



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**ANALÝZA ÚTOKOV NA AUTOMOBILY A ODCHYTENIE HTTPS
KOMUNIKÁCIE MEDZI MOBILNOU APLIKÁCIOU A KONKRÉTNYM
AUTOMOBILOM**

BEZPEČNÁ ZAŘÍZENÍ

Bc. Ján Jakub Kubík

29. marca 2022

Obsah

1	Úvod	2
2	Kľúčové komponenty automobilov a útoky	3
2.1	Kľúčové komponenty	3
2.2	Rozdelenie útokov	4
3	Výber konkrétného útoku a jeho realizácia	5
3.1	Použité nástroje	5
3.2	Obecný postup zachytenia requestov cez Charlesa	5
3.3	Konkrétny postup zachytenia requestov z aplikácie Kia Connect	6
4	Testovanie a experimenty	14
4.1	Overenie dĺžky session TTL	14
4.2	Skenovanie	15
5	Záver	17
	Literatúra	18
A	Súbory k Projektu	19

1 Úvod

Táto správa k projektu z predmetu *Bezpečná zařízení* sa zaoberá analýzou rôznych druhov útokov na automobily a implementáciou 1 konkrétneho útoku na 1 konkrétny automobil. Po naštudovaní literatúry popisujúcej rôzne druhy útokov a konzultácii s Ing. Ivanom Homoliakom Ph.D. sme sa dohodli, že sa pokúsim dekompilovať aplikáciu k jeho automobilu Kia XCeed, upraviť ju a ďalej sa pokúsim odchytiť komunikáciu medzi aplikáciou a automobilom a zistiť akými HTTPS requestami prebieha odomykanie a zamykanie daného automobilu.

Časť 2 popisuje kľúčové komponenty automobilov a rôzne druhy útokov na ne. Časť 3 obsahuje podrobný popis dekompilovania, upravovania a zbuildenia aplikácie a následne odchytenie HTTPS komunikácie. Časť 4 popisuje experimenty a diskutuje ich výsledky.

2 Kľúčové komponenty automobilov a útoky

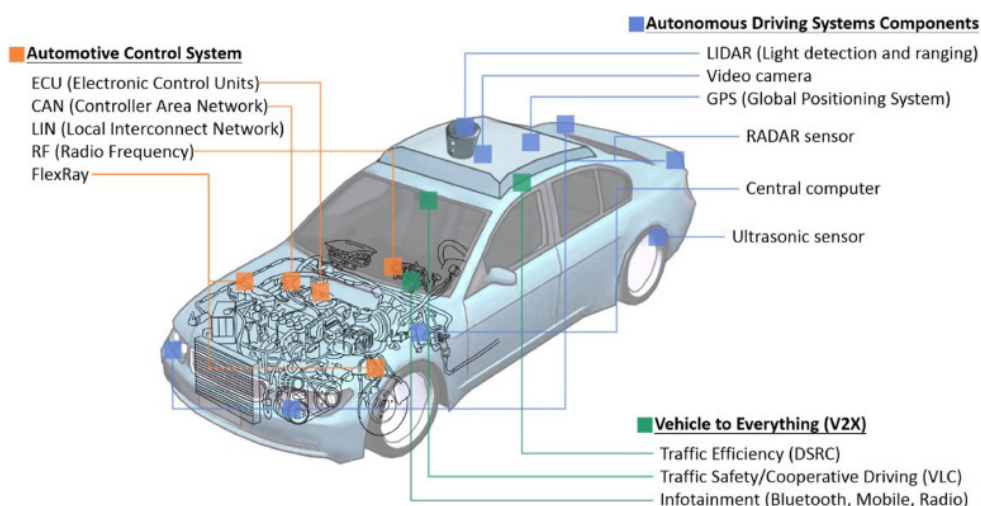
2.1 Kľúčové komponenty

V článku [1] sekcia 2 sú kľúčové komponenty automobilov rozdelené do 3 kategórii:

Automotive control system – pozostáva z vnútornej siete a komponentov automobilu, ktoré sieť prepája. Medzi najdôležitejšie komponenty patria electronic control units (ECUs). ECUs komunikujú a zodpovedajú od senzorov vnútri automobilu až po komponenty zodpovedné za bezpečnosť a pohodlie pasažierov ako je napríklad ABS, klimatizácia, osvetlenie, Komunikačnú sieť medzi jednotlivými ECUs tvorí controller area network (CAN), local interconnect network (LIN) a FlexRay.

Autonomous-Driving-System components – kľúčovými komponentami sú svetlo a vzdialenosť detekujúce senzory (LIDAR), videokamera, GPS, senzor na rádiovú detekciu a dosah (RADAR), centrálny počítač a ultrazvukový senzor.

V2X communication – je sieťová komunikácia medzi automobilom a externým prostredím alebo externými komponentami. Vehicle ad-hoc networks (VANETs) používa vyhradenú komunikáciu na krátku vzdialenosť (DSRC). VANET má široké spektrum použitia, ako napr. prevencia kolízií, monitorovanie stavu premávky v reálnom čase, výmena informácií medzi jednotlivými automobilmi, Infotainment zabezpečuje audio alebo video zábavu pre cestujúcich.



