Intro to security 2

Malware

Vulnerabilities

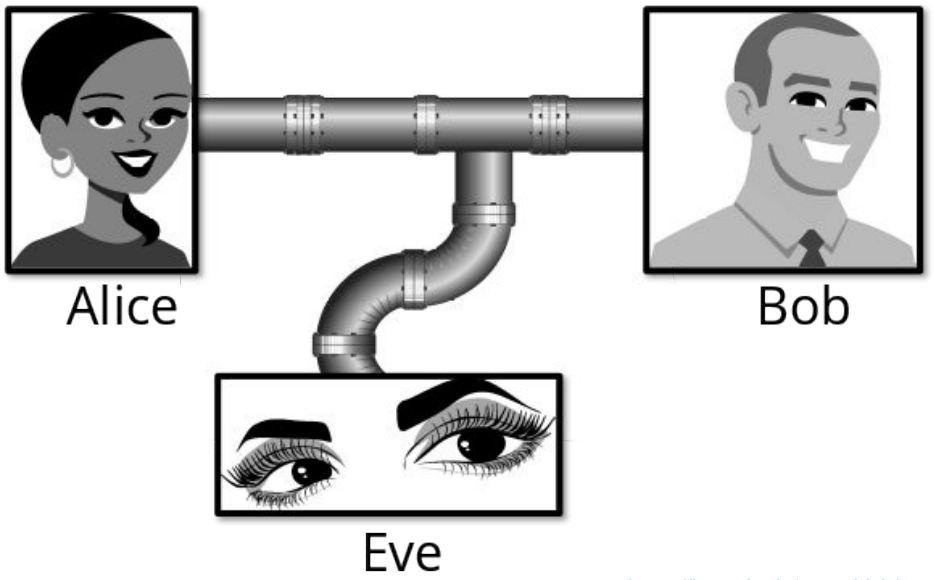
Network attacks

Web attacks

Tools

Cryptography

Encryption: a means to allow two parties, customarily called Alice and Bob, to establish confidential communication over an insecure channel that is subject to eavesdropping.



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Last lecture: Encryption and Decryption

The message M is called the **plaintext**.

Alice will convert **plaintext M** to an encrypted form

- Alice uses an **encryption algorithm E** that outputs **a ciphertext C** for M.

As equations:

C = E(M)

M = D(C)

The encryption and decryption algorithms are chosen so that it is infeasible for someone other than Alice and Bob to determine plaintext M from ciphertext C.

ciphertext C can be transmitted over an insecure channel that can be eavesdropped by an adversary.

Classical cryptosystems (single-key or symmetric cryptosystems)

Substitution Ciphers

A substitution cipher changes characters in the plaintext to produce the ciphertext.

- **Transposition ciphers**
- **Cesar cipher**
- **Vigenère cipher**
  - **A longer key, uses tableau**

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*
A	A	B	C	D	E	F	G	H	I	J	K	L	M
B	B	C	D	E	F	G	H	I	J	K	L	M	N
C	C	D	E	F	G	H	I	J	K	L	M	N	O
D	D	E	F	G	H	I	J	K	L	M	N	O	P
E	E	F	G	H	I	J	K	L	M	N	O	P	Q
F	F	G	H	I	J	K	L	M	N	O	P	Q	R
G	G	H	I	J	K	L	M	N	O	P	Q	R	S

EXAMPLE: The first line of a limerick is enciphered using the key "BENCH," as follows.

Key	B	ENCHBENC	HBENC	HBENCH	BENCHBENCH
Plaintext	A	LIMERICK	PACKS	LAUGHS	ANATOMICAL
Ciphertext	B	PVOLSMPM	WBGXU	SBYTJZ	BRNVVNMPCS

One-Time Pad: if the key as long as the text

Public Key Cryptography

Two keys: encipherment and decipherment keys

Public (encipherment) **key** is public!

Private (decipherment) **key** is know only to owner!

RSA

$$c = m^e \bmod n$$

$$m = c^d \bmod n$$

- **φ(n)**
  - **the number of numbers less than n with no factors in common with n**
- **Choose e: e < n relatively prime to φ(n).**
- **Find d: ed mod φ(n) = 1**

The public key is (e, n)

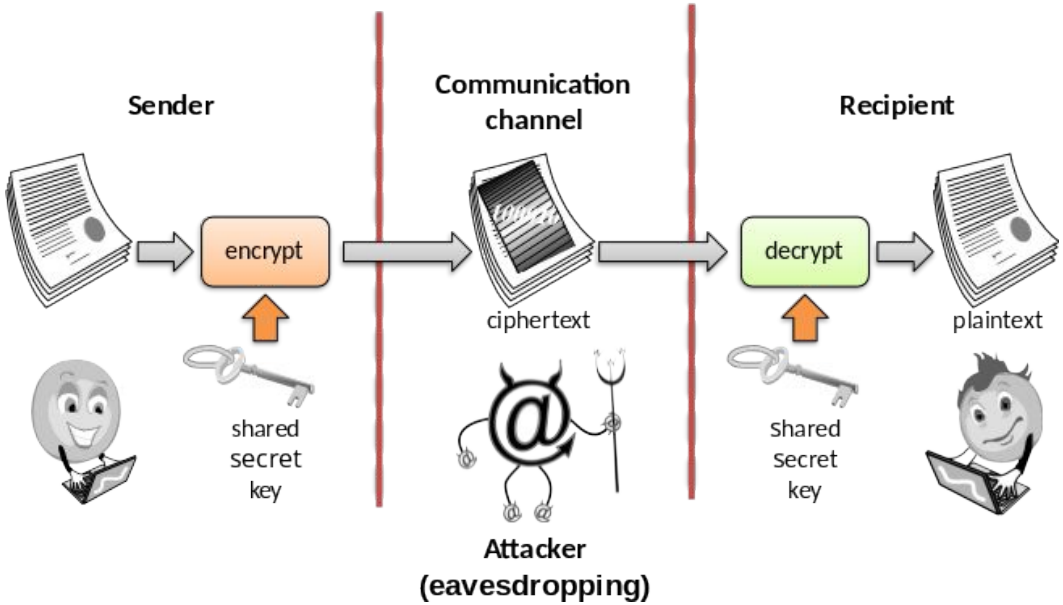
the private key is d

n = pq, p and q primes

In addition to confidentiality, RSA can provide data and origin authentication.

