Secure Applications without Secure Infrastructures

Dieter Gollmann
Hamburg University of Technology

TUHH
Technische Universität Hamburg-Harburg

Critical infrastructures

- We have to come to rely on the Internet to an extent that life without it becomes difficult to image.
  - Air travel: no more paper tickets, only e-tickets since 2008; booking via web sites.
  - Conference registration: via web sites
  - Payment: credit card details entered on web sites; PayPal.
  - Communication: via email
  - Plus e-banking, e-commerce, e-government, SCADA, …
- Internet & web have become critical infrastructures.

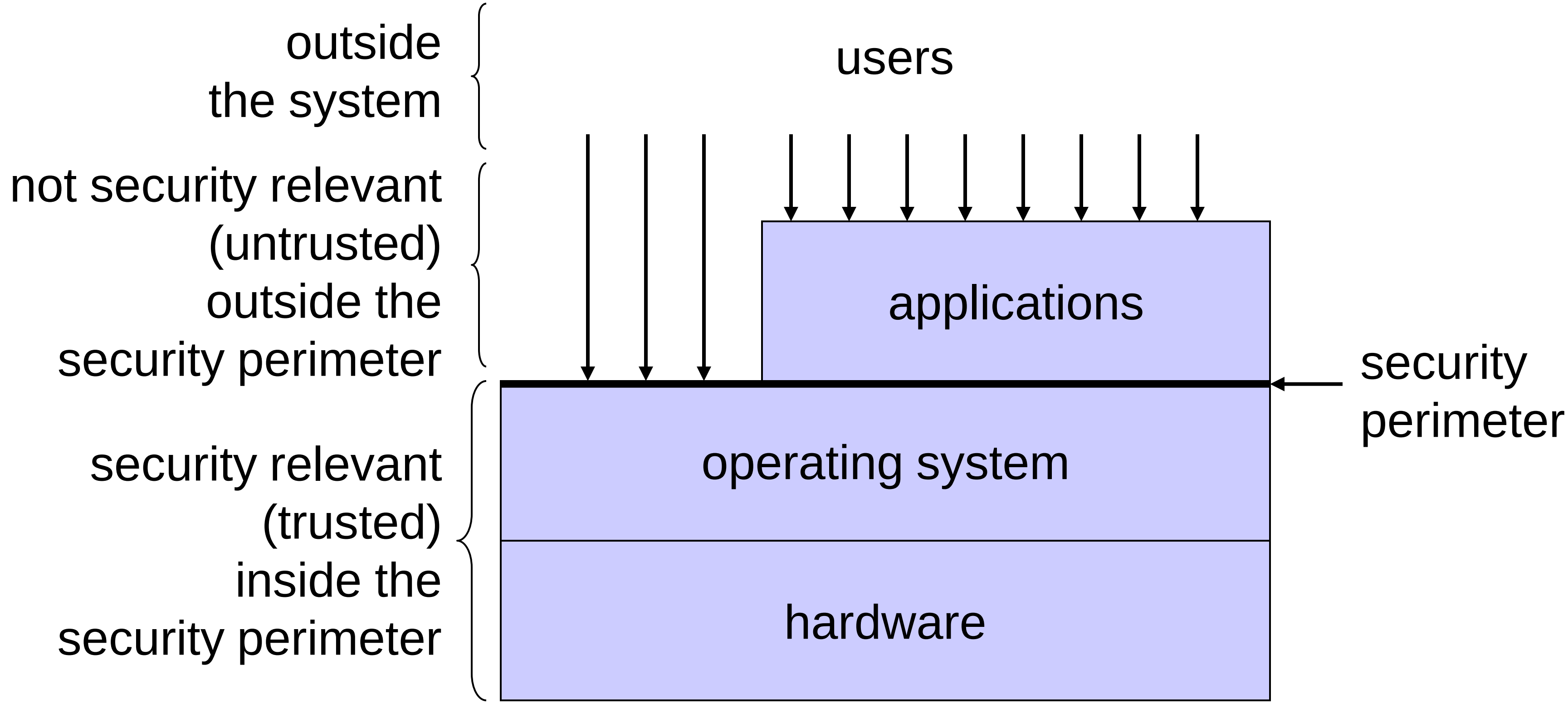
Do we have to secure this critical infrastructure?

