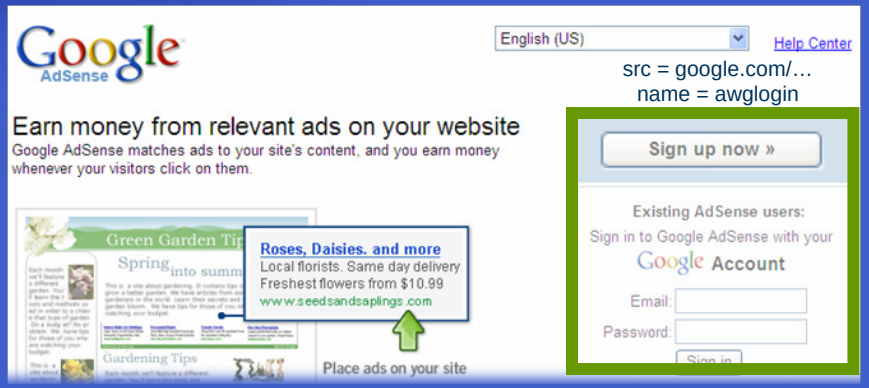
Securing Browser Frame Navigation and Communication

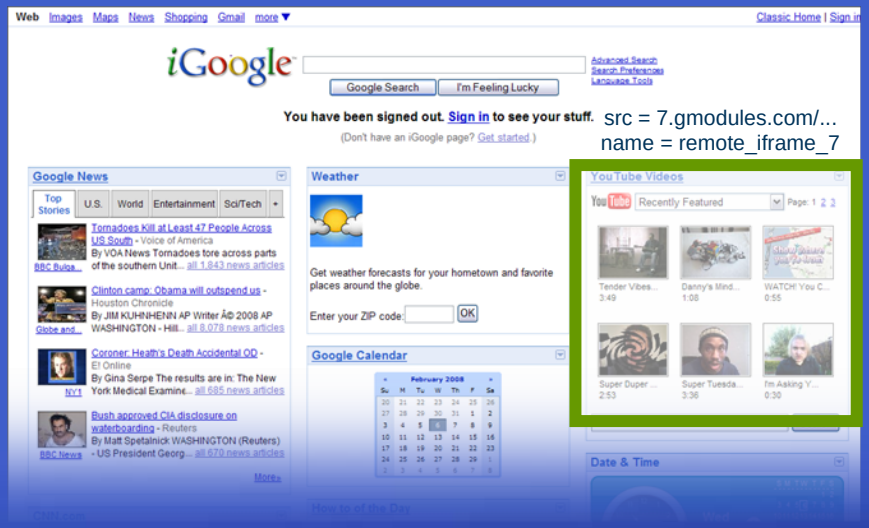
Collin Jackson

Joint work with Adam Barth and John C. Mitchell

Why use frames?

- Modularity
  - Brings together content from multiple sources
  - Client-side aggregation
- Isolation
  - Different frames can represent different principals
  - Can't script each other
  - Frame can draw only on its own rectangle
  - Easier than sanitization





Threat Model

- **Web attacker**
  - Controls attacker.com ($5)
  - Can obtain SSL/TLS certificate for attacker.com ($0)
  - User visits attacker.com
  - Optional additional assumption:

    Gets to embeds a malicious gadget (ad) on integrator site
- Stronger threat models
  - Network attacker: Can inspect or corrupt traffic
  - Malware attacker: Already escaped the from browser