Symmetric cryptosystems

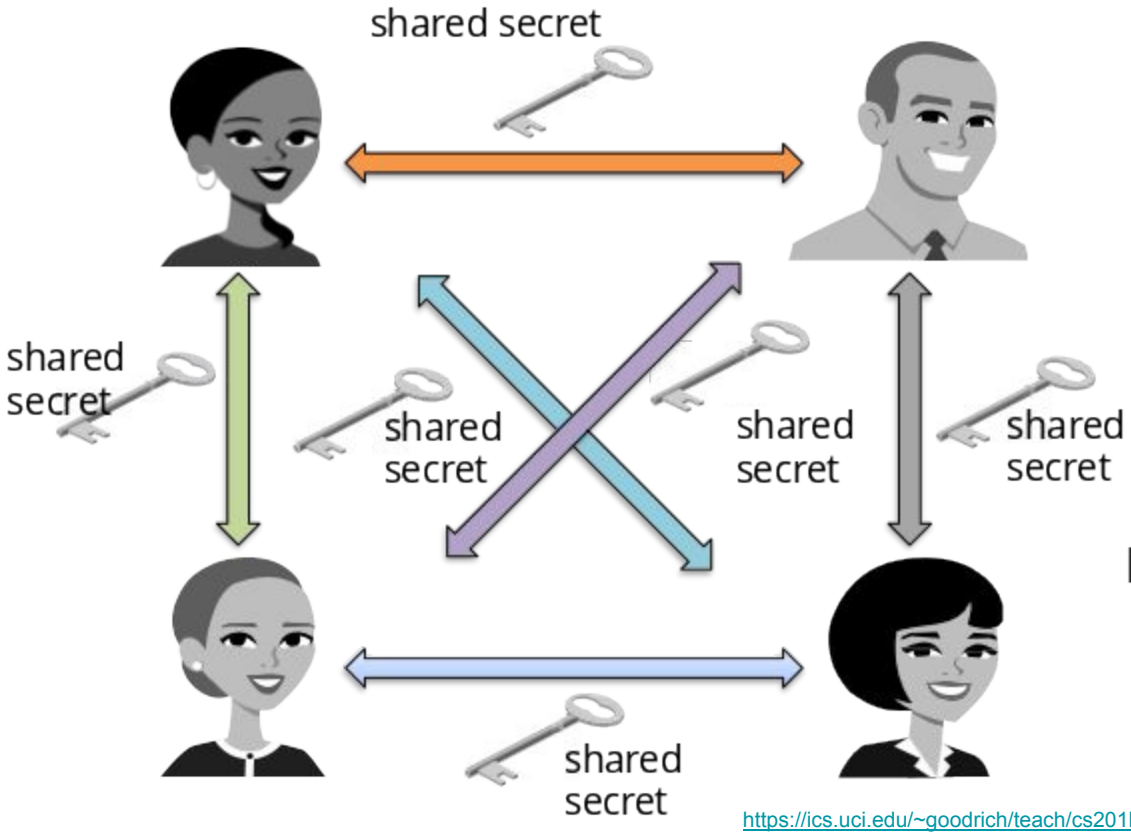
Alice and Bob share a secret key, which is used for both encryption and decryption.



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Symmetric key Distribution

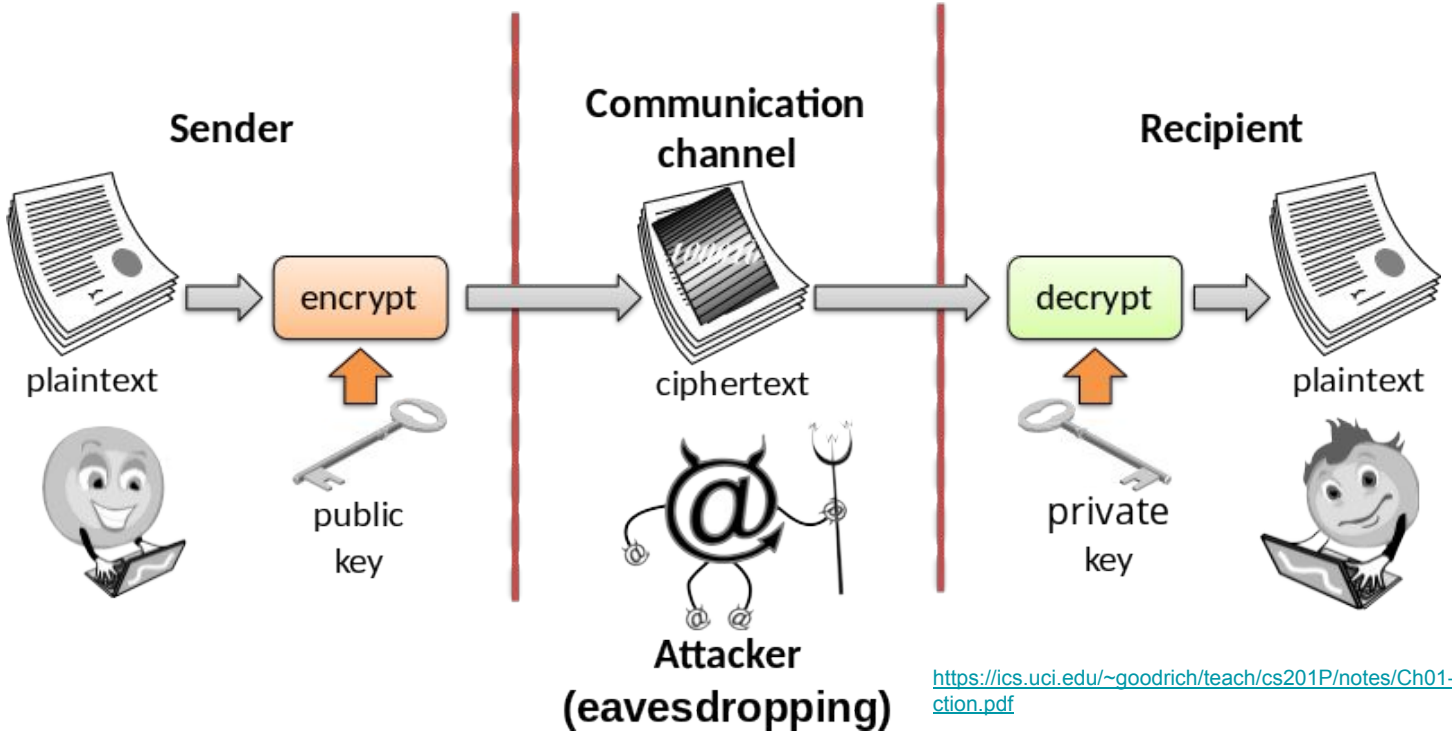
Requires each pair of communicating parties to share a (separate) secret key.



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Last lecture: Public-Key Cryptography

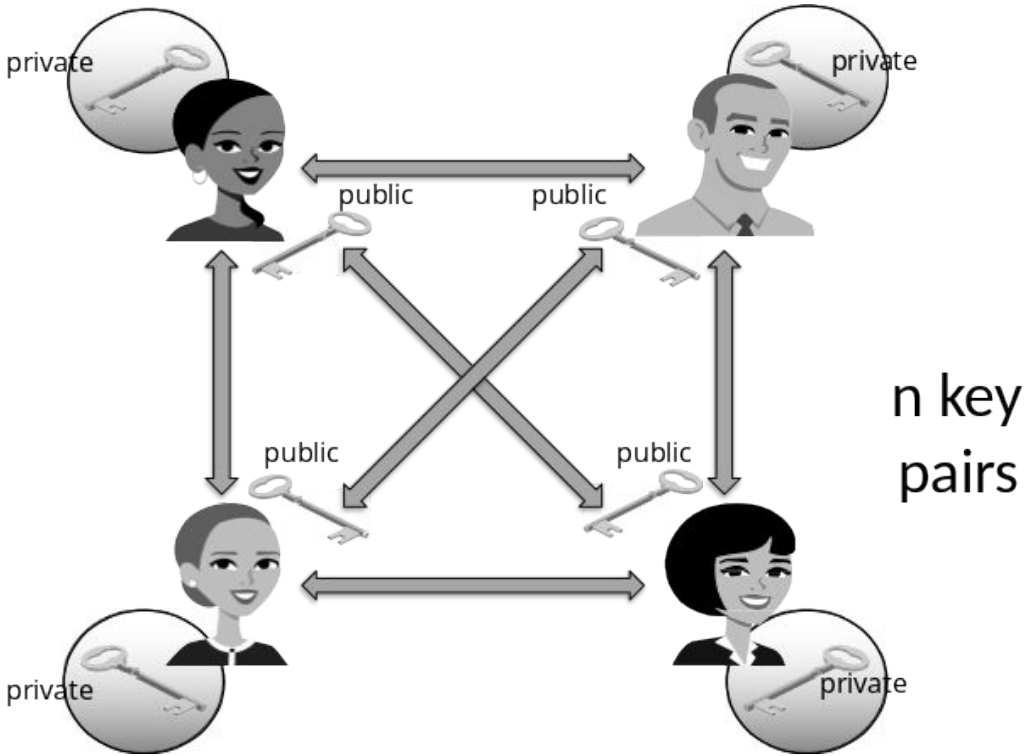
Separate keys are used for encryption and decryption.



