#### CSci 5271 Introduction to Computer Security Day 20: Firewalls, NATs, and IDSes

Stephen McCamant University of Minnesota, Computer Science & Engineering

#### Outline

#### Cross-site scripting

More cross-site risks Announcements intermission Confidentiality and privacy Even web more risks Firewalls and NAT boxes Intrusion detection systems

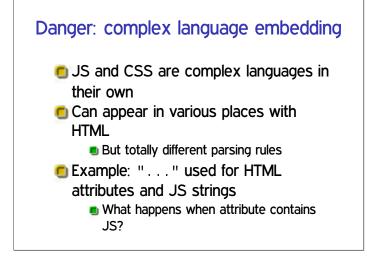
 XSS: HTML/JS injection (A3)
 Another use of injection template
 Attacker supplies HTML containing JavaScript (or occasionally CSS)
 OWASP's most prevalent weakness
 A category unto itself
 Easy to commit in any dynamic page construction

#### No string-free solution

- For server-side XSS, no way to avoid string concatenation
- Web page will be sent as text in the end

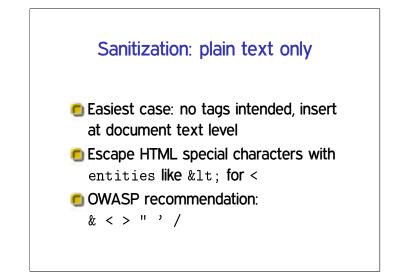
Research topic: ways to change this?

XSS especially hard kind of injection



#### Danger: forgiving parsers

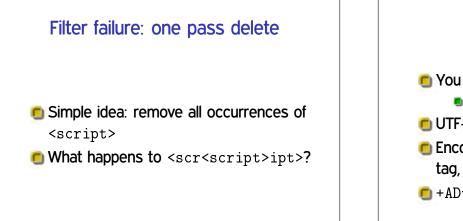
- History: handwritten HTML, browser competition
- Many syntax mistakes given "likely" interpretations
- Handling of incorrect syntax was not standardized

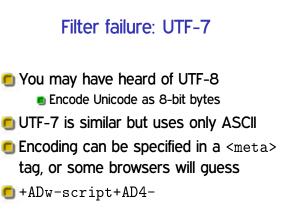


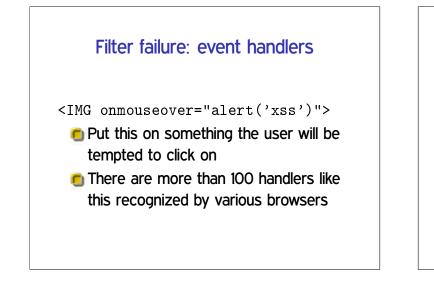
#### Sanitization: context matters

- An OWASP document lists 5 places in a web page you might insert text
   For the rest, "don't do that"
- Each one needs a very different kind of escaping









#### Use good libraries

- Coding your own defenses will never work
- Take advantage of known good implementations
- Best case: already built into your framework
  - Disappointingly rare

#### Content Security Policy

- New HTTP header, W3C candidate recommendation
- Lets site opt-in to stricter treatment of embedded content, such as:
  - No inline JS, only loaded from separate URLs
  - Disable JS eval et al.
- Has an interesting violation-reporting mode

#### Outline

Cross-site scripting

- More cross-site risks
- Announcements intermission
- Confidentiality and privacy
- Even web more risks
- Firewalls and NAT boxes
- Intrusion detection systems

#### HTTP header injection

- Untrusted data included in response headers
- Can include CRLF and new headers, or premature end to headers
- 🖲 AKA "response splitting"