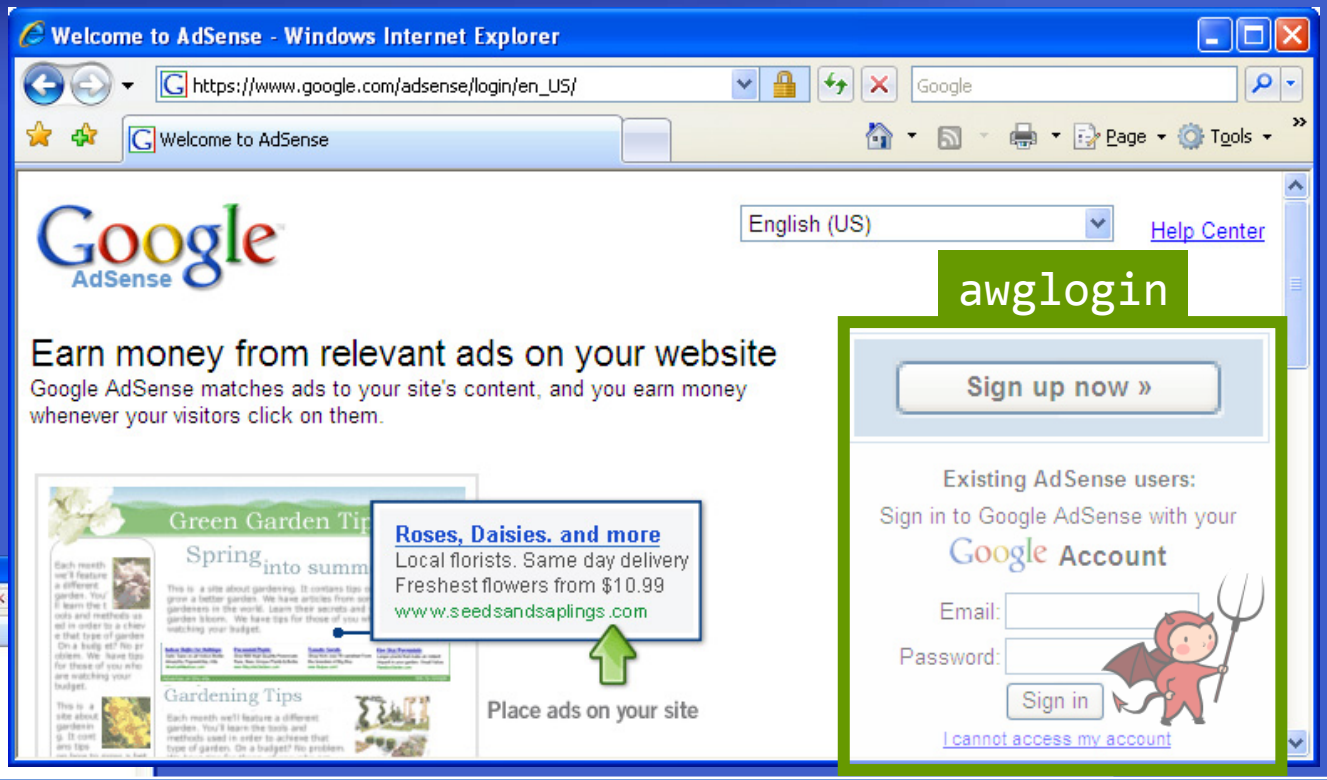
Frame Navigation

- Who decides a frame's content?

Permissive Policy

A frame can navigate any frame.

