FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Analýza bezpečnosti aktualizované vnitřní sítě vozu Tesla Model 3
Student:	Lukáš Nerad
Vedoucí:	Ing. Jiří Dostál, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Tesla Model 3 patří mezi nejnovější generaci elektromobilů s novou architekturou a infrastrukturou řídicích jednotek. Oproti ostatním automobilům Model 3 nepoužívá pouze sběrnici CAN, ale pro většinu interní komunikace síť typu Ethernet. Seznamte se s diplomovou prací „Tesla Model 3 Internal Network Security Analysis" (Analýza bezpečnosti vnitřní sítě vozu Tesla Model 3). Od doby sepsání práce se síťová komunikace elektromobilu architekturně změnila. Na testovacím voze proveďte bezpečnostní analýzu interní sítě propojující elektronické jednotky (ECU). Zaměřte se hlavně na jednotku řídící autopilota (ACU) a multimediální jednotku (MCU). Popište síťovou architekturu a použité komunikační protokoly. Identifikujte vybrané zranitelnosti a případně vytvořte exploity. V případě objevení „zero day" zranitelnosti zahajte proces zodpovědného odhalení (responsible disclosure). Vše zdokumentujte, vyhodnoťte možné bezpečnostní dopady a navrhněte jejich řešení.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 30. ledna 2022 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Analýza bezpečnosti aktualizované vnitřní sítě vozu Tesla Model 3

Lukáš Nerad

Katedra počítačových systémů
Vedoucí práce: Ing. Jiří Dostál, Ph.D.

3. ledna 2023

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce Ing. Jiřímu Dostálovi, Ph.D. za vedení a za čas, který mi během psaní této práce věnoval. Dále bych chtěl poděkovat jeho týmu za odborné rady a konzultace. V neposlední řadě děkuji své rodině a své partnerce Bc. Sáře Adámkové za podporu během psaní této práce i za veškerou podporu během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 3. ledna 2023 . . . . . .

Odkaz na tuto práci

Nerad, Lukáš. *Analýza bezpečnosti aktualizované vnitřní sítě vozu Tesla Model 3*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Abstrakt

Tato bakalářská práce se zabývá bezpečnostní analýzou aktualizované vnitřní sítě automobilu Tesla Model 3. Teoretická část práce popisuje síťové technologie používané automobilovým průmyslem a seznamuje čtenáře s využívanými technologiemi vozu Tesla Model 3. Praktická část se následně zaměřuje na vypracování bezpečnostní analýzy testované sítě. Bezpečnostní analýza využívá modifikovaného testovacího standardu PTES. Standard byl obohacen o experimentální část, která rozšiřuje přínosnost práce v odvětví penetračního testování. Analýza zjistila, že testovaná vnitřní síť vozu je bezpečně vytvořena s výjimkou videozáznamu ze zadních kamer. Útočník může přehrávaný videozáznam, po předešlém přístupu do vozu, měnit bez vědomí vlastníka automobilu. V závěru práce je uvedena konkrétní ukázka nalezené zranitelnosti, jejíž ukázkové video je k nalezení v příloze práce.

Klíčová slova Tesla Model 3, bezpečnostní analýza, penetrační testování, interní síťová komunikace, BroadR-Reach Ethernet, standard PTES

Abstract

The bachelor thesis focuses on the security analysis of the updated internal network of the Tesla Model 3 car. The theoretical part of the work describes the network technologies used by the automotive industry and introduces the reader to the technologies used in the Tesla Model 3. The practical part then focuses on a security analysis of the tested car. The security analysis uses a modified penetration testing standard PTES. The standard has been enriched with an experimental part that expands the usefulness of the work in the penetration testing industry. The analysis found that the tested internal network of the car is securely created apart from the video recording from the rear camera. An attacker could modify video recording, after the previous access to the car, without the knowledge of the car owner. At the end of the thesis, a specific example of the found vulnerability is presented and a sample video recording of the vulnerability can be found in the appendix of the thesis.

Keywords Tesla Model 3, security analysis, penetration testing, internal network communication, BroadR-Reach Ethernet, standard PTES

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Automobilový průmysl výrazně pokročil v elektrifikaci a celkové modernizaci automobilů za poslední dvě desetiletí. Automobily se díky vývoji naučily autonomně řídit a parkovat. Umí číst dopravní značení, rozpoznávat překážky na vozovce a upravovat jízdu vůči ostatním vozům či osobám na silnici. Díky těmto funkcionalitám je možné automobily označovat za počítače na čtyřech kolech. Výrazného pokroku těchto automobilových vlastností jsme se dočkali v posledních deseti letech. Tyto autonomní funkce nejsou zásluhou jedné řídící jednotky, ale vícero částí vozu jako např. pohybové senzory, kamery, řídící jednotka autopilota atd. Všechny tyto části vozu mezi sebou musí komunikovat v reálném čase. Protože se jedná o funkcionality, které ovlivňují lidské životy, musí tato vnitřní komunikace vozu probíhat zabezpečeně, spolehlivě a v neposlední řadě rychle.

Většina interní komunikace v automobilovém průmyslu je řešena pomocí CAN sběrnice (*Controlled Area Network*). Tato sběrnice umožňuje komunikaci mezi jednotkami za pomoci broadcasting komunikace to znamená, že všichni účastnící připojeni na sběrnici slyší veškerou komunikaci. Tato sběrnice je levné a ověřené řešení, avšak už není jediné na trhu, které automobilový průmysl používá. Mezi možné alternativy patří FlexRay, MOST (Media Oriented System Transport) a automobilový Ethernet neboli BroadR-Reach.

Cílem bakalářská práce je provedení bezpečnostní analýzy interní sítě vozu Tesla Model 3. Síť vozu je postavena na technologii BroadR-Reach Ethernet a propojuje jednotlivé elektronické řídící jednotky (ECU) automobilu. Hlavním zaměřením bezpečnostní analýzy je propojení mezi multimediální řídící jednotkou (MCU) a řídící jednotkou autopilota (ACU).

Na začátku práce si představíme používané technologie umožňující vnitřní komunikaci v automobilovém průmyslu. Následně si popíšeme základní teorii bezpečnostní analýzy a představíme si používané testovací standardy. Na teorii bezpečnostní analýzy navážeme navržením struktury bezpečnostního testování automobilů. A v závěru práce popíšeme proces penetračního testování a předneseme výsledky, které bezpečnostní analýza sítě přinesla.

Kapitola 1

Technologie vnitřních sítí automobilů

1.1 Používané technologie pro interní komunikaci

Interní komunikace elektrických vozů se ve své podstatě neliší od interní komunikace ve vozech se spalovacími motory. V obou případech je potřeba spolehlivé komunikace mezi jednotlivými částmi automobilu, pro zajištění jeho správného a bezpečného chodu.

Původně byla vnitřní komunikace v automobilech zajištěna jednoduchými měděnými dráty, které vytvářely spletitou síť mezi vnitřními systémy vozu dlouhou klidně i několik kilometrů Spletitost takovéto sítě bychom mohli přirovnat k spletitosti nervového systému člověka. Nejen, že takové řešení bylo nepraktické, ale výrazně zvyšovalo hmotnost automobilu a složitost jeho výroby. [1]

Částečným řešením těchto problémů byla technologie LIN (*Local Interconnect Network*), která přinášela základní síťovou topologii, až 16 připojených zařízení a strukturované rámce zasílané po LIN síti. K zásadní změně došlo vývojem a následnou implementací CAN (*Controller Area Network*) protokolu v automobilovém průmyslu. [2]

1.1.1 CAN sběrnice

Jedná se o *multidrop* sběrnici, to znamená, že všechna zařízení jsou připojena na jednu síť, kde probíhá broadcasting komunikace. O tom, jaké připojené zařízení v daný čas komunikuje, rozhoduje priorita rámce. Čím je priorita vyšší (nižší číslo v identifikátoru rámce) tím je komunikace důležitější a zařízení bude dříve na řadě s prioritním vysíláním. Během vysílání jednoho zařízení se ostatní zařízení přepnou do poslechového režimu a čekají až na ně přijde řada. CAN protokol zajišťuje, že další nejvyšší prioritní rámec, který nevyhrál

předchozí rozhodovací proces, bude v dalším rozhodovacím cyklu další na řadě. [3]

V protokolu je také implementováno znovu-posílání chybových rámců, za účelem zajištění spolehlivé komunikace po sběrnici. Pokud je zaregistrováno zaslání chybového rámce, je odesílající zařízení urgováno ostatními připojenými zařízeními k jeho opakovanému odeslání. Pokud toto zařízení bude opakovaně přijímat rámce s informací, že odeslal rámec chybně, tak se zařízení automaticky odpojí a vypne, aby nenarušovalo běh komunikace na sběrnici. Mechanismy na detekci chybných rámců fungují na všech připojených zařízeních nezávisle. [4]

CAN protokol verze 2.0 se skládá ze dvou částí, verze 2.0A a verze 2.0B. Verzi 2.0A bychom mohli nazvat jako základní verze CAN protokolu, jedná se totiž o část CAN protokolu v původní verzi s 11-bitovým identifikátorem rámce. Verze 2.0B rozšiřuje rámec o dalších 18 bitů, celkově má identifikátor rámce 29 bitů, tedy rozšířený rámec CAN protokolu. CAN protokol verze 2.0 je zpětně kompatibilní se základním CAN protokolem. Na jedné sběrnici se může komunikovat pomocí obou protokolů v jednu chvíli. [5]

CAN sběrnici můžeme dále rozdělit na dva standardy, vysoko-rychlostní CAN (VR-CAN) a nízko-rychlostní CAN (NR-CAN). Jak název napovídá, hlavní rozdíl je v maximální možné rychlosti. VR-CAN má maximální rychlost odesílání 1 Mb/s a NR-CAN má maximální rychlost odesílání 125 kb/s. Tyto rychlosti jsou umožněny jiným zapojením rezistorů na sběrnici. Kde VR-CAN sběrnice má na obou koncích po jednom tranzistoru o odporu 120 ohmů. Naproti tomu v NR-CAN sběrnici má každé připojené zařízení svůj vlastní odpor. Hlavní výhoda NR-CAN sběrnice je její implementace systémů odolných proti chybám, kde tyto systémy umožňují udržení komunikace mezi zařízeními, při selhání kabeláže na sběrnici. Z důvodů jiného zapojení tranzistorů, nelze mít VR-CAN zařízení na stejné sběrnici jako NR-CAN zařízení. [6]

TTCAN (*Time-Trigged CAN*) protokol umožňuje zasílání *time-triggered* rámců na CAN sběrnici. Jedná se pouze o dodatek ke CAN protokolu, díky němuž můžeme na stejné fyzické sběrnici posílat jak *time-triggered* rámce tak i *event-triggered* rámce. [7]

CAN FD (*CAN with Flexible Data-rate*) je dalším vylepšením CAN protokolu verze 2.0, která je s ní zpětně kompatibilní. CAN FD umožňuje flexibilní navýšení CAN zprávy, v zasílaném rámci, z 8 bajtů na 64 bajtů. Tímto způsobem se může rychlost rámce na sběrnici zvýšit až na 8 Mb/s. Díky CAN FD protokolu si mohou zařízení připojená na sběrnici dynamicky měnit velikost zprávy podle aktuální potřeby. [8]

Nejnovějším a nejrychlejším vylepšením CAN protokolu je CAN XL (*CAN Extra Long*). CAN XL díky implementaci pulzně šířkové modulaci kódování (PWM –*Pulse Width Modulation*) a většímu datovému poli v zasílaném rámci (až 2048 bajtů), dokáže dosáhnout rychlostí až 20 Mb/s. [9]

1.1.2 FlexRay

FlexRay je automobilová síť používající dvoukanálovou broadcasting sběrnici. Sběrnice tedy využívá dvě dvojlinky (dva kanály), které navyšují rychlost sběrnice a zlepšují odolnosti protokolu vůči chybám. Původní FlexRay sběrnice fungovala jako jednokanálová sběrnice, ale ztrácelo se tím na rychlosti a odolnosti vůči chybám. Kvůli dvoukanálové implementaci je FlexRay jednou z dražších síťových technologií v automobilovém průmyslu. [10]

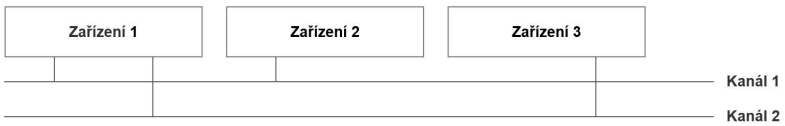
FlexRay protokol řeší problém rozhodování kdo bude vysílat rámce na sběrnici odlišným způsobem než CAN protokol. FlexRay je TTP (*Time-Triggered Protocol*), to znamená, že všechna, ke sběrnici připojená zařízení, mají svůj časový úsek kdy mohou zasílat rámce na sběrnici. Hlavní nevýhodou tohoto způsobu rozhodování je potřeba časové konfigurace zařízení před jeho připojením do FlexRay sítě. [11]

Základní vlastností FlexRay sítě je její dvoukanálová sběrnice, která umožňuje vícero topologických zapojení. Zařízení mohou být připojena do obou kanálů zároveň nebo jen do jednoho. Avšak pouze zařízení připojená na stejný kanál mohou mezi sebou komunikovat. Mezi-kanálová komunikace není implementována. Pokud je tedy zařízení připojeno do vícero FlexRay sítí, musí mezi-síťová komunikace probíhat prostřednictvím komunikačního řadiče. [12]

Podporované sítové topologie ve FlexRay síti:

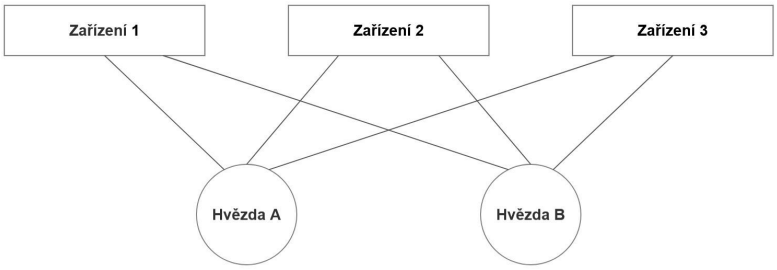
- **Pasivní sběrnicová topologie** (*Passive Bus Topology*), na obrázku (1.1), je konfigurace sběrnicové sítě, ve které mohou být zařízení připojena k jednomu nebo k oběma kanálům FlexRay sběrnice zároveň. V případě RayFlex sběrnice může být ke sběrnici připojeno až 2047 zařízení. [12]

Obrázek 1.1: Pasivní sběrnicová topologie sítě [13]



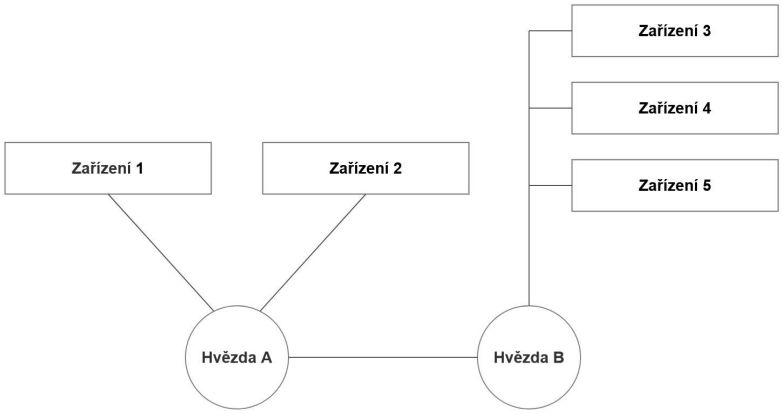
- **Aktivní topologie hvězda** (*Active Star Topology*), na obrázku (1.2), je síťová topologie, kde každý kanál musí být otevřený nebo uzavřený kruh a kde kanál nesmí obsahovat více jak dva hvězdové oddělovače. Signál odeslaný zařízením je hvězdným oddělovačem aktivně směrován k ostatním zařízením připojeným k oddělovači. Topologie hvězda také může vytvořit jednokanálovou síť, kdy jsou zařízení připojena na svůj hvězdný oddělovač a oddělovače jsou následně propojeny jednokanálovou sítí. [12]

Obrázek 1.2: Aktivní topologie sítě hvězda [14]



- **Hybridní topologie**, na obrázku (1.3), je kombinovaná sběrnicová síť využívající topologie pasivní sběrnice a aktivní hvězdy. Například můžeme do existující aktivní hvězdové sítě připojit na oddělovač jednokanálovou pasivní sběrnici. [12]

Obrázek 1.3: Hybridní topologie sítě [15]



1.1.3 Byteflight

Jedná se o předchůdce FlexRay sítě vyvíjeného pod dohledem automobilky BMW. Byteflight sběrnice umožňovala rychlost rámců až 10 Mb/s. Stejně jako jeho následovník FlexRay umožňoval využití topologie hvězda nebo sběrnicové topologie. [16]

