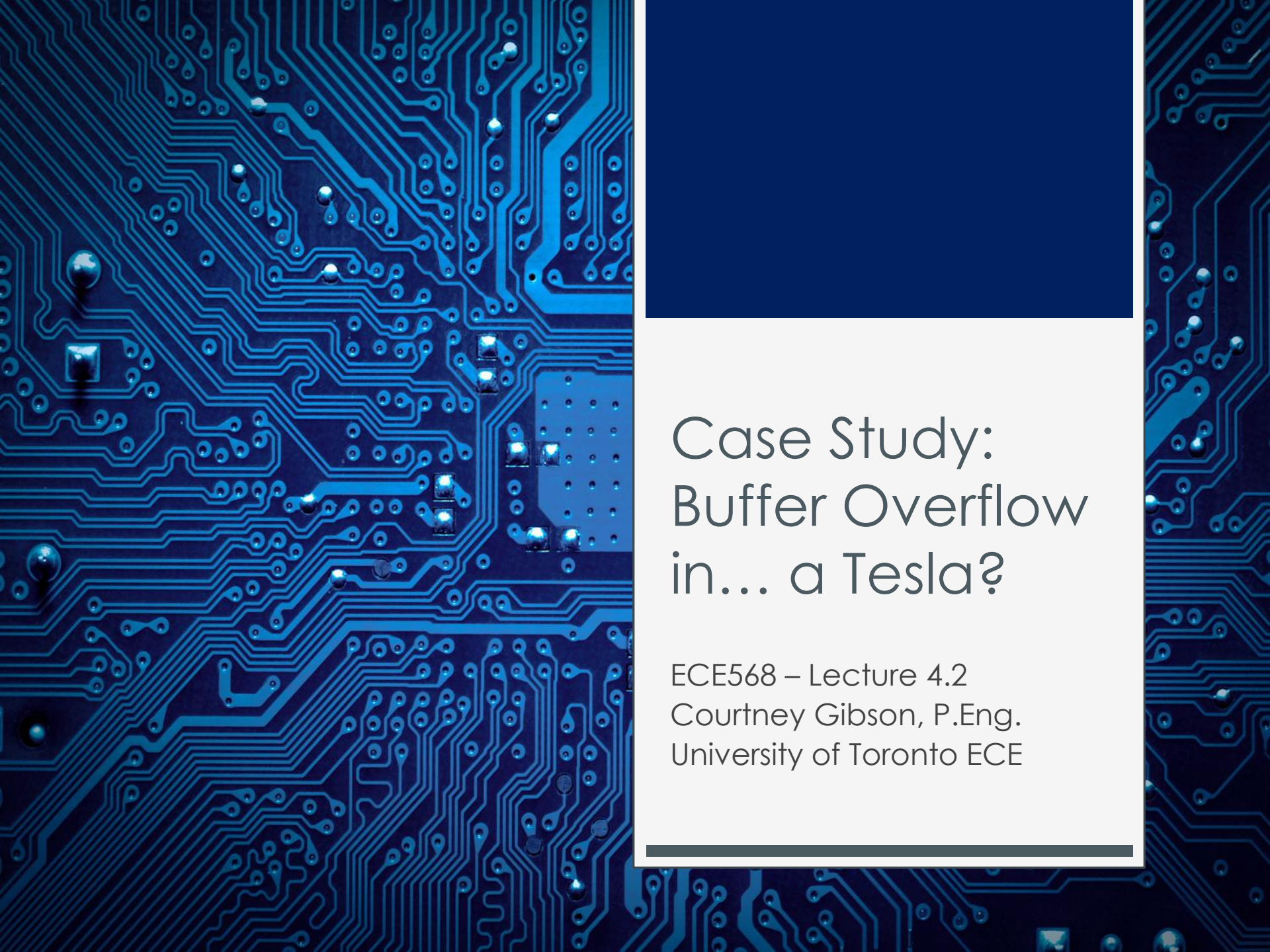


Case Study: Buffer Overflow in… a Tesla?

