

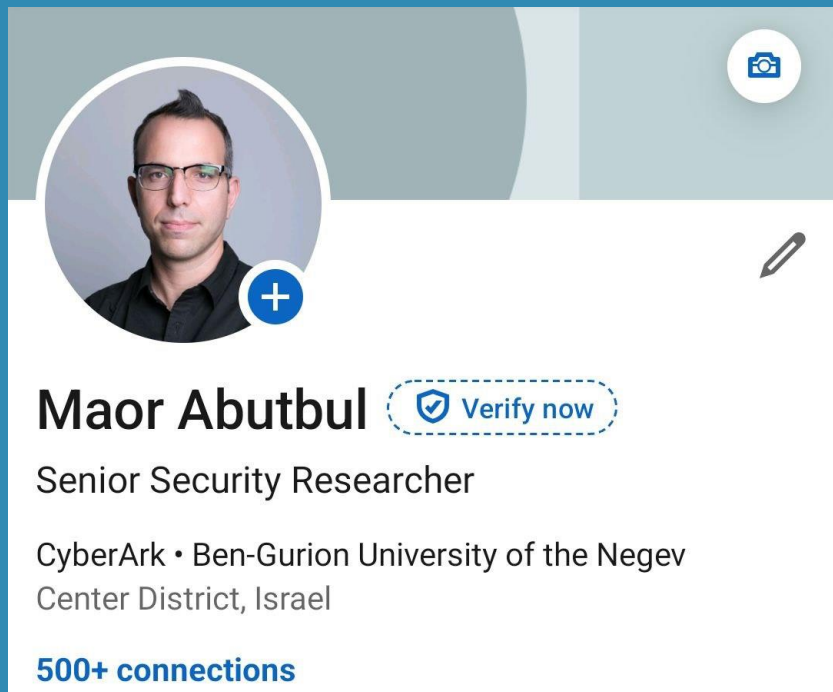
Web Technologies & Web Security

מאור אבוטבול
מגשימים 2024

25.06.2024



#whoami_____



<https://il.linkedin.com/in/maor-abutbul>

מי אני

• 39, יבנה

• רקע מקצועי

• לימודים

• היום - חוקר חולשות (אבטחה) - סייברארק

Agenda

- רקע תיאורטי
 - טכנולוגיות ווב (בזריזות)
 - אבטחת מערכות תקשורת - מבט מלמעלה
 - פרוטוקול HTTP
- כלי פיתוח ומחקר
- אבטחת מערכות ווב
 - חולשה נפוצה + מעבדה עצמית
- מקורות תרגול והעשרה

טכנולוגיות ווב (בפיתוח) – Web Technologies

Web Technologies - HTML CSS JavaScript

HTML - HyperText Markup Language

- HTML is the standard markup language for creating Web pages.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