Obr. 2.1: Kľúčové komponenty automobilu

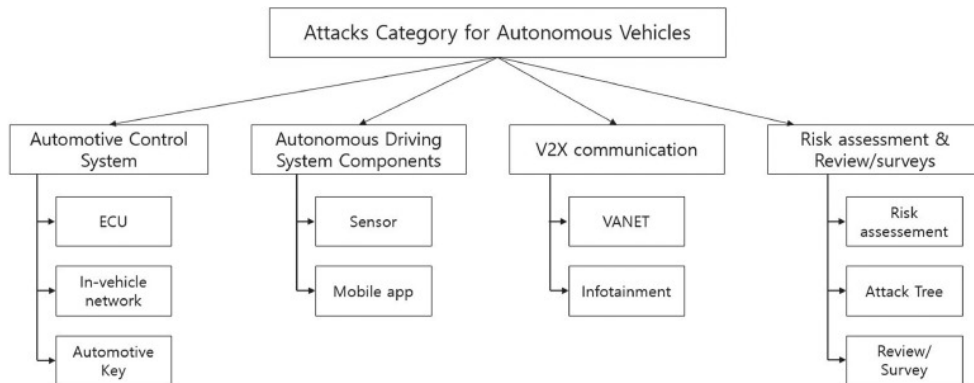
zdroj: [HTTPS://ars.els-cdn.com/content/image/1-s2.0-S0167404820304235-gr1.jpg](https://ars.els-cdn.com/content/image/1-s2.0-S0167404820304235-gr1.jpg)

2.2 Rozdelenie útokov

V článku [1] sekcia 3 sú útoky na automobily rozdelené do dvoch hlavných kategórií: jednoúčelové útoky a komplexné útoky. Jednoúčelové útoky sú zamerané na jednu komponentu z kategórii kľúčových komponentov, ktoré sú podrobne opísané v sekcii 2.1.

Obrázok 2.2 znázorňuje obe kategórie útokov. Do kategórie **jednoúčelových útokov** spadajú kategórie automotive control systems, autonomous driving system components a V2X communication.

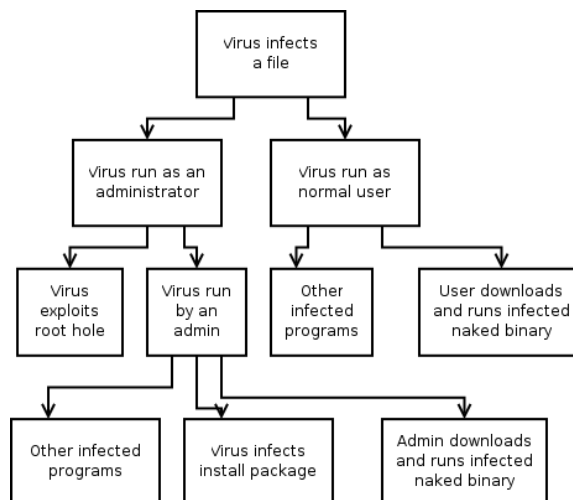
Do **komplexných útokov** spadá strom útokov z kategórie risk assesment.



Obr. 2.2: Kategórie útokov na autonómne vozidlá

zdroj: [HTTPS://ars.els-cdn.com/content/image/1-s2.0-S0167404820304235-gr3.jpg](https://ars.els-cdn.com/content/image/1-s2.0-S0167404820304235-gr3.jpg)

Stromy útokov sú viacúrovňové diagramy pozostávajúce z jedného koreňa, listov a detí. Zdola nahor sú detské uzly podmienkami, ktoré musia byť splnené, aby sa stal priamy nadradený rodičovský uzol true. Keď je koreň true, útok je dokončený. Rodičovský uzol môže vyžadovať aby všetci jeho potomci boli true (AND) alebo aspoň jeden (OR). Závysí to od konkrétného použitého diagramu.



Obr. 2.3: Ilustračný strom útoku

zdroj: [HTTPS://en.wikipedia.org/wiki/Attack_tree/media/File:Attack_tree_virus.png](https://en.wikipedia.org/wiki/Attack_tree/media/File:Attack_tree_virus.png)

3 Výber konkrétného útoku a jeho realizácia

Cieľom tejto práce bolo odchytiť komunikáciu medzi aplikáciou a automobilom Kia XCeed cez server a zistiť akými HTTPS requestami prebieha odomykanie a zamykanie daného automobilu a prípadne zistiť nejaké zraniteľnosti. Zraniteľnosti a samotný rozbor HTTPS requestov je kapitole 4.

3.1 Použité nástroje

Pre odchytenie HTTPS requestov na odomykanie a zamykanie auta bolo potrebné použiť:

Apktool¹ – nástroj na dekompilovanie/kompilovanie binárneho byte kódu aplikácie do/zo Smali kódu. Smali kód je niečo podobé ako dekompilovaný kód z bežnej C++ aplikácie.

Jadx² – dekompilátor binárneho byte kódu do Java kódu. Tento kód je podstatne lepšie čitateľný ako Smali kód. Jadx obsahuje aj gui, v ktorom sa dá dekompilovaný Java kód zobrazovať.