ECE568 – Lecture 4.2

Courtney Gibson, P.Eng.
University of Toronto ECE

ConnMan: Network Manager

- A **network manager** is the O/S component responsible for configuring and maintaining network connections
  - *e.g.*, Ethernet, Wifi, Bluetooth, etc.
  - Device configuration (MAC address, Wifi frequency, etc.)
  - Network parameters (SSID/password, DHCP, IP, netmask, etc.)
  - Proxy server for several network protocols: DNS, NTP, etc.
- **ConnMan** ("**Conn**ection **Man**ager") is a light-weight network manager used in many embedded systems
  - GENIVI In-Vehicle entertainment systems, Tesla
  - Smartphones
  - Nest thermostats, PVRs

DNS: Response Compression

- DNS responses include a special encoding for the hostnames; this helps the receiver parse the response and allocate appropriately-long buffers
- The fully hostname (the "FQDN") is divided into *labels*:
  - www.xyzindustries.com -> [www, xyzindustries, com]
- Each hostname is encoded as a series of labels, with a leading 8-bit integer that specifies the label length
  - **[3]**www**[13]**xyzindustries**[3]**com**[0]**

DNS: Response Compression

Each hostname is encoded as a series of labels, with a leading 8-bit integer that specifies the label length

- **[3]**www**[13]**xyzindustries**[3]**com**[0]**

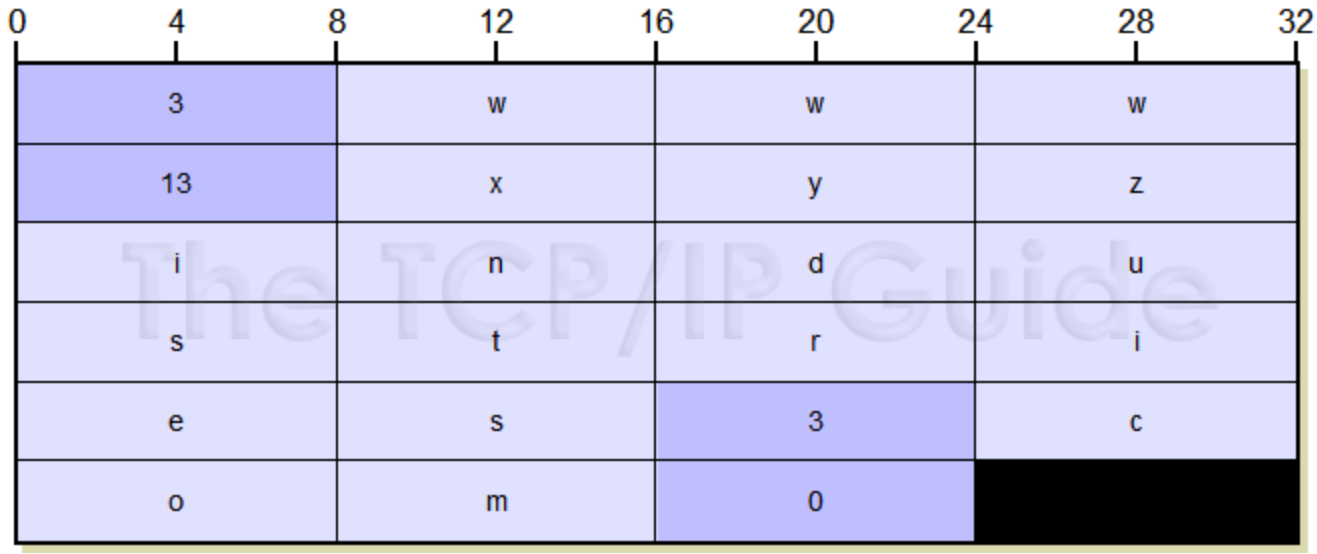


Image Source: www.tcpipguide.com

DNS: Response Compression

- When doing a DNS lookup, the response from the DNS server often contains a lot of repetitive information
  - (*i.e.*, the domain name can appear multiple times in the same response)
- As a result, a form of compression can be used.
  - Suppose the previous "www.xyzindustries.com" shows up at byte 47 of the DNS reply
  - If the same name (or a portion of it) is required again, it can be encoded as a special "field length" of 192, followed by the byte offset of the other copy of the name ("47" in this case)
  - This allows repetitions to be encoded as 2 bytes (which is a significant savings)

CVE-2021-26675

- Reported by Tesla (Feb 8, 2021)
- A bug in the DNS proxy plug-in in ConnMan allows a malicious DNS reply to uncompress into a large string that can overflow an internal buffer that exists on the process stack
  - Insufficient checking of whether the receiving buffer has enough space for the uncompressed string
- This means that a remote attacker who can control (or fake) a DNS response could perform a buffer overflow attack on the ConnMan application
- ConnMan runs as “root” on the devices it manages

CVE-2021-26676

- Remote buffer overflow attacks are harder to accomplish when the attacker doesn't have visibility to the victim's stack (e.g., what value to put into the Return Address?)
- A second bug was discovered in the DHCP client in ConnMan, where an uninitialized structure stored on the stack can leak stack values to a remote attacker (including stack-related addresses)
    - To be exploitable, this vulnerability requires the attacker to be on the same subnet as the victim

