虚拟汽车技术白皮书

VIRTUAL VEHICLE TECHNOLOGY WHITE PAPER

PROFESSIONAL
RELIABLE
RESPONSIBLE

NSFOCUS

NSFOCUS

绿盟科技集团股份有限公司(以下简称“绿盟科技”),成立于2000年4月,总部位于北京。公司于2014年1月 29日在深圳证券交易所创业板上市,证券代码:300369。绿盟科技在国内设有50 余个分支机构,为政府、金融、运营商、能源、交通、科教文卫等行业用户与各类型企业用户,提供全线网络安全产品、全方位安全解决方案和体系化安全运营服务。公司在美国硅谷、日本东京、英国伦敦、新加坡及巴西圣保罗设立海外子公司和办事处,深入开展全球业务,打造全球网络安全行业的中国品牌。

版权声明:

目录

CONTENTS

01

引言

车联网应用作为物联网落地最典型的场景之一，近年来不断涌现汽车电子电气架构、汽车网络安全、自动驾驶与协同控制等相关技术的研究成果。然而，这些研究成果，大都诞生于投入高成本建设的专用实验室环境，如大型硬件在环系统、高保真仿真平台等，这在客观上构成了较高得资金与技术门槛。对资金较少的学校、企业、科研院所并不友好，难以有效复现或深入参与相关技术研究。同时，在面向百人以上规模得班级教学、集中培训或大型技术竞赛等场景时，此类高成本、高复杂度得实验条件也带来了突出的规模化实施困境。若能有效突破这一成本与资源瓶颈，降低关键技术的研究与教学实训门槛，将有力的推动车联网领域的人才培养、技术普及与产业创新，从而为行业整体发展注入持续动力。

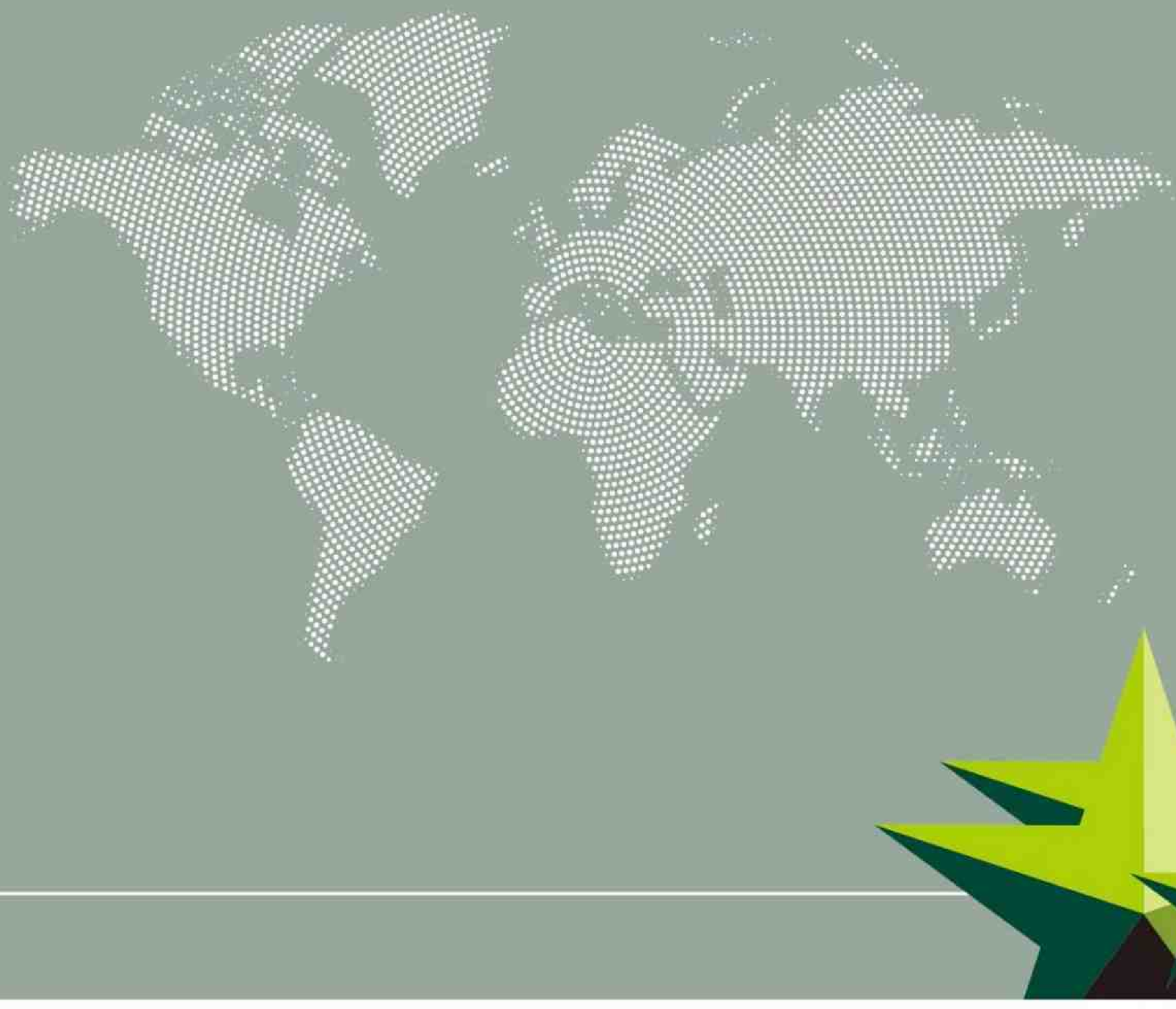
绿盟科技选择的方式是将硬件产品形态的汽车软件化、虚拟化。想要完成对汽车的高度仿真，尤其是汽车网络的仿真，对比多年实车研究成果后，我们对自己的研究提出了以下五点要求：

- 对汽车零部件的操作系统进行虚拟化，支持 Linux、Android 嵌入式操作系统的仿真。
- 操作系统运行在 ARM、X86 等多种指令集下，与实车的指令集使用情况相符。
- 在零部件之间建立无感、自然的 CAN 总线、车载以太网总线互联。
- 能够依照实车电子电气架构，复现某一具体车型的网络架构环境。
- 可依托成熟技术，如 Unity 3D、V2X、自动驾驶算法等构建单车驾驶、自动驾驶、多车协同的驾驶场景，覆盖从指令集到大规模交通仿真的能力。

满足这五项要求的重点在于对汽车电子电气架构的仿真，在后续的章节中将重点介绍。在第二章，将介绍汽车相关技术以做铺垫，如果读者对汽车电子系统较为了解，则可越过；第三章，将阐述电子电气架构的虚拟化路线，分 CAN 网络架构的虚拟化和零部件异构与操作系统级别的虚拟化形态这两方面来介绍；第四章，将设计虚拟汽车攻防系统，以仿真为底，靶标为入口，入侵检测机制为管控，教学、竞赛和电子电气架构业务为平台进行整合，设计出基于虚拟化的汽车攻防系统；第五章，将对我们构建的虚拟汽车系统进行实战测试，用局域网控车漏洞利用、模拟实车应用和产品研发测试这三个场景来评价虚拟汽车系统的实战表现。第六章，将阐述两个案例，来体现虚拟汽车在规模化运行以及虚实结合方面的优势。

02

汽车相关技术简介



对汽车的虚拟化，本质上是对电子电气架构的虚拟化，这其中涉及四方面的技术，分别为电子电气架构、汽车总线、虚拟化软件以及车内操作系统。

2.1 电子电气架构

电子电气架构（Electrical/Electronic Architecture），简单来说，即汽车内各个控制器的部署、连接、通信的一种架构形态。在实现其虚拟化仿真的过程中，可以将其分为零部件节点的仿真和汽车总线网络的仿真两大部分。具体形态，可以参考实车，如图 2.1 所示的结构。

