

# BLUEPRINT: Robust Prevention of Cross-Site Scripting Attacks for Existing Browsers

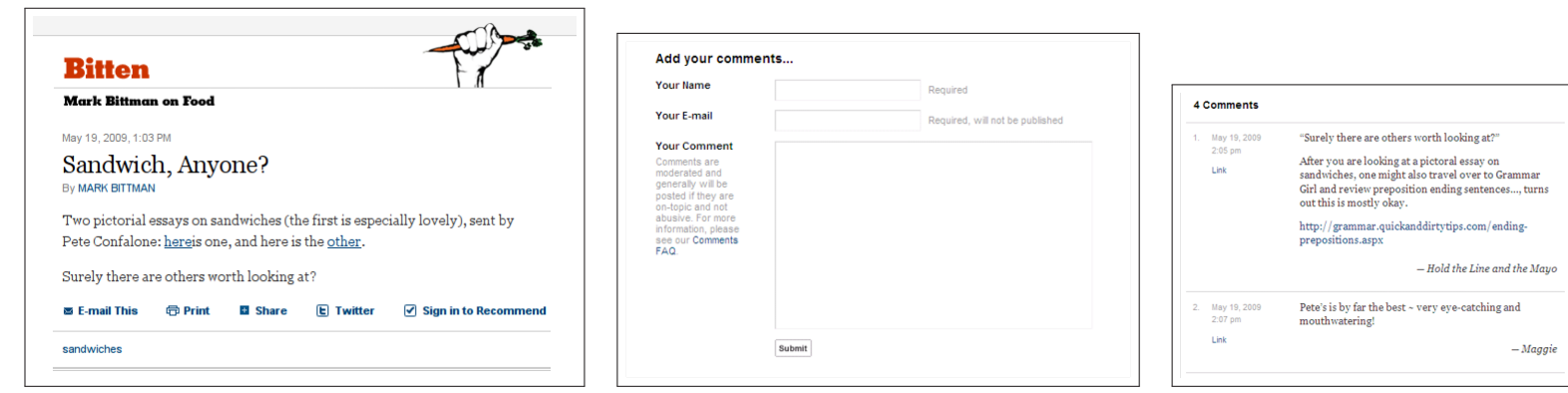
Mike Ter Louw V.N. Venkatakrisnan

Appeared in IEEE Symposium on Security and Privacy, 2009 (10.2% acceptance rate)

## 1. Web 2.0 and User Created Content (UCC)

A typical blog web application:

1. Author writes blog article
2. App encourages readers ("end users") to submit comments
3. Readers format their comments using HTML tags



(1) Blog article (2) Comment form (3) User comments

Most "Web 2.0" applications fueled by rich content created by end users.

Many more popular applications, including:

- Wikis (e.g., Wikipedia)
- Social networks (e.g., Facebook)
- Product reviews (e.g., Amazon)

## 2. Cross-Site Scripting (XSS) Attacks

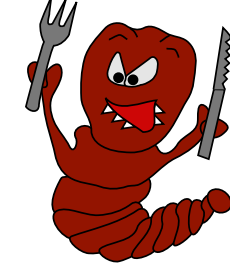
**Cross site scripting: Unauthorized script execution in a web app**  
The top security threat against web applications (OWASP Top Ten '07)

On a blog, user created content should not contain executable script code, because...

- JavaScript executed in browser with full app privileges
- End users not trusted to run privileged code
- Many attacks possible if malicious user (i.e., **attacker**) can run code

Harmful effects of XSS attacks:

- Administrator and user accounts hijacked (cookie theft)
- Sensitive page data leaked (confidentiality)
- Site content manipulated / defaced (integrity)
- Site content and services blocked (availability)
- Attack code propagate to other users and apps



## 3. Cross-Site Scripting Defense: Content Filtering

BLUEPRINT improves upon two main approaches to XSS defense:

1. Content filtering
2. Browser-server collaboration

**Content filtering (sanitization)**

- Analyzes untrusted user content for scripts
- Uses HTML parser, regular expressions
- Removes any scripts found during analysis

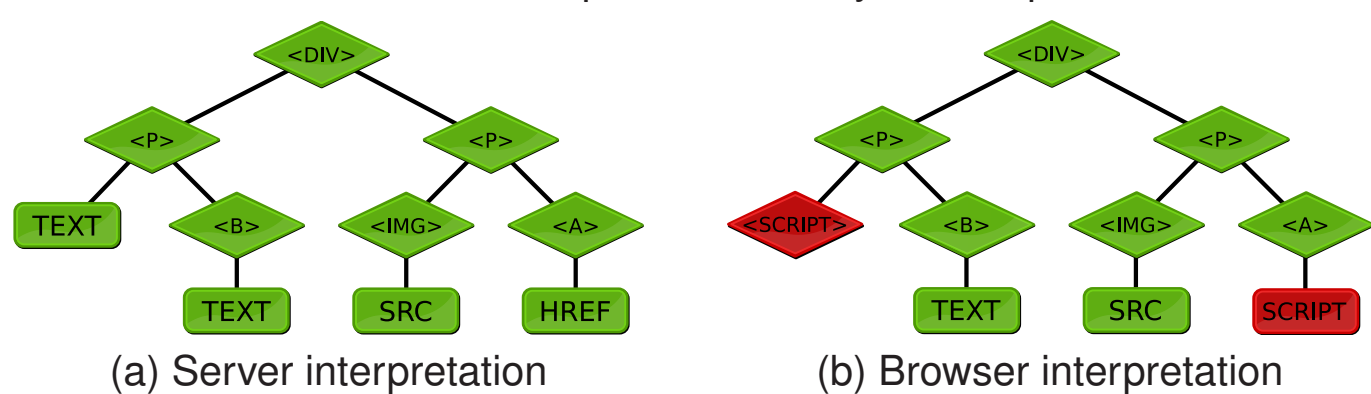
**Weakness: Browser still interprets filtered HTML, executes scripts content filter didn't detect**

- Filter must interpret HTML, and thus identify all scripts, exactly as browser would
- Browsers have anomalous parsing behaviors ("quirks"); difficult to predict and model
- When browser parses differently than content filter, XSS attacks can succeed
- Consider a **malicious blog comment**:

```

1 <div>
2 <p>
3   I found this picture +ADw-SCRIPT+AD4-attack ();
4   <b>hilarious!</b>
5 </p>
6 <p>
7   
8   <a href=" &#14; javasc&#x0A;ript:attack (...); ">click me</a>
9 </p>
10 </div>
    
```

Content filter and browser parse differently due to quirks:



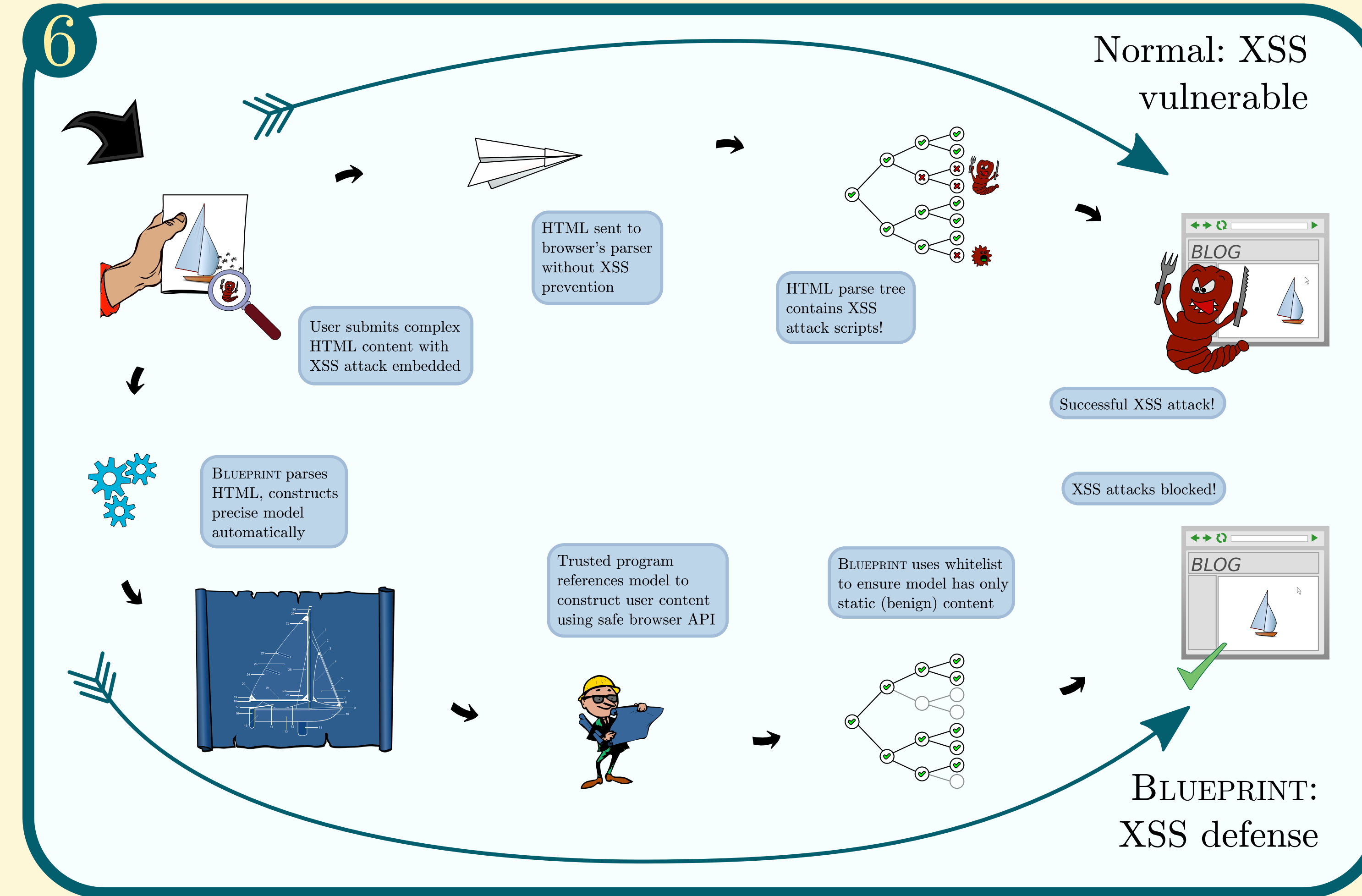
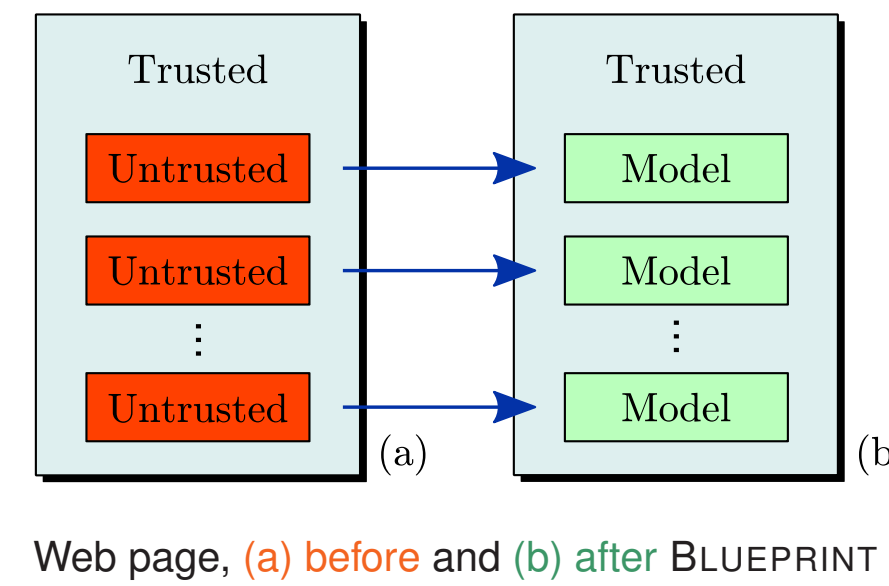
## 4. BLUEPRINT Objectives

