



Using Replicated Execution for a More Secure and Reliable Browser

Hui Xue, Nathan Dautenhahn, Sam King
University of Illinois at Urbana Champaign

Browsers: Important App Platforms

- Email, Banking, Shopping, Social network...

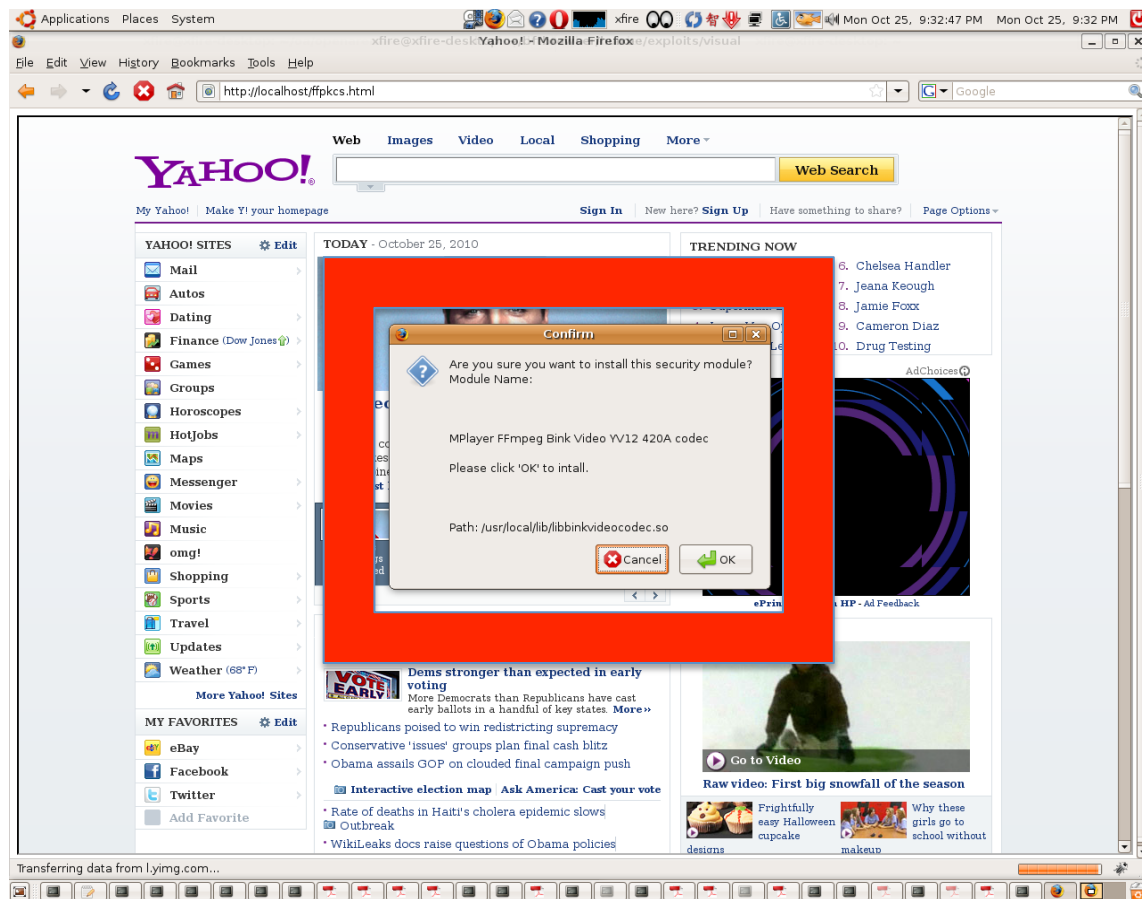


Browsers Are Not Safe

- Browsers are plagued with vulnerabilities
 - Internet Explorer: 59 new vulnerabilities in 2010
 - Firefox: 100 new vulnerabilities in 2010
 - Safari: 119 new vulnerabilities in 2010
 - Chrome: 191 new vulnerabilities in 2010
- Attackers target browsers
 - Studies from Washington, Microsoft, and Google