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Public Distribution

Only one key is needed for each recipient



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Digital Signatures

Public-key encryption provides a method for doing digital signatures

To sign a message, M,

- Alice just encrypts it with her private key, SA, creating **$C = E_{SA}(M)$.**

Anyone can decrypt this message using Alice's public key,

- $M' = D_{PA}(C)$, and compare that to the message M.

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Cryptographic Hash Functions

A checksum on a message, M, that is:

One-way:

- it should be easy to compute Y=H(M),
- but hard to find M given only Y

Collision-resistant:

- it should be hard to find two messages, M and N, such that H(M)=H(N).

Examples: SHA-1, SHA-256.

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

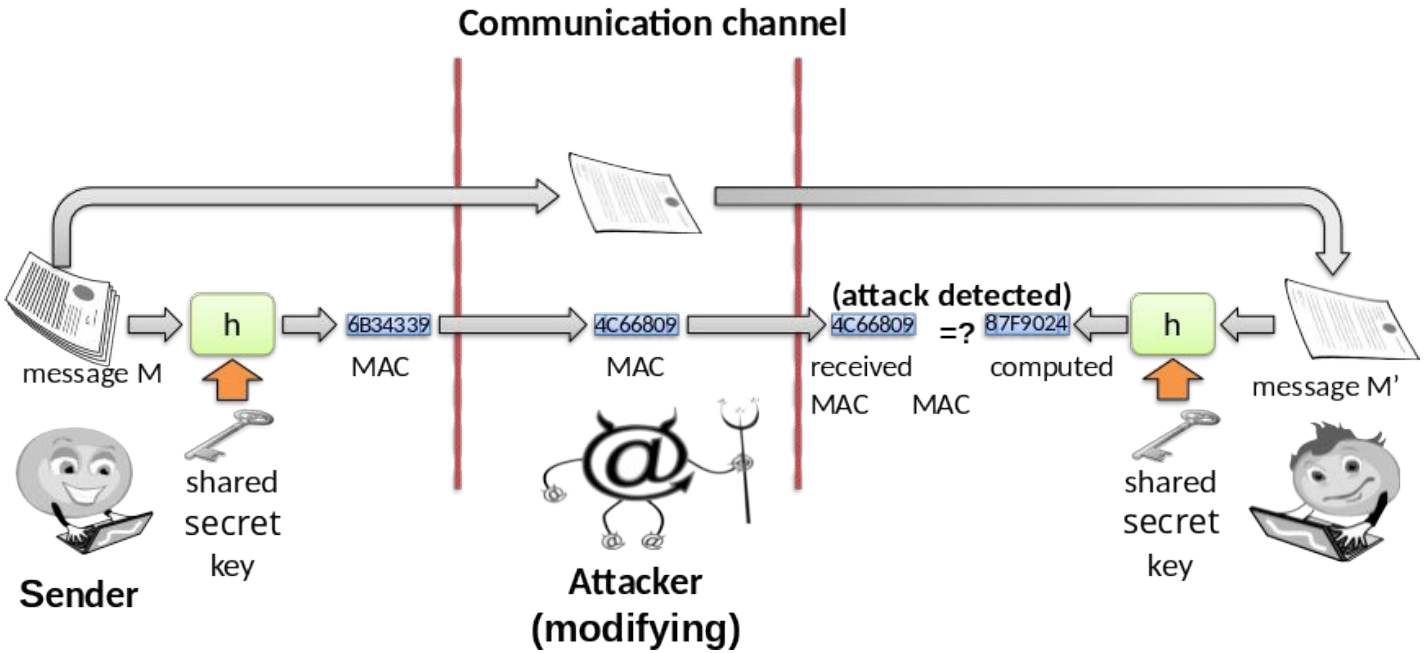
Message Authentication Codes

Allows for Alice and Bob to have data integrity, if they share a secret key.

Given a message M,

Alice computes H(K||M)

and sends M and this hash to Bob.



https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Digital Certificates

certificate authority (CA) digitally signs a binding between an identity and the public key for that identity.

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Passwords

A short sequence of characters used as a means to authenticate someone via a secret that they know.

Userid:

Password:

What is a strong password

- UPPER/lower case characters
- Special characters
- Numbers

When is a password strong?

Seattle1

M1ke03

P@$$w0rd

TD2k5secV

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Password complexity

Odd characters make pass strong

A fixed 6 symbols password:

- Numbers
  - 10^6 = 1,000,000
- UPPER or lower case characters
  - 26^6 = 308,915,776
- UPPER and lower case characters
  - 52^6 = 19,770,609,664
- 32 special characters
  - (&, %, $, £, “, |, ^, §, etc.)
  - 32^6 = 1,073,741,824

94 practical symbols available

- 94^6 = 689,869,781,056

ASCII standard 7 bit 2^7 =128 symbols

- 128^6 = 4,398,046,511,104

Password length

5 characters: 94^5 =	7,339,040,224
6 characters: 94^6 =	689,869,781,056
7 characters: 94^7 =	64,847,759,419,264
8 characters: 94^8 =	6,095,689,385,410,816
9 characters: 94^9 =	572,994,802,228,616,704

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Password Validity: Brute Force Test

Password does not change for 60 days

how many passwords should I try for each second?

