Corruption

Dave Aitel

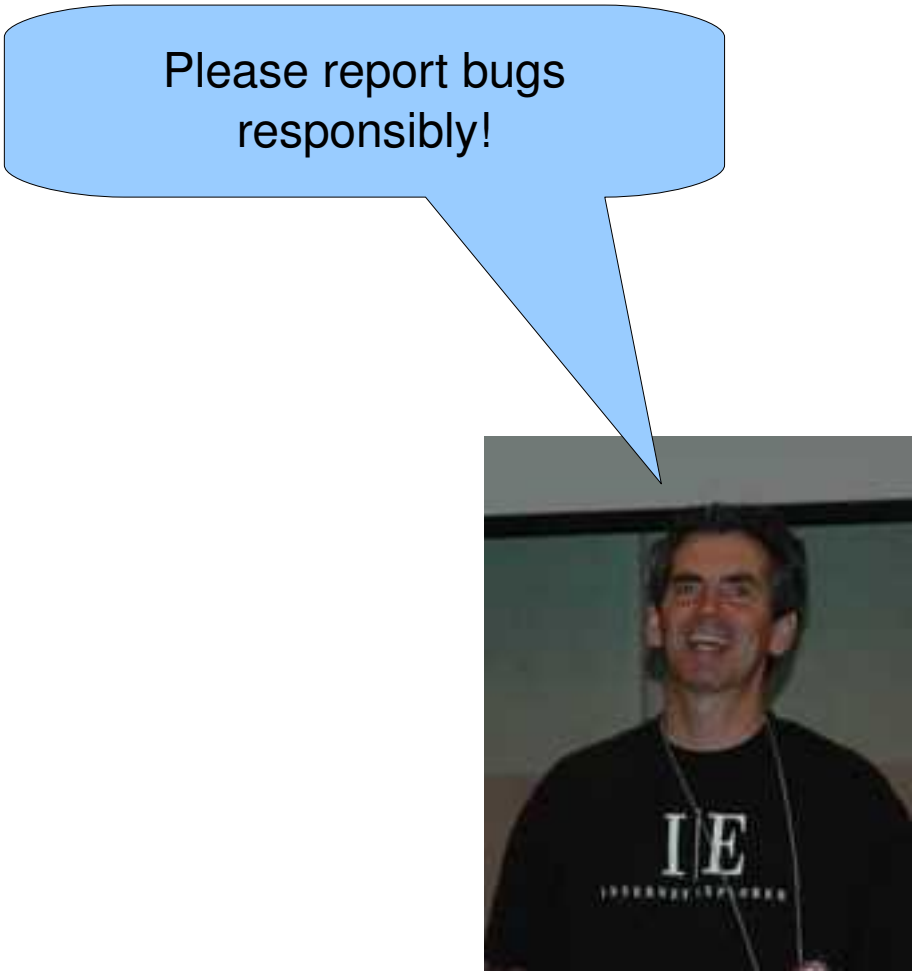
Immunity, Inc.

Thesis

- As the cost of writing memory corruption vulnerabilities goes up, we will no longer defend against them
- Unseen chaos will then ensue

“Public” knowledge drives our “threat model”

- Hence, the focus on:
  - the disclosure debate
  - vulnerability windows
  - advisories
  - patching
  - response times
  - etc.



Vendors stoke this fire

KNOWING YOU'RE SECURE

- "To date, no customers have reported security breaches" - CitectSCADA

30% of MS Advisories don't have this - why?

When this security bulletin was issued, had this vulnerability been publicly disclosed?

No. Microsoft received information about this vulnerability through responsible disclosure. Microsoft had not received any information to indicate that this vulnerability had been publicly disclosed when this security bulletin was originally issued.

The Bazaar

KNOWING YOU'RE SECURE

- Good offensive security research is being driven underground
  - It's too expensive to give away!
  - Not just a “Vulnerability Marketplace”:
    - Audit technologies
    - Bug classes
    - Exploit techniques

Show me the MONEY!

