“Think Like a Hacker”

Assaf Harel, Chief Scientist and Co-Founder

Gartner COOL VENDOR

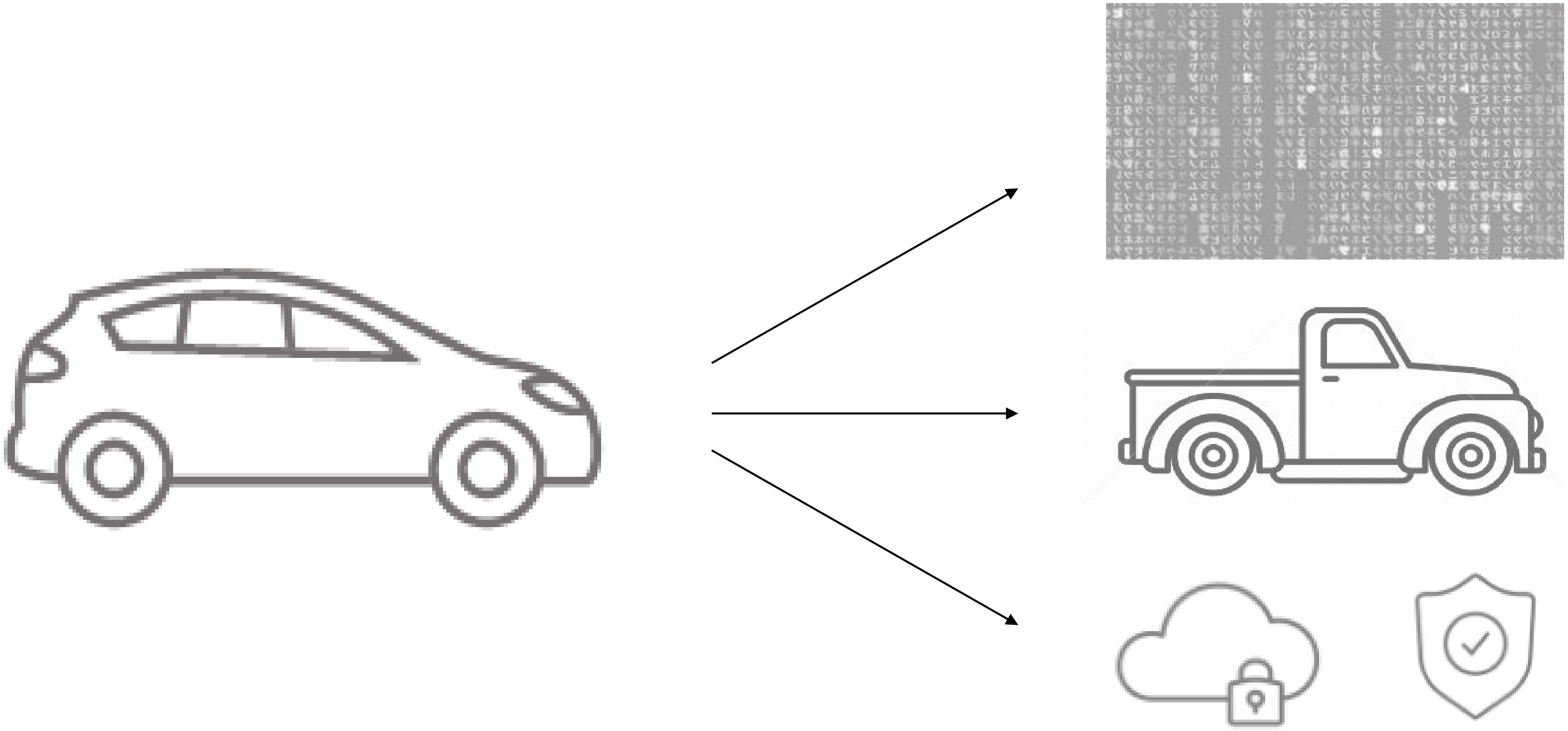
TU Automotive AWARDS WINNER 2017+2018

FROST & SULLIVAN 2017 New Product Innovation Award

auto connected car news WINNER Tech CARS Awards

Karamba Security

What Does it Mean?



Why Would a Hacker Want to Hack a Car?