Infrastructure security

- Historically, computer and communications security are infrastructure security.
- Computer security = operating system security: O/S is the infrastructure for users and applications.
  - Provides process isolation, access control, …
  - Once data are with the application the job is done.
- Communications security = secure channels: infrastructure carrying data from sender to receiver.
  - Once data are with the receiver the job is done.

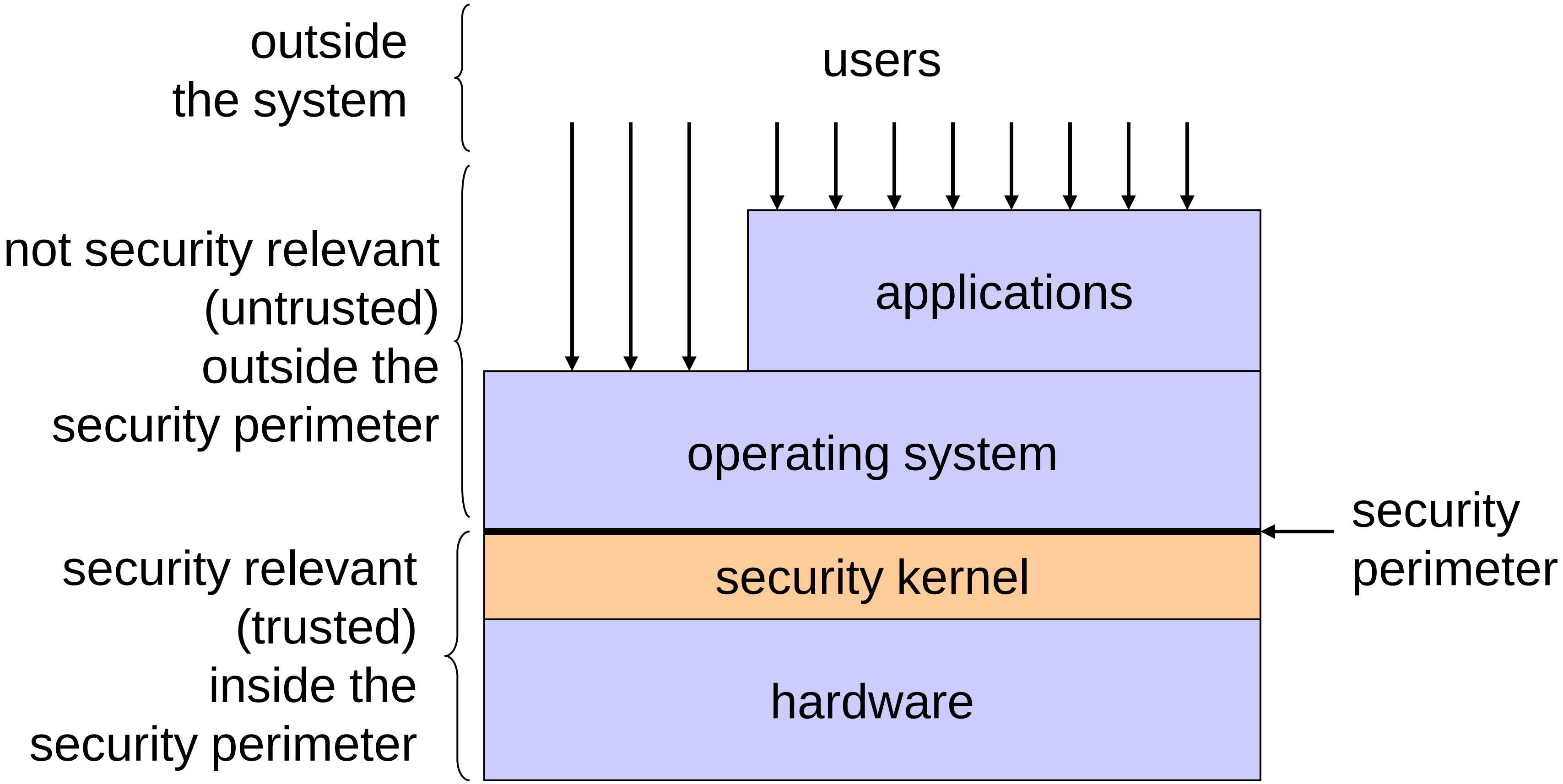
Computer security, 1988

Morrie Gasser, Building a Secure Computer System, Van Nostrand Reinhold



Computer security+, 1988

Morrie Gasser, Building a Secure Computer System, Van Nostrand Reinhold



outside
the system

users

not security relevant
(untrusted)
outside the
security perimeter

applications

operating system

security perimeter

security kernel

hardware

security relevant
(trusted)
inside the
security perimeter

Security strategy – the past

- Higher security by small, verifiable security kernel that implements the reference monitor.
- Security guaranteed by the lower system layers (the infrastructure).
- Applications need not be trusted.
- Security evaluation (Orange Book): focus on infrastructures managed by IT professionals.
