Recap: Protection

• Protection
  – Prevent unintended/unauthorized accesses

• Protection domains
  – Class hierarchy: *root* can to everything a normal *
    user* can do + alpha

• Access control matrix
  – Domains (Users) \(\leftrightarrow\) Resources (Objects)
  – Resource oriented: Access control list
  – Domain oriented: Capability list
Recap: Security

• Stack and buffer overflow
  – Failure to check bounds on inputs, arguments
  – Write past arguments on the stack into the return address on stack
  – Unauthorized user or privilege escalation
Recap: Code with Buffer Overflow

```c
#define BUFFER_SIZE 256
int process_args(char *arg1)
{
    char buffer[BUFFER_SIZE];
    strcpy(buffer, arg1);
    ...
}

int main(int argc, char *argv[])
{
    process_args(argv[1]);
    ...
}
```

• What is wrong in this code?
Recap: The Attack: Buffer Overflow

Before executing `strcpy(buffer, arg1)` the crafted string containing the illegitimate code
Heartbleed Bug

• Synopsis
  – Due to a bug in OpenSSL (popular s/w for encrypted communication), web server’s internal memory can be dumped remotely
Heartbleed Bug

Image source: xkcd.com
Heartbleed Bug

Image source: xkcd.com
Heartbleed Bug

```c
struct {
    HeartbeatMessageType type;
    uint16 payload_length;
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];
} HeartbeatMessage

int tls1_process_heartbeat(SSL *s) {
    ...
    /* Read type and payload length first */
    hbtype = *p++;
    n2s(p, payload); // payload = recv_packet.payload_length
    pl = p;
    ...
    if (hbtype == TLS1_HB_REQUEST) {
        ...
        buffer = OPENSSL_malloc(1 + 2 + payload + padding);
        bp = buffer;
        memcpy(bp, pl, payload);
        r = ssl3_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);
        ...
    }
}
```
Shellshock Bug

• Synopsis
  – You can remotely execute arbitrary programs on a server running a web server by simply sending a specially crafted http request.
  – Example

    curl -H "User-Agent: () { ;; }; /bin/eject" http://example.com/

• The problem
  – Fail to check the validity of a function definition before executing it

For detailed explanation: security.stackexchange.com
Roadmap

- CPU management
- Memory management
- Disk management
- Network and security
- Virtual machine
Cloud Computing

Image Source: http://btstrategy.com/wp-new/2013/10/18/is-everything-really-going-to-the-cloud-advice-for-business-owners/
Cloud Computing

Amazon EC2

Google Compute Engine

Microsoft Azure
Virtual Machines

• Enabling technology of cloud computing
• Basic idea: Provide **machine** abstractions
Virtual Machines

• Benefits
  – Can run **multiple OSes**, each in its own virtual machine
  – Can **copy** a VM image and run it on a different machine
  – Can create a **snapshot** of the state and restore it later
  – Can create a **customized** VM with specific OS version and libraries to avoid version dependency problems
  – More **efficient** resource utilization is possible

• Downsides?
  – Overhead
  – Interference
History

• Late 1960s
  – IBM introduced first full VMM on mainframes

• Late 1990s
  – Xen was developed for Intel PCs

• Mid 2000s
  – Hardware support was introduced (e.g., Intel VT-x)
  – Widely adopted in data centers.
Topics

• How to implement VMMs?
• How to reduce overhead?