Cryptocurrency Mining (any ECU)

Personal Information (Infotainment/TCU)

Ransomware (Infotainment)

Car/Cargo Theft (BCM)

Data Manipulation (Fleets) (TCU)

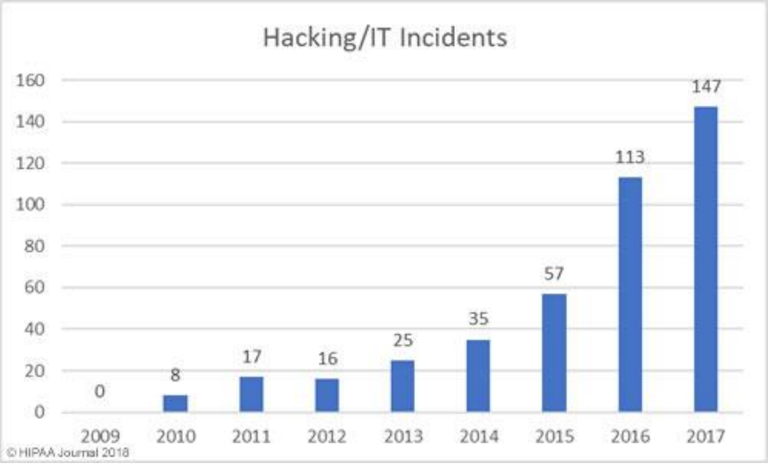
Controlling the Car (Speed & Steering ECUs)

The Automotive Industry is Doing a Great Job

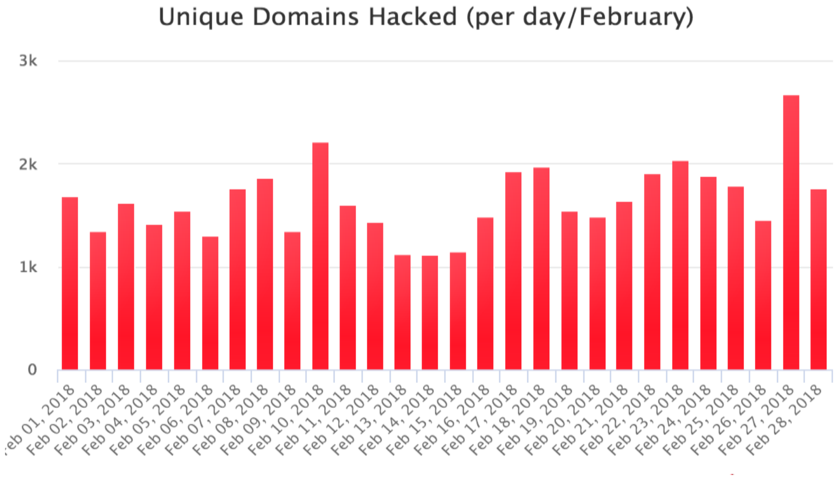
- Separating Domains
- Securing Connectivity
- Signing and Encrypting Images
- Pen Testing