5 characters:	1,415 PW /sec
6 characters:	133,076 PW /sec
7 characters:	12,509,214 PW /sec
8 characters:	1,175,866,008 PW /sec
9 characters:	110,531,404,750 PW /sec

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

Secure Passwords

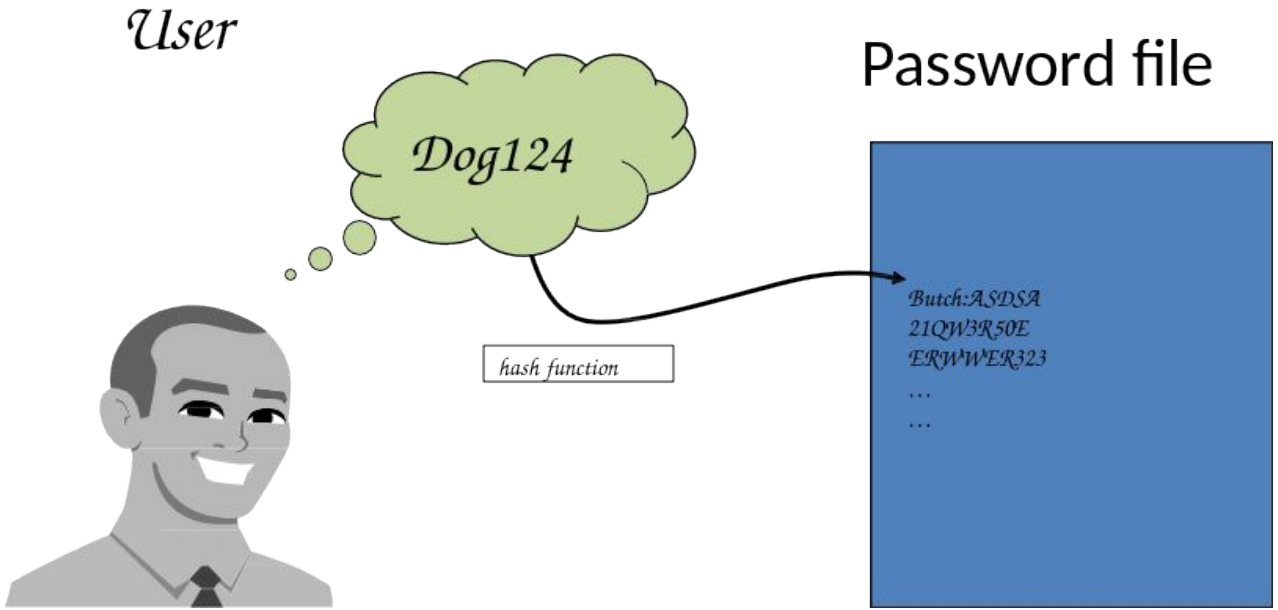
A strong password includes characters from at least three of the following groups:

Group	Example
Lowercase letters	a, b, c, ...
Uppercase letters	A, B, C, ...
Numerals	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Non-alphanumeric (symbols)	( ) ` ~ ! @ # $ % ^ & * - + = | \ { } [ ] : ; " ' < > , . ? /
Unicode characters	€, Γ, ƒ, and λ

eg. "I re@lly want to buy 11 Dogs!"

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

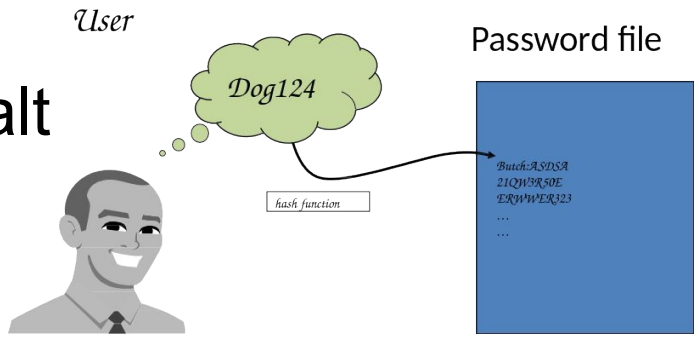
How a password is stored?



★ Add “salt” to avoid the same password being encrypted to the same value
- H(dog124+salt)
- Save salt+ h(dog124+salt)

https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch01-Introduction.pdf

How a password is stored: random salt



Username	String to be hashed	Hashed value = SHA256
user1	**password123**	EF92B778BAFE771E89245B89ECBC08A44A4E166C06659911881F383D4473E94F
user2	**password123**	EF92B778BAFE771E89245B89ECBC08A44A4E166C06659911881F383D4473E94F

Username	Salt value	String to be hashed	Hashed value = SHA256 (Password + Salt value)
user1	D;%yL9TS:5PalS/d	**password123**D;%yL9TS:5PalS/d	9C9B913EB1B6254F4737CE947EFD16F16E916F9D6EE5C1102A2002E48D4C88BD
user2	)<,-<U(jLezy4j>*	**password123**)<,-<U(jLezy4j>*	6058B4EB46BD6487298B59440EC8E70EAE482239FF2B4E7CA69950DFBD5532F2

https://en.wikipedia.org/wiki/Salt_(cryptography)

Malware (malicious software)

A virus

It replicates itself:

- modifies other programs
- and inserts its own code into those programs

A worm

It actively transmits itself over a network to infect other computers and can copy itself without infecting files.

Creeper was the first computer worm, while **Reaper** was the first antivirus software, designed to eliminate Creeper.

https://en.wikipedia.org/wiki/Creeper_and_Reaper

Trojan horse

A Trojan horse misrepresents itself to masquerade as a regular, benign program or utility in order to persuade a victim to install it

Droppers are a sub-type of Trojans that solely aim to deliver malware upon the system that they infect with the desire to subvert detection through stealth and a light payload

Spyware

Programs designed to monitor users' web browsing, display unsolicited advertisements, or redirect affiliate marketing revenues are called spyware.

Spyware programs do not spread like viruses; instead they are generally installed by exploiting security holes.

They can also be hidden and packaged together with unrelated user-installed software.

Ransomware

Ransomware prevents a user from accessing their files until a ransom is paid.

Rootkits

Once malicious software is installed on a system, it is essential that it stays concealed, to avoid detection.

Software packages known as rootkits allow this concealment, by modifying the host's operating system so that the malware is hidden from the user.

Rootkits can prevent a harmful process from being visible in the system's list of processes, or keep its files from being read.

Vocabulary

Bug - An error only exist in source code

- When noticed easily correctable

Flaw - An error in the understanding or particularly in the design

- difficult and costly to correct

https://cs116.org

Vulnerability

A vulnerability is

- a weakness,
- flaw
- or software bug

A vulnerability can be in

- an application
- a complete computer,
- an operating system,
- or a computer network that is exploited by malware to bypass defences or gain privileges it requires to run.

Common Vulnerabilities and Exposures

CVE created in 1999

Does NOT provide proof of concept (PoC) or exploit!

Common Weakness Enumeration

Common Weakness Enumeration

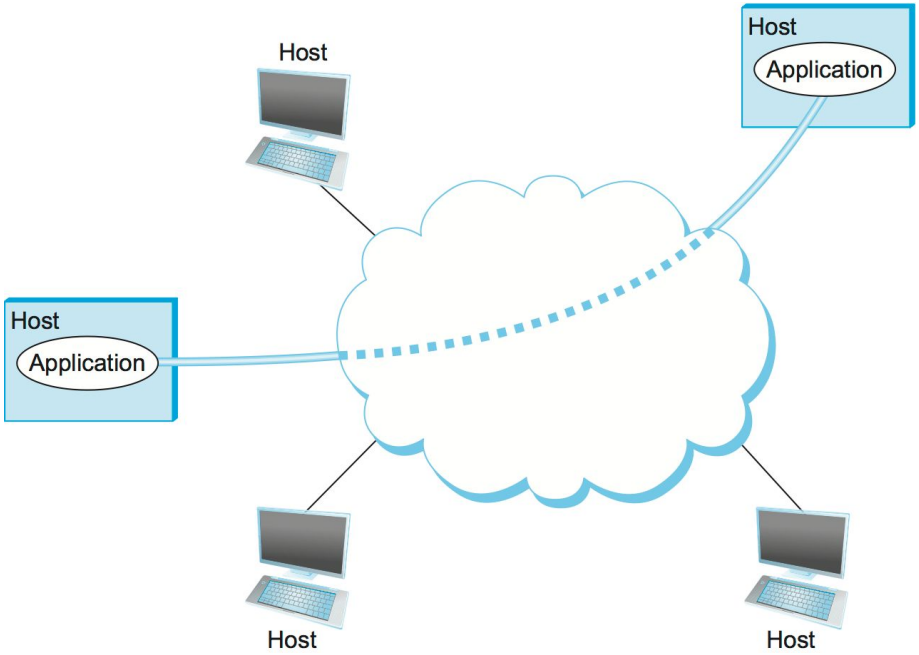
Some vulnerabilities and proof of concepts