Zásadně se však lišil v otázce snížení elektromagnetického rušení. Zatímco FlexRay upřednostňoval použití stíněné kabeláže, Byteflight se vydal cestou optických kabelů, konkrétně polymerových optických kabelů (POF – *Polymer Optical Fiber*). Pro Byteflight byl vyvinut speciální obousměrný optický kabel, kde optický vysílač/přijímač je zabudován v jednom čipu jako světelná dioda a fotodioda. [16]

1.1.4 MOST

MOST sběrnice je multimediální síťová technologie se zaměřením na posílání videa, audia a informačních signálů po sběrnici. MOST je možné využít pro komunikace jak po optickém drátě, tak po elektrickém kabelu. Pro jeho multimediální zaměření se MOST technologie převážně používala pro komunikaci mezi multimediální jednotkou automobilu a ostatními zařízeními typu radio, GPS, DVD přehrávač atd. Multimediální jednotkou myslíme obrazovku umístěnou mezi sedadlem řidiče a spolujezdce, díky níž můžeme ovládat a monitorovat chování automobilu, která nejsou specifická pro jeho řízení. MOST sběrnice podporuje *Plug and Play* technologii umožňující rychlé přidání nového zařízení do sítě nebo výměnu připojeného zařízení. [17]

MOST protokol definuje datový tok jako *point to multi-point* to znamená, že odesílaná data mají zdrojové zařízení a libovolný počet cílových zařízení. V MOST síti je potřeba aby jedno zařízení, z až 64 možných zapojených MOST zařízení, bylo nastaveno jako tzv. *TimingMaster*. Toto zařízení je nejčastěji hlavní multimediální jednotka. *TimingMaster* pravidelně odesílá rámec se systémovým časem, který ostatní připojená zařízení přijímají a tím je zajištěn stejný čas systémových hodin na sběrnici. [18]

MOST existuje ve třech verzích. MOST25 dosahoval rychlosti datového toku po sběrnici až 25 Mb/s. MOST50 dosahoval až 50 MB/s a MOST150 měl maximální rychlost 150 Mb/s. Jedná se o jednotlivé generace MOST technologie, které jsou mezi sebou zpětně kompatibilní. [18]

1.1.5 BroadR-Reach Ethernet

BroadR-Reach je *point-to-point* Ethernet technologie, kde zaslaná komunikace po síťové lince má jednoho konkrétního odesílatele a jednoho konkrétního příjemce. Využívá jednu kroucenou dvojlinku pro maximální rychlost přenosu dat 100 Mb/s a umožňuje tzv. *full-duplex* to znamená, že zařízení může odesílat a přijímat komunikaci po kabelu ve stejný čas. BroadR-Reach Ethernet byl navrhnut tak aby byl interoperabilní se specifikací standardu IEEE 802.3 (Ethernet Standard). [19]

BroadR-Reach technologie je modifikací standardů 1000BASE-T (Gigabit Ethernet) a 100BASE-TX (Fast Ethernet). BroadR-Reach se tedy řídí běžnou praxí definovanou ve standardu IEEE 802.3, kde přenosová cesta informace po síti je kompletně definována specifikací a definice přijímač přenesených dat je ponechána na jeho realizátorovi. Tato adaptace standardu umožňuje značnou míru využijí již existujících síťových prvků v automobilovém průmyslu. [19]

Full-duplex na jedné kroucené dvojlince zajišťuje snížení množství použité kabeláže vůči Fast Ethernetu, resp. Gigabit Ethernetu. Z důvodů automobilového použití, kde se vnitřní kabely mohou značně překrývat a stíněné kabely nejsou pro zachování nízké ceny a lepší flexibility kabelů vždy možné,

má BroadR-Reach Ethernet sníženou šířku pásma. Konkrétně jde o více jak poloviční snížení oproti Fast Ethernetu, resp. Gigabit Ethernetu, a to na šířku pásma 33 MHz. Toto řešení snižuje jak cenu kabeláže (mohou se použít méně kvalitní typy), tak umožňuje agresivnější EMC filtrování (*Electromagnetic Compatibility*) [19]. EMC filtrování zajišťuje, aby elektronická zařízení nevytvářela nebo nebyla ovlivňována elektromagnetickým rušením. [20]

Pro vysokou datovou propustnost se BroadR-Reach Ethernet hodí pro přenos většího objemu dat. Je tedy ideální pro komunikaci mezi zadními, resp. bočním kamerovým systémem a multimediální jednotkou nebo pro přenos informací mezi jednotkou autopilota a multimediální jednotkou. Protože se jedná o relativně novou technologii (2012 [19]) je využití její vysoké přenosové rychlosti především použito pro pasivní ADAS (*Advanced Driver-Assistance Systems*). ADAS je část elektronických zařízeních ve vozu, které napomáhají řidiči v řízení a parkování vozu. Pasivním ADAS myslíme převážně video komunikaci parkovacích kamer s řidičem a hlášení systémem vyhodnocených výstrah řidiči. V interní komunikaci vozu, která je pro řidiče standardně nepřístupná (např. komunikace s elektronickou řídící jednotkou pro vstřikování paliva do motoru) se prozatím dává přednost CAN sběrnici jakožto ověřené a funkční technologii, která oproti BroadR-Reach Ethernetu obsahuje více funkcionalit se zaměřením na detekci chyb a jejich samo-opravě. [21]

1.2 Vnitřní síť v Tesle Model 3

„Model 3 passed all regulatory requirements for production two weeks ahead of schedule.“ – Elon Musk, 3. 7. 2017 [22]

Tesla Inc. je současný největší dodavatel elektrických automobilů, který v roce 2021 měl necelých 14% podílu na trhu s elektrickými vozidly [23]. Tesla Model 3 je jejich nejúspěšnější elektrické vozidlo z roku 2017 [24], a které je nejprodávanějším elektrickým autem na světě s více jak jedním miliónem prodaných kusů, dle dat z roku 2021. [25]

Také jako jeden z mála produkčních automobilů využívá technologii BroadR-Reach Ethernet pro interní síťovou komunikaci. Model 3 používá dvě technologie pro interní komunikaci CAN sběrnici a BroadR-Reach Ethernet. Pro většinu interní komunikace dává Tesla Model 3 přednost systému založeném na Ethernet standardu před CAN sběrnicí, čímž se liší od konkurenčních řešení vnitřní komunikace.

Toto rozhodnutí přináší určité řešení některých problémů CAN sběrnice. Například BroadR-Reach Ethernet řeší autentizační proces při připojení nového zařízení, nemožnost pasivního odposlech připojeného zařízení do sítě přes RJ45 konektor, nemožnost šifrované komunikace atd. Avšak přináší nové problémy specifické pro Ethernet standard.

Tesla Model 3 jakožto EV z roku 2017 už prošlo mnoha změnami ať už na základě podnětů pro zlepšení uživatelské přivětivosti, tak na základě podnětů komunity lidí, kteří se zaměřují na bezpečnostní analýzy v automobilovém průmyslu. Ačkoliv se jedná o nevelkou skupinu lidí, kteří se tomu věnují na profesionální úrovni z důvodů finanční náročnosti koupi samotného automobilu, společnost Tesla se tématem zabezpečení svých automobilů, nejen Modelu 3, značně věnuje.

Číslo celkových počtu softwarových aktualizací prováděné společností Teslo je ovlivněné diverzifikací motorové a funkční výbavy jednotlivých vozů. Všichni uživatelé nemají zakoupený balíček s autonomním řízením, ne všude ve světě je autonomní řízení povoleno, a ne všichni mají pohon na všechny čtyři kola. Tyto rozdílné výbavy významně ovlivňují jednotlivé verze aktualizací, které společnost Tesla zasílá svým zákazníkům přes internet neboli *Over-the-Air Updates*. I přes tyto rozdílné výbavy je společnost Tesla velmi aktivní a dokáže vydat i více jak pět aktualizací v jednom měsíci pro většinu svých vozů. [26]

1.3 Známé bezpečnostní analýzy na Tesla automobilech

Tesla Model 3 už neřadíme mezi nejnovější automobily. Ale ačkoliv se nejedná o nejmodernější elektromobil a už vícero bezpečnostních analýz na něm bylo provedeno, stále se jedná elektromobil, který je průběžně aktualizován a možné nově nalezené zranitelnosti, jsou s ohledem na jeho využití Ethernetu, stále aktuální.

Pro pochopení aktuálního stavu vozu je potřeba se podívat na některé předešlé útoky na Tesla automobily, které byly zveřejněny. Tyto zdokumentované útoky nám pomohou k lepšímu pochopit problematiky automobilové bezpečnosti a umožní sestavení ideálních postupů bezpečnostní analýzy vozu.

V roce 2018 bylo skupinou COSIC zveřejněna zpráva [27] ohledně útoku na bez-klíčový systém moderních automobilů. Útok byl úspěšně proveden na voze Tesla Model S už v roce 2017, kdy poprvé kontaktovali Teslu. Jedná se o útok, který neměl s vnitřní komunikací vozu moc společného, spíše se jednalo o prolomení způsobu generování bezpečnostního klíče. Přesto jde o zranitelnost, kterou je potřeba zmínit.

Čínská společnost *Keen Security Lab of Tencent* v roce 2016 zveřejnila zprávu [28], která popisuje využití zranitelností v Linux CID. Tato zranitelnost umožňuje útočníkovy eskalovat privilegia a umožnit bezdrátový přístup do CAN sběrnice automobilu. Kontaktování Tesly Inc. bylo provedeno po úspěšném provedení útoku na voze Tesla Model S.

Kapitola 2

Teorie bezpečnostní analýzy

V této kapitole se zaměříme na teoretické znalosti potřebně pro úspěšné provedení bezpečnostní analýzy. Nejprve si představíme základní terminologii pro pochopení problematiky penetračního testování a používané hardwarové a softwarové nástroje. Popíšeme důležitost rozdílu mezi bezpečím a zabezpečením. A pro pochopení finálního rozhodnutí při výběru standardu pro penetrační testování si stručně popíšeme některé ve světe používané penetrační standardy. Závěrem si podrobně představíme standard PTES (*Penetration Testing Execution Standard*), který byl vybrán jako předloha pro účely naší bezpečnostní analýzy.

2.1 Používaná terminologie

Protože většina termínů v oboru bezpečnostní analýzy a penetračního testování je v anglickém jazyce, budeme pro většinu z nich používat překlady a definice z Výkladového slovníku kybernetické bezpečnosti [29].

Penetrační test (*Penetration test*) je konkrétní zkoumání funkcí počítačových systémů nebo sítí. Cílem penetračního testu je odhalení slabých míst v zabezpečení, pro účely jejich nápravy. [29]

Hacker a Cracker je nejčastěji osoba zabývající se studiem programovatelných systémů za účelem intelektuálního obohacení. Osoby často bývají programátoři a experti ve svém oboru. Cracker oproti hackeru zneužívá svých znalostí pro porušování zákonů při pronikání do zabezpečených počítačových systémů s cílem škodit jejich vlastníkům. [29]

Bezpečnost a zabezpečení, na první pohled se jedná o podobná slova a často mezi nimi dochází k záměně. Zabezpečením rozumíme nějakou konkrétní ochranu něčeho, co je naše, ať už je to náš dům nebo naše počítačová aplikace. Může se jednat o ochranu před počasím, zlodějem, zvířaty, crackerem atd.

Oproti tomu slovo bezpečnost/bezpečí má význam představující ochranu lidí před nebezpečím, které by mohlo být životu ohrožující. Zabezpečení je tedy z pohledu ochrany nějaké věci/systému a bezpečí zajišťujeme pro lidské osoby.

White hat je přístup k penetračnímu testování, který si zakládá na jeho legálnosti. Penetrační tester/etický hacker má tedy povolení vlastníka softwaru, resp. hardwaru, které je testováno. Veškeré nalezené zranitelnosti jsou následně reportovány vlastníkovy. Etický hacker je většinou motivován finanční odměnou za zaslanou závěrečnou zprávu, bez ohledu na nalezené zranitelnosti. [29], [30]

Black hat je přístup, který si zakládá na ilegálnosti testování softwaru, resp. hardwaru, tudíž se jedná o kybernetický zločin. Hlavním rozdílem oproti white hat přístupu je tedy provádění testů s cílem nalézt slabá místa v zabezpečení systému, bez povolení vlastníka softwaru, resp. hardwaru. V tomto přístupu je opět hlavní motivací finanční odměna. Odměna je však podmíněna nálezem zranitelnosti, která umožní kompromitaci testovaného softwaru, resp. hardwaru. [29], [30]

Slepé testování (*Black box testing*) je typ testování při němž má penetrační tester stejné informace o testovaném objektu a stejně omezený přístup k testovacímu objektu jako běžný uživatel tohoto softwaru, resp. hardwaru. Tester tedy testuje tzv. „naslepo“. [30]

Znalostní testování (*White box testing*) je způsob testování při němž má penetrační tester plný přístup k testovanému systému. Jeho privilegia při testování tedy odpovídají samotným vývojářům testovaného softwaru, resp. hardwaru. [30]

Hrozba (*Threat*) je potenciální nebezpečí, které může vést k poškození systému nebo organizace, při jeho zneužití. Hrozby vždy existují, ale se správným zabezpečením nepředstavují riziko. [29], [30]

Zranitelnost (*Vulnerability*) je slabé místo v systému, které může být zneužito v útočníkův prospěch. [30]

Vektor útoku (*Attack vector*) je cesta nebo série metod, které umožní útočníkovy přístup do počítačového systému nebo sítě. [29], [30]

Attack surface je sada/soubor vektorů útoku. [30]

Zneužití (*Exploit*) je popis způsobu jak prolomit zabezpečení systému za pomoci jeho nalezené zranitelnosti. Výsledkem zneužití je nezamýšlené chování systému nebo samotné narušení jeho zabezpečení. [29], [30]

Zero-day zranitelnost (*Zero-day vulnerability*) je nově nalezená zranitelnost, které byla doposud neznámá a vývojáři systému se zero-day zranitelností mají přesně nula dní na její nápravu. [30]

Sociální inženýrství (*Social engineering*) je „účelová manipulace lidí s cílem přimět je k provedení určité akce nebo k vyzrazení důvěrné informace“. [29]

Security by obscurity je způsob zabezpečení při němž se snažíme něco skrýt, např. nějakou vlastnost programu, namísto abychom cíleně použili metody umožňující správně zabezpečení systému, jako např. šifrování nebo autentizace přístupu. [31]

Útok hrubou silou (*Brute force attack*) je metodou k prolamování hesel za pomoci výpočetní síly útočníkova systému. Útočník zkouší všechny kombinace znaku, ze kterých by se heslo mohlo skládat. Jedná se o časově velmi náročný způsob, který je ukončen nalezením správně permutace znaků, které tvořící heslo. Časová náročnost je ovlivněna délkou hesla, složitostí hesla a výpočetní silou útočníkova stroje. [29]

Odmítnutí služby (*Denial of service*) neboli DoS je typ útoku, při němž dochází k přehlcení cílového stroje požadavky. Přehlcení způsobí pád cílového systému a jeho následovnou nedostupnost. [29]

Podvržení (*Spoofing*) je útok při němž se útočník podvrhuje svoji identitu, resp. komunikaci za účelem oklamání bezpečnostních složek systému a získat neautorizovaný přístup, resp. odpověď. [29]

Síťový port je specifické číslo, které v počítačové síti slouží ke komunikaci mezi dostupnými službami, které běží na zařízeních. Pomocí portu mohou mezi sebou konkrétní služby komunikovat. Čísla jsou z rozsahu 0 až 65535.

Proof of Concept bychom mohli přeložit jako důkaz/ověřitelnost tvrzení. Jedná se o pojem, který slouží jako důkaz něčeho, co tvrdíme, že umíme. Používá se jak v softwarovém, tak hardwarovém inženýrství.

2.2 Softwarové a hardwarové nástroje

Pro potřeby pochopení práce je potřeba se seznámit se softwarovými a hardwarovými nástroji, které byly při bezpečnostní analýze použity.

brctl [32] je volně dostupný nástroj v příkazové řádce který se používá k nastavení, údržbě a kontrole konfigurace ethernetového přemostění v linuxovém jádře. Ethernetový most je způsob propojení dvou ethernetových sítí dohromady, takže se tyto dvě sítě budou jevit připojeným zařízením jako jedna síť.

arping [33] je volně dostupný nástroj v příkazové řádce sloužící k posílaní ARP (*Address Resolution Protocol*) paketů, za účelem nalezení MAC (*Media Access Control*) adresy zařízení, v jedné Ethernet síti, pomocí námi známé IP adresy hledaného zařízení.

arp-scan [34] je volně dostupný nástroj v příkazové řádce umožňující skenování Ethernet sítě pomocí předdefinovaného IP rozsahu. Pro prohledávání v síti připojených zařízení využívá protokol ARP.

nmap [35] je open source nástroj pro průzkum sítě a bezpečnostní audit. Nmap pomocí nezpracovaných IP paketů dokáže určit, kteří hostitelé jsou v síti dostupní, jaké služby (jejich názvy a verze) používají, jaké porty používají, jaký mají operační systém a jeho verzi a mnoho dalších funkcí, které my nebudeme potřebovat.

