Improving Web Security:

Finding and fixing vulnerabilities in web security mechanisms

Devdatta Akhawe, Adam Barth,
Peifung E. Lam, John C. Mitchell and Dawn Song

Stanford Computer Security Lab

Introduction

- The Web is complex and fast evolving.
- New browser features, protocols, and standards added at a rapid pace.
- Vulnerabilities and security invariants assumed by web applications.
- We believe that abstract yet informed models of the Web will be amenable to automation, reveal practical attacks, and support useful evaluation of alternate designs.

Introduction (cont.)

- The Web mechanisms we have studied include:
  - HTML5 Forms
  - Referer validation
  - WebAuth protocol
- Our analysis reveals previously unknown attacks
- Countermeasures proposed for each attack

Introduction (cont.)

- These web mechanisms were analyzed using a common approach we have developed which involves:
    - A formal model of the web
    - Implementation of the formal model in Alloy
    - Modeling of the web mechanisms under study in Alloy

Outline of the talk

- Attacks and countermeasures for
  - HTML5 Forms
  - Referer validation
  - WebAuth protocol

Outline (Cont.)

Modeling the Web

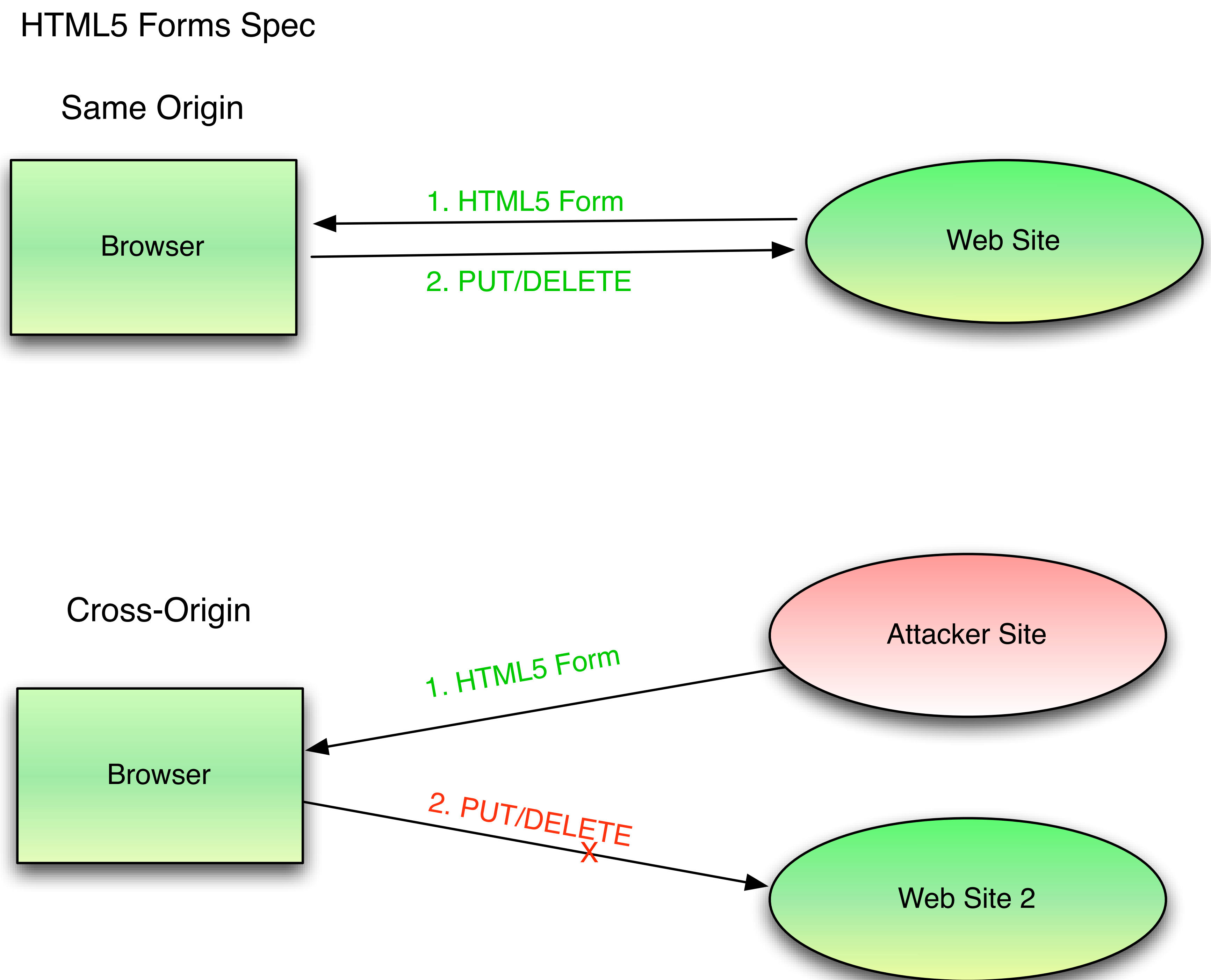
- A formal model of the Web
- Implementation of the model in Alloy
- Statistics of Alloy implementation

