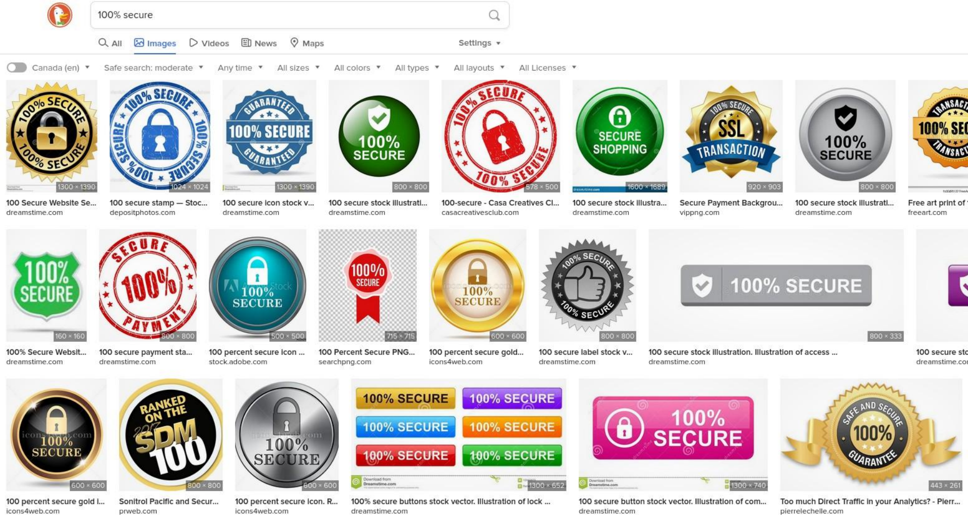
Lecture 01
Introduction

Software Security Engineering

Winter 2023
Thompson Rivers University

General Themes

- Computers do precisely what they’re told
- Code is data and data is code
- Features and convenience creates vulnerabilities
    - this includes features of programming languages
- no such thing as 100% secure, goal is risk management



100% secure

Canada (en) ▾ Safe search: moderate ▾ Any time ▾ All sizes ▾ All colors ▾ All types ▾ All layouts ▾ All Licenses ▾

1300 × 1390 100 Secure Website Se... dreamstime.com

1024 × 1024 100 secure stamp — Stoc... depositphotos.com

1300 × 1390 100 secure icon stock v... dreamstime.com

800 × 800 100 secure stock illustrati... dreamstime.com

578 × 500 100-secure - Casa Creatives Cl... casacreativesclub.com

1600 × 1689 100 secure stock illustra... dreamstime.com

920 × 903 Secure Payment Backgrou... vippng.com

800 × 800 100 secure stock illustrati... dreamstime.com

Free art print of freeart.com

160 × 160 100% Secure Websit... dreamstime.com

800 × 800 100 secure payment sta... dreamstime.com

500 × 500 100 percent secure icon ... stock.adobe.com

715 × 715 100 Percent Secure PNG... searchpng.com

600 × 600 100 percent secure gold... icons4web.com

800 × 800 100 secure label stock v... dreamstime.com

800 × 333 100 secure stock illustration. Illustration of access ... dreamstime.com

100 secure sto dreamstime.cor

600 × 600 100 percent secure gold l... icons4web.com

800 × 800 Sonitrol Pacific and Secur... prweb.com

600 × 600 100 percent secure icon. R... icons4web.com

1300 × 652 100% secure buttons stock vector. Illustration of lock ... dreamstime.com

1300 × 740 100 secure button stock vector. Illustration of com... dreamstime.com

443 × 261 Too much Direct Traffic in your Analytics? - Pierr... pierrelechelle.com

Computer Security

- keep systems functioning as intended
  - free of abuse
- keep data accessed only as desired
- secure access to resources and capabilities
- enable privacy and anonymity
- do all of this
  - with an adversary
  - on a budget

Computer security - Definition

“We define computer security as the combined art, science and engineering practice of protecting computer-related assets from unauthorized actions and their consequences, either by preventing such actions or detecting and then recovering from them.” [1]

Goals of Computer Security

- Confidentiality
  - non-public information accessible only to authorized parties
  - stored (at rest) or in transmission (in motion)
  - technical means: encryption
  - procedure means:
    - offline storage in secured sites
      e.g., guards, guns
- Integrity
  - data, software, and hardware remains unaltered
  - checksums detect this
  - preventing changes is harder
    - includes integrity of people
      e.g., bribery, corruption

Goals of Computer Security

- authorization
  - resources accessed only by authorized entities approved by resource owner
  - achieved by **access control mechanisms**

    e.g., passwords, keycards
- availability
  - information, services, and resources **can** be used
  - protect against intentional deletion or denial of service (DoS)

CIA: confidentiality, integrity, availability

Security Policies and Attacks

- security protects **assets**
  - information, software, hardware, computing and communication services
- a **security policy** specifies system’s rules and practices
  - what is and is not allowed
- a **security mechanism** implements a security policy
  - ideally the mechanism enforces the rules outlined in policy
  - mechanism can include protocols humans should follow
    - e.g., locking valuables in a safe

Example: Phone Security

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Mechanism: keep phone on person at all times

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Mechanism: keep phone on person at all times

Mechanism: keep phone on person or in a locked compartment at all times

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Mechanism: keep phone on person at all times

Mechanism: keep phone on person or in a locked compartment at all times

These have assumptions

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Mechanism: keep phone on person at all times

Mechanism: keep phone on person or in a locked compartment at all times

These have assumptions

e.g., locked compartment can only be physically accessed by the same owner.

Example: Phone Security

Policy: “work phone must never be physically handled except by owner.”

Mechanism: keep phone on person at all times

Mechanism: keep phone on person or in a locked compartment at all times

These have assumptions

e.g., locked compartment can only be physically accessed by the same owner.

e.g., the integrity of the person’s pockets cannot be compromised

Example: My bicycle

Example: My bicycle
Policy: “only I may use my bike”

Example: My bicycle
Policy: “only I may use my bike”
Mechanism: I use a bike lock or store it in a locked space when I don’t use it

Example: My bicycle
Policy: “only I may use my bike”
Mechanism: I use a bike lock or store it in a locked space when I don’t use it
Assumption: no one can use my bike while I’m using it or when its locked

Example: My produce shopping

Example: My produce shopping

Policy: “no minors allowed in Cannabis store”

Example: My produce shopping

Policy: “no minors allowed in Cannabis store”

Mechanism: inspection of government-issued ID

Example: My produce shopping

Policy: “no minors allowed in Cannabis store”

Mechanism: inspection of government-issued ID

Assumption: IDs unforgeable, or forged IDs are easy to detect

Example: Bank Security

Example: Bank Security

Policy: bank only gives information about account to account owner

Example: Bank Security

Policy: bank only gives information about account to account owner

Mechanism: they ask for your birthday when you call

Example: Bank Security

Policy: bank only gives information about account to account owner

Mechanism: they ask for your birthday when you call

Assumption: only person who knows your birthday is you

Example: Bank Security

Policy: bank only gives information about account to account owner

Mechanism: they ask for your birthday when you call

Assumption: only person who knows your birthday is you (same with mother’s maidan name, or your grade two teacher’s name)

Every security mechanism **implies** a policy objective

Every security mechanism **implies** a policy objective

I want you to think in the reverse way

Every security mechanism **implies** a policy objective

I want you to think in the reverse way

you see a security mechanism and you infer a policy

Every security mechanism **implies** a policy objective

I want you to think in the reverse way

you see a security mechanism and you infer a policy

and then you figure out an attack

Attacks often result from the mechanism’s assumptions.

Attacks often result from the mechanism’s assumptions. And you notice it when you start seeing everything in terms of security mechanisms attempting to fulfill security policies.

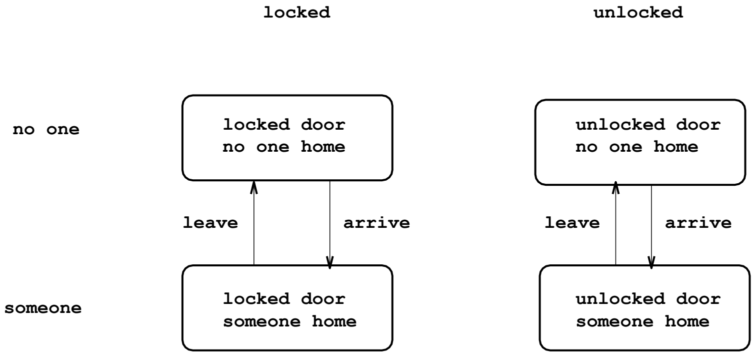
Theoretical Security

- system has **states**
- policy defines which states are **authorized (secure)** and **unauthorized (insecure)**
- e.g., “lock the door when nobody’s home” policy
  - four states for two binary variables
- policy is **violated** if the system moves into an unauthorized state
  - e.g., someone other than you gets your bank info
- the goal of a mechanism is to prevent the system from being able to go from a secure state to an insecure state

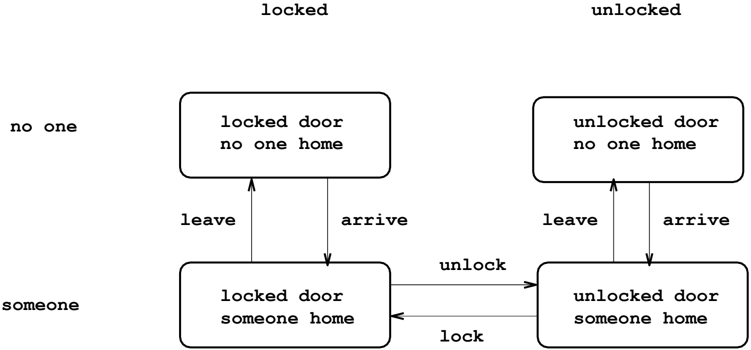
	locked	unlocked
no one		
someone		

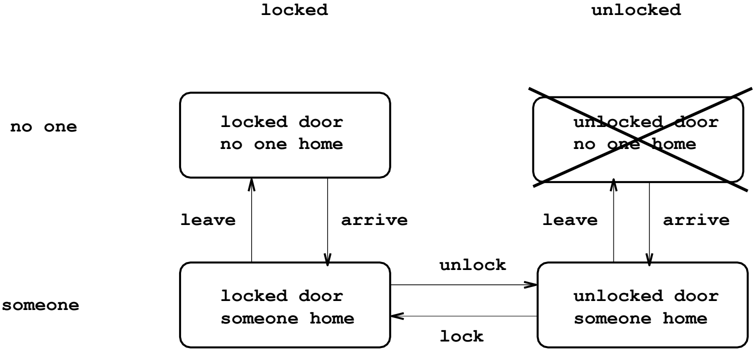
	locked	unlocked
no one	locked door no one home	unlocked door no one home
someone	locked door someone home	unlocked door someone home