Burp Suite [36] (verze zdarma) je grafický nástroj pro provádění bezpečnostních testů webových aplikací. Pomocí různých integrovaných nástrojů dokáže provést celý proces penetračního testu, od počátečního mapování a analýzy webu až po nalezení a zneužití bezpečnostních zranitelností. Pro naše účely jsou použity jeho vlastnosti modifikace HTTP dotazů a čtení HTTP hlaviček.

wireshark [37] je volně dostupný síťový analyzátor. Umožňuje sledovat veškerou komunikaci, která na sledovaném rozhraní probíhá. Wireshark umí jak, podrobně analyzovat jednotlivé zachycené pakety a rámce, tak vytvářet analýzu nad veškerou zachycenou komunikací jako celkem.

tcpbridge [38] je další volně dostupný nástroj umožňující přemostění dvou ethernetových sítí. Na rozdíl od nástroje brctl umožňuje tcpbridge větší manipulaci s přeposlanými pakety, umí například přepisování cílové nebo zdrojové IP adresy nebo navyšovat TTL (*time-to-live*) paketů. TTL číslo představuje maximální množství možných Ethernet kompatibilních zařízení, přes které může být paket přeposlán. Při každém přeposlání se TTL číslo sníží o jedna. TTL tedy představuje, jak dlouho bude paket v síti existovat, než bude zahozen.

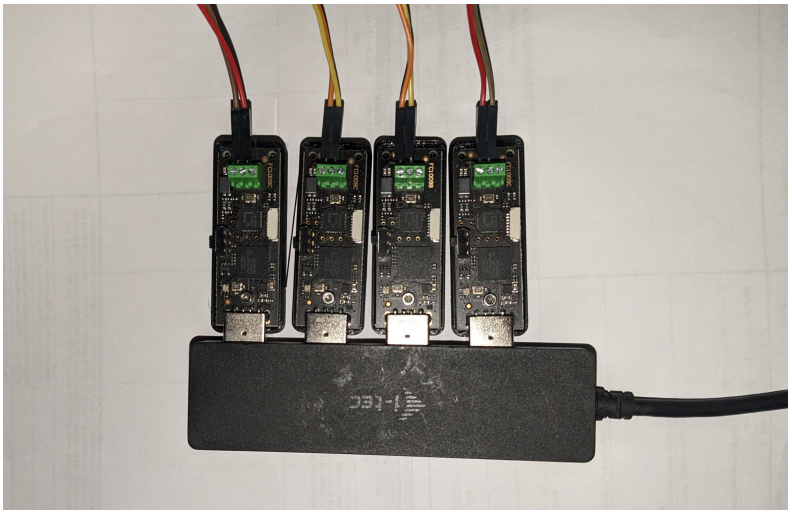
tcpreplay-edit [39] je volně dostupný nástroj umožňující přeposílání paketů ze souborů ve stejné rychlosti ve které byly pakety zachyceny nebo až v maximální rychlosti, kterou hardware umožňuje. Mimo změny rychlosti odesílání umožňuje další možnou editaci paketů, jako je změna TTL, změna cílové nebo zdrojové adresy a další úpravy.

Firefox je volně dostupný webový prohlížeč, který jsme pro potřeby našeho testování použili pro zkoumání otevřených portů na připojených zařízení v síti.

FC602 USB Stick [40] je kompaktní hardwarové rozhraní propojující PC, pomocí USB 2.0, s automobilovým síťovým rozhraním používající jednu Ethernet dvoulinku (v našem případě BroadR-Reach Ethernet). FC602 se po připojení do PC chová jako běžné Ethernet rozhraní, což umožňuje testování kabelů a diagnostiku automobilové sítě. Rozhraní můžeme vidět připojené v USB Hubu na obrázku (2.1).

USB 3.0 Hub je jednoduché zařízení umožňující připojit více USB zařízení do jednoho USB rozhraní.

Obrázek 2.1: FC602 rozhraní připojené do USB Hubu



2.3 Standardy penetračního testování

Pro účely penetračního testování byly vytvořeny konkrétní penetrační standardy, které je potřeba si stručně popsat. Jedná se známé a ve světě používané testovací standardy. Vlastnosti a funkcionalita standardů, nám pomůže lépe vybrat, pro naše potřeby, nejlepší testovací standard.

2.3.1 OWASP – Open Web Application Security Project

OWASP je bezplatný a otevřený projekt komunity z celého světa. Cílem projektu je lepší zabezpečení softwaru webových aplikací. Hlavním zaměřením projektu je tedy testování webových aplikací se zaměřením jak na lidskou činnost, tak na technologickou stavbu aplikace. Nejedná se tedy o univerzální návod pro penetrační testování. Webové zaměření projektu jej činí neideálním pro naše potřeby testování automobilové sítě. [41]

2.3.2 NIST – National Institute of Standards and Technology

Jedná se o národní institut pod správou Ministerstva obchodu Spojených států amerických. NIST klade důraz na kybernetickou bezpečnost kritických infrastruktur, které jsou v souladu s NIST rámcem spadajícím pod vládu USA. Přestože NIST podporuje testování technologií všech velikostí, od testování mikroprocesorů až po testování globálních komunikačních sítí, vládní zaměření NIST institutu nepředstavuje ideálního kandidáta pro naší bezpečnostní analýzu. [42]

2.3.3 ISSAF – Information Systems Security Assessment Framework

Ačkoliv ISSAF je již starší metodologií bez žádné aktuální správy je ve světě stále využíván z důvodů jeho propojování jednotlivých kroků penetračního testování s konkrétními nástroji pro jejich testování. Hlavním zaměřením této metodologie jsou informační systémy firem a jak s nimi síťový uživatelé zachází. Z důvodů zastaralosti a firemního zaměření jsme ISSAF nevybrali jako rámec našeho penetračního testování. [43]

2.3.4 OSSTMM – Open Source Security Testing Methodology Manual

Jde o doporučující manuál o bezpečnostních zkouškách a bezpečnostních analýzách. Metodologie testování je běžně aktualizováno a proto současné. Má aspekty zaměřující se jak na zabezpečení lidských interakcí s technologii, tak na zabezpečení technologie samotné. Bohužel OSSTMM neumožňuje dostatečný prostor pro modifikaci postupů, který je pro naši bezpečnostní analýzu kritický. [44]

Déle existuje standard PTES (*Penetration Testing Execution Standard*), který jsme si vybraly jako metodologii pro naše penetrační testování. Podrobně je popsán v následující kapitole [2.4].

2.4 PTES – Penetration Testing Execution Standard

Pro potřeby této práce byl vybrán standard PTES (*Penetration Testing Execution Standard*), jako návodu postupu při penetračním testování. Standard je univerzální a jednoduše modifikovatelný, a proto ideální pro testování interní sítě v automobilu, pro které zatím žádný konkrétní standard neexistuje.

PTES obsahuje jednotlivé pokyny pro úspěšné penetrační testování. Skládá se ze 7 hlavních sekcí, které pokrývají veškeré potřebné pokyny pro testování. Jednotlivé sekce standardu pouze popisují teorii penetračního testování v dané sekci. Konkrétní nástroje, které se mohou využít k testování jsou dostupné samostatné v technickém návodu [45] dodávaném standardem PTES.

2.4.1 Návrh penetračního testování

Před samotným testováním je potřeba se se zákazníkem, pro kterého budeme penetrační testování provádět, domluvit na rozsahu testování (ang. *Testing scope*). Předem domluvený rozsah testování nám umožňují vytvořit takový způsob testování, který bude na míru vytvořený pro potřeby zákazníka.

Rozsah testování může, mimo jiné, obsahovat jaké konkrétní zařízení se mají testovat, jaké webové aplikace jsou v testovacím rozsahu anebo jaký rozsah IP adres a síťových domén se má otestovat. Zákazníci mohou požadovat test na ověření bezpečného chování svých zaměstnanců, takže je potřeba nadefinovat způsob sociálního inženýrství, které se bude na zaměstnancích testovat. Bude mít penetrační tester fyzický přístup do testované sítě nebo pouze vzdálený? Takovéto a další otázky je potřeba před samotným testem zadefinovat a sepsat pro budoucí potřeby vytváření testů. V návrhu penetračního testování nesmí chybět ani cíle testování, jejichž splnění se v závěru testování promítne na hodnocení. [46]

Po dohodě detailů samotného testování se musí domluvit časové rozsahy testování a jejich uzávěrky. Musí sepsat legální kontrakt, který bude obsahovat všechny detaily a informace o penetračním testu. Tento dokument je hlavně důležitý z pozice penetračního testera, kvůli jeho ochraně před možnými komplikacemi ze strany zákazníka, například kdyby chtěl měnit parametry testů už po tom co byly provedeny podle parametrů ve smlouvě. [46]

2.4.2 Shromažďování informací

Jedná se o průzkum cílové infrastruktury zákazníka s cílem nashromáždit co nejvíce informací, které by se dali využít k jeho průniku. Čím více informací získáme, tím větší je náš attack surface pro budoucí použití. Jedná se tedy o aktivitu za účelem získat přehledu o tom, co kde běží a jaké služby se tam k běhu infrastruktury používají.

PTES využívá OSINT (*Open Source Intelligence*) strukturu, která má tří podoby:

- **Pasivní shromažďování informací** je způsob sběru dat, když je hlavní důraz kladen na požadavek, aby cíl nikdy nezjistil činnosti shromažďování informací. Při tomto shromažďování dat, nikdy neposíláme žádný provoz do testované infrastruktury, a proto se jedná o velmi obtížný typ sběru dat. Můžeme tedy používat pouze informace archivované nebo uložené, avšak tyto informace mohou být zastaralé a tudíž zavádějící. [47]

- **Semi-pasivní shromažďování informací** je typ shromažďování dat, kdy shromažďujeme data za pomocí odesílání požadavků, které se vydávají za běžnou, tudíž nepodezřelou komunikaci. Neprovádíme sken portů na konkrétních IP adresách a sledujeme pouze metadata, data o datech, v publikovaných dokumentech a zdrojích. Cíl testování by tedy neměl detekovat, žádnou nezvyklou aktivitu. [47]

- **Aktivní shromažďování informací** je metoda shromažďování informací, při které je naše testování cílem detekováno a označeno za hrozbu. Při tomto typu sběru dat aktivně mapujeme síťovou infrastrukturu, aktivně skenujeme zranitelnosti služeb na otevřených portech a vyhledáváme nepublikované soubory a dokumenty. [47]

Na závěr by se měla provést identifikace ochraných mechanismů, které zákazník využívá. Může se jednat o šifrování dat na síťové komunikaci, o povolení připojení pouze konkrétním oprávněným osobám nebo o samotnou ochranu uživatelů pomocí emailových filtrů na spam. [47]

2.4.3 Modelování hrozeb

PTES nevyužívá konkrétní model hrozeb, ale vyžaduje konzistentní návrh modelu, pro účely možného opakování testů se stejnými výsledky. Standard se zaměřuje na dva prvky, aktiva (*assets*) a útočníka. Aktiva se dále dělí na firemní aktiva a firemní procesy. Prvky útočníka se dělí na skupiny ohrožení a jejich schopnosti. Minimálně tyto čtyři prvky by měli být identifikovány v rámci penetračního testování. [48]

Během analýzy firemních aktiv je analýza zaměřena na všechna aktiva a jejich firemní procesy, které je podporují. Tato analýza probíhá shromažďováním

dat z dokumentace a rozhovory s příslušnými pracovníky. Díky tomuto je penetrační tester schopen identifikovat aktiva, na která se útočník s největší pravděpodobností zaměří. Tester dokáže určit hodnotu aktiv a dopad při jejich částečné/úplné ztrátě. [48]

V analýze obchodních procesů rozlišujeme kritické firemní procesy a nekritické firemní procesy. Pro každou kategorii je analýza stejná a bere v potaz stejné prvky. Hlavní rozdíl je ve váze, která je přiřazena k možné hrozbě, při ohrožení firemního procesu. Váha hrozeb se však vztahuje na jednotlivé procesy, může totiž nastat situace, při které spojením vícero nekritických hrozeb vznikne hrozba kritická. Takové scénáře hrozeb by měly být také identifikovány a zmapovány pro pozdější použití v penetračním testu. [48]

Při definování relevantních ohrožujících skupin by měla být poskytnuta jasná identifikace hrozeb, pokud jde o umístění skupiny vůči organizaci (interní nebo externí) a jakékoliv další relevantní informace, které by pomohly ohrožující skupině k zneužití hrozby. [48]

Po vytvoření ohrožujících skupin se musí analyzovat jejich schopnosti, za účelem zjištění pravděpodobnosti, že konkrétní skupina úspěšné kompromituje testovanou organizaci. V této analýze by se měla provést jak technická analýza, tak příležitostní analýza. [48]

2.4.4 Analýza zranitelností

Samotná analýza zranitelností je proces, při kterém se se snažíme odhalit nedostatky v testovaném systému, které by mohl útočník využít ve svůj prospěch. Samotné hledání chyb/nedostatků se liší svou podobou v ohledu na to jakou komponentu testujeme. Je tedy zásadní rozdíl v procesu testování webové aplikace a služby běžící na otevřeném portu. Analýzu zranitelností často rozdělujeme na aktivní a pasivní. [49]

Aktivní testování zahrnuje přímou interakci s komponentou, která je testována na bezpečnostní zranitelnosti. Existují dva různé způsoby interakce s cílovou komponentou, automatizovaný a manuální. [49]

Automatizované testování využívá software k interakci s cílovou komponentou, ke zkoumání jejích reakcích a k určování, zda existuje zranitelnost na základě těchto odpovědí. Automatizovaný proces často sníží požadavky na čas a práci testera. Při automatizovaném testování často získáme data, která jsou následně prozkoumána a často využita k další, už manuální, analýze. [49]

Pasivní analýza obsahuje zkoumání metadat a odposlech provozu mezi zařízeními, kteří mezi sebou komunikují po konkrétní síti. Jedná se o časově náročnou analýzy, například pro kvalitní analýzu odposlechu sítě potřebujeme dostatečné množství dat, které někdy musíme sbírat i několik hodin. [49]

Po dokončení analýzy zranitelností dochází k validaci zjištěných skutečností a následný průzkum zneužitelnosti konkrétních nalezených zranitelností. [49]

2.4.5 Zneužití zranitelnosti

V této fázi bezpečnostní analýzy dochází k samotnému testování zneužití zranitelností, které byly nalezeny během analýze zranitelností. Důležitost v této fázi je kladena na to aby útok nebyl detekován firmou, na kterou se útok provádí. Jako tester musíme být schopni upravovat již existující metody zneužití zranitelnosti, které se nalezly během fáze analýzy zranitelností, aby se daly zneužít v naší konkrétní situaci. [50]

Pří útoku se často setkáváme s protiopatřeními jejichž cílem je zabránit úspěšnému zneužití zranitelnosti. Metody zabraňující úspěšné zneužití zranitelnosti obsahují různé alarmy, které detekují pokusy o útok. Úkolem této fáze je tedy zůstat nenápadný a nespustit žádný alarm. Mezi různá protiopatření patří, anti-virus, zakódování, zašifrování nebo bílá listina (*whitelisting*). [50]

Pokud nedošlo k nalezení zranitelnosti, která by se dala zneužít, tak se musíme pokusit o nalezení nových zranitelností tzv. zero-day zranitelností. Jedná se o velmi náročný postup, který vyžaduje značné znalosti o testovaném cílovém systému. Mezi nejčastější metody k nalezení zero-day zranitelností patří fuzzing a analýza zdrojového kódu. [50]

Fuzzing je schopnost znovu vytvořit protokol nebo aplikaci a pokusit se odeslat data do cílové aplikace v naději identifikovat zranitelnost. Nejčastěji dochází k selhání cílové aplikace a díky tomu k následnému vytvoření zranitelnosti. V případě *fuzzingu* se útočník pokouší vytvořit konkrétní zranitelnost z něčeho, co dosud nebylo objeveno. [50]

Analýza zdrojového kódu je další možností nalezení nové zranitelnosti. Bohužel možnost nahlédnout do zdrojového kódu není běžná záležitost a často je tedy nedostupný pro penetračního testera. [50]