- **However…**

It is All About Motivation



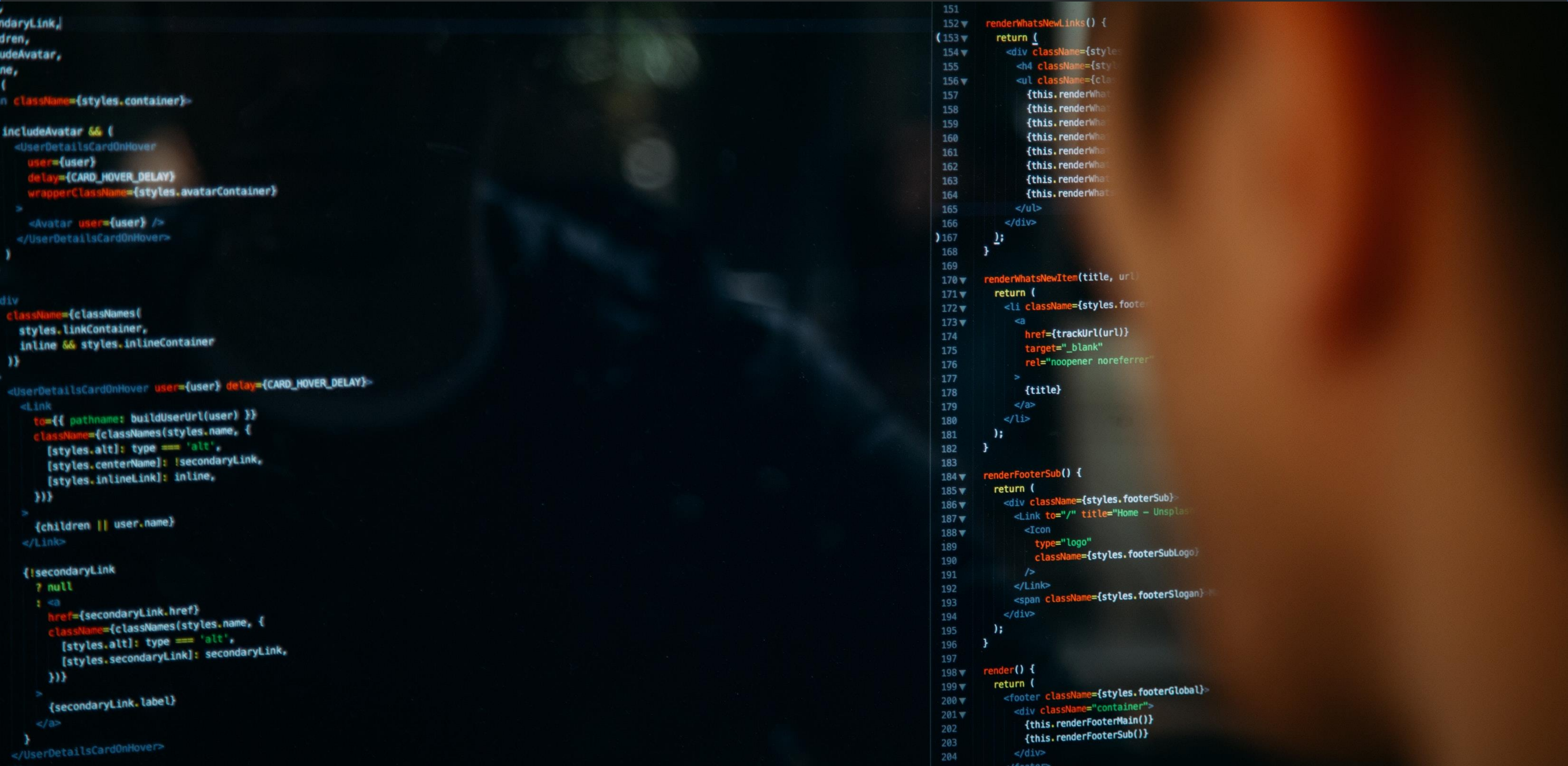
Healthcare Data Breach Statistics



Domain Hacking

Defcon – Car
Hacking Village

So How Does a Hacker Think?



```
ndaryLink,
dren,
udeAvatar,
ne,
(
n className={styles.container}>

 includeAvatar && (
   <UserDetailsCardOnHover
     user={user}
     delay={CARD_HOVER_DELAY}
     wrapperClassName={styles.avatarContainer}
   >
     <Avatar user={user} />
   </UserDetailsCardOnHover>
 )


div
 className={classNames(
   styles.linkContainer,
   inline && styles.inlineContainer
 )}

 <UserDetailsCardOnHover user={user} delay={CARD_HOVER_DELAY}>
   <Link
     to={{ pathname: buildUserUrl(user) }}
     className={classNames(styles.name, {
       [styles.alt]: type === 'alt',
       [styles.centerName]: !secondaryLink,
       [styles.inlineLink]: inline,
     })}
   >
     {children || user.name}
   </Link>

   {!secondaryLink
     ? null
     : <a
       href={secondaryLink.href}
       className={classNames(styles.name, {
         [styles.alt]: type === 'alt',
         [styles.secondaryLink]: secondaryLink,
       })}
     >
       {secondaryLink.label}
     </a>
   }
 </UserDetailsCardOnHover>
```

```

renderWhatsNewLinks() {
  return (
    <div className={style
      <h4 className={sty
      <ul className={cla
        {this.renderWhat
        {this.renderWha
        {this.renderWha
        {this.renderWha
        {this.renderWha
        {this.renderWha
        {this.renderWhat
        {this.renderWhat
      </ul>
    </div>
  );
}

renderWhatsNewItem(title, url
  return (
    <li className={styles.foote
      <a
        href={trackUrl(url)}
        target="_blank"
        rel="noopener noreferrer"
      >
        {title}
      </a>
    </li>
  );
}

renderFooterSub() {
  return (
    <div className={styles.footerSub}>
      <Link to="/" title="Home - Unsplas
        <Icon
          type="logo"
          className={styles.footerSubLogo}
        />
      </Link>
      <span className={styles.footerSlogan}>
    </div>
  );
}

render() {
  return (
    <footer className={styles.footerGlobal}>
      <div className="container">
        {this.renderFooterMain()}
        {this.renderFooterSub()}
      </div>
```