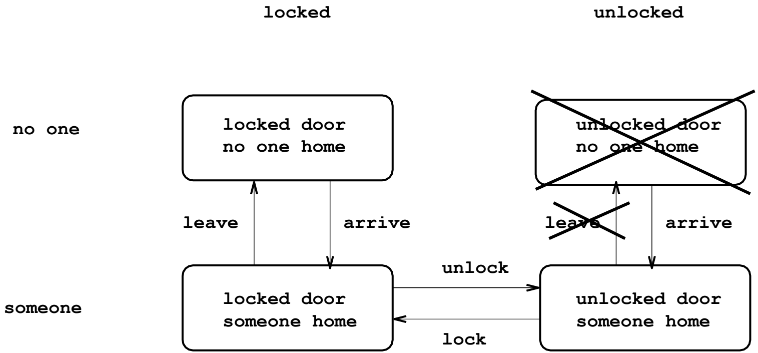
	locked	unlocked
no one	locked door no one home	unlocked door no one home
someone	locked door someone home	unlocked door someone home



locked
unlocked
no one
someone
locked door
no one home
leave
arrive
locked door
someone home
unlocked door
no one home
leave
arrive
unlocked door
someone home


locked
unlocked
no one
someone
locked door no one home
unlocked door no one home
leave
arrive
leave
arrive
unlock
locked door someone home
unlocked door someone home
lock


locked
unlocked
no one
someone
locked door
no one home
unlocked door
no one home
leave
arrive
leave
arrive
unlock
lock
locked door
someone home
unlocked door
someone home


locked
unlocked
no one
someone
locked door no one home
~~unlocked door no one home~~
leave
arrive
~~leave~~
arrive
unlock
lock
locked door someone home
unlocked door someone home

Security Attacks

- deliberate action
  - if successful causes a **security violation**
- **attack vector** is sequence of steps to do this
- attacks exploit **vulnerabilities**
  - misconfigurations
  - unsafe defaults
  - design flaws
  - implementation flaws
- source of attack (threat agent) is called **adversary** (theory) or **attacker** (systems)

Security Threat

- **threat** is any combination of circumstances and entities that may harm assets through a security violation
- the mere existence of a threat agent and a vulnerability do not imply an attack
    - indifference, insufficient incentive, insufficient resources
- attacker has a goal and a budget
    - goal: harness a resource, extract data, denying service, tampering with data, causing mischief
    - budget: time, money, abilities

Example: House Security Policy

Example: House Security Policy

No one permitted inside unless accompanied by a resident.

Example: House Security Policy

No one permitted inside unless accompanied by a resident.

Only residents may remove objects from the house.

Example: House Security Policy

No one permitted inside unless accompanied by a resident.

Only residents may remove objects from the house.

What is a security violation, vulnerability, attacker, attack vector, and threat?

Example: Implementation Flaw

Example: Implementation Flaw

Policy: gate may only be opened by someone inside the courtyard.

Example: Implementation Flaw

Policy: gate may only be opened by someone inside the courtyard.

Mechanism is a lever on the courtyard side of door.

Example: Implementation Flaw

Policy: gate may only be opened by someone inside the courtyard.

Mechanism is a lever on the courtyard side of door.

Assumption is that lever can only be turned by someone in courtyard.

No perfect security

- security violations have costs
- security **countermeasures** or **protections** have costs
- **risk assessment** analyzes these factors to estimate risk
  - **quantitative risk assessment** computes numerical estimates of risk
  - **qualitative risk assessment** ranks or orders risks
    - very low to very high for probability and cost
    - e.g., establish priorities for vulnerabilities
- $R = T \cdot V \cdot C$
  - risk is threat times existence of vulnerability times cost

Example: risk due to lava flows

Example: risk due to lava flows
houses are vulnerable to lava flows

Example: risk due to lava flows

houses are vulnerable to lava flows

cost of lava flows to an asset like house is large

Example: risk due to lava flows

houses are vulnerable to lava flows

cost of lava flows to an asset like house is large

risk vanishes if no volcanoes nearby

Example: risk due to lava flows

houses are vulnerable to lava flows

cost of lava flows to an asset like house is large

risk vanishes if no volcanoes nearby

R=0 if T=0 even when C is huge

Why is Security Challenging?

- intelligent adaptive adversary
    - can induce **zero probability** or **low probability** faults
    - can do arbitrary behaviour
    - e.g., give values as input that would never normally be given
- computer systems are built on abstractions
    - we forget these details when building systems
    - attackers use these details

Why is Security Challenging?

- an evolving field
  - adversary evolves with defenses
  - **arms race**
- computers also evolve faster than security
  - features, patches, complexity
  - vulnerabilities outscale lines of code
  - backwards compatibility

Why is Security Challenging?

- asymmetries
  - defender must defend **all fronts**
  - attacker needs only one weakness
  - defenses are public, attacks are private
    - e.g., you see my locks, guards, and cameras
    - e.g., I don’t see your plans and schemes
  - attackers are **nimble**, defenders have **sunk costs**
  - attackers have no rules, defenders have protocols
  - attackers can do nothing, defenders offer services
  - attackers are criminals, defenders follow laws

Why is Security Challenging?

- minimal deterrence
  - Internet hugely facilitates anonymity
  - attacks of great scale at little cost
  - attackers from anywhere on the planet

Why is Security Challenging?

- security has costs
  - overhead, burden, time to deploy
- security is hard to measure
  - was the investment worth it?
  - what is the value of a **lack of disaster**?
  - breach seen later, distance from attack and problems that allowed it
- market economics
  - those in position to allocate resources to security don’t benefit the most
  - security is a tax that we all pay everywhere
    - e.g., store security raises costs

Why is Security Challenging?

- bad design
  - users bypass or undermine security that is inconvenient and without obvious benefit
  - security mechanisms that are hard to use properly
    - “one click is one click too many”
  - no formal training required
- security gets in the way
  - dancing pigs problem
  - getting in the way is a cost
- social engineering works
- government obstacles
  - desire to monitor communications results in hindering sound policies like strong encryption by default
  - e.g. NSA Scandal (the Guardian - 2013)

