

Security

Chapter 18

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

5 Security Best Practices

6 Common Threat Vectors

7 Summary

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

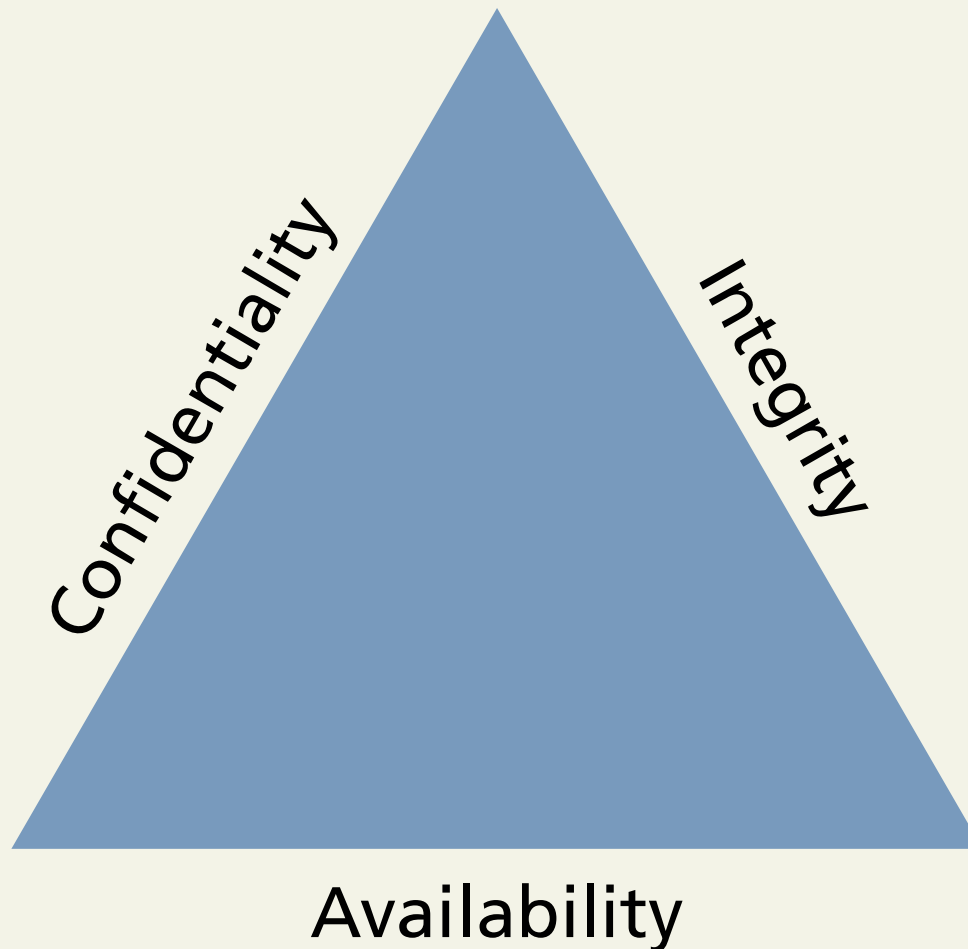
5 Security Best Practices

6 Common Threat Vectors

7 Summary

Security Principles

Information Security



Security Principles

Risk Assessment and Management

Actors, Impact, Threats, and Vulnerabilities

Security Principles

Actors

Internal actors are the people who work for the organization. They can be anywhere in the organization from the cashier through the IT staff, all the way to the CEO.

External actors are the people outside of the organization. They have a wide range of intent and skill, and they are the most common source of attacks.

Partner actors are affiliated with an organization that you partner or work with. If your partner is somehow compromised, there is a chance your data is at risk as well because quite often partners are granted some access to each other's systems (to place orders, for example).

Security Principles

Impact

- **A loss of availability** prevents users from accessing some or all of the systems.
- **A loss of confidentiality** includes the disclosure of confidential information to a (often malicious) third party
- **A loss of integrity** changes your data or prevents you from having correct data. This might manifest as an attacker hijacking a user session, perhaps placing fake orders or changing a user's home address.

Security Principles

Threats

Broadly, threats can be categorized using the STRIDE mnemonic

- **Spoofing**—The attacker uses someone else's information to access the system.
- **Tampering**—The attacker modifies some data in nonauthorized ways.
- **Repudiation**—The attacker removes all trace of their attack, so that they cannot be held accountable for other damages done.
- **Information disclosure**—The attacker accesses data they should not be able to.
- **Denial of service**—The attacker prevents real users from accessing the systems.
- **Elevation of privilege**—The attacker increases their privileges on the system thereby getting access to things they are not authorized to do.

Security Principles

Vulnerabilities

Vulnerabilities are the security holes in your system. The top five classes of vulnerability from the Open Web Application Security Project³ are:

1. Injection
2. Broken authentication and session management
3. Cross-site scripting
4. Insecure direct object references
5. Security misconfiguration

Security Principles

Security Policy

Usage policy defines what systems users are permitted to use, and under what situations.

Authentication policy controls how users are granted access to the systems.

Legal policies define a wide range of things including data retention and backup policies as well as accessibility requirements (like having all public communication well organized for the blind).

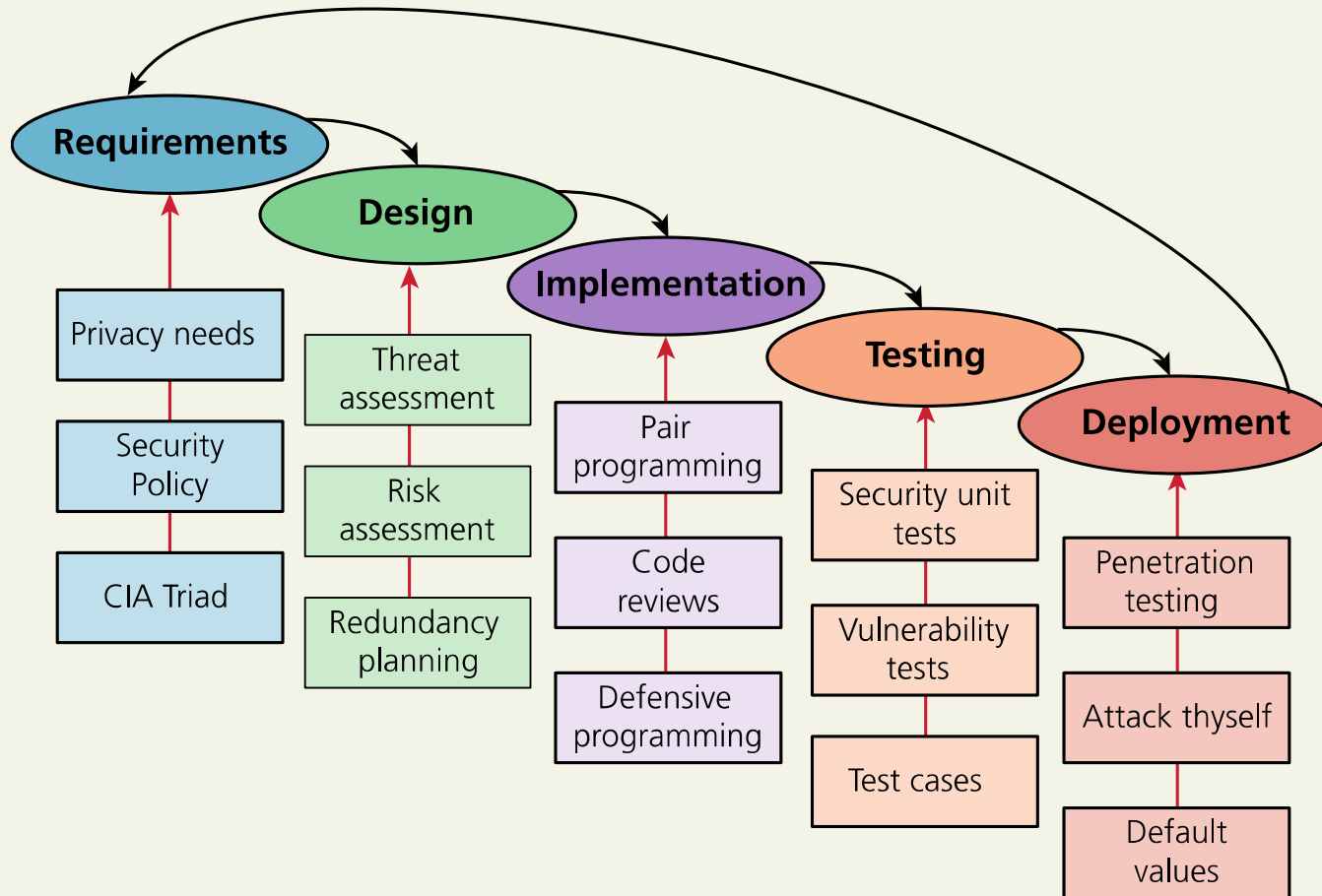
Security Principles

Business Continuity

- Admin Password Management
- Backups and Redundancy
- Geographic Redundancy
- Stage Mock Events
- Auditing

Security Principles

Secure by Design



Security Principles

Social Engineering

In security circles, software engineering takes on the meaning referring to the techniques used to manipulate people into doing something, normally by appealing to their baser instincts.

