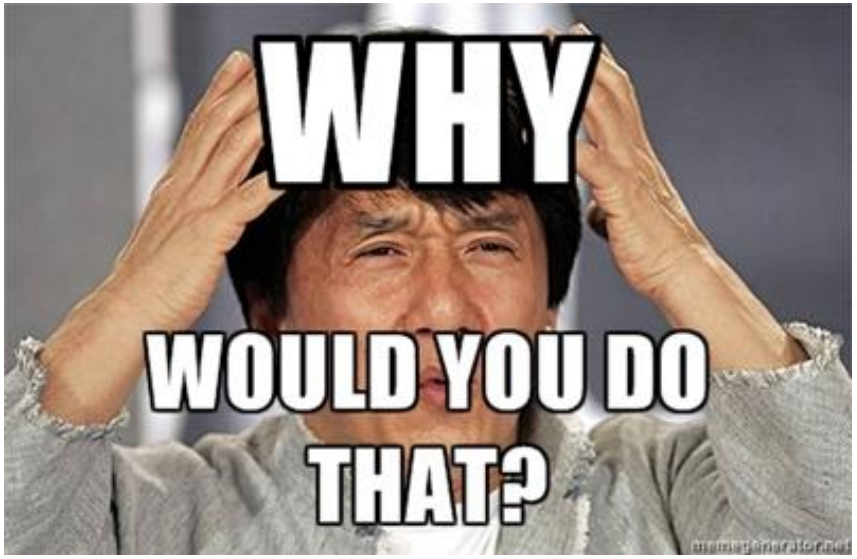
Attackers, lies and you

OWASP-Italy Day2012
Rome, 23° November 2012

Copyright © 2008 - The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License.

The OWASP Foundation
http://www.owasp.org

Are we doing things properly in security?


WHY
WOULD YOU DO
THAT?
memegenerator.net

How does offense work?

- Attacker’s mindset
- Gaining access
- Keeping access/stealing data

We currently fail badly at the understanding the first two

First problem: spot the difference

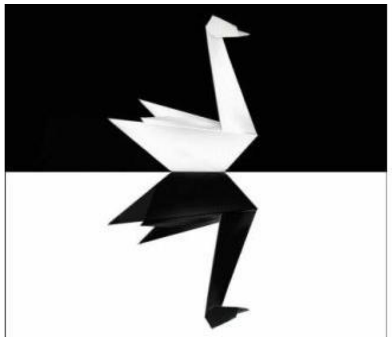
Black swans? What's that?

- A very interesting research result that is unlikely to happen in real life

Why black swans exist?

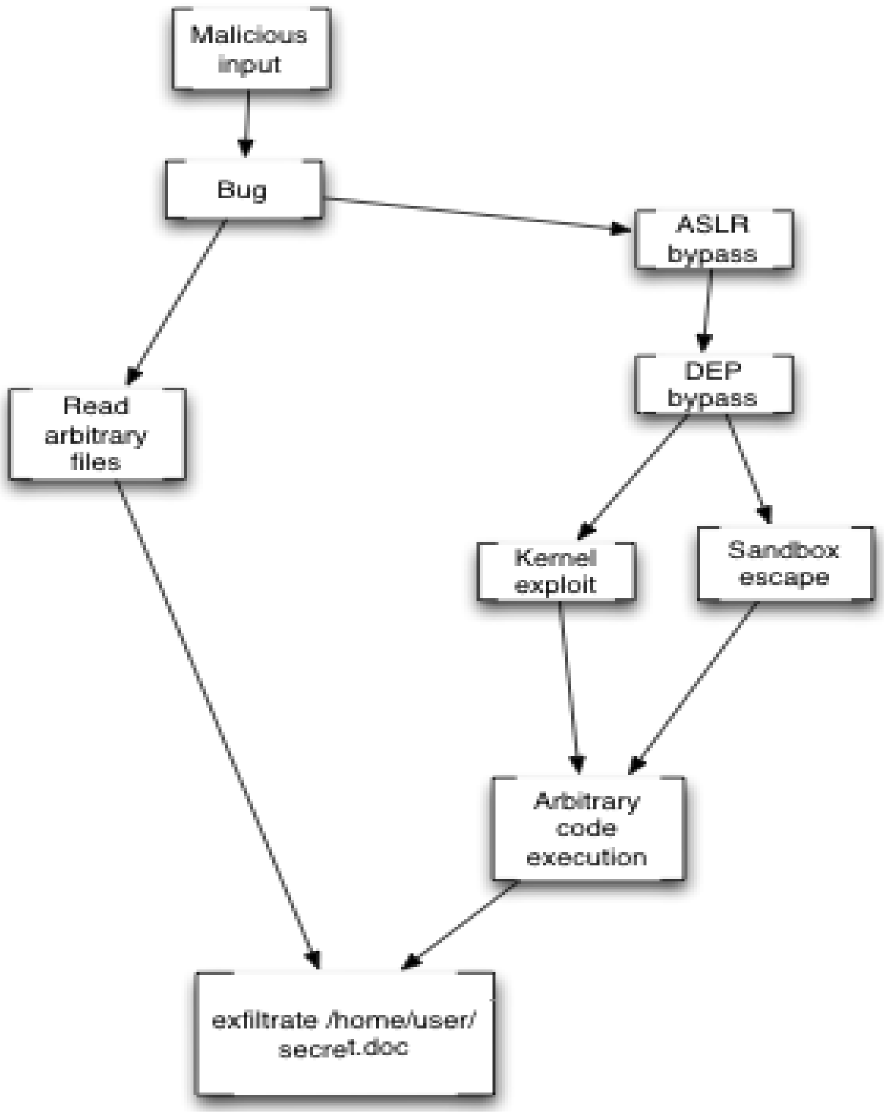
- "Machines can remain vulnerable longer than you can remain sane"
- The security community is fixated on persistance
- A lot of people forget the mantra: "whoever scores is right"
- Technical elegance is highly valued