There is no checklist to follow for security but there are **Design Principles**

Secure Design Principles

The Protection of Information in Computer Systems

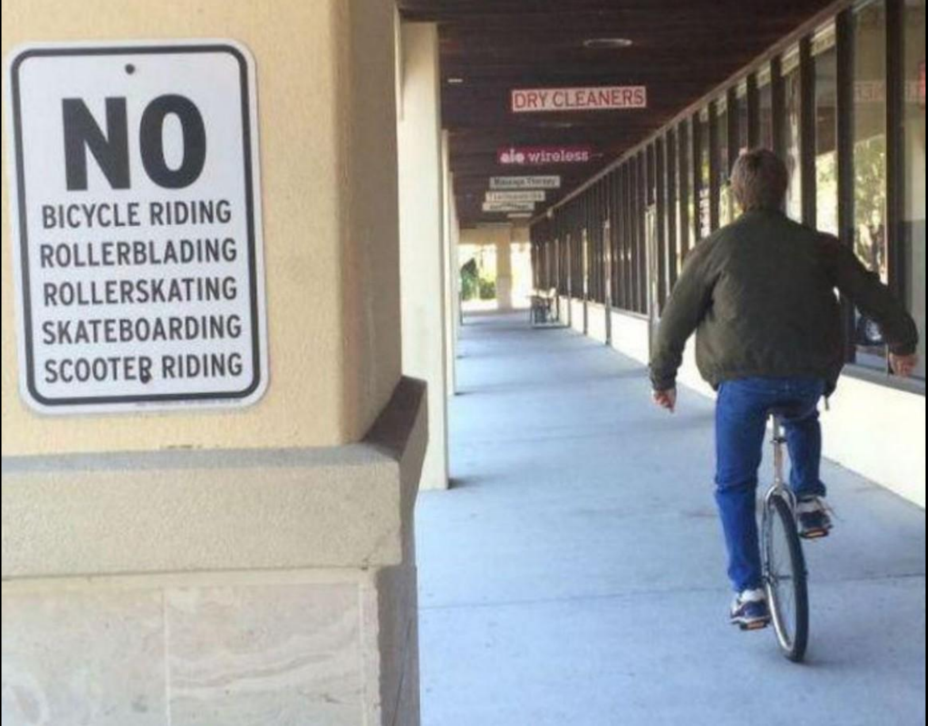
JEROME H. SALTZER, SENIOR MEMBER, IEEE, AND
MICHAEL D. SCHROEDER, MEMBER, IEEE

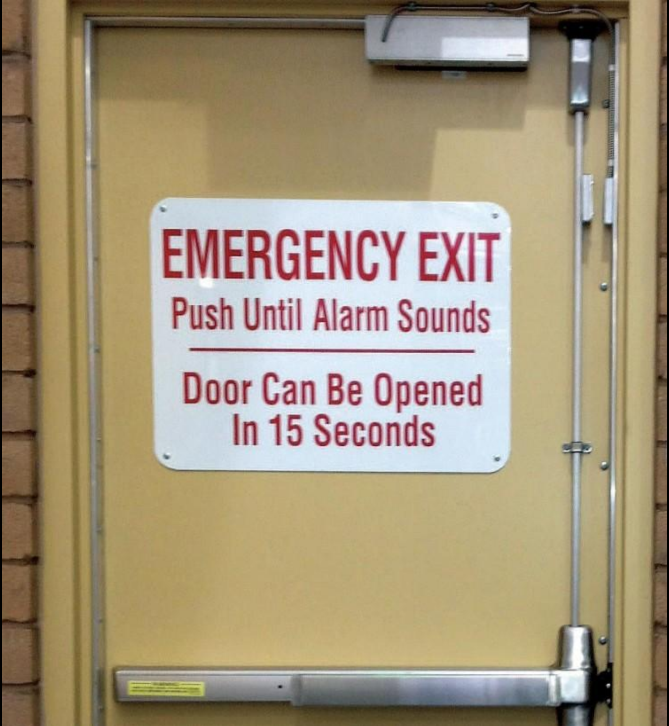
DP1: Economy of mechanism

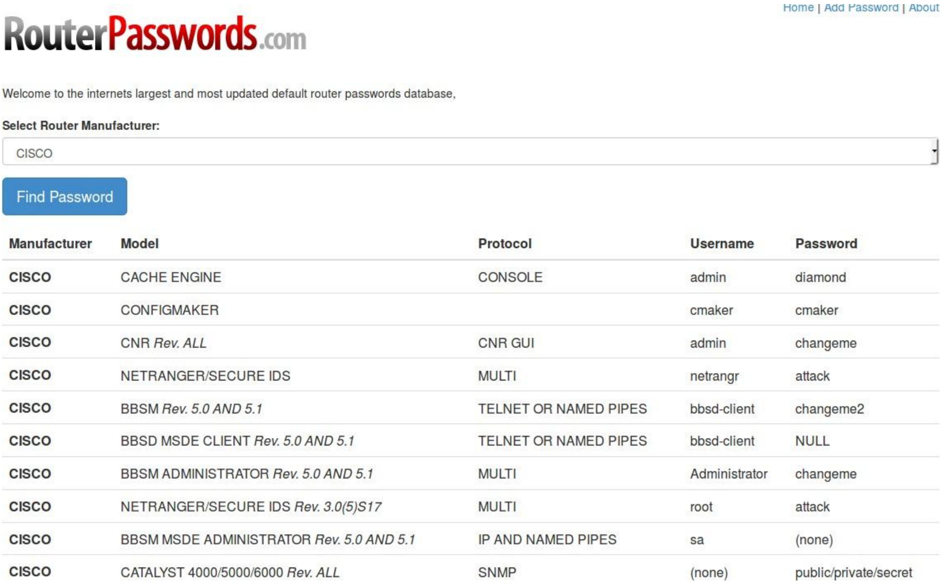
- keep designs small and simple
  - easier to analyze, test, and validate
- minimize functionality
  - disable unused functionality
  - disable by default
- this minimizes the **attack surface**
- well-used code tends to be less fragile
  - more code paths means less exercise per path

DP2: Fail-safe defaults

- use safe defaults settings
  - they aren’t usually changed
  - e.g., firewall block all ports by default
  - e.g., encrypt by default, use HTTPS by default
  - e.g., traffic lights blink red on failure
  - e.g., doors unlock during fire alarm
- favour explicit permission (allow-lists) over explicit exclusion (deny-lists)
  - base access on **permission** rather than **exclusion**
  - you may not think of all things to exclude
    - legitimate users denied access will complain
    - illegitimate users granted access won’t


NO
BICYCLE RIDING
ROLLERBLADING
ROLLERSKATING
SKATEBOARDING
SCOOTER RIDING
DRY CLEANERS
wireless


EMERGENCY EXIT
Push Until Alarm Sounds
Door Can Be Opened
In 15 Seconds



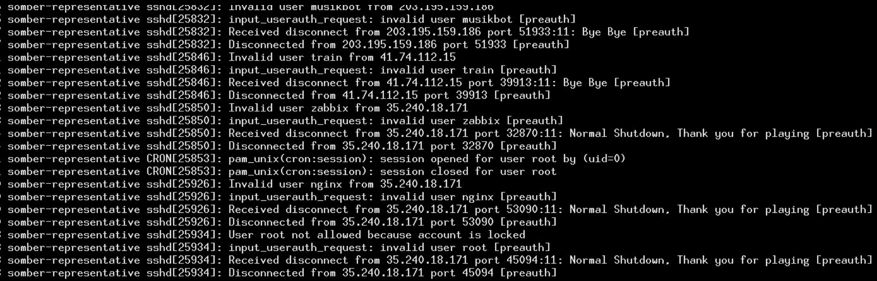
RouterPasswords.com

Welcome to the internets largest and most updated default router passwords database,

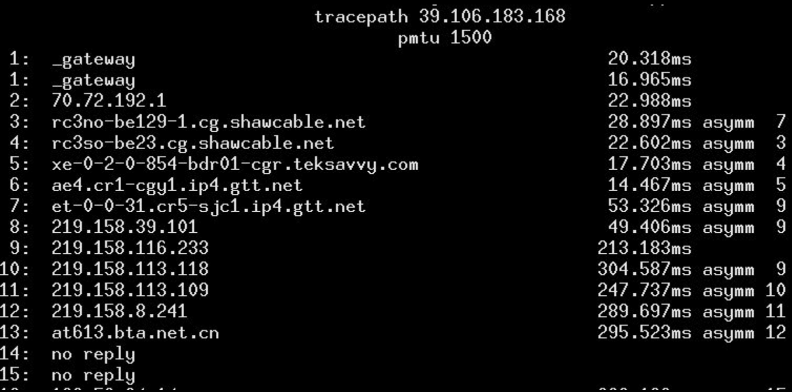
Select Router Manufacturer:

CISCO

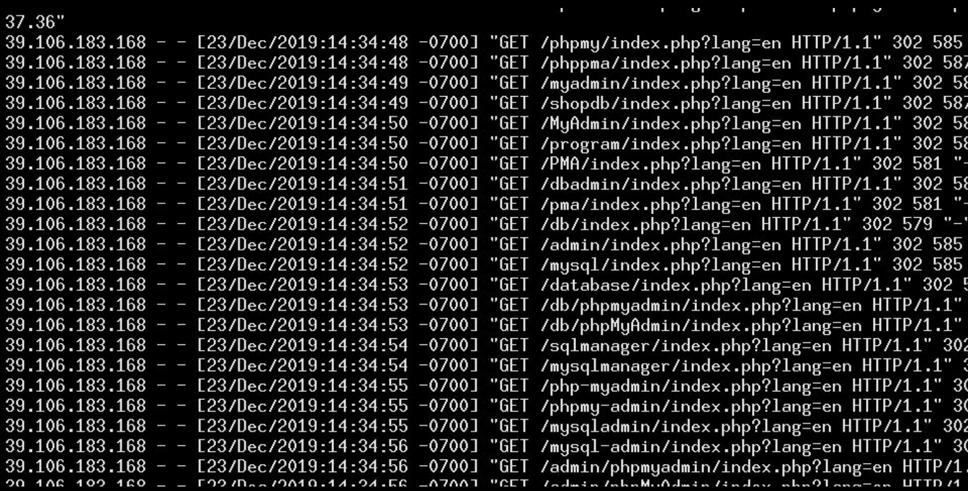
Manufacturer	Model	Protocol	Username	Password
CISCO	CACHE ENGINE	CONSOLE	admin	diamond
CISCO	CONFIGMAKER		cmaker	cmaker
CISCO	CNR *Rev. ALL*	CNR GUI	admin	changeme
CISCO	NETRANGER/SECURE IDS	MULTI	netrangr	attack
CISCO	BBSM *Rev. 5.0 AND 5.1*	TELNET OR NAMED PIPES	bbsd-client	changeme2
CISCO	BBSD MSDE CLIENT *Rev. 5.0 AND 5.1*	TELNET OR NAMED PIPES	bbsd-client	NULL
CISCO	BBSM ADMINISTRATOR *Rev. 5.0 AND 5.1*	MULTI	Administrator	changeme
CISCO	NETRANGER/SECURE IDS *Rev. 3.0(5)S17*	MULTI	root	attack
CISCO	BBSM MSDE ADMINISTRATOR *Rev. 5.0 AND 5.1*	IP AND NAMED PIPES	sa	(none)
CISCO	CATALYST 4000/5000/6000 *Rev. ALL*	SNMP	(none)	public/private/secret



```
somber-representative sshd[25832]: Invalid user musikbot from 203.195.159.186
somber-representative sshd[25832]: input_userauth_request: invalid user musikbot [preauth]
somber-representative sshd[25832]: Received disconnect from 203.195.159.186 port 51933:11: Bye Bye [preauth]
somber-representative sshd[25832]: Disconnected from 203.195.159.186 port 51933 [preauth]
somber-representative sshd[25846]: Invalid user train from 41.74.112.15
somber-representative sshd[25846]: input_userauth_request: invalid user train [preauth]
somber-representative sshd[25846]: Received disconnect from 41.74.112.15 port 39913:11: Bye Bye [preauth]
somber-representative sshd[25846]: Disconnected from 41.74.112.15 port 39913 [preauth]
somber-representative sshd[25850]: Invalid user zabbix from 35.240.18.171
somber-representative sshd[25850]: input_userauth_request: invalid user zabbix [preauth]
somber-representative sshd[25850]: Received disconnect from 35.240.18.171 port 32870:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25850]: Disconnected from 35.240.18.171 port 32870 [preauth]
somber-representative CRON[25853]: pam_unix(cron:session): session opened for user root by (uid=0)
somber-representative CRON[25853]: pam_unix(cron:session): session closed for user root
somber-representative sshd[25926]: Invalid user nginx from 35.240.18.171
somber-representative sshd[25926]: input_userauth_request: invalid user nginx [preauth]
somber-representative sshd[25926]: Received disconnect from 35.240.18.171 port 53090:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25926]: Disconnected from 35.240.18.171 port 53090 [preauth]
somber-representative sshd[25934]: User root not allowed because account is locked
somber-representative sshd[25934]: input_userauth_request: invalid user root [preauth]
somber-representative sshd[25934]: Received disconnect from 35.240.18.171 port 45094:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25934]: Disconnected from 35.240.18.171 port 45094 [preauth]
```



```
                              tracepath 39.106.183.168
                                      pmtu 1500
 1:  _gateway                                              20.318ms
 1:  _gateway                                              16.965ms
 2:  70.72.192.1                                           22.988ms
 3:  rc3no-be129-1.cg.shawcable.net                        28.897ms asymm  7
 4:  rc3so-be23.cg.shawcable.net                           22.602ms asymm  3
 5:  xe-0-2-0-854-bdr01-cgr.teksavvy.com                   17.703ms asymm  4
 6:  ae4.cr1-cgy1.ip4.gtt.net                              14.467ms asymm  5
 7:  et-0-0-31.cr5-sjc1.ip4.gtt.net                        53.326ms asymm  9
 8:  219.158.39.101                                        49.406ms asymm  9
 9:  219.158.116.233                                      213.183ms
10:  219.158.113.118                                      304.587ms asymm  9
11:  219.158.113.109                                      247.737ms asymm 10
12:  219.158.8.241                                        289.697ms asymm 11
13:  at613.bta.net.cn                                     295.523ms asymm 12
14:  no reply
15:  no reply
```



```
37.36"
39.106.183.168 - - [23/Dec/2019:14:34:48 -0700] "GET /phpmy/index.php?lang=en HTTP/1.1" 302 585
39.106.183.168 - - [23/Dec/2019:14:34:48 -0700] "GET /phpppma/index.php?lang=en HTTP/1.1" 302 587
39.106.183.168 - - [23/Dec/2019:14:34:49 -0700] "GET /myadmin/index.php?lang=en HTTP/1.1" 302 58
39.106.183.168 - - [23/Dec/2019:14:34:49 -0700] "GET /shopdb/index.php?lang=en HTTP/1.1" 302 587
39.106.183.168 - - [23/Dec/2019:14:34:50 -0700] "GET /MyAdmin/index.php?lang=en HTTP/1.1" 302 58
39.106.183.168 - - [23/Dec/2019:14:34:50 -0700] "GET /program/index.php?lang=en HTTP/1.1" 302 58
39.106.183.168 - - [23/Dec/2019:14:34:50 -0700] "GET /PMA/index.php?lang=en HTTP/1.1" 302 581 "-
39.106.183.168 - - [23/Dec/2019:14:34:51 -0700] "GET /dbadmin/index.php?lang=en HTTP/1.1" 302 58
39.106.183.168 - - [23/Dec/2019:14:34:51 -0700] "GET /pma/index.php?lang=en HTTP/1.1" 302 581 "-
39.106.183.168 - - [23/Dec/2019:14:34:52 -0700] "GET /db/index.php?lang=en HTTP/1.1" 302 579 "-"
39.106.183.168 - - [23/Dec/2019:14:34:52 -0700] "GET /admin/index.php?lang=en HTTP/1.1" 302 585
39.106.183.168 - - [23/Dec/2019:14:34:52 -0700] "GET /mysql/index.php?lang=en HTTP/1.1" 302 585
39.106.183.168 - - [23/Dec/2019:14:34:53 -0700] "GET /database/index.php?lang=en HTTP/1.1" 302 5
39.106.183.168 - - [23/Dec/2019:14:34:53 -0700] "GET /db/phpmyadmin/index.php?lang=en HTTP/1.1"
39.106.183.168 - - [23/Dec/2019:14:34:53 -0700] "GET /db/phpMyAdmin/index.php?lang=en HTTP/1.1"
39.106.183.168 - - [23/Dec/2019:14:34:54 -0700] "GET /sqlmanager/index.php?lang=en HTTP/1.1" 302
39.106.183.168 - - [23/Dec/2019:14:34:54 -0700] "GET /mysqlmanager/index.php?lang=en HTTP/1.1" 3
39.106.183.168 - - [23/Dec/2019:14:34:55 -0700] "GET /php-myadmin/index.php?lang=en HTTP/1.1" 30
39.106.183.168 - - [23/Dec/2019:14:34:55 -0700] "GET /phpmy-admin/index.php?lang=en HTTP/1.1" 30
39.106.183.168 - - [23/Dec/2019:14:34:55 -0700] "GET /mysqladmin/index.php?lang=en HTTP/1.1" 302
39.106.183.168 - - [23/Dec/2019:14:34:56 -0700] "GET /mysql-admin/index.php?lang=en HTTP/1.1" 30
39.106.183.168 - - [23/Dec/2019:14:34:56 -0700] "GET /admin/phpmyadmin/index.php?lang=en HTTP/1.
39.106.183.168 - - [23/Dec/2019:14:34:56 -0700] "GET /admin/phpMyAdmin/index.php?lang=en HTTP/1.
```