Why are memory corruption bugs so expensive

- **Heap/Stack cookies (/gS)**
- SafeSEH
- **ASLR**
- **DEP/NX/W^X/PAX**
- Process Isolation
- System call ACLs
- Automated code review programs
- **Managed languages**

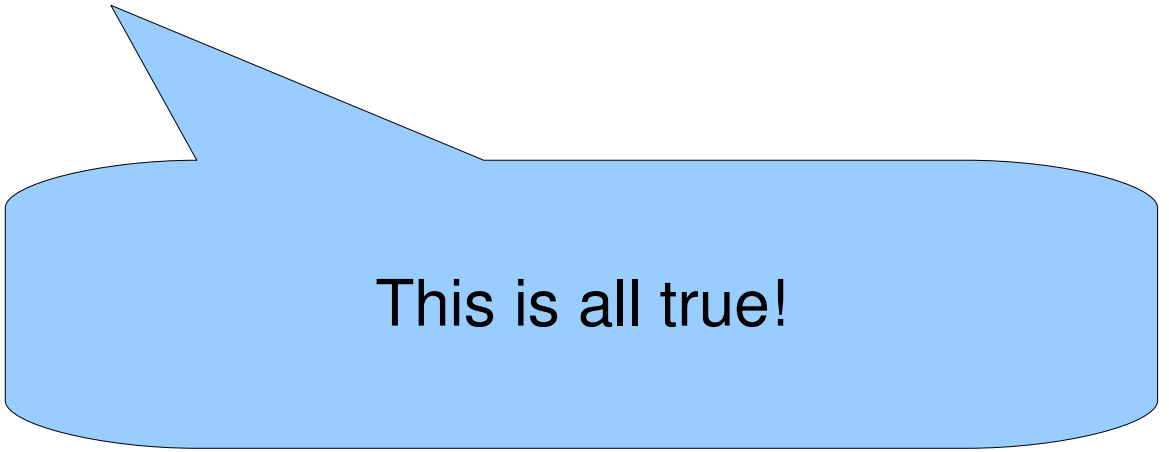
Security made the news

- Security built into development lifecycles
  - And compiler tools
- Security responses driving vendor differentiation
- Security being built into platforms

A gathering storm

KNOWING YOU'RE SECURE

- “But this doesn't affect me - I write web applications in Ruby on Rails”
- “There hasn't been a real remote overflow in IIS since version 5”
- “What part of **managed** language don't you understand?”



- Vulnerability research is so expensive it cannot be funded out of your marketing budget anymore
- Not only are bugs expensive but the techniques for reliably exploiting bugs becomes expensive
  - You no longer know if you are really at risk!

The market is reacting

- The memory corruption problem is “solved”
- The worm problem is “solved”
  - Hence, the slow takeup of IPS – it's just not worth the pain!
- Microsoft Exploitability Index
  - Q: What are you going to do when for months on end everything is “pretty much not exploitable”
  - A: Stop patching
  - A: Stop investing in security

Heap overflows make a good case study

KNOWING YOU'RE SECURE

- Heap overflows were never a problem
  - No worms, no problem!
  - No exploits on milw0rm, no problem!
  - Essentially, if it is not freely available, it does not exist