- The defenders retreat to the security kernel.

Communications security

- Focus on the design of secure channels: IPsec, IPsec over IPsec, SSL/TLS, …
- Infrastructure services at network and transport layer.
- Protect against attackers (“spies”) who can read, modify, delete, insert, replay messages.
- Job done once messages are delivered.
- No protection against attacks in the end systems (“hackers”).

Secure channels

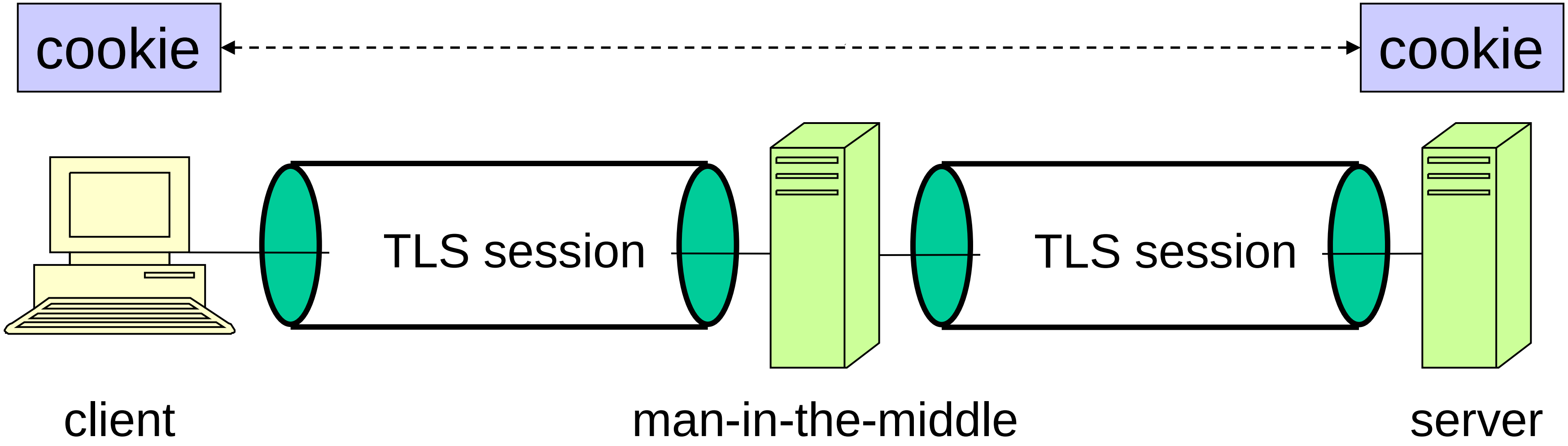
- Formal security proofs for some protocols exist (TLS).
- Challenge: combining channels at different layers.
- Attacks become possible when there is a mismatch of channel end points.
- Tunnels at different layers by definition have different end points!
- ‘Server’ and ‘client’ are dangerous simplifications.

Case study: TLS & e-commerce

- Server is authenticated to client when setting up the TLS tunnel (unilateral authentication).
- User has account at server, protected by password.
- Password based authentication within SSL tunnel.
  - Attacker cannot sniff user password.
- Server authenticates user, returns HTTP cookie.
- Client browser includes cookie in further requests to server; requests are attributed to authenticated user.

Man-in-the-middle attack

Attacker splices two TLS sessions.