Firefox Browser Exploit Example

- Firefox 3.0.x malicious popup by CVE-2009-3076



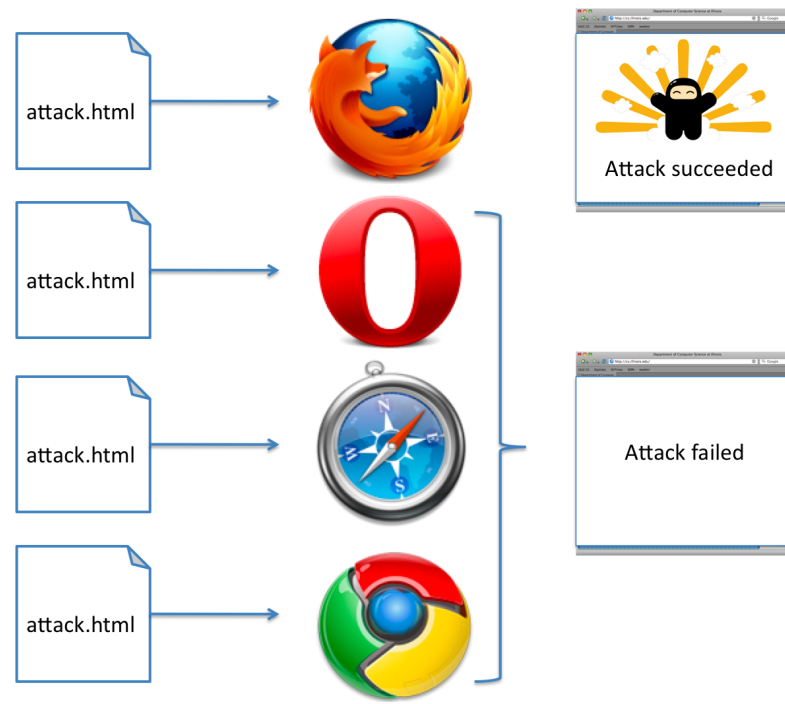
Opera: Exploit Fails

- Opera shows no popup

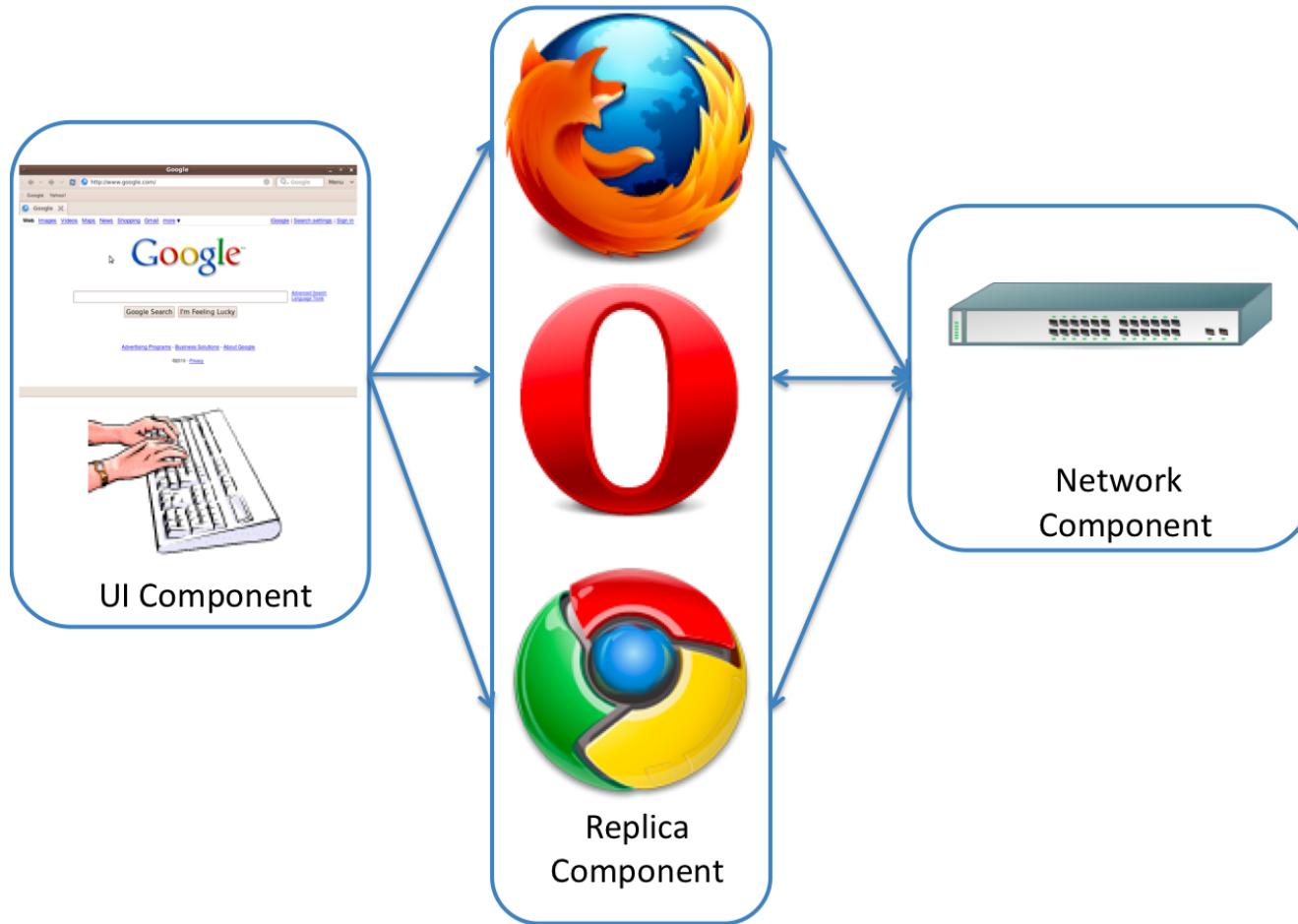


