשמרות במאגר מידע אשר מתעדכן בתכיפות מול שרתים ברשת האינטרנט. מערכות IDS משוות בכל רגע נתון מידע העובר ברשת לתבניות השמורות במאגר המידע. מערכות IDS מסוגלות לזהות מגוון התקפות, כגון: Trojan Horses, Port Scanning, D\Denial of Service Attacks, Known Exploits, Redirections, Tunneling ועוד. מערכות IDS מסוגלות גם לבצע פעולות מנע. ביכולתן להגיב לפי

רשימת פעולות מוגדרת מראש עבור תבניות התקפה מוכרות, ולנהל תיעוד ודיווח יסודיים, המאפשרים למנהל הרשת לדעת שפריצה מתרחשת ולהגיב בהתאם.

## SNORT Intrusion Detection Technology

את מערכת SNORT יצר מרטין רוזך בשנת 1998 והיא הפכה במהירות לטכנולוגית ה-IDS הנפוצה ביותר. גם חברות מחוץ לקהילת הקוד הפתוח מתכננות ומשווקות מוצרים המושתתים על טכנולוגיה זו. מערכת SNORT מזהה חדירות בזמן אמת, ומודיעה למנהל הרשת על כל ניסיון פריצה המתרחש ברשת.

SNORT הינה מערכת קלה להתקנה ולתפעול. ניתן להתחיל להשתמש בה בתוך מספר דקות, והיא אינה מתנגשת עם מערכות קיימות ברשת. מאגר הנתונים שלה מאפשר זיהוי של למעלה מאלף סוגי התקפות ופריצות שונים, והמאגר מתעדכן על בסיס קבוע – בדיוק כמו מאגרי נתונים של תוכנת אנטיוירוס. מערכת SNORT פועלת בשלושה מישורים, לפי הגדרת מנהל הרשת:

1. Packet Sniffer – האזנה לכלל התעבורה ברשת.
2. Packet Logger – תיעוד התעבורה, לזיהוי בעיות וטיפול בהן.
3. Intrusion Detector - זיהוי התקפות לפי כל התעבורה הנסרקת.

תכונות עיקריות של מערכת SNORT IDS:

1. זיהוי והתראה על תעבורה התואמת לתבניות מוכרות, כגון:

1. Buffer overflows, Stealth Port Scans, CGI attacks, SMB probes and NetBIOS queries, NMAP and other port scanners, well-known backdoors and system vulnerabilities, DDoS, ועוד.
2. שימוש ב syslog או הודעות WinPopUp להודעה על התרעות.
3. ניסוח תבניות תקיפה וחוקי תגובה מותאמים אישית, לפי צורכי האירגון.
4. זיהוי תעבורה לא מורשית, כגון רשתות שיתוף קבצים, חיבורי NFS, וכו׳.
5. ניתנת להגדרה כ-Host IDS, וככזו לעקוב אחר אפליקציות ושירותים הרצים במערכת מחשב מסוימת ולהתריע על התנהגות חשודה.
6. שומרת מידע אשר ניתפס ברשת (התעבורה עצמה – Packets) בצורה ניתנת לקריאה, ומסדרת את המידע באופן היררכי, להקלה על ניתוח שלו.

עלויות שוטפות של מערכת SNORT IDS:

מערכת SNORT היא מוצר חינמי, ואינה דורשת השקעה כספית בהטמעה בתוך מערכות האירגון הקיימות. גם העידכון התקופתי אינו מצריך תשלום.

### HoneyPot

מערכת HoneyPot יוצרת מערכות מחשב מדומות המדמות פעולה של מחשבים ושרתים ברשת קיימת, ומהווים "מלכודת דבש" עבור תוקפים פוטנציאליים, המסיטה את תשומת לבם ומאמצי הפריצה שלהם מהרשת האמיתית. תצורת המחשבים המדומים ניתנת להגדרה בנפרד, וכך ניתן ליצור רשת הטרוגנית של מחשבים מדומים (HoneyNet), המדמה רשת ארגונית מלאה בדיוק רב. בין השירותים נמצא: Web services, Mail services, FTP services, Terminal services ונוספים. ניתן אפילו לדמות התנהגות של מערכות הפעלה שונות, כך שפורץ שיבצע FootPrinting לזיהוי השירותים ומערכות הפעלה יקבל את התחושה של עבודה מול מערכת אמיתית.

