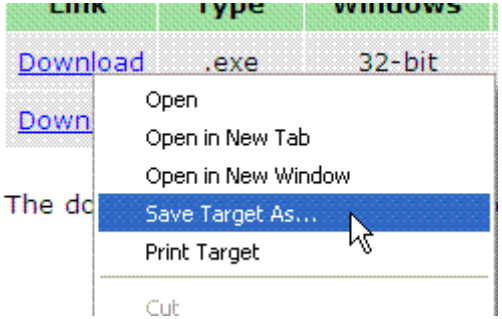
Detecting drive-by-downloads using human behavior patterns

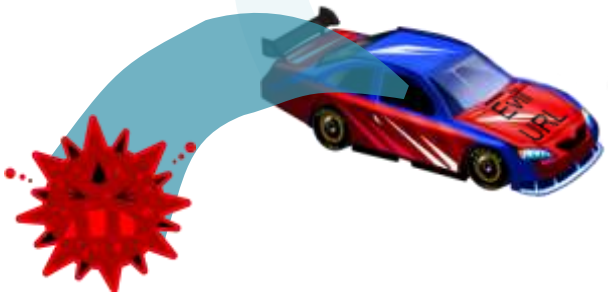
Alex Crowell, Rutgers University
Computer Science and Mathematics

Advisor: Prof. Danfeng Yao,
Computer Science Department

Drive-by-download attacks and detection

- *drive-by-download* - when visiting a URL causes malware to be installed on a computer
- Most approaches to detecting drive-by-downloads focus only on server-side solutions or browser security
- *We can use the user's input to validate each download when it occurs*