A Hacker Looks for Two Attacks Type

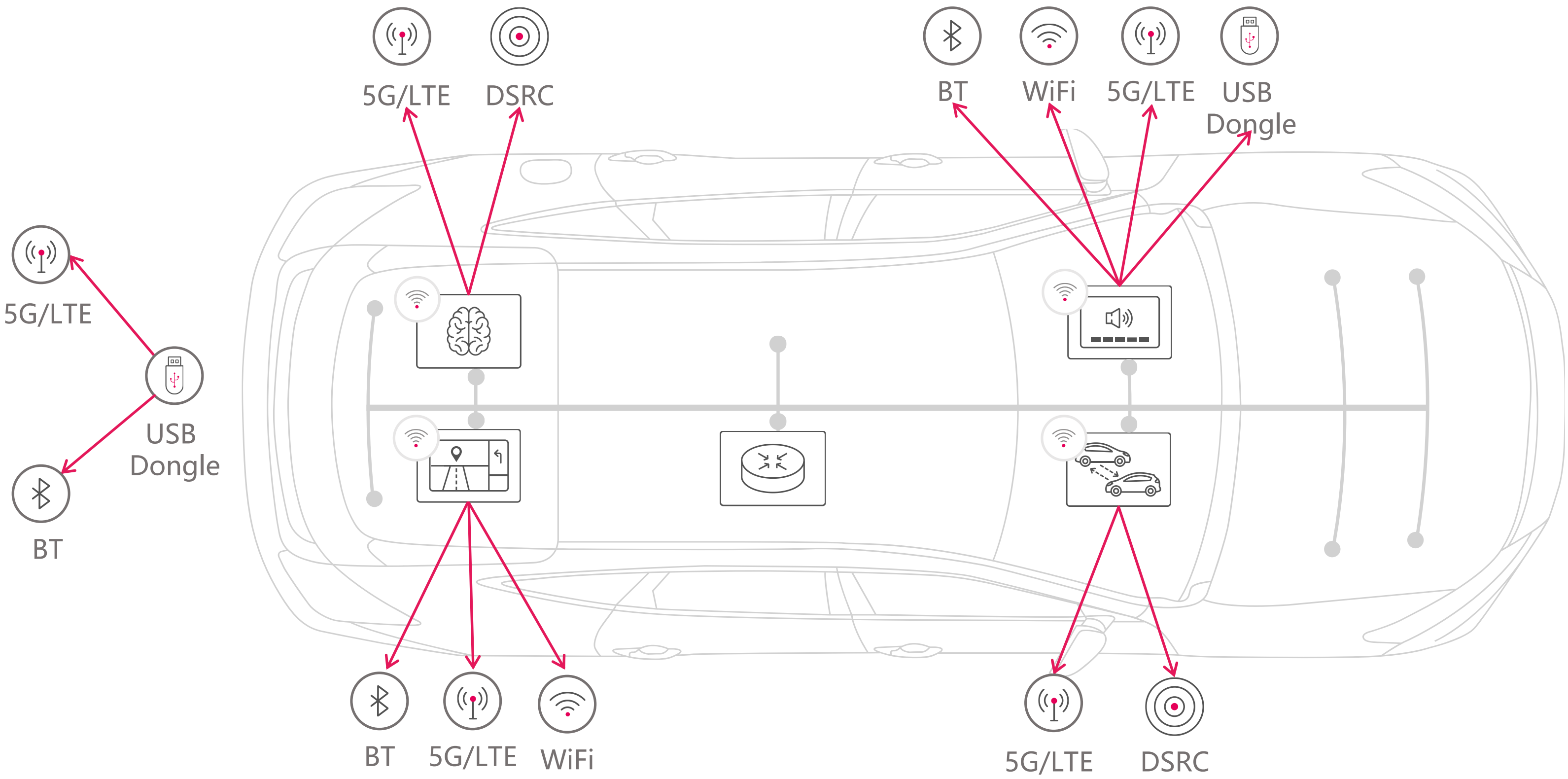
- **Logical attacks** – using existing functionality in unexpected scenarios

