Testing the Security and Performance of Automotive Ethernet ECUs

Bogdan Tenea
Product Specialist

íxíá
Security

Automotive Ethernet ECUs

SECURITY CONCEPTS
The connected car

- Connected cars use multiple new interconnects
- The number of ECUs in the car is growing fast
- This makes security even more complicated
Hacking Possible Results

• Possible results of car hacking:
  – Loss of privacy
  – Damage to the vehicle
  – Theft
  – Bodily injury

Security is important because it affects functional safety

• When revealed, vulnerabilities cause negative media exposure, even if not exploited

Definition: a vulnerability is a flaw or weakness in a system’s design or implementation that can be exploited in order to violate the system’s integrity or security policy
Security Concerns

• Proven car hacking examples
  – In the past, done by directly connecting to the CAN bus
  – Most recently, via wireless networked interfaces
• High severity – hackers might take control of steering and brakes
• Moving the in-car network to Automotive Ethernet and TCP/IP will affect the security landscape - for hackers this lowers the entry barriers:
  – Automotive Ethernet device connectivity is relatively inexpensive
  – Lots of protocol analyzers available
  – Lots of experience from the IT world
Layered Security

- Using IT world technology implies that you can benefit from its experience on security!
- A chain is only as strong as its weakest link – security of the ECUs should be seen as layered components.
Securing the Layers

Question: Which is the weakest link?
Answer: Every layer needs to be tested!

• Security on the physical layer is generally not seen as an issue, at least for wired networks in controlled environments, such as Ethernet in a car.
• Security on the Application layer is very dependent on the actual implementation of the applications running on each ECU.
• Security on the Network layer can benefit from IT world experience, because it is using standard protocols from the IT world.
Application Security

- Well known vulnerabilities are the easiest way for a hacker to get access to a system
- IT security professionals generally first start with the Common Vulnerabilities and Exposures (CVE) database
  - Lists publicly known vulnerabilities
  - Some of the vulnerabilities are generic (i.e. HTTP servers)
  - Use of the CVE is standardized by the ITU, NIST
  - Products for testing over 50,000 identified vulnerabilities are available on the market at this moment
1. Proper architecture, design, and implementation of network architecture is the first line of defense

2. **Conformance testing** is the second line of defense
   - Implementation conforms to design specification and standards

3. **Robustness testing** is the third line of defense
   - Does the device survive a Denial of Service (DoS) attack?

Will all future Automotive Ethernet ECUs go through this?
Robustness Testing

- **Robustness testing**, in the context of network communication stack testing, means subjecting the stack to **abnormal protocol behavior**:  
  - Corner case protocol state/functionality  
  - Invalid protocol field values  
  - Invalid protocol message order, timing, number  
  - Invalid protocol state triggers  
  - Abnormal communication load up to network saturation  
    - Known in the IT industry as Denial of Service **flood attacks**

- **Expectation for automotive** is that the stack will continue to function and protect itself
Robustness Importance

Considering a gateway that may have a TCP/IP stack connected to:

- other ECUs through the Automotive Ethernet network
- the wireless LTE or Wi-Fi connections on board
- or and maybe even on the diagnostics port

Abnormal protocol behavior on one port should not crash the stack and disable the gateway functionality on the CAN network.
Robustness Testing

Automotive Ethernet and TCP/IP

ROBUSTNESS EXAMPLES
Network Layers

Layer 1
- Automotive Ethernet Physical Layer – 100BASE-T1

Layer 2
- IEEE Ethernet + 802.1Q VLANs

Layer 3
- IP

Layer 4
- TCP
- UDP

Layer 5-7
- Applications

Protocol Layering:
- ARP
- DHCP
- ICMP
- SD
- SOME/IP
- Ethernet + TCP/IP

Application

Middleware

Ethernet + TCP/IP
Ethernet Robustness

1. Does the device ignore unknown upper protocols?
2. What happens when you send unusual sized frames?
   • IEEE 802.3 – total length must be between 64 and 1536
   • send frames smaller than 64 bytes (runt frames)
   • send frames bigger than 1536 bytes (jumbo frames)
3. Does the device drop frames with an invalid FCS?
   • invalid data should not be delivered to upper layers
4. Does device maintain functionality with high frame rate and/or data rate?
   • frame processing is usually tied up to CPU interrupts
   • overwhelming the CPU could lead to crashing the device
# Ethernet Example Test