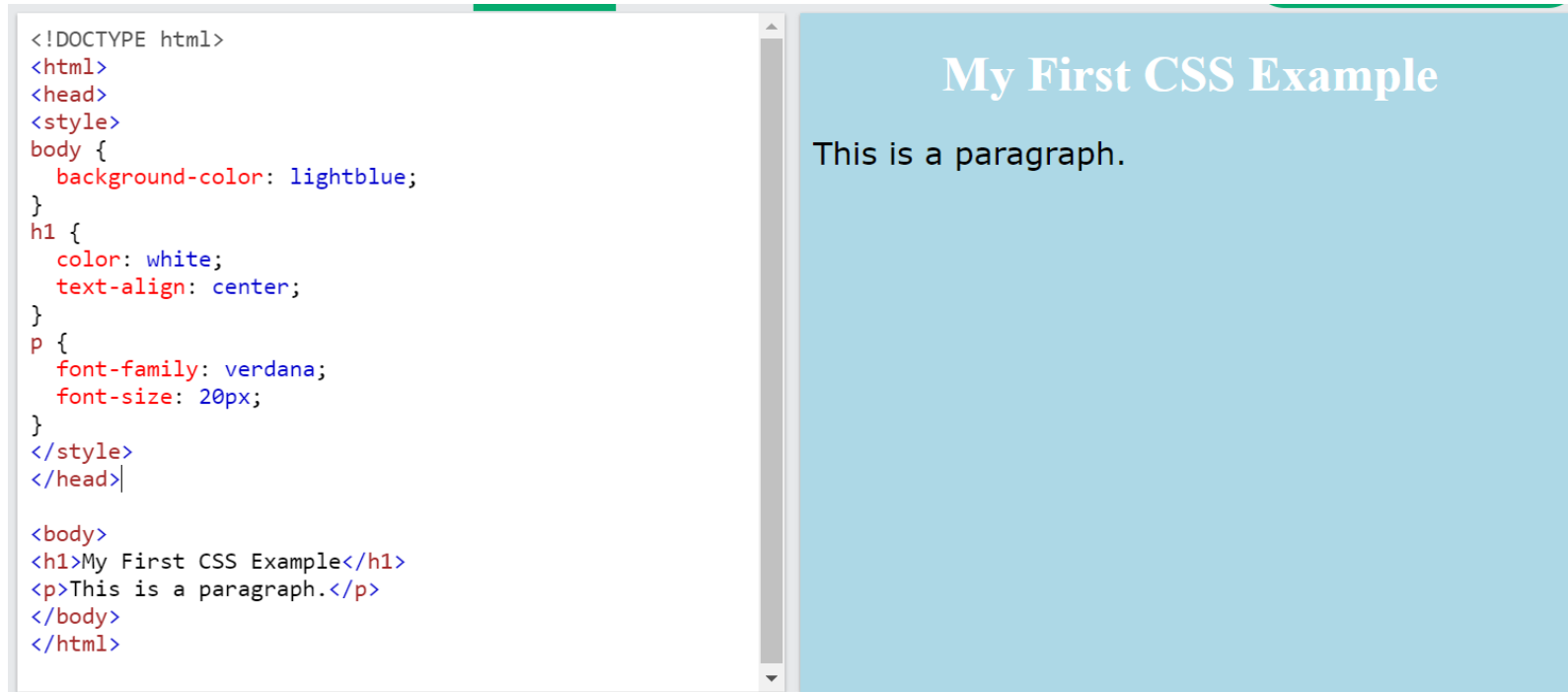
This is a Heading

This is a paragraph.

- <https://www.w3schools.com/html/default.asp>

CSS - Cascading Style Sheets

- CSS is the language we use to style a Web page.



- <https://www.w3schools.com/css/default.asp>

JavaScript

- The world's most popular programming language.
- JavaScript is the programming language of the Web.

<pre><!DOCTYPE html> <html> <body> <h2>My First JavaScript</h2> <button type="button" onclick="document.getElementById('demo').innerHTML = Date()"> Click me to display Date and Time.</button> <p id="demo"></p> </body> </html></pre>	<h2>My First JavaScript</h2> <p>Click me to display Date and Time.</p> <p>Sun Jun 27 2021 12:17:51 GMT+0300 (Israel Daylight Time)</p>
---	--

הרצת קוד בצד לקוח (דפדפן)

<https://www.w3schools.com/js/default.asp>

טכנולוגיות צד שרת

- Front-end development refers to the client-side (how a web page **looks**).
- Back-end development refers to the server-side (how a web page **works**).
- Server Side
 - PHP
 - ASP
 - Node.js
- עיבוד הבקשה נעשה בצד השרת
- <https://www.w3schools.com/>