图 2.1 汽车 CAN 网络域架构示例

至于在产品中的呈现形式，由于电子电气架构的多样性，需要结合具体需求对其架构进行定制后实现。以上述电子电气架构为例，需要对各个零部件节点位置进行拖拽配置、上线提醒、车内网信息展示、域网络结构等效果，如图 2.2 所示。

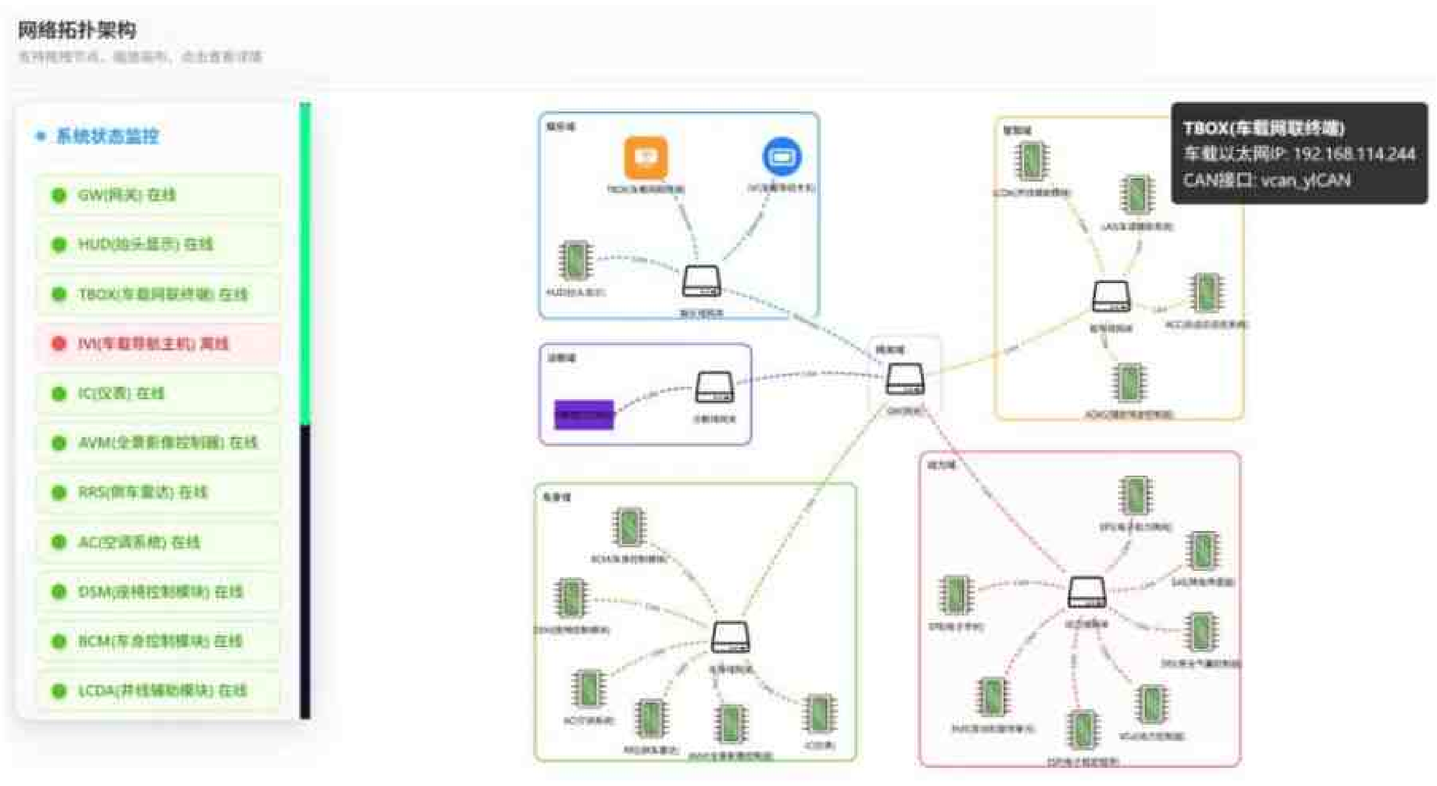


图 2.2 建议实现的车联网拓扑效果

2.2 汽车总线

车内总线有 LIN、CAN、FlexRay、MOST、以太网总线这几类，FlexRay、MOST 这两类总线，由于其成本高昂，和伴随汽车以太网总线技术的不断成熟，终将淡出市场。CAN、LIN、汽车以太网总线将逐渐成为市场主流。

对汽车总线的虚拟化，至少需要将 CAN 网络、车内以太网进行虚拟化。车内以太网在应用层和传统以太网几乎没有区别，都是基于 IP 报文进行通信，所以，针对以太网的虚拟化，依托虚拟化软件的 NAT、桥接等功能支撑虚拟零部件的以太网通信即可。需要解决的是 CAN 网络的虚拟化，以及各个零部件基于虚拟 CAN 总线网络的互联难题。

2.3 虚拟化软件

汽车内的控制器具备异构性，也就是说，车内有 ARM、X86、MIPS、Power PC 等多个处理器架构，而能够满足虚拟多种处理器架构的软件只有 QEMU，[①]支持的处理器架构情况如表 2.1 所示。

表 2.1 QEMU 支持的处理器架构

CPU 架构	虚拟化技术
i386 / x86_64	Haswell、Broadwell、Skylake-Server、EPYC 等
arm	ARM926, ARM1176, Cortex-A7/A8/A9/A15, Cortex-M3/M4
aarch64	Cortex-A53/A57/A72/A76 等
mips / mipsel / mips64	MIPS32/MIPS64、部分 Loongson
ppc / ppc64	e500、G3/G4/G5、POWER7/8/9（部分）
riscv32 / riscv64	sifive-e、sifive-u、virt
s390x	IBM z10/z13 等
sparc	LEON3 等
sparc64	UltraSPARC
m68k	Motorola 68000 系列
avr	ATmega 系列
xtensa	Tensilica Xtensa（如早期 ESP 系列）

① QEMU，https://www.qemu.org/

rx	Renesas RX 系列

谷歌基于 QEMU，开发了 emulator[①]软件，可以运行虚拟的 Android 镜像，如编译完成 Android 镜像后，可以运行“emulator”启动 Android 虚拟机；也可以在 Android Studio 中选择预编译的 Android 镜像，启动 Android 虚拟机。