Black swans and attacker math

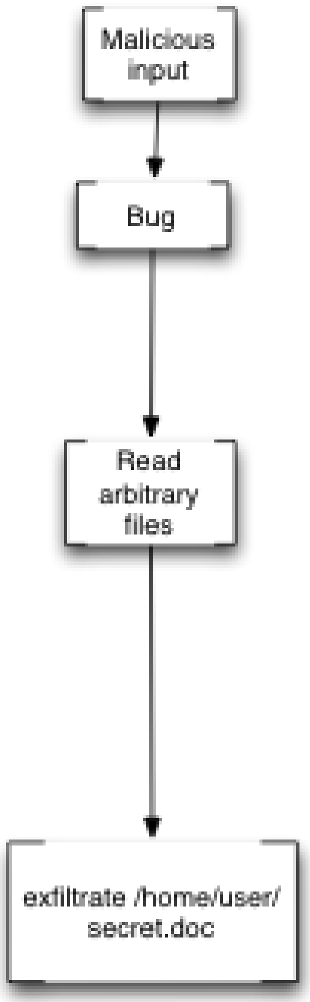


- Attackers are resource-constrained: "The Exploit Intelligence Project" (Dan Guido)
- Attackers are rational human beings
- **Attackers will take a given exploitation path IFF no cheaper paths are available**