- Phishing Scams
- Security Theater

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

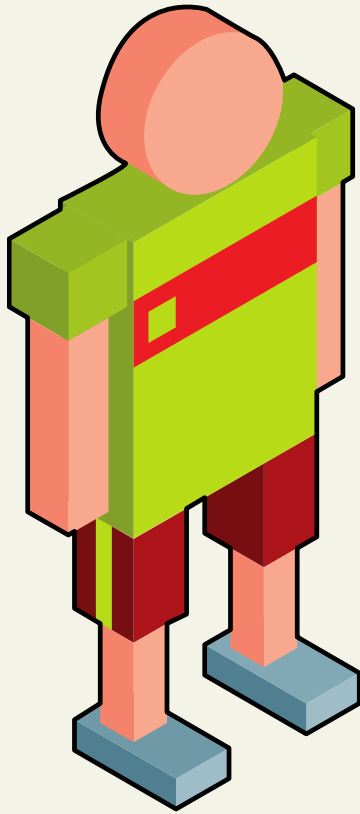
5 Security Best Practices

6 Common Threat Vectors

7 Summary

Authentication

Authentication Factors



What you **know** (Knowledge)

Passwords, PIN, security questions, ...



What you **have** (Ownership)

Access card, cell phone, cryptographic FOB, ...



What you **are** (Inherence)

Retinas, fingerprints, DNA, walking gait, ...

Authentication

Authentication Factors

- **Single-factor authentication** is the weakest and most common category of authentication system where you ask for only one of the three factors.
- **Multifactor authentication** is where two distinct factors of authentication must pass before you are granted access.

Authentication

HTTP Authentication

HTTP supports several different forms of authentication via the `www-authenticate` response header.

- HTTP Basic Authentication
- HTTP Digest Authentication
- Form-Based Authentication

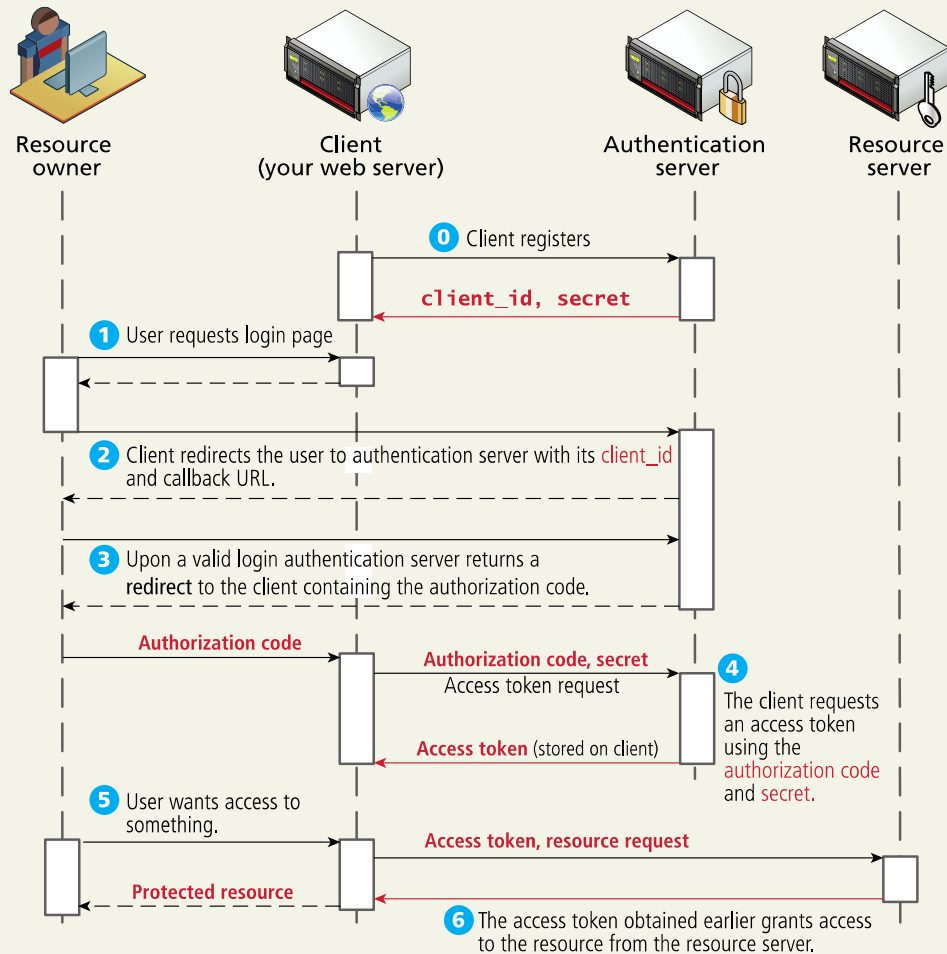
Authentication

Third-Party Authentication

Many popular services allow you to use their system to authenticate the user and provide you with enough data to manage your application

Authentication

Third-Party Authentication - OAuth



Authentication

Authorization

Authorization defines what rights and privileges a user has once they are authenticated. Some principles:

- Using a separate database user for read and write privileges on a database.
- Providing each user an account where they can access their own files securely.
- Setting permissions correctly so as to not expose files to unauthorized users.
- Using Unix groups to grant users permission to access certain functionality rather than grant users admin access.
- Ensuring Apache is not running as the root account (i.e., the account that can access everything).