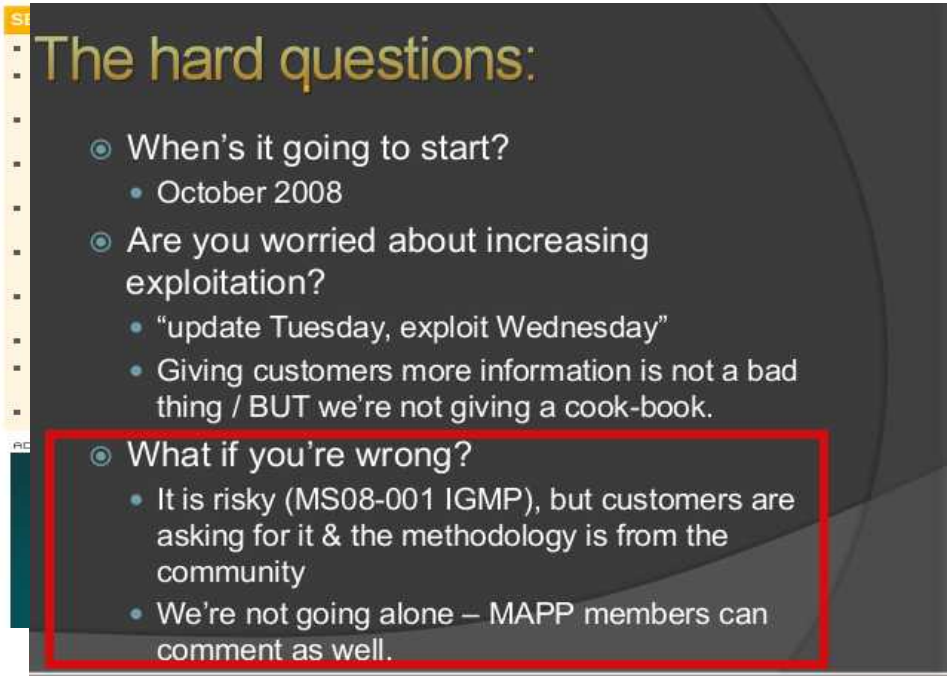
Mitigating factors aren't

KNOWING YOU'RE SECURE

Microsoft Security Research Team Blog

> The first question you are probably asking is "How likely is exploitation of this issue?"
>
> Even though this bulletin is rated Critical for XP and Vista (the bulletin describes mitigating factors that lower the severity on Windows Server 2003) there are a number of factors that make exploitation of this issue difficult and unlikely in real-world conditions.

Kostya



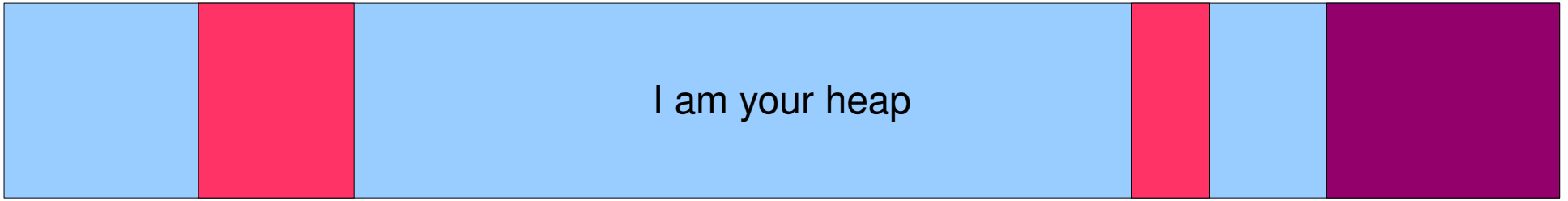
Microsoft BlackHat Vegas 2008

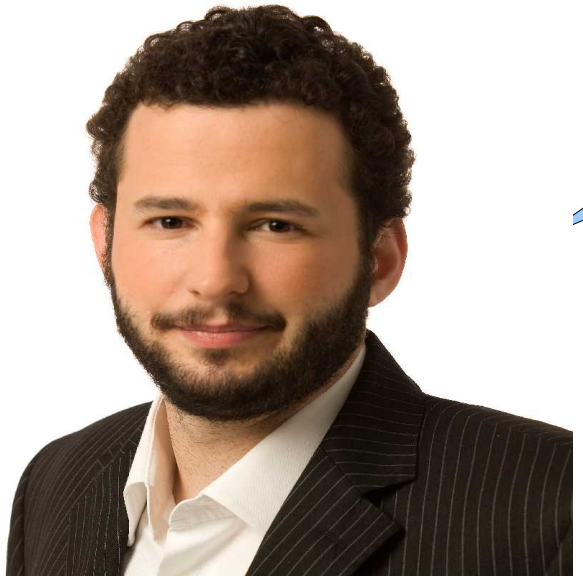
Answer: Yes

- Default remote Ring0 on Windows XP SP2
- Patch was triaged by lots of customers
  - Why? No free exploits.
- See your local CANVAS install for how Kostya did it.