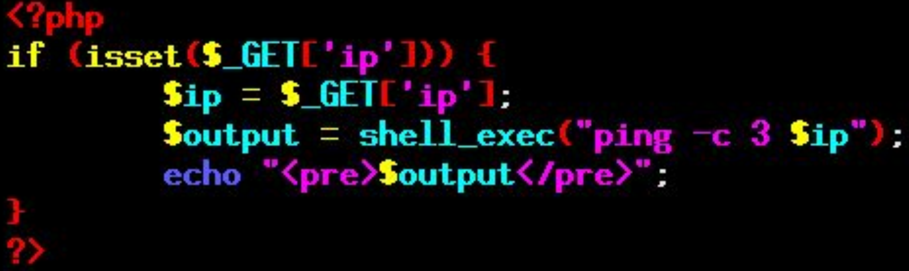
DP3: Complete mediation

- every access to every asset must be checked for authority
- access right are validated every time
  - authority may change
  - access level may change
  - attacker might have bypassed earlier validation code

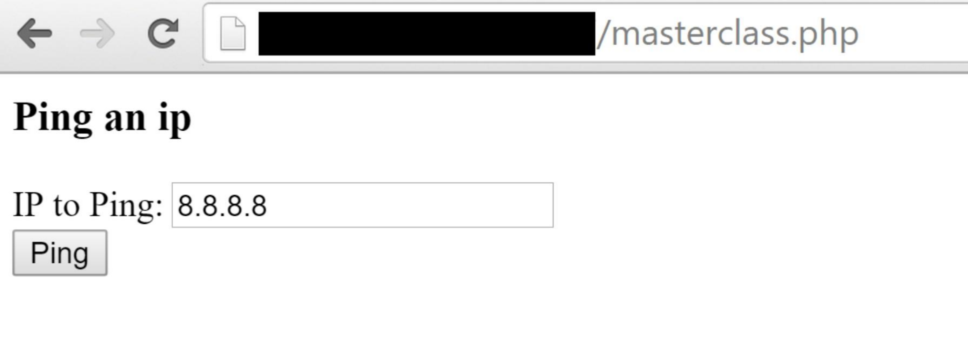
User-supplied data

- inputs to programs are often supplied by untrusted users
  - e.g., web applications and authentication dialogs
  - uesars can someitmes msistype as they intput
- verify all received data conforms to expected or assumed properties
  - never assume anything about input data
  - especially when it is spurious input from the Internet
- sanitize inputs
- canonicalize inputs

Malicious users can craft special input to change how programs behave



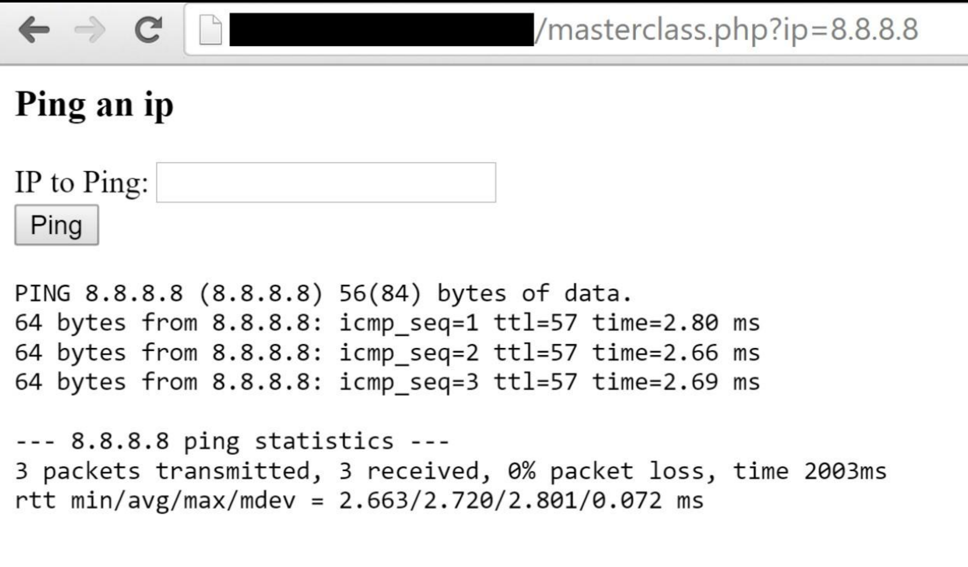
```php
<?php
if (isset($_GET['ip'])) {
        $ip = $_GET['ip'];
        $output = shell_exec("ping -c 3 $ip");
        echo "<pre>$output</pre>";
}
?>
```



[redacted]/masterclass.php

Ping an ip

IP to Ping: 8.8.8.8

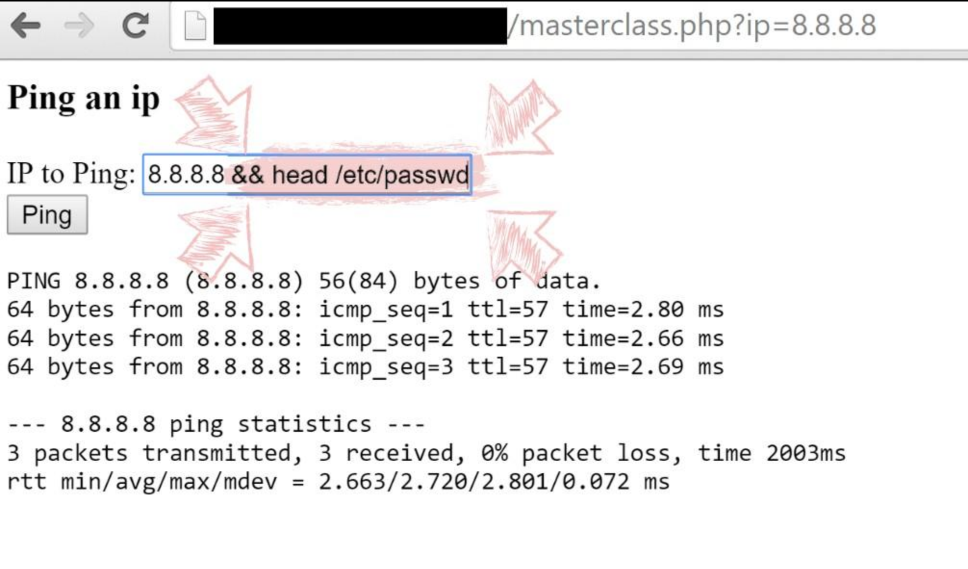


Ping an ip

IP to Ping:

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=2.80 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=2.66 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=2.69 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.663/2.720/2.801/0.072 ms
```



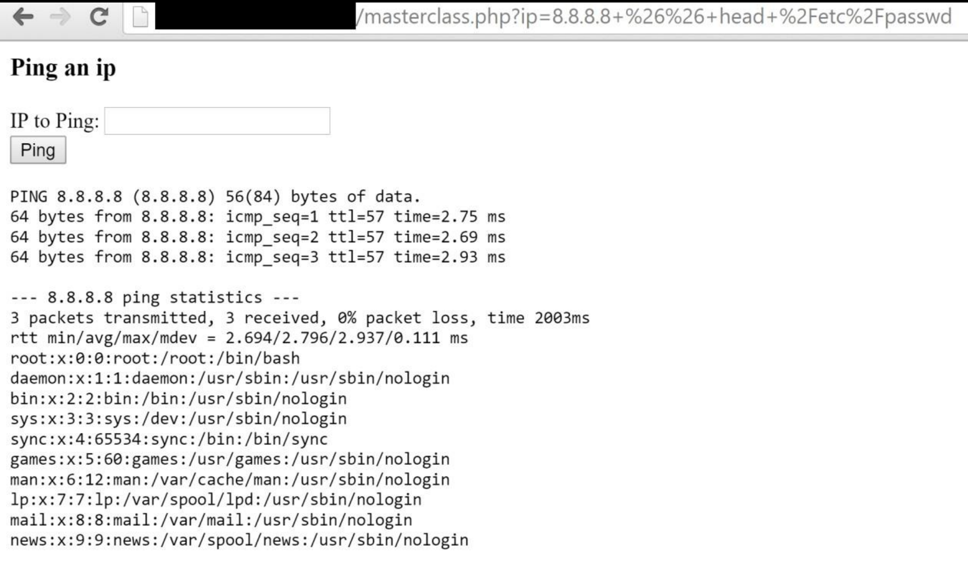
[redacted]/masterclass.php?ip=8.8.8.8

Ping an ip

IP to Ping: 8.8.8.8 && head /etc/passwd

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=2.80 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=2.66 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=2.69 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.663/2.720/2.801/0.072 ms
```



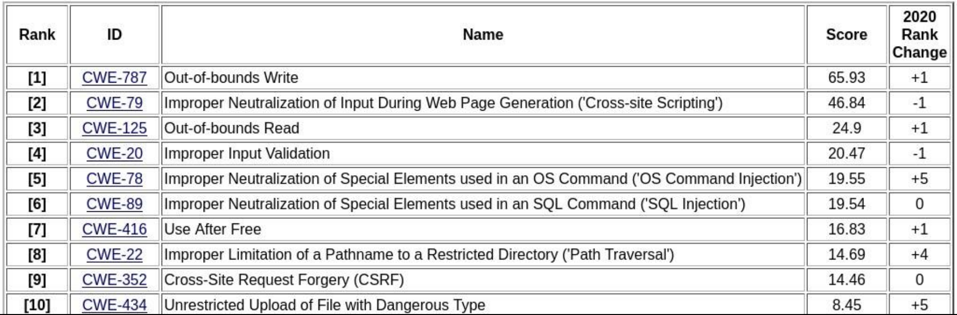
[redacted]/masterclass.php?ip=8.8.8.8+%26%26+head+%2Fetc%2Fpasswd

Ping an ip

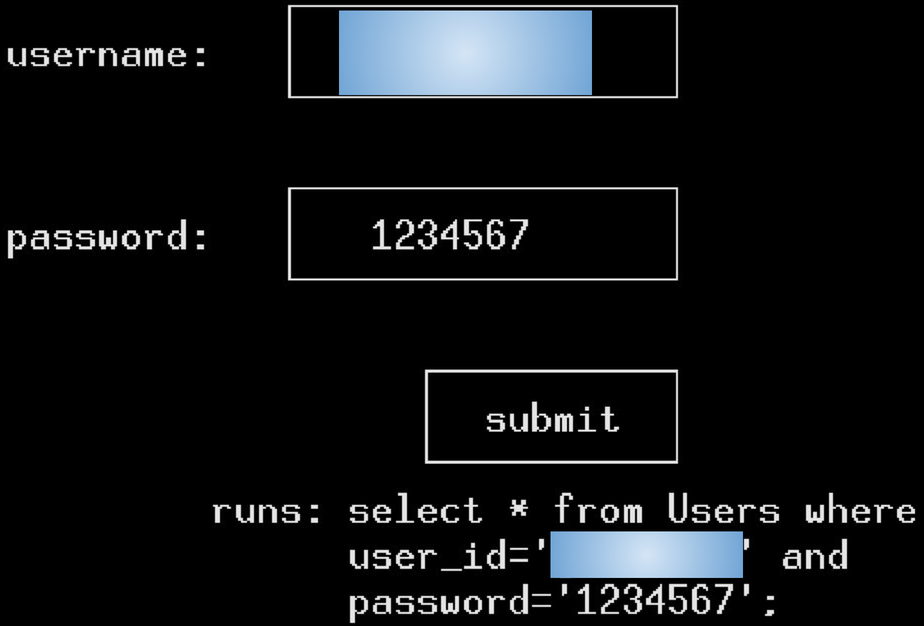
IP to Ping:

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=2.75 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=2.69 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=2.93 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.694/2.796/2.937/0.111 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```



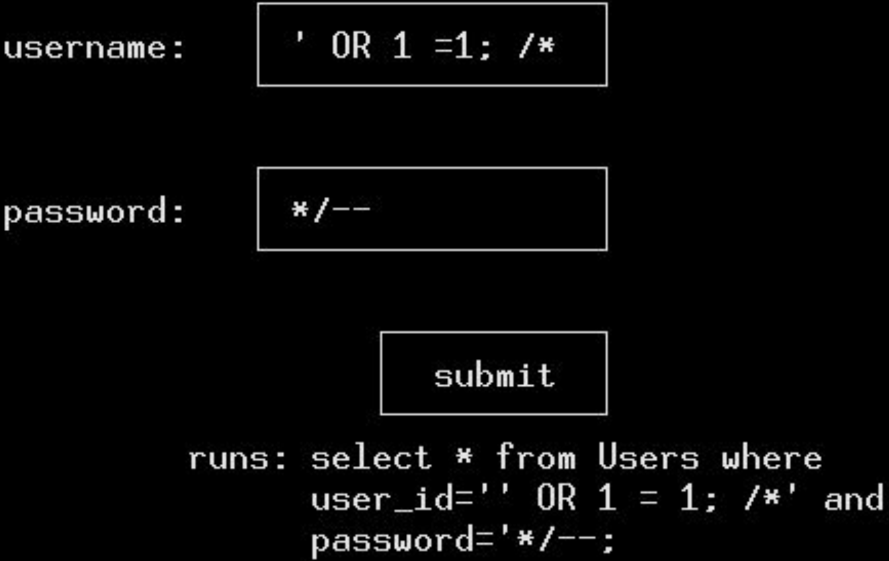
Rank	ID	Name	Score	2020 Rank Change
[1]	CWE-787	Out-of-bounds Write	65.93	+1
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	46.84	-1
[3]	CWE-125	Out-of-bounds Read	24.9	+1
[4]	CWE-20	Improper Input Validation	20.47	-1
[5]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	19.55	+5
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	19.54	0
[7]	CWE-416	Use After Free	16.83	+1
[8]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.69	+4
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	14.46	0
[10]	CWE-434	Unrestricted Upload of File with Dangerous Type	8.45	+5



username:

password: 1234567

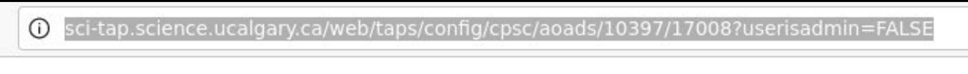
```
runs: select * from Users where
      user_id='' and
      password='1234567';
```



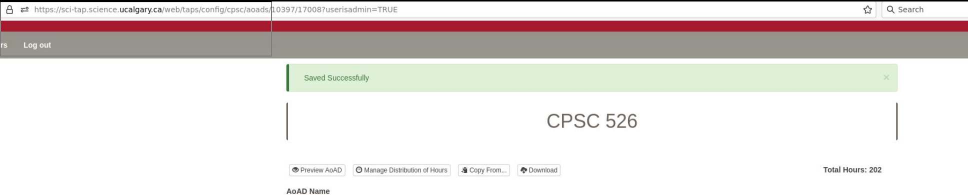
username: ' OR 1 =1; /*

password: */--

```
runs: select * from Users where
      user_id='' OR 1 = 1; /*' and
      password='*/--;
```



sci-tap.science.ucalgary.ca/web/taps/config/cpsc/aoads/10397/17008?userisadmin=FALSE


https://sci-tap.science.ucalgary.ca/web/taps/config/cpsc/aoads/10397/17008?userisadmin=TRUE
Search
rs
Log out
Preview AoAD
Manage Distribution of Hours
Copy From...
Download

Saved Successfully

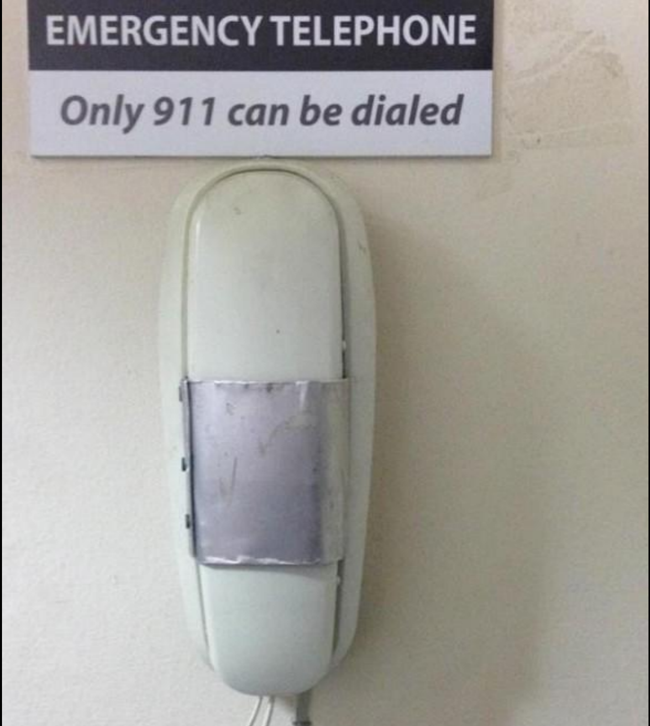
CPSC 526

Total Hours: 202

AoAD Name

Client-side Mediation

- many web forms perform client-side mediation
  - clicking “submit” triggers JavaScript code that validates data before sending to server
- many websites keep client-side state
  - data in hidden fields, cookies, URLs
- problems with this?
  - user can disable JavaScript
  - user can edit hidden form fields, cookies, URLs
  - user can interact with server using, e.g., telnet