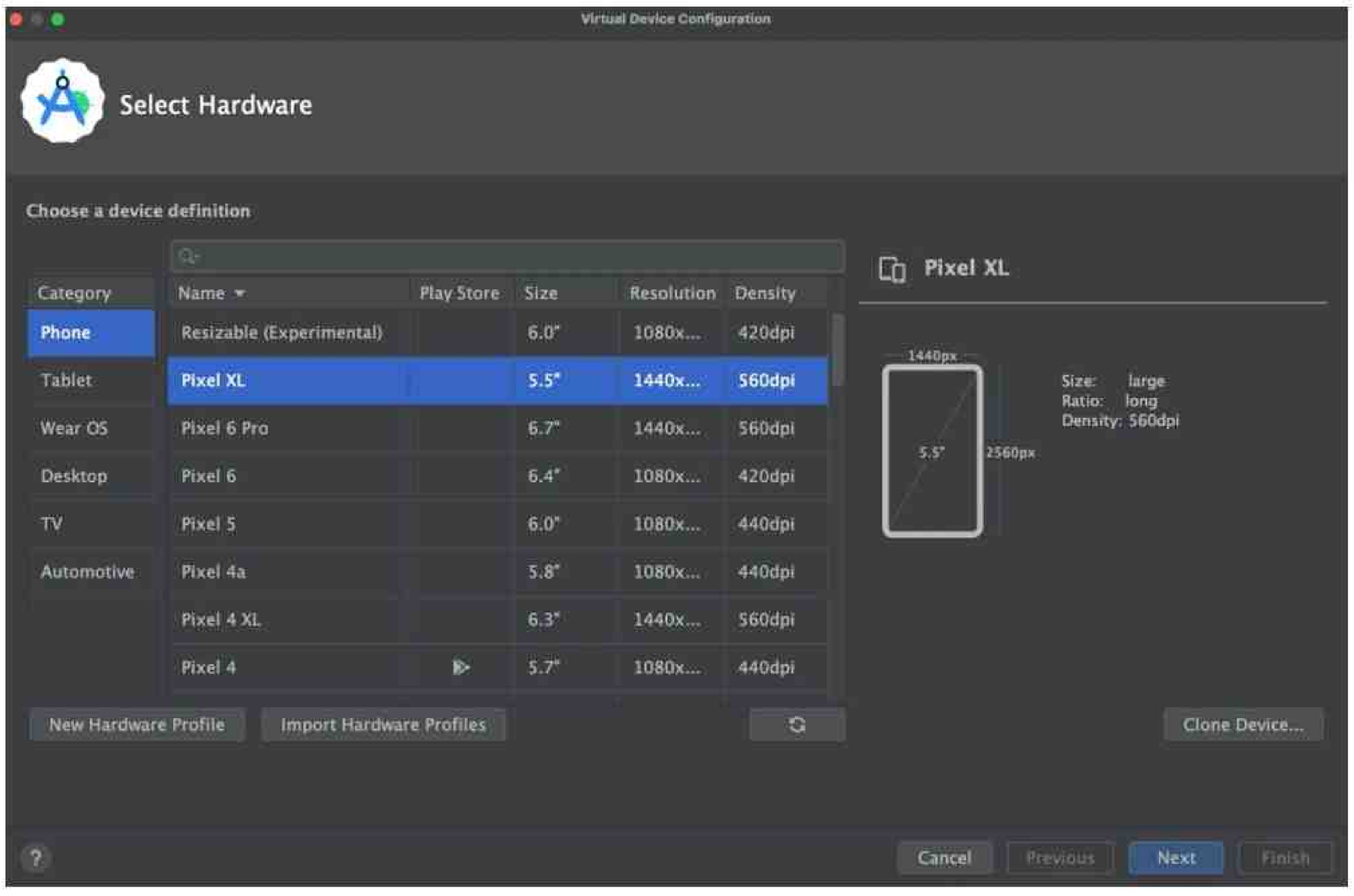


图 2.3　谷歌官方对 Android 模拟器的支持情况

综上所述，可以使用 QEMU 虚拟常规的 Linux 操作系统的零部件，使用 Emulator 来虚拟 Android 多媒体系统。

2.4　操作系统

在车内，操作系统有 Android、Linux、AGL[②]、QNX、RTOS 等，前三个操作系统是开源的，也都有虚拟化的版本，启动一个简单的虚拟主机相对容易。QNX 相对封闭，RTOS 的逻辑简单，使得对后两种操作系统虚拟化的必要性较小。

在车内，使用 Linux 操作系统的零部件有 T-BOX、以太网网关、自动驾驶控制器、行车记录仪，部分车机系统也会使用 Linux 操作系统，如特斯拉使用的是 Ubuntu；使用 Android 操作系统的零部件是车机系统；使用 AGL 操作系统的零部件为车机系统、仪表、T-BOX 等。Android 相比 AGL 的娱乐性更为

① Emulator，https://developer.android.com/studio/run/emulator-commandline?hl=zh-cn

② AGL，https://www.automotivelinux.org/

丰富，拓展性更强。所以，根据当前汽车应用操作系统的现状，以及操作系统本身功能的定位，将嵌入式 Linux、Android 操作系统进行虚拟化即可满足构建虚拟电子电气架构的需求。

2.5 小结

虚拟一辆汽车，最为关键的，是电子电气架构的虚拟化，即对零部件进行指令集、操作系统及 CAN 总线通信能力的仿真，对车内网通信进行 CAN 总线以及车内以太网的通信互联。只有将电子电气架构做到完整的仿真，汽车的仿真才不至于沦为空壳。

03

电子电气架构的虚拟化

本章通过构建虚拟车内网和虚拟零部件来虚拟汽车，以支撑远程控车业务相关的软件在 TBOX、车身控制器等零部件中运行，和车内操作系统漏洞、软件系统漏洞的靶标集成。

3.1 汽车 CAN 总线的虚拟化

汽车 CAN 总线的虚拟化相对容易，尤其是基于 SocketCAN[①]技术的 CAN 网络虚拟化。假设运行虚拟零部件的宿主机为网关，以创建四个功能域为目标，实现起来如图 3.1 所示。在 Linux 系统中，创建 SocketCAN 接口也较为简单。在启动脚本中，加载 vcan 虚拟 CAN 驱动后，使用“ip”命令创建并激活 CAN 接口即可。如下图所示，使用“ip”命令创建了四个 SocketCAN 接口。

```
iovns@iovns:~/Desktop$ cat /etc/rc.local
#!/bin/bash
echo "1" > /home/iovns/Desktop/startup.txt
sh /home/iovns/Desktop/config.sh
iovns@iovns:~/Desktop$ cat /home/iovns/Desktop/config.sh
sudo modprobe vcan
sudo ip link add dev can0 type vcan
sudo ip link set can0 up
sudo ip link add dev can1 type vcan
sudo ip link set can1 up
sudo ip link add dev can2 type vcan
sudo ip link set can2 up
sudo ip link add dev can3 type vcan
sudo ip link set can3 up
```

图 3.1　创建四个 SocketCAN 接口

图 3.2 所示，我们成功创建了四个 SocketCAN 接口，每个接口可以定义为一个功能域，如可以定义“can0”为信息域，“can1”为 ADAS 域，“can2”为动力域，“can3”为车身域。将多媒体系统、T-BOX 接入到代表信息域的“can0”中，将 ADAS 控制器接入到代表辅助驾驶域的“can1”中，将整车控制器 VCU、方向盘、脚踏板等接入到代表动力域的“can2”中，将车门、车灯等虚拟化控制器接入到代表车身域的“can3”中即可构建小型的电子电气架构。

① SocketCAN，https://en.wikipedia.org/wiki/SocketCAN

```
iovns@iovns:~/Desktop$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:d6:f8:5e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.91.141/24 brd 192.168.91.255 scope global dynamic noprefixroute ens33
       valid_lft 1035sec preferred_lft 1035sec
    inet6 fe80::20c:29ff:fed6:f85e/64 scope link
       valid_lft forever preferred_lft forever
3: can0: <NOARP,UP,LOWER_UP> mtu 72 qdisc noqueue state UNKNOWN group default qlen 1000
    link/can
4: can1: <NOARP,UP,LOWER_UP> mtu 72 qdisc noqueue state UNKNOWN group default qlen 1000
    link/can
5: can2: <NOARP,UP,LOWER_UP> mtu 72 qdisc noqueue state UNKNOWN group default qlen 1000
    link/can
6: can3: <NOARP,UP,LOWER_UP> mtu 72 qdisc noqueue state UNKNOWN group default qlen 1000
    link/can
iovns@iovns:~/Desktop$
```

图 3.2　创建的四个接口详细情况

当然，汽车总线不止有 CAN 总线，还有以太网总线。使用虚拟化软件对以太网的桥接、NAT 的方法，将虚拟零部件接入以太网即可满足零部件车载以太网的通信。本文不再赘述。

3.2　零部件指令集异构、操作系统级别的虚拟化

针对电子电气架构的虚拟化，需要做的是创建多个零部件，并将其按照电子电气架构的特点进行连接。这个“特点”可以是仿照实车的架构，也可以是创造一个新的架构。下面我们创造一个最小架构。

以往，T-BOX 和车机都会连接在信息域，站在网关的角度看，他们都在统一组 CAN 接口上相连。而在我们的架构中，为了最简单地描述电子电气架构的可定制性，我们将车机和 T-BOX 分开，放在两个不同的功能域。自动驾驶控制器连接到驾驶域中，这一点保持不变，具体如图 3.3 所示。