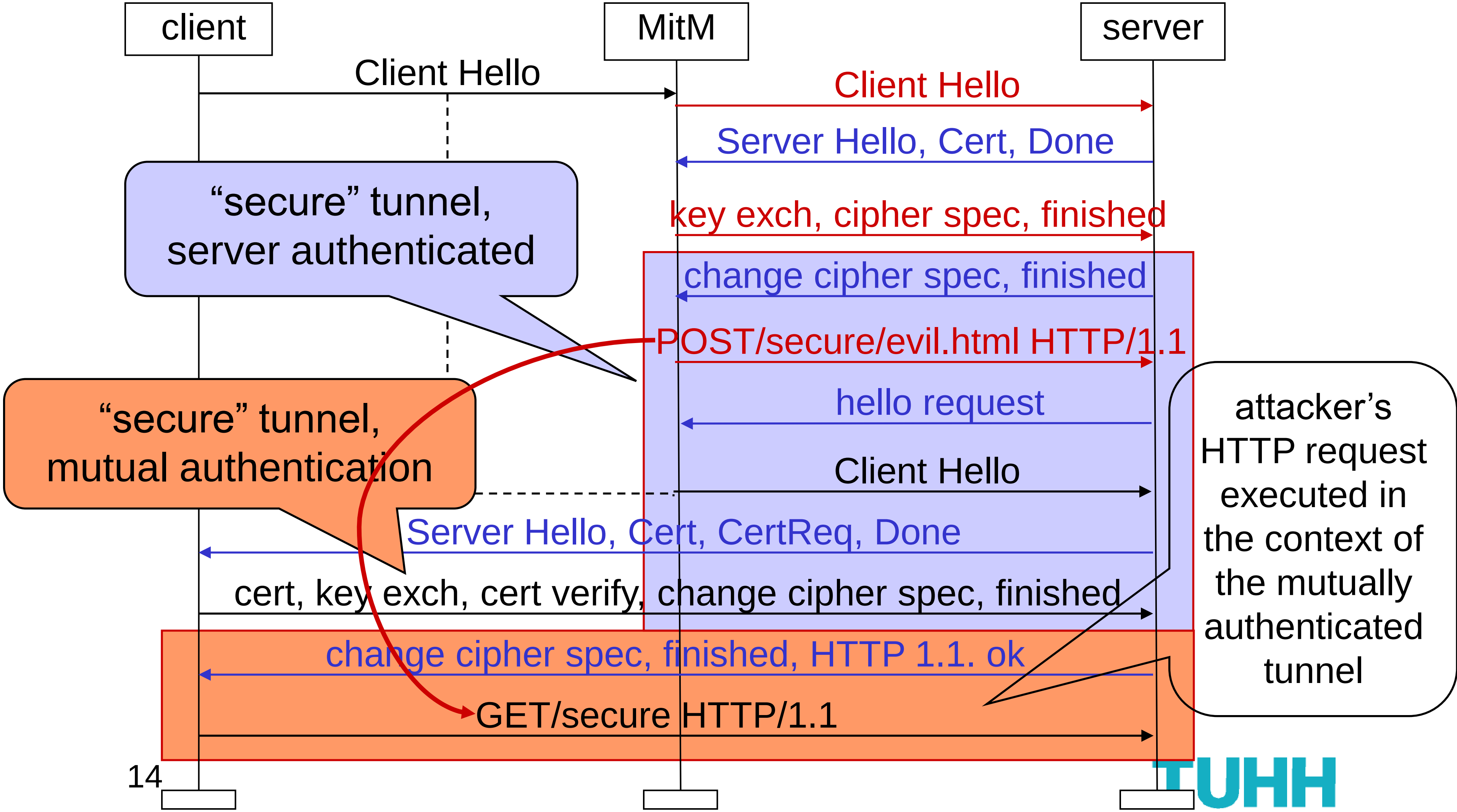
As a defence, bind cookie to TLS session.

Recent TLS security scare

- “Flaw” of TLS widely reported.
  - Marsh Ray, Steve Dispensa: Renegotiating TLS, 4.11.2009
- Background: TLS employed for user authentication when accessing a secure web site.
- Common practice for web servers to let users start with an anonymous TLS session.
- Request for a protected resource triggers TLS renegotiation; mutual authentication requested when establishing the new TLS tunnel.