Charles³ – debugging proxy serverová aplikácia, v ktorej sa po presmerovní trafficu dajú sledovať HTTP, HTTPS a aj HTTP2 dotazy.

3.2 Obecný postup zachytenia requestov cez Charlesa

Pre zachytenie requestov cez Charlesa je vždy potrebné stiahnuť z internetu .apk súbor danej aplikácie, dekompilovať ho, nastaviť certifikát z Charlesa na testovacom telefóne, povoliť tento certifikát alebo aj všetky certifikáty v dekompilovanej aplikácii. Znovu skompilovať aplikáciu. Vygenerovať kľúč pre ňu, podpísať skompilovanú aplikáciu, nainštalovať ju do testovacieho zariadenia, nastaviť proxy na danom zariadení na IP adresu zariadenia na ktorom je spustený Charles, spustiť aplikáciu na testovacom zariadení a v Charlesovi je vidieť celý HTTPS traffic. Obe zariadenia musia byť v rovnakej sieti. Dost' často sa stáva, že aplikácia má ešte nejaký druh ochrany, ktorý môže byť jednoduchý alebo aj sofistikovaný web application firewall (WAF), ktorý zabráni takému spôsobu odchyťavania trafficu cez Charlesa. WAF pomáha chrániť webové aplikácie tým, že monitoruje a filtruje HTTPS komunikáciu medzi aplikáciou a serverom.

¹[HTTPS://ibotpeaches.github.io/Apktool/](https://ibotpeaches.github.io/Apktool/)

²[HTTPS://github.com/skylot/jadx](https://github.com/skylot/jadx)

³[HTTPS://www.Charlesproxy.com/](https://www.Charlesproxy.com/)

3.3 Konkrétny postup zachytenia requestov z aplikácie Kia Connect

Pre automobil Kia XCeed bolo potrebné nájsť najnovšiu verziu aplikácie Kia Connect. Aktuálne to je verzia 2.1.2⁴. Túto aplikáciu som dekompiloval.

```
~/Desktop/kia $ ls
Kia Connect_2.1.2_apkcombo.com.apk
~/Desktop/kia $ apktool d Kia\ Connect_2.1.2_apkcombo.com.apk -o kia
I: Using Apktool 2.4.1 on Kia Connect_2.1.2_apkcombo.com.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/jakubkubik/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values ** XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
~/Desktop/kia $ ls
Kia Connect_2.1.2_apkcombo.com.apk  kia
```

Do Manifest.xml som pridal do application tagu cestu k network security configu. Manifest obsahuje dôležité informácie pre buildovanie aplikácie, android os a Google Play. Network security config slúži prevažne na nastavovanie certifikačných autorít, ktorým má aplikácia dôverovať.



Ďalej som pridal network security config súbor do res/xml adresára. V ňom som povolil všetky certifikačné autority.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <network-security-config>
3      <base-config>
4          <trust-anchors>
5              <certificates src="system" overridePins="true" />
6              <certificates src="user" overridePins="true" />
7          </trust-anchors>
8      </base-config>
9  </network-security-config>
```

Takto upravenú aplikáciu som skompiloval.

⁴[HTTPS://apkcombo.com/kia-connect/com.kia.connect.eu/download/apk](https://apkcombo.com/kia-connect/com.kia.connect.eu/download/apk)

```
~/Desktop/kia $ apktool b kia/ -o kia.apk
I: Using Apktool 2.4.1
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
~/Desktop/kia $ ls
Kia Connect_2.1.2_apkcombo.com.apk kia kia.apk
```

Následne bolo potrebné vygenerovať kľúč⁵,

```
~/Desktop/kia $ keytool -genkey -v -keystore my-release-key.keystore -alias bza -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing my-release-key.keystore]
~/Desktop/kia $ ls
Kia Connect_2.1.2_apkcombo.com.apk kia kia.apk my-release-key.keystore
```

a podpísať ním aplikáciu. Podpisovanie aplikácie je nutné z dôvodu identifikovania autora aplikácie. Bez podpísania sa aplikáciu nepodari správne nainštalovať⁶.

```
~/Desktop/kia $ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore kia.apk bza > /dev/null
```

V testovacom telefone som musel povoliť developer options⁷.

Daný telefon som pripojil k PC s upravenou aplikáciou. Takto upravenú aplikáciu som nainštaloval do telefónu.

⁵<https://stackoverflow.com/questions/10930331/how-to-sign-an-already-compiled-apk>

⁶<https://source.android.com/security/apksigning>

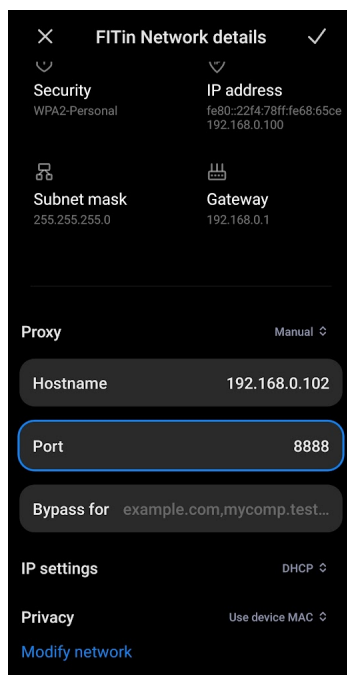
⁷<https://developer.android.com/studio/debug/dev-options>