Reason: Browser Specific Vulnerabilities

- Different browsers different code bases
- The same bug often only in one browser



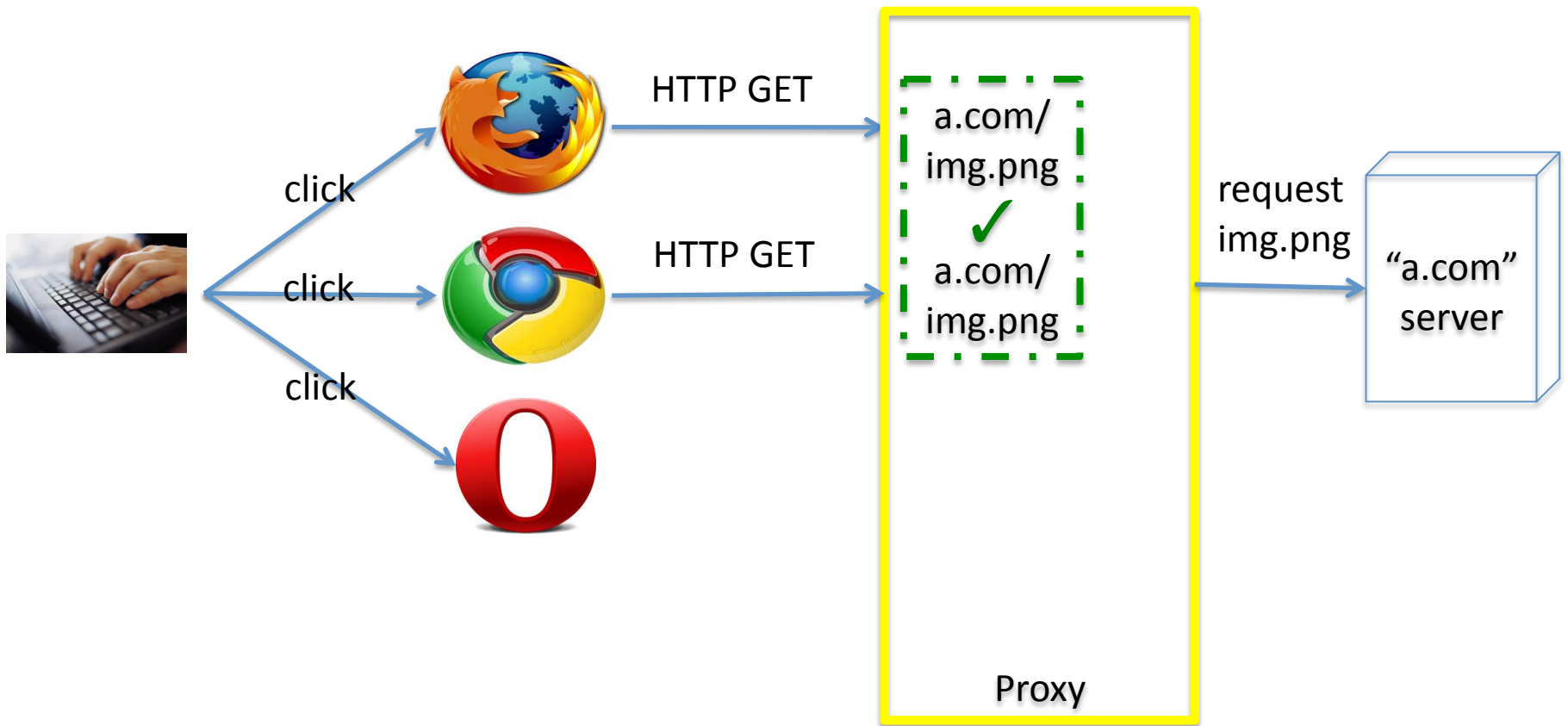
Cocktail: Mixing Browsers For Better Security