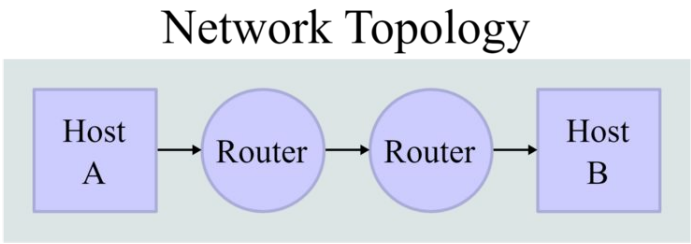
- https://www.metasploit.com/
- https://www.exploit-db.com/
- https://cxsecurity.com/
- https://packetstorm.news/
- Vx Underground
- https://www.seebug.org/

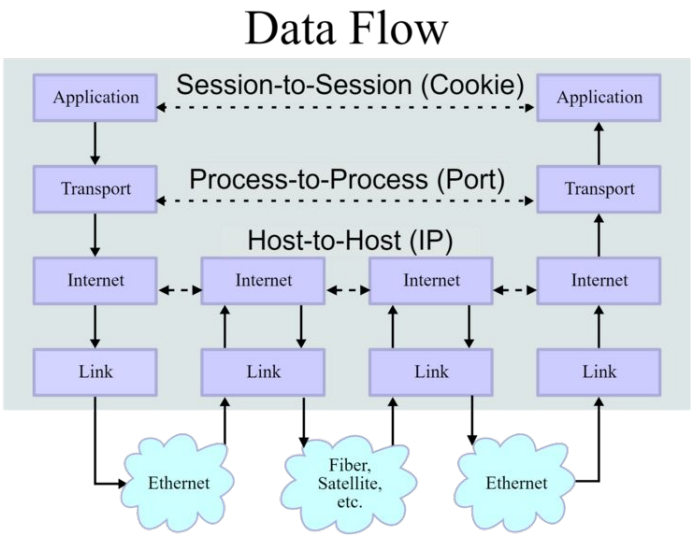
To search vulnerabilities in network and Iot devices

- ZoomEye
- Shodan
- CVE Search Engine - Security Vulnerabilities and Exploits Search Tool

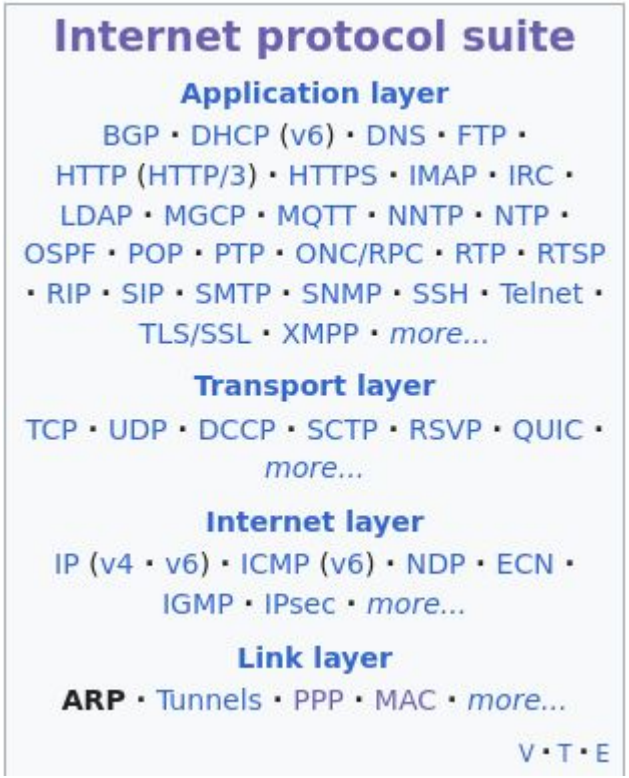
Reminder: Computer networks







Packet and protocols (encapsulation)



A packet contains implementation of used protocols in every layer

- Source&destination IP, port numbers, MAC address, TCP, HTTPS etc.

Address Resolution Protocol (ARP)

- IPv4 address
- + MAC address (medium access control address)
    - A unique identifier assigned to network interface card (NIC) for communications at the data link layer of a network segment

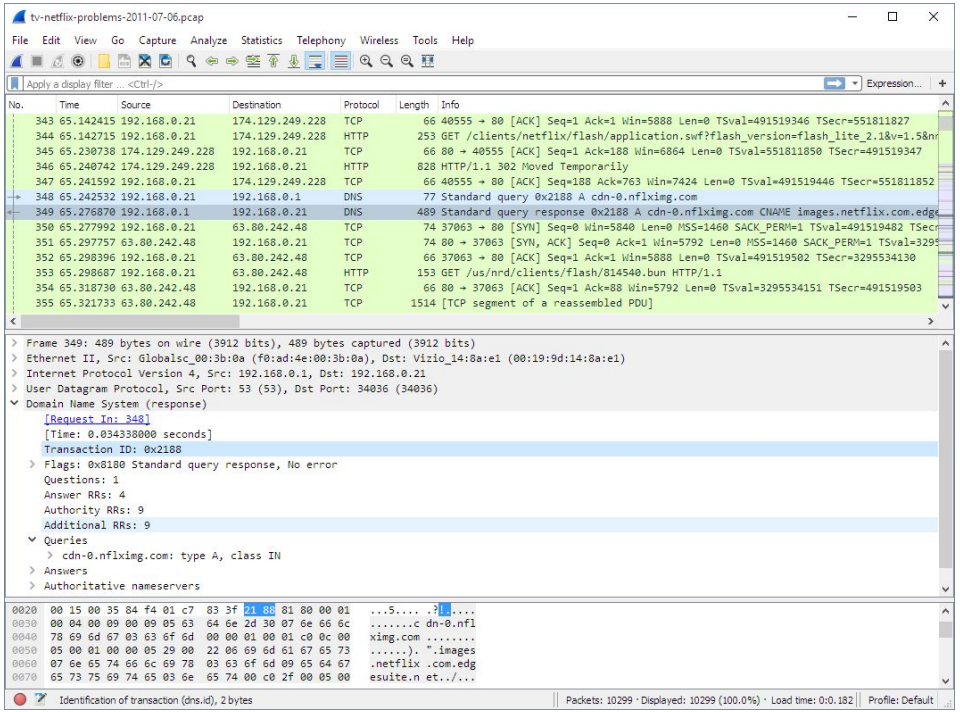
[Address Resolution Protocol - Wikipedia]

Tool wireshark

A popular network protocol analyzer

https://www.wireshark.org/

Wireshark captures packets and lets you examine their contents.