2.4.6 Postup po zneužití zranitelnosti

Účelem této fáze je určit hodnotu ohroženého systému a zachovat kontrolu útočníka nad systémem pro pozdější použití. Hodnota stroje pod kontrolou útočníka je určena citlivostí dat na něm uložených a jeho užitečností při dalším kompromitování sítě. Identifikujeme a zdokumentujeme citlivá data a konfigurační nastavení, komunikační kanály a vztahy s jinými síťovými zařízeními, které lze zneužít k dalšímu získání přístupu do sítě. V této fázi bychom si měli nastavit na stroji metody k pozdějšímu přístupu ke stroji, tzv. „zadní vrátka“. [51]

Během tohoto postupu nesmíme zapomínat na dodržování předem dohodnutých pravidel se zákazníkem, který je testován. Tyto pravidla chrání jak zákazníka, aby například nedošlo k omezení služeb, které jsou potřeba pro plynulý chod firmy, tak nás jako penetračního testera, aby nedošlo k porušení zákona. [51]

Po úspěšném přístupu k cílovému systému, po splnění cílů z fáze návrhu penetračního testování a po vytvoření, způsobu jak se do stroje dostat i v bu-

doucnu musíme po sobě uklidit. Odstraníme všechny spustitelné soubory, skripty a dočasné soubory z kompromitovaného systému. Systém vrátíme do původního nastavení, pokud bylo během penetračního testu změněno. Na závěr odstraníme veškeré metody, které sloužili k zpětnému přístupu do systému. [51]

2.4.7 Nahlášení výsledků penetračního testování

PTES nepopisuje konkrétní kroky při hlášení a zveřejňování výsledků penetračního testu, ale pouze popisuje, jaké položky v závěrečné zprávě nesmí chybět. Standard dělí hlášení výsledků na dvě části, shrnutí a technickou zprávu. [52]

Shrnutí je část závěrečné zprávy o penetračním testu, která je primárně určena pro vedení testovaného subjektu a skupiny zasažené nalezenými zranitelnostmi. Stručně popíšeme účel testů, nalezené hrozby a zhodnotíme jejich závažnost pomocí předem domluveného uceleného hodnotícího systému. Na závěr doporučíme kroky k nápravě nalezených zranitelností. [52]

Technická zpráva obsahuje podrobný popis celého procesu penetračního testování. Cílová skupina zprávy jsou lidé, kteří se budou nalezenými zranitelnostmi aktivně zabývat a jsou zodpovědní za následné kroky vedoucí k jejich nápravě. [52]

2.5 Pokyny společnosti Tesla k odpovědnému zveřejňování zranitelností

Společnost Tesla má své vlastní pokyny, jak postupovat při zveřejňování výsledků penetračních testů, které jsou prováděny na jejich automobilech/strojích. Pokud budeme jakožto penetrační tester své nálezy hlásit společnosti Tesla, tak se společnost Tesla vzdává svého práva provádět jakékoliv právní kroky proti naší straně. Toto platí za předpokladu, že se budeme řídit následujícími kroky:

- Poskytněte podrobnosti o zranitelnosti, včetně informací potřebných k reprodukci a ověření zranitelnosti a *Proof of Concept*. Jakákoliv zranitelnost, která implikuje funkci, která se nenachází ve vozidle registrovaném pro výzkum, musí být nahlášena do 168 hodin a nula minut (7 dní) od identifikace zranitelnosti.
- Snažte se v dobré víře vyhnout narušení soukromí, zničení dat a přerušení nebo zhoršení našich služeb.
- Neupravujte ani nepřistupujte k údajům, které vám nepatří.
- Před zveřejněním jakýchkoli informací dejte společnosti Tesla přiměřený čas na nápravu problému.

- Upravujte pouze vozidla, která vlastníte nebo máte oprávnění k přístupu.
- Neohrožujte bezpečnost vozidla ani nevystavujte ostatní nebezpečnému stavu.
- Bezpečnostní výzkum je omezen na bezpečnostní mechanismy binárních souborů infotainmentu, binárních souborů Gateway, ECU vyvinutých společností Tesla a energetických produktů.

Plné originální znění nalezneme na [53].

Kapitola 3

Koncept bezpečnostní analýzy

V této kapitolu si popíšeme provedené úpravy standardu PTES pro jeho maximální využití pro naše potřeby testování a nadefinujeme rozsah našeho testování. Také probereme možné systémy hodnotící zranitelnosti a navrhneme model hrozeb penetračního testování automobilu Tesla Model 3.

3.1 Modifikace standardu PTES

Jak již bylo zmíněno, jako metodologie pro tuto práci byl zvolen standard PTES pro jeho jednoduchou modifikovatelnost a univerzálnost v odvětví penetračního testování. Na PTES bylo pohlíženo jako na návod při vypracovávání bezpečnostní analýzy, proto některé části, které standard doporučuje byly vynechány nebo zásadně upraveny.

Standard PTES byl navrhován se zaměřením na bezpečnostní analýzy firemních sítí a jejich subjektů, ale naším testovacím subjektem je pouze automobil Tesla Model 3, proto společnost Tesla a její pracovníci nejsou subjekty této penetrační analýzy. A tudíž veškeré části/fáze standardu PTES, které se zaměřují na firemní aktiva a jejich procesy, nebyly testovány. Ani sociální inženýring ani byznys analýza na společnosti Tesla nebyli prováděny.

Post Exploitation [2.4.6] fáze standardu PTES byla vynechána a nahrazena Pokyny společnosti Tesla k odpovědnému zveřejňování zranitelností [2.5]. Tato fáze popisuje jedinou interakci se společností Tesla, která byla provedena, kdy jsme společnosti sdělili nalezené zranitelnosti a byl ji dán prostor na její vyjádření.

3.1.1 Experimenty v rámci penetračního testování

Nejdůležitější změnou standardu PTES byla struktura postupu při penetračním testování. Fáze shromažďování informací [2.4.2], analýza zranitelností [2.4.4] a zneužití zranitelnosti [2.4.5] byly upraveny aby více odpovídali postupům,

resp. experimentům ve vědeckých pracích. Tyto experimenty mají běžné fáze testování: cíl, příprava, provedení a závěr.

Přístup kdy k penetračnímu testování přistupujeme jako k vědeckým experimentům umožňuje lepší reprodukovatelnost bezpečnostní analýzy a zajišťuje ucelenou strukturu postupu při penetračním testování. Experimenty dále umožňují lepší zachycení procesu bezpečnostního testování, kdy na počátku nic o testovaném subjektu nevíme, ale s dalšími experimenty se dozvídáme více informací, které nám otevírají další možnosti, jak subjekt testovat. Tento přístup nám tedy umožní, na konci bezpečností analýzy, vizualizaci jednotlivých kroků testování, které byly během analýzy provedeny.

3.2 Systém hodnocení zranitelností

Pomocí hodnotícího systému hodnotíme veškeré nalezené zranitelnosti a proto je třeba aby byl objektivní. Toto hodnocení napomáhá při určování priority, které hrozby jsou bezprostřední a musí se na jejich nápravě pracovat co nejdříve a naopak, které hrozby mohou být ponechány na pozdější řešení.

Tato práce využívá hodnotící systém CVSS v3.1 (*Common Vulnerability Scoring System version 3.1*), konkrétně jeho veřejně dostupný kalkulátor [54]. Hodnotící rozsahu můžeme vidět na tabulce (3.1). Jedná se o aktualizovanou verzi, která zahrnuje nová hodnotící měřítka jako je fyzický vektor útoku, nové měřítko pro hodnocení uživatelské interakce pro úspěšný útok a nové pole zahrnující rozsah útoku, jestli útok nezasahuje i do jiných částí ohroženého sytému. [55]

Tabulka 3.1: Tabulka rozsahů hodnotícího systému CVSS v3.1

Popis hodnocení	**Rozsah hodnocení**
Žádný (*none*)	0.0
Nízký (*low*)	0.1 – 3.9
Střední (*medium*)	4.0 – 6.9
Vysoký (*high*)	7.0 – 8.9
Kritický (*critical*)	9.0 – 10.0

Ani verze 3.1 není dostatečná pro hodnocení zranitelností v automobilovém průmyslu. Její hlavní nedostatek je neexistence hodnocení zranitelnosti s ohledem na lidské životy. Tento nedostatek má většina ve světě standardně používaných hodnotících systémů. Proto musíme vytvořit speciální dodatek, který bude ve svém hodnocení zohledňovat bezpečí lidských životů. Tento dodatek bude ke každé nalezené zranitelnosti přidán vedle CVSS v3.1 hodnocení.

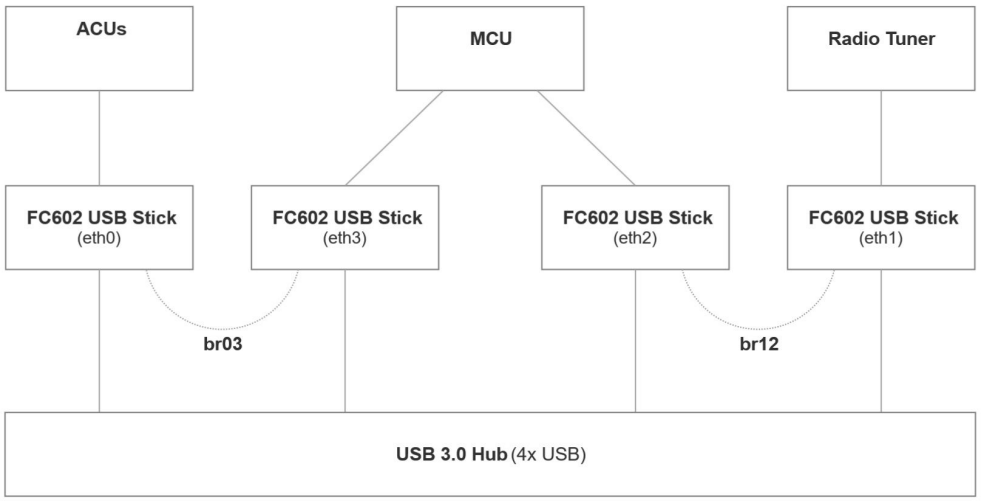
Dodatek má následující hodnotící kritéria:

- **Žádné ohrožení** – lidské životy nejsou v ohrožení ani přímo, ani nepřímo po interakci útočníka s automobilem. Například může jít o únik informací o uživateli auta nebo o neautorizované odemčení vozu.
- **Střední ohrožení** – lidské životy jsou v nepřímém ohrožení po úspěšném útoku útočníka na vůz. Například k tomu může dojít při neautorizovaném restartu obrazovky multimediální jednotky, které ale nezpůsobí žádné zhoršení schopnosti ovládat automobil řidičem.
- **Vysoké ohrožení** – lidské životy jsou v přímém ohrožení při útoku na automobil. Například při převzetí plné kontroly útočníkem nad vozem.

3.3 Rozsah testování

Rozsah testování je BroadR-Reach Ethernet propojení mezi multimediální řídící jednotkou (MCU – *Media Control Unit*) a řídícími jednotkami autopilota (ACUs – *Autopilot Control Units*), kde jedna jednotka je v pohotovostním režimu, a tunerem rádia automobilu. Testování propojení těchto zařízeních bylo umožněno pomocí FC602 USB rozhraní [2.2] a USB hubu. Konkrétní propojení a přemostění mezi zařízeními je vidět na schématu (3.1). Testování má charakter slepého testování [2.1].

Obrázek 3.1: Schéma zapojení testovací vnitřní sítě



3.4 Model hrozeb

Firemní aktiva byly nahrazeny aktivy, které jsou důležité pro majitele automobilu a firemní procesy s aktivy byly nahrazeny procesy, které souvisí s využitím automobilu. Skupiny ohrožení a jejich schopnosti byly ve fázi modelování hrozeb vynechány pro jejich firemní orientaci.

Protože tato bakalářské práce má jako jeden z úkolů aktualizovat řešení diplomové práce [56], je modelování bezpečnostních hrozeb mezi pracemi velmi podobné, v některých případech totožné. Tato skutečnost také vyplývá z podobnosti zadání obou prací.

3.4.1 Aktiva

Seznam aktiv (3.2), které jsou součástí testovaného rozsahu a jsou důležitá pro vlastníka vozu.

Tabulka 3.2: Seznam aktiv

Název aktiva	Popis aktiva
Samotný automobil	Samotný automobil jako takový je důležitým aktivem pro vlastníka vozu. Jakýkoliv neautorizovaný průnik do vozu by mohl mít škodlivé následky.
Data o uživateli	Jedná se o data, které jsou v automobilu uložena a mohou být spojena s vlastníkem vozu (např. lokace vozu, cestovní logy nebo personalizované informace o uživateli).

3.4.2 Procesy

Seznam procesů (3.3), které jsou spojeny s užíváním vozu.

Tabulka 3.3: Seznam procesů

Název procesu	Popis procesu
Lidská interakce	Jakákoliv lidská interakce s vozem od jízdy až po lokalizaci automobilu na mapě na displeji multimediální jednotky.
Procesy vozu na pozadí	Jedná se o procesy, které fungují nezávisle na interakci uživatele s vozem a jsou zodpovědné za korektní běh vozu.

3.4.3 Hrozby

Za pomoci aktiv a procesů automobilu můžeme definovat možné hrozby (3.4). Při definování hrozeb jsme také použili model pro identifikaci hrozeb s názvem STRIDE. STRIDE je akronym pro následující anglická slova představující jednotlivé kategorie bezpečnostních hrozeb: *Spoofing* [2.1], *Tampering* (manipulování s daty), *Repudiation* (nemožnost ověřit kdo způsobil danou situaci), *Information Disclosure* (únik informací), *Denial of Service* [2.1] a *Elevation of Privileges* (získání vyšších oprávnění).

Tabulka 3.4: Seznam hrozeb

Přístupové zařízení	**Popis hrozby**	**STRIDE**
CAN debug konektor*	Připojení do CAN sítě by mohlo vést k neautorizovanému přístupu. Útočník by mohl zasílat jakékoliv zprávy posílané po sběrnici a tím podvrhnout síťovou komunikaci.	S, T
Nabíjecí konektor*	Nabíjecí konektor je součástí CAN sběrnice (jedna z mnoha ve voze). Nabíjecí stanice by neměla mít možnost posílat data přímo na sběrnici a nabíjecí konektor by neměl odesílat stanici data, které nepotřebuje k účelům správného dobíjení baterie.	S, T
Člověk*	Vlastník automobilu má celkem 4 možnosti jak získat přístup do vozu. Tři z těchto možností mohou být ukradeny nebo ztraceny (karta, klíčenka a telefonní mobil s Tesla aplikací). Tyto 3 zařízení mohou útočníkovi dát plný přístup k ovládání automobilu. Čtvrtou možností je volitelný PIN kód bez nějž nejde automobil nastartovat. PIN je pouze čtyřmístný kód, který může být slabý a tudíž pro útočníka lehce prolomitelný.	T, R, I, E
Wi-Fi připojení*	Wi-Fi hotspot, pomocí něhož může automobil komunikovat s internetem, může odposlouchávat veškerou příchozí i odchozí komunikaci. Odesílané pakety mohou být hotspotem upravovány a automobil se o tom nikdy nedozví. Také otevřené porty internetového rozhraní automobilu a na nich běžící služby, které jsou viditelné z internetu, by mohly být útočníkem zneužity.	S, T
LTE připojení*	Stejné hrozby jako u Wi-Fi hotspotu, monitorování internetové komunikace a útok na běžící služby.	S, T
RJ45 konektor*	Připojení do interní sítě by mohlo být kritickou hrozbou. Útočník by mohl podvrhnout síťovou komunikaci a tím získat možnost ovládat automobil.	S, T, D, E
Přímé připojení do Ethernet sítě	Jedná se o stejný rozsah hrozeb jako přes připojení pomocí RJ45 konektoru. Při přímém připojení můžeme navíc filtrovat veškerou síťovou komunikaci, kterou přemosťujeme a tím ovlivnit chování připojených zařízeních.	S, T, D, E

* mimo rozsah testování

Kapitola 4

Bezpečnostní analýza

V závěrečné kapitole si popíšeme proces bezpečnostního testování podle nadefinovaných postupů v předešlé kapitoly. V experimentální části kapitoly se zaměříme na penetračního testování interní sítě vozu. Následně si shrneme nalezené zranitelnosti a zdokumentujeme je v souladu s pokyny společnosti Tesla a s doporučeními standardu PTES. Tato bakalářská práce představuje aktualizované řešení bezpečnostní analýzy provedené v diplomové práci [56], proto kapitolu uzavřeme srovnáním získaných dat z obou prací.

4.1 Bezpečnostní experimenty

V experimentech se budeme věnovat prvotní analýze interní sítě, následné hlubší analýze připojených zařízení a na závěr se budeme věnovat možným nalezeným zranitelnostem. Samotné penetrační testování bylo rozděleno do dvou částí, aktivní a pasivní.

Aktivní část je zaměřená na penetrační testování jednotlivých běžících služeb na každém připojeném zařízení. Pro tuto část procesu využijeme automatizované softwary. Pasivní část sestává pouze z monitorování provozu mezi připojenými zařízeními. Žádná metadata nebyla analyzována.

