זיוף שידור של ה-Ring Doorbell

מאת אור צינגיסר

הקדמה

מטעם עבודתי כחוקר אבטחה בדוג'ו בולגארד את מרבית יומי אני מבלה בחיפוש אחר חולשות IoT. ב-27 לפברואר הדגמנו על הבמה ב-Mobile World Congress Barcelona 2019 את הפריצה שלנו לשידור האינטרקום החכם של Amazon, אשר זכתה לסיקור תקשורתי רחב היקף. לצד ההדגמה החיה המוצלחת פרסמנו כתבה בבלוג החברה בה פירטנו אודות המתקפה. מאמר זה הינו ברובו תרגום הפוסט וכן מרחיב בנושא השמשת ההתקפה, אשר מאפשרת האזנת סתר לשידור וידאו ושמע מהאינטרקום וכן הזרקה של פקטות אלו כרצוננו. דרישת הקדם היחידה לביצוע המתקפה היא להיות עם רגל באותה הרשת כמו הטלפון בו האפליקציה מותקנת.

נתחיל בהקדמה על המוצר המדובר: Ring היא חברת בת של אמזון ועוסקת במוצרי IoT כמו מצלמות פנים, אינטרקומים, תאורה חכמה ועוד. ה-Ring Doorbell מקובע מחוץ לבית ומחליף את הפעמון הקלאסי. המשתמש מתקין את אפלקציית Ring ומקבל התראה כאשר מישהו מזמזם בפעמון. המשתמש יכול לראות ולשמוע את השידור מה-Doorbell ואם יבחר לענות, יוכל לדבר עם מי שמחוץ לדלת דרך הרמקול המובנה. בהנחה שמותקנת גם מערכת מנעול חכם כדוגמת Amazon Key, ניתן לפתוח את הדלת באמצעות האפליקציה הרלוונטית, למשל בשביל לפתוח למנקה או לשליח של אמזון.

כפי שראוי לעשות, התחלנו את המחקר על ידי סריקת ידיעות ופרסומים קודמים אודות המוצר. אכן, ב-2015 נמצאה חולשה שמאפשרת על ידי לחיצה על כפתור ההתקנה ב-Doorbell גניבה של פרטי ה-WiFi של הבית. הבעיה הייתה כפי שקורה רבות בעולם ה-IoT, בשאריות מה-SDK שמציעה חברת הצ'יפים המייצרת את ה-SoC שבשימוש. דרך הרשת WiFi שה-Doorbell פותח במצב התקנה אפשר לגשת לעמודי אדמיניסטרציה אשר כוללים את פרטי ה-WiFi. שווה להזכיר ש-Ring מהרה לבצע עידכון תוכנה על כלל המכשירים בתוך החלון Disclosure שהוקצה.

מבוא לתעבורת זמן אמת

כדי להבין כיצד ממומשת המצלמה נתחיל בהסבר קצר על איך עובדת תשדורת זמן אמת: כאשר יש טריגר, בדרך כלל פיזי על ידי חיוג או זמזום, נשלחת הודעת הקמת שיחה בפרוטוקול הנקרא SIP (Session Initiation Protocol). ההודעה, שעוברת מעל UDP, מעבירה מספר פרטים חשובים - מזהה המחייג והמחויג, מזהה השיחה, סוג קידוד ה-stream, פורט פנוי שישמש להעברת התוכן ועוד. לאחר הודעה זו שנקראת INVITE, המחויג שולח הודעת SIP TRYING להבהיר שהוא בתהליך עיבוד הבקשה, ולבסוף SIP OK שמסמן שהערוץ מוכן, בו כתוב הפורט אליו יש לגשת. על מנת לפתוח את פורט ה-Data בסביבת NAT השירות מוציא פקטת STUN (Session Traversal Utilities for NAT) אשר מפעילה הפניה של פורט מסוים. לאחר מכן נשלח ACK סופי בדומה ל-TCP וניתן להתחיל לשלוח הודעות Data על גבי הפורטים שנפתחו. נציין ש-SIP הוא פרוטוקול ורסטילי שתומך במספר משתמשים מאחורי מרכזיה אחת (ריבוב מעל IP יחיד) וכן בניתוב דרך פרוקסים בדרך. בשלב זה משתמשים ב-RTP (קיצור של Real Time Protocol Transport) בשביל להעביר את ה-Codec (שיטת קידוד השמע / וידאו) שנבחר, פרוטוקול התומך ב-Sequencing וריבוב מקורות שונים.