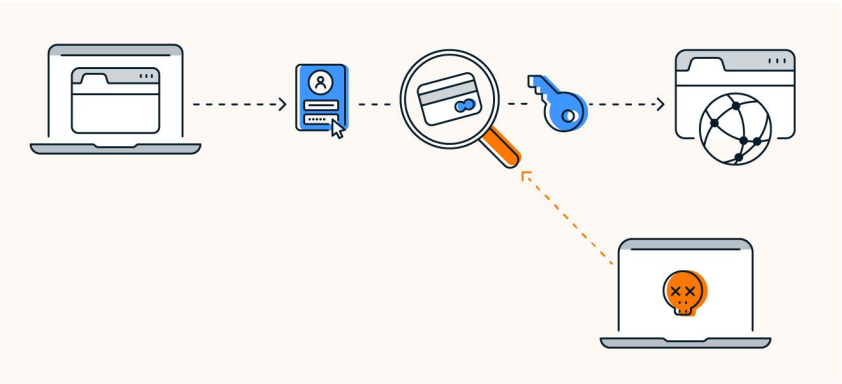


https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroPurposes

Attacking networks: network sniffing

Look at network traffic

Most of the traffic on a network is still unencrypted, plaintext ("in the clear")



Things you can do with network sniffing:

- Troubleshoot networking issues
- Record communications (e.g., email, voice, chat)
- Catch usernames and passwords, personal information, and other sensitive information

Image source: https://www.avast.com/c-packet-sniffing
https://cs116.org/slides/03_attacking_networks.pdf

Network sniffing tools

Ettercap and :: bettercap

Is not intended for network traffic analysis but has capabilities for:

- Capturing passwords
- Conducting man-in-the-middle (eavesdropping) attacks
- Hijacking sessions

dsniff | Kali Linux Tools

Suite of networking sniffing tools including

- dsniff - password sniffer
- webspy - intercepts URLs entered
- mailsnarf - intercepts POP or SMTP-based mail

https://cs116.org/slides/03_attacking_networks.pdf

How to prevent network sniffing

Use encryption and encrypted network protocols

E.g.

- Use HTTPS instead of HTTP
- Use SSH instead of RSH or Telnet
- Use SCP instead of FTP
- Use IMAP or POP3 over SSL
- Use a Virtual Private Network (VPN)

https://cs116.org/slides/06_web_security.pdf

Attacking networks: Network Scanning

Why? Network reconnaissance. Warfare 101

- What devices and computers are up?
- What ports are open on a computer?
- What services are running?
- Determine possible vulnerabilities?
- Still extremely relevant today
- Think poking holes, "ask questions”
- Poking holes: finding interesting and unwanted stuff on networks

Zoomeye and shodan can be used for this purpose

Fping: fping man-page ,

Nmap: Chapter 15. Nmap Reference Guide | Nmap Network Scanning

Netcat: nc(1): arbitrary TCP/UDP connections/listens - Linux man page

Shodan: Shodan

Zoomeye: ZoomEye

https://cs116.org/slides/06_web_security.pdf

Attacking networks: Distributed Denial of Service (DDoS) Attacks

The idea: to make a resource unavailable (the “A” in the CIA Triad) using many remote computing devices.

That is, overwhelm or flood a target (e.g., with so much network traffic)

• Imagine if an important service you use like Gmail, Google, Netflix, Twitter, GitHub is down

- Terabit scale Distributed Denial of Service (DDoS) attacks from September 2016 to late 2016
- How: using thousands of infected devices, mostly cameras. Devices infected via weak username:password hardcoded on device (e.g.,root:root, admin:admin)
- Results: took down Brian Kreb’s blog in September 2016; GitHub, Twitter, Netflix, and many major services were affected via DDoS on Dyn DNS in October 2016 (i.e., “all eggs in one basket”)
- Source code of Mirai botnet: GitHub - jgamblin/Mirai-Source-Code
- Rob Graham’s presentation on “Mirai and IoT Botnet Analysis” at RSA Conference 2017:
- https://vinceinthebay.files.wordpress.com/2017/02/rsac-slides-hta-w10-mirai-1.pdf

https://cs116.org/slides/06_web_security.pdf

DDOS flood Types

SYN Flood

- exhaust states in the TCP/IP stack

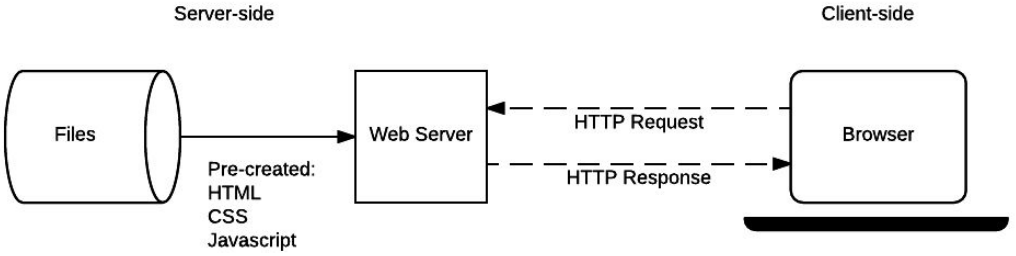
Ping of death

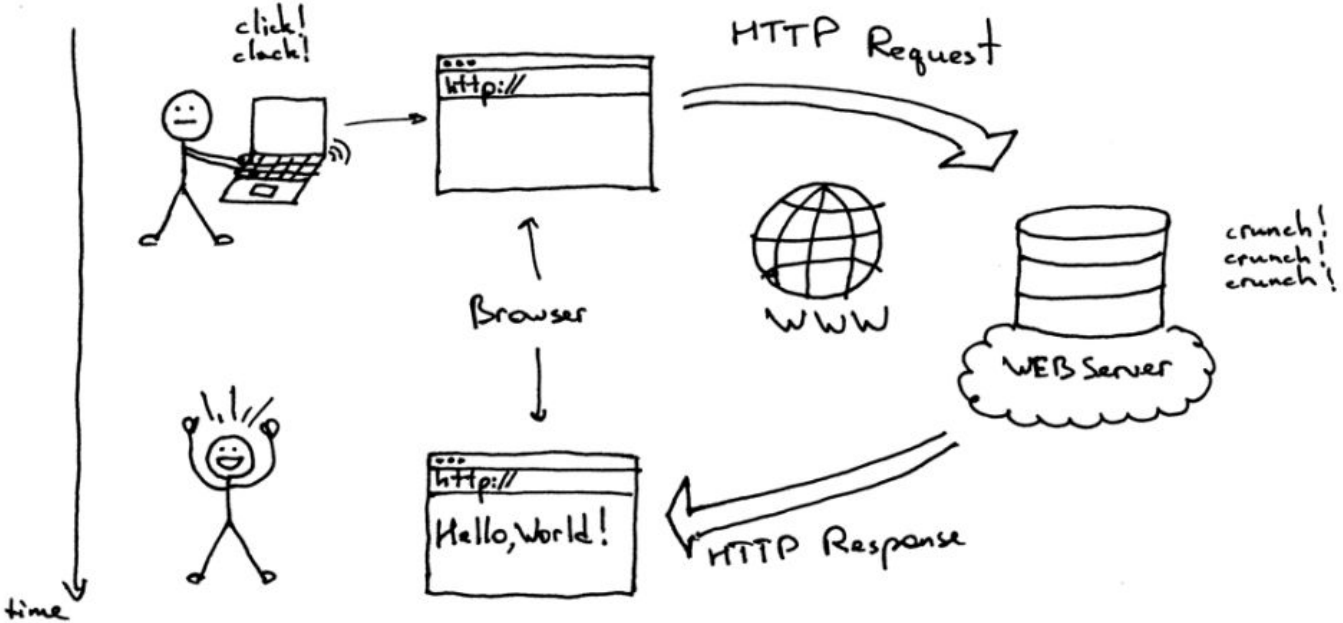
- violate the IP contract, send large packets to cause unexpected errors