- Attacks and countermeasures
  - HTML5 Forms
  - Referer validation
  - WebAuth protocol

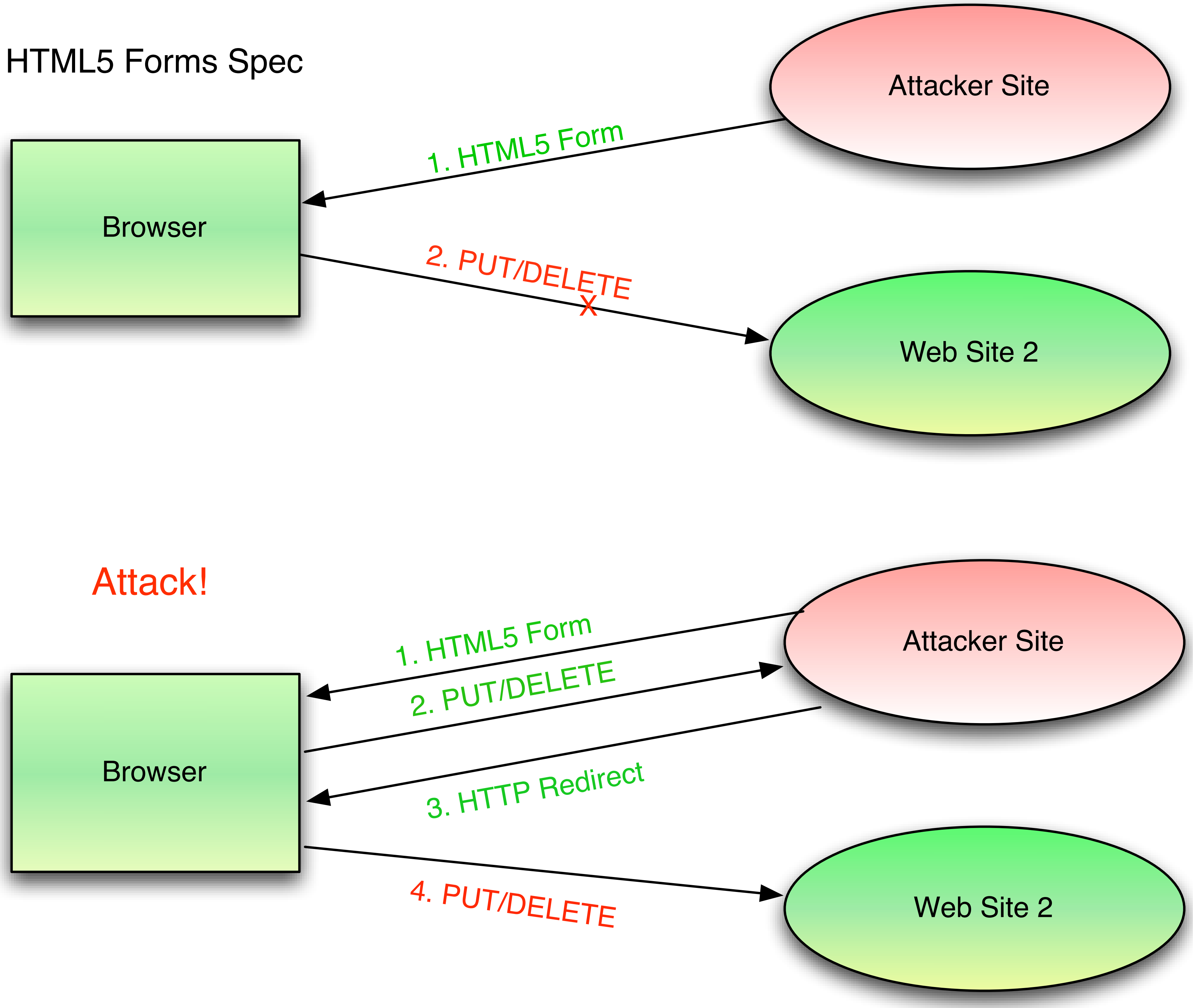
HTML5 Forms

- HTML5 is the next major revision of HTML
- FormElement API in HTML5 can generate HTTP requests with PUT and DELETE methods
- Same origin policy applies to such requests

HTML5 Forms (Cont.)