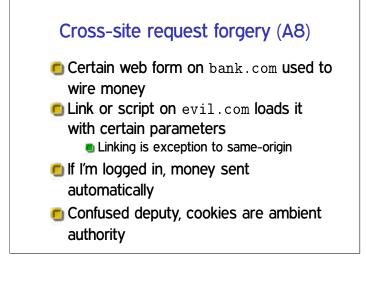
#### Content sniffing

Browsers determine file type from headers, extension, and content-based guessing

Latter two for ~ 1% server errors

- Many sites host "untrusted" images and media
- Inconsistencies in guessing lead to kind of XSS

E.g., "chimera" PNG-HTML document



#### **CSRF** prevention

Give site's forms random-nonce tokens

 E.g., in POST hidden fields
 Not in a cookie, that's the whole point

 Reject requests without proper token

 Or, ask user to re-authenticate

 XSS can be used to steal CSRF tokens

#### Open redirects (A10)

- Common for one page to redirect clients to another
- Target should be validated
  - With authentication check if appropriate
- Open redirect: target supplied in parameter with no checks
  - Doesn't directly hurt the hosting site
  - But reputation risk, say if used in phishing
  - We teach users to trust by site

#### Outline

Cross-site scripting

- More cross-site risks
- Announcements intermission
- Confidentiality and privacy
- Even web more risks
- Firewalls and NAT boxes
- Intrusion detection systems

## Upcoming assignments Exercise set 4 posted late last week, due 11/21

A week from this Thursday

HW2 almost ready

#### Note: more readings this week

- More details on how to set up firewalls
- Burglar alarms and "mimicry" attack on IDSes
- Containing high-speed worms
- Virus evolution in 2012
- Use bookmarklet for on-campus download links

#### Outline

Cross-site scripting More cross-site risks Announcements intermission

#### Confidentiality and privacy

Even web more risks

Firewalls and NAT boxes

Intrusion detection systems

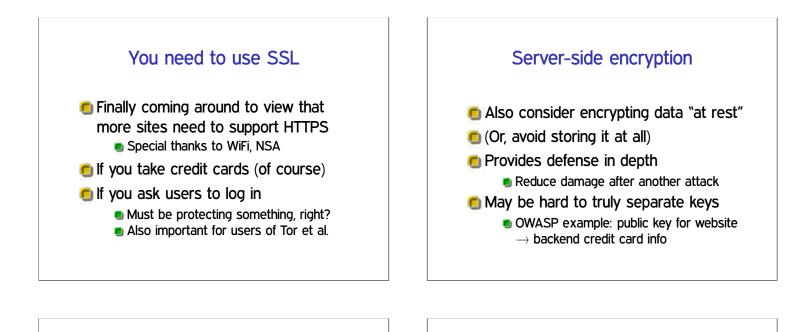
#### Site perspective (A6)

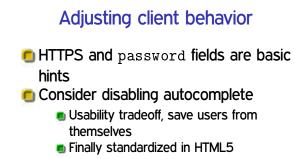
Protect confidentiality of authenticators

Passwords, session cookies, CSRF tokens

#### Duty to protect some customer info

- Personally identifying info ("identity theft")
- Credit-card info (Payment Card Industry Data Security Standards)
- Health care (HIPAA), education (FERPA)
- Whatever customers reasonably expect