ניתוח פעילות רשת

לאחר הבנת ייסודות התקשורת זמן אמת, נתמקד כעת על תעבורת הרשת שמתרחשת ב-Doorbell וכיצד השידור מגיע לאפליקציה. הארכיטקטורת תקשורת ש-Ring בחרה היא להשתמש ב-AWS (Amazon Web Services) כשרתי תמסורת. האפליקציה והמכשיר משדרים לענן ומקבלים ממנו את השידור של המקביל להם. תחילה הפעמון נלחץ ונשלחת בקשת REST לענן, אשר שולח notification לאפליקציה. האפליקציה וה-Doorbell שולחים במקביל SIP INVITE לשרת, כל אחד עם פורטים מוקצים משלו. ה-Doorbell עושה זאת ב-SIP עם headers מיוחדים ש-Ring הוסיפו:

```
> Ethernet II, Src: Tp-LinkT_15:0e:f7 (14:cc:20:15:0e:f7), Dst: D-Link_6a:3d:76 (5c:d9:98:6a:3d:76)
> Internet Protocol Version 4, Src: 192.168.36.129, Dst: 18.197.187.54
> User Datagram Protocol, Src Port: 15063, Dst Port: 15063
v Session Initiation Protocol (INVITE)
  > Request-Line: INVITE sip:5nlmmc40em6gm-2037kgh65bhbni@18.197.187.54 SIP/2.0
  v Message Header
    > Via: SIP/2.0/UDP 192.168.36.129:15063;rport;branch=z9hG4bK705780083
    > From: <sip:f4844c55c577@ring.com>;tag=TUF0000000020030F38
    > To: <sip:5nlmmc40em6gm-2037kgh65bhbni@18.197.187.54>
      Call-ID: KIYQRJDLGONFQMWPNXQJXZUDTJNYFINPK0000000020030F38
    > CSeq: 1 INVITE
    > Contact: sip:f4844c55c577@192.168.36.129
      Content-Type: application/sdp
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, MESSAGE
      Max-Forwards: 70
      User-Agent: Device/dpdv3/1.21.63
      Subject: Ding
    v X-SSRC-A: 0247647989
      > [Expert Info (Note/Undecoded): Unrecognised SIP header (x-ssrc-a)]
    v X-SSRC-V: 0704495836
      > [Expert Info (Note/Undecoded): Unrecognised SIP header (x-ssrc-v)]
    v X-Session-Hash:i9N27wjQZo3IKeToDvDQfXiQwwaPhLDg9mMRPRm/g6g=,2576
      > [Expert Info (Note/Undecoded): Unrecognised SIP header (x-session-hash)]
      Content-Length:  391
  > Message Body
```

[הודעת SIP מה-Doorbell לענן]

בתמונה רואים שימוש ב-X-SSRC-A, X-SSRC-V ו-X-SESSION-HASH, שדות שמרחיבים את הפרוטוקול אשר Ring הוסיפה. ככל הנראה מדובר בחתימה על הפרמטרים של השיחה (פורט מקור ויעד, IP מקור ויעד, מזהה שיחה) כדי שלא יהיה ניתן לשנותם, וכן מפתח ההצפנה שמשמש בשביל העברת ה-RTP שבאה לאחר מכן. נציין רק שהשיטה הסטנדרטית להעביר SIP מאובטח היא מעל TLS, ול-RTP יש וריאנט בשם SRTP בשביל שידור מאובטח.