HTML5 Forms - Attack



Exploitation

- Attacker could illegitimately modify/delete resources on a RESTful website

Countermeasure

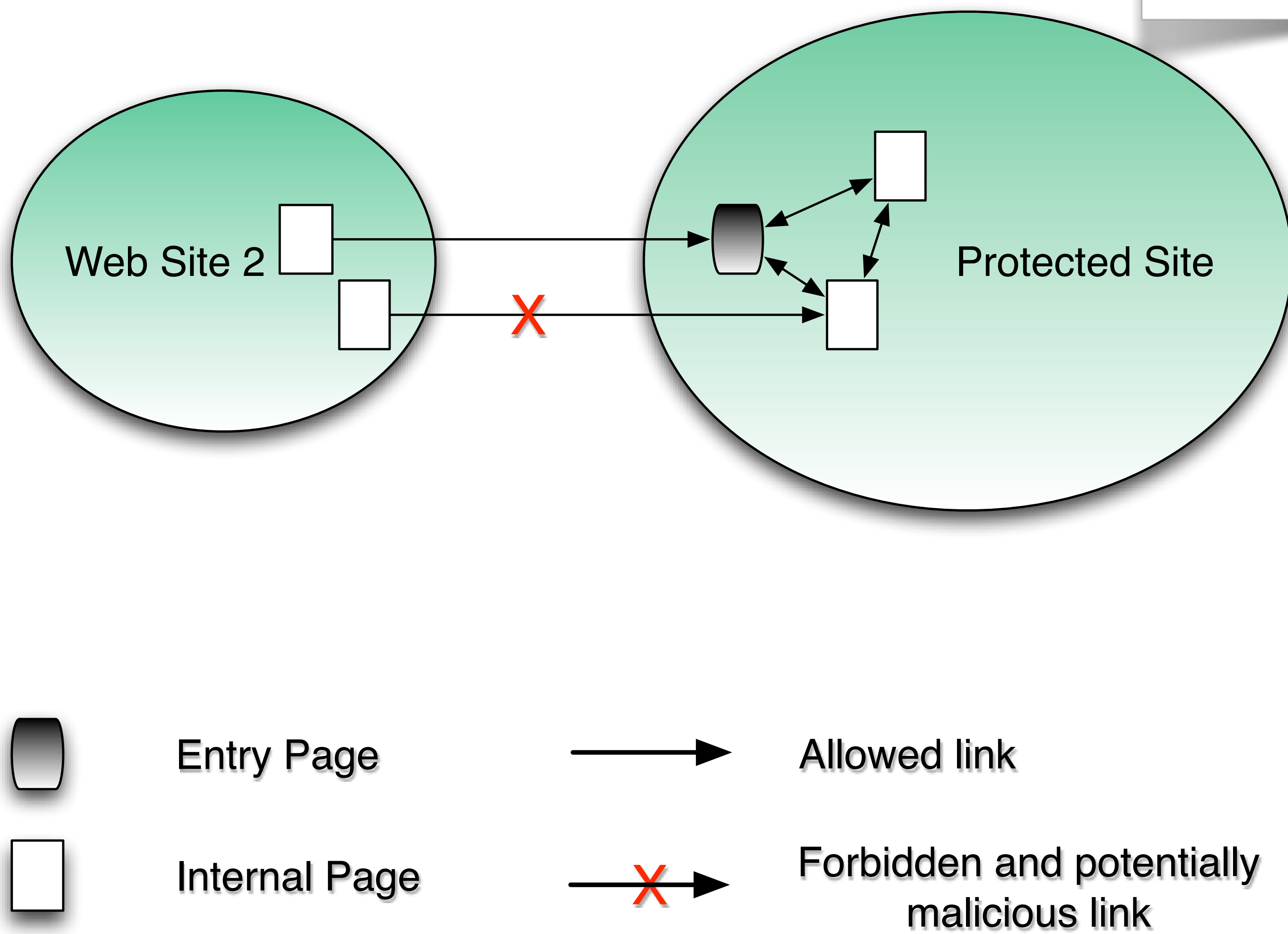
- Refuse to follow redirects of PUT/DELETE requests generated from HTML Forms
- Verified the fix up to a finite size in our model
- Recommendation accepted by the HTML5 working group