And many others

https://cs116.org/slides/06_web_security.pdf

Web security and attacks





https://cs116.org/slides/06_web_security.pdf

Remember URL

Example: https://www.google.com/search?**q=grand+theft+auto**&**lr=lang_zh-TW**

(returns Google results on "Grand Theft Auto" in Chinese Traditional language)

- q => Google’s key in query string for “query”
- lr => Google’s key in query string for “language”

Notice example URL uses https. That is HTTP + Transport Layer Security (TLS)

https://cs116.org/slides/06_web_security.pdf

HTTP Request

header

- GET - Download data from server.
  - This is always the HTTP command used when you type in a URL into address bar
- POST - Sent to server from a form
- PUT - Upload
- DELETE

body

data to be sent to server including **query string key-value pairs**

Additional HTTP commands: HTTP - Wikipedia

https://cs116.org/slides/06_web_security.pdf

HTTP response

header

- 200 - OK
- 301 - Moved Permanently
- 302 - Found (the request was redirected to another URL/URI)
- 401 - Unauthorized
- 403 - Forbidden
- 404 - Not Found
- 500 - Internal Server Error

body

the content data (e.g., HTML, text, JSON, etc.)

https://cs116.org/slides/06_web_security.pdf

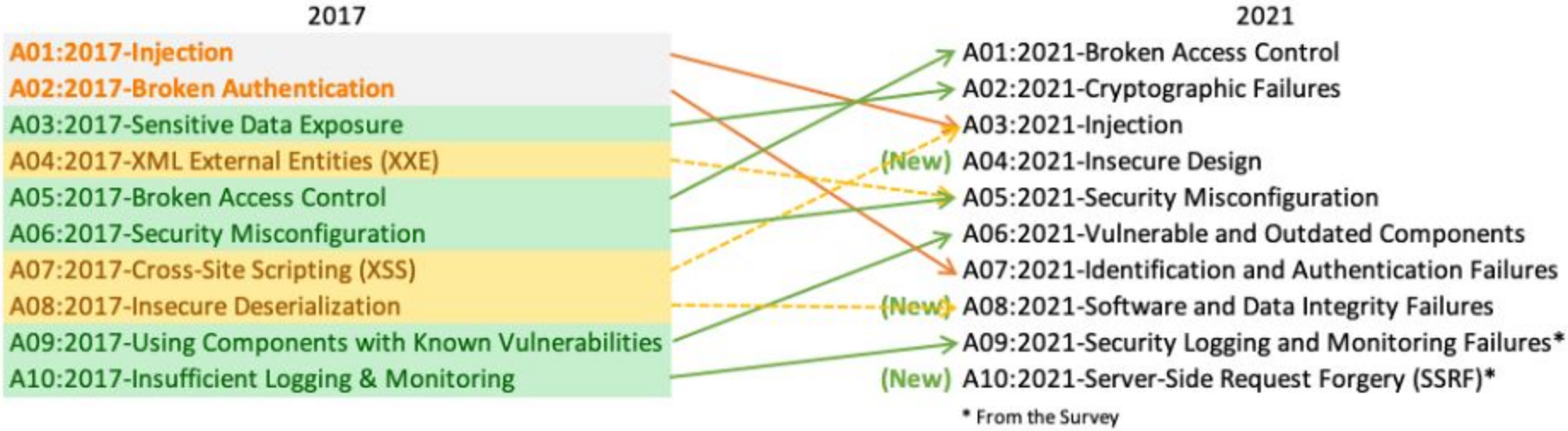
Web security

The Open Worldwide Application Security Project (OWASP) OWASP Foundation is a nonprofit foundation that works to improve the security of software. Our programming includes:

- Community-led open source projects including code, documentation, and standards
- Over 250+ local chapters worldwide
- Tens of thousands of members
- Industry-leading educational and training conferences

https://cs116.org/slides/06_web_security.pdf

OWASP Top 10 Web Application Security Risks



[OWASP Top Ten]

2024 CWE Top 25 Most Dangerous Software Weaknesses

2024 CWE Top 25

Rank	ID	Name	Score	CVEs in KEV	Rank Change vs. 2023
1	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	56.92	3	+1
2	CWE-787	Out-of-bounds Write	45.20	18	-1
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	35.88	4	0
4	CWE-352	Cross-Site Request Forgery (CSRF)	19.57	0	+5
5	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	12.74	4	+3
6	CWE-125	Out-of-bounds Read	11.42	3	+1
7	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	11.30	5	-2
8	CWE-416	Use After Free	10.19	5	-4
9	CWE-862	Missing Authorization	10.11	0	+2
10	CWE-434	Unrestricted Upload of File with Dangerous Type	10.03	0	0
11	CWE-94	Improper Control of Generation of Code ('Code Injection')	7.13	7	+12
12	CWE-20	Improper Input Validation	6.78	1	-6
13	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	6.74	4	+3
14	CWE-287	Improper Authentication	5.94	4	-1
15	CWE-269	Improper Privilege Management	5.22	0	+7
16	CWE-502	Deserialization of Untrusted Data	5.07	5	-1
17	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor	5.07	0	+13
18	CWE-863	Incorrect Authorization	4.05	2	+6
19	CWE-918	Server-Side Request Forgery (SSRF)	4.05	2	0
20	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	3.69	2	-3
21	CWE-476	NULL Pointer Dereference	3.58	0	-9
22	CWE-798	Use of Hard-coded Credentials	3.46	2	-4
23	CWE-190	Integer Overflow or Wraparound			-9
24	CWE-400	Uncontrolled Resource Consumption			+13
25	CWE-306	Missing Authentication for Critical Function			-5

Notice the similarities with the OWASP Top 10 list

https://cs116.org/slides/06_web_security.pdf

Is There a Legal Way or Place to Practice Attacking Web Applications?

IMPORTANT: NEVER DEPLOY THESE WEB APPLICATIONS TO THE PUBLIC INTERNET OR ON A PRODUCTION SYSTEM!

