dns

- הכניסה לרשת לחנות אינטרנט: google.com בודקים בקובץ ה hosts ואם יש השמה לדומיין שהכניסה, אם לא שולחים query dns לרזולבר שמוגדר ב resolv.conf והוא מבצע בקשה רקורסיבית/איטרטיבית ל root server
זה root שולח את הבקשה לשרת שמתאר את ה TLD (.com) והשרת ההוא שולח לשרת שמתאר את כל הדומיינים של google והשרת ההוא שולח לשרת שמתאר את הסאב דומיין www
וגם יש רזולבר של כתובות IP והוא חוזר כל הדרך לרזולבר שמבקש אותה אלינו.
(ניתן את הפלוף הרקורסיבי, האיטרטיבי הרזולבר שולח לכל שרת בדרך והשרת נכבה אומר לשרת הבא)
- אחרי שעשים handshake יש לבנות סשן עם הדומיין.
- אחרי handshake כלשהו שבו מסכימים על פרוטוקול הצפנה. וגם הקליינט יודע מפתח פומבי, מצפין אותו עם הפרטי קי של השרת ושולח לשרת. והם מתקשרים עם הסימטרי. (לא ידוע כרגע, נראה בהמשך)
- פורמט URL:
https://shlomi.cloud.com/user/shlomke/index?...
פרוטוקול | תת דומיין | דומיין | path | סאב דומיין | פרוטוקול

- בשרת עצמו הוא בודק רשימה של root servers ושולח לכולם כדי לבדוק מי מהם קיים, ובשביל מהם שלח לו הוא בקש ממנו את השרת הבא בתור.
- קובץ zone:

name	ttl	class	type
שם הדומיין	תאריך כדי שבקאש יהיה עדכני	מן הקורס (in, ch..)	

- query recursive - הבקשה נשלחה לשרת והוא שואל את השרת הבא ששואל את מה שאתה מחפש עד שיש רזולבר (או אין) ואז המשמעות חוזרת כל הדרך חזרה.
- " iterative - הבקשה נשלחה לשרת והוא מחזיר את השרת הבא שצריך לשאול ואז שולחים בקשה לאותו שרת והוא מחזיר את השרת הבא שצריך לשאול.
- zone transfer - העברת מתוך של כלכך מהקלטות בין שרת אחד לשני, השרת שצריך את הקובץ יכול לבקש אותו. מקשת AXFR

- ARP: מישהו שונל בברודקאסט מה ה mac של ה IP הזה.
כל הנטס שיקבלו את ההודעה בודקים אם הוא מה ה IP הזה
מי שזה הוא מחזיר לשולח reply ARP עם ה-MAC שלו. (מחזיר src ip ו-dst mac)
ואז הוא מעדכן את זה ב ARP cache שלו.

כלים:

- WAF - FW שמגן רשכבות העליונות של osi, על אפליקציות web (בקשות http) אבל עם פעולות שאפשר להעביר אישית לפי הצרכים. זה reverse proxy ~~עם יכולות הפעולות משפרות לעומת מתאר רגיל (בהמשך אפשר)~~.
- IPS - ניטור יותר מדויק, לא מבטן http אלא dns, Rdp, ssh, ftp, smtp וגם. נעשה בשכבות 4-3 בדר', עם פעולות וגם תגובות.
- DDoS - מונעים loss מלא והגבה של מידע בדיוק מהאירגון. מקבל המבאים וכו'. אפשר לשים אותו גם לפני שרת מייל כדי שלא יוציאו מילים מידע רגיש.
- relay mail - לשים לפני שרת מייל כדי לסנן מיילים בבטחה ולמנוע ספאם וכיוצ'.
- IDS - כמו IPS, בדרך הטוב ביותר ומשווה עם תגובות ומצהה מלוגים, סריקות פורטים ועוד.
- EDR - מגן על endpoints, אולם בלא ל agents ומצהה אירועים לי תגובות וגם אירועים שלא קרו לי כידוע. יכול להעביר את הבלגן ל SIEM / SOAR אם רוצים.
- NDR - מגן על סוגים נוספים של endpoints כמו שרתי מדפסת, מצלמות וכו'.
- SOAR - מקבל אירועים מגוון כלים ובונה תיק מידע לפי אירוע ומאפשר לאנליסט לנהל בדרך נוחה. וגם מאפשר לשלוח אוטומציה לאירועים מוגדרים.
- NGFW - FW שמגן על הקליינטים שיוצאים לאינטרנט, עשה filtering לרשת, AV, ולפעמים גם IPS. זה סוג של proxy
- לפעמים שניים כלי לפני ה FW כדי שיוכל לדעת איזה תקיפות נסו אליו ושהם יודע לא יחסום אותם. ואז נוכל לשנות להעביר את התנועה לאתר שבדוק כמו שלנו אבל זה honeypot
- CDR (reconstruction של content disarm) - לא לקבל קבצים, בודקים שהם לא יהיה שיכול לשנות תקיפה, initial access. יכול אתר של הקבוצה לי לינקים ודברים כאלה.
- MDR - שירות בלתקן ורספונס לאירועים אנשי soc ו- IR ד/24 שמגיבים מהאירוע. תוקפים Alerts, אנשים מיומנים לאירועים, remediation, threat hunt
- FW stateful - שומר את ה-state של מי תיזר פעל וכך יודע לזהות אם קיבלת לשלח רספונס לפני שהיה בקש או שמישהו שלח חצי אחרי שכבר קיבל syn-Ack
- ~~FW עסקים - שכבה ד לפי פרוטוקולים שבה ד. אפשר לשים אותו מול שרתים~~
- FW לא יכול להבין מה זה שלא אבר הנתון (מס קובץ, סיומת וכו').
- UTM - כלי שמרכז המון כלי הגנה: FW, AV, מסר anti spam ועוד. (unified threat mgmt)
- Firewall limit - מגבר את ה-endpoints מאיר אחרים בגלל ומלוים שם הרשתות מקום מרכזי. שיש סרטרים (מקליטים) דרמת קבע, שנמצאים בחוטים, ארנטים, FS ועוד.

כללי:

- אין לנו הרבה זמן אמת לתקיפת המחדר לתשדורן של נושאי דרגים (את השמנים והמוסמכים וה-CA) - לכן מיקום גיאוגרפי של המחדר, דרך תוקפים משתמשים ב מסע (לכי לאיזה מקום ש אין תוקפים)
- DDoS - מניעת שירות - אי אפשר לגשת, אומר לשכנע שיש מישהו שמנסה לתקוף אותו. כדי ה שת נתל
- TLS - handshake v1.2 - קליינט שולח hello ומשיבים בשאלות שמתאים אליהם (ופרוטוקול) הפסה
  השרת שולח hello בדיוק אם ה - cert שלו (ההצהרה קי שלו) שמחוברת לפרטי הפרטי
  הקליינט שולח שיוצרים את המפתח של המחדר premaster secret והקליינט מצפיר ביחד מהרשומים ופרמטרים
- handshake v1.3 - משתמשים כן בדיפי הלמן שזה הרבה יותר טוב מ-RSA ש ב-1.2 מאשר
  קליינט שולח את הרשומה
  ומשיבים בשאלות
  הכל בלוגים מוצפנים.
  הנתונים אחר אינה שולחים cert ושלח ויצר פוג פוג
  נשלח גם לקליינט שולח DH בהמשך
  נאל כל מהם לוקח את המפתח לסשן ע"י הצרוף של
  לכן זה הרבה יותר קצר וצרכי (פחות הרבה מקום מ)
- RSA - הפרוטוקול בהרבה יותר קצר דרשים (אבל שהוא מנצל את הבאלנס של השרת)
  ואולי יותר טוב כי הם יצרו את הפרוטוקול בלעדיהם. הם יצרו את הפרוטוקול שלהם הבאלנס.
- סרטיפיקט - לתלק השם לבמה תחי בשנת. למשל השם של שבל נשי השרת של 25/ שם CSR
- signatures - CA חותם על cert עם המפתח הפרטי שלו, זה יוצר נצפין הוא שם המפתח הפרטי ושם שם, ואז מי שנכנס לי לאתר הדפדפן שלו דרך שמשתמש דקרוט עם הציבורי של ה CA זה תואם להוא של ה - CSR
- socket - קובץ שמשמש לתקשורת בין קליינט לסרבר. לכל קליינט נפתח socket חדש.
- application - נפרסת את הדטא ומשה אינו מה שהאפליקציה אמורה לעשות.
- presentation - הצורה שזה המידע נוצר (חוצנו לשל). היום זה מהלך ודאי ה אומ כדר.
- session - של האפליקציה עדיה. זה שורם שהתנתק הינטרנט ואנחנו מרגלן יציב, הטרגלן מאמין אותה נקודה, אפלו שה socket התנתק. כי האפליקציה מתחברת מחדש.
- forward ip - הם לאשר להתקלות שלא מעבירת אליך לשרת הזה, או להפיל אותן.
- https - גם יש https תקשורת מאוטחת ומוצפנת, ואינטרנט ע"י cert (בפרוט לאלה).
- NAT - הצפנו לשים את מאחי/מחל כי זה פרטית חשוב, ומדבר כאוטרס מיה. אם תשלח פרוטוקול אחר, גם אם תכתוב שזה פורט 53 לשל, הראוטר יחה שזה מאבטחת מחל
- stack ip - אוסף הפרוטוקולים של ip/tcp בתוכנת הבלה נמרצת, אז ארכיב נמוכים (ביב).
- מחאי - פרוטוקול בקרה לשכבת הרשת, לדוגמת תיבור, ניתוב קשר יעד (redirect) ועוד.
- NAT/PAT - 10.10.10.2:80 = 63.63.63.1:8001 10.10.10.1:80 = 63.63.63.1:8000
- little indian - 0x15234567 הכתובת הקטנה יותר תאחסן את הבתים הגדול יותר
  " big -