- **Code-Injection attacks** – creating a new functionality in an existing module

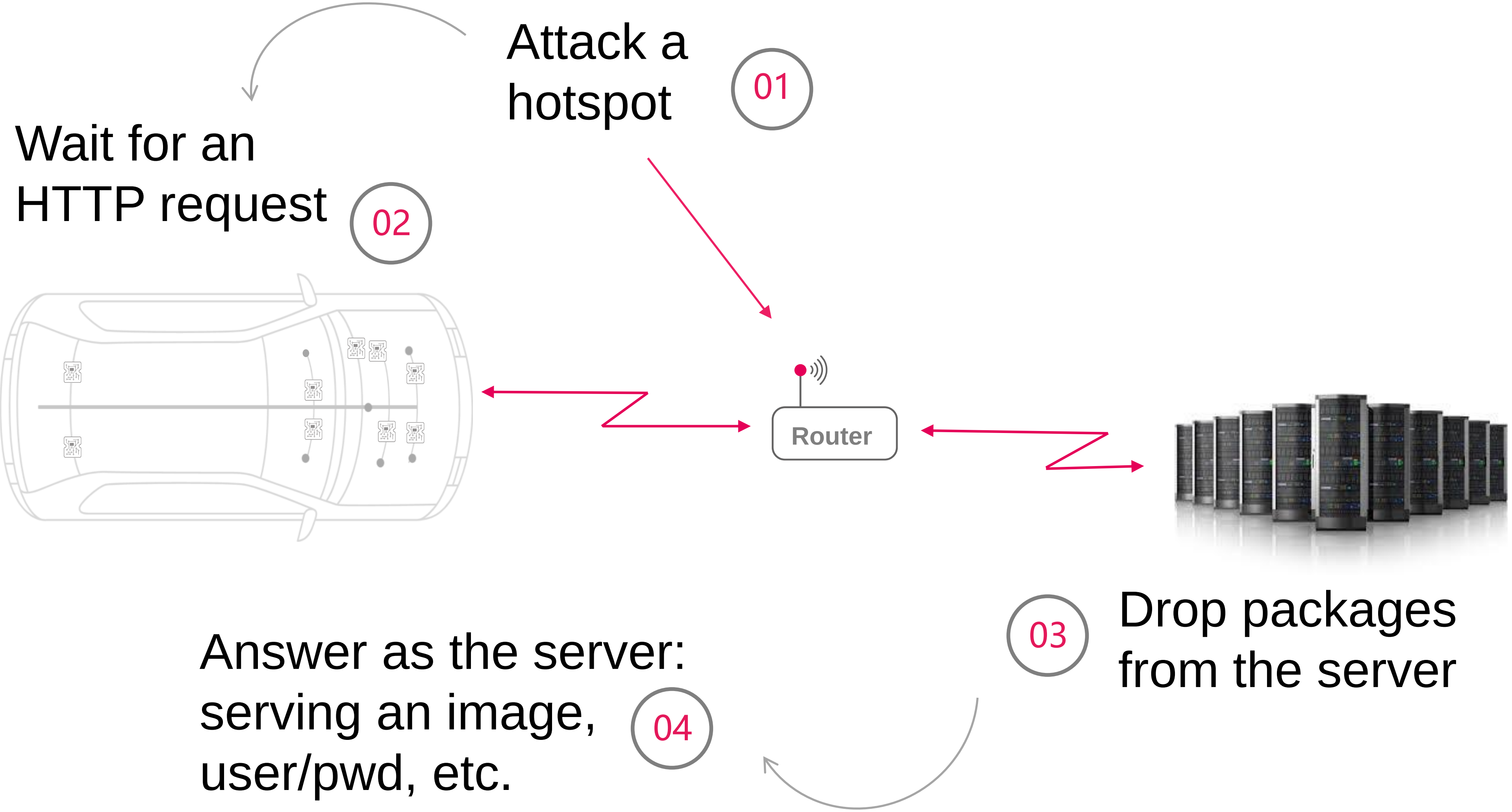
Getting into the Car – A Foot in the Door

Why Connectivity?

- Diagnostics
- FOTA
- Remote Control
- Data monetization
- Internet Services
- V2X
- Autonomous vehicle



Getting into the Car – Impersonation Example



01 Attack a hotspot

02 Wait for an HTTP request

03 Drop packages from the server

04 Answer as the server: serving an image, user/pwd, etc.

Getting into the Car – Other Ways?