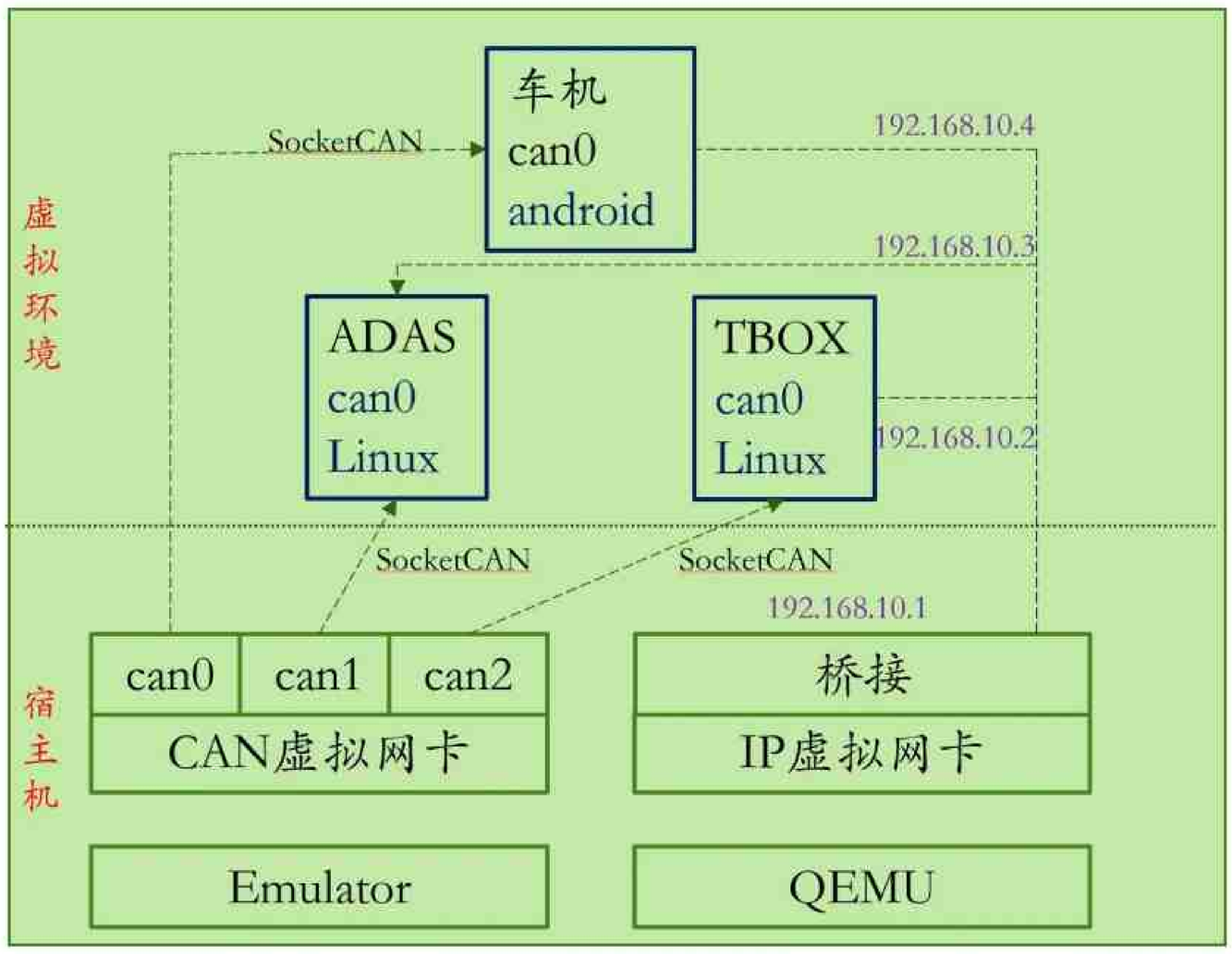


图 3.3　不同域零部件之间交互逻辑

这三个零部件之间可以通过 CAN 总线互通。如图 3.4 所示实验显示，在 TBOX 发送 CAN 总线消息，在 Android、网关上都可以接收到同样的 CAN 报文，证明虚拟零部件与宿主机和其他零部件的 CAN 总线是互通的。

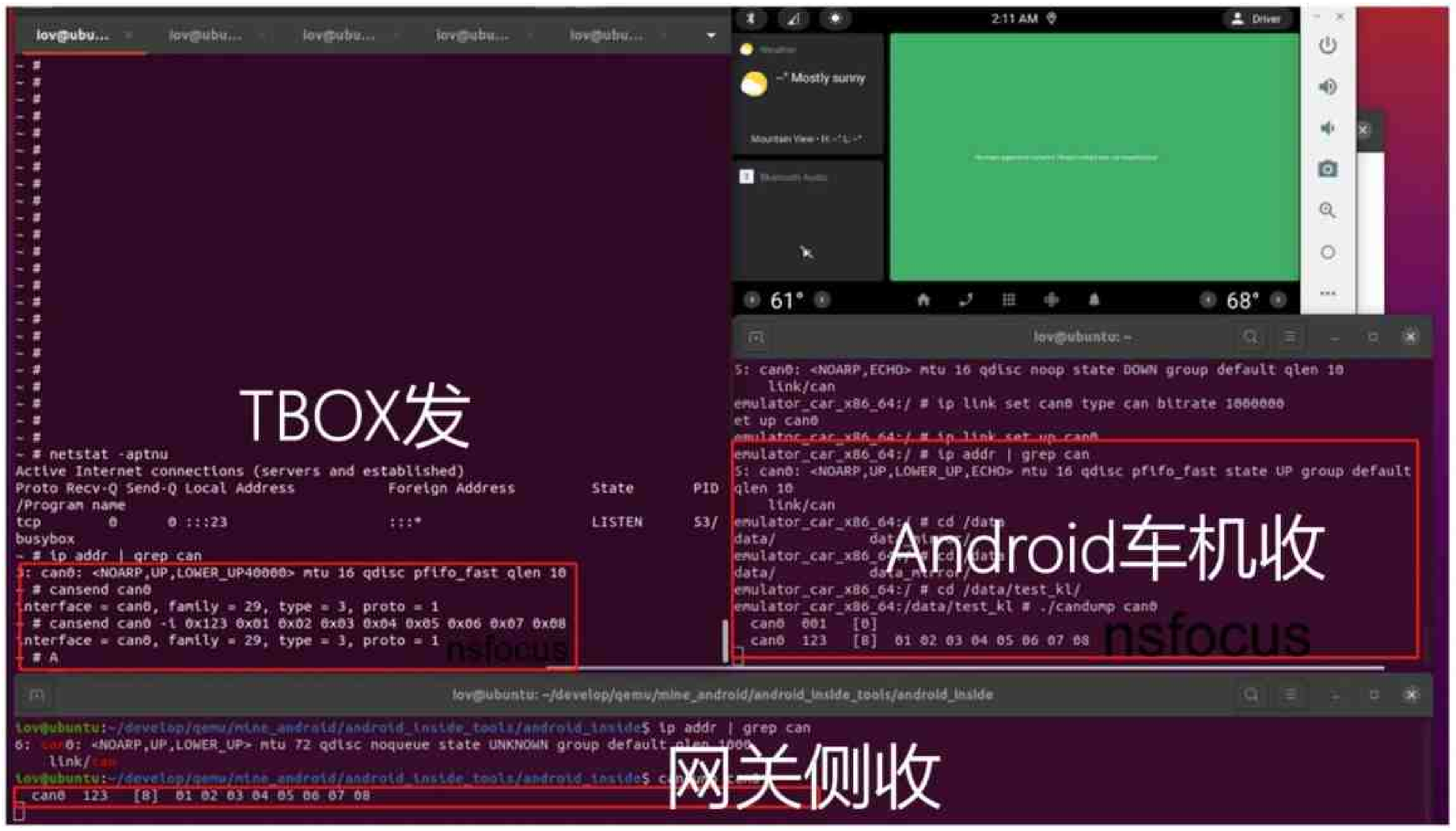


图 3.4 三个零部件互相收发

3.3 小结

本章通过介绍汽车 CAN 网络的虚拟化、Linux 和 Android 零部件的虚拟化以及虚拟汽车系统的构建，介绍了整个虚拟汽车电子电气架构的设计和实现思路。并就零部件的虚拟化实现进行了详细的阐述。