Referer Validation

- A proposed defense against Cross-Site Request Forgery (CSRF) and Cross-Site Scripting (XSS) [F. Kerschbaum, 2007]
- Websites would reject a request **unless**
  1. the referer header is from the same site, or
  2. the request is directed at an "entry" page vetted for CSRF and XSS vulnerabilities

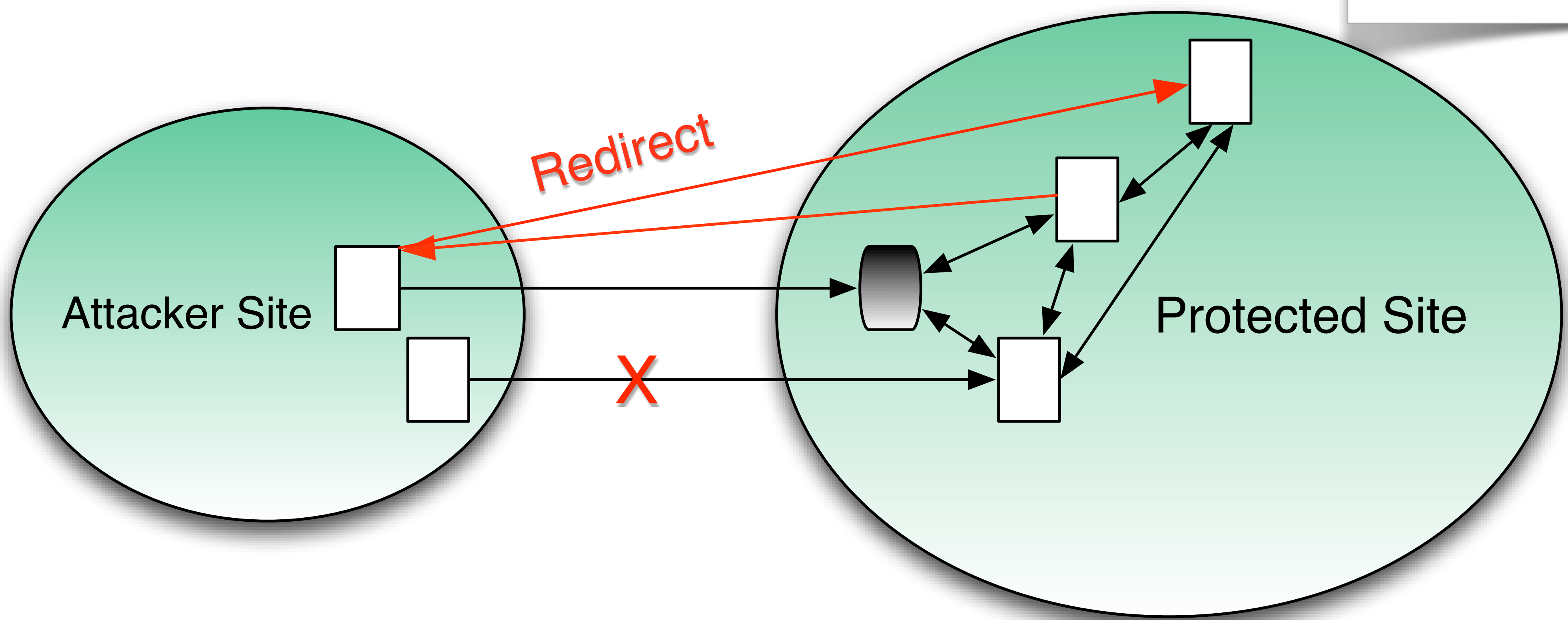
Referer Validation - proposal

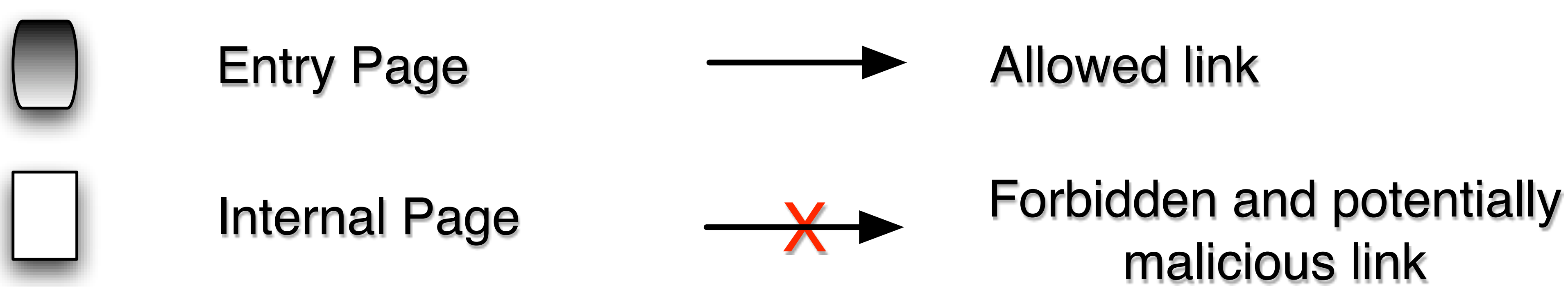
Figure adapted from F. Kerschbaum, “Simple cross-site attack prevention,” 2007, with attack (in red) added.



