JAVA JAVA, PROXY PROXY

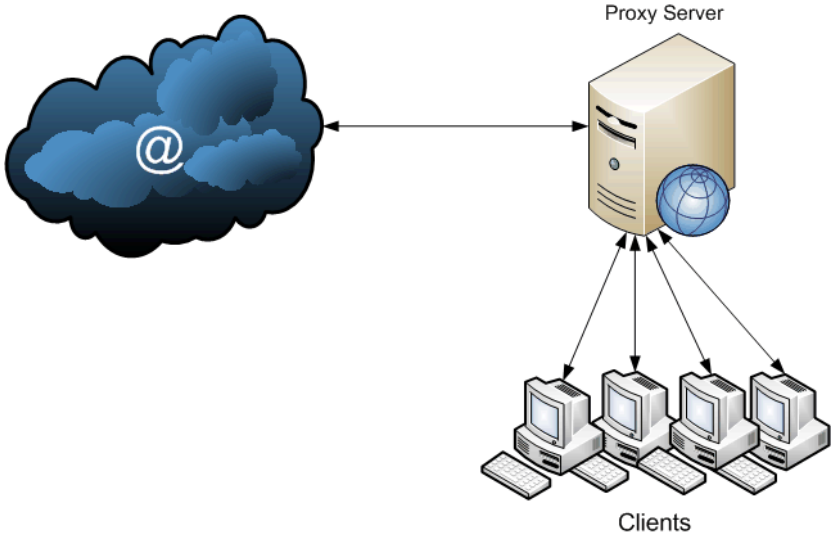
מאת רועי חורב (AGNil)

זוכרים את המערכון העתיק של שי ודרור? במערכון רואים בחור המחפש עבודה במחשבים והבחורה בלשכת העבודה לא כל כך מבינה במה הבחור עוסק. "אולי תיקח את ה-JAVA לכיוון המכונאות רכב?". יתכן והמאמר הזה לא יסביר לגברת מה הוא בעצם Proxy, אך הוא בהחלט יעשה קצת סדר במושגים שבדרך כלל צצים בשיחות סלון הנוגעות לנושא.

המאמר הבא מנוסח בלשון פורט 80 מטעמי נוחות, אך הוא מתייחס גם לפורט 443.

הקדמה

אם נשאל את המילון מה משמעות המילה Proxy הוא יסביר לנו בנימוס שזה "ייפוי כוח", "נציג", "בא כוח" או אפילו "שליח", ואכן פחות או יותר כך הדבר. שרת Proxy הוא שרת שמטרתו הצגת בקשות בשם מחשב אחר. אפשר לפגוש את המושג בהרבה תחומים ואנחנו נדון כאן בנפוץ מביניהם – WEB Proxy, שרת שמטרתו התעסקות עם תעבורת HTTP. ברוב הארגונים היום מוטמע שרת שכזה (ולפעמים הרבה יותר). הרעיון הוא מאוד פשוט – תחנות העבודה מציגות את בקשות ה-HTTP שלהם לשרת ה-Proxy, הוא מבחינתו מציג את הבקשה לאתר האינטרנט ומחזיר בשמו את התשובה לקליינט:



למה בעצם צריך את השרת הזה בדרך? מה אנחנו מרוויחים

1. **מהירות** – רבים מהשרתים הללו (במיוחד אצל ספקיות האינטרנט) מספקים שירותי caching, או בעברית צחה – זיכרון מטמון.
ננית שהגעתי היום בבוקר לעבודה וגלשתי לאתר ynet.co.il. כאשר פניתי לאתר, שרת ה-Proxy שמר אצלו ב-cache את התמונות, הסרטונים והפרסומות מהאתר. בעוד כשעתיים יגיע שכני למשרד גלעד וגם הוא יגלוש לאתר ynet. ההבדל הוא שהוא יוריד את מרבית משקל האתר משרת שנמצא יחד איתו ברשת, סביר להניח שברוחב פס של 100M לפחות.

חשוב לציין ששרתים ששומרים caching לא שומרים את כל האתר, אלא רק אובייקטים מסוימים מתוך העמודים. אנשים רבים חוששים לשים caching בכדי למנוע שמירת סיסמאות של משתמשים או קבלת מידע לא מעודכן מאתרים דינאמיים – תופעות שלא אמורות לקרות בפעילות תקינה של שרת Cache.

