Authentication Vulnerabilities

Bernard Nongpoh

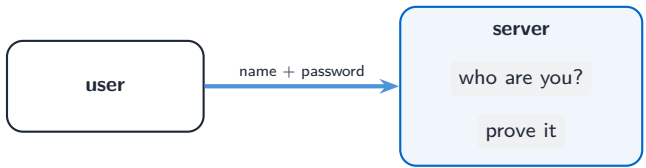
Assistant Professor, Department of CSE



ভাৰতীয় প্ৰযুক্তিবিদ্যা প্ৰতিষ্ঠান গুৱাহাটী
भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

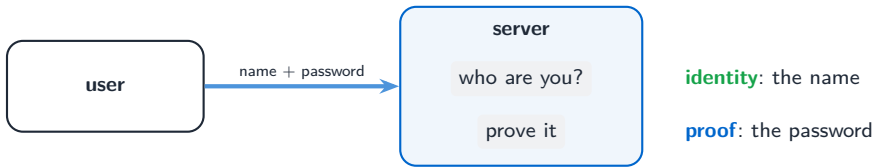
Authentication: Who Are You? (1/3)

A login asks two things. The password answers only one of them.



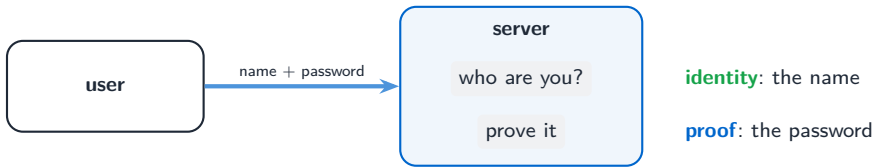
Authentication: Who Are You? (2/3)

A login asks two things. The password answers only one of them.



Authentication: Who Are You? (3/3)

A login asks two things. The password answers only one of them.



Today we open an account **three ways**, and only one of them is the password.

Four Ways Past a Password (1/2)

Where does an attacker push when the password holds?

1. Guess it



weak password,
no limit on tries

Password Strength

2. Reset it

security question
with a public answer

Reset Jim

3. Forge the token

`h.p.sig`

server trusts an
unsigned token

Unsigned JWT

4. Steal the session



reuse or steal
a live token

2:00 session

Four Ways Past a Password (2/2)

Where does an attacker push when the password holds?

1. Guess it



weak password,
no limit on tries

Password Strength

2. Reset it

security question
with a public answer

Reset Jim

3. Forge the token

`h.p.sig`

server trusts an
unsigned token

Unsigned JWT

4. Steal the session



reuse or steal
a live token

2:00 session

The first three are now, on Juice Shop. The fourth is this afternoon.

Guessing the Password (1/4)

Two failures have to meet for guessing to work.



Guessing the Password (2/4)

Two failures have to meet for guessing to work.



Guessing the Password (3/4)

Two failures have to meet for guessing to work.



Guessing the Password (4/4)

Two failures have to meet for guessing to work.



Weak password and **no limit on tries**: remove either one and guessing stops.

Password Strength in Juice Shop (1/3)



We know the admin's email. The password is on the list.

```
email:    admin@juice-sh.op
password: admin123
```

Juice Shop challenge *Password Strength* (A07)

Password Strength in Juice Shop (2/3)



We know the admin's email. The password is on the list.

```
email:    admin@juice-sh.op
password: admin123
```

weak password (on every common list) + **no lockout** (try forever)

Juice Shop challenge *Password Strength* (A07)

Password Strength in Juice Shop (3/3)



We know the admin's email. The password is on the list.

```
email:    admin@juice-sh.op
password: admin123
```

weak password (on every common list) + **no lockout** (try forever)

Lab (6 min). Log in as `admin@juice-sh.op` with `admin123`. Score board: *Password Strength*.

Juice Shop challenge *Password Strength* (A07)

The Password Reset Back Door (1/4)

If you cannot guess the password, make the site give you a new one.



OWASP Forgot Password Cheat Sheet

The Password Reset Back Door (2/4)

If you cannot guess the password, make the site give you a new one.