Develop a cross-site scripting defense that:

1. **Robustly protects** against XSS attacks, despite parsing quirks,
2. **Supports** benign HTML user input, and is
3. **Compatible** with existing browsers.

## 5. Approach Overview

- Replace every instance of untrusted content on a web page with a **model**
- Carefully render models in browser to avoid XSS attacks



## 8. Cross-Site Scripting Defense: Browser - Server Collaboration

**Cross-site scripting can be defeated with changes to web standards and web browsers:**

1. Server notifies browser of precisely which HTML markup is untrusted user content
  2. Browser enforces *no-script* policy over untrusted HTML
- Ref. **Browser-Enforced Embedded Policies** by Jim et al. [2]  
Ref. **Document Structure Integrity** by Saxena, et al. [3]
- This defense can take **many years** to reach vast majority of end users
    - Need consensus: standardized collaboration protocol
    - Browsers must implement support for collaboration
    - End users need to install updated browsers
  - Need **robust XSS defense for near term**

## 9. XSS Defense for Today's Web

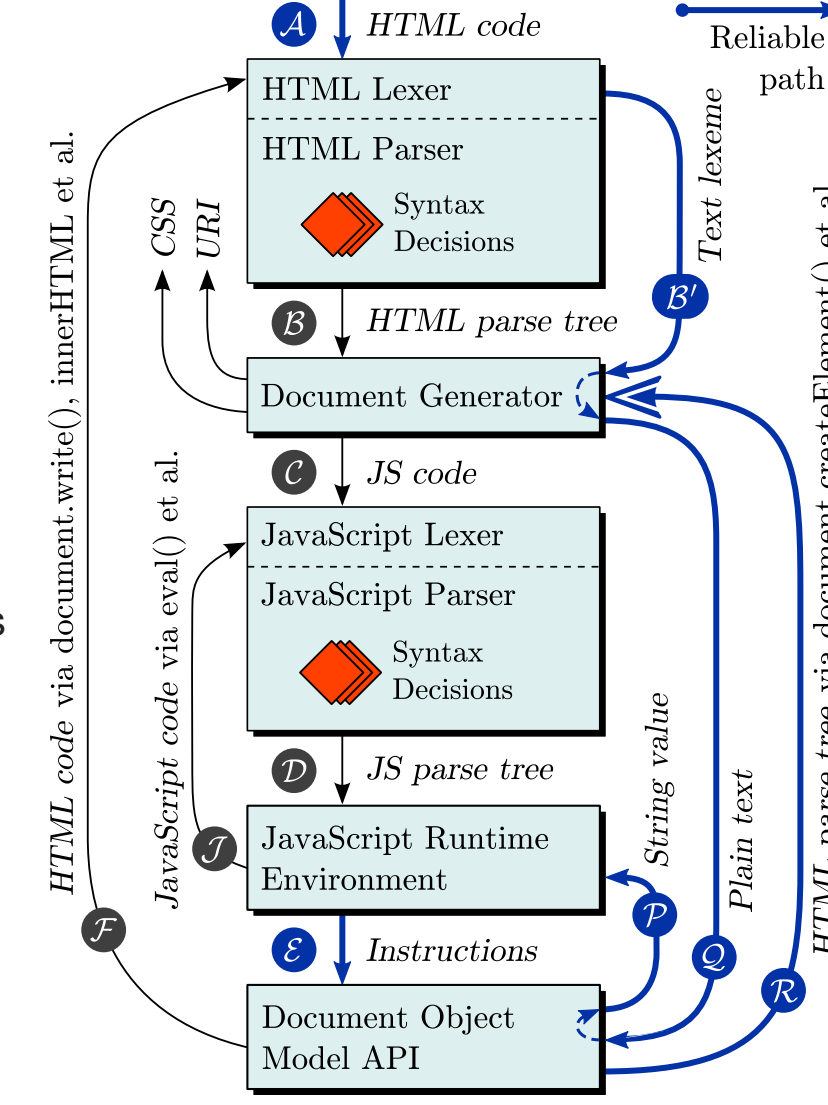
**BLUEPRINT was specifically designed to work on existing browsers.**

