Ryan Naraine, InternetNews, 2004.09.10:

“MS Premium customers get early security warnings

“Microsoft is giving premium customers advance notice of security bulletins, internetnews.com has learned.

“The company plans to release two security bulletins, one with a ‘critical’ rating, on Tuesday September 14, in order to plug holes in multiple software products, according to an advance notice sent to select customers. . . .

“The U.S. government’s Computer Emergency Readiness Team (US-CERT) has also been heavily criticized for providing advisories to paying customers ahead of coordinated public release.”

Assignment due 2004.08.25: read foreword and preface of textbook.

Assignment due 2004.08.27: read textbook Chapter 1 pages 1–14, up to “The Trinity of Trouble.”

Assignment due 2004.08.30: read the rest of Chapter 1.

Assignment due 2004.09.03: Read Gaim. `http://cr.yp.to/2004-494/gaim.html`

Assignment due 2004.09.08: read textbook Chapter 7 pages 277–308.

Assignment due 2004.09.15: read textbook Chapter 7 pages 309–336.

Last time: alphabetic machine language `hABCDhABCDYIQDYAQDYIIQDYAAQD` puts 40 43 41 46 into memory near `*sp`.

Can vary number of `A`'s and `I`'s to store any four bytes into memory.

Can repeat to store (e.g.) 40 bytes.

Can use `TYkaDA` (i.e., `*--sp = sp; cx = *sp++; sp = ((int*)cx)[17] * 65`) to change where these bytes are stored; in particular, to store them at the end of the alphabetic payload.

So a fairly long alphabetic payload can take control of the computer.

Complete alphabetic payload:

`AAAAAAA...`
   (no-ops)
`hABCDhABCD`
   (put known bytes on stack)
`YI...QDYI...QDYI...QDYI...QD`
   (change those bytes)
`LLLLL...TYkaDA`
   (point `sp` inside the `A`'s below)
`hABCDhABCDhABCD...`
   (put known bytes on stack)
`YI...QDYI...QDYI...QD...`
   (change those bytes)
`AAAAAAA...`
   (no-ops, changed before they're run)

Alternatives to alphabetic payload

1. Can put non-alphabetic payload somewhere else in memory.
Payload and smasher can be separate.

2. Check more carefully: maybe the input doesn't have to be alphabetic!
In particular, I do not think `isalpha` means what you think it means.

3. Often can take over using a pure smasher without a payload.
This also dodges NX "protection."
Will come back to this.

Example of technique 1 for Gaim:

`handle_receive_message` reads any amount of data into `msg` buffer. Some alphabetic bytes from `msg` are copied into `keyword` buffer.

Attacker strategy:
Put non-alphabetic payload into `msg`.
Have smasher jump into `msg`.

Smasher still has to be alphabetic.
If the legitimate return address was `0x08056fbe` then some working payload addresses are `0x41414141`, `0x7a7a7a7a`, `0x08004141`, etc.
Force memory allocation to make sure that payload is at a usable address.