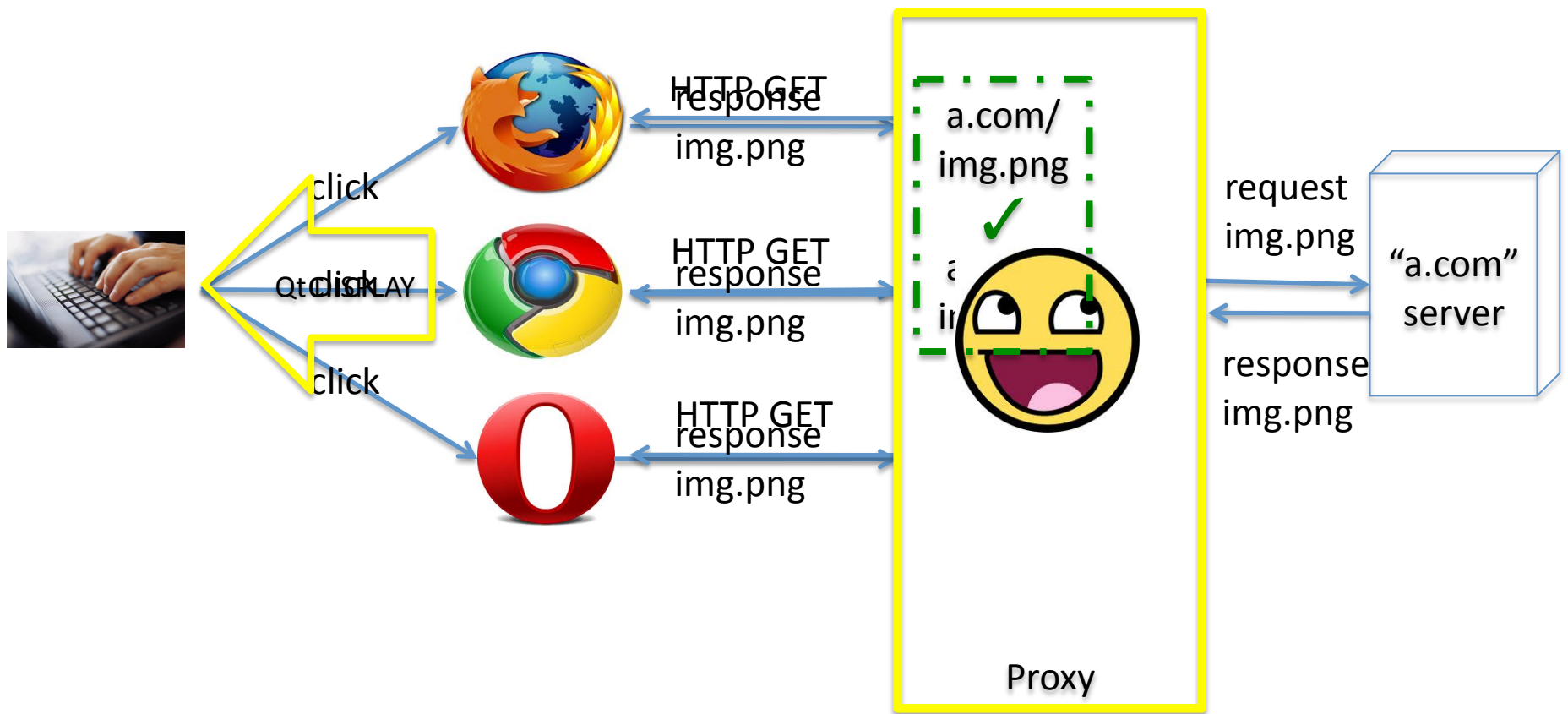
Overview

- Example
- Observation
- Design
 - Non-determinism
- Implementation
- Evaluation
- Conclusion