- Effective on over **96% of web browsers** in active use (Global Web Browser Marketshare Statistics by Net Applications [5])
  - Chrome 1
  - Firefox 3
  - Firefox 2
  - Internet Explorer 7
  - Internet Explorer 6
  - Opera 9.6
  - Safari 3.2
  - Safari 3.1
- Requires no browser modifications, plug-ins, nor any changes to default settings
- Web sites can shield the vast majority of end users against XSS attacks **today!**

## 7. Why it Works: Rendering Strategy for Untrusted Web Content

Avoid **XSS vectors** by ensuring untrusted content takes **reliable path** through browser's HTML interpreter:

- A** Send model of approved, XSS-free parse tree to browser
- B'** Encode model using **syntactically inert** character alphabet (Base64), mitigating risk of browser quirks
- Q** Trusted JavaScript code reads model from web page as text
- P** Process model data as string
- E** Procedurally construct parse tree from model using safe DOM API
- R** Send approved, XSS-free parse tree to page renderer

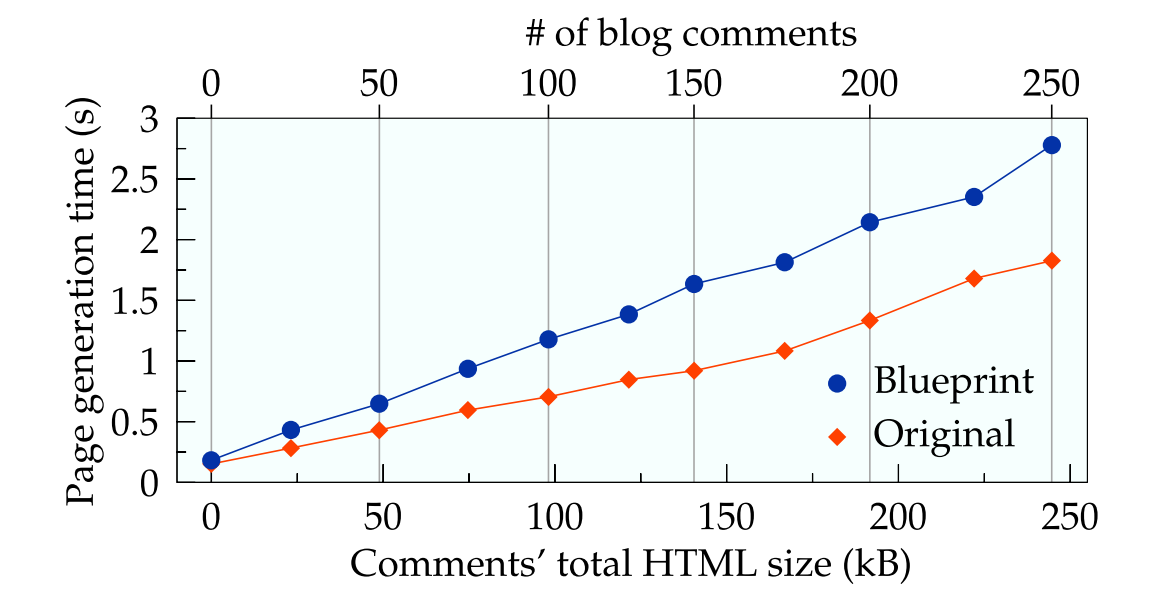


## 10. Evaluation: Defensive Effectiveness

- Evaluated against attack samples from XSS Cheat Sheet [4].
- 94 sample attack HTML sequences designed to penetrate XSS defenses
- Tested 8 browsers comprising 96% market share
- All attacks prevented in all browsers.**