- Damn Vulnerable Web Application (DVWA) - http://www.dvwa.co.uk/
- Mutillidae - https://sourceforge.net/projects/mutillidae/
- Hacme Casino - https://www.mcafee.com/us/downloads/freetools/hacme-casino.aspx (old; Ruby on Rails based)
- WebGoat - https://github.com/WebGoat/WebGoat/wiki ; by OWASP
- A plethora deliberately vulnerable web applications to install and practice on

https://cs116.org/slides/06_web_security.pdf

Metasploitable 2

An **intentionally vulnerable Linux virtual machine (VM)**

https://sourceforge.net/projects/metasploitable/

- **Uses VMware by default; can run on VirtualBox**
- **Contains Damn Vulnerable Web Application, Mutillidae, phpMyAdmin, etc.**
- **Great practice environment**
- **References:**
- **https://community.rapid7.com/docs/DOC-1875**
- **https://www.offensive-security.com/metasploit-unleashed/requirements/**

https://cs116.org/slides/06_web_security.pdf

Some other tools

A web proxy will be an important tool for testing and breaking web applications

Many web proxie software available:

- Burp Suite
- OWASP Zed Attack Proxy (ZAP)
- Tamper Data for Firefox
- mitmproxy

https://cs116.org/slides/06_web_security.pdf

The vulnerabilities

The Principle of Least Privilege –Or Lack thereof

- **The issue1:** connecting to a database or system as root or as administrator -- which has the power to do anything

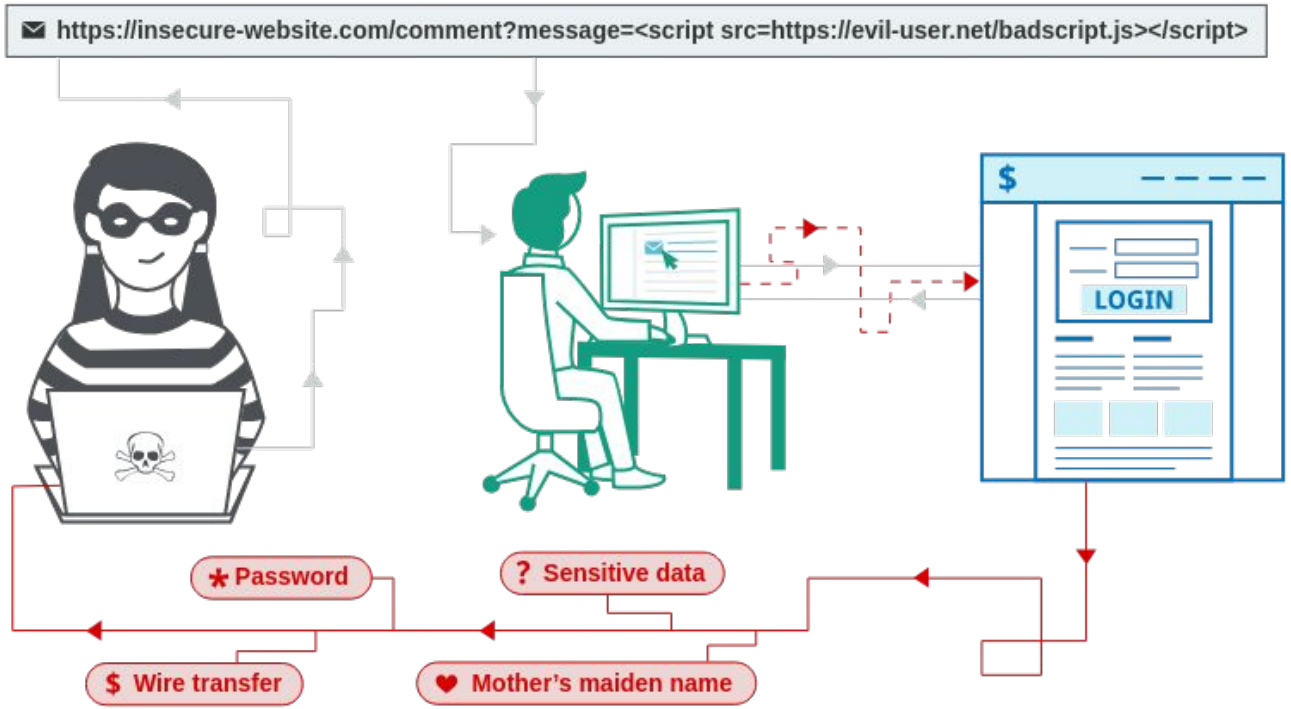
Hard-Coded Credentials

- **The issue2:** username and password or key are hard-coded in source code
- God forbid if you push source code with the credentials to GitHub

```
<?php
 $conn =mysql_connect( "127.0.0.1",
                                    "root",
                                    "pass");
?>
```

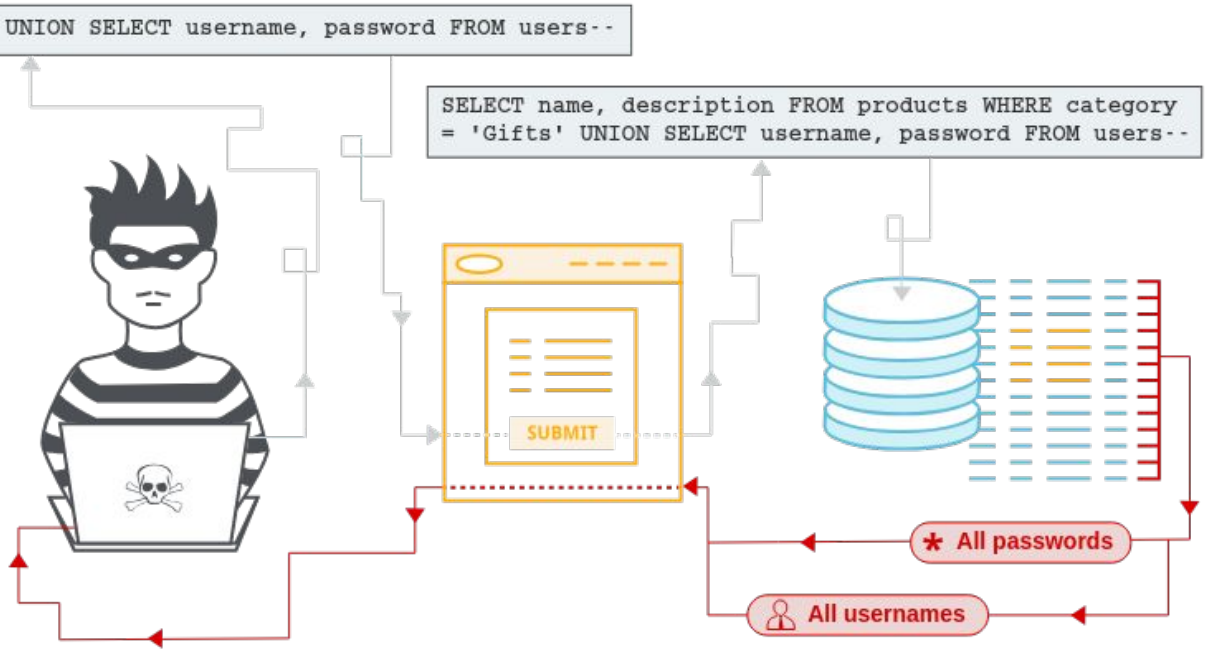
https://cs116.org/slides/06_web_security.pdf

cross-site scripting (XSS)



https://portswigger.net/web-security/cross-site-scripting, attacker's script executes in the user's browser

SQL Injection



https://portswigger.net/web-security/sql-injection

End of semester