04

虚拟汽车攻防系统设计

本章就虚拟汽车构建和攻防系统构建这两方面进行阐述。虚拟汽车的构建过程保证了远程控制业务的完整运行，攻防系统的构建保证了全车安全状态可控的上帝视角。

4.1 虚拟汽车构建（单车与远控业务）

只有零部件是不够的，既然是汽车，就应该可以被驾驶。所以还需要有方向盘以控制汽车驾驶，需要有 3D 的界面以切换驾驶员视角。所以，汽车的运行环境需要有 3D 车路的呈现，运行虚拟汽车的主机必须配备性能足够强的显卡，以满足车路效果的稳定呈现。另外，还需要具备方向盘和脚踏板这类外设，可以是实车移植，也可以购买商业产品来进行虚实结合。主机需要读取方向盘的按键编码数据，将控制类信号转发到 CAN 总线后，经由虚拟零部件，转化为前端汽车的控制数据，这一转化，经由“控制引擎”完成汽车的运动控制和车灯、车门等 ECU 的控制。对于信息安全而言，需要向汽车零部件中植入安全探针，如安全 SDK、IDPS 等，以满足对汽车信息安全的实时监控。

车外，有云服务和手机 APP，所以需要具有远程控制服务、手机 APP 这两个模块，以满足汽车的远程控制功能。另外，还需要具备安全 SDK 和安全运营平台以满足监管需求。具体如图 4.1 所示。

图 4.1 虚拟汽车模块化构建思路

4.2 构建车联网攻防系统（车联网、攻击检测）

将自动驾驶控制器、车路协同模块、智能座舱等关键组件仿真后，可构建现代化的智能网联虚拟汽车，结合远程控车业务、TBOX 管理业务、汽车租赁等业务环境，构造汽车信息安全相关靶标，构成攻防场景下的虚拟汽车，其高并发特性，可满足大规模的仿真应用场景，如教学、竞赛等，其电子电气架构的定制特性可仿真多个型号的汽车架构。结合入侵检测、主动防护、威胁溯源等技术，可实现对车联网攻防系统的构建。具体可参考图 4.2 所示结构。

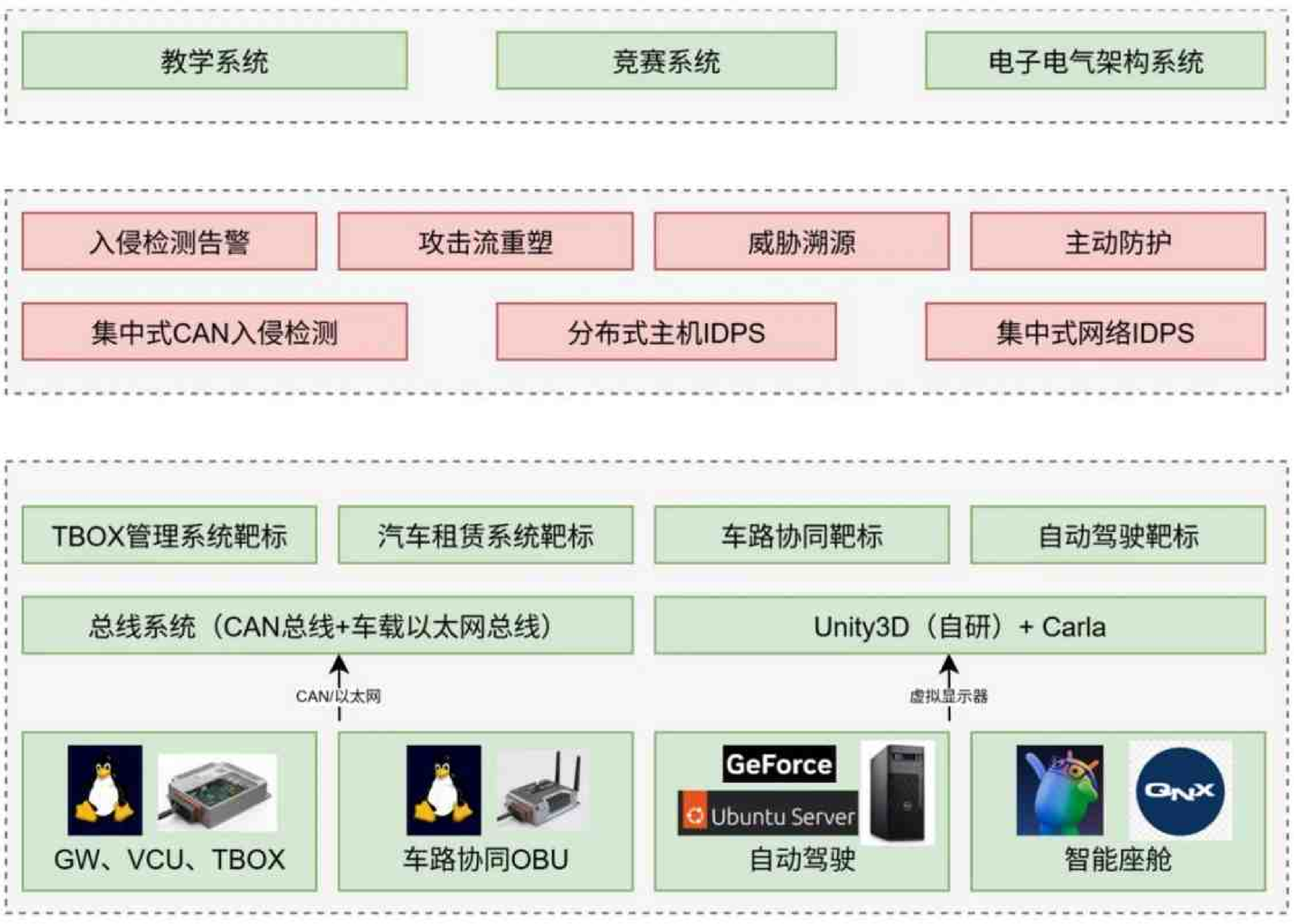
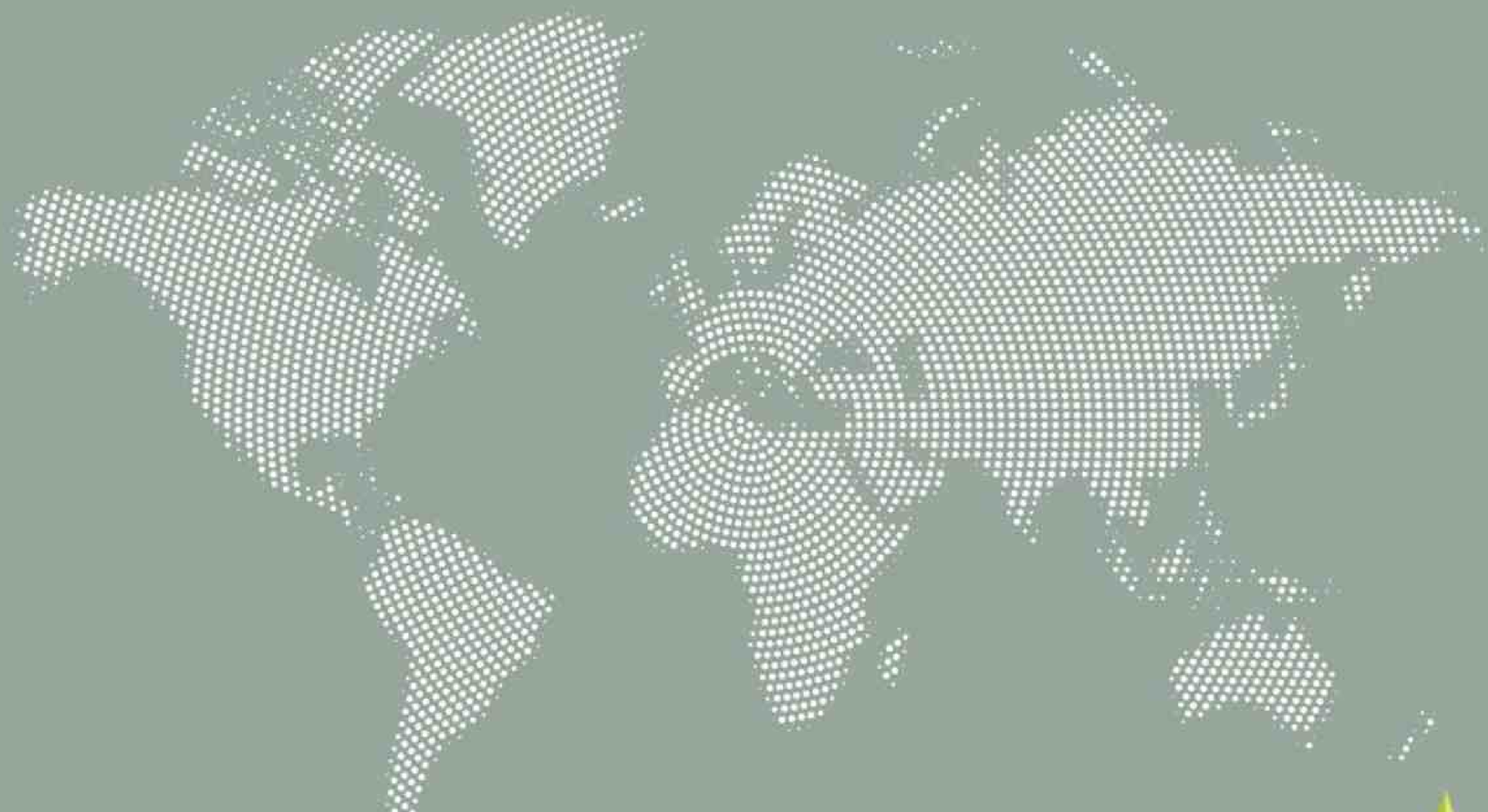