- Impersonation – act as the original service
  - Can I send a “key fob” command as the key?
  - Can I serve an update?
- Undocumented opened service
  - Was a debug port left open?
  - Are admin & password connectivity enabled?
- Exploiting coding vulnerabilities
  - Is command injection an option?
  - Can I manipulate the input?

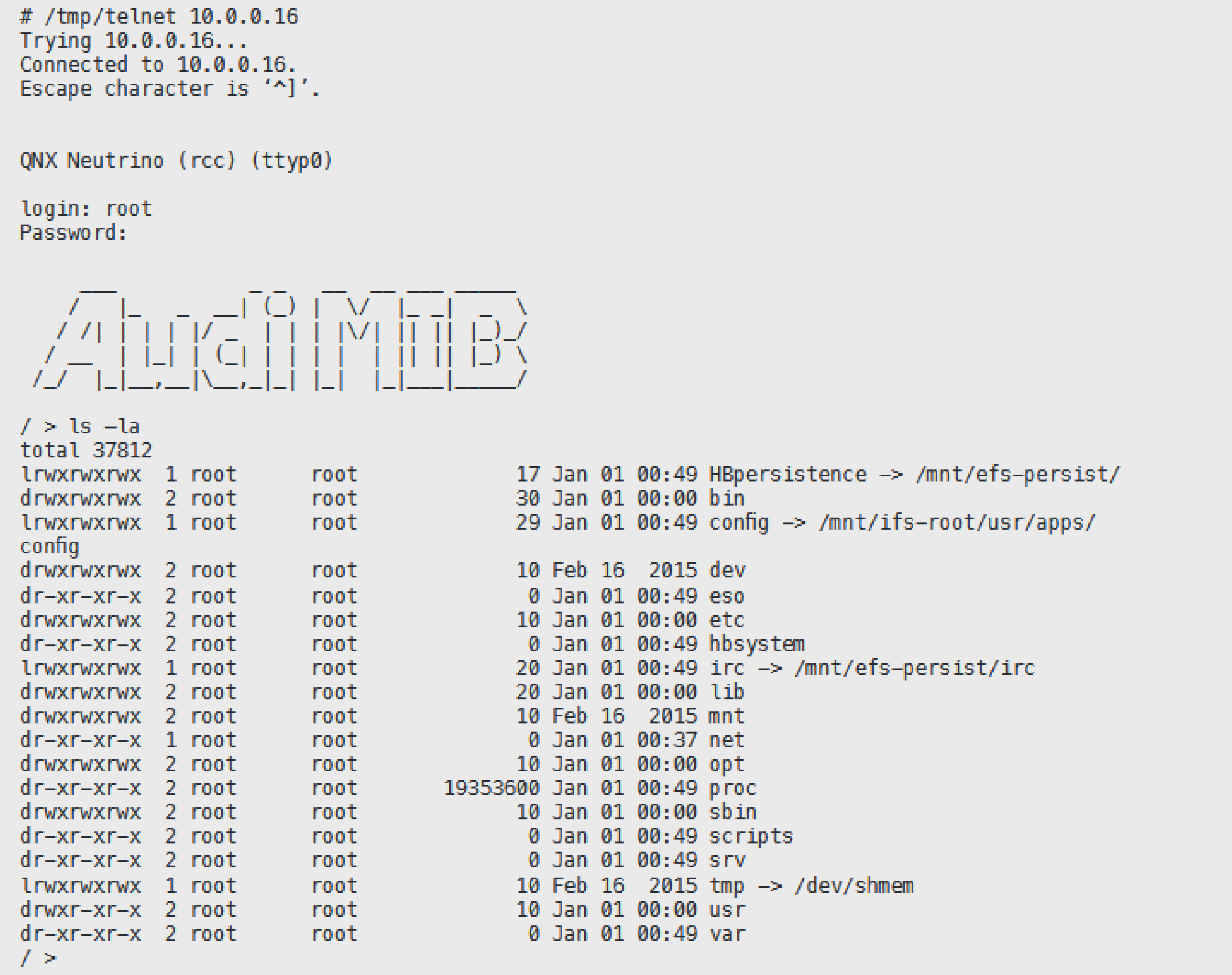
Getting into the Car – Hackers Look for Code