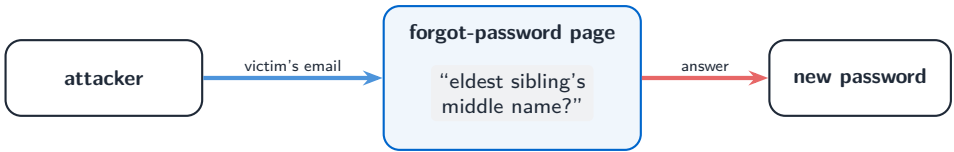


the answer is **public**:
a second password the user cannot keep secret

OWASP Forgot Password Cheat Sheet

The Password Reset Back Door (3/4)

If you cannot guess the password, make the site give you a new one.

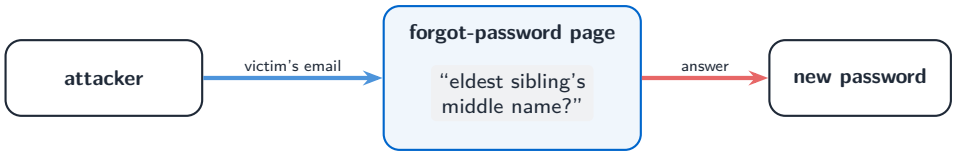


the answer is **public**:
a second password the user cannot keep secret

OWASP Forgot Password Cheat Sheet

The Password Reset Back Door (4/4)

If you cannot guess the password, make the site give you a new one.



Fix: no security questions. Reset through a **one-time link** to a verified email.

OWASP Forgot Password Cheat Sheet

Reset Jim's Password in Juice Shop (1/2)



Jim is a real account. His question has a public answer.

```
email:    jim@juice-sh.op
question: Your eldest sibling's middle name?
answer:   Samuel   (Jim = James T. Kirk)
```

Juice Shop challenge *Reset Jim's Password* (A07)

Reset Jim's Password in Juice Shop (2/2)



Jim is a real account. His question has a public answer.

```
email:    jim@juice-sh.op
question: Your eldest sibling's middle name?
answer:   Samuel   (Jim = James T. Kirk)
```

Lab (6 min). *Forgot your password?* → Jim's email → answer `Samuel` → set a new password.

Juice Shop challenge *Reset Jim's Password* (A07)

Quiz: Which Way In? (1/5)

Guess it, reset it, or forge the token? Watch for the case that is not obvious.

1. The link in a “forgot password” email works forever

2. A bank app takes a 4-digit PIN, with no limit on tries

3. The site reads your user id from a value your browser keeps, and never checks who made it

4. To recover an account you name your favourite team, which is on your public profile

OWASP A07

Quiz: Which Way In? (2/5)

Guess it, reset it, or forge the token? Watch for the case that is not obvious.

1. The link in a "forgot password" email works forever	`reset it`
2. A bank app takes a 4-digit PIN, with no limit on tries	
3. The site reads your user id from a value your browser keeps, and never checks who made it	
4. To recover an account you name your favourite team, which is on your public profile	

OWASP A07

Quiz: Which Way In? (3/5)

Guess it, reset it, or forge the token? Watch for the case that is not obvious.

1. The link in a "forgot password" email works forever	`reset it`
2. A bank app takes a 4-digit PIN, with no limit on tries	`guess it`
3. The site reads your user id from a value your browser keeps, and never checks who made it	
4. To recover an account you name your favourite team, which is on your public profile	

OWASP A07

Quiz: Which Way In? (4/5)

Guess it, reset it, or forge the token? Watch for the case that is not obvious.

1. The link in a "forgot password" email works forever — `reset it`
2. A bank app takes a 4-digit PIN, with no limit on tries — `guess it`
3. The site reads your user id from a value your browser keeps, and never checks who made it — `forge the token`
4. To recover an account you name your favourite team, which is on your public profile

OWASP A07

Quiz: Which Way In? (5/5)

Guess it, reset it, or forge the token? Watch for the case that is not obvious.

1. The link in a "forgot password" email works forever — `reset it`
2. A bank app takes a 4-digit PIN, with no limit on tries — `guess it`
3. The site reads your user id from a value your browser keeps, and never checks who made it — `forge the token`
4. To recover an account you name your favourite team, which is on your public profile — `reset it`

OWASP A07

The Token After Login: a JWT (1/5)

After you log in, the server hands you a token. What is in it?



RFC 7519: JSON Web Token

The Token After Login: a JWT (2/5)

After you log in, the server hands you a token. What is in it?