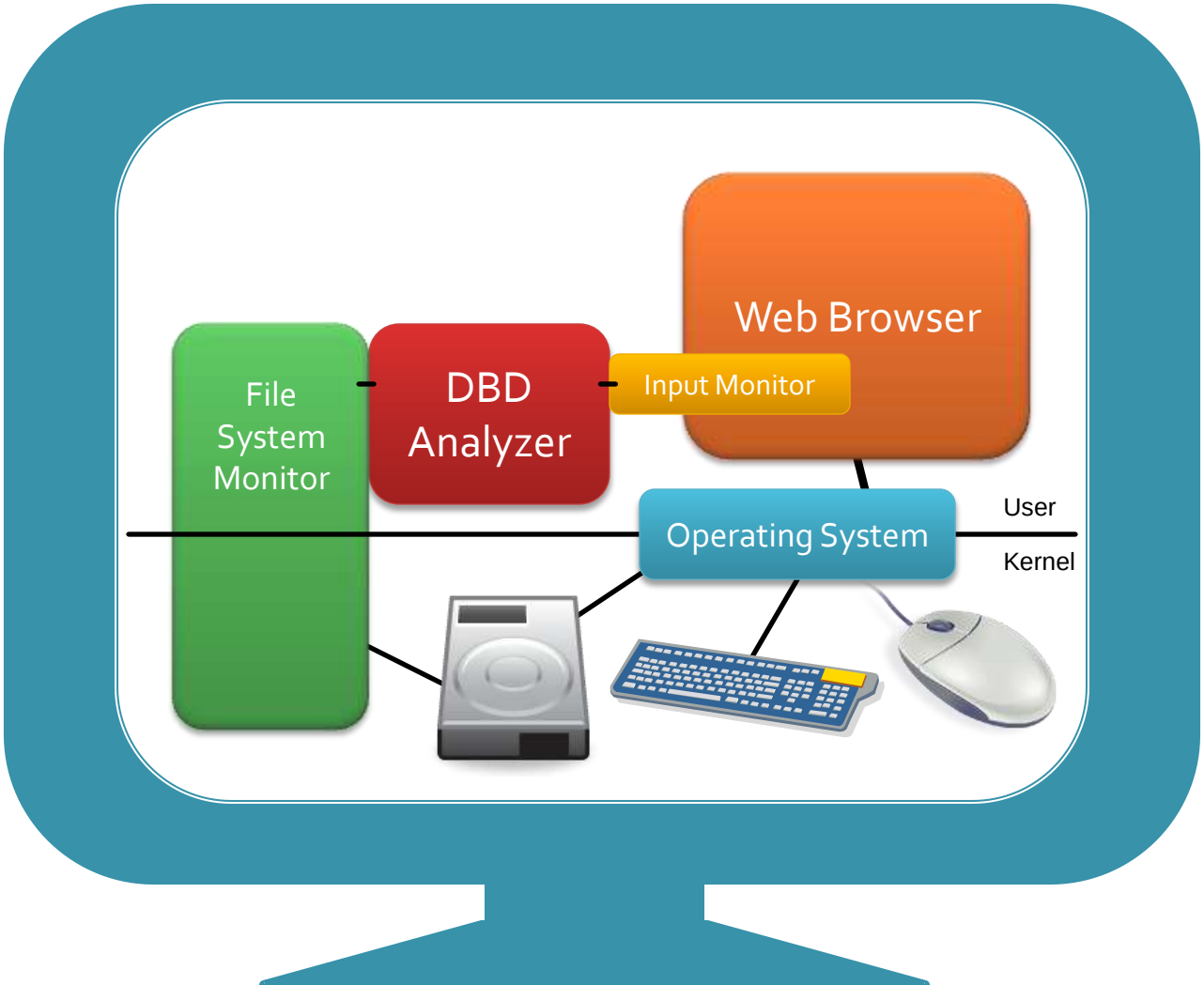
Our approach

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

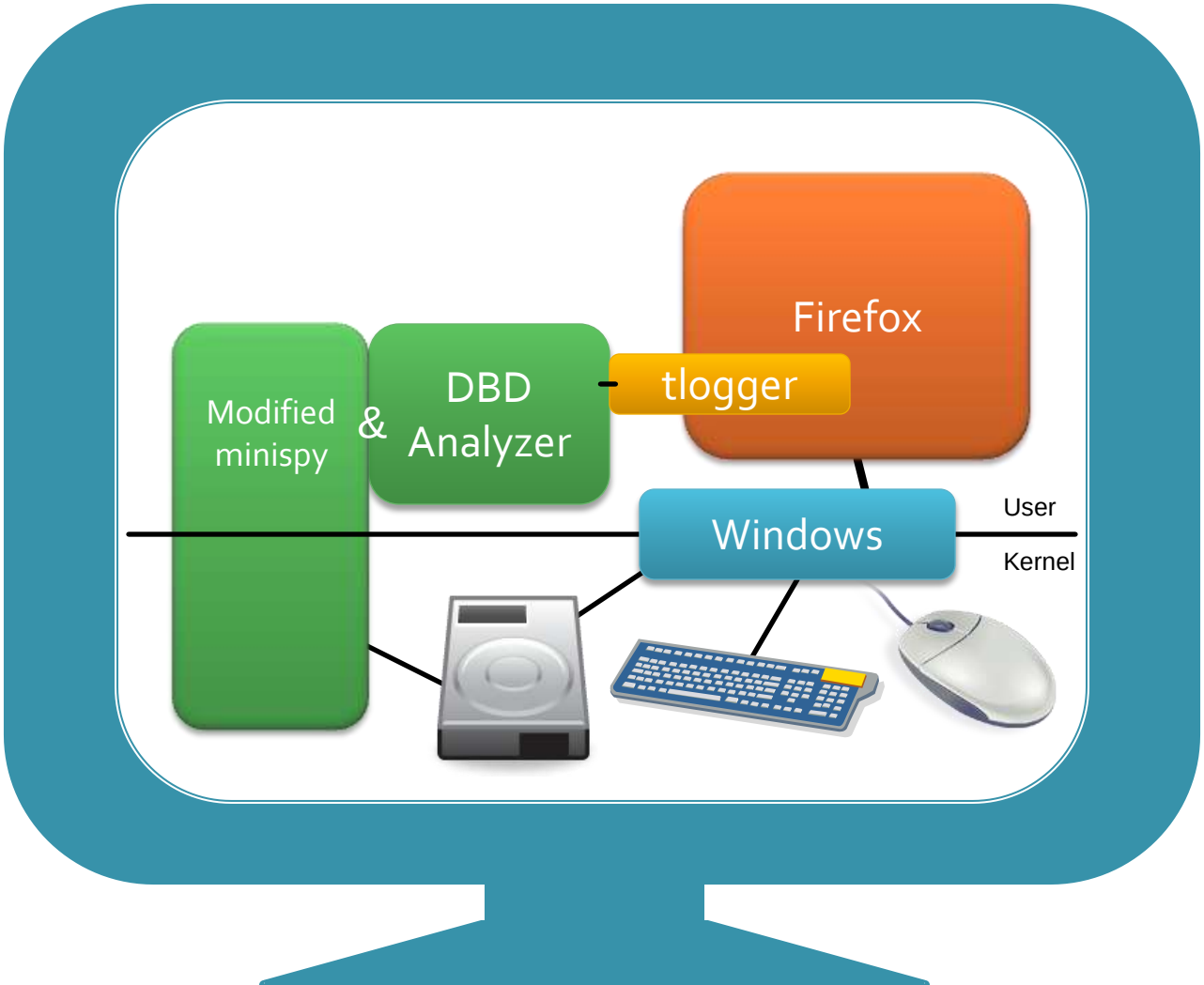
Some Problems

- ProcMon doesn't save its data in real-time
  - minispy is a sample program supplied with the Windows Driver Kit that works just like ProcMon
- Some websites redirect through a chain of pages before reaching the download (e.g. download.com)
- In practice, there is a long lag time between a link click and file creation
  - It may not be possible to track the user clicking the 'Save File' button

Our General Architecture



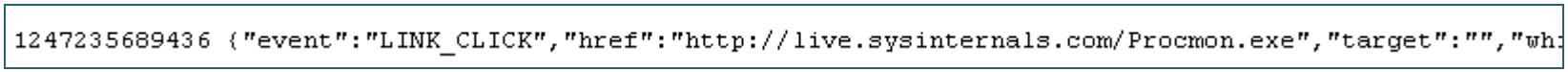
Our Implementation



How the Analyzer works

- Tracks, using ProcMon/minispy, for the creation of files by Firefox

- When a file is created by Firefox, the analyzer searches through the entries in the tlogger data file for a corresponding user input



- As long as the input occurred within a time limit from the file creation, it is a valid download

```
range(0, len(tlogger_data) - 1):
clicktime = float(tlogger_data[i].split(None, 1)[0]) / 1000
clicklinkname = json.loads(tlogger_data[i].split(None, 1)[1])["href"].split('/')[-

if abs(entrytime - clicktime) <= TOLERANCE and entryfilename == clicklinkname:
        foundmatch = True
        del tlogger_data[i]
        break
```

Some Major Assumptions

- Windows is not compromised
- Firefox and tlogger are not compromised
- No file overwrites occur in any file downloads
- File creation occurs in legitimate downloads within a short time of the user input that initiated it

Plans for Evaluation

- Want to test:
  - Effectiveness of solution
    - Particularly false positive/negative rates
  - Performance and Usability
    - Overhead on system
    - Whether it is obtrusive to the user
- Will do both:
  - User study
  - Partially automated testing

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
- Find better input to track
  - Find some way to track the user's clicking the 'Save File' button

Acknowledgements

- Thanks to:
  - Mentor Danfeng Yao
  - Qiang Ma
  - DIMACS Faculty

Questions