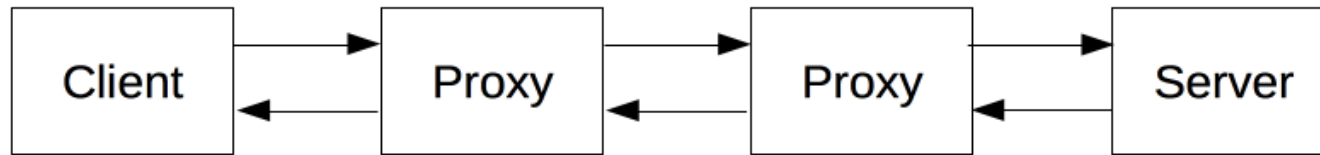
אבטחת מערכות תקשורת - מבט מלמעלה

Network Security in 2 minutes

רכיבי תקשורת ואבטחה -עבור פרוטוקול HTTP

- רכיבי תקשורת (ואבטחה) ללא התערבות ב-HTTP

- Firewall
- נתב (Router)
- Network Load Balancer
- ועוד



- רכיבים המנתחים תעבורת HTTP

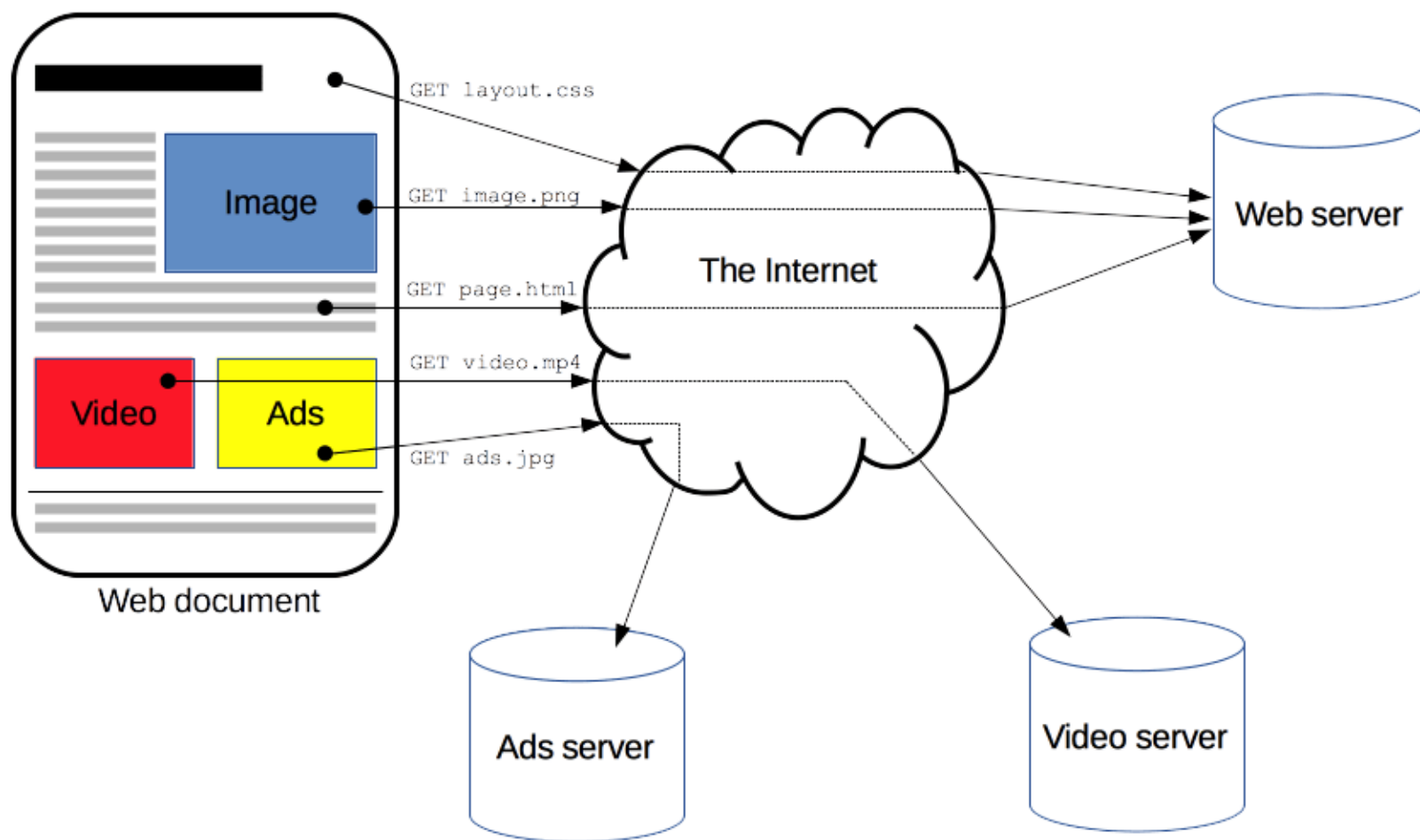
- רכיב פרוקסי (Proxy)
- Application Load Balancer
- Web Cache
- WAF – Web Application Firewall
- Web Server