How does this magical code work?!?

KNOWING YOU'RE SECURE





You just fill the little red holes to make it all predictable

Heap Overflow Conclusions

KNOWING YOU'RE SECURE

- Heap overflows can be made MORE reliable than other vulnerabilities
  - But it takes 1-6 man months to write the exploit
- Q: Would you patch that Bob's Server heap overflow?
  - No free exploits
  - You installed Bob on Windows 2003 SP2 with DEP and heap cookies, etc.
  - Bob's is supposed to be firewalled anyways.
- A: No, instead, you got owned.

Vulnerability is a float

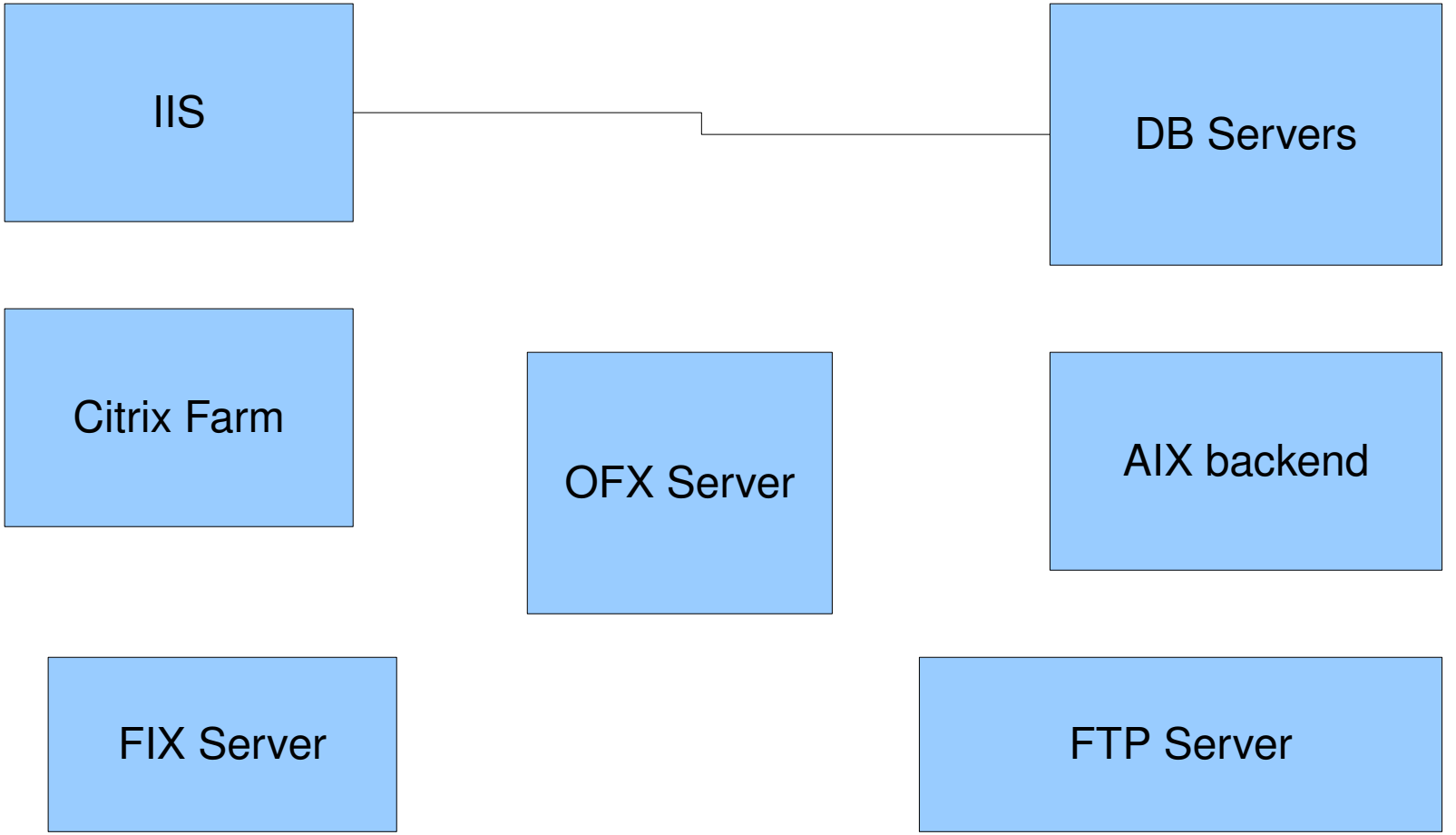
KNOWING YOU'RE SECURE

- Assuming 1% of the time my exploit works
  - Why would you protect yourself?