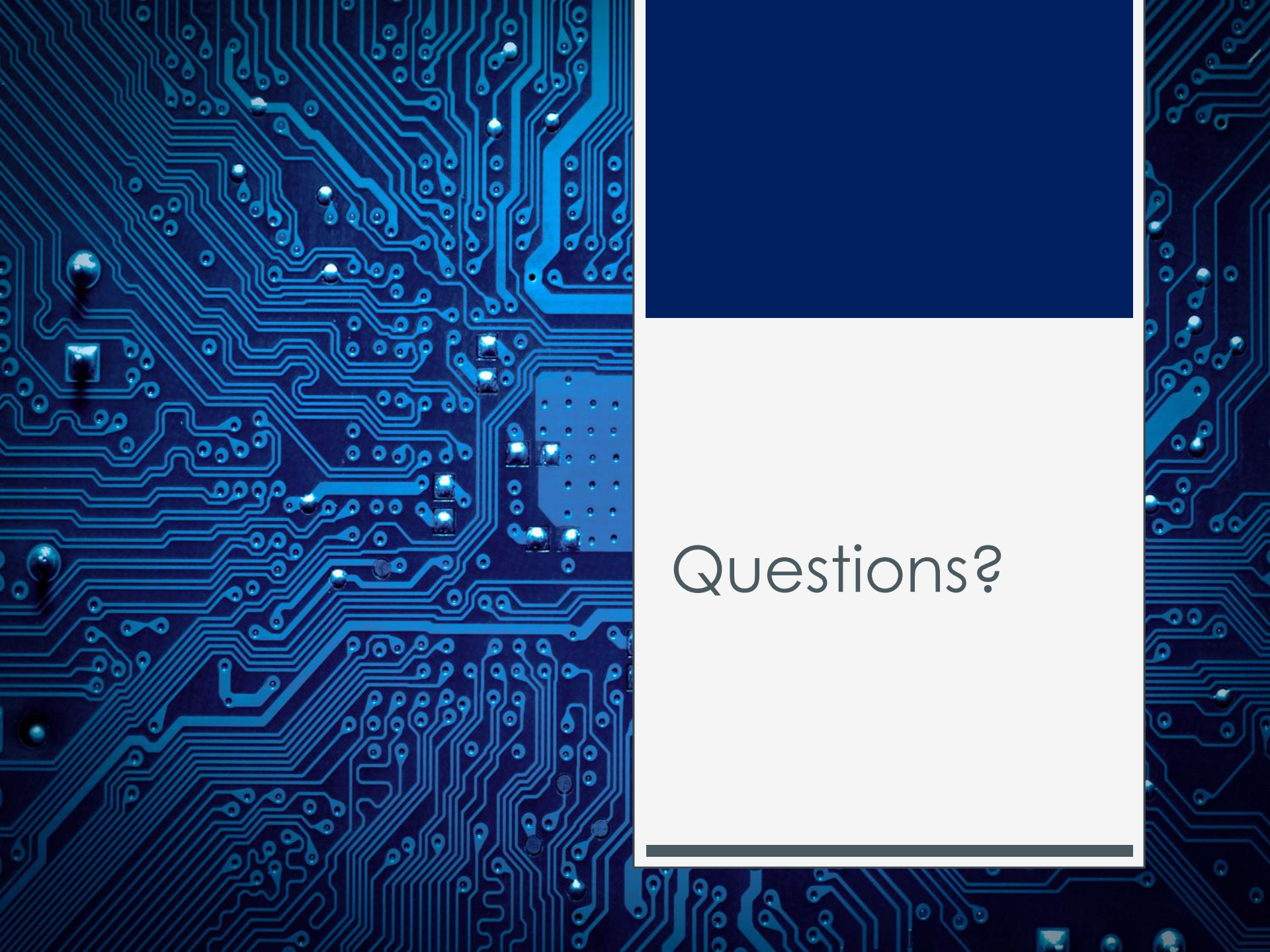
CVE-2021-26676

```
static gboolean listener_event(...)
{
        GDHCPClient *dhcp_client = user_data;
        struct sockaddr_in dst_addr = { 0 };
        - struct dhcp_packet packet;
        + struct dhcp_packet packet = { 0 };
        struct dhcpv6_packet *packet6 = NULL;
        …
```

Implications

Potentially impacts a large number of embedded systems, ranging from smart-home IoT devices to vehicles

- High risk of consumer safety impact
- Devices may not support easy/automatic upgrades
- Unclear of risk if attacker is not on the same subnet: some devices (*e.g.*, cars, mobile devices) may share a subnet with a large number of strangers


Questions?