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

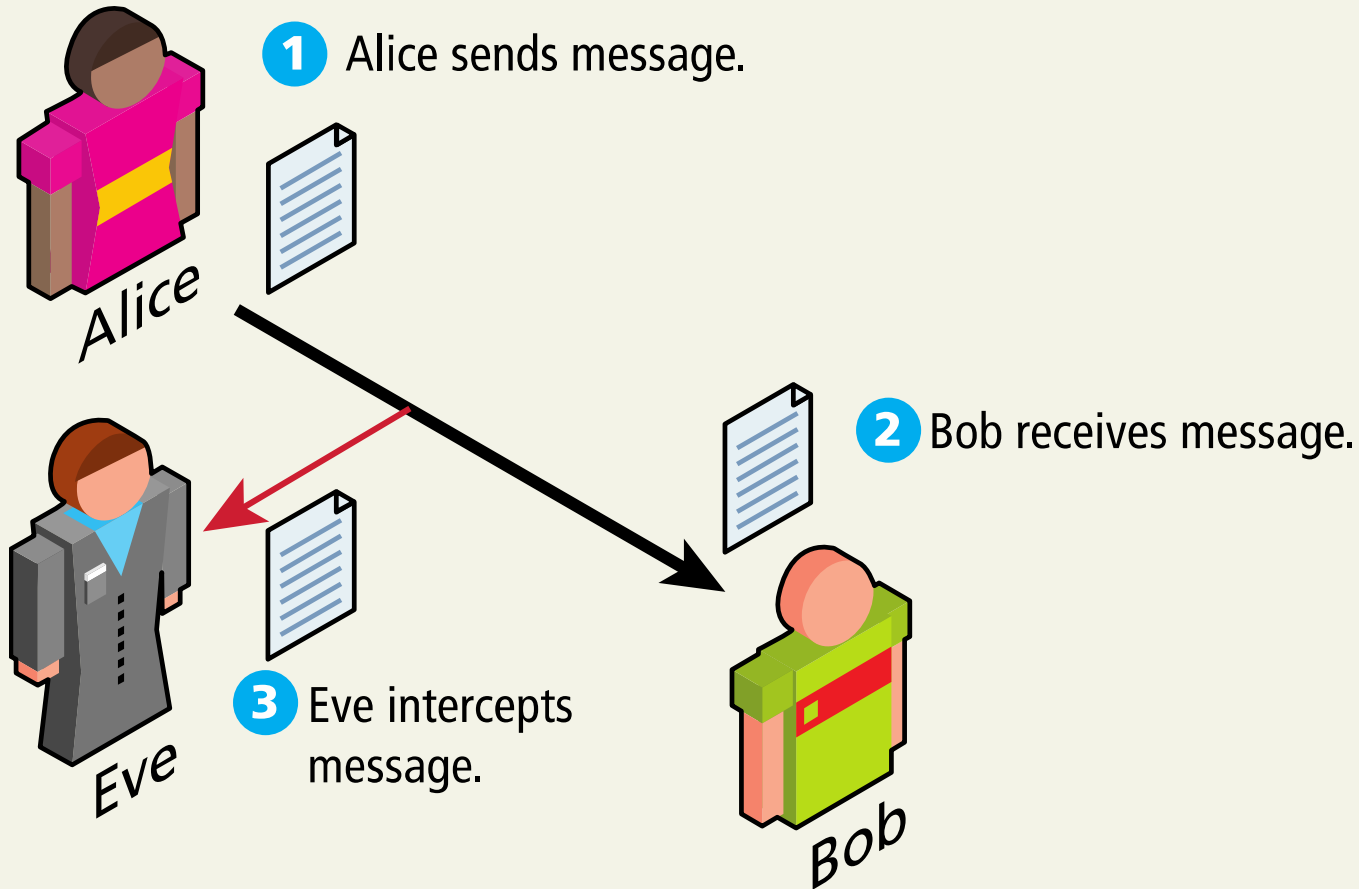
5 Security Best Practices

6 Common Threat Vectors

7 Summary

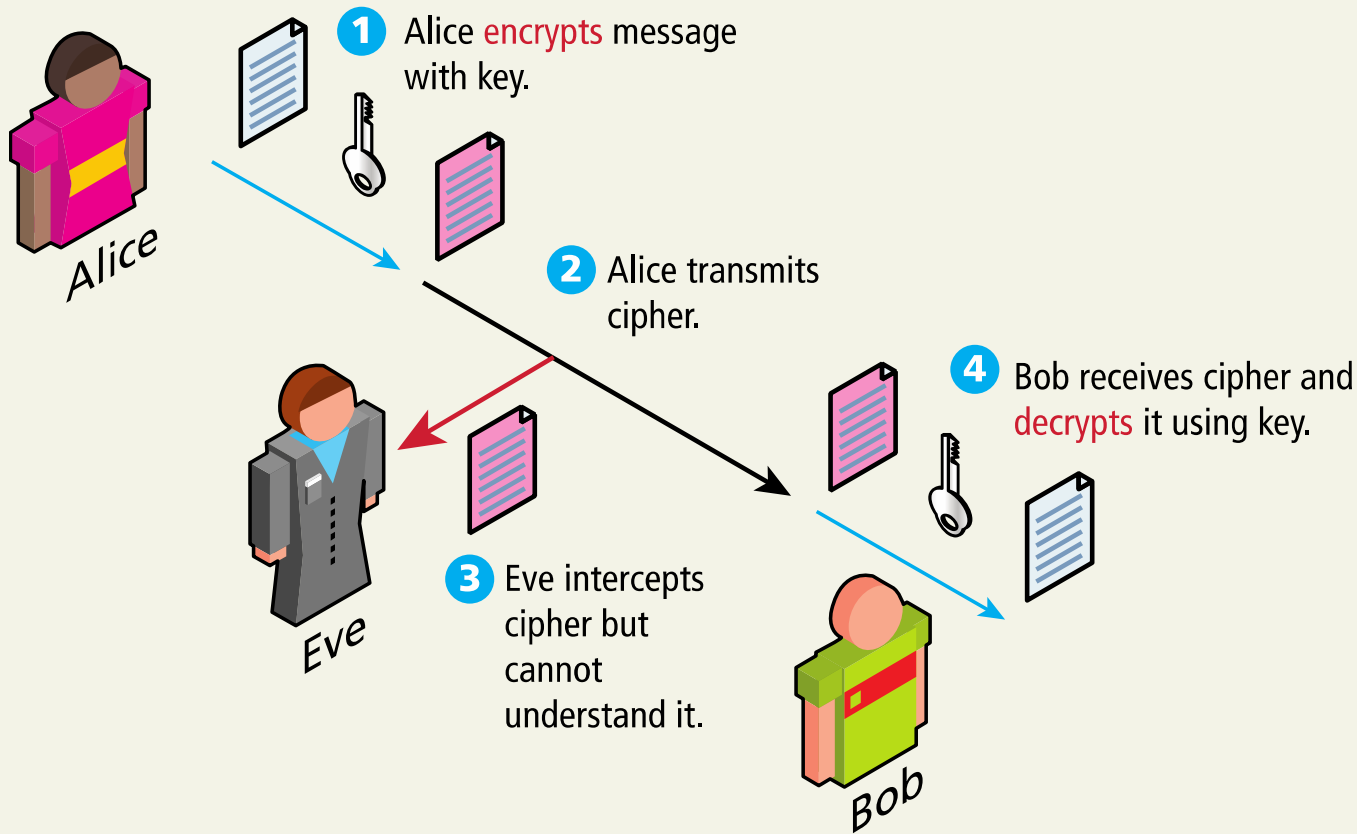
Cryptography

An overview



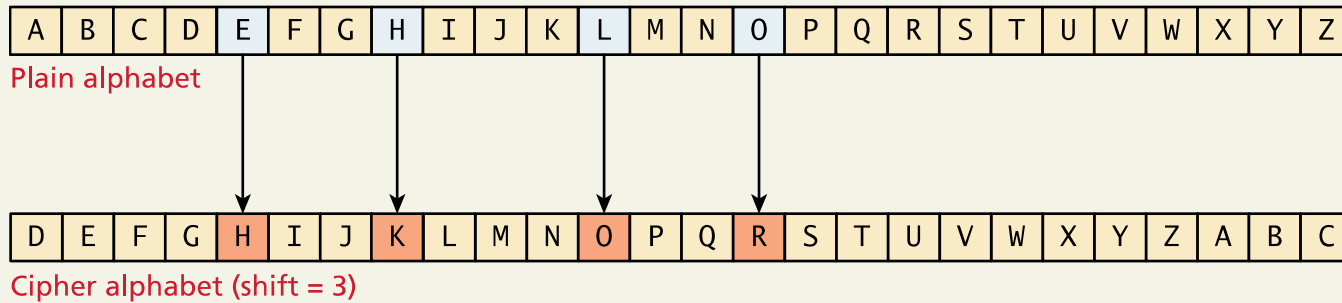
Cryptography

Symmetric encryption



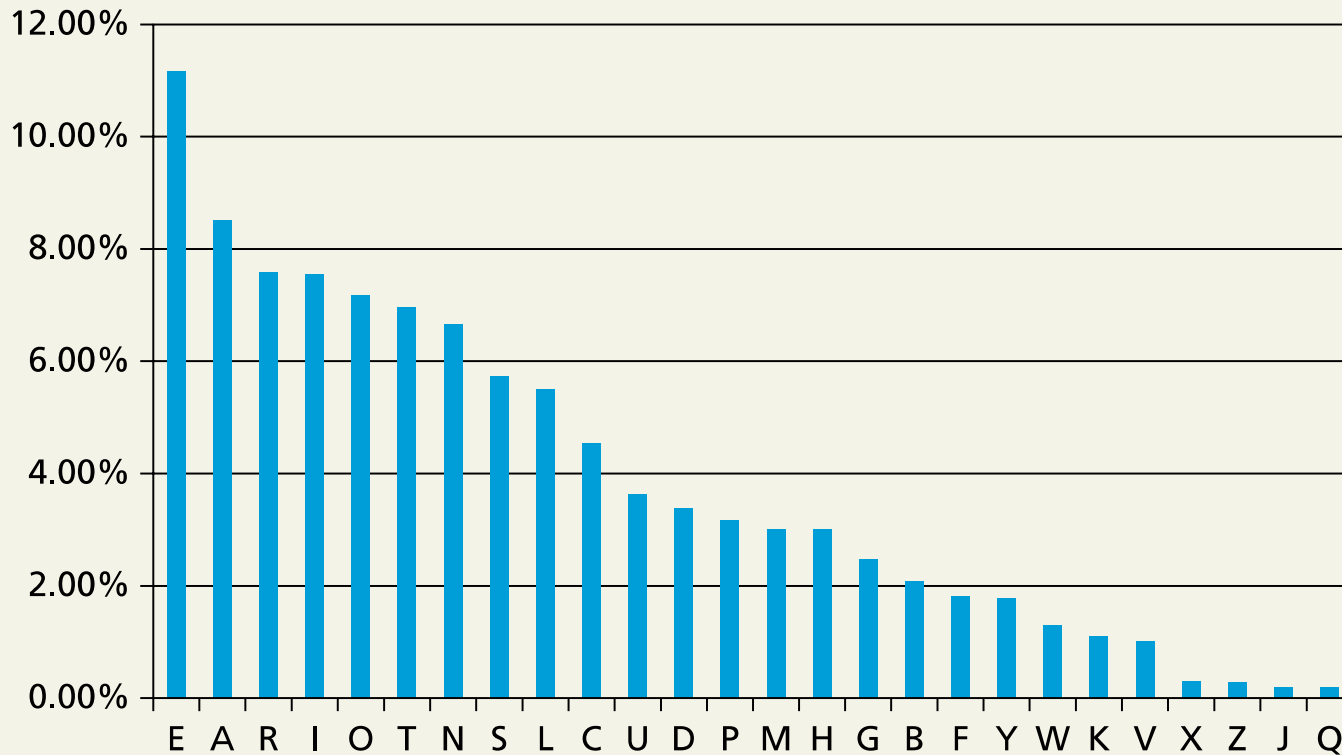
Cryptography

Substitution Ciphers



Cryptography

Ciphertext open to frequency analysis



Cryptography

Vigenere

Plain text message

H E L L O D E A R R E A D E R S

+

H O T D O G H O T D O G H O T D