- Getting the image
  - Download updates from official sites
  - Get from flash (JTAG, UART)
  - Extract from memory
- …and source is the best

Recent Automotive Research (Foot in the Door 1)

*“**Volkswagen Golf GTE and Audi A3 Sportback e-tron models** …The two researchers said used a car's **WiFi connection to exploit an exposed port** and gain access to the car's IVI”*

(*) https://www.bleepingcomputer.com/news/security/volkswagen-and-audi-cars-vulnerable-to-remote-hacking/

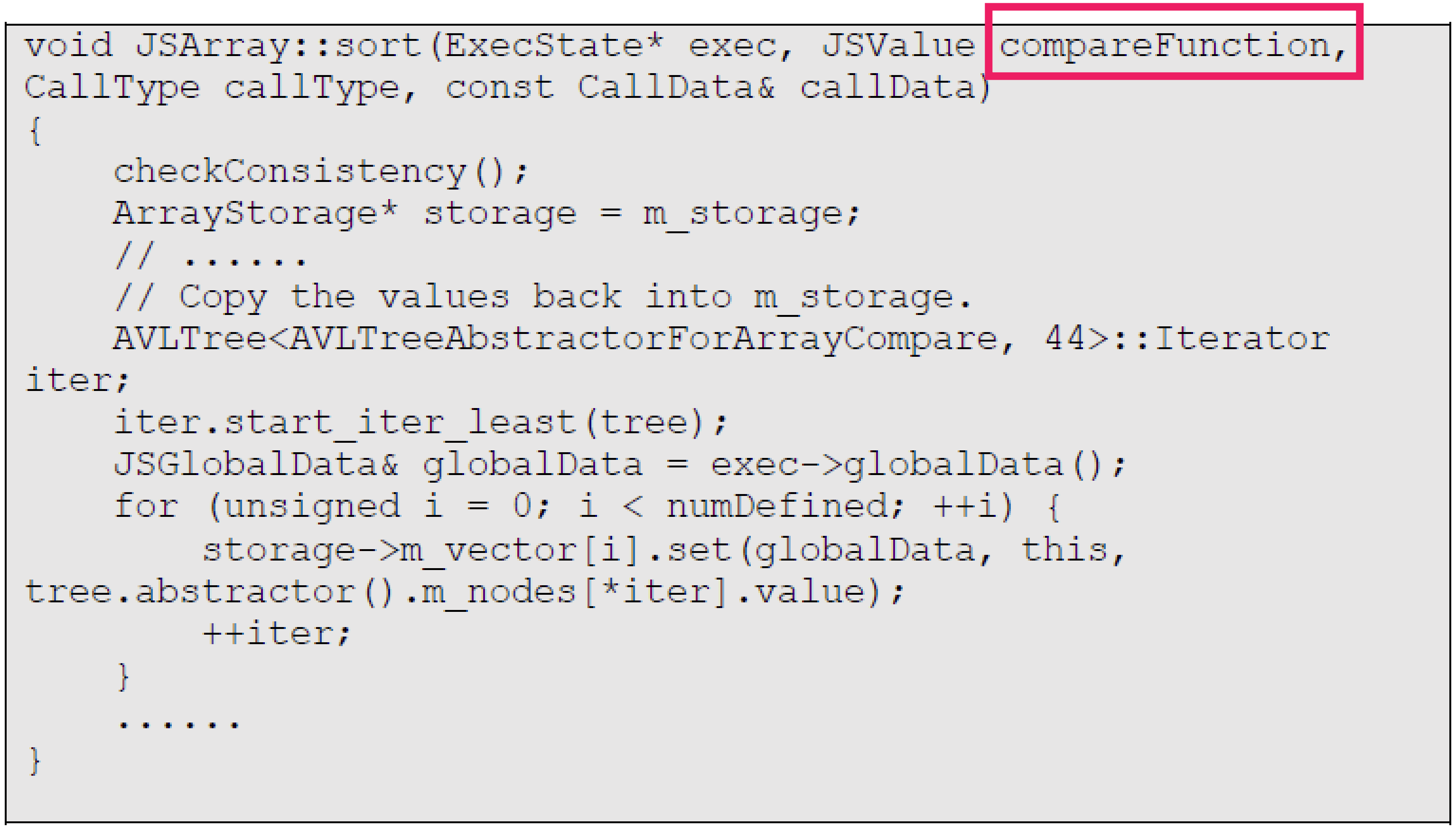


```
# /tmp/telnet 10.0.0.16
Trying 10.0.0.16...
Connected to 10.0.0.16.
Escape character is ‘^]’.


QNX Neutrino (rcc) (ttyp0)

login: root
Password:

     _             _ _ __  __ ___ ____
    / \  _   _  __| (_)  \/  |_ _| __ )
   / _ \| | | |/ _` | | |\/| || ||  _ \
  / ___ \ |_| | (_| | | |  | || || |_) |
 /_/   \_\__,_|\__,_|_|_|  |_|___|____/

/ > ls -la
total 37812
lrwxrwxrwx  1 root      root             17 Jan 01 00:49 HBpersistence -> /mnt/efs-persist/
drwxrwxrwx  2 root      root             30 Jan 01 00:00 bin
lrwxrwxrwx  1 root      root             29 Jan 01 00:49 config -> /mnt/ifs-root/usr/apps/
config
drwxrwxrwx  2 root      root             10 Feb 16  2015 dev
dr-xr-xr-x  2 root      root              0 Jan 01 00:49 eso
drwxrwxrwx  2 root      root             10 Jan 01 00:00 etc
dr-xr-xr-x  2 root      root              0 Jan 01 00:49 hbsystem
lrwxrwxrwx  1 root      root             20 Jan 01 00:49 irc -> /mnt/efs-persist/irc
drwxrwxrwx  2 root      root             20 Jan 01 00:00 lib
drwxrwxrwx  2 root      root             10 Feb 16  2015 mnt
dr-xr-xr-x  1 root      root              0 Jan 01 00:37 net
drwxrwxrwx  2 root      root             10 Jan 01 00:00 opt
dr-xr-xr-x  2 root      root       19353600 Jan 01 00:49 proc
drwxrwxrwx  2 root      root             10 Jan 01 00:00 sbin
dr-xr-xr-x  2 root      root              0 Jan 01 00:49 scripts
dr-xr-xr-x  2 root      root              0 Jan 01 00:49 srv
lrwxrwxrwx  1 root      root             10 Feb 16  2015 tmp -> /dev/shmem
drwxr-xr-x  2 root      root             10 Jan 01 00:00 usr
dr-xr-xr-x  2 root      root              0 Jan 01 00:49 var
/ >
```