EMERGENCY TELEPHONE
Only 911 can be dialed

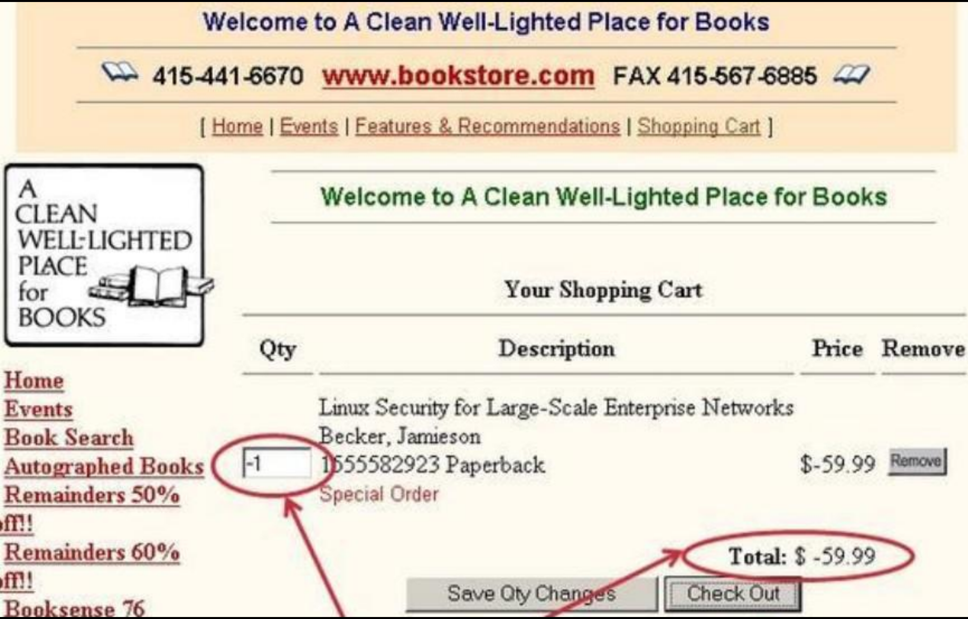
A
B
C
1
WXYZ
9

Scenario

- A user wishes to purchase a widget from an online store.
- Server replies with a form asking for shipping and billing info
- Form has the following hidden fields:

```
<input type=“hidden” name=“productid” value=“42”>
<input type=“hidden” name=“quantity” value=“1”>
<input type=“hidden” name=“unitprice” value=“111.00”>
```

- What happens if user changes “unitprice” to “0.00” before submitting?



Welcome to A Clean Well-Lighted Place for Books

415-441-6670 www.bookstore.com FAX 415-567-6885

[Home | Events | Features & Recommendations | Shopping Cart]
A CLEAN WELL-LIGHTED PLACE for BOOKS
Home
Events
Book Search
Autographed Books
Remainders 50% off!!
Remainders 60% off!!
Booksense 76

Welcome to A Clean Well-Lighted Place for Books

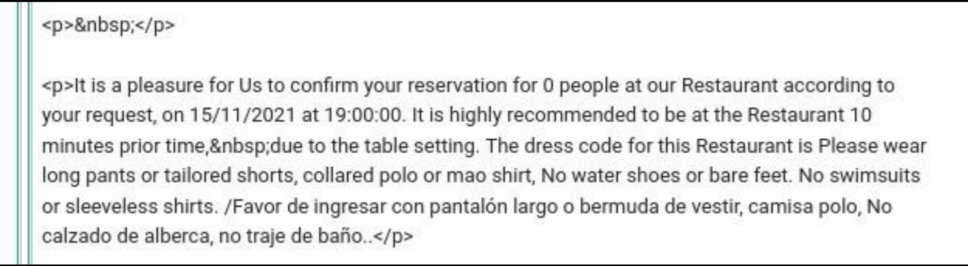
Your Shopping Cart

Qty	Description	Price	Remove
-1	Linux Security for Large-Scale Enterprise Networks Becker, Jamieson 1555582923 Paperback Special Order	$-59.99	Remove

Total: $ -59.99

Client-side mediation

- Client-side mediation is useful for friendlier user interfaces
  - but it’s useless for security purposes!
- Always, always, ALWAYS do security-relevant mediation at the server!
- Values can be arbitrary
  - never assume text fields only contain only valid ASCII

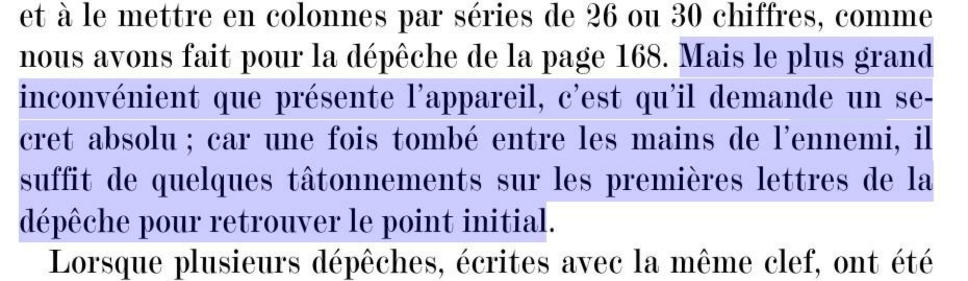


<p>&nbsp;</p>

<p>It is a pleasure for Us to confirm your reservation for 0 people at our Restaurant according to your request, on 15/11/2021 at 19:00:00. It is highly recommended to be at the Restaurant 10 minutes prior time, due to the table setting. The dress code for this Restaurant is Please wear long pants or tailored shorts, collared polo or mao shirt, No water shoes or bare feet. No swimsuits or sleeveless shirts. /Favor de ingresar con pantalón largo o bermuda de vestir, camisa polo, No calzado de alberca, no traje de baño..</p>

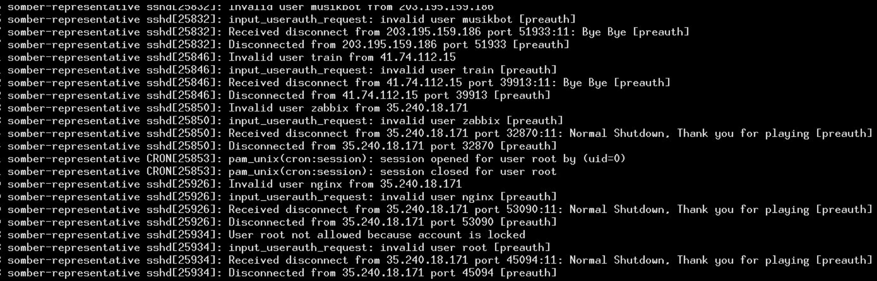
DP4: Open design

- don’t rely on secret design or attacker ignorance
  - “don’t rely on security through obscurity”
  - “the enemy knows the system”
  - Kerckhoff’s principle
- invite open review and analysis
- yet, leverage unpredictability if there is no disadvantage
  - e.g., no gain to publish blueprints, or your vacation schedule



et à le mettre en colonnes par séries de 26 ou 30 chiffres, comme nous avons fait pour la dépêche de la page 168. Mais le plus grand inconvénient que présente l'appareil, c'est qu'il demande un secret absolu ; car une fois tombé entre les mains de l'ennemi, il suffit de quelques tâtonnements sur les premières lettres de la dépêche pour retrouver le point initial.

Lorsque plusieurs dépêches, écrites avec la même clef, ont été



```
somber-representative sshd[25832]: Invalid user musikbot from 203.195.159.186
somber-representative sshd[25832]: input_userauth_request: invalid user musikbot [preauth]
somber-representative sshd[25832]: Received disconnect from 203.195.159.186 port 51933:11: Bye Bye [preauth]
somber-representative sshd[25832]: Disconnected from 203.195.159.186 port 51933 [preauth]
somber-representative sshd[25846]: Invalid user train from 41.74.112.15
somber-representative sshd[25846]: input_userauth_request: invalid user train [preauth]
somber-representative sshd[25846]: Received disconnect from 41.74.112.15 port 39913:11: Bye Bye [preauth]
somber-representative sshd[25846]: Disconnected from 41.74.112.15 port 39913 [preauth]
somber-representative sshd[25850]: Invalid user zabbix from 35.240.18.171
somber-representative sshd[25850]: input_userauth_request: invalid user zabbix [preauth]
somber-representative sshd[25850]: Received disconnect from 35.240.18.171 port 32870:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25850]: Disconnected from 35.240.18.171 port 32870 [preauth]
somber-representative CRON[25853]: pam_unix(cron:session): session opened for user root by (uid=0)
somber-representative CRON[25853]: pam_unix(cron:session): session closed for user root
somber-representative sshd[25926]: Invalid user nginx from 35.240.18.171
somber-representative sshd[25926]: input_userauth_request: invalid user nginx [preauth]
somber-representative sshd[25926]: Received disconnect from 35.240.18.171 port 53090:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25926]: Disconnected from 35.240.18.171 port 53090 [preauth]
somber-representative sshd[25934]: User root not allowed because account is locked
somber-representative sshd[25934]: input_userauth_request: invalid user root [preauth]
somber-representative sshd[25934]: Received disconnect from 35.240.18.171 port 45094:11: Normal Shutdown, Thank you for playing [preauth]
somber-representative sshd[25934]: Disconnected from 35.240.18.171 port 45094 [preauth]
```

These attacks all stop when ssh is moved from port 22 to port 2222

DP5: Separation of privilege

DP5: Separation of privilege

Where feasible, a protection mechanism that requires 2 keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

DP5: Separation of privilege

Where feasible, a protection mechanism that requires 2 keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

Prevent unilateral action by a subverted individual.