Testování bylo možné přes síťové mosty přímo připojené do BroadR-Reach sítě. Každý test byl spuštěn, když byly dveře automobilu odemčeno, ale schopnost vozu řídit byla deaktivována (tj. motor byl uzamčen a některé ze senzorů vozu byly deaktivovány).

Testování bylo prováděno na firmwaru ve verzi v10.2 2021.40.6 eed31525bfea.

4.1.1 Experiment: Prvotní průzkum interní sítě automobilu

Cílem prvotního průzkumu je zjistit podobu síťové topologie testované interní sítě automobilu, IP adresy připojených zařízení a o jaká zařízení se jedná.

Pro potřeby tohoto experimentu využijeme softwarové nástroje brctl, arp-scan a Wireshark [2.2]. Následně použijeme 4-krát FC602 USB Stick a USB 3.0 Hub pro připojení k interní síti vozu [2.2].

Dvoje přemostění (3.1) byla vytvořena pomocí nástroje brctl za účelem monitorování a testování sítě. Přes tyto dva mosty prochází veškerá komunikace mezi MCU a ACUs a mezi MCU a rádio tunerem. Mosty jsou pojmenovány *br03* (propojení *eth0 – eth3*) a *br12* (propojení *eth1 – eth2*).

4.1.1.1 Přemostění br03

Nejprve jsme provedli ARP sken (4.1) na přemostění *br03*, abychom nalezli zapojená zařízení a jejich IP adresy.

Obrázek 4.1: ARP sken na přemostění *br03* (arp-scan 192.168.90.0/24 -I br03)

```
┌──(root㉿ASUS-KALI)-[~]
└─# arp-scan 192.168.90.0/24 -I br03
Interface: br03, type: EN10MB, MAC: fc:c2:3d:11:34:52, IPv4: 192.168.90.101
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.90.100  a4:34:d9:01:02:03       Intel Corporate
192.168.90.103  00:02:5a:c5:4f:00       Catena Networks
192.168.90.105  00:43:58:85:4c:02       (Unknown)

99 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.002 seconds (127.87 hosts/sec). 3 responded
```

Z ARP skenu (4.1) jsme zjistili, že na přemostění *br03* jsou připojena tři zařízení. Pomocí nástroje Wireshark, ve kterém jsme analyzovali síťovou komunikaci, jsme zjistili, že IP adresu 192.168.90.100 má jednotka MCU a adresy 192.168.90.103 a 192.168.90.105 patří jednotkám autopilota. Autopilot v pohotovostním režimu je na IP adrese končící na .105.

4.1.1.2 Přemostění br12

Obdobně jako při skenování přemostění *br03* jsme začali s nástrojem arp-scan (4.2) na síťovém mostu *br12*.

Obrázek 4.2: ARP sken na přemostění *br12* (arp-scan 192.168.90.0/24 -I br12)

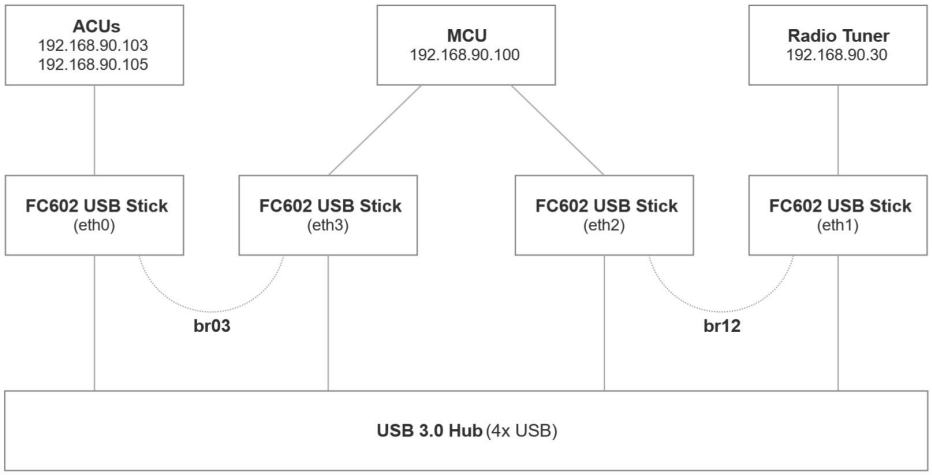
```
┌──(root㉿ASUS-KALI)-[~]
└─# arp-scan 192.168.90.0/24 -I br12
Interface: br12, type: EN10MB, MAC: fc:c2:3d:11:49:f2, IPv4: 192.168.90.102
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.90.30   2c:6b:7d:c4:41:3f       Texas Instruments
192.168.90.100  a4:34:d9:01:02:03       Intel Corporate

65 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.017 seconds (126.92 hosts/sec). 2 responded
```

Opět pomocí nástroje wireshark, jsme zjistili, že na IP adrese 192.168.90.30 se nachází rádio tuner a na adrese 192.168.90.100 se nachází jednotka MCU, která má stejnou MAC adresu jako na přemostění *br03*.

Díky získaným informacím z průzkumu sítě jsme dokázali vytvořit síťovou topologii (4.3) interní sítě vozu. Tato topologie nám pomůže k lepšímu otestování funkčnosti interní sítě automobilu.

Obrázek 4.3: Síťová topologie vnitřní sítě vozu Tesla Model 3



4.1.2 Experiment: Sken zranitelností na zařízeních připojených do sítě

Účelem tohoto experimentu je zjistit jaké služby běží na zařízeních připojených do interní sítě. Běžící služby, konkrétně jejich verze, poslouží jako podklad k nalezení jejich možných zranitelností.

Pro provedení skenu potřebujeme softwarové nástroje brctl a nmap [2.2] a hardwarové nástroje FC602 USB Stick a USB 3.0 Hub pro připojení k interní síti vozu [2.2].

4.1.2.1 Přemostění br03

Nejprve jsme použili nmap port skener na jednotku MCU. Tomuto postupu však bránila skutečnost, že MCU odpovídá na SYN pakety, které zasílá nmap skener, pouze z IP adres, které náleží autopilotům (4.4).

Obrázek 4.4: Security by Obscurity na síťovém přemostění *br03*

```
┌──(root㉿ASUS-KALI)-[~]
└─# ip a a 192.168.90.101/24 dev br03

┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -T4 192.168.90.100 -e br03
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:10 CEST
Nmap scan report for 192.168.90.100
Host is up (0.00035s latency).
All 1000 scanned ports on 192.168.90.100 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: A4:34:D9:01:02:03 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 21.32 seconds

┌──(root㉿ASUS-KALI)-[~]
└─# ip a d 192.168.90.101/24 dev br03 && ip a a 192.168.90.103/24 dev br03

┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -T4 192.168.90.100 -e br03
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:10 CEST
Nmap scan report for 192.168.90.100
Host is up (0.00034s latency).
Not shown: 998 filtered tcp ports (no-response), 1 filtered tcp ports (port-unreach)
PORT     STATE SERVICE
8443/tcp open  https-alt
MAC Address: A4:34:D9:01:02:03 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 5.13 seconds

┌──(root㉿ASUS-KALI)-[~]
└─# ip a d 192.168.90.103/24 dev br03 && ip a a 192.168.90.105/24 dev br03

┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -T4 192.168.90.100 -e br03
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:11 CEST
Nmap scan report for 192.168.90.100
Host is up (0.00044s latency).
Not shown: 998 filtered tcp ports (no-response), 1 filtered tcp ports (port-unreach)
PORT     STATE SERVICE
8443/tcp open  https-alt
MAC Address: A4:34:D9:01:02:03 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 7.47 seconds
```

Po tomto zjištění všechny další nmap skeny proběhli bez obtíží. Pro nmap sken jsem použili následující možnosti:

- *-Pn*: předpokládáme, že cílové zařízení je dostupná
- *-n*: přeskakujeme DNS překlad
- *-sV*: pokus o zjištění služby a její verze na oskenovaném portu
- *-sC*: pokus o zjištění běžícího skriptu na portu
- *-O*: pokus o detekci operačního systému cílového zařízení
- *-T4*: šablona rychlosti skenování (čím vyšší, tím rychlejší)
- *-p-*: oskenovat veškeré porty (0-65535)
- *-e*: specifikace síťového rozhraní na kterém bude sken zařízení proveden

Výsledný sken jednotky MCU můžeme vidět na obrázku (4.5).

Obrázek 4.5: Sken softwarem nmap na jednotce MCU (*br03*)
(nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.100)

```
┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.100
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:42 CEST
Nmap scan report for 192.168.90.100
Host is up (0.00045s latency).
Not shown: 65530 filtered tcp ports (no-response), 1 filtered tcp ports (port-unreach)
PORT      STATE SERVICE      VERSION
8443/tcp  open  https-alt?
8444/tcp  open  pcsync-http?
8900/tcp  open  http-proxy   (bad gateway)
| fingerprint-strings:
|   FourOhFourRequest, GetRequest, HTTPOptions:
|     HTTP/1.0 502 Bad Gateway
|     Date: Thu, 26 May 2022 07:47:53 GMT
|     Content-Length: 0
|   GenericLines, Help, NessusTPv10, RTSPRequest, SSLSessionReq, Socks5:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|_    Request
|_http-title: Site doesn't have a title.
20564/tcp open  unknown
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 404 Not Found
|     Content-Type: text/html
|     Content-Length: 0
|     Connection: close
|     Date: 22 Feb 2012 06:58:25 GMT
|_    Server: updater/9927adf99122b47f
MAC Address: A4:34:D9:01:02:03 (Intel Corporate)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: WAP|phone
Running: Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/h:sonyericsson:u8i_vivaz
OS details: Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Ericsson U8i Vivaz mobile phone
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 362.03 seconds
```

Obdobné skenování proběhlo na IP adresách autopilota (4.6), (4.7).

Obrázek 4.6: Sken softwarem nmap na jednotce hlavního autopilota
(nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.103)

```
┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.103
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:51 CEST
Nmap scan report for 192.168.90.103
Host is up (0.00071s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT      STATE  SERVICE             VERSION
22/tcp    closed ssh
8081/tcp  open   ssl/blackice-icecap?
| ssl-cert: Subject: commonName=1462554-01-H-B7S19109A00948/organizationName=Tesla/stateOrProvinceName=California/countryName=US
| Not valid before: 2021-02-05T08:58:08
|_Not valid after:  2023-02-05T08:58:08
8901/tcp  open   http                Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
25974/tcp closed unknown
28496/tcp open   http                Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
MAC Address: 00:02:5A:C5:4F:00 (Catena Networks)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.1
OS details: Linux 5.1
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 111.41 seconds
```

Obrázek 4.7: Sken softwarem nmap na jednotce pohotovostního autopilota (nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.105)

```
┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -sV -sC -O -T4 -p- -e br03 192.168.90.105
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 09:55 CEST
Nmap scan report for 192.168.90.105
Host is up (0.00056s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT      STATE  SERVICE            VERSION
22/tcp    closed ssh
8081/tcp  open   ssl/blackice-icecap?
| ssl-cert: Subject: commonName=1462554-01-H-B7S19109A00948-B/organizationName=Tesla/stateOrProvinceName=California/countryName=US
| Not valid before: 2021-02-05T10:01:57
|_Not valid after:  2023-02-05T10:01:57
8901/tcp  open   http               Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
25974/tcp closed unknown
28496/tcp open   http               Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
MAC Address: 00:43:58:85:4C:02 (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.1
OS details: Linux 5.1
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 109.36 seconds
```

4.1.2.2 Přemostění br12

Na tomto přemostění jednotka MCU zcela ignorovala IP adresu síťového mostu. MCU na tomto přemostění jednoduše neměla žádné otevřené porty, resp. běžící služby (4.8).

Obrázek 4.8: Sken softwarem nmap na jednotce MCU (*br12*) (nmap -Pn -n -sV -sC -O -T4 -p- -e br12 192.168.90.100)

```
┌──(root㉿ASUS-KALI)-[~]
└─# nmap -Pn -n -sV -sC -O -T4 -p- -e br12 192.168.90.100
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 11:10 CEST
Nmap scan report for 192.168.90.100
Host is up (0.00031s latency).
All 65535 scanned ports on 192.168.90.100 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)
MAC Address: A4:34:D9:01:02:03 (Intel Corporate)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1318.40 seconds
```

Následoval nmap sken (4.9) rádio tuneru.

Obrázek 4.9: Sken softwarem nmap na jednotce radio tuneru (nmap -Pn -n -sV -sC -O -T4 -p- -e br12 192.168.90.30)

```
┌──(root@ASUS-KALI)-[~]
└─# nmap -Pn -n -sV -sC -O -T4 -p- -e br12 192.168.90.30
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-26 11:05 CEST
Nmap scan report for 192.168.90.30
Host is up (0.00059s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Dropbear sshd 2015.68 (protocol 2.0)
1488/tcp  open  docstor?
| fingerprint-strings:
|   Help:
|_    ! 5f
3744/tcp  open  sasg?
| fingerprint-strings:
|   GenericLines:
|     HTTP/1.1 200 OK
|     Content-type: text/plain
|     Content-transfer-encoding: 7-bit
|     Content-length: 0
|     Connection: close
|   GetRequest, HTTPOptions:
|     HTTP/1.1 200 OK
|     Content-type: text/plain
|     Content-transfer-encoding: 7-bit
|     Content-length: 10696
|     Connection: close
|     servicebroker master
|     compile settings: tcp iface, tcp client, tcp timeo supervision, http iface, cache
enabled, ipc iface, tcp send/recv timeo=10000/10000 ms
|     version: 4.0.57
|     Statistics:
|     Servers: 29|29 , cached : 0
|     Clients: 22|22
|     Notifications : 88|93 , proxies: 0|0
|     Server Connect: 10|15
|     Server Disconnect: 34|34
|     Client Detach: 44|44
|     Jobs: 0|0
|     Connected Servers:
|     Process ServerId Interface
|     TunerApp <1000.500026> TunerPresCtrlAdapter.TunerStation
|     TunerApp <1000.500024> TunerPresCtrlAdapter.TunerEpg
|     TunerApp <1000.500025> TunerPresCtrlAdapter.TerrestrialTunerControl
|_    TunerApp <1000.500015> TunerPresCtrlAdapter.TerrestrialTunerAnn
5555/tcp  open  freeciv?
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 404 Not Found
|     Content-Length: 41
|     Date: Thu, 01 Jan 1970 02:21:41 GMT
|     {"valid":"false", "reason":"parse_error"}
|   GetRequest, HTTPOptions:
|     HTTP/1.0 404 Not Found
|     Content-Length: 41
|     Date: Thu, 01 Jan 1970 02:21:26 GMT
|     {"valid":"false", "reason":"parse_error"}
|   RTSPRequest:
|     HTTP/1.1 404 Not Found
|     Content-Length: 41
|     Date: Thu, 01 Jan 1970 02:21:26 GMT
|     {"valid":"false", "reason":"parse_error"}
|   SIPOptions:
|     HTTP/1.1 404 Not Found
|     Content-Length: 41
|     Date: Thu, 01 Jan 1970 02:21:46 GMT
|_    {"valid":"false", "reason":"parse_error"}
30508/tcp open  unknown
30509/tcp open  unknown
30510/tcp open  unknown
30511/tcp open  unknown
30512/tcp open  unknown
30513/tcp open  unknown
30520/tcp open  unknown
30530/tcp open  unknown
50361/tcp open  unknown
52196/tcp open  unknown
65218/tcp open  tandem-print Sharp printer tandem printing
65219/tcp open  unknown
MAC Address: 2C:6B:7D:C4:41:3F (Texas Instruments)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; Device: printer; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 208.26 seconds
```

Díky skenu zranitelností jsme získali přehled o běžících službách na jednotlivých zařízeních. Tento seznam služeb nám pomůže k další síťové analýze možných zranitelností připojených zařízení.

4.1.3 Experiment: Analýza zranitelností připojených zařízení

Hlavním cílem analýzy zranitelností je odhalit nedostatky v testované síti, které by mohl útočník zneužít. Hledání těchto chyb rozdělíme do dvou kategorií, pasivní testování a aktivní testování.

Nástroje, které potřebujeme pro úspěšnou analýzu zranitelností jsou prohlížeč Firefox, nástroj Burp Suite a příkazová řádka v Kali Linux [2.2].