Karamba Security

Recent Automotive Research (Foot in the Door 2)

- Browser hacking
  - “QtCarBrowser Safari/534.34“
  - Changing the compare function in Java Script
  - Gaining access to the ECU



```
void JSArray::sort(ExecState* exec, JSValue compareFunction,
CallType callType, const CallData& callData)
{
    checkConsistency();
    ArrayStorage* storage = m_storage;
    // ......
    // Copy the values back into m_storage.
    AVLTree<AVLTreeAbstractorForArrayCompare, 44>::Iterator
iter;
    iter.start_iter_least(tree);
    JSGlobalData& globalData = exec->globalData();
    for (unsigned i = 0; i < numDefined; ++i) {
        storage->m_vector[i].set(globalData, this,
tree.abstractor().m_nodes[*iter].value);
        ++iter;
    }
    ......
}
```

Vulnerable Function

(*) FREE-FALL: Hacking TESLA from Wireless to CAN Bus (Keen Security Lab, 2017)

In the Car – How Can We Pass the Gateway ?

- Flash the Gateway
- Hack the Gateway
- Bypass the Gateway –
  - using approved CAN commands in unexpected scenarios

In the Car – How Can We Pass the Gateway ?

- Hack it – Errors in Ethernet packet handling

  (Internal Research for Tier-1 company)

  - Sending the same packets 10 times has caused buffer overflow
  - Enables running a shell command (left on the device)
  - Enables changing the GW configuration

- Bypass it – Activating Park Assistant

  (Internal Research for OEM company)

  - Setting the Park Assistant ECU to diagnostic mode while engine is running
  - Sending Park Assistant messages from another ECU, causing the wheel to turn
  - Relatively easy to do over CAN (no authentication)

In the Car – What About Other ECUs?

- We can Flash ECUs using UDS commands
  - Many ECUs do not apply secure boot
  - Extract encryption keys from binary
  - Use a vulnerable older version
  - Send UDS commands (thru the Gateway)
- Find Buffer Overflow
  - UDS protocol has potential for vulnerabilities
  - Enables running malicious code on the ECU

“Think Like a Hacker”

Questions?