2. **אבטחה** – כמובן שהחלק העיקרי שנתעסק בו הוא הפן האבטחתי ששרת ה-proxy מספק לנו. מרבית הארגונים שמטמיעים שרת Proxy נוטים לסגור את הגלישה מתחנות הארגון בשלב שלאחר מכן ובכך כופים מצב בו כל תחנה שרוצה לגשת לאינטרנט חייבת לעבור דרך שרת ה-Proxy. דרך אחת לבצע מהלך שכזה היא פשוט לחסום את PORT 80 לכיוון העולם ב-FW. נכון, מדובר בדרך עקיפה די בקלות, אבל זה בדרך כלל מספיק עבור המשתמש הממוצע.

לאחר שגרמנו לכך שהתחנות יגלשו דרך שרת ה-Proxy, אפשר להתחיל לטפל בתעבורה עצמה. לדוגמה:

- לדאוג שהגלישה תעבור סריקת אנטי-וירוס.
- לדאוג שאנשים לא יגלשו לאתרים מפוקפקים.
- לנטר ולהתבונן באיזה אתרים המשתמשים גולשים.

אנו מקבלים כאן תוספת הגנה מפני עולם האינטרנט (שבסך הכל הוא לא מזיק ומאוכלס על ידי אנשים עם כוונות טובות בלבד) ומסייעים לעובדים לגלוש בצורה חכמה באמצעות הפעלת שיקול הדעת במקומם.

3. **אימות** – באמצעות השמת שרת Proxy ניתן לקבל תמונת מצב יפה וברורה של אילו משתמשים גולשים לאן, מה האתרים הפופולאריים, ואפילו על מה הולך הרוחב פס – נתונים שהרבה יותר קשה לנתח ללא שרת Proxy (כתובות IP דינאמיות).

אז איך "מכריחים" משתמשים לעבור דרך Proxy?

השלב הבא אחרי ההפנמה שמדובר בשרת חשוב והתקנתו, הוא לגרום למשתמשים לגלוש דרכו. אני מדלג באלגנטיות על שלב ההתקנה כי המאמר עוסק בהבנה של הנושא ולא בהתקנה של מוצר כזה או אחר. אציין חריגה קטנה ואגיד שמי שבכל מקרה רוצה "לשחק" עם שרת שכזה, שיעיף מבט באינטרנט על Squid- שרת Proxy בקוד פתוח, שנפוץ מאוד, נוח מאוד לשימוש, ויודע לנבא תוצאות כדורגל ברמה גבוהה מאוד.

השיטות הנפוצות להסתת התעבורה אל ה-Proxy הן כדלקמן:

1. **קובץ PAC** – קובץ JS קטן היודע להפנות את הדפדפן (IE) לכיוון ה-Proxy. היותו קובץ JS מאפשר גמישות גבוהה בניתוב, כגון אפשרות של איזון עומסים ויתירות גבוהה (יתירות = מילה יפה בעברית למילה redundancy בלעז). את הקובץ ניתן לכפות על הדפדפן על ידי שימוש ב-GPO. החיסרון הגדול בשיטה זאת הוא חוסר יכולת השליטה בדפדפנים האיכותיים יותר(firefox, chrome, wget וכו').

2. **http_mapped** – שיטה שהייתה נפוצה בעבר ופחות נתקלים בה כיום . הרעיון הוא שברגע שה-Firewall מזהה פניה החוצה לאינטרנט בפורט 80, הוא מנתב אותה לשרת ה-Proxy במקום. השיטה די נדירה היום בארגונים, אולי משום שהשיטה הבאה די החליפה אותה.

3. **WCCP** – פרוטוקול ניתוב מבית סיסקו, אשר זלג עם השנים גם לחברות המתחרות, שמטרתו ניתוב התעבורה ב-L2 או L3 לכיוון שרת Proxy. האותיות בשם מסמלות web caching communication protocol. הפרוטוקול יעיל יותר מזה שהוזכר בסעיף הקודם כי הוא כולל בתוכו מנגנוני יתירות ושרידות מובנים.

4. **Transparent Proxy** – תצורה נפוצה מאוד בארגונים כיום. ניתוב התעבורה פיזית דרך השרת, שבדרך כלל יישב כ-Bridge. אמנם התצורה לא בדיוק עונה למינוח Proxy, היות והבקשה מתבצעת בין הקליינט לאתר – אך לרוב יש לשרתים אלו את אותן היכולות כמו לשרתי ה-Proxy האחרים.

איך המנגנונים השונים עובדים?