לאחר הקמת השיחה ניתן לראות ש-Wireshark אינו יודע לפענח את תעבורת ה-RTP שעוברת:

ip.addr == 192.168.36.129

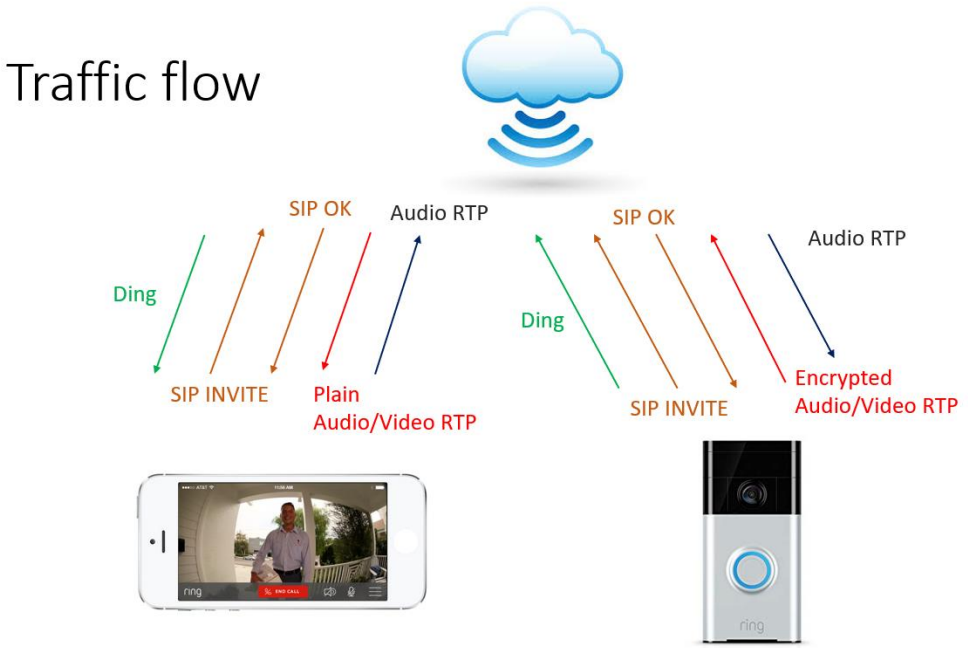
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
143	5.406392567	192.168.36.129	18.197.187.54	H264	600	35	PT=H264, SSRC=0x29FDC0DC, Seq=35, Time=30000, Mark PPS
144	5.420818303	192.168.36.129	18.197.187.54	H264	1331	36	PT=H264, SSRC=0x29FDC0DC, Seq=36, Time=36000 SEI
145	5.429096494	192.168.36.129	18.197.187.54	H264	1336	37	PT=H264, SSRC=0x29FDC0DC, Seq=37, Time=36000 Prefix
146	5.434674554	192.168.36.129	18.197.187.54	H264	1336	38	PT=H264, SSRC=0x29FDC0DC, Seq=38, Time=36000 IDR-Slice
147	5.443508968	192.168.36.129	18.197.187.54	H264	584	39	PT=H264, SSRC=0x29FDC0DC, Seq=39, Time=36000, Mark PPS
148	5.444573354	192.168.36.129	18.197.187.54	H264	1331	40	PT=H264, SSRC=0x29FDC0DC, Seq=40, Time=42000 EXT Unknown Subtype (16)
149	5.452881888	192.168.36.129	18.197.187.54	H264	1336	41	PT=H264, SSRC=0x29FDC0DC, Seq=41, Time=42000 End-of-Seq
150	5.463684223	192.168.36.129	18.197.187.54	H264	888	42	PT=H264, SSRC=0x29FDC0DC, Seq=42, Time=42000, Mark Reserved
151	5.488399791	192.168.36.129	18.197.187.54	H264	1331	43	PT=H264, SSRC=0x29FDC0DC, Seq=43, Time=48000 SPS
152	5.488453112	192.168.36.129	18.197.187.54	H264	728	44	PT=H264, SSRC=0x29FDC0DC, Seq=44, Time=48000, Mark End-of-Stream
153	5.491474228	192.168.36.129	18.197.187.54	H264	1331	45	PT=H264, SSRC=0x29FDC0DC, Seq=45, Time=54000 Slice-C
154	5.491516012	192.168.36.129	18.197.187.54	H264	888	46	PT=H264, SSRC=0x29FDC0DC, Seq=46, Time=54000, Mark EXT Unknown Subtype (29)
155	5.491539933	192.168.36.129	18.197.187.54	H264	1219	47	PT=H264, SSRC=0x29FDC0DC, Seq=47, Time=60000, Mark Reserved
156	5.494382503	192.168.36.129	18.197.187.54	H264	1331	48	PT=H264, SSRC=0x29FDC0DC, Seq=48, Time=66000 MTAP16 [Bad NAL Length]
157	5.506305783	192.168.36.129	18.197.187.54	H264	1320	49	PT=H264, SSRC=0x29FDC0DC, Seq=49, Time=66000, Mark IDR-Slice
158	5.508303475	192.168.36.129	18.197.187.54	H264	1267	50	PT=H264, SSRC=0x29FDC0DC, Seq=50, Time=72000, Mark Slice-C
159	5.515659724	192.168.36.129	18.197.187.54	H264	1331	51	PT=H264, SSRC=0x29FDC0DC, Seq=51, Time=78000 STAP-A [Bad NAL Length]
160	5.516410252	192.168.36.129	18.197.187.54	H264	1240	52	PT=H264, SSRC=0x29FDC0DC, Seq=52, Time=78000, Mark Reserved
161	5.528463180	192.168.36.129	18.197.187.54	H264	1331	53	PT=H264, SSRC=0x29FDC0DC, Seq=53, Time=84000 FU-A Start:SPS-Ext
162	5.539767242	192.168.36.129	18.197.187.54	H264	136	54	PT=H264, SSRC=0x29FDC0DC, Seq=54, Time=84000, Mark Reserved
163	5.539815603	192.168.36.129	18.197.187.54	H264	66	55	PT=H264, SSRC=0x29FDC0DC, Seq=55, Time=90000, Mark FU-A Start:MTAP16 [Bad NAL Length]