67	45	23	15
15	23	45	67
0x1000	0x1001	0x1002	0x1003

InfinityLabs R&D

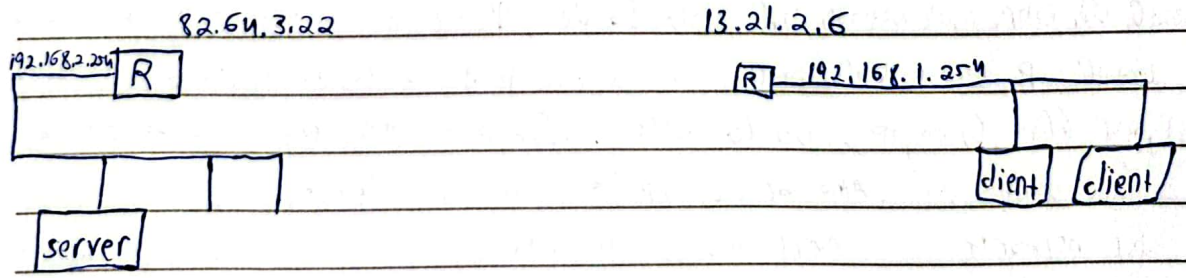
TCP:

- keep alive - אפשר לשמור על ה session של TCP גם כשלא נשלחו פאקטות פרק זמן מסוים.
- flow control - שליטה בקצב השליחה ע"י sliding window.
- sliding window - המקבל מודיע לכל פאקטה את גודל החלון הפנוי שלו, והשולח יכול לשלוח רק עד אותו גודל מבלי לחכות לאישור (אפילו חיובי)
- timestamp - שדה שעוזר להתמודד עם wrap around (כשהגענו למספר פאקטה של 65535 ואז מתחילים שוב מ-0) ולא להתבלבל שגרמה שקבר קיבלתי את המספר פאקטה הזה.
- congestion control - אם יש congestion הוא מקטין את גודל החלון שלח השולח, או אפס אותו. וכך מונע עומס נוסף ברשת. כך - קטן אם יש פאקטות עומס הגיהה של פאקטות יכולה להיכשל.
- הגודל חלון של המקבל הוא מה שמגדיל את השולח, והגודל נמדדים כל פאקטה
- ack שולח - כדי לחסוך במספר הפאקטות, המקבל יכול לחכות שישלחו לו 3 פאקטות לפני, ורק אז להשיב Ack על שניהם. 1000 + 100 + 100 = ack 1200
- retransmission - השולח שומר עותק של הפאקטות שנשלחו בקווים, ואם לא קיבל שום Ack עד הזמן מוגדר.
- + handshake - קליינט שולח פאקטה עם דגל מנע עלון ו-isn רנדומלי (2000 למשל) שרת מחזיר עם דגל מנע ו-Ack, isn רנדומלי (5000) והמספר Ack יהיה 2001. קליינט מחזיר Ack עם מספר Ack 5001