Bugtraq ID 36935: “Multiple vendors’ TLS protocol implementations are prone to a security vulnerability related to the session-renegotiation process.”

Recent https-Problem


client
MitM
server
Client Hello
Client Hello
Server Hello, Cert, Done
key exch, cipher spec, finished
change cipher spec, finished
POST/secure/evil.html HTTP/1.1
hello request
Client Hello
Server Hello, Cert, CertReq, Done
cert, key exch, cert verify, change cipher spec, finished
change cipher spec, finished, HTTP 1.1. ok
GET/secure HTTP/1.1
"secure" tunnel, server authenticated
"secure" tunnel, mutual authentication
attacker's HTTP request executed in the context of the mutually authenticated tunnel

Comment

- Application developers using session renegotiation for user authentication made assumptions about renegotiation I failed to spot in RFC 5246.
- Fact: typical use case for renegotiation suggests that the new session is a continuation of the old session.
  - Plausible assumptions about a plausible use case are treated as a specification of the service.
- Fix: TLS renegotiation cryptographically tied to the TLS connection it is performed in (RFC 5746).
  - TLS adapted to meet expectations of application.
- This had really been an application layer problem.
  - State at server persists over two TLS tunnels; attacker sends a malicious partially complete command in the first tunnel.

CompSec & CommSec – today

- Cross-Site Scripting
  - #1 in 2007 OWASP Top Ten Vulnerabilities
  - #1 in CVE seit 2005
- SQL Injection
  - #2 in CVE seit 2006
- Cross-Site Request Forgery
- JavaScript Hijacking
- DNS Rebinding
- Code injection attacks,
- …

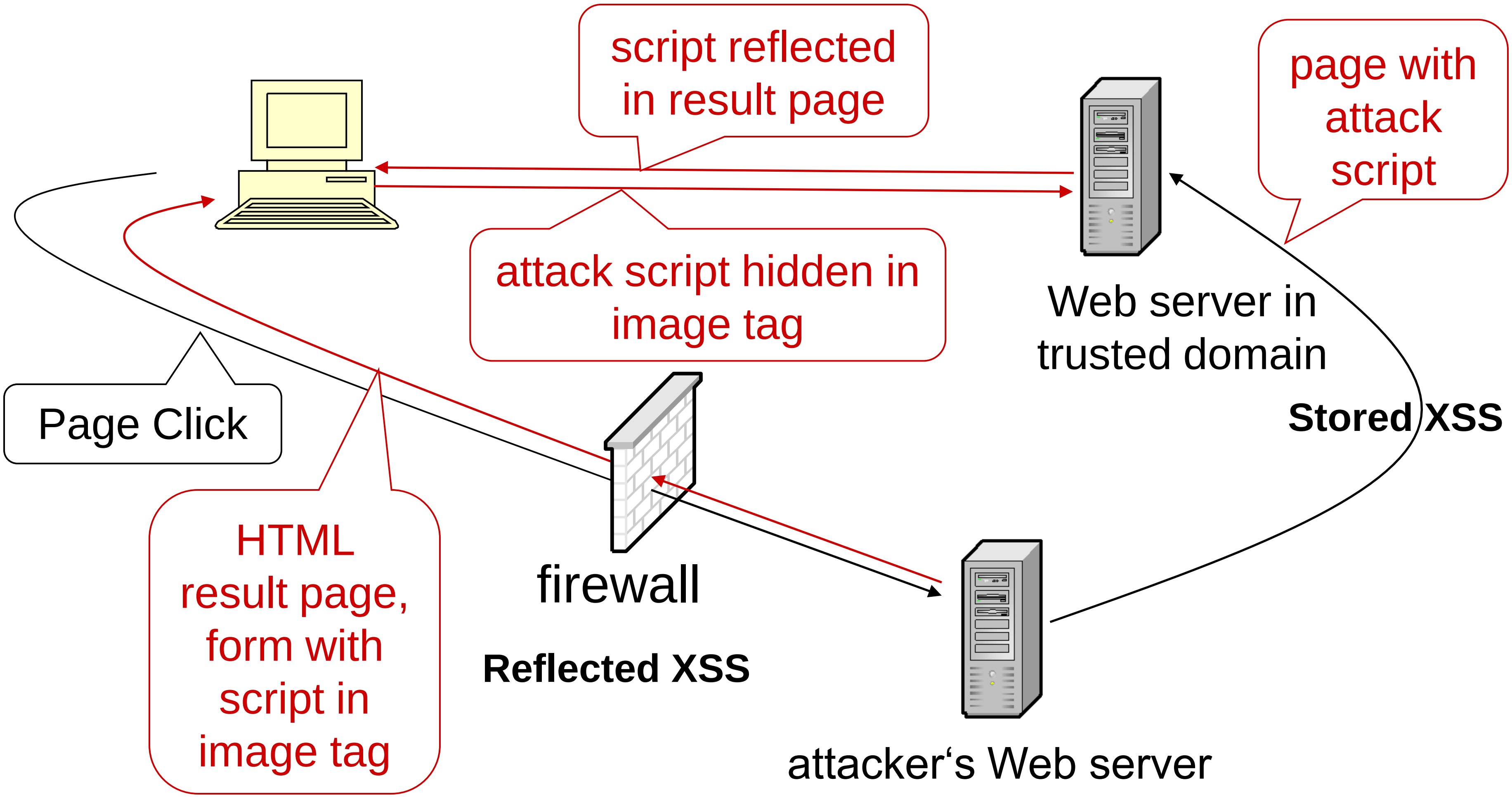
Client-side scripting (Web 1.0)