```
~/Desktop/kia $ adb devices
List of devices attached
aefdd9e8      device

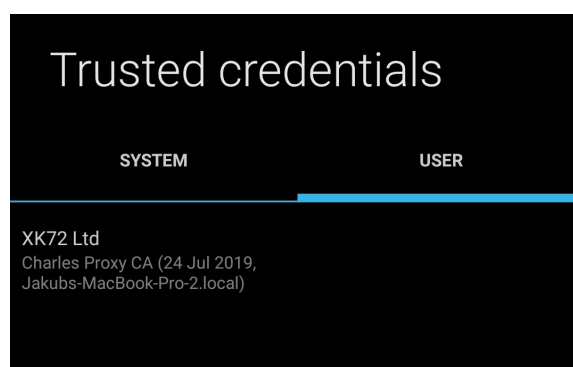
~/Desktop/kia $ adb -s aefdd9e8 install kia.apk
Performing Streamed Install
Success
```

Z

Na PC som spustil Charlesa. A na telefóne som nastavil proxy cez Charlesa na PC v rovnakej sieti.



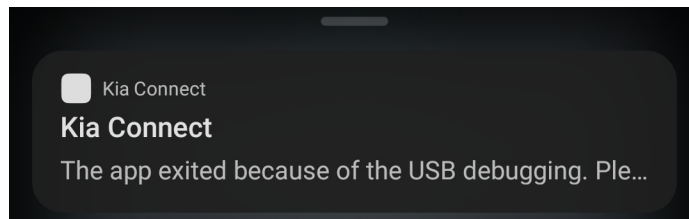
Na testovacom telefóne som musel ešte stiahnuť a nainštalovať Charles certifikát ⁸ pre správne zobrazovanie trafficu v Charlesovi.



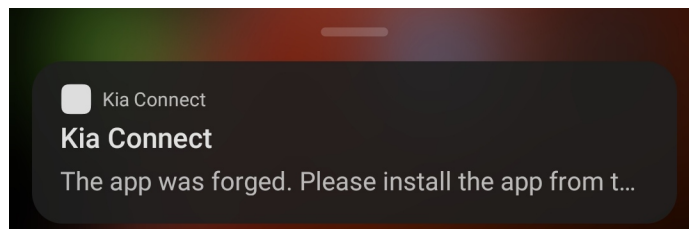
Obr. 3.1: Ilustračný strom útoku

⁸<https://gathelp.zendesk.com/hc/en-us/articles/360019067120-Charles-Certificate-on-Android-Setup-Instructions>

Aplikáciu sa mi nepodarilo spustiť. Dostal som chybovú hlášku USB debugging.



Tak som odpojil testovací telefón od PC a skúsil znovu spustiť aplikáciu. Hlásilo mi chybu, že aplikácia bola neoprávnene zmenená.



Z chybovej hlášky som usúdil, že pri spustení musí aplikácia nejako kontrolovať či bola zmenená. Predpokladal som, že bude kontrolovať asi hash skompilovanej aplikácie. Tak som musel znovu dekompilovať aplikáciu pomocou jadx.

```
^C~/Desktop/kia $ jadx-gui kia.apk
INFO - output directory: kia
INFO - loading ...
```

Pomocou neho som si aj zobrazil Java kód dekompilovanej aplikácie.