图 4.2　车联网攻防系统构建思路

当攻击者攻击虚拟汽车与对实车进行攻击的方法，是否一致？虚拟汽车是否可以模拟实车软件业务以及满足产品验证的需求呢？下一章我们将一一介绍。

05

虚拟汽车实战表现



本节通过在攻防、仿真、测试三个方面的应用阐述，表现虚拟汽车的攻防业务承载能力，侧面体现虚拟汽车的仿真粒度之细。

5.1 作为汽车构建攻防场景（以局域网控车为例）

首先，构造一个攻击环境，其次，使用端口扫描、木马植入等方式进行攻击，最后检查攻击效果。设计攻击环境时，攻击入口设置为两个，分别是车端入口和云端入口。其中，车端的攻击具体如图 5.1 所示。左侧所示为驾驶员使用方向盘和脚踏板驾驶汽车，方向盘的控制信号经由宿主机转化为 CAN 信号，发送给 VCU，VCU 将控制方向盘数据转化为前端车辆需要的转向、速度、刹车、倒车等信号，将信号发送给 CAN 转 Unity 的协议转换模块，最后经 Unity 引擎时车辆在车路上驾驶。这其中所有的控制经 CAN 总线。车载以太网方面，宿主机连接外部 Wi-Fi 路由器，QEMU 使用桥接技术将启动的虚拟零部件桥接到 Wi-Fi 路由器下，与宿主机共享一个网段，其中，外部 Wi-Fi 路由器可被当做车辆热点来看待，攻击者接入热点后可对汽车零部件发起攻击。

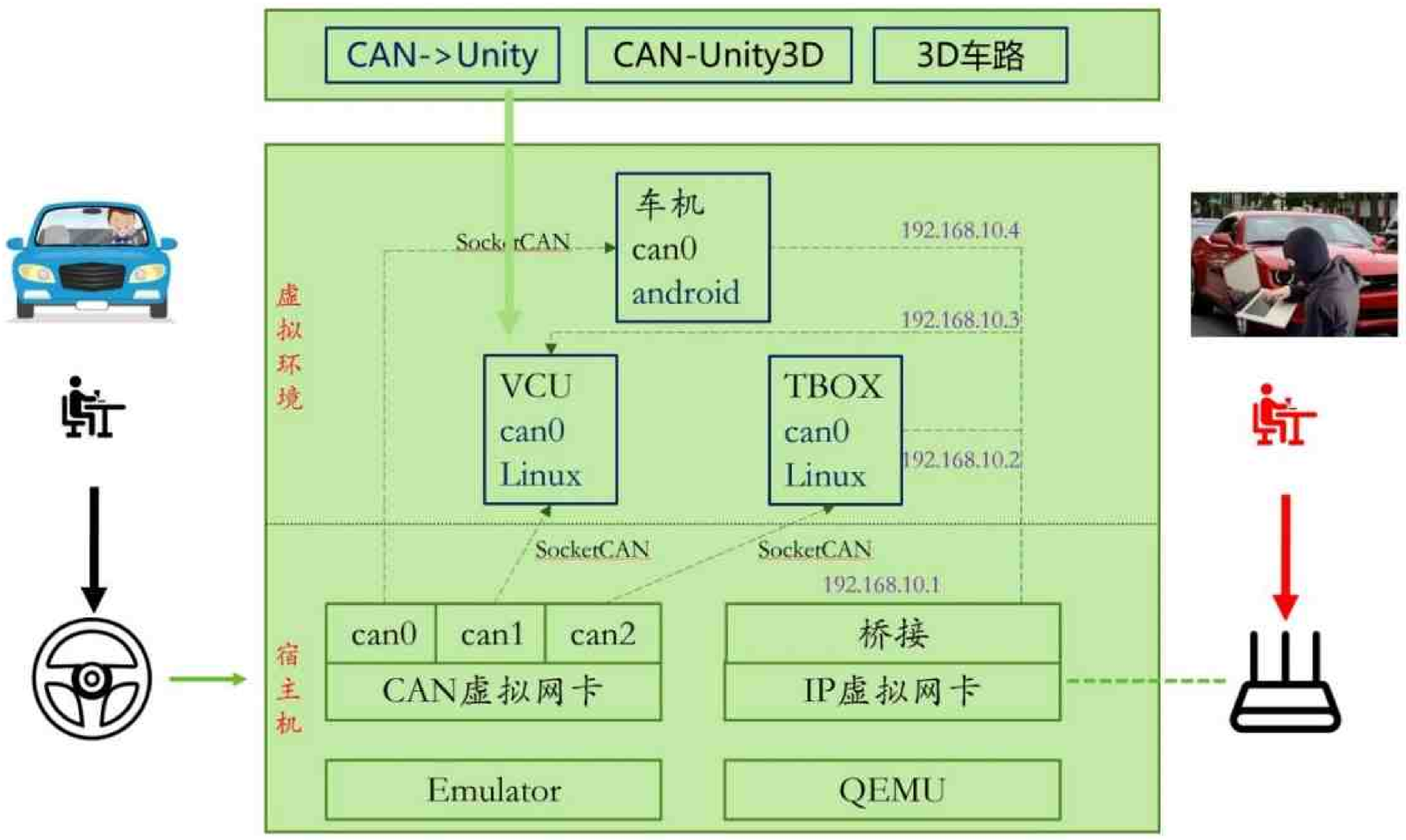


图 5.1 局域网控车场景构建思路

具体的攻击流程为：首先，攻击者接入汽车热点，然后，在热点下对汽车发起主机扫描、端口扫描、弱口令爆破、服务利用等攻击，最后，在获取主机权限后，对零部件发起攻击，如使用 ADB 服务控制车机的空调按钮控制空调，或者发送 CAN 报文实现控车效果。具体如图 5.2 所示。