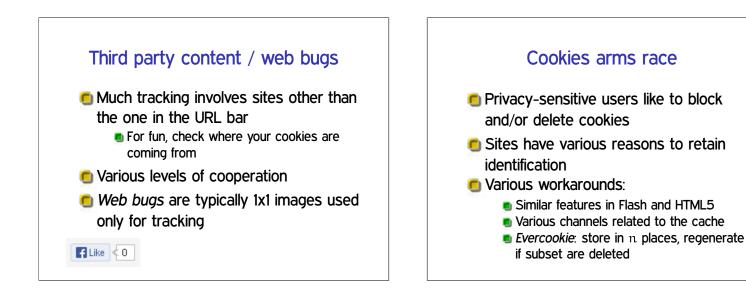


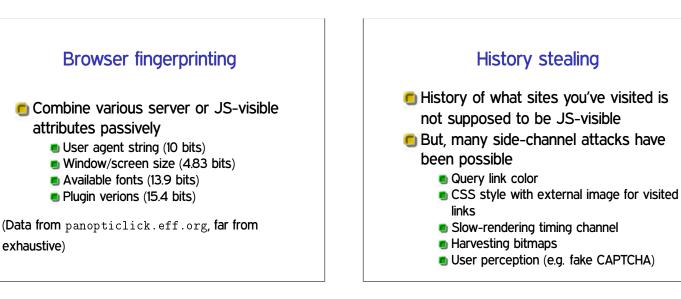


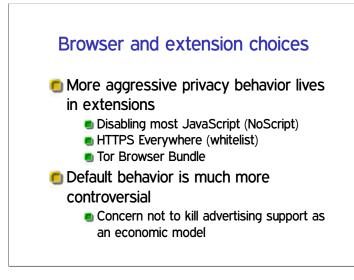
- Consider disabling caching
  - Performance tradeoff
  - Better not to have this on user's disk
  - Or proxy? You need SSL

#### User vs. site perspective

- User privacy goals can be opposed to site goals
- Such as in tracking for advertisements
- Browser makers can find themselves in the middle
  - Of course, differ in institutional pressures







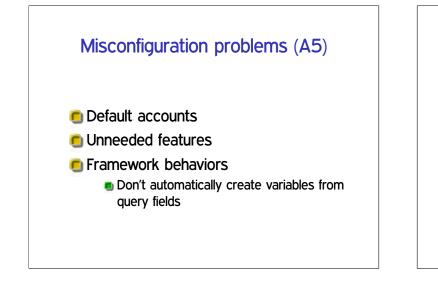
#### Outline

Cross-site scripting

- More cross-site risks
- Announcements intermission
- Confidentiality and privacy

#### Even web more risks

- Firewalls and NAT boxes
- Intrusion detection systems



#### **Openness tradeoffs**

- Error reporting

   Few benign users want to see a stack backtrace
   Directory listings
  - Hallmark of the old days
- Readable source code of scripts
  Doesn't have your DB password in it, does it?

# Using vulnerable components (A9) Large web apps can use a lot of third-part code Convenient for attackers too OWASP: two popular vulnerable components downloaded 22m times Hiding doesn't work if it's popular Stay up to date on security announcements

#### Clickjacking

- Fool users about what they're clicking on
  - Circumvent security confirmations
  - Fabricate ad interest
- Example techniques:
  - Frame embedding
  - Transparency
  - Spoof cursor
  - Temporal "bait and switch"

#### Crawling and scraping

- A lot of web content is free-of-charge, but proprietary
  - Yours in a certain context, if you view ads, etc.
- Sites don't want it downloaded automatically (web crawling)
- Or parsed and user for another purpose (screen scraping)
- High-rate or honest access detectable

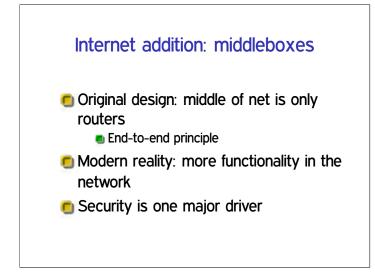
#### Outline

Cross-site scripting

- More cross-site risks
- Announcements intermission
- Confidentiality and privacy
- Even web more risks

#### Firewalls and NAT boxes

Intrusion detection systems



#### Security/connectivity tradeoff

- A lot of security risk comes from a network connection
  - Attacker could be anywhere in the world
- Reducing connectivity makes security easier
- Connectivity demand comes from end users

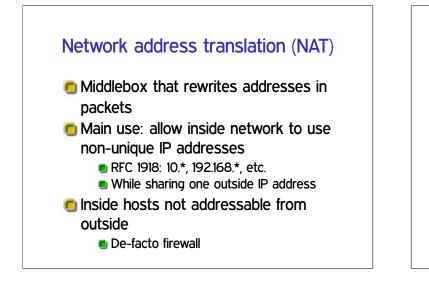


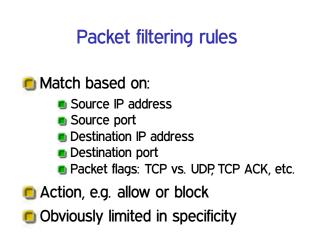
# Default: deny Usual whitelist approach: first, block