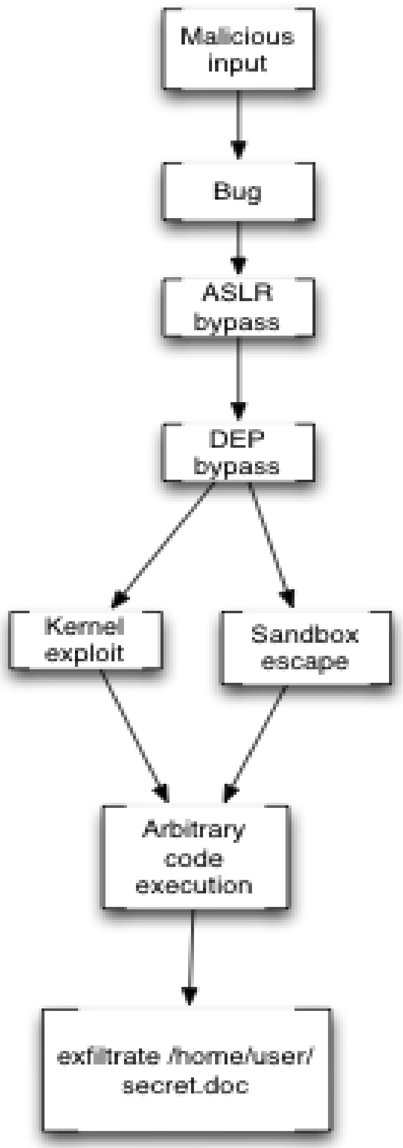
Exploitation paths



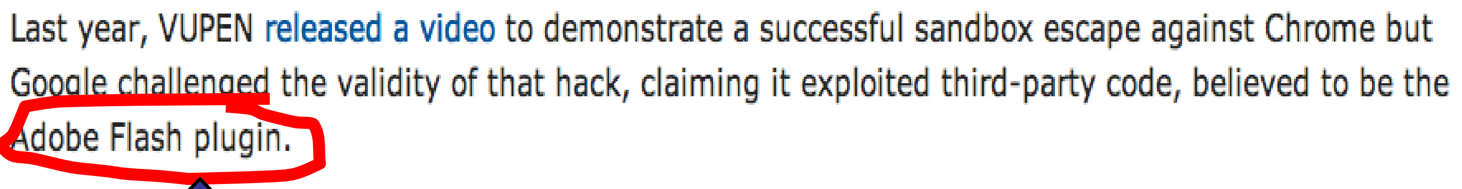
A rational attacker



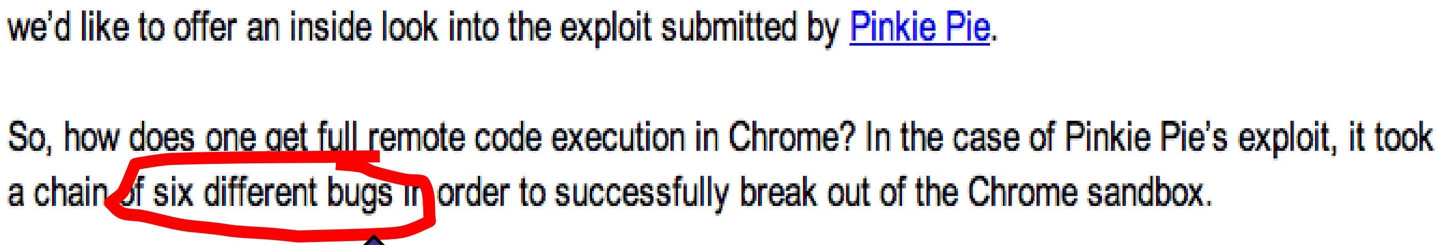
A black swan


Malicious input
Bug
ASLR bypass
DEP bypass
Kernel exploit
Sandbox escape
Arbitrary code execution
exfiltrate /home/user/secret.doc

Practical example


Last year, VUPEN released a video to demonstrate a successful sandbox escape against Chrome but Google challenged the validity of that hack, claiming it exploited third-party code, believed to be the Adobe Flash plugin.

A rational attacker


we'd like to offer an inside look into the exploit submitted by Pinkie Pie.

So, how does one get full remote code execution in Chrome? In the case of Pinkie Pie's exploit, it took a chain of six different bugs in order to successfully break out of the Chrome sandbox.

A black swan (AKA: are you nuts?)

So…

VS

Unless..

- The ROI on a black swan is higher, for some definition of “return”
- Flame md5 collision attack comes to mind
- Therefore our graph is weighted

Weight function

- That's very hard to calculate in the general case
- Some examples in "Attacker Math 101" (Dino Dai Zovi)
- A bit out of scope here
- But we can usually draw a line easily

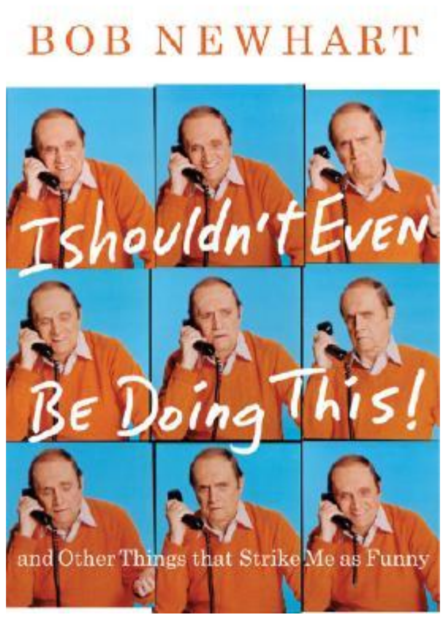
What if two paths are equally cost effective?

Gaining access..

- It's all about programming a "weird machine" (Sergey Bratus et al.)

The weird machine

- In short: “a machine that executes an unexpected series of instructions”


BOB NEWHART
I Shouldn't Even Be Doing This!
and Other Things that Strike Me as Funny

By examples

- ROP
- JIT Spraying – Dion Blazakis
- SpiderMonkey Bytecode Hijacking – Thomas Dullien
- JIT code hijacking – Chris Rohlf and Yan Ivnitskiy
- …

Exploitation

- Exploitation is setting up, instantiating, and programming the weird machine - Thomas Dullien

Controlling the machine

- You need write primitives
- You need infoleaks/memleaks
- For both you need some degree of control over the application.
- It’s either pure data or you can directly influence the application state (eg: through an interpreter of some kind)

Controlling the machine 2

- Just data = most likely you need multiple bugs (infoleak, write primitive, etc)
- Through interpreter = most likely you just need one (see comex jailbreaks for example)

Me no like exploits

- This process is challenged in a few ways:
  - Negate the initialization (fix bugs)
  - Make the setup hard (heap/stack mitigations, ASLR)
  - Make it hard to put together ‘weird instructions’ (ASLR, DEP, JIT hardening)
  - Reduce/Neutralize the effects of a running weird machine (sandboxing, code signing)
  - More to come in the future..

Get to the data/persistence

- How hard is to get your code on a target?
- How far away is the data you care for from you?

For future reference..

- So here’s the thing:
- In a few years everything an attacker cares for will be inside a browser/mobile app

- Do sandboxes help with that? *NO*

Let's wrap up

- Attacker's mindset: take the most cost-effective path
- When it comes to exploitation the most cost-effective path is:
- 1) As close as possible to your data
- 2) Reduces as much as possible the need for multiple bugs/exploits
- 3) Reduces maintenance cost

Conclusion

- If you don’t know *what* you’re protecting, you’ll fail
- Likewise if you don’t know what you’re protecting *against*, you’ll fail
- You don’t need a horde of code auditors & policy people, you need a CEO (chief exploitation officer)