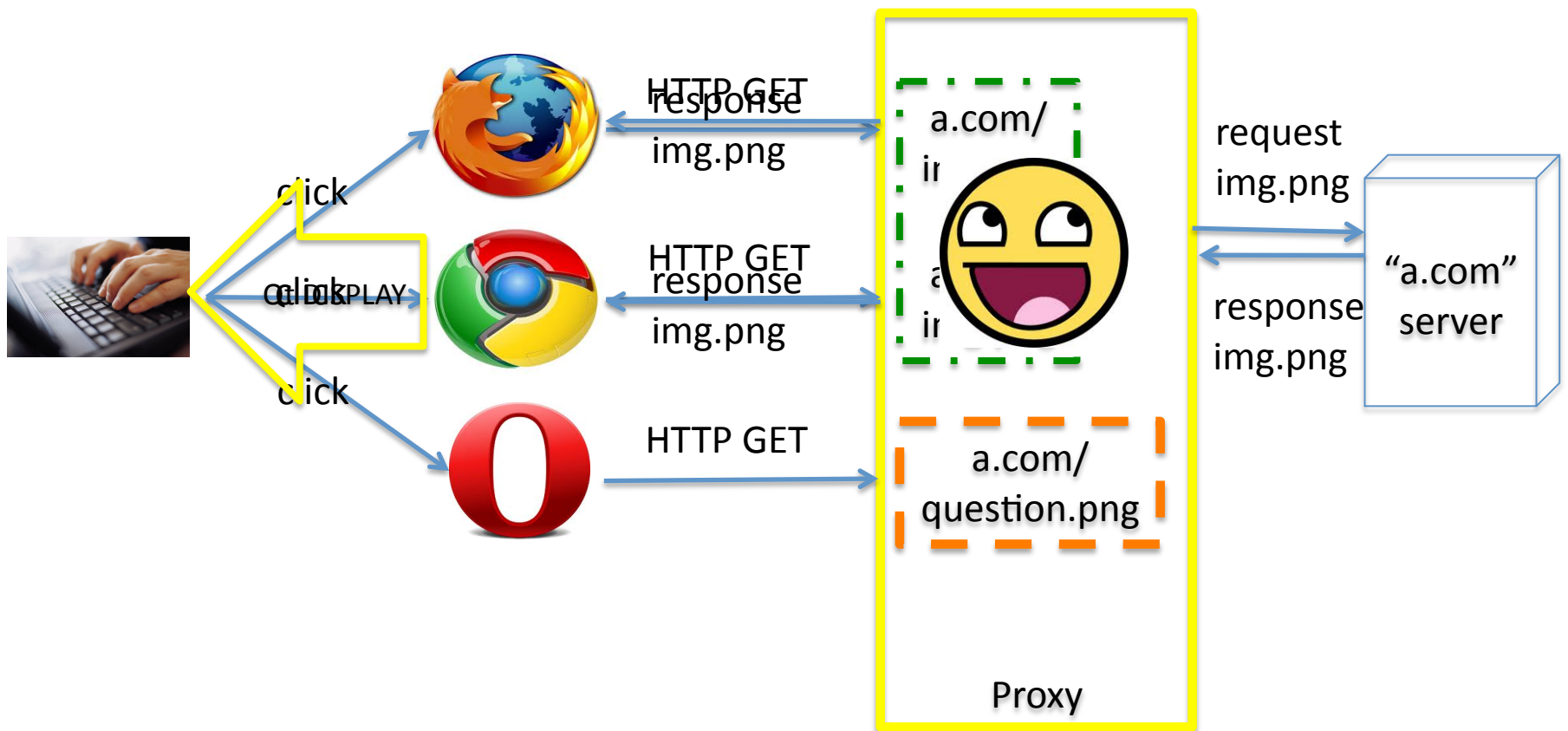
How Cocktail Works: Example



How Cocktail Works: Example



Withstanding False Positive/Attack



Observation: Opportunistic N-Version Programming



Different code base

+



DOM