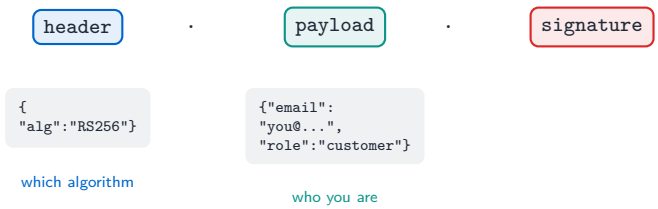

```
{
"alg":"RS256"}
```

which algorithm

RFC 7519: JSON Web Token

The Token After Login: a JWT (3/5)

After you log in, the server hands you a token. What is in it?



RFC 7519: JSON Web Token

The Token After Login: a JWT (4/5)

After you log in, the server hands you a token. What is in it?



RFC 7519: JSON Web Token

The Token After Login: a JWT (5/5)

After you log in, the server hands you a token. What is in it?



Reading a token needs no key; only **making** a valid one needs the private key.

RFC 7519: JSON Web Token

Unsigned JWT: alg none (1/4)



What if the server accepts a token that says “no signature”?

the forged token

```
header: {"alg":"none"}
payload: {"email":
  "jwtn3d@juice-sh.op"}
signature: (empty)
```

Juice Shop challenge *Unsigned JWT* (A08); RFC 7519

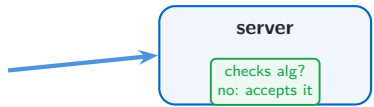
Unsigned JWT: alg none (2/4)



What if the server accepts a token that says “no signature”?

the forged token

```
header: {"alg":"none"}
payload: {"email":
  "jwtn3d@juice-sh.op"}
signature: (empty)
```



Juice Shop challenge *Unsigned JWT* (A08); RFC 7519

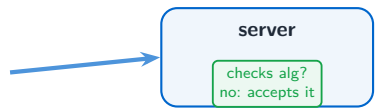
Unsigned JWT: alg none (3/4)



What if the server accepts a token that says “no signature”?

the forged token

```
header: {"alg":"none"}
payload: {"email":
  "jwtn3d@juice-sh.op"}
signature: (empty)
```



logged in as a user
we invented

Juice Shop challenge *Unsigned JWT* (A08); RFC 7519

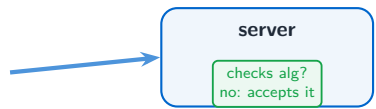
Unsigned JWT: alg none (4/4)



What if the server accepts a token that says “no signature”?

the forged token

```
header: {"alg":"none"}
payload: {"email":
  "jwtn3d@juice-sh.op"}
signature: (empty)
```



logged in as a user
we invented

Fix, one line: accept **only the exact algorithm** you issue; reject `none`.

Juice Shop challenge *Unsigned JWT* (A08); RFC 7519

Defenses (1/4)

Weakness	Fix
weak password, no lockout	strong-password rules **and** lock or delay after a few tries

OWASP Authentication Cheat Sheet

Defenses (2/4)

Weakness	Fix
weak password, no lockout	strong-password rules **and** lock or delay after a few tries
security question	drop it: reset by one-time link to a verified email

OWASP Authentication Cheat Sheet

Defenses (3/4)

Weakness	Fix
weak password, no lockout	strong-password rules **and** lock or delay after a few tries
security question	drop it: reset by one-time link to a verified email
unsigned token (alg none)	accept only the exact algorithm; reject `none`

OWASP Authentication Cheat Sheet

Defenses (4/4)

Weakness	Fix
weak password, no lockout	strong-password rules **and** lock or delay after a few tries
security question	drop it: reset by one-time link to a verified email
unsigned token (alg none)	accept only the exact algorithm; reject `none`
all of the above	add a second factor (MFA): a code or an app

The theme of the day again: never trust what the client sends. Check it on the server.

OWASP Authentication Cheat Sheet

Summary (1/3)

Way in	**What we did**	**Juice Shop**
guess it	admin with a common password	Password Strength

Summary (2/3)

Way in	**What we did**	**Juice Shop**
guess it	admin with a common password	Password Strength
reset it	answered a public security question	Reset Jim's Password

Summary (3/3)

Way in	**What we did**	**Juice Shop**
guess it	admin with a common password	Password Strength
reset it	answered a public security question	Reset Jim's Password
forge the token	a JWT with no signature	Unsigned JWT

OWASP A07