Advantages of an Adversarial Mindset

Wesley McGrew
wesley@mcgrewsecurity.com

Mississippi State University
Center for Computer Security Research
Critical Infrastructure Protection Center

Questions that drive development

Is this [x] efficient?

How can I optimize [x] for efficiency?

code

configuration

design

implementation

What are the weaknesses in [x]?

How could [x] be attacked?

What could be gain by attacking [x]?

Good design and careful implementation will only take you so far

For example...

"Every access to every object must be checked for authority"

Saltzer & Schroeder's principle of Complete Mediation

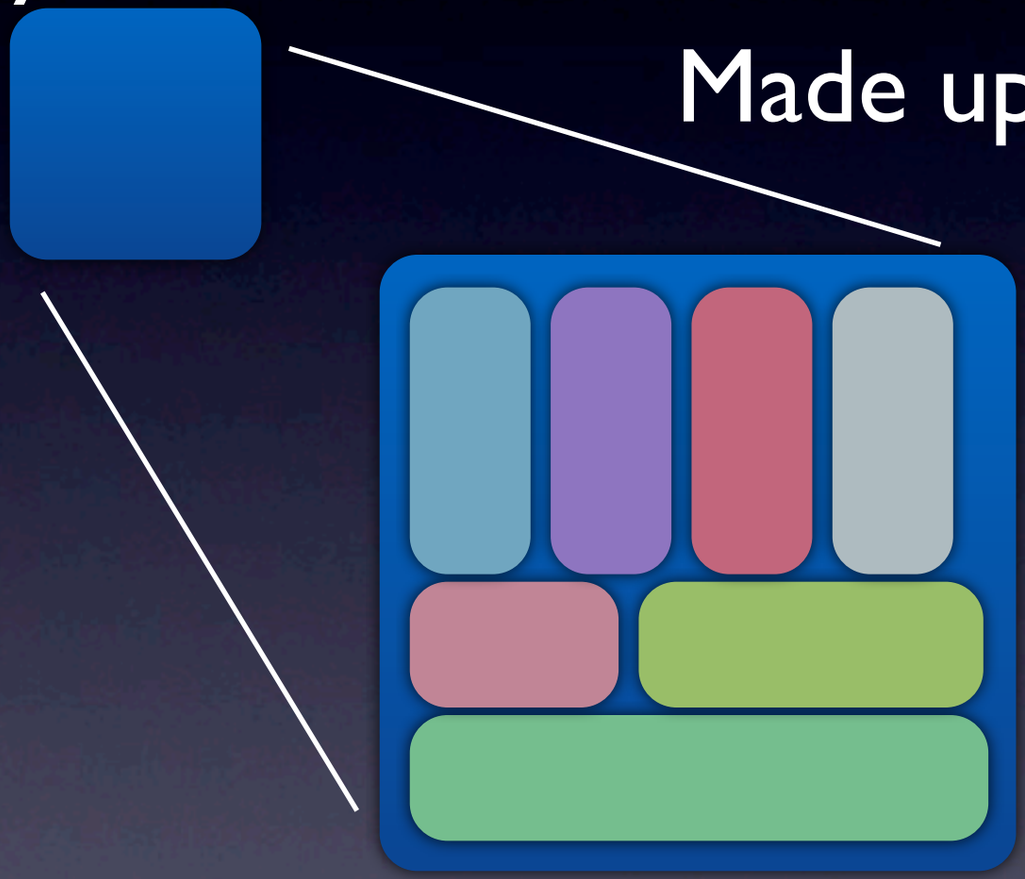
Good design and careful implementation will only take you so far