- Response page may contain scripts (often written in JavaScript) that will be executed in browser.
- Attack might be launched by placing malicious script in a response page.
- Browsers enforce a JavaScript same origin policy:
  - Script may get access to its own DOM only.
  - Script may only connect to the DNS domain it came from.

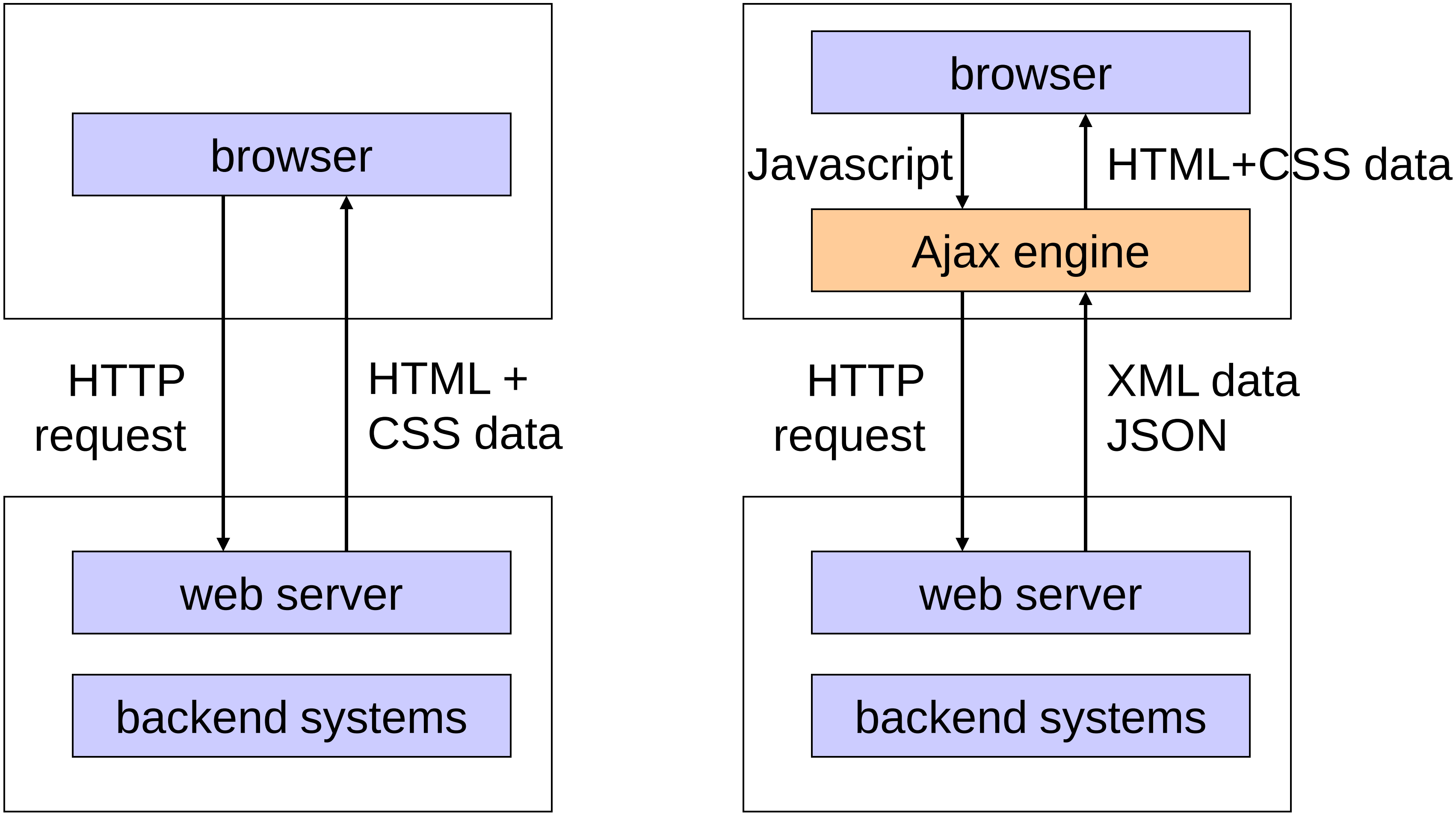
Cross Site Scripting – XSS

- Elevation of privilege attack involving attacker, client (victim), server (‘trusted’ by client).
  - Trust: code in pages from server executed with higher privileges at client (origin based access control).
- Attacker places malware on a page at server (stored XSS) or in a hidden form in a page on the attacker’s host (reflected XSS).
  - Stored XSS: e.g. via a bulletin board.
  - Reflected XSS: e.g. via a search term.
- Malware returned by server to client in result page; executed at client with permissions of trusted server.

Cross-site scripting


script reflected in result page
page with attack script
attack script hidden in image tag
Web server in trusted domain
Stored XSS
Page Click
HTML result page, form with script in image tag
firewall
Reflected XSS
attacker‘s Web server

Web 1.0 & Web 2.0


browser
HTTP request
HTML + CSS data
web server
backend systems
browser
Javascript
HTML+CSS data
Ajax engine
HTTP request
XML data JSON
web server
backend systems

JavaScript hijacking (Web 2.0)

- Exploits that a client side Ajax engine sitting between browser and web server performs many actions automatically.
- Exploits the fact that Web 2.0 applications may use JavaScript (JSON) for data transport.