כפי שהוזכר, ישנם כמה וכמה תהליכים שיכולים לעבור על תעבורת ה-HTTP. מנגנונים אלו אמנם שונים ממוצר למוצר, אך ברוב המקרים הרעיון הכללי נשאר דומה:

- **סריקת Anti Virus** – אחת התרומות הגדולות של ה-Proxy, כזכור, היא יכולת סינון התוכן המגיע מפני מזיקים. כאשר המשתמש מוריד קובץ מהאתר, שרת ה-Proxy סורק את תוכן ההורדה ובודק אותו אל מול חתימות ה-AV. אותם מנועים מוכרים של Anti Virus מתחנות העבודה הם אלו שסורקים ברמת ה-Proxy. ישנם מוצרים שמעדיפים להוריד את תוכן הקובץ כולו אל שרת הסריקה ולאחר מכן להעביר אותו למשתמש, וישנם כאלה שמחלקים את הקובץ לחלקים וכל חלק שעבר סריקה מתחיל לעבור למשתמש. מוצרים אלה משתמשים ב-trickling (טפטוף המידע) על מנת לשמור על שקיפות מבחינת המשתמש.

- **URL Filtering** – בכדי למנוע מהמשתמשים לשחק פוקר כל היום במשרד או לסחור בנשק מתחומי החברה, נעשה שימוש במסדי נתונים שמקטלגים את האינטרנט. זה נשמע קצת מופרך לנסות לקטלג את כל האתרים באינטרנט, אך ישנן חברות שמתמחות בנושא והרוב נעשה בצורה אוטומטית בהתאם למילות מפתח למשל. מכיוון שיש מסד נתונים הכולל קטלוג ברמה כזו או אחרת של כל אתרי האינטרנט, ניתן לאכוף מדיניות האוסרת על גלישה לאתרי פורנו לדוגמה.

- **DLP** – מניעת זליגת מידע מהארגון. נושא שמתחזק בשנים האחרונות, ומספק יכולת למנוע העלאה לרשת של קבצים בעלי תוכן רגיש כגון מספרי ת.ז, פוליסות ביטוח, קורות חיים וכדומה. הקבצים שמועלים מושווים אל מול תבניות ידועות מראש ואל מסדי נתונים ארגוניים, בכדי לוודא כי הקבצים לא מכילים מידע פנימי.

- **Authentication** – מכיוון שבכדי לנהל מדיניות ע"ב משתמשים, ובכדי לקבלת תמונת מצב אמיתית של מי מהמשתמשים גולש לאן – צריך לבצע הזדהות מול שרת ב-Proxy.
ישנן כמה גישות לביצוע ההזדהות, כאשר הנפוצה ביותר והיעילה ביותר היא שימוש ב-NTLM על מנת לגלות את שם המשתמש. שיטה נוספת כוללת מיפוי של כתובת IP לשם משתמש ב-Login, או הזדהות ישירה מול ה-Proxy.

איך מתמודדים עם תעבורת SSL ?

בכדי שאנשים לא יצותתו כאוות נפשם לתעבורת HTTP, הוכנס לתמונה פרוטוקול HTTPS. מטרת הפרוטוקול היא יצירת תווך מאובטח שרץ על תקשורת לא מאובטחת. במילים פשוטות יותר – תעבורת HTTP בצורה מוצפנת. משום שהתעבורה מוצפנת, שרתי ה-Proxy לא יכולים לראות את המידע שעובר בין התחנה לבין שרת האינטרנט ואינם מסוגלים לבצע סריקה או קטלוג של התוכן. הדרך הנפוצה להתמודדות עם בעיה זו נקראת SSL Termination. בצורה מאוד דומה להתקפות man in the middle, שרת ה-Proxy יחתוך את ה-session המוצפן, יעבד את הדף ולאחר מכן יחזיר אותו לקליינט. איך זה יראה למשתמש הקצה? ישנן שתי אפשרויות:

1. לאחר שה-Proxy יסיים לעכל את המידע, הוא יצפין מחדש את ה-SESSION אל מול המשתמש. משמע- המשתמש עדיין יראה את התעבורה כמאובטחת ואם הוא סומך על הסרטיפיקט שהוצג לו, העניין יהיה תקין מבחינתו. זאת השיטה שבדרך כלל משתמשים בה היות והיא מאובטחת יותר ומונעת התקפות MITM נוספות בין שרת ה-Proxy למשתמשים. הסרטיפיקט שמוצג למשתמש יונפק על ידי CA ארגוני או CA אמין אחר.