## Synopsis

An Ethernet stack must be able to withstand receiving **traffic at line rate, without crashing or disabling the whole device on which the stack is running on**. To do this the stack may activate **defensive behavior** either by rate limiting incoming traffic, or by disabling the specific network interface entirely. **If DUT has any other exposed interfaces, it must maintain service on those interfaces while under extreme load** on the tested interface. After incoming line rate traffic is stopped, DUT **must resume normal operation on the tested interface** within a reasonable amount of time.

## Prerequisites

| **ARP** must be enabled on the tested interface |

## Test setup

| Tester directly connected to the Device under Test |

## Test Input Parameters

| DIFACE_O_MAC_ADDR, FloodTime, RecoveryTime |

## Test Procedure

| 1. TESTER: <HOST-1> Sends **Ethernet line rate to DUT** through <DIface-0> for <FloodTime> seconds:  
    - Destination MAC Address set to <DIFACE_O_MAC_ADDR>  
    - Source MAC Address set to <RANDOM-MAC>  
    - Ethertype set to “0x0B07” (reserved protocol value)  
  
  2. DUT: While receiving line rate traffic DUT must maintain functionality on other exposed network interfaces  
  
  3. TESTER: Wait for <RecoveryTime> seconds for DUT to recover from any defensive behavior  
  
  4. TESTER: Sends ARP Request to DUT through <DIface-0>  
  
  5. DUT: Sends ARP Response |

## Pass Criteria

| 2. DUT: While receiving line rate traffic DUT must maintain functionality on other exposed network interfaces  

  5. DUT: Sends ARP Response |
ARP Robustness

1. For Ethernet, HW length must be 6 and for IP, Proto Length must be 4
   • does the device correctly discard frames with invalid lengths specified?
2. Operation can be either request (0x1) or reply (0x2)
   • invalid ID specified for selecting the operation from a table could crash the stack
3. Source Protocol Address needs to match the hosts sub-network
   • if not respected this enables protocol address spoofing
4. Does device maintain functionality on high ARP Request rate?
   • ARP table cache is usually populated from incoming ARP Requests
   • does the device maintain functionality while ARP table is full?
## ARP Example Test

<table>
<thead>
<tr>
<th><strong>Synopsis</strong></th>
<th>„When an address resolution packet is received, the receiving Ethernet module gives the packet to the Address Resolution module which goes through an algorithm similar to the following: Negative conditionals indicate an end of processing and a discarding of the packet. ?Is the opcode ares_op$REQUEST?“ (Note: In this test TESTER sends an ARP Packet with opcode field set to an invalid value, and expects that DUT will not send any ARP Response)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prerequisites</strong></td>
<td>Device under Test needs to have an IP stack and ARP enabled</td>
</tr>
<tr>
<td><strong>Test setup</strong></td>
<td>Tester directly connected to the Device under Test</td>
</tr>
<tr>
<td><strong>Test Input Parameters</strong></td>
<td>HOST-1-IP, Dlface-0-IP, INVALID_OPERATION, ParamListenTime</td>
</tr>
</tbody>
</table>
| **Test Procedure** | 1. TESTER: <HOST-1> Sends ARP packet to DUT through <Dlface-0> containing:  
   - Source IP Address set to <HOST-1-IP>  
   - Destination IP Address set to <Dlface-0-IP>  
   - Operation code set to INVALID_OPERATION  
2. TESTER: <HOST-1> Sends ARP Request to DUT through <Dlface-0>  
3. TESTER: <HOST-1> Listens (up to <ParamListenTime>) on <Dlface-0>  
4. DUT: Sends only one ARP Response |
| **Pass Criteria** | 4. DUT: Sends only one ARP Response |
| **Reference** | Derived from RFC 826 "An Ethernet Address Resolution Protocol", section "Packet Reception" (MUST) |
IPv4 Robustness