Making things harder is good for professional attackers

Your web app: More than ports 80/443

KNOWING YOU'RE SECURE

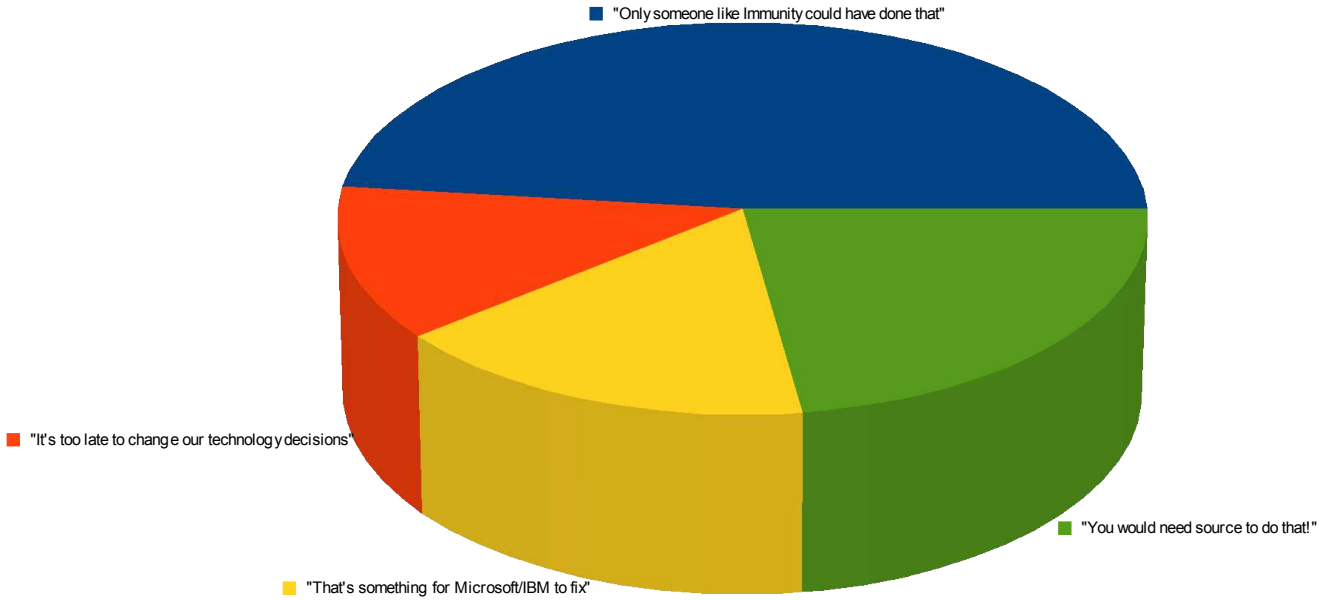


What Immunity Sees

- Overflows in ISAPIS we can exploit remotely without ever having the ISAPI
- Overflows in ISAPIS we can exploit remotely after downloading that eval copy on their website
- Overflows in the FTP server you use to transfer large files
- Overflows in the OFX library you use to connect your application to MS Money/Quickbooks

Most common excuses

KNOWING YOU'RE SECURE



- "Only someone like Immunity could have done that"
- "It's too late to change our technology decisions"
- "That's something for Microsoft/IBM to fix"
- "You would need source to do that!"