2. השיטה הפחות מאובטחת גורסת כי לאחר שה-Proxy פותח את תעבורת הHTTPS-, הוא מעביר את המידע הלאה למשתמש בתצורת HTTP. המשתמש מבחינתו גולש לאתר HTTP רגיל. כאמור, בשלב זה ניתן להאזין לתעבורה ולכן השיטה פחות נפוצה לשימוש.

למה נשים לא מצליחות לעשות Reverse Proxy?

אם אמרנו ש-Proxy מיועד לכך שכל מי שגולש לאינטרנט יצא דרך השרת המיועד הזה, Reverse Proxy הוא בדיוק הפוך. כל מי שמגיע מהאינטרנט לאתר שלנו יצטרך לגלוש דרך שרת ה-Proxy שלנו.
ה-Proxy ההפוך מיועד לכמה מטרות עיקריות:

- שימוש ב-load balancer על מנת לאזן עומסים בין כמה שרתי WEB, והוספת יתירות לאתר. המשמעות היא שאם שרת אחד נופל, שרת אחר יספק את השירות במקומו- הישרדות.

- הגנה- שימוש ב-Proxy הפוך מספק לנו הגנה בכך שהוא זה שבעצם נמצא מול האינטרנט, ולא שרת ה-WEB בעצמו. אם למשל שרת ה-WEB רץ על IIS לא צריך לדאוג (צריך אבל פחות) מפגיעויות נפוצות שלו, כי הוא לא פתוח ישירות מול האינטרנט.

- פעמים רבות נעשה שימוש ב-Web Application Firewall, נושא ששווה להקדיש לו לפחות מאמר אחד משל עצמו. בכמה מילים, ה-FW הזה בודק את התעבורה ב-L7 ומוודא כי התעבורה שמגיעה אל האתר היא לגיטימית ולא בעלת אופי התקפי. למשל הגנה מפני SQL Injection, ה-WAF ידע להבחין שבשדה טקסט שאמור להזין שם או שם משפחה, כנראה שאסור שייכנס הסימן "=".

- ביצועים – לעתים קרובות ה-Reverse Proxy לוקח על עצמו תפקידים צורכי משאבים כגון הצפנת התווך או ביצוע caching, ובכך יכול להקל על שרת ה-WEB בעצמו.

Anonymous Proxies

שרתי Proxy בעולם התחילו לקבל שימוש נוסף- עזרה למשתמשים מכל קצוות תבל לגלוש בצורה אנונימית. ישנם שרתי Proxy חופשיים המאפשרים גלישה דרכם, ועל ידי כך בעצם מסתירים את כתובת ה-IP ממנה הם מגיעים. בנוסף, חלק משרתים אלה מצפינים את התעבורה בכדי שלא ניתן יהיה להאזין לתעבורה. הבעיה הקטנה עם אתרים אלה היא שהם בדרך כלל איטיים ומסורבלים. הבעיה הגדולה באתרים אלה היא שהם מפוקפקים עד מאוד ורובם נועד על מנת להאזין לתעבורה שעוברת דרכם, ולשמור את השמות והסיסמאות של הגולשים במרתף חשוך.

פרויקט מפורסם מאוד של גלישה אנונימית נקרא Tor (The Onion Router), למרות שיוצרי הפרויקט מכחישים בתוקף. הפרויקט הוא צאצא של onion routing – פרויקט אחר שמטרתו הייתה הסתרת נתיבי הרשת. Tor לקח רשת של מתנדבים, שיחד יוצרים המון מסלולי גלישה פוטנציאלים. כל אחד שינסה לגלוש דרך רשת Tor יועבר דרך רשת מסועפת שיהיה קשה מאוד להתחקות אחר המסלול בה. אך גם Tor אינה אנונימית לחלוטין, ולמרות שהיא כביכול אופציית הגלישה הבטוחה כיום- אם משקיעים אנרגיה ניתן להתחקות אחר מקור הגלישה. השמועה הרווחת היא שממשלת ארצות הברית מנטרת את כל רשת Tor מקצה לקצה.

סיכום

שרתי Proxy מאפשרים שליטה ובקרה על תעבורה הנחשבת למסוכנת. הם נותנים יכולת לאכוף גישה לאתרים ע"ב מדיניות ארגונית ומשמשים למניעת זליגת מידע מארגונים. קיימים עשרות שרתי Proxy חופשיים למשתמש הביתי, וחלקם אף מגיעים בתצורת תוסף לדפדפן. אמנם רובם "מסוכנים" לשימוש, אך הפעלת שיקול דעת נכון יכול להפוך אותם ליעילים במקרים מסוימים.