=

Cipher

P T F P D K M P L V T H L T L W

-

H O T D O G H O T D O G H O T D

=

H E L L O D E A R R E A D E R S

Encrypt
(+)

KEY

H O T D O G

Decrypt
(-)

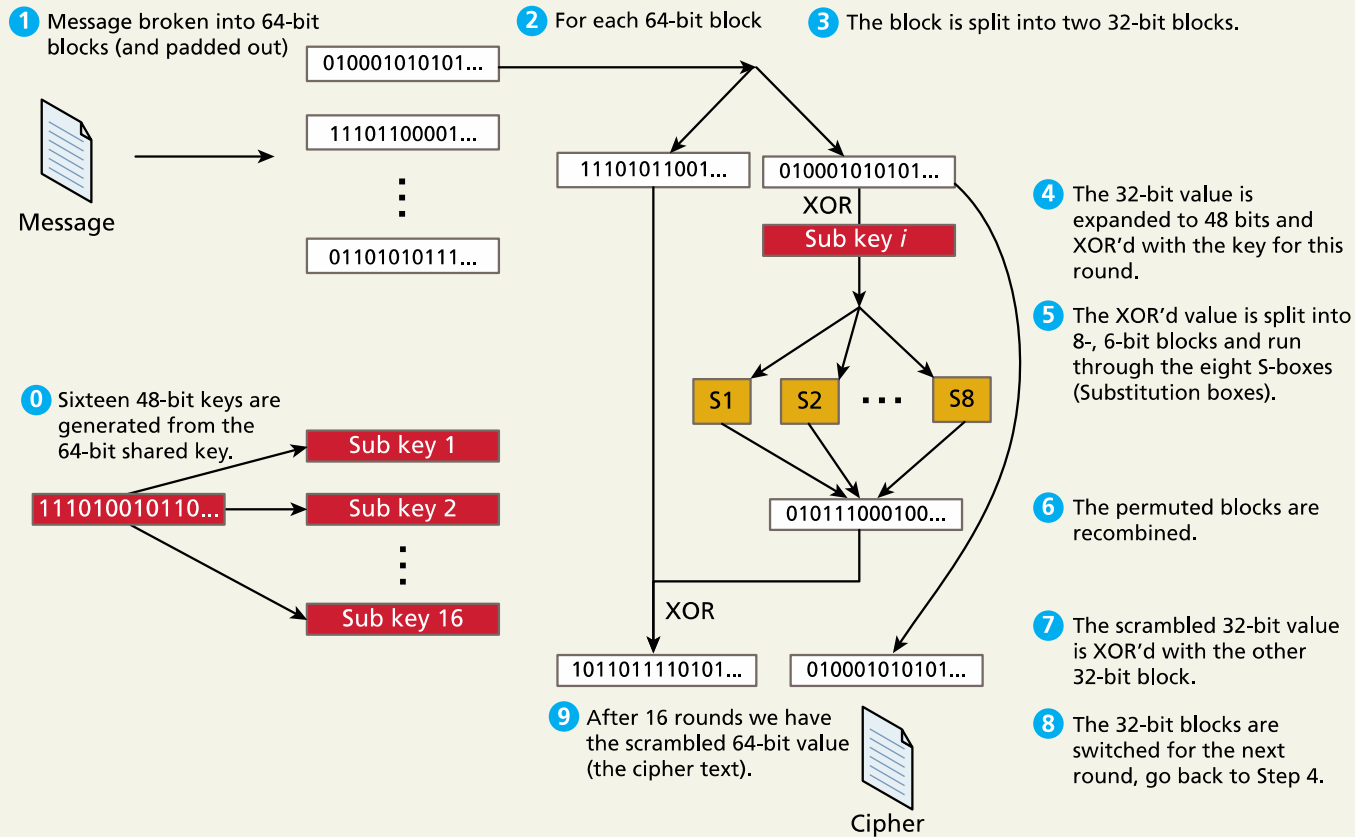
Cryptography

One Time Pad

Theoretically Perfect

Cryptography

Modern Block Ciphers



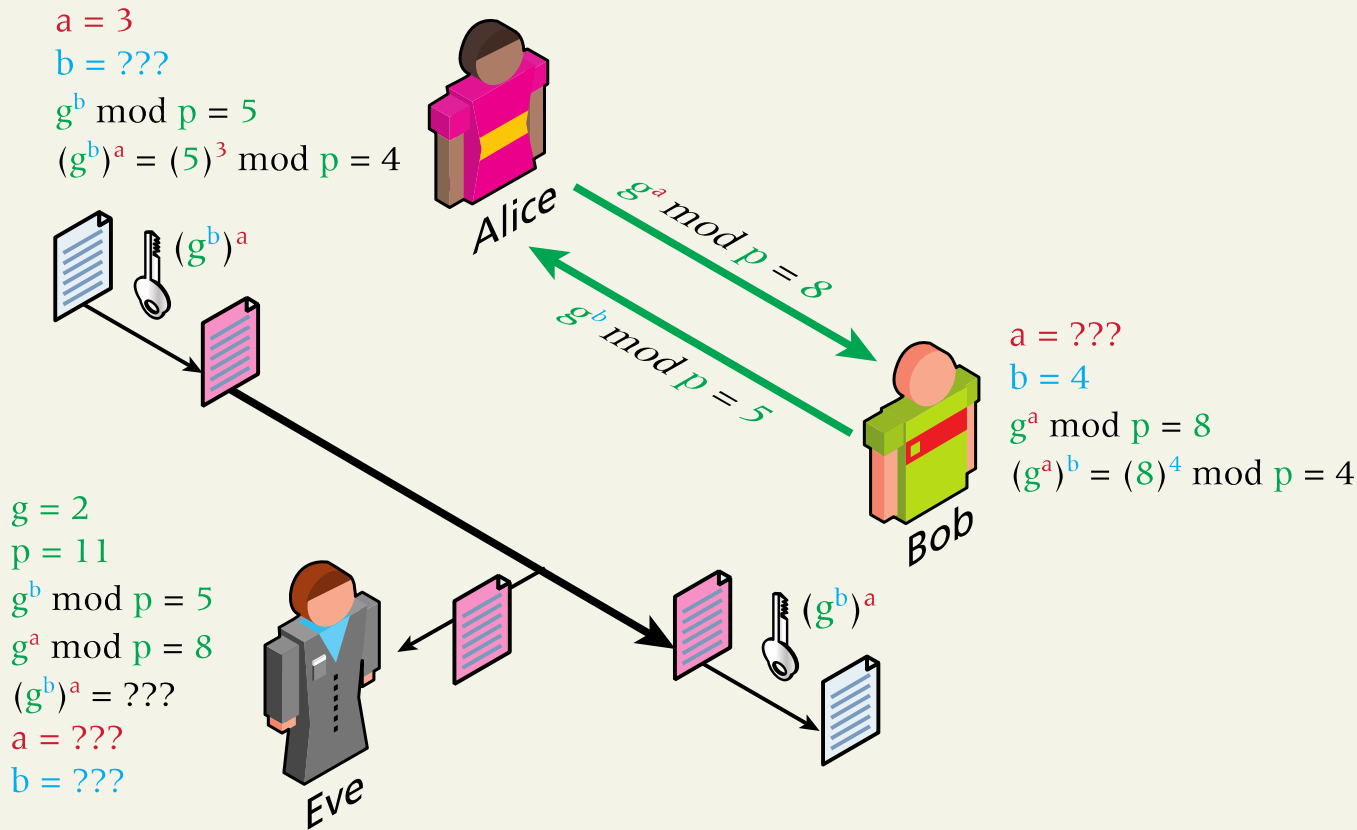
Cryptography

Public Key Cryptography

Public key cryptography (or asymmetric cryptography) solves the problem of the secret key by using two distinct keys: a public one, widely distributed and another one, kept private.