```
329  /* JADX WARNING: Removed duplicated region for block: B:16:0x0024 A[SYNTHETIC, Splitter:B:16:0x0024] */
330  /* JADX WARNING: Removed duplicated region for block: B:22:0x0034 A[SYNTHETIC, Splitter:B:22:0x0034] */
331  /* Code decompiled incorrectly, please refer to instructions dump. */
332  public static int getDexLen(java.lang.String r4) {
333      /*
334       *
335       *
336       *
337       *
338       *
339       *
340       *
341       *
342       *
343       *
344       *
345       *
346       *
347       *
348       *
349       *
350       *
351       *
352       *
353       *
354       *
355       *
356       *
357       *
358       *
359       *
360       *
361       *
362       *
363       *
364       *
365       *
366       *
367       *
368       *
369       *
370       *
371       *
372       *
373       *
374       *
375       *
376       *
377       *
378       *
379       *
380       *
381       *
382       *
383       *
384       *
385       *
386       *
387       *
388       *
389       *
390       *
391       *
392       *
393       *
394       *
395       *
396       *
397       *
398       *
399       *
400       *
401       *
402       *
403       *
404       *
405       *
406       *
407       *
408       *
409       *
410       *
411       *
412       *
413       *
414       *
415       *
416       *
417       *
418       *
419       *
420       *
421       *
422       *
423       *
424       *
425       *
426       *
427       *
428       *
429       *
430       *
431       *
432       *
433       *
434       *
435       *
436       *
437       *
438       *
439       *
440       *
441       *
442       *
443       *
444       *
445       *
446       *
447       *
448       *
449       *
450       *
451       *
452       *
453       *
454       *
455       *
456       *
457       *
458       *
459       *
460       *
461       *
462       *
463       *
464       *
465       *
466       *
467       *
468       *
469       *
470       *
471       *
472       *
473       *
474       *
475       *
476       *
477       *
478       *
479       *
480       *
481       *
482       *
483       *
484       *
485       *
486       *
487       *
488       *
489       *
490       *
491       *
492       *
493       *
494       *
495       *
496       *
497       *
498       *
499       *
500       *
501       *
502       *
503       *
504       *
505       *
506       *
507       *
508       *
509       *
510       *
511       *
512       *
513       *
514       *
515       *
516       *
517       *
518       *
519       *
520       *
521       *
522       *
523       *
524       *
525       *
526       *
527       *
528       *
529       *
530       *
531       *
532       *
533       *
534       *
535       *
536       *
537       *
538       *
539       *
540       *
541       *
542       *
543       *
544       *
545       *
546       *
547       *
548       *
549       *
550       *
551       *
552       *
553       *
554       *
555       *
556       *
557       *
558       *
559       *
560       *
561       *
562       *
563       *
564       *
565       *
566       *
567       *
568       *
569       *
570       *
571       *
572       *
573       *
574       *
575       *
576       *
577       *
578       *
579       *
580       *
581       *
582       *
583       *
584       *
585       *
586       *
587       *
588       *
589       *
590       *
591       *
592       *
593       *
594       *
595       *
596       *
597       *
598       *
599       *
600       *
601       *
602       *
603       *
604       *
605       *
606       *
607       *
608       *
609       *
610       *
611       *
612       *
613       *
614       *
615       *
616       *
617       *
618       *
619       *
620       *
621       *
622       *
623       *
624       *
625       *
626       *
627       *
628       *
629       *
630       *
631       *
632       *
633       *
634       *
635       *
636       *
637       *
638       *
639       *
640       *
641       *
642       *
643       *
644       *
645       *
646       *
647       *
648       *
649       *
650       *
651       *
652       *
653       *
654       *
655       *
656       *
657       *
658       *
659       *
660       *
661       *
662       *
663       *
664       *
665       *
666       *
667       *
668       *
669       *
670       *
671       *
672       *
673       *
674       *
675       *
676       *
677       *
678       *
679       *
680       *
681       *
682       *
683       *
684       *
685       *
686       *
687       *
688       *
689       *
690       *
691       *
692       *
693       *
694       *
695       *
696       *
697       *
698       *
699       *
700       *
701       *
702       *
703       *
704       *
705       *
706       *
707       *
708       *
709       *
710       *
711       *
712       *
713       *
714       *
715       *
716       *
717       *
718       *
719       *
720       *
721       *
722       *
723       *
724       *
725       *
726       *
727       *
728       *
729       *
730       *
731       *
732       *
733       *
734       *
735       *
736       *
737       *
738       *
739       *
740       *
741       *
742       *
743       *
744       *
745       *
746       *
747       *
748       *
749       *
750       *
751       *
752       *
753       *
754       *
755       *
756       *
757       *
758       *
759       *
760       *
761       *
762       *
763       *
764       *
765       *
766       *
767       *
768       *
769       *
770       *
771       *
772       *
773       *
774       *
775       *
776       *
777       *
778       *
779       *
780       *
781       *
782       *
783       *
784       *
785       *
786       *
787       *
788       *
789       *
790       *
791       *
792       *
793       *
794       *
795       *
796       *
797       *
798       *
799       *
800       *
801       *
802       *
803       *
804       *
805       *
806       *
807       *
808       *
809       *
810       *
811       *
812       *
813       *
814       *
815       *
816       *
817       *
818       *
819       *
820       *
821       *
822       *
823       *
824       *
825       *
826       *
827       *
828       *
829       *
830       *
831       *
832       *
833       *
834       *
835       *
836       *
837       *
838       *
839       *
840       *
841       *
842       *
843       *
844       *
845       *
846       *
847       *
848       *
849       *
850       *
851       *
852       *
853       *
854       *
855       *
856       *
857       *
858       *
859       *
860       *
861       *
862       *
863       *
864       *
865       *
866       *
867       *
868       *
869       *
870       *
871       *
872       *
873       *
874       *
875       *
876       *
877       *
878       *
879       *
880       *
881       *
882       *
883       *
884       *
885       *
886       *
887       *
888       *
889       *
890       *
891       *
892       *
893       *
894       *
895       *
896       *
897       *
898       *
899       *
900       *
901       *
902       *
903       *
904       *
905       *
906       *
907       *
908       *
909       *
910       *
911       *
912       *
913       *
914       *
915       *
916       *
917       *
918       *
919       *
920       *
921       *
922       *
923       *
924       *
925       *
926       *
927       *
928       *
929       *
930       *
931       *
932       *
933       *
934       *
935       *
936       *
937       *
938       *
939       *
940       *
941       *
942       *
943       *
944       *
945       *
946       *
947       *
948       *
949       *
950       *
951       *
952       *
953       *
954       *
955       *
956       *
957       *
958       *
959       *
960       *
961       *
962       *
963       *
964       *
965       *
966       *
967       *
968       *
969       *
970       *
971       *
972       *
973       *
974       *
975       *
976       *
977       *
978       *
979       *
980       *
981       *
982       *
983       *
984       *
985       *
986       *
987       *
988       *
989       *
990       *
991       *
992       *
993       *
994       *
995       *
996       *
997       *
998       *
999       *
1000      */
1001      java.util.zip.ZipEntry r1 = r2.getNextEntry() // Catch:{ Exception -> 0x004b }
1002      if (r1 != 0) goto L_0x0018
1003      if (r2 == 0) goto L_0x0017
1004      r2.close() // Catch:{ Exception -> 0x003f }
1005      L_0x0017:
1006      return r0
1007      L_0x0018:
1008      int r0 = r0 + 1
1009      goto L_0x000c
1010      L_0x001d:
1011      r1 = move-exception
1012      r2 = r3
1013      L_0x001f:
1014      r1.printStackTrace() // Catch:{ all -> 0x0046 }
1015      if (r2 == 0) goto L_0x0017
1016      r2.close() // Catch:{ Exception -> 0x002a }
1017      goto L_0x0017
1018      .....
```

Niektoré funkcie sa nepodarilo správne dekompilovať, tak som musel použiť prepínač⁹.

⁹<https://stackoverflow.com/questions/41424442/jadx-output-decompile-error-code-decompiled->