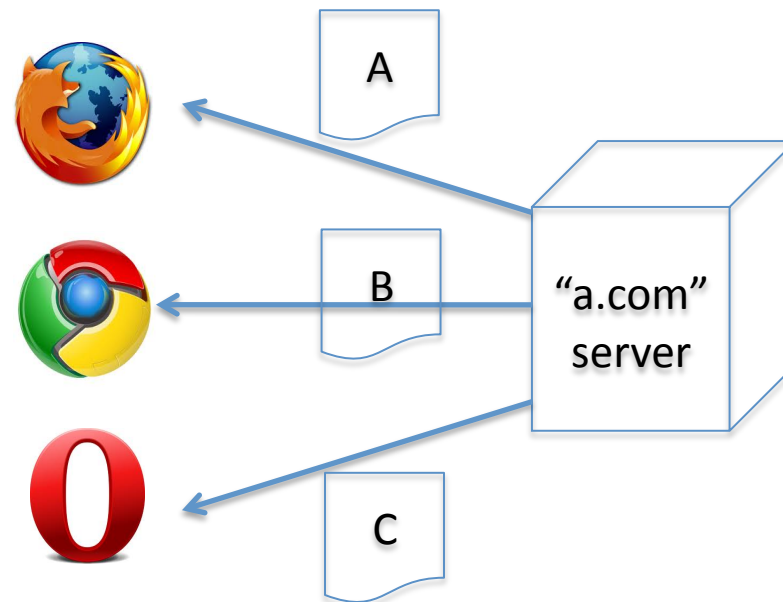
Same specification “roughly” followed

How to Compare Different Browsers?

- States to compare: display + network
 - Display: vision based page layout abstraction
- Interaction with server
- Client side non-determinism