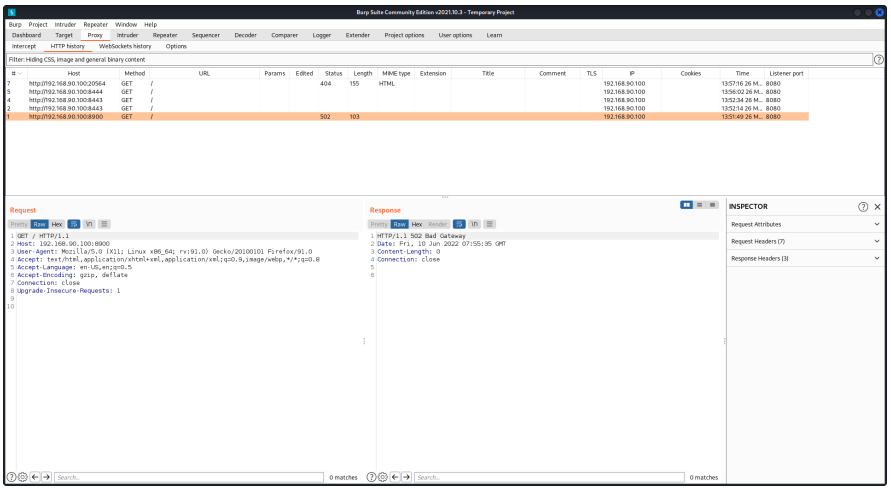
Z výsledků prvotní analýzy předpokládáme, že většina portů je otevřena pouze pro účely poslechu a pokud bude docházet k odpovědím ze služeb, které na portech běží, budou mít převážně informativní charakter.

4.1.3.1 Aktivní penetrační testování

Během skenování zranitelností na připojených zařízeních, jsme získali kompletní seznam běžících služeb na jednotlivých zařízeních. Většinu služeb nebylo možno během testování konkrétněji identifikovat.

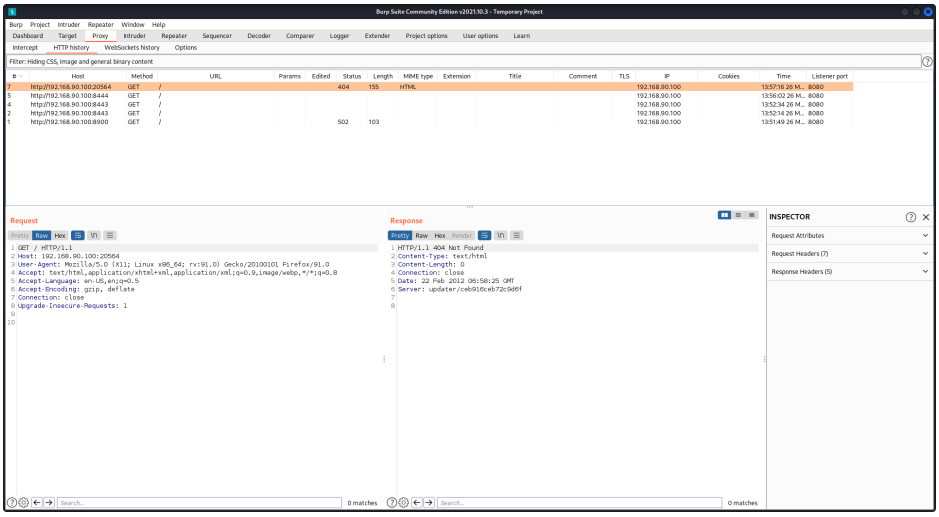
Jednotka MCU s IP adresou 192.168.90.100 má 4 otevřené TCP porty (4.5). Porty 8443 a 8444 nereagovali na žádné naše pokusy navázat spojení. Porty 8900 a 20564 reagovali na naše dotazy pouze jako „*Bad Gateway*“ (4.10) nebo „*Page Not Found*“ (4.11). Změna našeho dotazu nezměnila výslednou odpověď jednotky.

Obrázek 4.10: Burp Suite odpověď na portu 8900



Dvě jednotky autopilota, kde hlavní ACU je na IP adrese 192.168.90.103 a jednotka v pohotovostním režimu je na adrese 192.168.90.105, byly identicky otestovány se stejnými závěry testování.

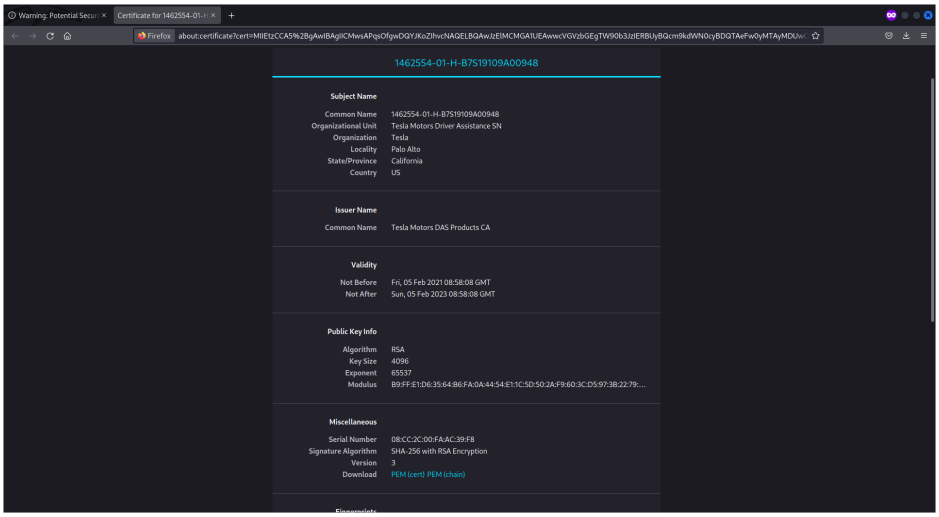
Obrázek 4.11: Burp Suite odpověď na portu 20564



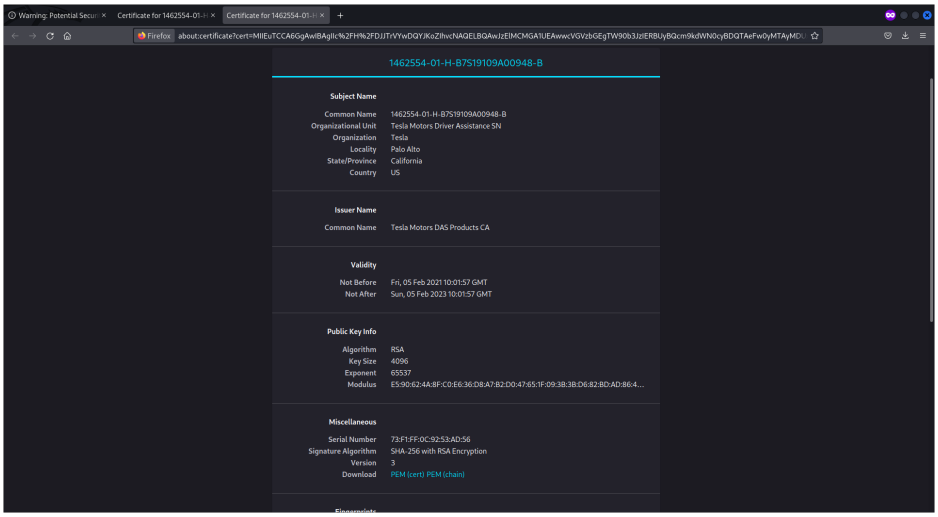
Porty 22 a 25974 byly cíleně zavřeny na námi testované verzi firmwaru automobilu. Porty 8901 a 28496 pouze odpovídali chybou „*Page Not Found*“, kde změna zasílaného dotazy chování nezměnila.

Na portu 8081 běží SSL certifikační služba (4.12), na které nebyla nalezena žádná zranitelnost. Jedná se o služby s informativním charakterem, které bude nejspíše mít proprietární účely v rámci společnosti Tesla. Jednotky mají rozdílné certifikáty pouze ve jménu certifikátu (pohotovostní jednotka ACU má certifikát zakončený na „-B“ (4.13)).

Obrázek 4.12: SSL certifikační služba na hlavní jednotce autopilota



Obrázek 4.13: SSL certifikační služba na pohotovostní jednotce autopilota

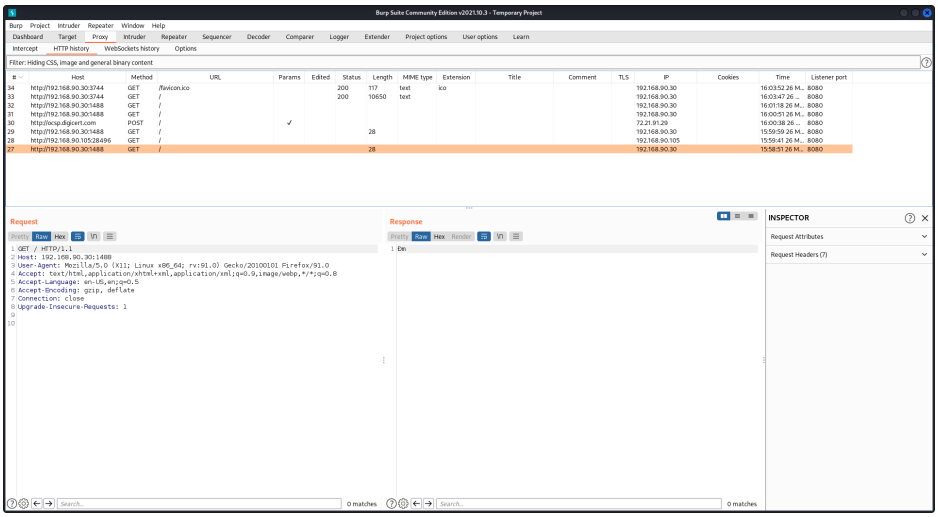


Rádio tuner na síťovém mostu *br12* s IP adresou 192.168.90.30 měl nejvíce otevřených portů z testovacích zařízení. Jejich funkcionalita je nejspíše proprietární a nebo má informativní charakter o nastavení rádia (naladěná frekvence, stupeň hlasitosti atd.).

Port 22 byl otevřen a běžela na něm SSH služba. SSH služba používá pouze certifikáty jako autentizaci při pokusu o připojení. Používání certifikátů jako autentizace pro připojení nám znemožňuje veškeré pokusy o připojení do rádia přes port 22.

Port 1488 vrací pouze hexadecimální výstup (4.14).

Obrázek 4.14: Ukázka odpovědi radio tuneru na zaslaný dotaz na port 1488



Služba běžící na portu 3744 odpovídala na zasílané dotazy. Odpověď obsahuje pouze nastavení rádio tuneru (4.15). Tyto hodnoty nastavení nebylo možné změnit ani zneužít.

Obrázek 4.15: Ukázka odpovědi služby radio tuneru běžící na portu 3744

```
HTTP/1.1 200 OK
Content-type: text/plain
Content-transfer-encoding: 7-bit
Content-length: 10529
Connection: close

servicebroker master
compile settings: tcp iface, tcp client, tcp timeo supervision, http iface, cache enabled, ipc iface, tcp send/recv timeo=10000/10000 ms
version: 4.0.57

Statistics:
   Servers: 22|22 , cached : 0
   Clients: 21|21
   Notifications : 94|108 , proxies: 0|0
      Server Connect: 12|26
      Server Disconnect: 40|40
      Client Detach: 42|42
   Jobs: 0|0

Connected Servers:
     pid  Process          ServerId      Interface
     835  TunerApp         <1000.500022>  TunerPresCtrlAdapter.TunerStation 1.3
     835  TunerApp         <1000.500019>  TunerPresCtrlAdapter.TunerEpg 2.0
     835  TunerApp         <1000.500018>  TunerPresCtrlAdapter.TerrestrialTunerControl 0.2
     835  TunerApp         <1000.500020>  TunerPresCtrlAdapter.TerrestrialTunerAnnouncement 0.4
     778  sysetcomposite   <1000.500009>  SysetPageUpdater.DSYSSysetClientGet 1.0
     778  sysetcomposite   <1000.500008>  SysetIOSrv.DSYSSysetIOSrvAdmin 1.1
     778  sysetcomposite   <1000.500012>  SysetDispatcher.DSYSSysetClientStore 2.0
     778  sysetcomposite   <1000.500010>  SysetDispatcher.DSYSSysetAdmin 3.0
     778  sysetcomposite   <1000.500011>  SysetDispatcher.DSYSOnOff 1.1
     778  sysetcomposite   <1000.500013>  SysetDispatcher.DSYSErrorMemoryAppClient 1.1
     767  Persistency      <1000.500005>  Persistency.DSYSPersistencyAdmin 2.3
     767  Persistency      <1000.500004>  Persistency.DSYSPersistency 2.4
     767  Persistency      <1000.500006>  Persistency.DSYSOnOff 1.1
     767  Persistency      <1000.500007>  Persistency.DSYSErrorMemoryAppClient 1.1
     835  TunerApp         <1000.500014>  PTunerOnOffHandler.DSYSOnOff 1.1
     835  TunerApp         <1000.500021>  PTunerDiagnosisHandler.TunerIntrospection 4.3
     684  OnOff            <1000.500003>  POnOffComponent.OnOff 1.1
     682  swdlctrl         <1000.500001>  PHTTPServerComponent.Swdlctrl 0.2
     835  TunerApp         <1000.500017>  GAnnouncementMaster.DMMTunerAnnouncement 3.2
     681  Diagnosis        <1000.500002>  Diagnosis.Diagnosis 1.3
     835  TunerApp         <1000.500016>  AMFMTunerDSIDevice.DMMTunerStation 7.2
     835  TunerApp         <1000.500015>  AMFMTunerDSIDevice.DMMAmFmTunerControl 7.5

Connected Clients:
     pid  Process          ClientId      ServerId       Interface
     682  swdlctrl         <1000.100001> <1000.500002>  Diagnosis.Diagnosis 1.3
     681  Diagnosis        <1000.100002> <1000.500002>  Diagnosis.Diagnosis 1.3
     683  Engineering      <1000.100003> <1000.500002>  Diagnosis.Diagnosis 1.3
     683  Engineering      <1000.100004> <1000.500003>  POnOffComponent.OnOff 1.1
     684  OnOff            <1000.100005> <1000.500005>  Persistency.DSYSPersistencyAdmin 2.3
     681  Diagnosis        <1000.100006> <1000.500003>  POnOffComponent.OnOff 1.1
     681  Diagnosis        <1000.100007> <1000.500009>  SysetPageUpdater.DSYSSysetClientGet 1.0
     684  OnOff            <1000.100008> <1000.500011>  SysetDispatcher.DSYSOnOff 1.1
     681  Diagnosis        <1000.100009> <1000.500012>  SysetDispatcher.DSYSSysetClientStore 2.0
     835  TunerApp         <1000.100010> <1000.500009>  SysetPageUpdater.DSYSSysetClientGet 1.0
     835  TunerApp         <1000.100011> <1000.500004>  Persistency.DSYSPersistency 2.4
     835  TunerApp         <1000.100012> <1000.500004>  Persistency.DSYSPersistency 2.4
     836  TunerPresCtrl    <1000.100013> <1000.500018>  TunerPresCtrlAdapter.TerrestrialTunerControl 0.2
     836  TunerPresCtrl    <1000.100014> <1000.500019>  TunerPresCtrlAdapter.TunerEpg 2.0
     683  Engineering      <1000.100015> <1000.500021>  PTunerDiagnosisHandler.TunerIntrospection 4.3
     681  Diagnosis        <1000.100016> <1000.500021>  PTunerDiagnosisHandler.TunerIntrospection 4.3
     836  TunerPresCtrl    <1000.100017> <1000.500020>  TunerPresCtrlAdapter.TerrestrialTunerAnnouncement 0.4
     836  TunerPresCtrl    <1000.100018> <1000.500021>  PTunerDiagnosisHandler.TunerIntrospection 4.3
     683  Engineering      <1000.100019> <1000.500022>  TunerPresCtrlAdapter.TunerStation 1.3
     684  OnOff            <1000.100020> <1000.500022>  TunerPresCtrlAdapter.TunerStation 1.3
     836  TunerPresCtrl    <1000.100021> <1000.500022>  TunerPresCtrlAdapter.TunerStation 1.3

Server Connect Notifications:
     pid  Process          Notification
     835  TunerApp         -76- SDARSTunerDSIDevice.DMMTunerAntenna 2.0
     835  TunerApp         -75- SDARSTunerDSIDevice.DMMSdarsTunerControl 0.13
     836  TunerPresCtrl    -70- SdarsPresCtrlAdapter.SdarsTunerControl 1.0
     836  TunerPresCtrl    -68- SdarsPresCtrlAdapter.TunerEpg 2.0
     835  TunerApp         -58- TunerAppTv.DSYSOnOff 1.1
     836  TunerPresCtrl    -53- SdarsPresCtrlAdapter.TunerStation 1.3
     778  sysetcomposite   -47- ErrorMemory.DSYSErrorMemoryApp 1.1
     684  OnOff            -30- SdarsPresCtrlAdapter.SdarsTunerControl 1.0
     767  Persistency      -24- ErrorMemory.DSYSErrorMemoryApp 1.1
     683  Engineering      -19- SdarsPresCtrlAdapter.TunerStation 1.1
     683  Engineering      -17- SdarsPresCtrlAdapter.SdarsTunerControl 1.0
     682  swdlctrl         -1- SwdlRoot.swdlCSB 2.1

Server Disconnect Notifications:
     pid  Process          Notification
     836  TunerPresCtrl    -102- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     836  TunerPresCtrl    -101- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     684  OnOff            -100- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     684  OnOff            -99- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     683  Engineering      -98- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     683  Engineering      -97- <1000.500022> TunerPresCtrlAdapter.TunerStation 1.3
     836  TunerPresCtrl    -94- <1000.500021> PTunerDiagnosisHandler.TunerIntrospection 4.3
     836  TunerPresCtrl    -93- <1000.500021> PTunerDiagnosisHandler.TunerIntrospection 4.3
     836  TunerPresCtrl    -90- <1000.500020> TunerPresCtrlAdapter.TerrestrialTunerAnnouncement 0.4
     836  TunerPresCtrl    -89- <1000.500020> TunerPresCtrlAdapter.TerrestrialTunerAnnouncement 0.4
     681  Diagnosis        -82- <1000.500021> PTunerDiagnosisHandler.TunerIntrospection 4.3
     681  Diagnosis        -81- <1000.500021> PTunerDiagnosisHandler.TunerIntrospection 4.3
```

Porty 30508, 30513, 65218 a 65219 neodpovídaly na žádné námi zaslané dotazy. Poslední dva porty (65218 a 65219) měli po každém restartu automobilu náhodné hodnoty. Tyto hodnoty se pohybovali v mezi 34000 až 65500. Nicméně jejich funkcionalita se tímto jevem neměnila.