Systems

Good design and careful implementation will only take you so far

Systems

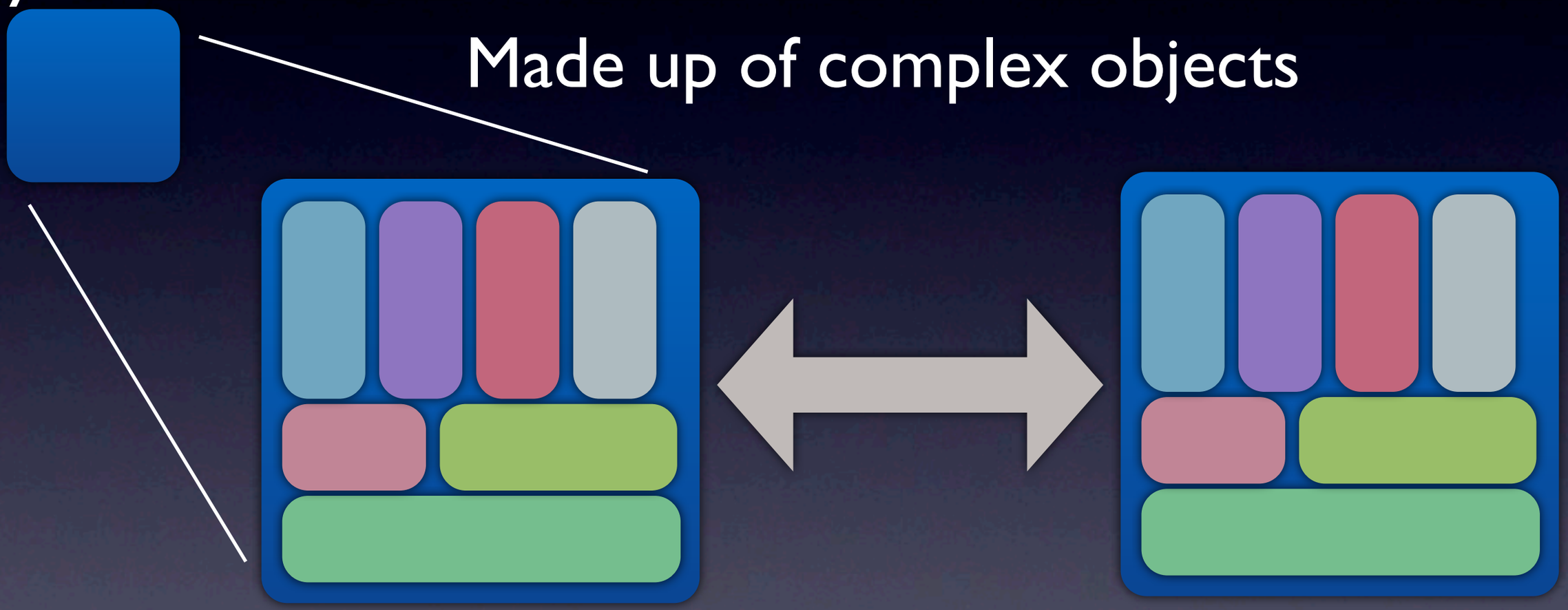
Made up of complex objects



Good design and careful implementation will only take you so far

Systems

Made up of complex objects



Communicating with other systems/objects

Good design and careful implementation will only take you so far

"Every access to every object must be checked for authority"

What consitutes an object?
Where does the check take place?
Access by humans, other objects?

The devil is in the details...

Good design and careful implementation will only take you so far

```
            wcscpy_s(previous_slash, (end_of_path-previous_slash)/2, p+2);
#else // XP
            wcscpy(previous_slash, &p[2]);
#endif

            if (ch == L'\0')
                return 1;

            current_slash = previous_slash;
            p = previous_slash;

            // find the slash before p

            // BUG: if previous_slash points to the beginning of the
            // string, we'll go beyond the start of the buffer
            //
            // example string: \a\..\

            q = p-1;

            while (*q != L'\\' && q != path)
                q--;

            if (*p == L'\\')
                previous_slash = q;
            else
                previous_slash = NULL;
        }
        else if (p[1] == L'\\') {
            // we have \.\ or ^.\
#ifdef VISTA
            if (current_slash != NULL) {
                if (current_slash >= end_of_path)
                    return 0;
                wcscpy_s(current_slash, (end_of_path-current_slash)/2, p+2);
                goto end_of_loop;
            }
            else {  // current_slash == NULL
                if (p >= end_of_path)
                    return 0;
                wcscpy_s(p, (end_of_path-p)/2, p+2);
                goto end_of_loop;
```

Implementation is even worse:

Subtle errors with serious consequences

Image Source: ms08-067 decompilation by Alexander Sotirov
http://www.phreedom.org/blog/2008/decompiling-ms08-067/

“Mainstream” IT

- Lots of experience and solid research on security
- Still has plenty of problems discovered and resolved as a result of an adversarial mindset
  - Local, Remote, Web vulnerabilities
  - Protocol issues: DNS, WEP, TKIP

Form

- Adversarial mindset takes the form of
  - Red-Teaming
  - Independent research/verification
  - Educating developers on attacks

Focus Areas

- Protocols
  - Boundaries between independent systems
  - Communications over distance
    - Attack Surface
- Interfaces: Human-Computer
  - Plug for the In-Depth Discussion on HMI :)

- It's not perfect:
  - Different attackers find different problems
- ...but it does fill in gaps
  - Mistakes made in design, implementation
  - Part of a process