Cryptography

Public Key Cryptography – Diffie Hellman



Cryptography

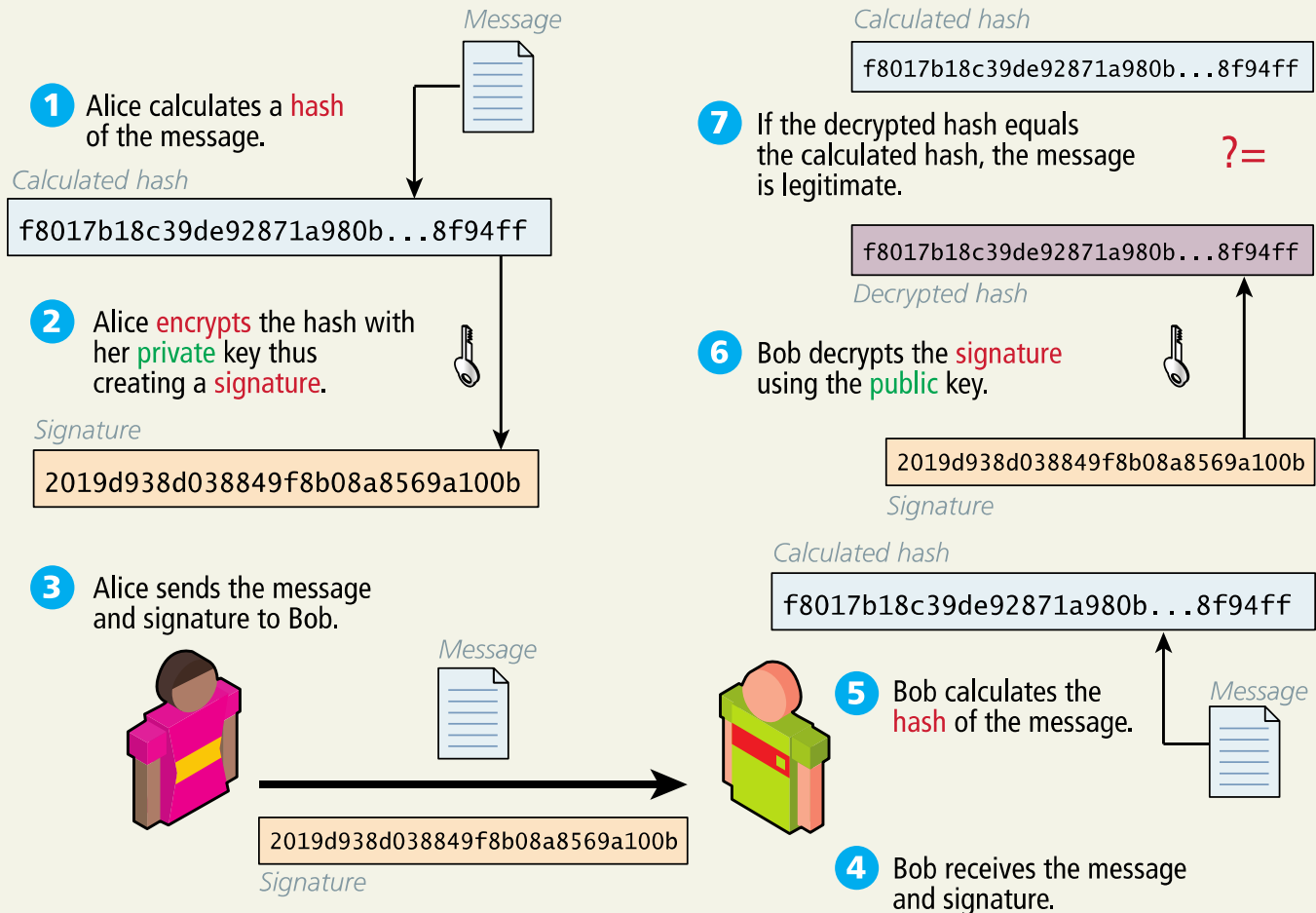
Digital Signatures

A **digital signature** is a mathematically secure way of validating that a particular digital document was

- created by the person claiming to create it (authenticity),
- was not modified in transit (integrity), and
- cannot be denied (nonrepudiation).

Cryptography

Digital Signatures



Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

5 Security Best Practices

6 Common Threat Vectors

7 Summary

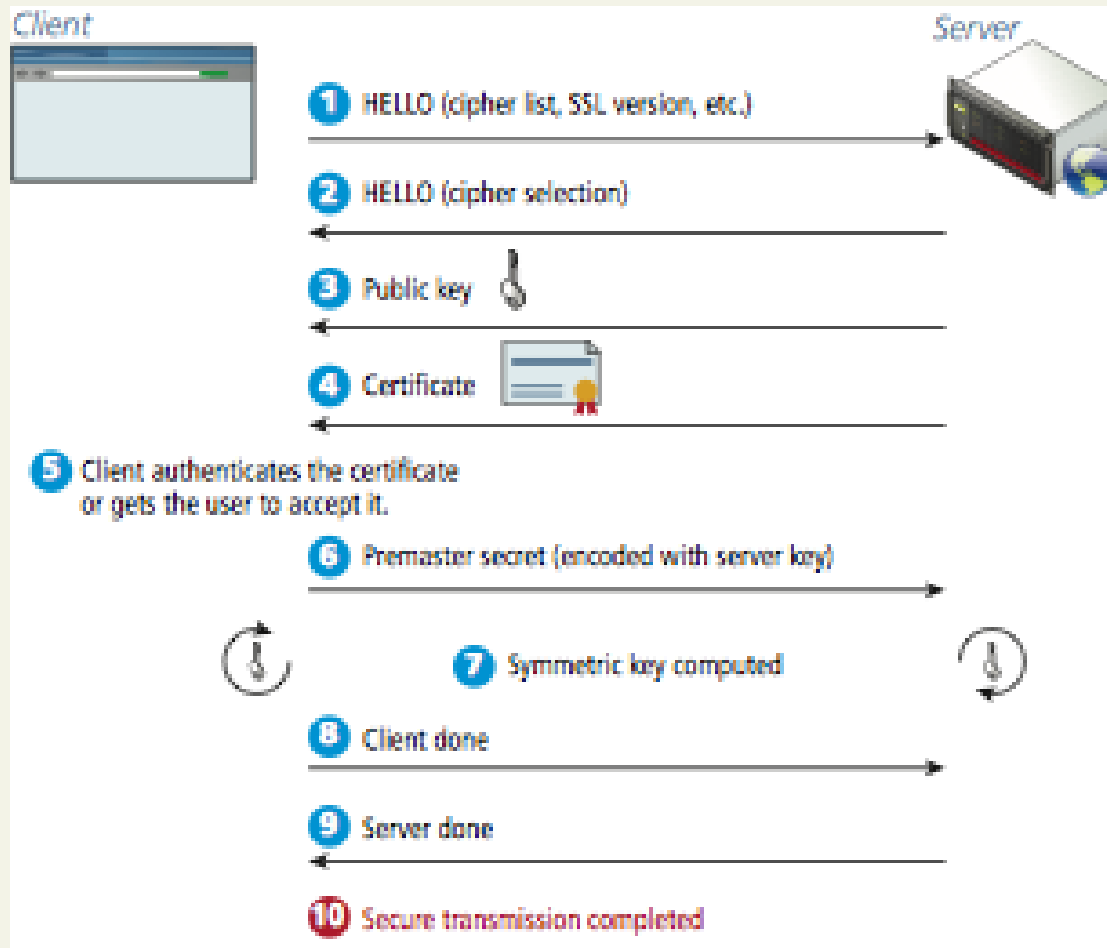
Hypertext Transfer Protocol Secure (HTTPS)

What the browsers say



Hypertext Transfer Protocol Secure (HTTPS)

Secure Handshakes



Hypertext Transfer Protocol Secure (HTTPS)

Certificates and Authorities

Certificate Authority (CA) allows users to place their trust in the certificate since a trusted, independent third party signs it.