```
~/Desktop/kia $ jadx-gui --show-bad-code kia.apk
INFO - output directory: kia
INFO - loading ...
INFO - Can't find 'R' class in app package: com.kia.connect.eu
```

Potom sa už všetky funkcie zobrazili.

```
311  /* JADX WARNING: Removed duplicated region for block: B:16:0x0024 A[SYNTHETIC, Splitter:B:16:0x0024] */
312  /* JADX WARNING: Removed duplicated region for block: B:22:0x0034 A[SYNTHETIC, Splitter:B:22:0x0034] */
313  public static int getDexLen(String str) {
314      Throwable th;
315      ZipInputStream zipInputStream;
316      Exception e2;
317      int i2 = 0;
318      ZipInputStream zipInputStream2 = null;
319      try {
320          zipInputStream = new ZipInputStream(new FileInputStream(str));
321          while (zipInputStream.getNextEntry() != null) {
322              try {
323                  i2++;
324              } catch (Exception e3) {
325                  e2 = e3;
326                  try {
327                      e2.printStackTrace();
328                      if (zipInputStream != null) {
329                          return i2;
330                      }
331                  } catch (Throwable th2) {
332                      th = th2;
333                      zipInputStream2 = zipInputStream;
334                      if (zipInputStream2 != null) {
335                          throw th;
336                      }
337                  }
338              }
339          }
340      }
```

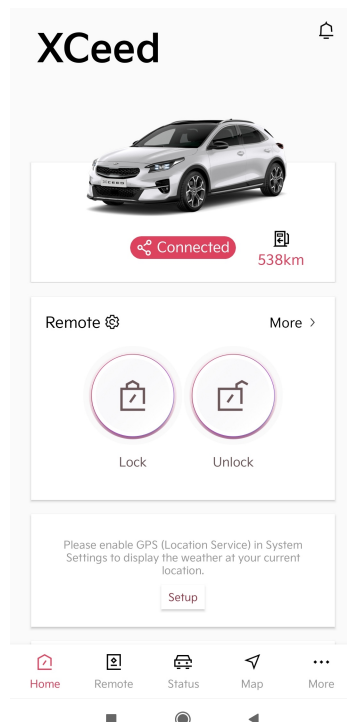
Ďalej som potreboval zistiť čo sa vlastne deje v aplikácii. Tak som si našiel kde je vstupný bod aplikácie ¹⁰. Vstupným bodom je metóda onCreate.