The Cloud City

KNOWING YOU'RE SECURE

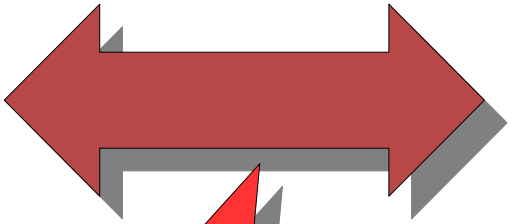
- Virtualization Clouds
  - You don't just own one machine – you own **everything** (data/applications) in the cloud
- C.F.: Google App Engine integer overflow

If you kill me, I'll just become useful for hypervisor attacks.

Penetration testing is corrolation

KNOWING YOU'RE SECURE

Only Publishers can
Edit content on my site

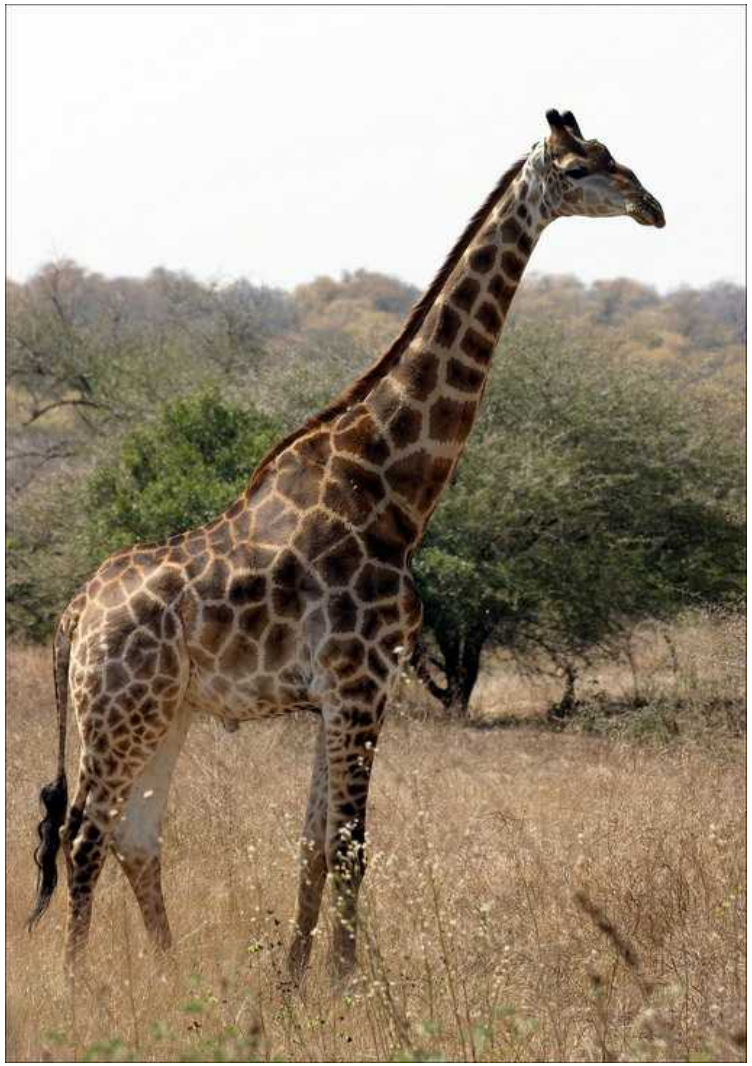


If member in PUBLISHER
Group then allowed
File types += [“.html”]

Penetration lies in the
Difference between
Technical and Business
measures

Conclusion and Questions

KNOWING YOU'RE SECURE



- Don't underestimate the reach of your attackers
- Questions?