The CA's primary role is to validate that the requestor of the certificate is who they claim to be, and issue and sign the certificate containing the public keys so that anyone seeing them can trust they are genuine.

In browsers, there are many dozens of CAs trusted by default

You can also self-sign certificates (generates warnings)

Hypertext Transfer Protocol Secure (HTTPS)

Certificates and Authorities



Hypertext Transfer Protocol Secure (HTTPS)

Certificates and Authorities



This Connection is Untrusted

You have asked Firefox to connect securely to **funwebdev.com**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

▼ Technical Details

funwebdev.com uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

(Error code: sec_error_untrusted_issuer)

▶ I Understand the Risks

Hypertext Transfer Protocol Secure (HTTPS)

Migrating to HTTPS from HTTP

Coordinating the migration of a website can be a complex endeavor

- Mixed Content
 - Internal links within the site.
 - External links to frameworks delivered through a CDN.
 - Any links or references generated by PHP code that might include a hardcoded http.
 - References to http within any HTML markup outside of PHP blocks.
- Redirects from old Site

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

5 Security Best Practices

6 Common Threat Vectors

7 Summary

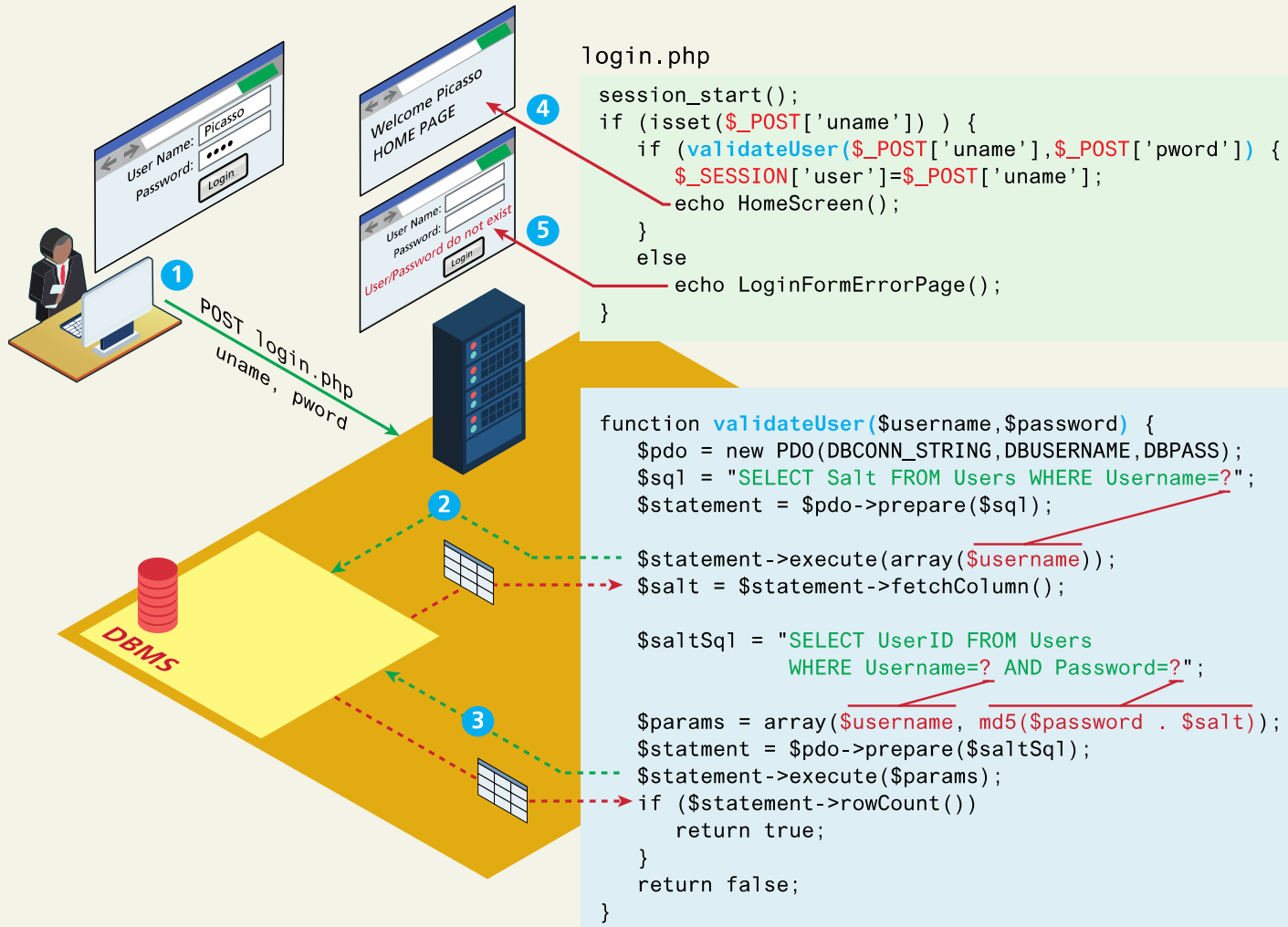
Security Best Practices

Data Storage

- Secure Hash
- Salting the Hash

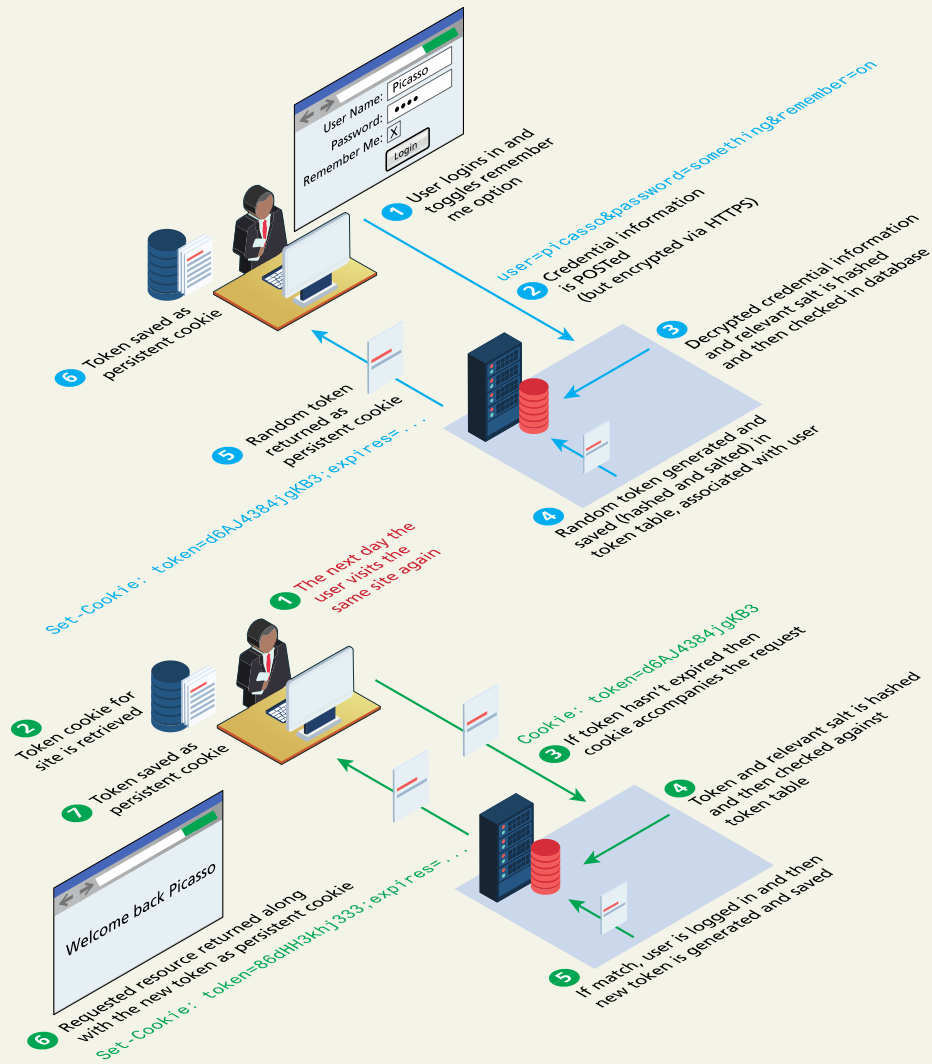
Security Best Practices

Data Storage



Security Best Practices

Remembering a user login



Security Best Practices

Monitor Your Systems

- System Monitors
- Access Monitors
- Automated Intrusion Blocking
- ...