```
587  public void onCreate() {
588      super.onCreate();
589      int i2 = 1;
590      while (true) {
591          if (i2 >= 10) {
592              break;
593          } else if (i2 >= 7) {
594              fjs.a(this);
595              if ("RES_NOT_ENCRYPTED".equals("RES_ENCRYPTED") && "AHOPE_IS_UNITY_APP".equals("TRUE")) {
596                  callMethod(i[8], getPackageName());
597              }
598          } else {
599              if (i2 == 1 || i2 >= 5) {
600                  if (i2 == 1) {
601                      if (i0 == "value".length()) {
602                          int i3 = "system.permission".length() > 0 ? 1 : 0;
603                          for (int i4 = 0; i4 < 10; i4++) {
604                              i3 += "system.permission".substring(i4, i4 + 1).charAt(0);
605                              if (i3 > 100) {
606                                  }
607                          }
608                      }
609                      e = this;
610                      yks.apply();
611                      f = skdb.vhqp(e);
612                  } else if (i2 > 4) {
613                      for (int i5 = 0; i5 < 1; i5++) {
614                          if (i5 > 10) {
615                              int[] iArr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3, 2, 1};
616                              int i6 = 7;
617                              for (int i7 = 0; i7 < iArr.length; i7++) {
618                                  int i8 = (i6 * iArr[i7]) / 2;
619                                  i6 = iArr[i7] % 2 != 0 ? i8 << iArr[i7] : i8 >> iArr[i7];
620                              }
621                          }
622                      }
623                  }
624              }
625              i2 += 2;
626          }
627      }
628      if (d != null) {
629          d.onCreate();
630      }
631  }
```

Postupným skúmaním kódu som sa dostal cez **skdb.vhqp** k podozrivým metódam **b**, **d**, **e**.

incorrectly-please-refer-to-instr

¹⁰<https://nphau.medium.com/android-when-does-applications-oncreate-method-get-called-5d5019c1a862>
















Aj traffic sa správne zachytával a zobrazoval v Charlesovi.

Code	Method	Host	Path	Start	Duration	Size	Status
200	POST	prd.eu-ccapi.kia.com:8080	/api/v1/spa/notifications/register	19:02:02	470 ms	33.80 KB	Complete
200	POST	prd.eu-ccapi.kia.com:8080	/api/v1/spa/devices/version	19:02:03	157 ms	34.32 KB	Complete
200	POST	prd.eu-ccapi.kia.com:8080	/api/v1/user/oauth2/token	19:02:03	200 ms	34.37 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/user/profile	19:02:03	182 ms	34.33 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles	19:02:04	195 ms	34.38 KB	Complete
200	POST	prd.eu-ccapi.kia.com:8080	/api/v1/spa/notifications/8013e98f-bfed-46d0-819b-d069cd5da7c8/r...	19:02:05	146 ms	34.23 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/devices/8013e98f-bfed-46d0-819b-d069cd5da7c8/settin...	19:02:05	151 ms	34.14 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/settin...	19:02:05	192 ms	34.35 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/profile/updateinfo?lang=en&car_id=a9c73bf7-abc1-4ecf-88aa...	19:02:06	131 ms	34.15 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/profile	19:02:06	184 ms	35.66 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/notifications/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/co...	19:02:07	172 ms	34.15 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/locati...	19:02:08	190 ms	34.20 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/status...	19:02:08	253 ms	34.57 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/profile/users/b8c9f1f6-5b3f-4b45-9cfa-dcf8619e53f5/cars/a9...	19:02:08	193 ms	34.10 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/finald...	19:02:08	153 ms	34.12 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/settin...	19:02:08	199 ms	34.35 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/status...	19:02:09	435 ms	35.73 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/profile/users/b8c9f1f6-5b3f-4b45-9cfa-dcf8619e53f5/cars/a9...	19:02:10	997 ms	34.10 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/profile/users/b8c9f1f6-5b3f-4b45-9cfa-dcf8619e53f5/cars/a9...	19:02:13	357 ms	34.19 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/settin...	19:02:24	572 ms	34.35 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/profile/users/b8c9f1f6-5b3f-4b45-9cfa-dcf8619e53f5/cars/a9...	19:02:24	437 ms	34.10 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/locati...	19:02:24	530 ms	34.20 KB	Complete
200	GET	prd.eu-ccapi.kia.com:8080	/api/v1/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/status...	19:02:24	442 ms	34.57 KB	Complete

Ďalším prezeraním kódu a googlením som zistil, že metódu **e** nemusím vôbec zakomentovať. Ďalej som zistil, že metóda **b** má na starosti blokovanie aplikácii pri USB debugingu a metóda **d** je zodpovedná za kontrolovanie správnosti hashu danej aplikácie.

Na záver som ešte skúsil dekompilovať aplikáciu pre auto škoda a zachytiť requesty. Podarilo sa mi to tiež. Ale tam na rozdiel od Kia Connect nebolo potrebné nič upravovať priamo v Smali kóde. Stačilo len pridať network security config.

Nejaké requesty som aj z tejto aplikácie zachytil.

Structure		Sequence							
Code	Method	Host	Path	Start	Duration	Size	Status	Info	
	403	POST	firebaseinstallations.googleapis.com	/v1/projects/cz-skoda-auto-connect/installations	19:10:28	592 ms	24.84 KB	Complete	
	200	GET	api.connect.skoda-auto.cz	/api/v1/users/a4cb7eee-f7e9-45ec-9feb-be247eda9152/identities	19:10:29	765 ms	24.35 KB	Complete	
	200	GET	api.connect.skoda-auto.cz	/api/v2/preferences	19:10:32	1.10 s	24.59 KB	Complete	
	200	GET	api.connect.skoda-auto.cz	/api/v3/garage	19:10:32	551 ms	2.53 KB	Complete	
	302	GET	identity.vwgroup.io	/oidc/v1/authorize?redirect_uri=skodaconnect://oidc.login/&nonce=f73...	19:10:44	534 ms	28.02 KB	Complete	
	302	GET	identity.vwgroup.io	/oidc/v1/authorize?redirect_uri=skodaconnect://oidc.login/&nonce=ab8...	19:10:45	286 ms	28.33 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:47	744 ms	37.26 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:48	98 ms	19.52 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:48	336 ms	75.82 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:48	299 ms	48.79 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:49	66 ms	1.83 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:49	68 ms	3.41 KB	Complete	
	200	GET	cdn.emea.vwapps.io	/assets/be108820-9b1a-4906-a2e1-3f39150c43b7/production/emea/1...	19:10:55	35 ms	771 bytes	Complete	





Ale keďže som nemal dostupné auto na ktorom by som mohol vyskúšať napr. odomykanie, zamykanie, tak som s touto aplikáciou ďalej nič nerobil.

4 Testovanie a experimenty

V testovaní som overil či a za akú dlhú dobu expirujú requesty na odomykanie. A oskenoval som na dostupné zraniteľnosti endpoint pre tieto requesty.

4.1 Overenie dĺžky session TTL

Z vykonaných experimentov sa mi podarilo zistiť, že request z Charlesa môže byť znovu úspešne spustený napr. cez curl, s tým že sa úspešne odomknú alebo zamknú dvere. Session TTL pre odchytený request je približne 8 min a nezáleží na IP adrese odosielateľa. Čiže by toho mohol útočník teoreticky zneužiť.

Structure		Sequence								
	Code	Method	Host	Path	Start	Duration	Size	Status		
	200	POST	prd.eu-ccapi.kia.com:8080	/api/v2/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/control...	10:59:59	237 ms	34.30 KB	Complete		
	200	POST	prd.eu-ccapi.kia.com:8080	/api/v2/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/control...	11:00:25	183 ms	34.30 KB	Complete		
	200	POST	prd.eu-ccapi.kia.com:8080	/api/v2/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/control...	11:00:42	183 ms	34.30 KB	Complete		
	200	POST	prd.eu-ccapi.kia.com:8080	/api/v2/spa/vehicles/a9c73bf7-abc1-4ecf-88aa-fba3d0e0ae68/control...	11:01:20	174 ms	34.30 KB	Complete		

Filter: door

Overview	Contents	Summary	Chart	Notes
----------	----------	---------	-------	-------

