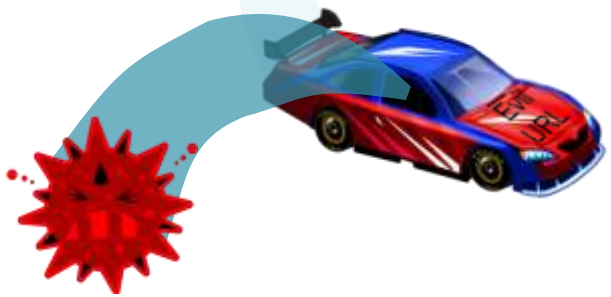
Detecting drive-by-downloads using human behavior patterns

Alex Crowell, Rutgers University
Computer Science and Mathematics

Advisor: Prof. Danfeng Yao,
Computer Science Department

What are drive-by-downloads?

- *drive-by-download* - when visiting a URL causes malware to be installed on a computer
- This is a 'pull-based' attack
- Made possible by:
  - Web server security flaws
  - Browser security flaws
  - Social engineering



Video taken from: http://www.watchguard.com/education/video/play.asp?vid=dbd-cubecast

How are they spread?

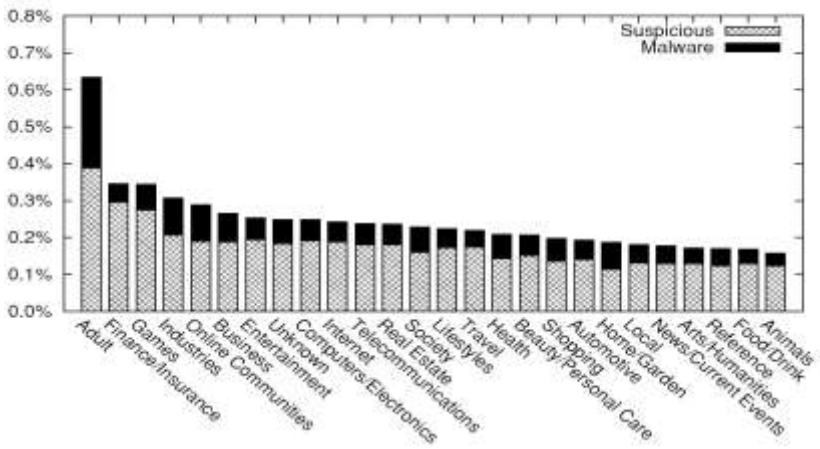
- There are many ways to put a drive-by-download exploit online:
  - Launch your own website
  - Break into someone else's website
  - Post user contributed content to a website
  - Use third-party online advertising
  - Use a third-party widget (i.e. a traffic counter)

From: *Provos N., McNamee, D., Mavrommatis P., Wang, K., and Modadugu, N.* The Ghost in the Browser: Analysis of Web-based Malware. In Proceedings of the first USENIX workshop on hot topics in Botnets (HotBots'07). (April 2007)

How prevalent are they?

- Search of pages indexed by Google found over 3 million unique malicious URLs executing drive-by-downloads
- Distribution of malicious sites not significantly skewed towards 'gray content'

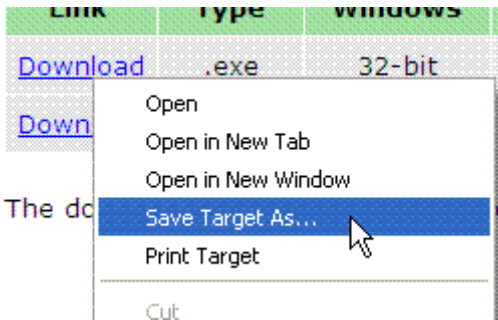
Data collection period	Jan – Oct 2007
Total URLs checked in-depth	66,534,330
Unique suspicious landing URLs	3,385,889
Unique malicious landing URLs	3,417,590
Unique malicious landing sites	181,699
Unique distribution sites	9,340



From: *Provos N., Mavrommatis P., Rajab M. A., and Monrose, F.* All Your iFRAMEs Point to Us. In Proceedings of the USENIX Security Symposium (July 2008)

Our Approach

- Most approaches to detecting drive-by-downloads focus only on the computer itself
- A lot can be seen by considering the user's input as well
  - User usually clicks a link or 'Save Target As…' before downloading an executable
- We can clearly make use of this to help create a much stronger detection method



Our Approach (continued…)

- Taking this approach to detect drive-by-downloads, we will:
  - Check for user clicks and associate them with downloads recorded in file system data
  - If we cannot find user input to associate with a download, consider it suspicious
  - Ensure the user input is not faked by the attacker

First Steps

- Will be implemented on Windows
  - Popular; most drive-by-downloads on Windows
  - Has convenient tool for monitoring file system events (`FileMon` or `ProcMon`)
  - Closed source; parts of API unavailable
- We use the Firefox extension `tlogger` to handle user input
- Write a program that takes the file system data from `FileMon` and user action data from `tlogger` and flags any 'suspicious' downloads

Plans for Improvement

- Authenticating the user input
  - Trusted Platform Module (TPM) can be used
- Making input logger platform independent
- Test on both real-world techniques and synthesized ones
- Improve performance accuracy
  - Find a good tolerance for the time between user click and start of download