Security Best Practices

Audit and Attack Thyself

There are a number of companies that you can hire (and grant written permission) to test your servers and report on what they've found.

If you prefer to perform your own analysis, you should be aware of some open-source attack tools such as w3af, which provide a framework to test your system including SQL injections, XSS, bad credentials, and more.

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

5 Security Best Practices

6 Common Threat Vectors

7 Summary

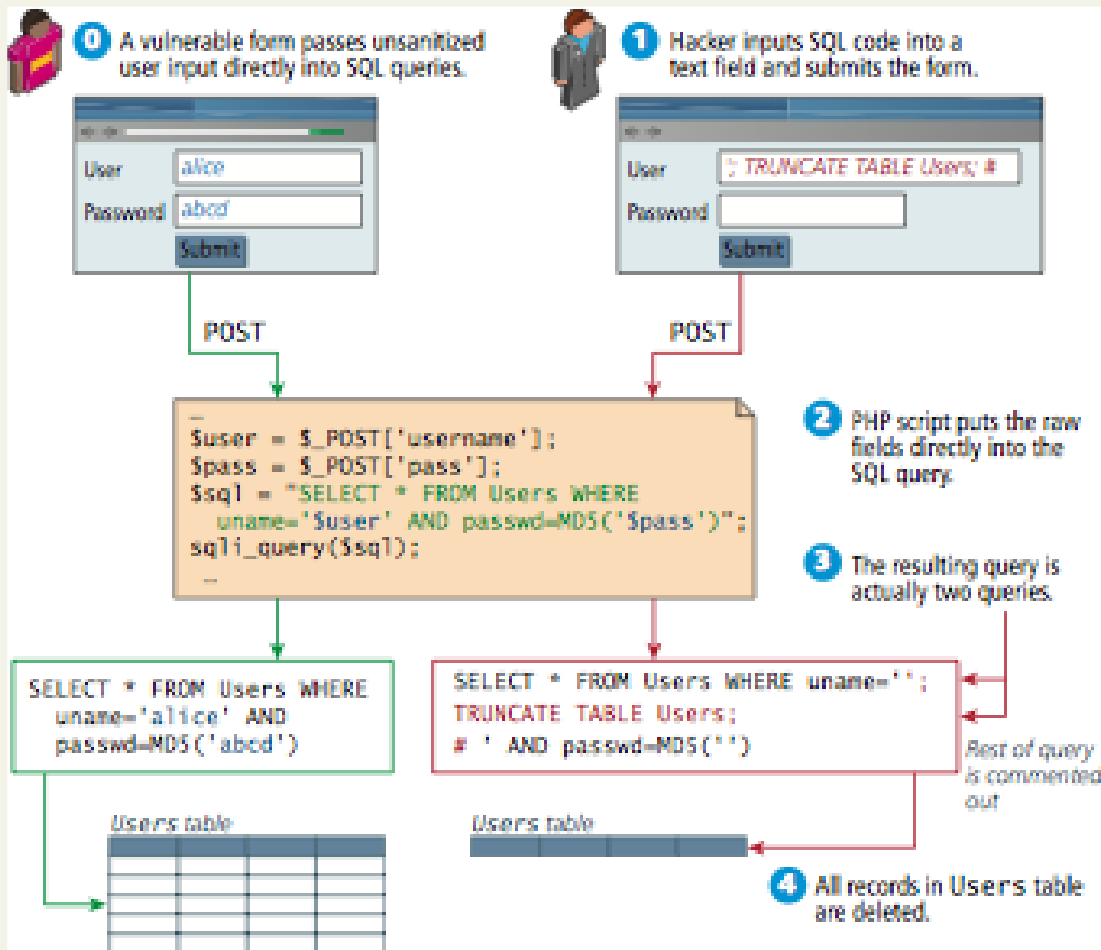
Common Threat Vectors

Brute-Force Attacks

- throttle login attempts
 - Limit number of guesses
 - CAPTCHA


Common Threat Vectors

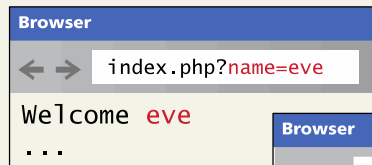
SQL Injection



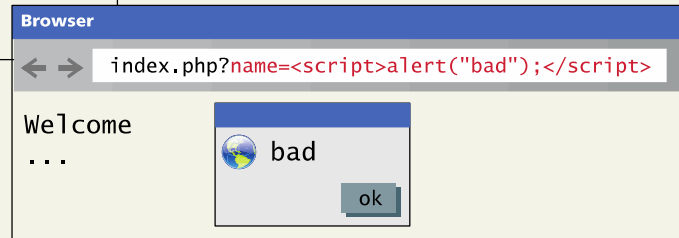
Common Threat Vectors

Cross-Site Scripting (XSS)

-  **1** A malicious user targets a site that is obviously reflecting data from the user back to them.



- 2** The malicious user tests a simple XSS to see if it works.



- 3** The malicious user crafts a more malicious URL.

index.php?name=<script>...</script>

The malicious user might shorten it with a URL shortening service.

http://bit.ly/au83n9/

- 4** The malicious user sends an email to potential users of the site that contains the malicious URL as a link.



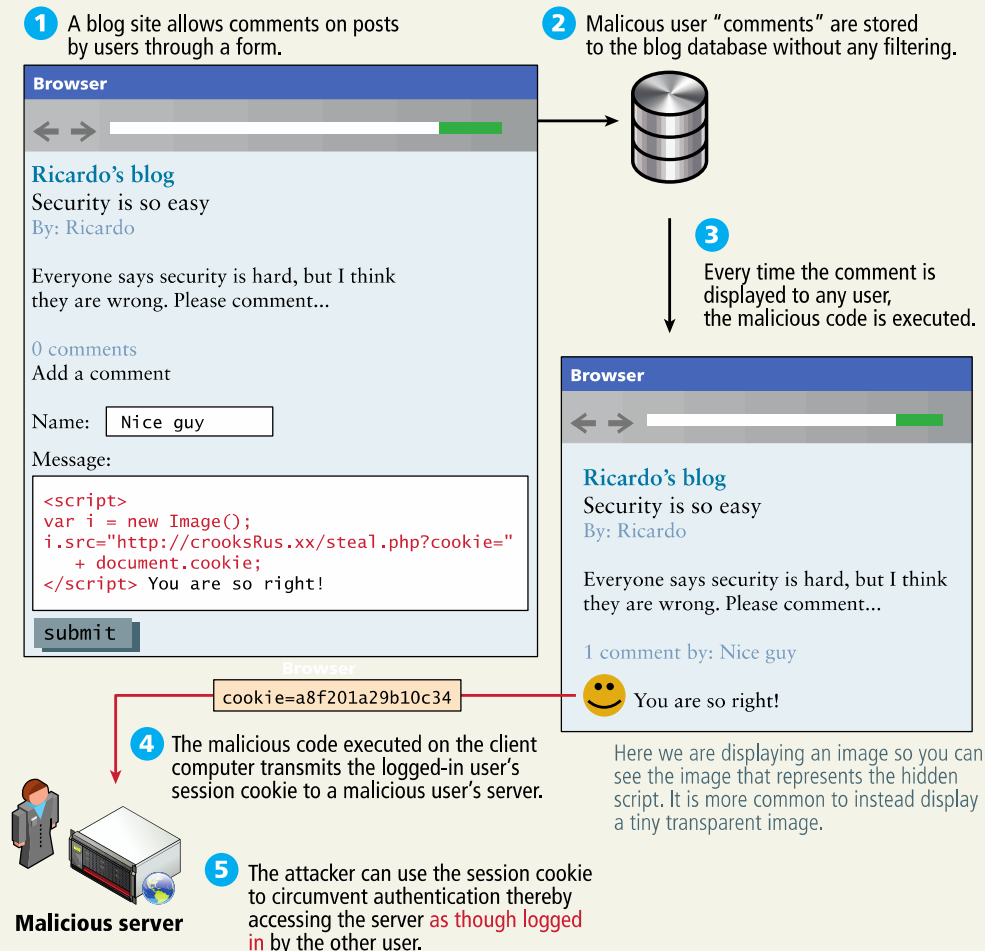
- 5** The victim clicks the link, and the site reflects the script into the user's browser.