Guninski Attack

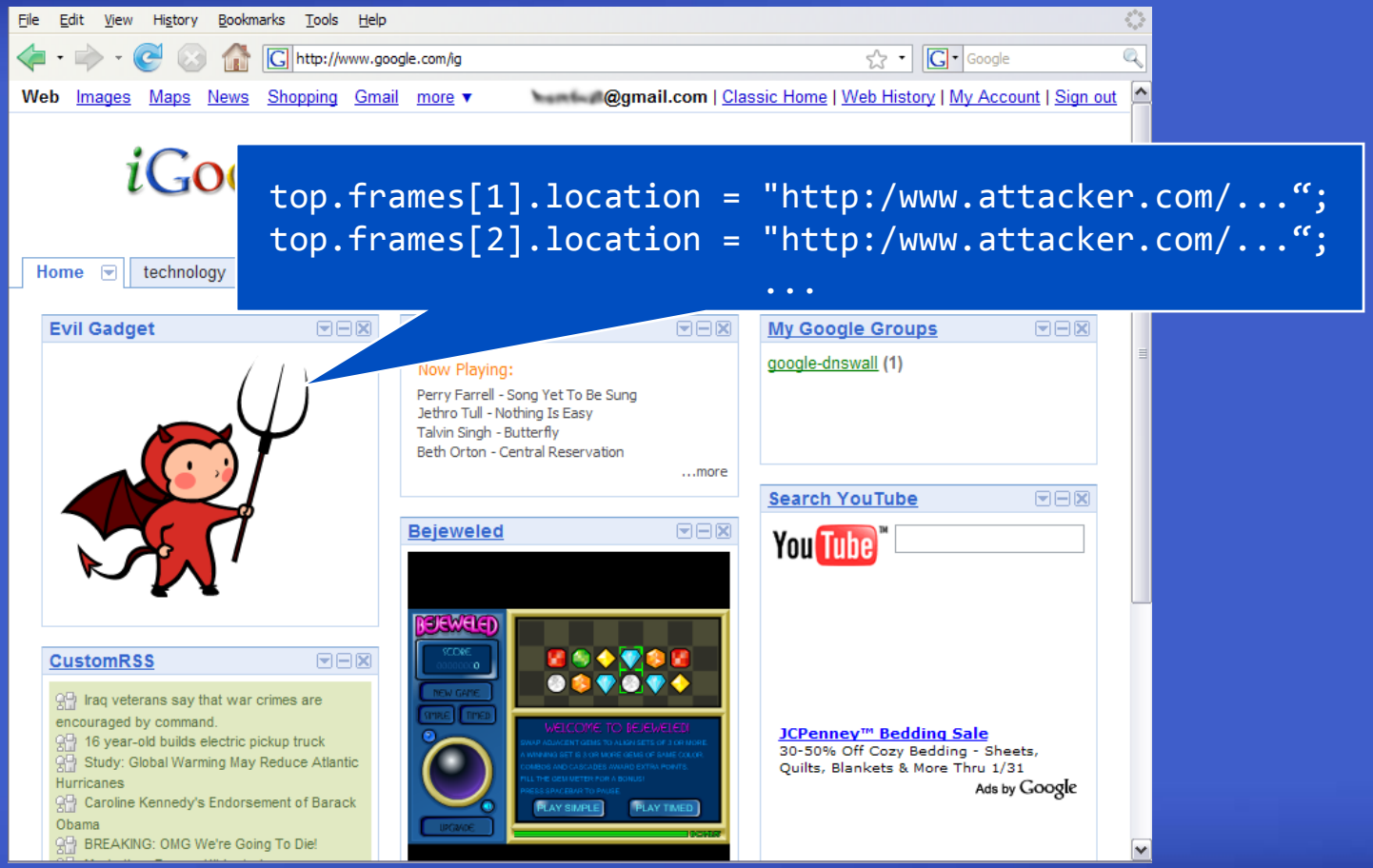



```
window.open("https://www.attacker.com/...", "awglogin")
```