Referer Validation - Attack

Figure adapted from F. Kerschbaum, "Simple cross-site attack prevention," 2007, with attack (in red) added.





Referer Validation - Countermeasure

Exploitation

- CSRF and XSS can be carried out on websites protected with Referer Validation

Countermeasure

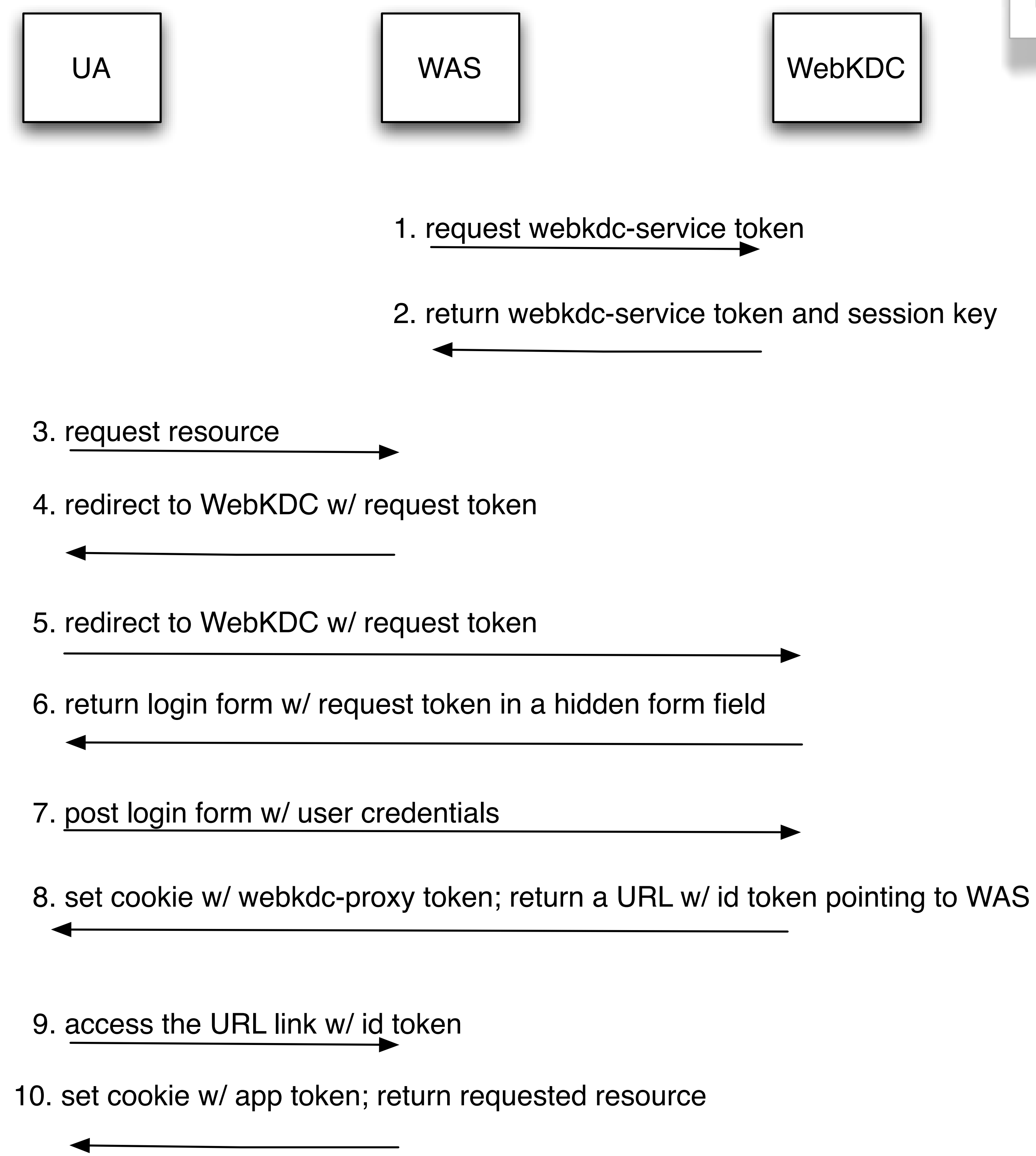
- This vulnerability is difficult to correct as Referer header has been widely deployed
- Websites can try to suppress all outgoing Referer headers using, for example, the noreferrer relation attribute on hyperlinks.