```

{
  "action": "open",
  "deviceId": "1e8cba06-1a1c-48ac-b85a-8323ce427fe9"
}

```

Headers	Authentication	Text	Hex	JavaScript	JSON	JSON Text	Raw
---------	----------------	------	-----	------------	------	-----------	-----

```

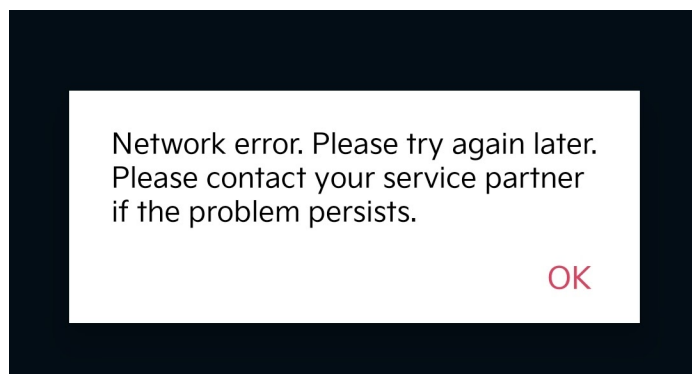
{
  "retCode": "S",
  "resCode": "0000",
  "msgId": "bf008ce8-01ef-4253-b564-ad7ba5509c0c"
}

```

Obr. 4.1: Odchytené requesty Charlesom

[illegible]

Obr. 4.2: Úspešné odomknutie útočníkom



Tak som si skúsil cez VPN zmeniť IP adresu. Pomohlo to. Čiže pravdepodobne nejako detekovali, že som z ich z mojej IP adresy skenoval, a tak zablokovali moju IP adresu.

5 Záver

Cieľom tejto práce bolo zistiť niečo o rôznych útokoch na automobily, vybrať si 1 útok a zrealizovať ho. Zvolený útok mal odchytiť HTTPS requesty na odomykanie/zamykanie auta Kia XCeed a prípadne zistiť potenciálne zraniteľnosti.

Podarilo sa mi úspešne dekompilovať aplikáciu pre dané auto, upraviť ju, naspäť skompilovať a odchytiť HTTPS requesty na odomykanie a zamykanie auta. Ďalej som overil ich expiráciu. A na záver som oskenoval použité endpointy pre odomykanie/zamykanie auta na potenciálne zraniteľnosti.

Do budúcnosti by určite stálo za to zistiť ako presne funguje vytváranie sessions pre jednotlivé requesty a pokúsiť sa to zreprodukovať. Takto by sa útočník mohol teoreticky stať absolútne nezávislý na pravidelnom odchyťovaní requestov. Stačilo by mu to len raz. A mohol by si odomykať/zamykať auto ako by sa mu zachcelo.

Taktiež by mohlo byť zaujímavé skúsiť exploitovať potenciálne zraniteľnosti, ktoré boli zistené zo skenovania.

Literatúra

- [1] KIM, K., KIM, J. S., JEONG, S., PARK, J.-H. a KIM, H. K. Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers Security*. 2021, roč. 103, s. 102150. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0167404820304235>. ISSN 0167-4048.

A Súbory k Projektu

Dodatok obsahujem zoznam súborov k projektu dostupných na [Google Drive](#)¹.

- kia.apk – upravená aplikácia pre auto Kia
- kia.chls – Charles session ku Kia XCeed automobilu
- imgs/ – adresár s použitými obrázkami

¹https://drive.google.com/drive/folders/13F6h_3jN0_u8Qa1hs9WnjOJPi2AqBfE-?usp=sharing