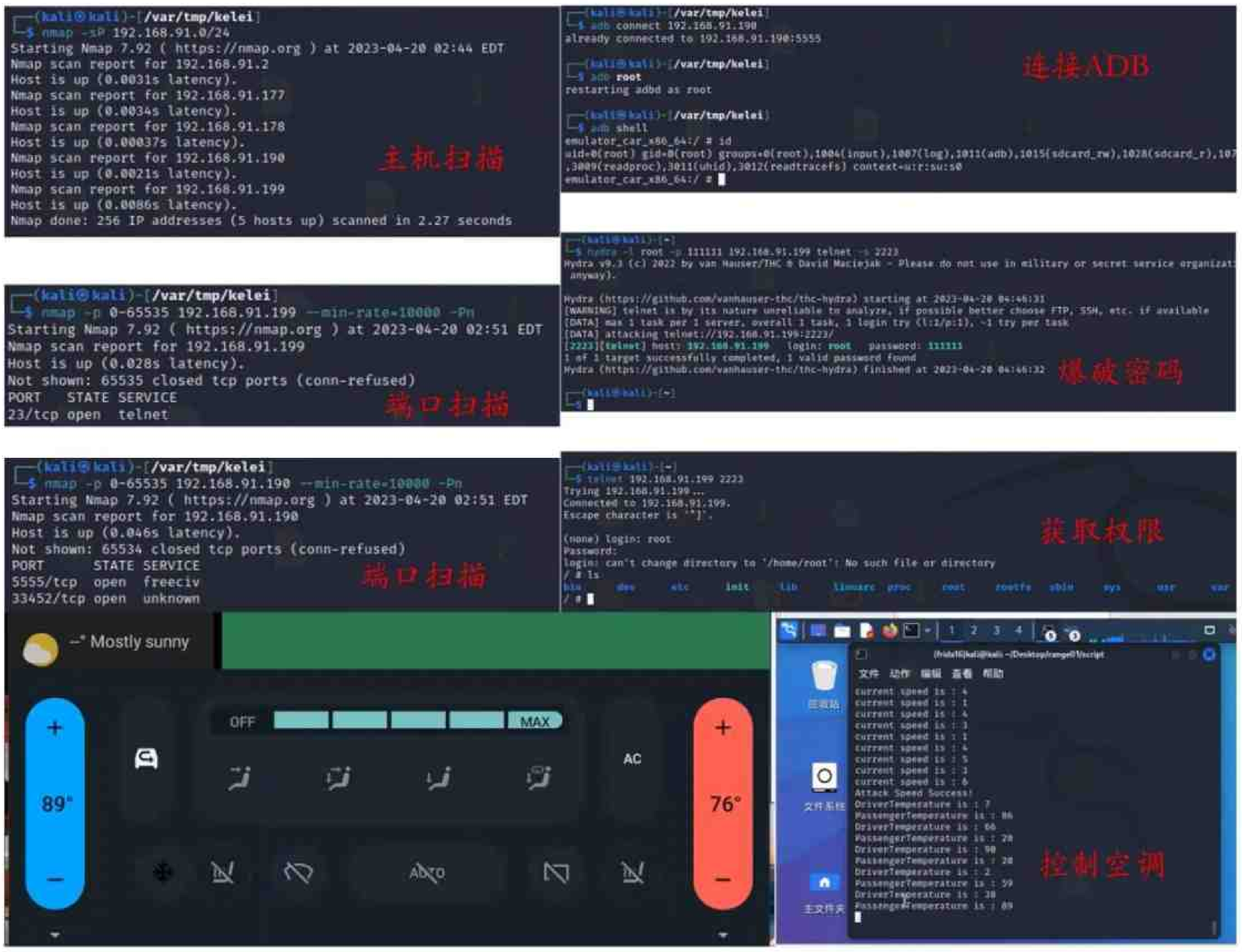


图 5.2　虚拟汽车可支撑的与实车无异的攻防流程

攻击者对云端的攻击与其他领域（如金融、运营商）的攻防技术几乎一致，其中主要涉及对云端资产和手机 APP 的攻击，这里不再赘述。

综上所述，对比历年的汽车网络安全事件和漏洞披露信息可知，在攻击的流程方面，虚拟汽车与实车的控制流程差异不大，不影响攻击效果。

5.2　作为模拟环境运行实车应用

除了可以对其进行攻击以外，虚拟汽车还可以模拟汽车业务。在零部件的模拟过程中，使用的是 QEMU 启动的 ARM64 架构的 Linux 系统零部件，所以，一些使用该架构的实车中的零部件应用，也可以迁移到虚拟零部件中运行。最为基础的操作为，将实车零部件的根 Linux 文件系统迁移到虚拟零部件中，经过定制操作系统内核和驱动后，运行 chroot 命令切换到实车的固件环境下，运行需要被模拟的应用。具体如图 5.3 所示。应用程序名为“mv_app”的应用经过虚拟零部件启动后，与真实零部件环境下的表现一致，这个“一致”体现在两方面，分别是网络侧监听的端口一致，和对 CAN 总线的控制、收发消息的逻辑一致。

图 5.3 对实车固件的仿真表现

对监听的端口和应用，可以进行逆向分析、流量抓取等方式进行漏洞挖掘，对 CAN 总线消息的收发，可以分析车内网总线控车逻辑，利用前者，攻击者可获取零部件权限，利用后者，攻击者可以明确控车报文对部分车辆功能进行控制。

5.3 作为实验平台支撑科研和产品测试

虚拟汽车对汽车产品验证的能力，不仅限于车载 IDPS 这类汽车安全产品，还可以是负责汽车空调控制业务、ADAS 辅助驾驶业务等相关的车载产品。本节以车载 IDPS 为例进行阐述。

车载 IDPS 不应该开启任何对外暴露的端口，它只需要将零部件产生的流量、文件、进程等方面的异常状况进行分析后，向安全运营服务器上传分析结果。将其交叉编译后，可以运行在 ARM64 位架构的零部件内部。

异常的种类非常多，这里我们以两个端口的命令注入漏洞为例。云端可以配置零部件的规则，云端更新规则后，客户端将更新规则到本地生效。此时，攻击者对零部件发起命令注入攻击，IDPS 识别到攻击并将其及时阻断，同时将识别到的攻击上传到运营平台。下图所示为车端的 IDPS 本地日志。具体如图 5.4 所示。