WebAuth

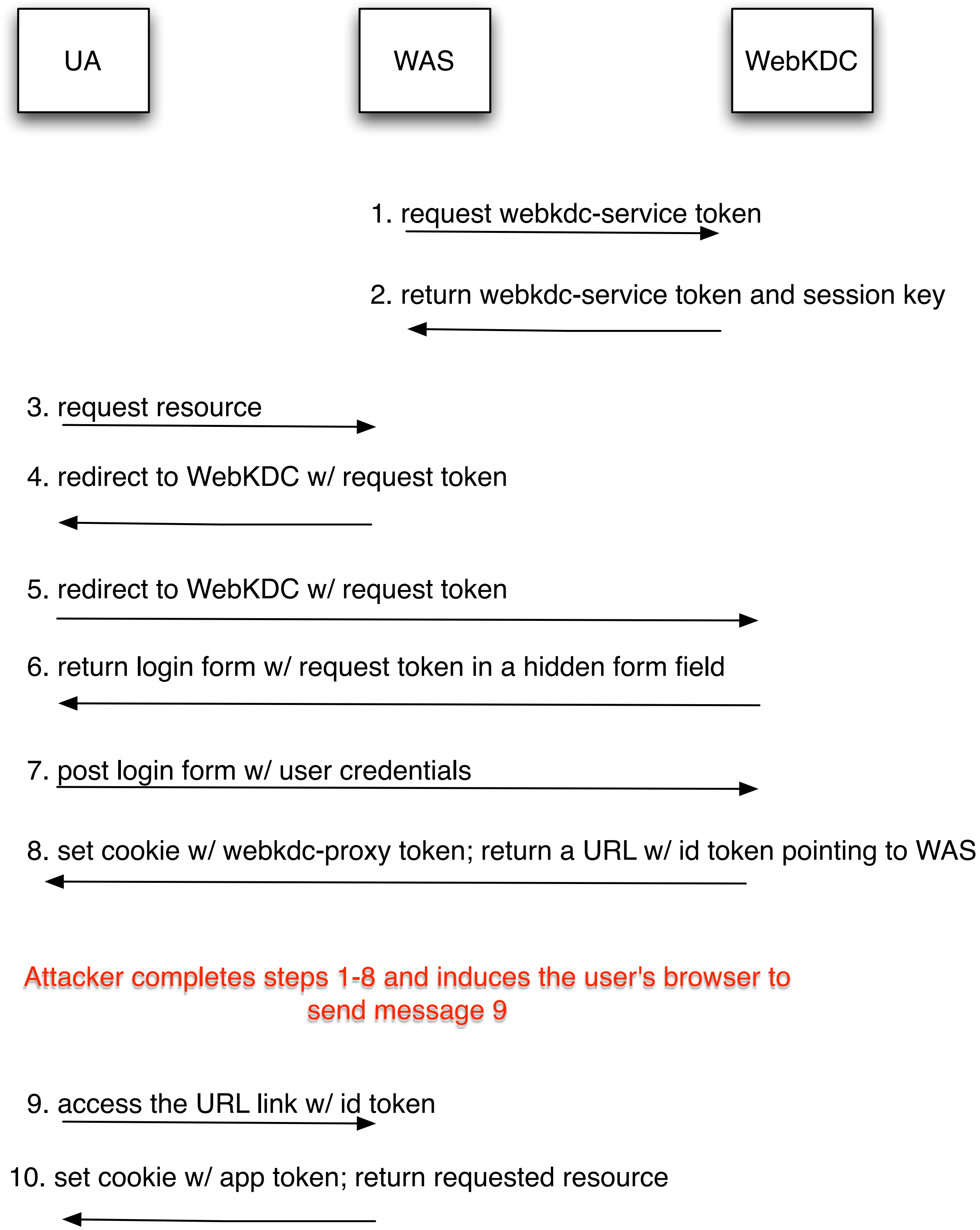
- Web-based Single Sign-On protocol
- WebAuth and a similar protocol, Central Authentication Service (CAS), are deployed at over 80 universities worldwide
- Although we analyze WebAuth specifically, we have verified the same vulnerability exists in CAS