וזה דרך שהם יודעים אם הקווים 2000 והפאקטה 100 בייט ה Ack יהיה 2102

- meltdown - מאוד מעניין ל TCP עם מה אם הפרוטוקול Ack הוא יגיע שהוא מודע לזה שם פאקטה מידע נשלחת שוב וזה יגרר בעיות ו מעמיס (כי אז הפרוטוקול יבדוק את הזמן שלו מול המידע הוא עדיין יהיה קטן יותר וכו')

מקר:



10.2.0.1	VNic	VNic	192.168.1.13
192.168.2.16	NIC	VNic	10.2.0.2

הראוטר יעשה דיקפסולציה לפאקטה ואז ישלח את הבקשה לגוגל עם מאסקרייד.

packet to google.com

src:	192.168.1.13
dst:	8.8.8.8

src:	10.2.0.2
dst:	82.64.3.22

encapsulating packet

כללי:

- tracerouter - שולח פאקטה עם TTL 1 ואז הראוטר הראשון מוריד אותה ל-0 ומחזיר icmp שגה לא הוגן, ואז הוא שומר את אותו ראוטר ושולח שוב עם TTL 2.
- בקבלת אחסון על פרוטוקול הנתבים, בדרך כלל הם הם יעד וחוזרים בראוטינג טיבל דרך אלה אינטרפיס יצאים לאותו IP (ב- MPLS לעל G בתעבורה יעד ידוע דרך ה - VNIC)
- AS - אוסף כתובות בעלי מאפיינים משותפים בתיבה, יכול להכיל המון רשתות.
- RIP - מאפשר רק עד 15 hops, מיועד לרשתות קטנות. distance vector.
- BGP - פרוטוקול נתיב בין AS ובין AS (EBGP \ IBGP)
- OSPF - לא יעיל ברשתות גדולות כי G neibor מעדכן את המידע 6 פעם שיש שינוי והם מעדכנים את המידע שלהם. אל ברשתות גדולות זה גורם לעומס רב.

גם הוא לא עדיף לגרף עם נתיב שרוצה למשל שמ המעגלה עליה קודם לראוטר נמצא. ב EGP כן אפשר.

- AS - PATH - ב - BGP אפשר לראות דרך איזה AS's צריך לעבור כדי להגיע ליעד נמצא. וזה גם מונע loop כי אם ראוטר מוצא את ה AS שלו בתוך ה PATH הוא לא יקבל את זה.
- sysmon - נותן system events דומים (יצירת קבצים, שינויים ברג'יסטרי). מלווים הרבה אינפו.

positives false - קבצה התראה, אבל זה היה false

negative false - לא קבצה התראה על אירוע אמיתי.

- חלון - חלשה שמאפשר לנקב, בין אם זה נגד קונפיגורציה, סיסמא חלשה, שימוש בתוכנה לא עדכנית.
- sniffing - בפרוטוקול לא מוגן וקיבוע סוויץ' ראוטר/שרת ואז מידע.
- A=3 דפאלט - יוצר אובייקט מסוג חדש שווה ל- 3 , A 1 יהיה פוינטר לאותו אובייקט.
- slide NOP - הנקל לא צריך לעשות לזה דיבוג התנהגות תקינה, אז הוא מכניס הוראה NOP ומזיז שם את הקוד שלו, וכך גם זה נגד אנשים באמצע ה NOP זה יחל לא כלום ואז יבצע את הקוד שלו.
- bit locker - הצפנת דיסק, full disk enc, דיסקן AES. כולל דוגמאות, מיגוקים ו Sets
- secure boot - בודק התימה דיגיטלית של כל תוכנה לפני שמריצים אותה, מונע שלא טומפרה
- TPM - בא לוקח נשמש בו לאחסן נתונים בצורה בטוחה מונע שרצים לא טומפרו
- DORA DHCP - Discovery בחבקאט, offer של כתובת IP, מאדנט, בקשת request אחרי שיבא שה IP לא תפוס עם ARP. ואז Ack
- טבלת ניתוב: dst network next hop interface
- אם אין בטבלה לא תקבל unreachable host
- redirect icmp - הראוטר קבל פאקטה והוציא אותה לאותו אינטרפיס שנכנסה. (ניתן לא יעיל)
- MPLS - יצירת רשת וירטואלית בפנים עם הרשת הגדולה, ואז מכל של איזון יכול לעבור בהבטחה הרשת הווירטואלית ולבאות לרשת התיבונית דרך הרשת של האיזון בריבוע.
- רגיסטרים שמתחילים ב E - 32 ביט. (extended), ב- R - 64 ביט. (register) אם אין קידומת (x) - 16 ביט. אם אין קידומת (l) - 8 ביט