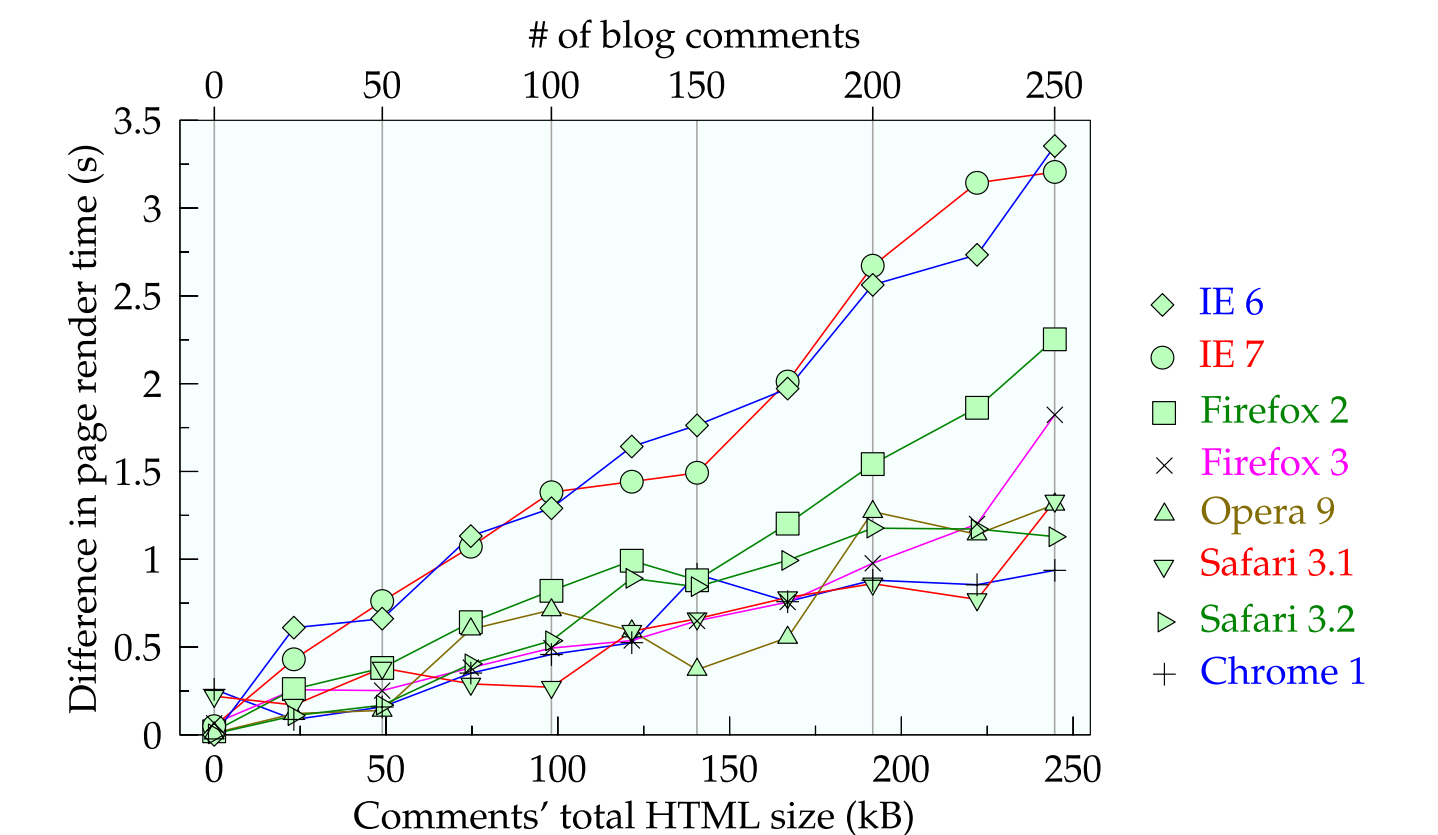
## 11. Evaluation: Performance overhead

**Server page generation latency:**



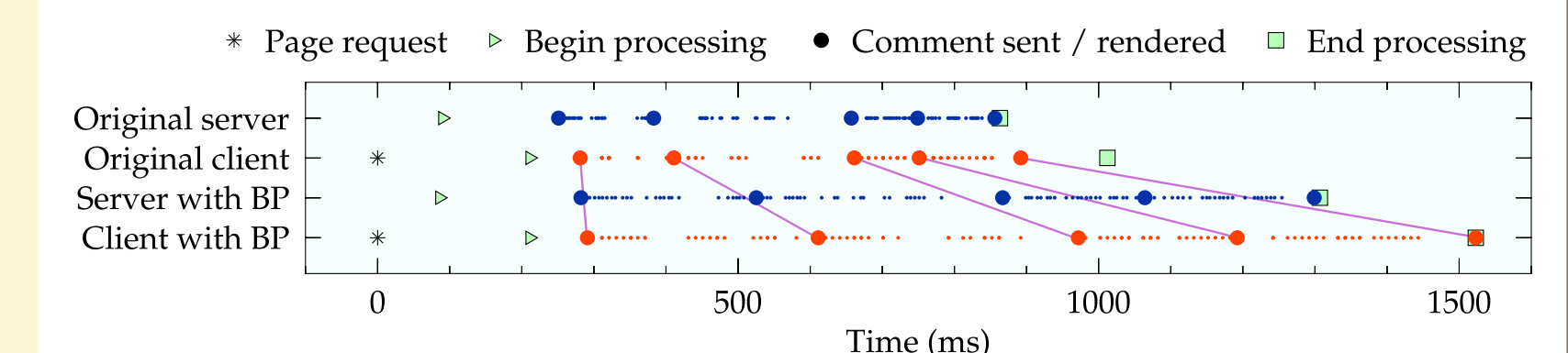
- Added BLUEPRINT to WordPress blog application
- Measured time to generate blog pages with and without BLUEPRINT
- Some significant overhead, partly due to redundant content filtering

**Client rendering latency:**



- Measured page rendering latency for 8 browsers under varied workloads
- Highest workload (250 blog comments) had only **3.5 seconds** latency
- Internet Explorer browsers significantly stressed by our defense

**End user experience:**



- Measured end user page experience with and without BLUEPRINT
- For 100 blog comments, only **1/2 second** latency overall
- Initial comments appear without perceptible delay

## 12. References

- [1] M. Ter Louw and V.N. Venkatakrisnan, "BLUEPRINT: Robust prevention of cross-site scripting attacks for existing browsers," in *30th IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2009.
- [2] T. Jim, N. Swamy, and M. Hicks, "Defeating script injection attacks with browser-enforced embedded policies," in *16th International World Wide Web Conference*, Banff, AB, Canada, May 2007.
- [3] P. Saxena, D. Song, and Y. Nadji, "Document structure integrity: A robust basis for cross-site scripting defense," in *16th Annual Network & Distributed System Security Symposium*, San Diego, CA, USA, Feb. 2009.
- [4] R. Hansen, "XSS (cross site scripting) cheat sheet esp: for filter evasion," 2008. [Online]. Available: <http://hackers.org/xss.html>
- [5] Net Applications, "Browser version market share." Statistics for Q4 2008. [Online]. Available: <http://marketshare.hitslink.com/browser-market-share.aspx?qprid=2&qptimeframe=Q&qpsp=39>