Phishing on Demand
Improvements & Survey

Yannick Hänni
What is “Phishing on Demand”? 

A portable phishing framework.

No programming knowledge required.

Supports Mac OS, Windows, and Linux.
Architecture

Phishing on Demand (Browser component)
- Tracking actions
- Rendering view

Phishing on Demand (Server component)
- Processing actions
- Processing screenshots & generating metadata

Selenium (Browser control)
- Replaying actions
- Taking screenshots
Loggen Sie sich bei PayPal ein

E-Mail-Adresse oder Handynummer

Weiter

oder

Neuanmelden

Deutsch | Französisch | Englisch
New functionality: Browser history
Challenges

Caching

State change detection

Correct rerouting

Cross site tracking
What we have tried...

Using the JavaScript history manipulation method `popstate(event e)`

Which button was pressed?

Workaround corrupts browser history!
What we have tried...

Using the favicon URL to determine the current state:

```javascript
document.querySelector("link[rel*='icon']").href
```

... but there was a new problem: browser caching.
How to avoid browser caching?

Random favicon file names!
Final workflow

Client (browser)

- Browsing history event
  (back / forward button pressed)

- Browser opens cached site
  (and re-executes JavaScript)

- Browser returns unique ID
  (based on the favicon name)

Server

- Server notifies client
  (to let the browser update the view)

- Server opens found website
  (and updates its internal state)

- Server uses ID to find website
  (internal bookkeeping with a map)
Result

Working navigation buttons

Authentic browser history

Cross site tracking

No caching issues

Performance loss

Minor favicon inconsistencies
Further improvements and future work

Scrolling
Webdriver issues
Code cleanup

Popup window placement
Horizontal scrollbars on some pages
Reaction and loading times
FPS
Demo
Survey design

Personal information
Website usability tests
Phishing experience
Demography

7 participants

- Age: 30 years on average
- Web experience: 18 years on average
- Developer experience: 1 year on average
Have participants been phished before?

- **Attempted**
- **Successful**

Bar chart showing attempted and successful rates in percentage.
Probability of phishing on website

estimation of participants

SRF | Google | Facebook | Amazon
Did participants notice the phishing?

No!

They noticed instead:

- Lower performance
- Repeated login
- Scrollbars
Other observations

- Google: 4.5
- SRF: 3.75
- Amazon: 2.5
- Facebook: 3.75

- Usability
- Regularity of their use

Estimation of participants
Conclusion

No more critical issues

Less minor issues

Better performance

Convincing results