The script executes (unbeknownst to them). The attack is successful!

Common Threat Vectors

Stored Cross-Site Scripting (XSS)



Common Threat Vectors

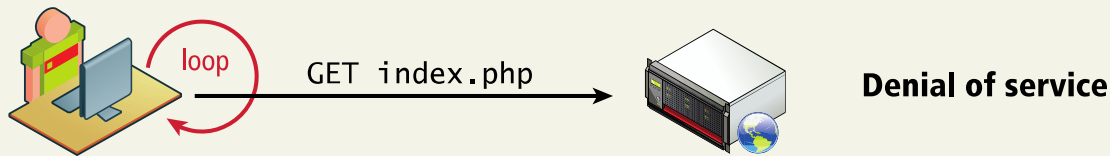
Insecure Direct Object Reference

An insecure direct object reference is a fancy name for when some internal value or key of the application is exposed to the user, and attackers can then manipulate these internal keys to gain access to things they should not have access to.

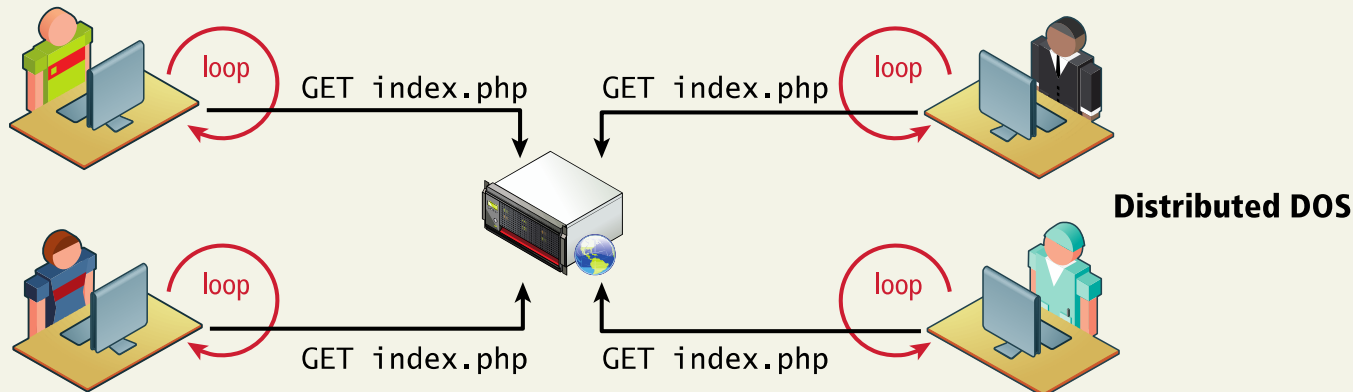
- URL hacking
- Obfuscate URLs

Common Threat Vectors

Denial of Service



This computer is running a program or script that is repeatedly requesting a page from the server.



Each computer in this **bot army** is running the same program or script that is bombarding the server with requests. These users are probably unaware that this is happening.

Common Threat Vectors

Security Misconfiguration

- Out of date software/patches
- Open Mail Relays
- More Input Attacks
- Virtual Open Mail Relay
- Arbitrary Program Execution

Common Threat Vectors

Security Misconfiguration – virtual open mail relay

0 A contact form transmits the email of the receiver within the HTML in the **to:** field.

Browser

Contact Us

From: youremail@example.com

To: Select one
rconolly@mtroyal.ca
rhoar@mtroyal.ca

Message: Type here ...

submit

Query string parameters

```
sender=some-person@where-ever.com  
receiver=rhoar@mtroyal.ca  
message=[Hello I love your book ...]
```

1 Malicious user sees that you are transmitting email addresses in HTML and creates a spam script to mail a list of addresses.



```
Aphrodite@abc.xyz  
Apollo@abc.xyz  
Ares@abc.xyz  
Artemis@abc.xyz  
Athena@abc.xyz  
...  
Zeus@abc.xyz
```

Query string parameters

```
sender=fakename@realbank.com  
receiver=Aphrodite@abc.xyz  
message=[spam (or worse)]
```

2 PHP script passes the query string input directly to the PHP mail() function.

```
...  
$from = $_POST['sender'];  
$to = $_POST['receiver'];  
$msg = $_POST['message'];  
$header = "From: " . $from . "\r\n";  
mail($to, "Form message", $msg, $header);  
...
```

3 The form thus acts as an open relay and lets the malicious user send many messages.

Mail from contact form

To: rhoar@mtroyal.ca

Spam mail from malicious user

To: Aphrodite@abc.xyz

To: Apollo@abc.xyz

To: Zeus@abc.xyz

Common Threat Vectors

Security Misconfiguration – command line pass through

0 The script is intended to echo the output of a ping command to the user for the IP or domain they want.

Browser

Ping an IP address

Enter IP:

submit

1 Malicious user inputs reserved characters and commands into the text field.

Browser

Ping an IP address

Enter IP:

submit

```
...  
$ip = $_POST['ip'];  
$ret = exec("ping -c 1 $ip 2>&1", $output);  
print_r($output);  
print_r($ret);  
...
```

2 PHP script passes the user input as a parameter to a Unix command (ping).

```
Array  
(  
[0] => PING funwebdev.com (66.147.244.79): ...  
[1] => 64 bytes from 66.147.244.79: icmp_seq=0 ...  
[2] => 64 bytes from 66.147.244.79: icmp_seq=1 ...  
[3] => 64 bytes from 66.147.244.79: icmp_seq=2 ...  
[4] => 64 bytes from 66.147.244.79: icmp_seq=3 ...  
[5] =>  
[6] => --- funwebdev.com ping statistics ---  
[7] => 4 packets transmitted, 4 packets ...  
[8] => round-trip min/avg/max/stddev = ...  
)  
round-trip min/avg/max/stddev = ...
```

Displayed to user (as intended)

3 The attacker executes arbitrary command (in this case 1s) and gains knowledge for further exploits and attacks.

```
Array  
(  
[0] => a182761.png  
[1] => b171628.png  
[2] => c998716.png  
[3] => super-secret.png  
[4] => top-secret.txt  
...  
)  
Z1928.png
```

Displayed to malicious user

Chapter 18

1 Security Principles

2 Authentication

3 Cryptography

4 Hypertext Transfer Protocol Secure (HTTPS)

5 Security Best Practices

6 Common Threat Vectors

7 Summary

Summary

Key Terms

asymmetric cryptography
auditing
authentication
authentication cookie
authentication factors
authentication policy
authorization
availability
block ciphers
Caesar cipher
Certificate Authority
cipher
CIA triad
code review
confidentiality
Content Security Policy
cross-site scripting
cryptographic hash
functions
decryption
denial of service attacks
digest
digital signature
encryption
external actors
form-based
authentication
high-availability
HTTP basic authentication
HTTP digest
authentication
Hypertext Transfer
Protocol
Secure (HTTPS)
information assurance
information security
inherence factors
input coupled control
insecure direct object
reference
integrity

Summary

Key Terms continued

internal actors	password policies	authentication
key	phishing scams	social engineering
knowledge factors	premaster secret	stored XSS
legal policies	principle of least privilege	SQL injection
logging	public key cryptography	STRIDE
man-in-the-middle attacks	rainbow table	substitution cipher
mixed content	reflected XSS	symmetric ciphers
multifactor authentication	salting	threat
OAuth	secure by default	token-based
one-time pad	secure by design	authentication
one-way hash functions	Secure Sockets Layer	unit testing
open mail relay	security testing	usage policy
ownership factors	security theater	Vigenère cipher
pair programming	self-signed certificates	vulnerabilities
partner actors	single-factor	

Summary

Questions?