Challenge: Interaction with Server

- Pages from server can be different



Request Duplication Is Bad

- Time difference
 - Page get updated
- Post requests
 - Output commit

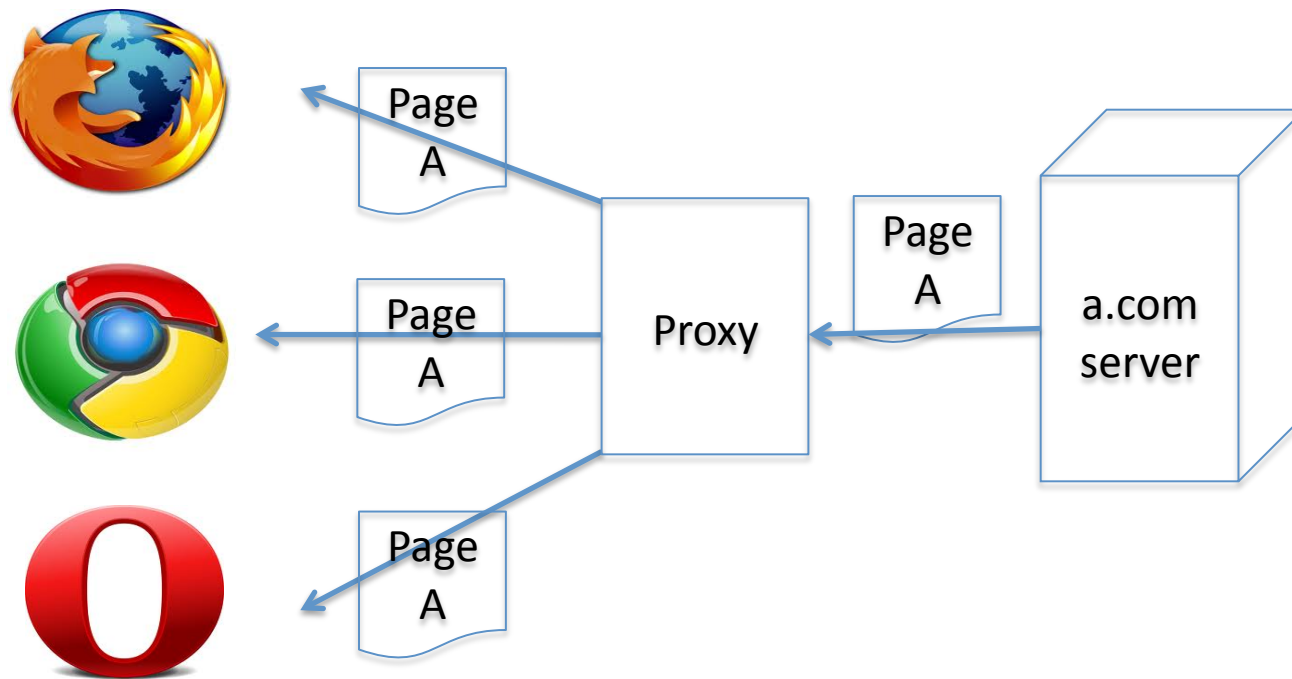


Solution

- Avoid major changes to browser
 - Browsers self-update is easy
 - Open source is not required
- Solution: proxy replication
 - Replicate incoming network data with proxy
 - HTTPS handling: Man-in-the-middle

Solution: Proxy Replication

- One browser as seen by server



Challenge: Client Side Non-determinism

- Same page content, different execution result

```
<html>
...
<script>
  randomId = Math.random();
  url = "doubleclick.com?ad=" + randomId;
</script>
...
</html>
```

Client Non-determinism Summary

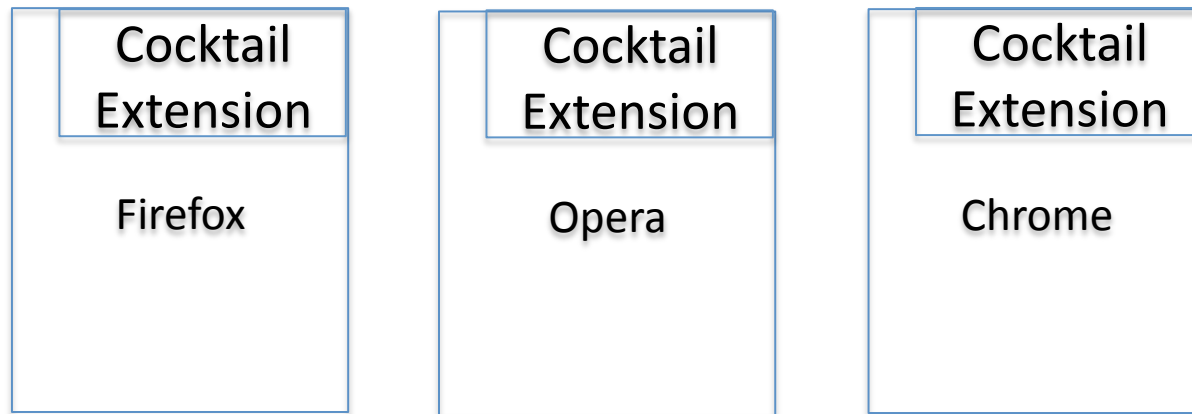
- Script related randomness

Random Number	Math.random()
Date and Time	new Date()
Browser Specific Value	window.opera; navigator.appName
Browser Locale	“en-us” VS “en-US”
...	...

- Browser specific behaviors
 - E.g., Opera community

Solution

- Extension modifies script execution
 - Overwrites Math, Date, window.opera



- Browser configuration change
 - Disable Opera community
 - Adjust browser locale