ENTRY RESTRICTED
TO
MCCC AND DMCCC
ON DUTY
DIEBOLD

DP6: Least privilege

- Allocate the fewest privileges needed for a task and for the shortest duration necessary
- e.g., use root to do something and then exit terminal
- e.g., sudo and sudo session riding
- e.g., don’t give every app access to the microphone
- “need-to-know basis”

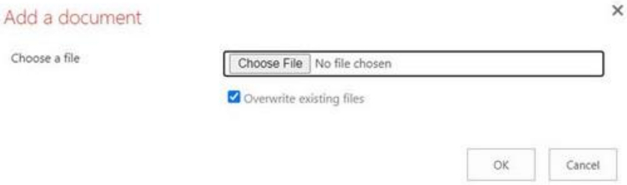
DP7: Least common mechanism

- Minimize the amount of mechanisms
  - common to more than one user and
  - depended on by all users
- examples:
  - shared variables
  - shared storage
- Shared mechanisms might provide unintended communication paths or means of interference

DP8: Psychological acceptability

- Design mechanism and interfaces to behave as users expect
- Align design with mental model
    - especially when errors are irreversible
- Beware of designs suited to trained experts or which require training
- “least surprise”

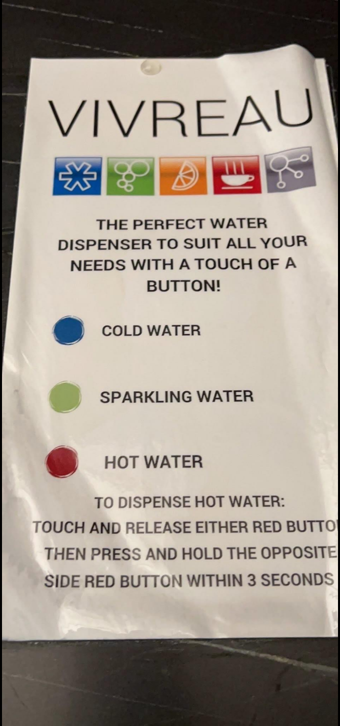
The files should be named differently. But the other way to not overwrite is to uncheck the 'overwrite existing files' check box.



Add a document

Choose a file: Choose File No file chosen

☑ Overwrite existing files



VIVREAU

THE PERFECT WATER DISPENSER TO SUIT ALL YOUR NEEDS WITH A TOUCH OF A BUTTON!

🔵 COLD WATER

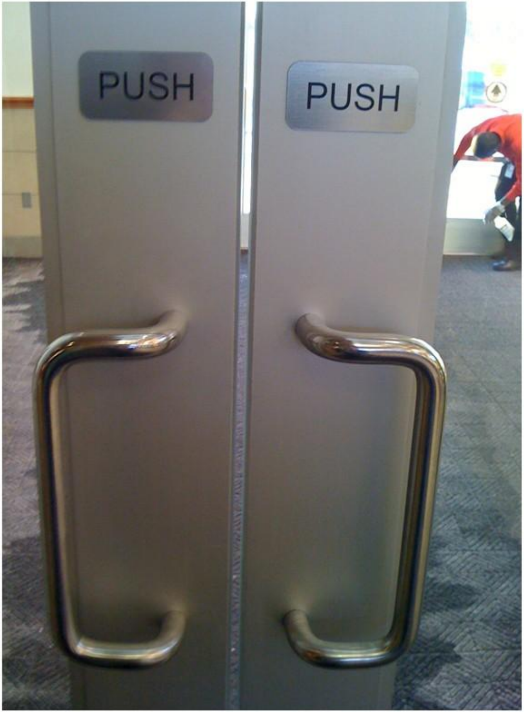
🟢 SPARKLING WATER

🔴 HOT WATER

TO DISPENSE HOT WATER:
TOUCH AND RELEASE EITHER RED BUTTO
THEN PRESS AND HOLD THE OPPOSITE
SIDE RED BUTTON WITHIN 3 SECONDS

PUSH

PULL
PUSH


PUSH
PUSH

DP9: Time-tested tools

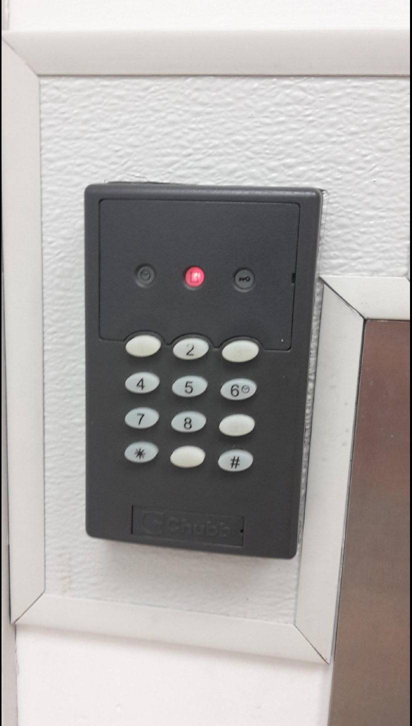
- Rely on established methods to accomplish security
    - protocols, primitives, toolkits
- Heavily scrutinized tools are less likely flawed
- “don’t roll your own crypto”
- Reinventing the wheel is a great way to learn but not a great way to do security

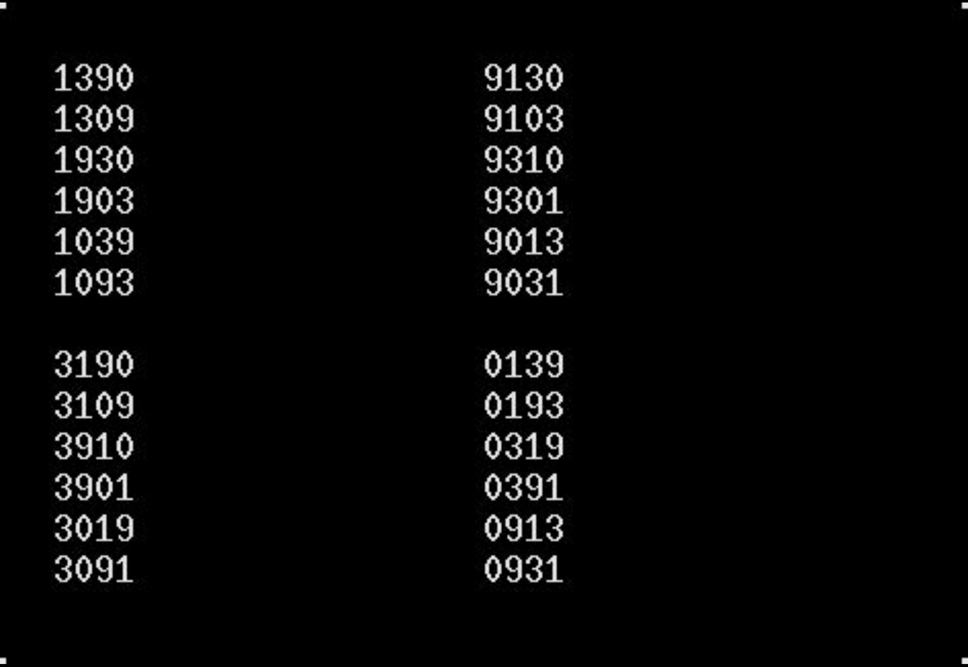
DP10: Evidence production

- log system activities that can promote accountability
  - e.g., when sudo is used by an authorized party
  - e.g., when someone logs into a server
  - e.g., when someone plugs in a USB stick
  - e.g., when someone accesses a file
  - e.g., or have decoy “honey” files that should never be accessed
  - e.g., when certain files or special directories are modified
- this helps discover attacks, determine effect
- help build intrusion-detection tools

DP11: remnant removal

- remove all traces of critical information
  - don’t store keys in RAM after done using
  - don’t save decrypted data to storage medium
  - **securely delete** files you do not want
  - don’t log all interactions with a program
    - unless supporting evidence production
    - have a plan for sanitizing long term logs


2
4
5
6
7
8
*
#
Chubb

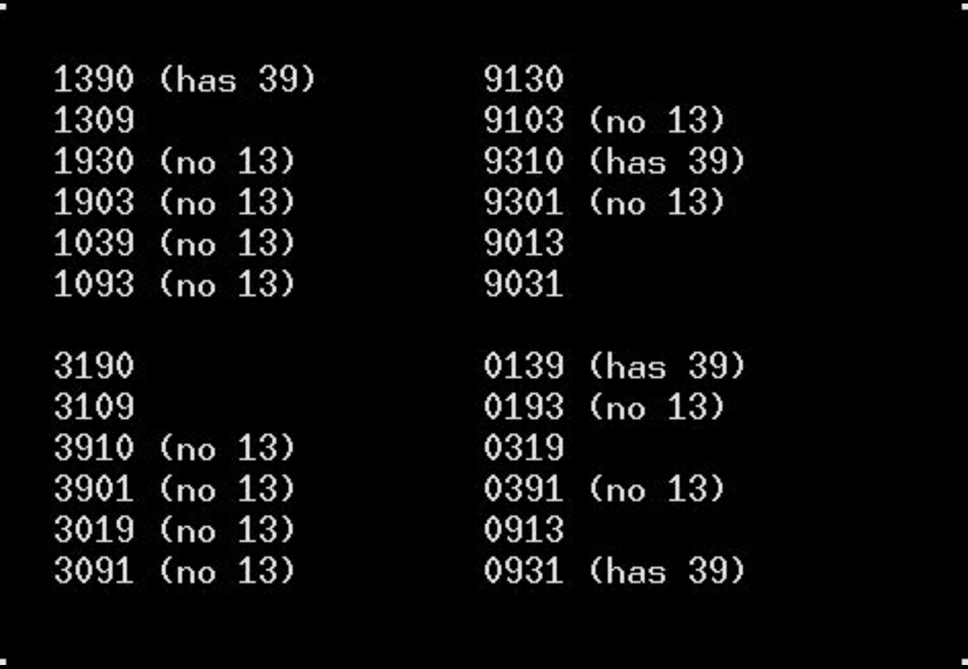


```
1390                 9130
1309                 9103
1930                 9310
1903                 9301
1039                 9013
1093                 9031

3190                 0139
3109                 0193
3910                 0319
3901                 0391
3019                 0913
3091                 0931
```