- JSON string is a serialized JavaScript object, turned back into an object with JavaScript by calling eval() with the JSON string as the argument using the JavaScript object constructor.
- Attack rewrites constructor to disclose confidential data to attacker; bypasses same origin policy.

DNS rebinding

- Same origin policy: Script can only connect back to the server it was downloaded from.
- To make a connection, the client’s browser needs the IP address of the server.
- Authoritative DNS server resolves DNS names in its domain to IP addresses.
- The client’s browser ‘trusts’ the DNS server when enforcing the same origin policy.
- Trust is Bad for Security!

DNS rebinding attack

- Client visits attacker.org; attacker's DNS server resolves this name to attacker's IP address with short time-to-live.
- Attack script waits before connecting to attacker.org.
- Binding at browser has expired; new request for IP address of attacker.org, now bound to target address.
- Defence: Don't trust the DNS server on time-to-live; pin host name to original IP address;
  - J. Roskind: Attacks against the Netscape browser. in RSA Conference, April 2001.

DNS rebinding attack

- More sophisticated authorisation system: client browser refers to policy obtained from DNS server when deciding on connection requests.
- A malicious DNS server can thus authorize connection to the victim.
- Defence: Double check policy with the host at the IP address the DNS name is being resolved to.
  - Related to reverse DNS lookup.
  - Similar defences against bombing attacks in network security.
- Digital signatures do not help against DNS rebinding.