False Positive


- Browsers treat malformed URL differently

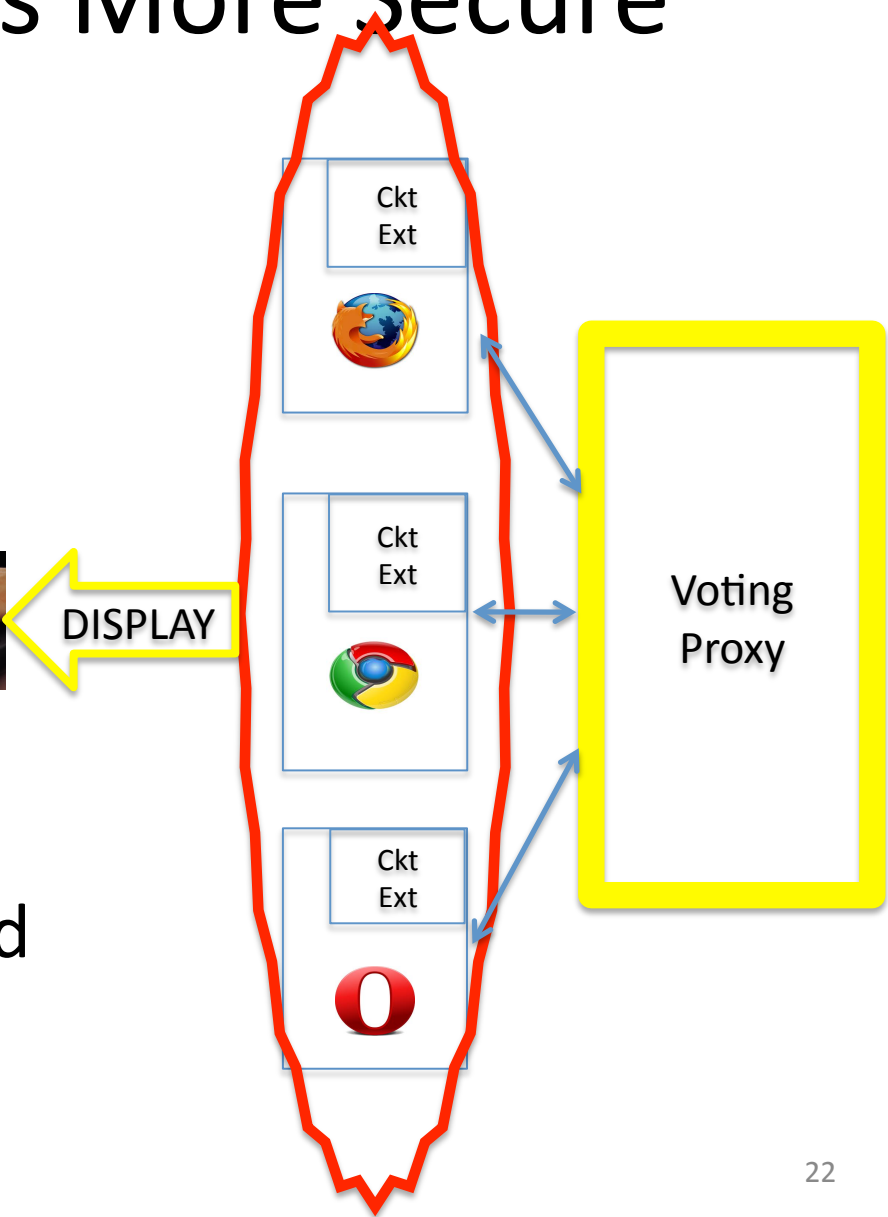
```
<iframe src="
http://www.adfusion.com/Adfusion.PartnerSite/ca
teoryhtml.aspx?userfeedguid=948fbed8-69ae-4659
-b3c1-b9863e5ab24e&clicktag=http://ads.bluelith
ium.com/clk?2,13%...%2Ffrec%2F,&CB={REQUES
TID}
```



```
width="300" height="250" scrolling="no"
frameborder="0" marginheight="0"
marginwidth="0"
></iframe>
```

Why Cocktail Is More Secure

- Voting == Security
- Withstand some F.P.
 - Only need 2 to proceed
- Ext. to eliminate non-determinism 
- Looks like a good one
Acts like a good one
It *is* one uncompromised browser



Implementation

- UI replication
 - Recorder and replayer: Extension
 - Passing UI events across browsers: Proxy
- UI Display capturing and voting
 - ImageMagick and OpenCV
- Proxy
 - OpenSSL for MITM

Evaluation

- Security analysis
 - User interaction: CVE-2009-3076
 - Heap overflow: CVE-2009-2477
 - DOS attack: Firefox 3.0.4 DOS, April 2009
 - Same origin policy bypassing: CVE-2007-0981
- Performance
 - 30% slower comparing to Firefox

Conclusion

- Mixing different browsers for better security
 - Practical N-Version programming for browsers
 - Solutions for design challenges
 - Security shifted to thin layer instead of big software

Thank you!

- Q/A
- Hui Xue (huixue2@uiuc.edu)
<https://netfiles.uiuc.edu/huixue2/www/>

Window of Exposure