WebAuth Protocol

Figure adapted from http://webauth.stanford.edu/protocol.html



WebAuth Protocol - Attack



WebAuth - exploitation

Exploitations

- An insider can share privileged web resources with unprivileged users without sharing login credentials
- Attacker can steal sensitive user information by logging users into attacker's account

Countermeasure

- Store a nonce in a host cookie to bind messages 3 and 9, and splice in messages in between by including the nonce in the request and id tokens.
- Verified the fix up to a finite size in our model

Modeling the Web

- A formal model of the Web
- Implementation of the model in Alloy
- Statistics of Alloy implementation

A formal model of the web

- We model web entities including browser, servers, and network
- Our threat models include attackers with various capabilities, such as:
  - web attacker with no special network privilege, and
  - network attacker that can eavesdrop and/or modify unencrypted traffic at will

A formal model of the web (cont.)

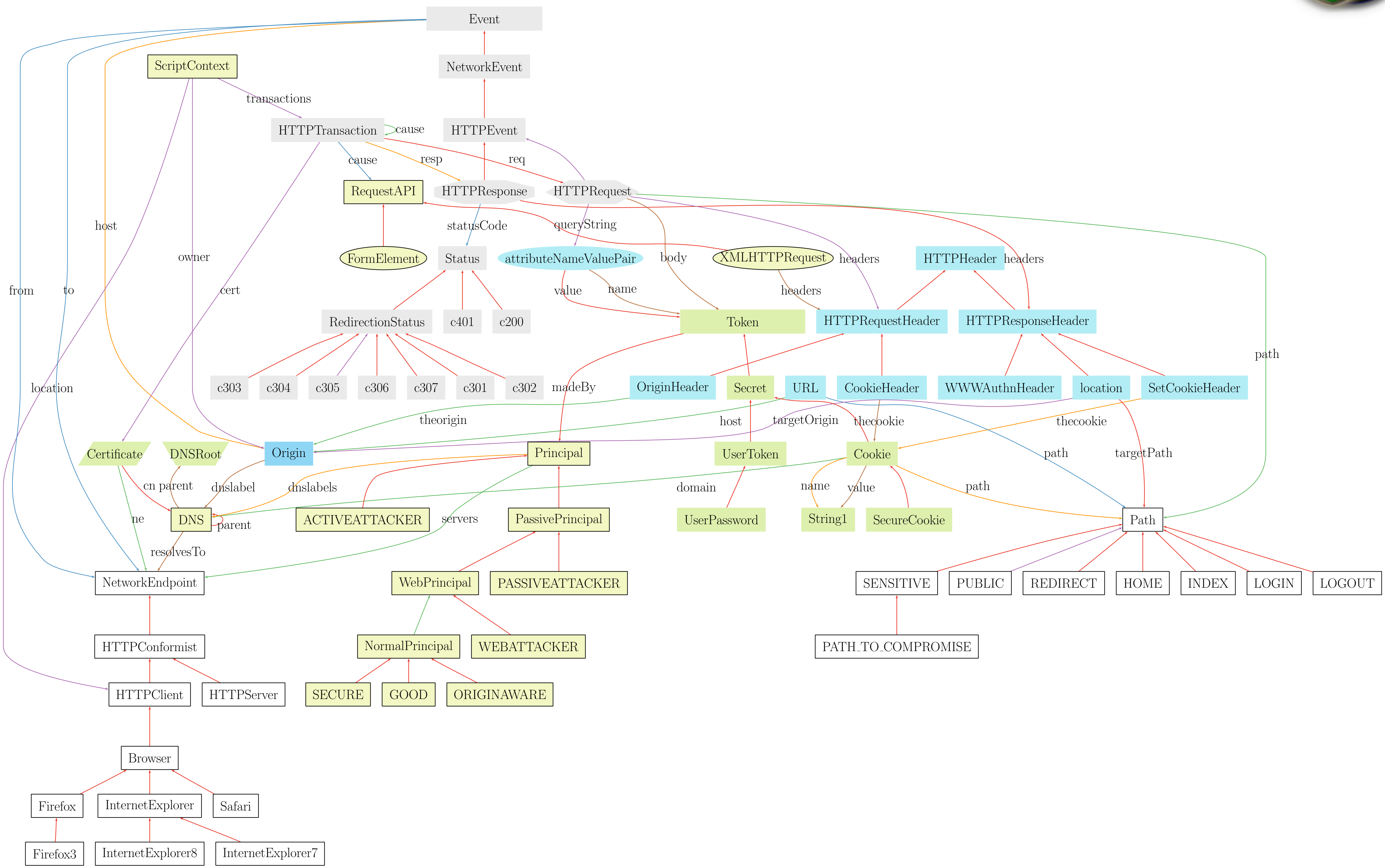
Main security goals we have identified include:

- Security invariants
  - Assumptions about how today's Web works
  - Example: no DELETE in cross-origin HTTP requests
- Session integrity
  - Attacker does not participate in the HTTP transaction

Alloy

- A declarative language based on first-order logic
- Facts and predicates about a model are declared
- The Alloy code is translated into a SAT instance
- SAT solver searches for counterexamples using bounded exhaustive search

MetaModel in Alloy



Example Alloy code

- Example code for session integrity

```