4.1.3.2 Pasivní penetrační testování

Data pro pasivní testování jsme sesbírali při monitorování komunikačního provozu na síťových mostech *br03* a *br12*. Doba nahrávání komunikace byla 30 minut. Monitorování proběhlo při odemknutém vozu i při zamčeném. Žádný rozdíl v síťové komunikaci mezi zamčeným a odemčeným stavem nebyl zaznamenán.

Komunikace na přemostění *br03* mezi ACUs a MCU probíhala převážně z aktivního ACU (192.168.90.103) na jednotku MCU (192.168.90.100). Tato komunikace přenášela pakety ze zadní videokamery a dvou postranních videokamer. Tato komunikace je podrobně popsána v dalším experimentu [4.1.4]. Jiná komunikace na tomto síťovém mostu není z bezpečnostního hlediska tak podstatná. Jedná se o neustálou kontrolu připojení jednotky autopilota pomocí ARP protokolu zasílaných z jednotky MCU. Dále aktivní jednotka ACU občas pošle TCP SYN paket na MCU, jednotka MCU na tyto pakety nikdy neodpovídá.

Na síťovém mostě *br12* je více jak 99% komunikace odesláno z rádio tuneru (192.168.90.30) na multimediální jednotku (192.168.90.100). Tato komunikace je převážně pomocí protokolu IEEE 1722, který slouží jako ukončující formát poslechu rádia nebo se jedná o multicast komunikaci přes kterou rádio tuner odesílá informace o svém nastavení pro informativní účely. Stejně jako na přemostění *br03* jednotka MCU nedopovídá na příchozí pakety a pouze jednou za čas kontroluje stav připojení rádio tuneru pomocí ARP protokolu.

Na začátku testování se zdálo, že většina otevřených portů je otevřená pouze pro účely poslechu. Tento předpoklad se z většiny potvrdil během pasivní části testování. Není však vyloučeno, že tyto služby reagují/mění se v závislosti na to v jakém režimu se vůz nachází. Z webového zdroje [57] jsme se dozvěděli o různých režimech, které jsou proprietárním řešením společnosti Tesla a slouží k internímu testování. Kódy pro změnu režimu nejsou veřejné, proto jsme tuto teorii nemohli potvrdit.

Během tohoto experimentu jsme nalezli žádné bezpečnostní riziko.

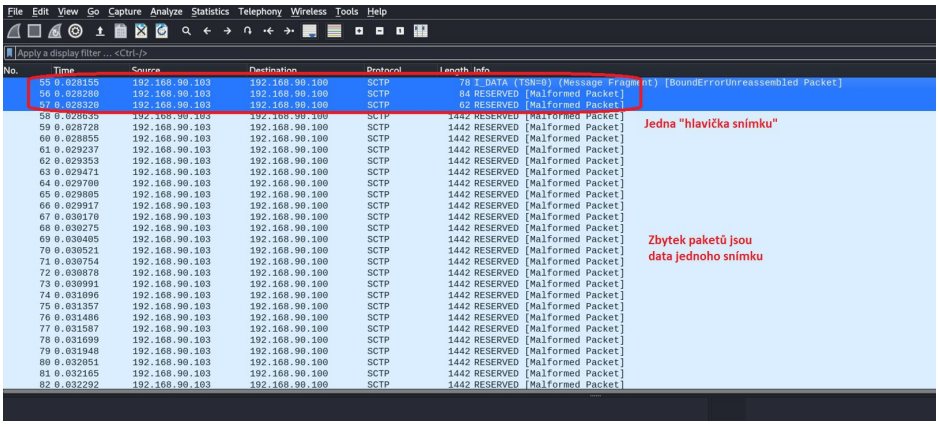
4.1.4 Experiment: Manipulace s videozáznamem zadních kamer

Cílem tohoto experimentuje je demonstrovat proveditelnost nalezené zranitelnosti. Podrobně popíšeme kroky k úspěšnému zneužití zranitelnosti, zhodnotíme nebezpečnost zranitelnosti za pomoci předem nadefinovaných hodnotících kritérií a navrhneme bezpečnostní opatření pro zamezení zneužití zranitelnosti.

Tcpreplay-edit a Wireshark jsou softwarové nástroje, které jsou pro demostraci nalezené zranitelnosti potřeba.

Jednotka ACU shromažďuje data (video snímky) ze zpětné kamery, resp. postranních kamer. Tyto snímky ve formě paketů jsou poté odeslány do jednotky MCU, kde se zpracují a přehrají na displeji. Jeden snímek se skládá ze tří paketů hlaviček (dále jen „hlavička snímku“) a více datových paketů obsahujících data jednoho snímku videa (dále jen „data snímku“) (4.16). Každá hlavička snímku je jiná (jiné označení paketů, ID paketů atd.), ale pouze první hlavička je kontrolována multimediální jednotkou. Všechny následující snímkové hlavičky neprochází kontrolou a mohou být vynechány a data snímků jsou stejně zpracována a zobrazena na displeji jako další snímky videa. Síťová komunikace probíhá pomocí protokolu STCP (*Stream Control Transmission Protocol*). S video snímky je možné manipulovat a celá videokomunikace tak se může být změněna nebo kompletně nahrazena. Komunikace může být změněna úpravou souborů PCAP (*Packer Capture*). Jedná se o soubory, které byly zachyceny pomocí nástroje Wireshark a obsahují nahraný síťový provoz mezi jednotkami ACU a jednotkou MCU.

Obrázek 4.16: Ukázka jednoho zachyceného video snímku softwarem Wireshark



4.1.4.1 Důkaz proveditelnosti

Pro demonstraci zranitelnosti jsme zachycenou síťovou komunikaci mezi jednotkami ACU a MCU uložili pomocí nástroje Wireshark do souboru ve formátu PCAP. Zkontrolovali jsme, že se nahrála i prvotní hlavička snímku, bez které by videozáznam nebyl multimediální jednotkou po zpracování zobrazen.

Nahraný PCAP soubor jsme následně pomocí nástroje tcpreplay-edit nahráli na jednotku MCU. V rámci nástroje tcpreplay-edit jsme použili následující nastavení:

- *-K*: načtení souboru PCAP do paměti RAM pro maximální možnou odesílací rychlosti souboru do cílené sítě
- *-l*: počet kolikrát bude PCAP soubor odeslán do cílené sítě
- *-i*: specifikace síťového rozhraní kam se bude PCAP soubor posílat

Odeslaný příkaz vypadal v našem případě následovně:
„tcpreplay-edit -K -l 1 -i eth3 /root/walk.pcap“

Výsledek odeslání souboru na síťové rozhraní jednotky MCU je okamžitý. Obraz začne přeskakovat a živý záznam zadní videokamery se stává nečitelným, kvůli překryvům se zaslaným videozáznamem útočníka.

Nahrávka ukázky zneužití zranitelnosti je v dostupná v příloze. Na videonahrávce je důkaz proveditelnosti zneužití zranitelnosti provedena za pomocí SSH připojení z mobilního telefonu. Toto bezdrátové provedení demonstruje možné zneužití zranitelnosti v reálném světě. Útočník, který má krátkodobý neomezený přístup k odemčenému automobilu se může během pár minut připojit na interní síť za pomoci Raspberry PI zařízení. Zařízení Raspberry PI umožňuje krátkodobé napájení z baterie a pomocí USB portu a FC602 USB zařízení se dokáže jednoduše připojit k interní síti vozu bez potřeby kabelového napájení. Malá velikost Raspberry PI umožňuje jeho umístění tak, aby nebylo uživatelem automobilu možné ho detekovat. Na předem připravené Raspberry PI se můžeme pomocí Wi-Fi sitě, kterou bude Raspberry PI vysílat, přihlásit pomocí SSH služby a pokud budeme v dostatečné vzdálenosti od automobilu, můžeme takto na dálku spustit námi předem připravený PCAP soubor.

Pokud by námi zvolené Raspberry PI zařízení umožňovalo LTE konektivitu, mohli bychom SSH službu na Raspberry PI vystavit do internetu a poté můžeme zadat příkaz o odeslání PCAP souboru kdekoliv, kde je internetové připojení.

4.1.4.2 Možné modifikace PCAP souboru

Při pokusech odeslat PCAP soubor pouze s daty snímků (bez hlaviček snímků), jednotka MCU videozáznam nezobrazila na displej. Poté jsme zkusili dát na začátek souboru jednu hlavičku snímku (tedy tři hlavičkové pakety videozáznamu) a zbytek souboru, který obsahoval pouze data snímků byl ponechán beze změny. MCU soubor zpracovala a odeslala všechny snímky do displej. Což vyvolalo otázku, co přesně MCU kontroluje na příchozích snímcích videa?

Kodek, který používá zadní kamera je chováním nejblíže ke kódování h.264. Ověření tohoto předpokladu bylo neúspěšné, neboť protokol STCP v tomto ohledu neuchovává dostatečné informace o originálním záznamu z videokamery. Nezapomínejme, že videozáznam je nejdříve zpracován jednotkou ACU a až poté odeslán na jednotku MCU.

Poté jsme se zaměřili na samotný protokol SCTP. Zkoušeli jsme posílat „rozbité“ pakety (nedokončené TAGy paketů, špatný údaj o počtu datových paketů, upravený kontrolní součet hlavičky atd.) v nekonečných smyčkách s cílem provést útok typu DoS (*Denial-of-Service*). Pokud bylo něco upraveno nebo změněno (kromě TAG označení paketu), video pakety byly zpracovány a odeslány na displej bez úspěšného provedení DoS útoku.

Poslední věc, kterou jsme zjistili, bylo, že hlavička snímku nemusí odpovídat jeho vlastním datům snímku. Při použití hlavičky snímku z jiného PCAP souboru se výsledné zobrazení videozáznamu nelišilo od originálního souboru se správnými hlavičkami snímků.

4.1.4.3 Normy pro videozáznam zadních kamer

Tesla Model 3 je v aktuální chvíli v rozporu s normami pro zadní videozáznam, které má splňovat.

Zadní obraz je definován jako vizuální obraz oblasti přímo za vozidlem, detekovaný pomocí jediného zdroje, který je poskytován na jednom konkrétním místě a umožňující nepřímý výhled sledovaného prostoru řidiči vozu. [58].

Dva hlavní standardy/normy se vztahují na automobil Tesla Model 3:

1. UN Regulation No. 158 - Devices for means of rear visibility or detection [59]

2. FMSVV 111 – 49 CFR § 571.111 - Standard No. 111; Rear visibility [58]

Obě normy vyžadují stejnou dobu odezvy maximálně 2 sekundy od okamžiku, kdy vůz začne couvat. Doba odezvy – Pohled dozadu splňující požadavky S5.5.1 a S5.5.2, je-li testován v souladu s S14.2, se musí zobrazit do 2,0 sekund od začátku akce couvání [58].

Což znamená, že pokud je zadní kamera vozu odpojena a vozidlo začne couvat, měl by být řidič alespoň informován o snížených jízdních vlastnostech, a nikoliv pouze zobrazit černou obrazovku na displeji namísto video obrazu jak je tomu v současnosti. Tesla Model 3 žádnou takovou informaci neuvádí, pokud není žádný signál ze zadní kamery. A proto je Tesla Model v rozporu s těmito normami.

4.1.4.4 Hodnocení zranitelnosti a její nebezpečí

Zranitelnost byla pomocí kalkulátoru CVSS v3.1 ohodnocena se skórem 5,6 (vektor hodnocení: AV:P/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:H).

Jedná se o zranitelnost se středním ohrožením na lidkých životech, lidský život je tedy nepřímo v ohrožení. V tomto případě by útočník mohl poskytnout jednotce MCU upravené videosnímky a řidič, spoléhající se pouze na vizuální pomoc videokamer auta, by se mohl snadno zranit nebo způsobit někomu zranění při couvání vozu, způsobené následkem nečitelnosti záznamu zadních kamer.

Další útok by mohlo přerušit spojení od zadních kamer a řidič, který je aktivně využívá (nejen při couvání) by mohl zpanikařit a způsobit dopravní nehodu, která by mohla ohrozit lidské životy.

4.1.4.5 Navrhovaná opatření

Doporučujeme kontrolovat, zda jsou hlavičky snímků správné vůči následujícím datovým paketům. Zjistili jsme, že každá hlavička snímku je jedinečná, a tím pádem souvisí s datovými pakety jednoho snímku. Kvůli tomuto vztahu by nemělo být obtížné vytvořit tento navrhovaný kontrolní mechanismus. Poté, co je tento mechanismus vytvořen a implementován, můžeme jej také použít ke splnění požadavků výše uvedených norem. A tedy při registraci chybných video paketů, nebo registraci jejich výpadku od jednotek autopilota, upozornit řidiče vozu, že jsou snížené jízdní vlastnosti.

Pouze pro implementaci norem tento kontrolní mechanismus není potřeba. Například mohla by být provedena jednoduchá kontrola na multimediální jednotce, aby se zjistilo, zda nějaká data/hlavičky vůbec přicházejí od jednotek autopilota. Pokud by tomu tak nebylo, opět upozornit řidiče na snížené jízdní vlastnosti.

4.2 Shrnutí nalezených zranitelností

Bezpečnostní analýza ukázala, že Tesla Model 3 (firmware: v10.2 2021.40.6 eed31525bfea) je poměrně bezpečný vůz s výjimkou možnosti modifikace video paketů zadních kamer vozu.

Po fyzickém přístupu k síťovému spojení mezi jednotkami MCU a ACU je možné plně modifikovat pakety ze zadních kamer. Dostupnost těchto ka-

mer je tak v rukou útočníka a MCU nedetekuje žádné úpravy nebo problémy s dostupností kamer v jakékoliv podobě.

Jedná se o zranitelnost s CVSS v3.1 hodnocením 5,6 (vektor hodnocení: AV:P/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:H) a se středním ohrožením na lidských životech (lidské životy jsou nepřímo ohroženy činností útočníka).

Po nalezení a zdokumentování zranitelnosti bylo postupováno podle pokynů v [2.5]. Společnost Tesla prozkoumala a zhodnotila nalezenou zranitelnost a vyjádřila souhlas s jejím plným zveřejněním.

4.3 Rozdíly s předešlou bezpečnostní analýzou interní sítě vozu

Hlavním rozdílem této bakalářské práce s diplomovou prací kolegy Machaly [56] je rozdílný rozsah testování. Diplomová práce se zaměřuje na Wi-Fi síť, síťové připojení LTE a na síťovou komunikaci mezi multimediální jednotkou a jednotkou autopilota. Díky většímu rozsahu diplomové práce je její závěr více obecného charakteru. Oproti tomu tato práce má konkrétnější bezpečnostní analýzu zaměřenou pouze na testování síťového propojení jednotky MCU s jednotkami autopilota a rádio tunerem. Zúžení testovaného rozsahu je navíc odůvodněno výrazným nezměněním funkcionality Wi-Fi a LTE konektivity. Nejvíce aktualizací proběhlo v řešení interní komunikace vozu, a proto je testována pouze jeho interní síť.

Zásadní změnou jednotky MCU je změna stavu otevřenosti některých síťových portů. Pouze jeden z portů, které jsou otevřeny, je stejný v obou pracích, a to konkrétně port 20564. Většina portů je na multimediální jednotce po aktualizacích uzavřena, početně zbyla méně jak polovina otevřených portů. Tyto změny způsobují snížení *Aattack Surface* možných zranitelností. Dále se objevila vlastnost jednotky MCU, že odpovídá pouze na dotazy odeslané z adres autopilotů (192.168.90.103 nebo .105).