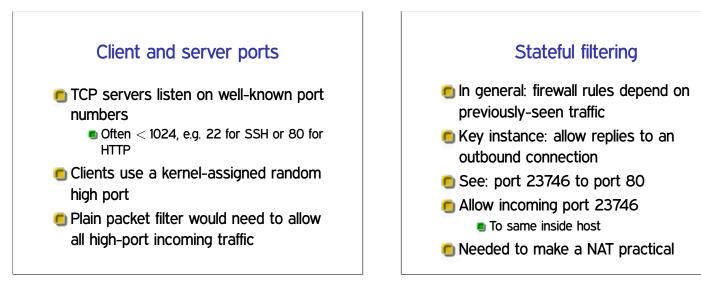
- everything
- 🖲 Then allow certain traffic
- Basic: filter packets based on headers
- More sophisticated: proxy traffic at a higher level

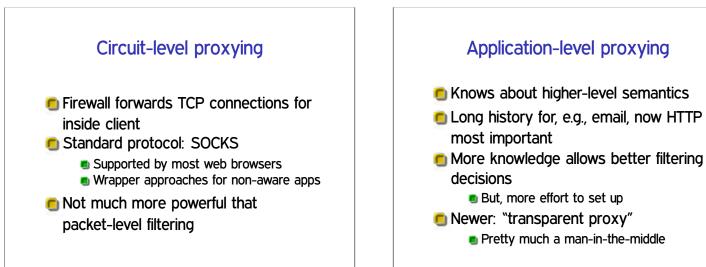
#### IPv4 address scarcity

- Design limit of 2<sup>32</sup> hosts
   Actually less for many reasons
- Addresses becoming gradually more scarce over a many-year scale
- Some high-profile exhaustions in 2011
- IPv6 adoption still very low, occasional signs of progress







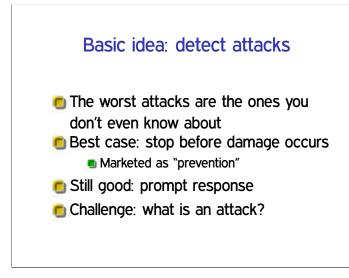


#### Tunneling

- Any data can be transmitted on any channel, if both sides agree
- E.g., encapsulate IP packets over SSH connection
  - Compare covert channels, steganography
- Powerful way to subvert firewall
  - Some legitimate uses

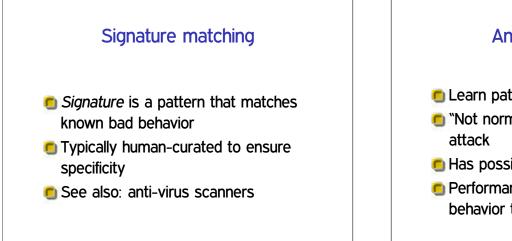
#### Outline

- Cross-site scripting More cross-site risks Announcements intermission Confidentiality and privacy Even web more risks
- Firewalls and NAT boxes
- Intrusion detection systems



#### Network and host-based IDSes

- Network IDS: watch packets similar to firewall
  - But don't know what's bad until you see it
     More often implemented offline
- Host-based IDS: look for compromised process or user from within machine



### Anomaly detection

- Learn pattern of normal behavior
- Not normal" is a sign of a potential attack
- Has possibility of finding novel attacks
- Performance depends on normal behavior too

#### Recall: FPs and FNs

- False positive: detector goes off without real attack
- False negative: attack happens without detection
- Any detector design is a tradeoff between these (ROC curve)

#### Signature and anomaly weaknesses