fun involvedServers[t:HTTPTransaction]:set NetworkEndpoint{
    (t.*cause & HTTPTransaction).resp.from
    + getTransactionOwner[t].servers
    }
pred webAttackerInCausalChain[t:HTTPTransaction]{
      some (WEBATTACKER.servers & involvedServers[t])
      }
```

Statistics for the case studies

Case Study	Lines of new code	No. of CNF clauses	CNF gen. time (sec)	CNF solve time (sec)
HTML5 Form	20	976,174	27.67	73.54
Referer Validation	35	974,924	30.75	9.06
WebAuth	214	355,093	602.4	35.44

- The base model contains some 2,000 lines of code
- Tests were performed on an Intel Core 2 Duo 3.16GHz CPU with 3.2 GB memory

Conclusion

- We identified previously unknown attacks in HTML5 Forms, Referer validation, and WebAuth
- Proposed countermeasures to the attacks.
- These attacks are identified based on a formal model the Web that we have developed, which is then implemented in the Alloy language.
- This modeling approach not only enables us to discover practical new attacks, but also serves to verify the security of alternate designs, up to a certain size of the model.

References

- HTML5 working group, "HTML5 Forms," 2010. [Online]. Available: http://www.whatwg.org/specs/web-apps/current-work/multipage/forms.html
- IP F. Kerschbaum, "Simple cross-site attack prevention," in Proceedings of the Third international workshop on Security and Privacy in Communication networks, 2007.
- R. Schemers and R. Allbery, "Webauth v3 technical specification," 2009. [Online]. Available: http://webauth.stanford.edu/protocol.html
- D. Mazurek, "CAS protocol," 2005. [Online]. Available: http://www.jasig.org/cas/protocol

Questions ?

Thank you!