Také funkcionalita síťových portů u jednotek autopilotů se změnila. SSH port 22 je cíleně uzavřen a přibyl nový port 8081, na kterém běží SSL certifikační služba s certifikátem, který je vystavený pro každého autopilota zvlášť.

Kvůli změnám v oblasti síťových portů došlo také ke změně samotné komunikace mezi propojenými jednotkami. Kontrolní jednotky komunikují převážně na jiných portech a například byla změněna TTL (*Time-To-Live*) hodnota paketů na hodnotu 1, aby bylo zabráněno přemostění mezi multimediální jednotkou jednotkami autopilota. Tato hodnota však jde na některých síťových přemostěních změnit na hodnotu vyšší a komunikace tak bude probíhat, jako kdyby pakety přes žádný síťový most neprocházely.

Závěr

Cílem bakalářské práce bylo provedení bezpečnostní analýzy interní sítě vozu Tesla Model 3. Práce navazuje na diplomovou práci z roku 2020, od doby zveřejnění práce ovšem došlo k aktualizaci architektury sítě. Testovaná síť umožňuje vnitřní komunikaci řídících jednotek v rámci vozu. Hlavním úkolem práce bylo zjistit, k jakým síťovým změnám došlo a zda mají dopad na bezpečnostní hledisko automobilu.

Bezpečnostní analýza byla prováděna fyzicky na reálném automobilu, který je schopen jízdy. Testování bylo prováděno během přímého připojení do sítě vozu. Po připojení bylo zjištěno, které řídící jednotky komunikují po síti a jaká komunikace mezi nimi probíhá. Následně byly jednotlivé jednotky aktivně otestovány na známé zranitelnosti a došlo k pokusům o nalezení zero-day zranitelností. V průběhu odposlechu komunikace mezi jednotkami ACU a MCU byla zachycena aktivní videokomunikace mezi zadní videokamerou, resp. postranními videokamerami. Analýza této nahrané videokomunikace prokázala existenci zero-day zranitelnosti. Zranitelnost umožňuje útočníkovi, který měl předešlý neomezený přístup k vozu, modifikaci videozáznamu ze zadních kamer automobilu a tím nepřímo způsobit újmu na zdravý lidem ve vozu či lidem v jeho okolí.

Výsledek práce tedy prokázal nedostatečnou kontrolu videokomunikace mezi jednotkami ACU a MCU. Zranitelnost byla nahlášena společnosti Tesla, která po ověření a zhodnocení zranitelnosti souhlasila s jejím zveřejněním. Práce může být využita jako předloha pro budoucí bezpečnostní testování v automobilovém průmyslu. Navazující bezpečnostní analýza by se mohla zaměřit na další interní síť, kterou vůz využívá. Jedná se o CAN sběrnici, která, jakožto zastaralejší standard, který neumožňuje využívání šifrovacích mechanismů v interní komunikaci, představuje rozsáhlý prostor pro penetrační testování.

Bibliografie

1. IVERSRUD, Rob. *Connector and cable considerations when designing for can bus*. 2022. Dostupné také z: https://connectorsupplier.com/connector-and-cable-considerations-when-designing-for-can-bus/. [Cit. 06-11-2022].
2. MARY TAMAR TAN; BRIAN BAILEY; HAN LIN. *LIN Basics and Implementation of the MCC LIN Stack Library on 8-Bit PIC® Microcontrollers*. 2017. Tech. zpr. Dostupné také z: http://ww1.microchip.com/downloads/en/appnotes/00002059b.pdf. [Cit. 02-11-2022].
3. VOSS, Wilfried. *Controller Area Network (CAN Bus) - Bus Arbitration*. 2018. Dostupné také z: https://copperhilltech.com/blog/controller-area-network-can-bus-bus-arbitration/. [Cit. 02-11-2022].
4. CIA INTEREST GROUPS. *Classical Controller Area Network (CAN)*. 2022. Dostupné také z: https://www.can-cia.org/can-knowledge/can/classical-can/. [Cit. 03-11-2022].
5. ROBERT BOSCH GMBH. *CAN Specification Version 2.0*. 1991. Tech. zpr. Dostupné také z: http://esd.cs.ucr.edu/webres/can20.pdf. [Cit. 03-11-2022].
6. NATIONAL INSTRUMENTS CORP. *CAN Physical Layer Standards: High-Speed vs. Low-Speed/Fault-Tolerant CAN*. 2022. Dostupné také z: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LzHSAU. [Cit. 03-11-2022].
7. CIA INTEREST GROUPS. *History of CAN technology*. 2022. Dostupné také z: https://www.can-cia.org/can-knowledge/can/can-history/. [Cit. 03-11-2022].
8. CIA INTEREST GROUPS. *CAN FD - The basic idea*. 2022. Dostupné také z: https://www.can-cia.org/can-knowledge/can/can-fd/. [Cit. 03-11-2022].

9. CIA INTEREST GROUPS. *Controller Area Network Extra Long (CAN XL)*. 2022. Dostupné také z: https://www.can-cia.org/can-knowledge/can/can-xl/. [Cit. 03-11-2022].
10. NATIONAL INSTRUMENTS CORP. *FlexRay Automotive Communication Bus Overview*. 2021. Dostupné také z: https://www.ni.com/cs-cz/innovations/white-papers/06/flexray-automotive-communication-bus-overview.html. [Cit. 04-11-2022].
11. EMBITEL. *FlexRay Protocol and the Modern ECU Network Architecture: An Insider's Perspective*. 2020. Dostupné také z: https://www.embitel.com/blog/embedded-blog/flexray-protocol-and-the-modern-ecu-network-architecture. [Cit. 04-11-2022].
12. VOJÁČEK, Antonín. *Sběrnice a komunikace FlexRay nejen pro automobily*. 2007. Dostupné také z: https://automatizace.hw.cz/sbernice-komunikace-flexray-nejen-pro-automobily. [Cit. 04-11-2022].
13. VOJÁČEK, Antonín. *Příklad připojení uzlů (Node) v čisté topologii sběrnice (Bus) [obrázek]*. 2007. Dostupné také z: https://automatizace.hw.cz/system/files/images/admin/smallFlexRay_Protocol_1.gif. [Cit. 31-12-2022].
14. VOJÁČEK, Antonín. *Příklad připojení uzlů (Node) a oddělovačů (Star 1A a 2A) v čisté topologii typu hvězda (star) [obrázek]*. 2007. Dostupné také z: https://automatizace.hw.cz/system/files/images/admin/smallFlexRay_Protocol_2.gif. [Cit. 31-12-2022].
15. VOJÁČEK, Antonín. *Příklad typické hybridní topologie sběrnice a hvězda [obrázek]*. 2007. Dostupné také z: https://automatizace.hw.cz/system/files/images/admin/smallFlexRay_Protocol_5.gif. [Cit. 31-12-2022].
16. BMW AG. *byteflight components and specifications*. 2007. Dostupné také z: https://web.archive.org/web/20070628003409/http://www.byteflight.com/presentations/index.html. [Cit. 05-11-2022].
17. SHEPARD, Jeff. *How is the MOST bus optimized for the automotive industry?* 2022. Dostupné také z: https://www.microcontrollertips.com/how-is-the-most-bus-optimized-for-the-automotive-industry-faq/. [Cit. 05-11-2022].
18. GRZEMBA, Andreas. *MOST The Automotive Multimedia Network*. Elektronická verze. Franzis Verlag GmbH, 2011. ISBN 978-3-645-65061-8. Dostupné také z: http://www2.ciando.com/img/books/extract/3645250611_lp.pdf.
19. BROADCOM CORPORATION. *BroadR-Reach® Physical Layer Transceiver Specification For Automotive Applications*. 2014-05. Tech. zpr. Dostupné také z: https://www.ieee802.org/3/1TPCESG/public/BroadR_Reach_Automotive_Spec_V3.0.pdf. [Cit. 07-11-2022].

20. DEM MANUFACTURING. *EMC Filtering For industrial Applications*. 2019. Dostupné také z: https://www.dem-uk.com/roxburgh/news/emc_filtering_for_industrial_applications.asp. [Cit. 07-11-2022].

21. ABAYE, Ali. *BROADR-REACH® TECHNOLOGY: ENABLING ONE PAIR ETHERNET*. 2012. Dostupné také z: https://www.ethercat.org/2013/mobile_applications/files/04_EtherCAT_Mobile_App_Broadcom.pdf. [Cit. 07-11-2022].

22. MUSK, Elon. 2017. Dostupné také z: https://twitter.com/elonmusk/status/881751358407299072. [Cit. 31-12-2022].

23. A., Julija. *How Many Teslas Have Been Sold? 25+ Tesla Car Sales Statistics & Facts*. 2012. Dostupné také z: https://fortunly.com/statistics/tesla-car-sales-statistics/. [Cit. 08-11-2022].

24. MOSEMAN, Andrew. *All About the Tesla Model 3*. 2017. Dostupné také z: https://www.popularmechanics.com/cars/a12983/35000-tesla-model-iii-coming-in-2017/. [Cit. 08-11-2022].

25. SHAHAN, Zachary. *Tesla Model 3 Has Passed 1 Million Sales*. 2021. Dostupné také z: https://cleantechnica.com/2021/08/26/tesla-model-3-has-passed-1-million-sales/. [Cit. 08-11-2022].

26. NOTATESLAAPP. *not a tesla app*. 2022. Dostupné také z: https://www.notateslaapp.com/software-updates/. [Cit. 09-11-2022].

27. WOUTERS, Lennert. *Fast, Furious and Insecure: Passive Keyless Entry and Start in Modern Supercars*. 2018. Dostupné také z: https://www.esat.kuleuven.be/cosic/news/fast-furious-and-insecure-passive-keyless-entry-and-start-in-modern-supercars/. [Cit. 02-11-2022].

28. SEN NIE; LING LIU; YUEFENG DU. *FREE-FALL: HACKING TESLA FROM WIRELESS TO CAN BUS*. 2016. Tech. zpr. Dostupné také z: https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf. [cit. 12-11-2022].

29. PETR JIRÁSEK, LUDĚK NOVÁK A JOSEF POŽÁR. *Výkladový slovník kybernetické bezpečnosti*. Páté elektronická vydání. Přel. KAREL VAVRUŠKA A PETR JIRÁSEK. Centrum kybernetické bezpečnosti, z.ú. a Česká pobočka AFCEA, 2022. ISBN 978-80-908388-4-0. Dostupné také z: https://cybersecurity.cz/data/Slovnik_523el.pdf.

30. DOSTÁL, Jiří. *Introduction to Ethical Hacking & Penetration Testing*. 2020. Dostupné také z: https://courses.fit.cvut.cz/BI-EHA/media/lectures/01_Introduction.pdf. [Cit. 16-11-2022].

31. MIESSLER, Daniel. *Secrecy (Obscurity) is a Valid Security Layer*. 2021. Dostupné také z: https://danielmiessler.com/study/security-by-obscurity/. [Cit. 16-11-2022].

32. BUYTENHEK, Lennert. *brctl - ethernet bridge administration*. 2022. Dostupné také z: https://www.root.cz/man/8/brctl/. [Cit. 16-11-2022].

33. ALEXEY KUZNETSOV A HIDEAKI YOSHIFUJI. *arping - send ARP REQUEST to a neighbour host*. 2022. Dostupné také z: https://www.root.cz/man/8/arping/. [Cit. 16-11-2022].

34. HILLS, Roy. *arp-scan*. 2022. Dostupné také z: https://github.com/royhills/arp-scan. [Cit. 16-11-2022].

35. LYON, Gordon Fyodor. *nmap - Network exploration tool and security / port scanner*. 2022. Dostupné také z: https://www.root.cz/man/1/nmap/. [Cit. 16-11-2022].

36. WEAR, Sunny. *Web Security Testing with Burp Suite*. 2022. Dostupné také z: https://www.pluralsight.com/paths/web-security-testing-with-burp-suite. [Cit. 16-11-2022].

37. SYSDIG INC. *Wireshark*. 2022. Dostupné také z: https://www.wireshark.org/. [Cit. 17-11-2022].

38. TURNER, Aaron. *tcpbridge - Bridge network traffic across two interfaces*. 2022. Dostupné také z: https://linux.die.net/man/1/tcpbridge. [Cit. 17-11-2022].

39. TURNER, Aaron. *tcpreplay-edit - Replay network traffic stored in pcap files*. 2022. Dostupné také z: https://tcpreplay.appneta.com/wiki/tcpreplay-edit-man.html. [Cit. 17-11-2022].

40. FIBRECODE. *FC602 USB 100BASE-T1 Stick for Automotive Single Pair Ethernet (SPE)*. 2022. Dostupné také z: https://www.fibrecode.com/fc602-usb-oabr-broadr-reach-100base-t1-stick.html. [Cit. 16-11-2022].

41. THE OWASP FOUNDATION INC. *OWASP Testing Guide 4.0)release(*. 2022. Dostupné také z: https://owasp.org/www-pdf-archive/OTGv4.pdf. [Cit. 17-11-2022].

42. NIST U.S. DEPARTMENT OF COMMERCE. *About NIST*. 2022. Dostupné také z: https://www.nist.gov/about-nist. [Cit. 17-11-2022].

43. OPEN INFORMATION SYSTEMS SECURITY GROUPS. *Information System Security Assessment Framework (ISSAF)*. 2022. Dostupné také z: https://www.futurelearn.com/info/courses/ethical-hacking-an-introduction/0/steps/71521. [Cit. 17-11-2022].

44. ISECOM - INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES. *OSSTMM 3 The Open Source Security Testing Methodology Manual*. 2022. Dostupné také z: https://www.isecom.org/OSSTMM.3.pdf. [Cit. 17-11-2022].
45. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *PTES Technical Guidelines*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines. [Cit. 18-11-2022].
46. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Pre-engagement*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Pre-engagement. [Cit. 18-11-2022].
47. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Intelligence Gathering*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Intelligence_Gathering. [Cit. 18-11-2022].
48. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Threat Modeling*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Threat_Modeling. [Cit. 18-11-2022].
49. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Vulnerability Analysis*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Vulnerability_Analysis. [Cit. 18-11-2022].
50. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Exploitation*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Exploitation. [Cit. 18-11-2022].
51. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Post Exploitation*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Post_Exploitation. [Cit. 18-11-2022].
52. PTES GROUP OF INFORMATION SECURITY PRACTITIONERS. *Reporting*. 2022. Dostupné také z: http://www.pentest-standard.org/index.php/Reporting. [Cit. 18-11-2022].
53. TESLA INC. *Product Security*. 2022. Dostupné také z: https://www.tesla.com/legal/security. [Cit. 20-11-2022].
54. NIST - NATION INSTITUTE OF STANDARDS AND TECHNOLOGY. *Common Vulnerability Scoring System Calculator*. 2022. Dostupné také z: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. [Cit. 06-12-2022].
55. FIRST - FORUM OF INCIDENT RESPONSE AND SECURITY TEAMS. *Common Vulnerability Scoring System version 3.1: User Guide*. 2019. Dostupné také z: https://www.first.org/cvss/user-guide. [Cit. 06-12-2022].

56. MACHALA, Filip. *Analýza bezpečnosti vnitřní sítě vozu Tesla Model 3*. Praha, 2020. Dostupné také z: https://dspace.cvut.cz/handle/10467/87976. Diplomová práce. ČVUT, Fakulta informačních technologií. Vedoucí práce Jiří DOSTÁL.

57. RICE, Tristan. *Hacking my Tesla Model 3 - Software Modes*. 2020. Dostupné také z: https://fn.lc/post/tesla-model-3-modes/. [Cit. 10-12-2022].

58. STANDARD NO. 111; REAR VISIBILITY. *49 CFR § 571.111 - Standard No. 111; Rear visibility*. 2014. Dostupné také z: https://www.law.cornell.edu/cfr/text/49/571.111. [Cit. 11-12-2022].

59. DEVICES FOR MEANS OF REAR VISIBILITY OR DETECTION. *UN Regulation No. 158 - Devices for means of rear visibility or detection*. 2021. Dostupné také z: https://unece.org/transport/documents/2021/07/standards/un-regulation-no-158-devices-means-rear-visibility-or-0. [Cit. 11-12-2022].

Příloha **A**

Seznam použitých zkratek

ACU Řídící jednotka autopilota

MCU Multimediální řídící jednotka

ECU Elektronická řídící jednotka

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `attack-example.mp4` ..... ukázka útoku na zranitelnost
  - `latex-dir` ..... soubor s LaTeX zdrojovými kódy práce
- `text` ..... text práce
  - `thesis-Nerad-Lukas.pdf` ..... text práce ve formátu PDF