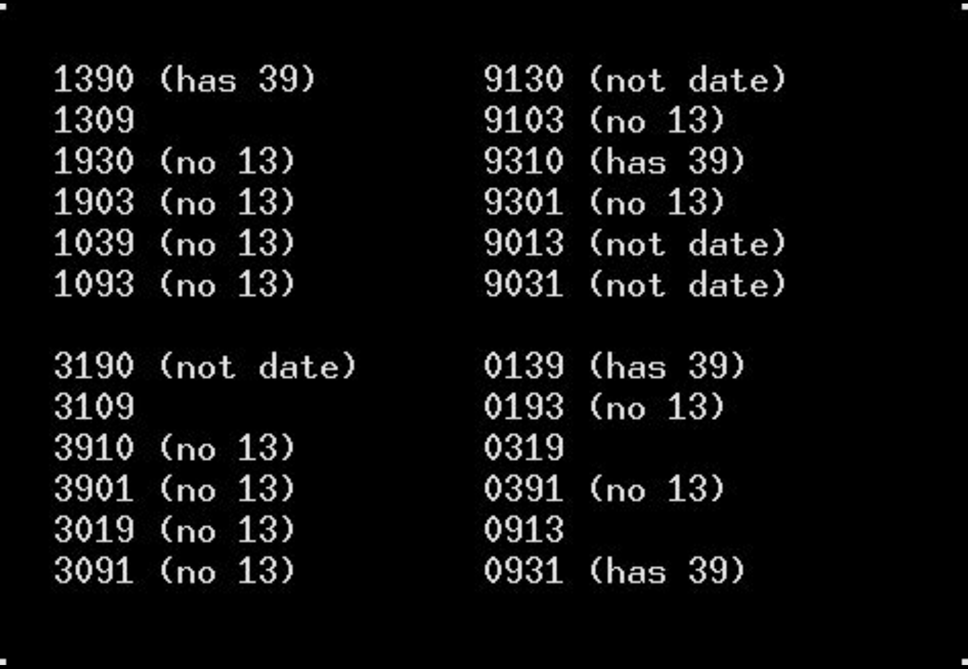
```
1390                    9130
1309                    9103 (no 13)
1930 (no 13)            9310
1903 (no 13)            9301 (no 13)
1039 (no 13)            9013
1093 (no 13)            9031

3190                    0139
3109                    0193 (no 13)
3910 (no 13)            0319
3901 (no 13)            0391 (no 13)
3019 (no 13)            0913
3091 (no 13)            0931
```



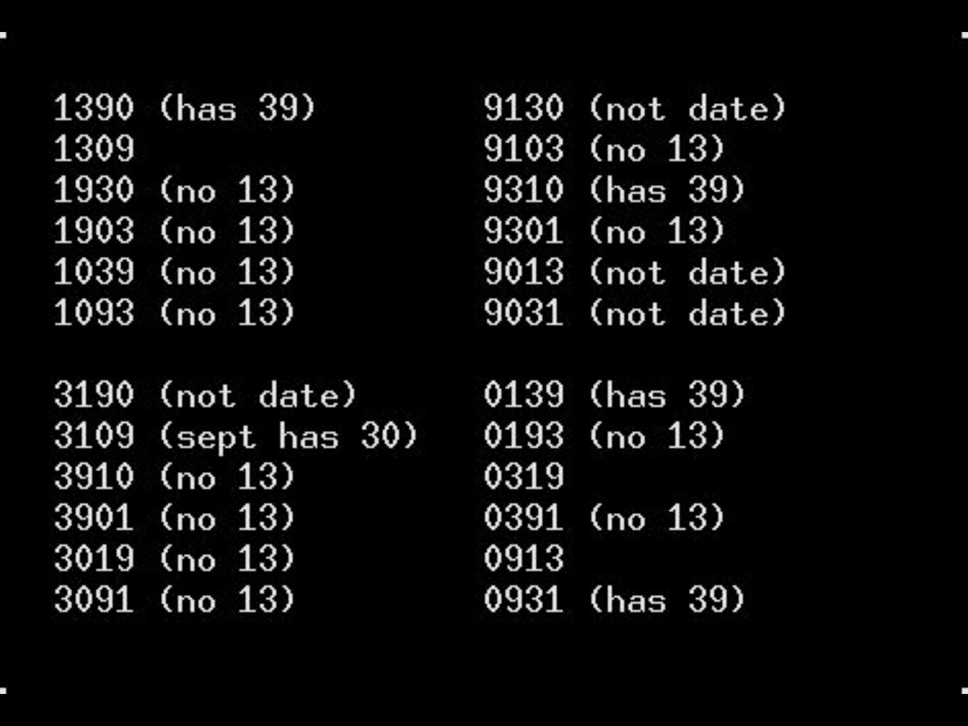
```
1390 (has 39)        9130
1309                 9103 (no 13)
1930 (no 13)         9310 (has 39)
1903 (no 13)         9301 (no 13)
1039 (no 13)         9013
1093 (no 13)         9031

3190                 0139 (has 39)
3109                 0193 (no 13)
3910 (no 13)         0319
3901 (no 13)         0391 (no 13)
3019 (no 13)         0913
3091 (no 13)         0931 (has 39)
```



```
1390 (has 39)          9130 (not date)
1309                   9103 (no 13)
1930 (no 13)           9310 (has 39)
1903 (no 13)           9301 (no 13)
1039 (no 13)           9013 (not date)
1093 (no 13)           9031 (not date)

3190 (not date)        0139 (has 39)
3109                   0193 (no 13)
3910 (no 13)           0319
3901 (no 13)           0391 (no 13)
3019 (no 13)           0913
3091 (no 13)           0931 (has 39)
```



```
1390 (has 39)          9130 (not date)
1309                   9103 (no 13)
1930 (no 13)           9310 (has 39)
1903 (no 13)           9301 (no 13)
1039 (no 13)           9013 (not date)
1093 (no 13)           9031 (not date)

3190 (not date)        0139 (has 39)
3109 (sept has 30)     0193 (no 13)
3910 (no 13)           0319
3901 (no 13)           0391 (no 13)
3019 (no 13)           0913
3091 (no 13)           0931 (has 39)
```

1309

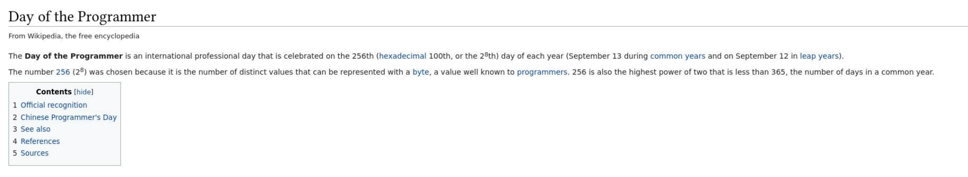
0319

0913

1309 - sept 13 (in day month?)

0319 (march 19)

0913 (sept 13)



Day of the Programmer

From Wikipedia, the free encyclopedia

The **Day of the Programmer** is an international professional day that is celebrated on the 256th (hexadecimal 100th, or the 2^8th) day of each year (September 13 during common years and on September 12 in leap years).

The number 256 (2^8) was chosen because it is the number of distinct values that can be represented with a byte, a value well known to programmers. 256 is also the highest power of two that is less than 365, the number of days in a common year.

Contents [hide]

DP12: reluctant allocation

- Be reluctant to expend effort or allocate resources
  - especially with unauthenticated external agents
- Be reluctant to extend privileges or act on someone’s behalf
- Place burden of proof of identity on those who initiate communication
  - e.g., the person who calls should not demand: “Who am I speaking with?”
  - e.g., if they are from the bank you should call the bank to reconnect

DP13: security by design

- Build security in from the start
- Don’t staple it on at a late stage
- Don’t add security purposes to something not designed for it
    - e.g., social insurance numbers
- explicitly state design goals of security mechanisms
- explicitly state what they are not designed to do
- explicitly state assumptions, especially involving trust

DP14: security is economics

- attacks and defenses have costs
- consequences of attacks have costs
- real world security balances these

DP15: Defence in depth

- use multiple layers, each backing up the other
  - attackers must defeat independent layers
- design each to be comparably strong
  - strengthen the weakest one first
  - “attackers break the weakest link”
- assume attacker will bypass some or layer will fail

DP16: know your adversary

- security is designed and defined relative to an adversary
- adversary wants to break security
- understanding the adversary gives better security design

Adversary Modelling

- an adversary model:
    - identifies objectives
        - e.g., target assets
    - methods
        - attacker techniques, types of attacks
    - capabilities
        - computing resources, skills, knowledge, opportunity
    - funding level
        - correlates with determination and persistence

Adversary Modelling

- A **categorical schema** classifies well-defined adversaries into groups
- for example
  - foreign intelligence
  - terrorists
  - politically motivated adversaries
  - industrial espionage agents
  - organized crime
  - lesser criminals, e.g., “script kiddies”
  - malicious insiders, e.g., disgruntled employees
  - non-malicious employees, e.g., USB stick pluggers-in
  - researchers, casual hackers, and bug bounty hunters

Types of Attacks

- Passive attack
  - nothing is different as a result of attacker being present
  - same data is exchanged, communication happens without interference
  - attacker eavesdrops
    - “man-in-the-middle” attack
- Active attack
  - attacker interferes with communication
  - inserts data, removes data, modifies data, replays data

Types of Attacks

- targeted attacks
  - aimed at specific individuals or organizations
    - e.g., stuxnet
    - log into CEO’s account
- opportunistic attacks or generic attacks
  - aimed at arbitrary victims
    - e.g., log into anyone’s account
    - bike locks are an example defence (won’t work for targeted)

Types of Attacks

- Outsider attack
  - has no special access to target network
- Insider attack
  - party has some advantage over outsiders
  - taking over one account may boost outsider to insider

Threat Model

- what attacks do you consider
- what assets you need to defend
- who is your adversary
- bad threat models
  - give false sense of security
  - have invalid assumptions and misplaced trust
  - focus on the wrong threats

The modelled adversary is meant to characterize the capabilities of the real attacker

The modelled adversary is meant to characterize the capabilities of the real attacker

This corresponds to the real-world scenario.

The modelled adversary is meant to characterize the capabilities of the real attacker

This corresponds to the real-world scenario.

The more accurate the model, the better suited any security meant to thwart them.

Example: insider threat

- the adversary
  - knows the system
  - has keys to rooms and passwords to machines
  - has friends who may be **confused deputies**
- frequently, systems defend well against external threats

Privilege escalation attack

- technique that insider can use
- given some amount of privileges, exploit the system to gain more
- e.g., master keys lock systems
- **social engineering** is often characterized by privilege escalation.

Resources

[1]. Chapter 1 - Computer Security and the Internet: Tools and Jewels