web:

- מסמך - הגדרת הבעיה" של הדפדפן, מגדיר איזה אובייקטים אפשר להכניס ומתי.
- CSRF - שולח לינק עם בעיה מרגיש התקפה, כשיוצר מאומת באתר לוחץ על הלינק בה כדי שנגלו אל הבעיה בלי ידיעתו. (למשל מה שעשינו בלתיבה על הלינק בה כדי שנבדקו סיסמא אוטומטית)
מניעה: להוסיף tokens CSRF לכל בקשות changing state, וכך תוקף לא יוכל בשום לינק כי הוא לא יוכל לחסל את הטוקן.

i • XSS - הכנסת סקריפט לאתר לגרום לאתר לבצע אותו, וכך יוצר שנכנס לאתר לשפוך. שם מיוצר הבר אתר. השלם יקל ושה לא יה
1) stored - נשים את הסקריפט לאתר ומבנו (בתגובות למשל) DB וכשיוצר נכנס לבלוג הסקריפט יתוכנת לו ורוץ.
2) reflected - שולח ליוצר לינק עם סקריפט שישלח בקשה לאתר והאתר יעשה echo לקוד לאותו יוצר.
3) dom - שהאפליקציה מקבלת client side אינפוט ומקבלים לא בטוחה ומתבצע אותו קוד ב-Dom
מניעה: שימוש ב API בלתי, אינפוט לא יכנס לדבר השרת, אסקייפ לתווים מיוחדים. ובמשך המאמר מדר. (שיוכנו מיוחדים אחרים)

- SSRF - כמו CSRF שמנצל קרדן שהוא קיים, וכאן מנצלים את השרת עצמו ומורים לו לבצע בקשה לנכסים שהוא לא נגיש וכך לשלם לתשלם מידע רגיש. דוג': אוטוקנה שאתה מכניס לה URL והיא מבצעה לך את הבקשה של אותו אתר, ואתה שם לה כתובת של הפנים שלה מניעה: לא לאפשר לבקש מהשרת, או לתת לשרת יוצר בלי הרשאות רגישות וכך הוא לא יוכל לבצע בקשות שיתשפו מידע רגיש. לא להכניס את האינפוט של היוצר לשרתה שיתבצע בקשות. בטוח בפרוטוקולים בדומים ל SSRF (FTP, smtp). מניעה נוספת לדומיים הרגישים מהשרת בחיצוני
WAF, user access control, patch mgmt
ניהול: לנטרל (מה של WAF), ניטור השרת (ids\ips), edr, sifm

ii • XSS נגמן לתוקף גישה לכל הפרטים של הדפדפן, הוא יכולה להתרגל מנגש קוד שחיצוני לאתר, להתקנות פיוצר שמתקף ולבצע כל בעיה שאותו יוצר יכול לבצע, או מידע שהוא יכול לקרוא, לעדכן חשבון (לפנה)
- ניהול ובדיקה בגלגלים ל - XSS אפשר ירוצים, ואפשר עם scanner חלוש משו suite פרוש
מניעה: לבלבל אינפוט מהיוצר בגלל שאתר בהמשך שאתה מקבל כאינפוט לפיך!
ראשית encoding לדעתו שהיוצר הכניס בשאלה יחזיר אותו ל response, כך שלא יהיה active