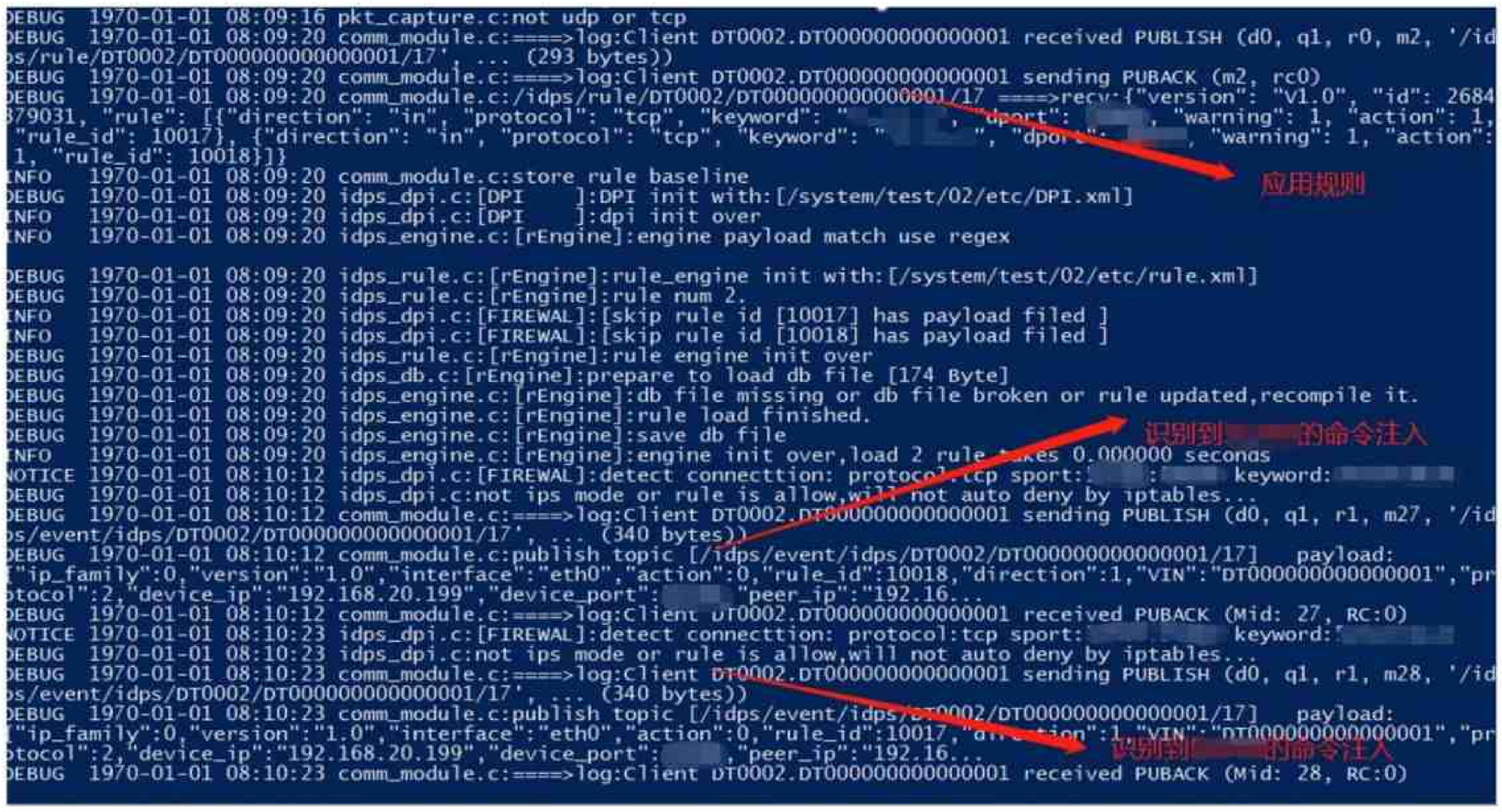


图 5.4　公司级 IDPS 研发测试表现

同时，结合 IDPS 的告警能力，可以实时对攻击的检测，如图 5.5 所示，汽车在监测期间遭受了 90 次攻击，其中 17 次攻击成功，零部件 root 权限被攻击者获取，彻底沦陷。

图 5.5　公司级 IDPS 赋能靶场的表现

5.4　小结

本章就虚拟汽车在攻击场景构建、固件仿真、产品测试这三个方面进行了阐述，首先，从攻击角度看，虚拟汽车作为被攻击的靶标，其在虚拟电子电气架构、虚拟汽车零部件、虚拟车内网通信这三个方

面与实车几乎一致。其次，固件模拟执行和业务复现，在虚拟汽车上表现不错，也可达到多零部件联动仿真的效果。最后，在产品开发测试方面，虚拟汽车可以支撑车载应用的开发工作，甚至可构建大规模攻击检测防护的能力。

06

应用案例/成绩

6.1 赛事案例（大规模并发能力显著）

在“2023 年第四届电信和互联网行业职业技能竞赛暨第十二届信息通信网络安全管理员职业技能竞赛”中，绿盟科技荣获突出贡献奖。车联网作为大赛的创新赛道，靶场给多达 114 支参赛队伍配置了独立的、队员可同时访问的虚拟汽车参赛环境，保证了每支队伍的赛题环境的公平、公正。大赛参赛队员超过 400 名，车联网大赛决赛规模史无前例。

序号	竞赛名称	主办单位	竞赛职业（工种）	决赛时间	联系人	联系电话
25	第四届全国电信和互联网行业职业技能竞赛	中国通信企业协会 中国就业培训技术指导中心	信息通信网络运行管理员 S(5G 网络优化方向） 智能硬件装调员（信创设备方向）	11 月	陈颜芹	010-68200122

图 6.1 竞赛详情

图 6.2 获奖情况

6.2 项目案例（虚实结合过渡自然）

还有一个比较典型的虚实结合的案例。需求源于老师们希望构建一个实物，既能接入实车的零部件，又能构建多个虚拟零部件。综合众多需求建议，为老师构建了一个智能座舱，放在实验室环境下进行体验式教学，大大提高了学生对知识的掌握度，整体交付模式如图 6.3 所示。纯虚拟的汽车系统放在教学环境下支撑理论教学，适合老师们在课堂直接验证攻防效果。实物座舱作为实车零部件的载体，部署 PC 作虚拟汽车软件的载体，将二者使用 USBCAN 适配器（支持 socketcan）结合，综合下来，每一个 CAN 网络域只需要 50 元左右即可。虚拟汽车的车内网仿真能力与实车的差异，小到 50 元即可解决虚实结合的难点，并能实现与实车的自然结合，这再次证实了虚拟汽车攻防系统的强大仿真能力。

图 6.3　虚实结合的形态

07

总结

本文重点阐述了虚拟化技术、汽车电子电气架构的虚拟化、虚拟汽车系统、虚拟汽车实战演练和应用案例这 5 个方面的内容。虚拟汽车系统基于成熟且高度灵活的虚拟电子电气架构实现，能够完整承载整车业务逻辑与通信交互。在此环境中开展的安全攻击测试，无论从攻击路径、技术手段还是流程逻辑上，均与针对实车的攻击高度一致，具备良好的仿真真实性。此外，该系统也可有效支持车载系统及应用的功能开发与测试验证，尤其能够与车载入侵检测系统等安全产品形成协同测试与迭代优化的闭环。

当前行业普遍倡导降本增效，实车业务的虚拟化显著降低了车联网安全研究的设备与环境成本，，甚至在一定程度上实现零边际成本的仿真推演与攻防实验复现，从而支撑汽车业务场景的高效、大规模、可复现运转，这为汽车安全前沿技术研究、产品快速研发及近实战化攻防演练创造了全新的技术条件与操作空间。

我们将持续探索信息安全领域的前沿学术方向，从实践出发，结合公司资源和先进技术，实现概念级的原型系统到成熟产品的转化链条，不断打造具有市场高竞争力的解决方案。我们将积极与有车联网相关授课需求、竞赛需求以及产品测试需求的高校、科研院所、车企和其他类型的客户合作，共同构建开放、协作、可持续的产业生态，助力车联网行业的良性健康发展。

NSFOCUS

THE EXPERT BEHIND GIANTS

巨 人 背 后 的 专 家

多年以来，绿盟科技致力于安全攻防的研究，
为政府、金融、运营商、能源、交通、科教文卫等行业用户和各类型企业用户，
提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。在这些巨人的背后，他们是备受信赖的专家。

NSFOCUS

THE EXPERT BEHIND GIANTS

巨人背后的专家

多年以来，绿盟科技致力于安全攻防的研究，
为政府、运营商、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。
在这些巨人的背后，他们是备受信赖的专家。

总部：北京市海淀区北洼路4号院绿盟科技园
绿盟科技（股票代码300369）

邮编：100089

电话：010-68438880

传真：010-68437328

邮箱：webadmin@nsfocus.com

www.nsfocus.com

扫描绿盟科技官方二维码
可在手机端直接观看报告电子书