הנחת "מלכודת דבש" כזו מעניקה לארגון שני יתרונות עיקריים:

1. יכולת איסוף של חומר מפליל על תוקפים פוטנציאלים.
2. איסוף וניתוח מידע על שיטות חדירה פוטנציאליות, וניבוי תקיפה אמיתית על האירגון.

קיימת עדיין מחלוקת משפטית לגבי השימוש במידע הנאסף על ידי מערכות HoneyPot, על רקע אי-הבהירות לגבי החוקיות של "עידוד" הפריצה – פעולה שעשויה להחשב כהכשלה (entrapment). אך גם אם אינו קביל כראיה בבית משפט, המידע הנאשף הוא בעל חשיבות עליונה, ובנוסף, במהלך נסיונות התקיפה מבזבז הפורץ זמן יקר על פריצת מכשולים שלא יובילו אותו אל היעד האמיתי שלו, ולא יגרמו לנזק לארגון.

### HoneyD

HoneyD היא מערכת HoneyPot מבוססת קוד פתוח, שיצר נילס פרובוס מאוניברסיטת מישיגן. המערכת מאפשרת יצירת מחשבים ורשתות מדומים המסוגלים להטעות סורקי רשת כגון: Nmap, xProbe, ונוספים.

תכונות עיקריות של מערכת HoneyD:

1. הדמייה בו-זמנית של אלפי מערכות מדומות.
2. הגדרת מגוון שירותים בסביבת הרשת המדומה: HTTP, HTTPS, SMTP, SNMP, POP3 ונוספים.
3. יצירת טופולוגית ניתוב עצמאית, המאפשרת למחשבים המדומים לתקשר זה עם זה ולנתב מידע ביניהם.
4. יצירת סביבת הרצה לשירותים ויישומים במחשב המדומה, שמגיבים לכל פעולות התוקף - כולל נסיונות פריצה לאותם שירותים / יישומים.

## סיכום

אם עד לפני שנים ספורות היה מנהל מערכות מידע מרים גבה למשמע ההצעה להקמת מערך אבטחה המבוסס על קוד פתוח, הרי שכיום, הוא יתייחס להצעה מסוג זה בכובד ראש.

בכך, הוא עשוי ליהנות ממספר יתרונות:

• חיסכון ניכר במשאבים כספיים, שהיו נדרשים לרכישת מוצרי אבטחה מסחריים.

• בנייה מושכלת של מערכת IT על-גבי מצע בטוח, יעיל ומוגן להפליא, לקראת הכניסה הבלתי-נמנעת לסביבת קוד פתוח.

• שחרור מהתלות ביצרני תוכנה מסחריים, תוך שיפור היכולת המבצעית של צוות ה-IT לטיפול עצמי במערכות האבטחה מבוססות קוד פתוח.

• רכישת היכולת לשלב כלים פתוחים תחת מטריה של ממשק אחיד, יתרון שנמנע מיחידות IT המצטיידות בכלים קנייניים סגורים מיצרנים שונים.

• גישה לבסיס מקוון רחב של כלי עזר ומידע היקפי, במסגרת קהילת פיתוח פעילה.

החסרונות שאותם עליו לשכלל בהחלטה:

• הצורך להחזיק מומחה לקוד פתוח עשוי להתבטא בהוצאה כספית (בעיקר בסביבה עתירת מערכות קנייניות).

• תשתית תמיכה מבוזרת בה מתאפיינות מערכות קוד פתוח עשויה לעורר את הצורך ברכישת חבילת תמיכה מחברת צד שלישי.

• העדר כוח אדם המיומן בעבודה סביבה פתוחה (למרות שגם בכוח אדם המיומן בתפעול מערכות אבטחת מידע בסביבה קניינית קיים מחסור מתמיד).

לסיכום, אנו ממליצים על הצטיידות במערכות אבטחת מידע מבוססות קוד פתוח, לשני סוגים של ארגונים.

1. לארגונים גדולים במיוחד, שההוצאות השנתיות על רכש "כלי אבטחת מידע" עולה על 100,000 דולר. החסכון שארגונים כאלה עשויים להפיק ממעבר למערכות פתוחות גדול בדרך כלל מההשקעה הנחוצה לביצוע המעבר.

2. לארגונים קטנים במיוחד, שמוציאים לא יותר מ-20,000 דולר בשנה על מערכות אבטחה קנייניות. בארגונים מסוג זה פועל בדרך כלל צוות IT קטן, וחלק סמלי מהחסכון על מערכות קנייניות יכול לשמש להכשרת מומחה לסביבת קוד פתוח, ולמעבר הדרגתי לסביבה פתוחה לגמרי.