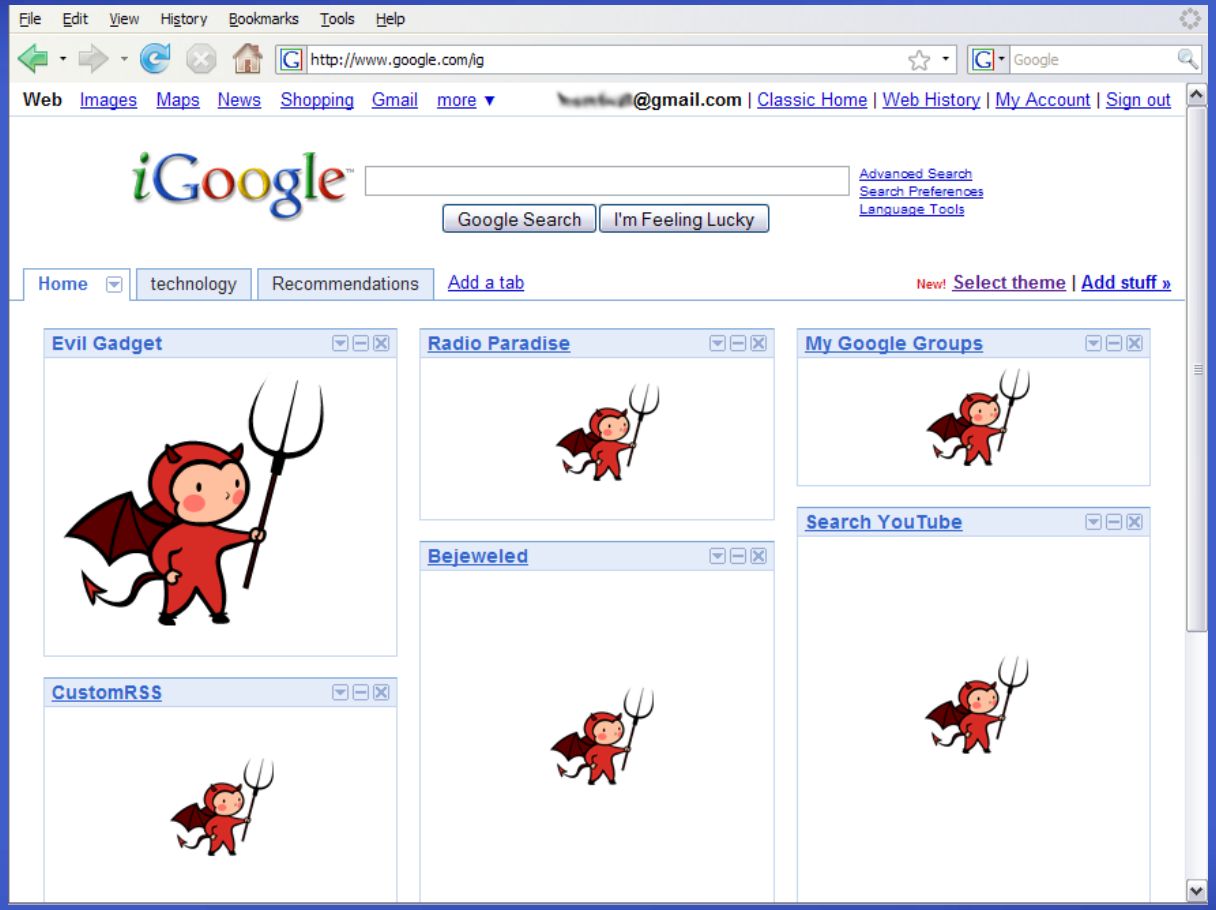
Window Policy

A frame can navigate
frames in its own window.