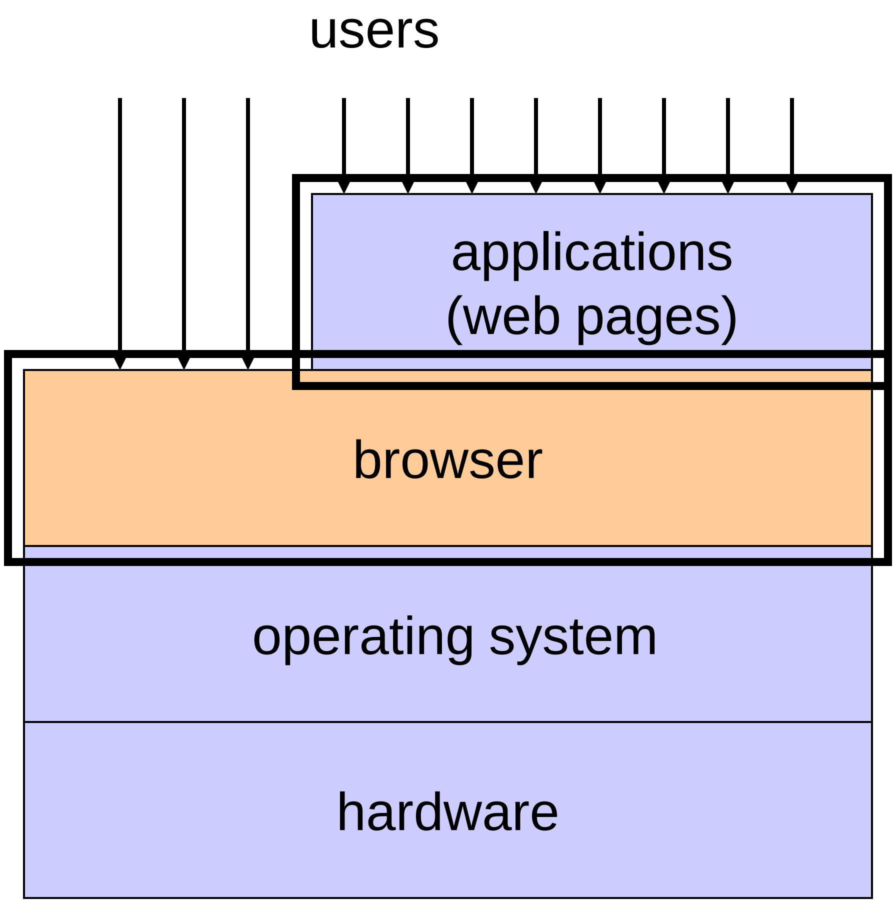
Computer security – today

- Same Origin Policies: who may read cookies, where may a script connect to?
- Reference monitor in the browser (sandbox).
- The attacks just sketched exploit vulnerabilities in this reference monitor.

- Web pages filter inputs, e.g. to defend against SQL injection attacks.
- Reference monitor moves to the application layer; application developers have to include relevant filters.

Computer security – today

security perimeter?



Summary

- Mechanisms in the traditional security kernel hardly defend against today’s new attacks.
- Traditional secure channels hardly defend against today’s new attacks.
- The line of defence against current attacks moves up to the application layer.
- Security mechanisms are moving out of the infrastructure into the applications.
- Defenders meet the attackers in front of the gates.

Research challenges

- Access control – mechanisms: modelling the mechanisms in the browser
  - considering Web 2.0, plug-ins, mashups, …
  - considering new mechanisms for authenticating data origin.
- Access control – policies: specification and enforcement of Cross Domain Policies.
  - Ajax Cross Domain Policies
  - HTTP Access Control Headers for Cross-Domain Policies, http://www.w3.org/TR/access-control/
  - Re-evaluating the role of the end user in setting policies.

Research challenges

- Authentication: examining necessary conditions on ‘authentication’.
  - Authentication can go beyond ‘corroborating the identity of the sender.
  - Would ‘recognition’ or ‘know thyself’ suffice?
- Interaction between protocol layers: understanding how to build tunnels in tunnels.
  - Re-evaluating which security services should be provided by the lower layers (the infrastructure) and which are provided within the application.

- We do not have to secure the infrastructure but the critical applications.
- Securing the critical infrastructure is neither sufficient nor necessary.
- Thank you very much for your attention.