1. Invalid field values for version/protocol should be ignored
2. IHL gives the header length including IP Options
   • what happens when the indicated IHL is less than the minimum 20 bytes?
   • survive receiving invalid header options with valid IHL?
3. Total length must be set to the total length of the diagram
   • what happens if the indicated length is bigger that the actual received one?
4. Invalid checksum data should not be delivered to upper layers
5. Fragmentation is one of the biggest sources of IP attacks
   • send the device lots of incomplete fragments, exhaust cache buffers
## IPv4 Example Test

### Synopsis
An TCP/IP stack must be able to withstand receiving incomplete IPv4 fragments at high rate without crashing or disabling the whole device on which the stack is running on. To do this the stack may activate defensive behavior either by rate limiting incoming traffic, or by disabling the specific network interface entirely. **If DUT has any other exposed interfaces, it must maintain service on those interfaces while under extreme load** on the tested interface. After line rate traffic is stopped, DUT must resume normal operation on the tested interface within a reasonable amount of time.

### Test setup
Tester directly connected to the Device under Test

### Test Procedure
1. **TESTER**: <HOST-1> Sends valid IPv4 fragments at high rate to DUT through `<Dlface-0>` for `<FloodTime>` seconds containing the following:
   - Source MAC Address set to `<RANDOM-MAC>`
   - Source IP Address set to `<RANDOM-IP>`
   - MF (More Fragments) in Flags field set to 1
   - Length set to a value bigger than total packet data
2. **DUT**: **While receiving line rate traffic DUT must maintain functionality on other network interfaces**
3. **TESTER**: Wait for `<RecoveryTime>` seconds for DUT to recover from any defensive behavior
4. **TESTER**: Sends ARP Request to DUT through `<Dlface-0>`
   - Sender Protocol Address is not in the list of previously sent `<RANDOM-IP>`
5. **DUT**: Sends ARP Response

### Notes
The `<RANDOM-MAC>`, `<RANDOM-IP>` pair shall be consistent such that the IP packets do not to trigger any spoofing protection on the DUT
TCP Robustness

1. Flood with new connection requests (SYN flood)
   • Usually handled by not allocating resources on first SYN
   • Delay allocation until the 3-rd phase by using an sequence number from a hash on source IP and port that can be used to validate the ACK message

2. Flood with open connections (SYN+ACK flood)
   • Attacker can deduce the correct seq. number in phase 3
   • A DUT can defend by closing unused open connections
   • Attacker can fake transmitting data on the connections
   • DUT may defend by limiting connections from same source IP
OPEN Alliance TC8

- OPEN Alliance Technical Committee 8 (TC8) responsible for ECU and network testing has recently created TCP/IP conformance test cases.
- A lot of attention was directed towards also adding robustness tests.
Performance Testing

- Performance testing - how does a system perform in terms of responsiveness and stability under a particular workload?
- Above definition is quite extensive
The typical IT test for switch performance is specified by RFC 2544. It determines **throughput and latency** under specific conditions:

- How many ports transmitting
- How many are receiving
- One to one, many to many or fully meshed?
- One direction/bi-directional
- Protocols
- Broadcast/Multicast
- Test duration
- Test different frames size

Because automotive is an engineered network, it has the luxury of testing only for specific engineered traffic configurations.

Tools exist to precisely emulate usual and malicious traffic patterns that could flow in the car network.
• Audio Video Bridging was introduced to allow switches to provide guarantees regarding throughput and latency for A/V traffic
• The specific configuration and traffic pattern in which the switch network will operate needs to be tested
• **Tests need to be run for a long duration**, as lab results have shown performance is not constant
• If reliable enough, AVB/TSN can be used for body control functions
TCP/IP Performance

• Throughput scenarios:
  – Sending or receiving data
  – Varying window sizes
  – Varying number of connections

• Latency/responsiveness of the stack
  – Varying number of connections
  – Different throughput scenarios

• Session establishment rate

• Number of concurrent connections
Thank you!

More @
www.ixiacom.com/automotive-ethernet

ixia