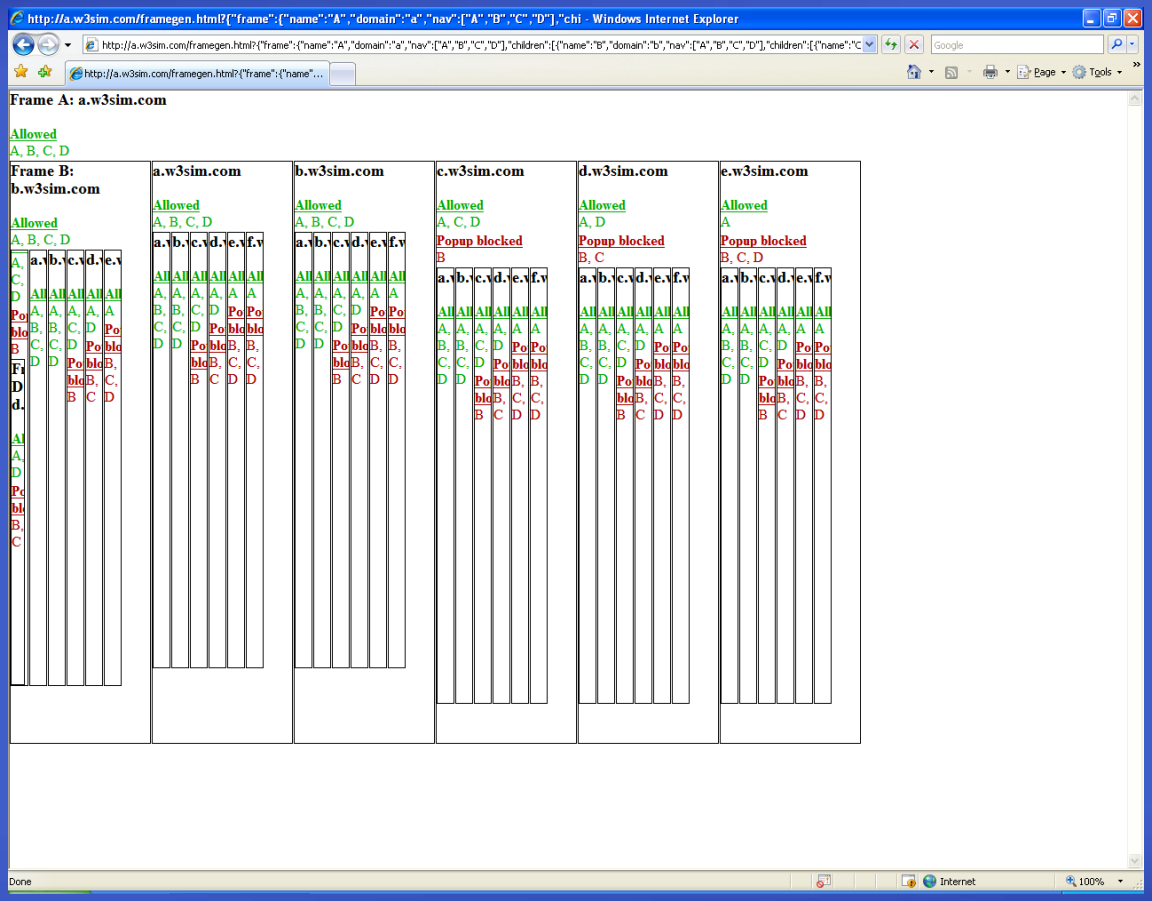
Gadget Hijacking



Gadget Hijacking

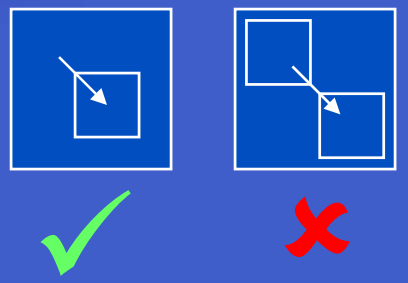


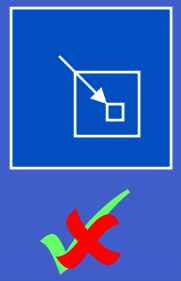
Policy Testing



Parent Policy

A frame can navigate its children.





Ancestor Policy

A frame can navigate its descendants.

Frame Navigation Policies

Browser	Policy	Propagation
 IE 6 (default)	Permissive	N/A
 IE 6 (option)	Parent	No
 IE7 (no Flash)	Ancestor	Yes
 IE7 (with Flash)	Permissive	N/A
 Firefox 2	Window	Sometimes
 Safari 2	Permissive	N/A

Frame Navigation Policies

Browser	Policy	Propagation
 IE7 (no Flash)	Ancestor	Yes
 IE7 (with Flash)	Ancestor	Yes
 Firefox 3	Ancestor	Yes
 Safari 3	Ancestor	Yes