התוכנה חכמה מספיק להשליך את הצימודי קידוד-פורט שהוצהרו בשלב ה-SIP ל-dissectors העתידיים, אבל אפשר לראות שהתעבורה לא נראית הגיונית. הפירוק בעזרת codec של H264 (בשימוש ב-MP4) זורק שגיאות של Unknown Subtype, Bad NAL Length, Reserved ועוד מכיוון שהשכבה השביעית מוצפנת. לא הצלחנו לפענח את ההצפנה בעזרת הפרמטרים המיוחדים שעברו ב-SIP ולא הייתה לנו גישה ל-Firmware על מנת להבין כיצד היא מתבצעת.

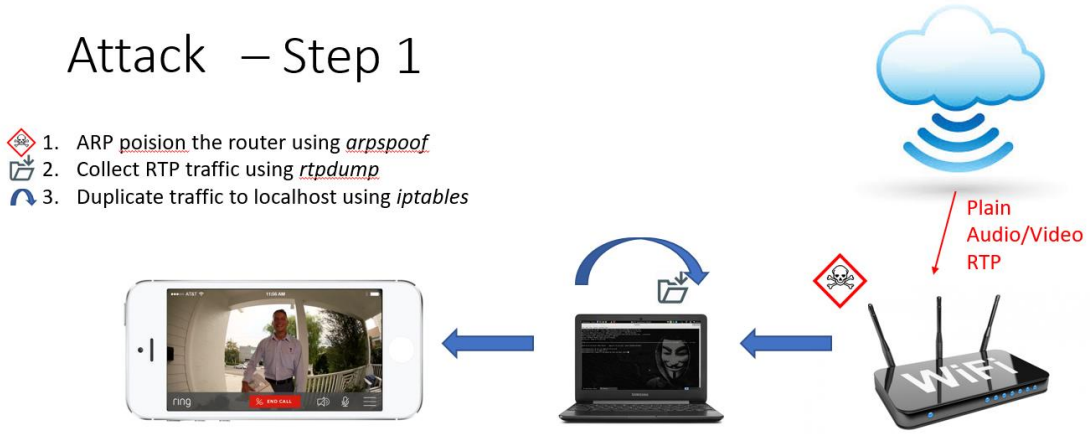
בשלב זה עברנו להסניף את האפליקציה. לצערנו ראינו שהתעבורת התראות, עדכונים, נתונים ו-SIP מוצפנת לשרת של Ring, כך שלא ניתן לראות את הקמת השיחה ישירות (אפשר בעזרת שיטות הסנפה יותר מתקדמות שמעבר ל-scope כאן). למרות זאת, אחרי שמצלצלים ורואים את השידור של המצלמה אפשר לראות את התוכן RTP ב-plain:

1310	11.878801752	18.197.187.115	192.168.36.124	H264	328	514 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=514, Time=660000, Mark FU-A End
1318	11.922528715	18.197.187.115	192.168.36.124	H264	1331	515 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=515, Time=666000 FU-A Start:non-IDR-Slice
1320	11.934711849	18.197.187.115	192.168.36.124	H264	1336	516 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=516, Time=666000 FU-A
1322	11.936379847	18.197.187.115	192.168.36.124	H264	1336	517 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=517, Time=666000 FU-A
1323	11.938401968	18.197.187.115	192.168.36.124	H264	344	518 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=518, Time=666000, Mark FU-A End
1356	12.003577768	18.197.187.115	192.168.36.124	H264	1331	519 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=519, Time=672000 FU-A Start:non-IDR-Slice
1357	12.003811649	18.197.187.115	192.168.36.124	H264	1336	520 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=520, Time=672000 FU-A
1359	12.005795296	18.197.187.115	192.168.36.124	H264	1336	521 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=521, Time=672000 FU-A
1362	12.007092537	18.197.187.115	192.168.36.124	H264	360	522 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=522, Time=672000, Mark FU-A End
1378	12.074726072	18.197.187.115	192.168.36.124	H264	1331	523 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=523, Time=678000 FU-A Start:non-IDR-Slice
1383	12.095755588	18.197.187.115	192.168.36.124	H264	1336	524 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=524, Time=678000 FU-A
1384	12.095763412	18.197.187.115	192.168.36.124	H264	1336	525 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=525, Time=678000 FU-A
1385	12.095767842	18.197.187.115	192.168.36.124	H264	280	526 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=526, Time=678000, Mark FU-A End
1401	12.145133318	18.197.187.115	192.168.36.124	H264	1331	527 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=527, Time=684000 FU-A Start:non-IDR-Slice
1416	12.152027842	18.197.187.115	192.168.36.124	H264	1336	528 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=528, Time=684000 FU-A
1431	12.216097001	18.197.187.115	192.168.36.124	H264	1336	529 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=529, Time=684000 FU-A
1432	12.216104244	18.197.187.115	192.168.36.124	H264	264	530 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=530, Time=684000, Mark FU-A End
1434	12.222939150	18.197.187.115	192.168.36.124	H264	1331	531 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=531, Time=690000 FU-A Start:non-IDR-Slice
1436	12.224437516	18.197.187.115	192.168.36.124	H264	1336	532 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=532, Time=690000 FU-A
1437	12.224441613	18.197.187.115	192.168.36.124	H264	1336	533 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=533, Time=690000 FU-A
1439	12.230072848	18.197.187.115	192.168.36.124	H264	408	534 PT=DynamicRTP-Type-96, SSRC=0x51F63391, Seq=534, Time=690000, Mark FU-A End

אלו פקטות ש-Wireshark מבין כהגיוניות בפורמט H264 ולכן רצינו לראות אם אפשר לשחזר מהן את הסרטון. השתמשנו בכלי בשם videosnarf, שמחלץ סרטונים / שיחות מתוך קובץ pcap, והצלחנו לראות את הסרטון מהמצלמה (H264) ולשמוע מהמיקרופון (G711)! המשמעות היא שכל מי שיכול לשים את ידו על הפקטות של האפליקציה יכול לראות את השידור. התמונה התבהרה ונראית כך:



אז איך נוכל לצפות בשידור? בתור התחלה יש להיות ברשת משותפת עם הטלפון. אפשר להיות ברשת Wi-Fi ציבורית משותפת כמו ברכבת או בבית קפה, להרים AP זדוני שהטלפון ייצטרף אליו או להשתמש במכשיר ביתי פרוץ בשביל pivoting. לאחר מכן, מבצעים מתקפת ARP בין הנתב והטלפון ומתחזים להם בשכבה שתיים. כעת נקבל את התעבורה המקורית ונוכל להעביר אותה הלאה גם לטלפון. התרחיש תקיפה ייראה כך:



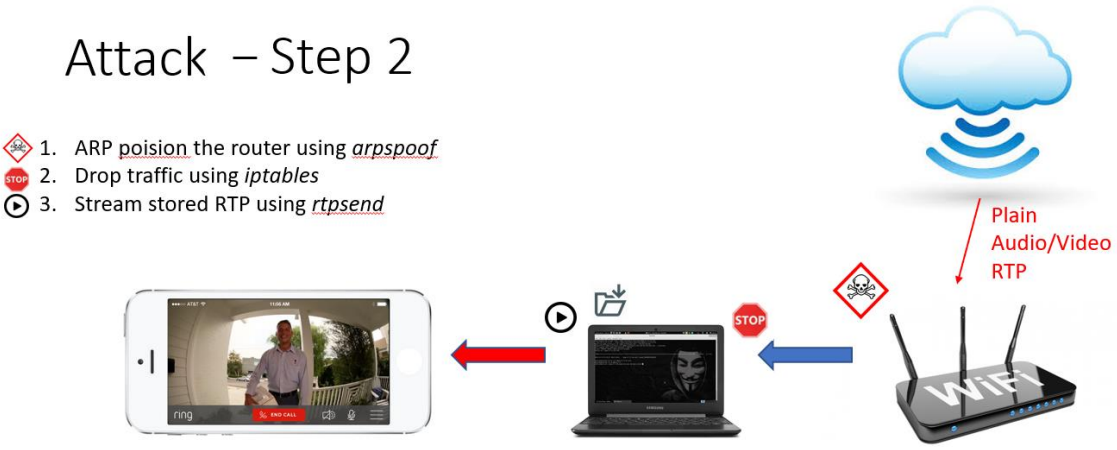
מרימים Kali ומריצים:

1. Arpspoof - מרעילים את הrouter ואת הטלפון כפי שהוסבר
2. Rtpdump - כלי לכתיבת תעבורת RTP מפורט לקובץ, אותו ניתן לקרוא
3. Iptables - שימוש ב-target בשם TEE לשכפל את התעבורה ליעד 127.0.0.1

כמובן שצריך להפוך את המחשב שלנו לנתב כדי להעביר את הפקטות ליעדם האמיתי על ידי:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

בעזרת setup זה יש לנו יכולת מרשימה של האזנת סתר לדלת כניסה של אנשים, בעזרתה ניתן לעשות הנדסה חברתית ולאסוף פרטים מעניינים רבים אודות ההרגלים בבית הקורבן. השלב הבא המתבקש הוא לזייף תעבורה:



שני שינויים על מנת לעבור למצב התקפה:

1. Iptables - הופכים את חוקי ה-TEE ממקודם לחוקי DROP ובכך מפילים תעבורת שמע ווידאו
2. Rtpsend - משדרים את קבצי ה-RTP ששמרנו בשלב הקודם לעבר הקורבן

בכך אנו משלימים התקפת replay בסיסית. בעזרתה, עשויים לשכנע את הקורבן לפתוח את הדלת מרחוק למי שהוא חושב שמכיר ושהוא סומך עליו. ההשלכות יכולות לנוע בסקלה שבין מתיחה מצחיקה לבין אירועים נוראיים כמו חטיפת ילדים שההורה שלהם פתח מרחוק למי שנראתה כמו הבייביסיטר.

דיווח וסגירת החולשה

החולשה דווחה לחברת Ring ובגרסה 3.4.7 באנדרואיד הבעיה תוקנה.

סיכום

התקיפה שהוצגה במאמר זה מראה שגם בחברות הענק ובמוצרי דגל ניתן למצוא חולשות לוגיות חמורות שדרוש מעט מאוד ידע ומיומנות לנצלן. ניכר שנעשה מאמץ לאבטח את השידור בעזרת שכבת ה-TLS באפליקציה והשדות המיוחדים ב-SIP מהמכשיר כמו גם ה-RTP המוצפן. למרות זאת, מערך ההגנה חזק רק כחוזק החולייה החלשה שלו, שבמקרה זה הייתה התעבורת RTP לטלפון. אירוני שדווקא מוצר אבטחה שאמור לחזק את תחושת הביטחון פותח הזדמנויות תקיפה חדשות שלא היו קודם, וזהו מוטיב שחוזר על עצמו שוב ושוב בעולם האבטחת מידע.

מקורות

- תמונת Ring Doorbell:

https://target.scene7.com/is/image/Target/GUEST_6ae6702b-c0b5-427b-8b74-cf2761aaca3b?wid=488&hei=488&fmt=pjpeg