Frame Communication

Fragment Identifier Messaging

- Send information by navigating a frame
  - http://gadget.com/**#hello**
- Navigating to fragment doesn't reload frame
  - No network traffic, but frame can read its fragment
- Not a secure channel
  - Confidentiality 
  - Integrity 
  - Authentication 

Fix: Improve the protocol

- Proposed Needham-Schroeder-Lowe

$$
\begin{aligned}
A &\rightarrow B : N_A, \mathrm{URI}_A \\
B &\rightarrow A : N_A, N_B, \mathrm{URI}_B \\
A &\rightarrow B : N_B \\
&\ldots \\
A &\rightarrow B : N_A, N_B, \mathrm{Message}_i \\
B &\rightarrow A : N_A, N_B, \mathrm{Message}_j
\end{aligned}
$$

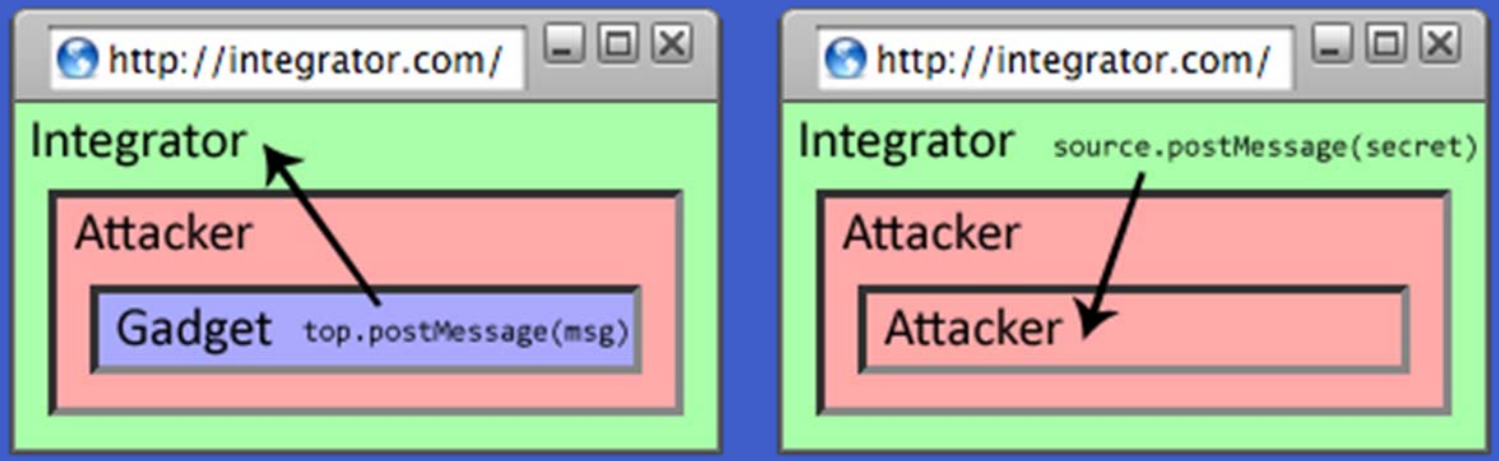
- Adoption
  - Microsoft: Windows Live Channels library
  - IBM: OpenAjax Hub 1.1

postMessage

- New API for inter-frame communication
- Supported in latest betas of many browsers

- Not a secure channel
  - Confidentiality ✗
  - Integrity ✓
  - Authentication ✓

Reply Attack



Fix: Improve the API

- Let the sending specify the recipient
  - frame[0].postMessage(“Hello”, “http://gadget.com”)
  - Can omit argument if confidentiality not required
- Adoption
  - Firefox 3
  - Internet Explorer 8
  - Safari 3.1

Conclusion

- All proposals deployed to real users
- Frame isolation
  - Improved frame navigation policy
    - Fixed Guninski and Gadget Hijacking
  - Drive-by-downloads still a concern…

- Frame communication
  - Secured fragment identifier messaging
  - Secured new postMessage API

Microsoft IBM