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/>

פרוטוקול - HTTP

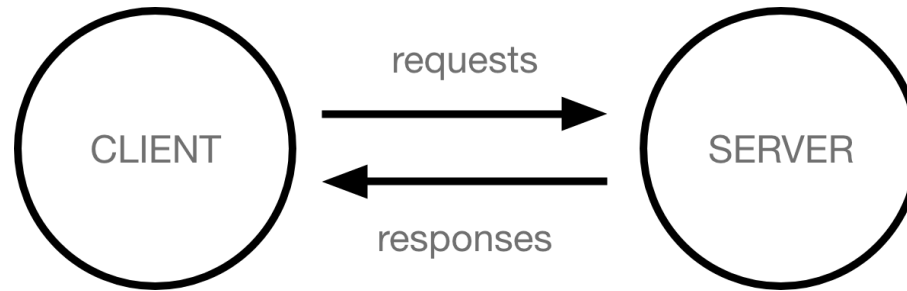
Hypertext Transfer Protocol (HTTP)

פרוטוקול HTTP



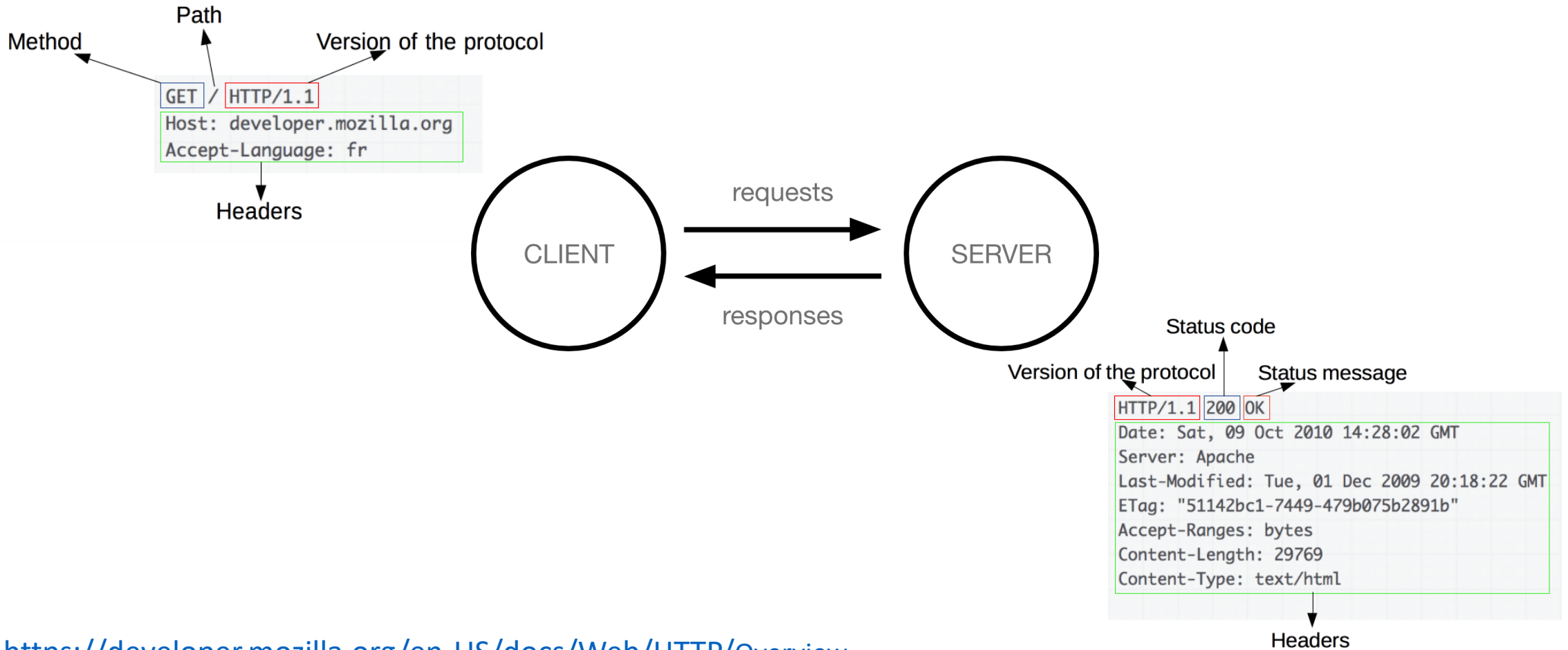
פרוטוקול - HTTP מודל שרת לקוח והודעות

- מודל שרת-לקוח
- Stateless, but not Session-less
- הודעות - בקשה תשובה \ תגובה



- <https://developer.mozilla.org/en-US/docs/Web/HTTP/>

פרוטוקול - HTTP מודל שרת לקוח והודעות



<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

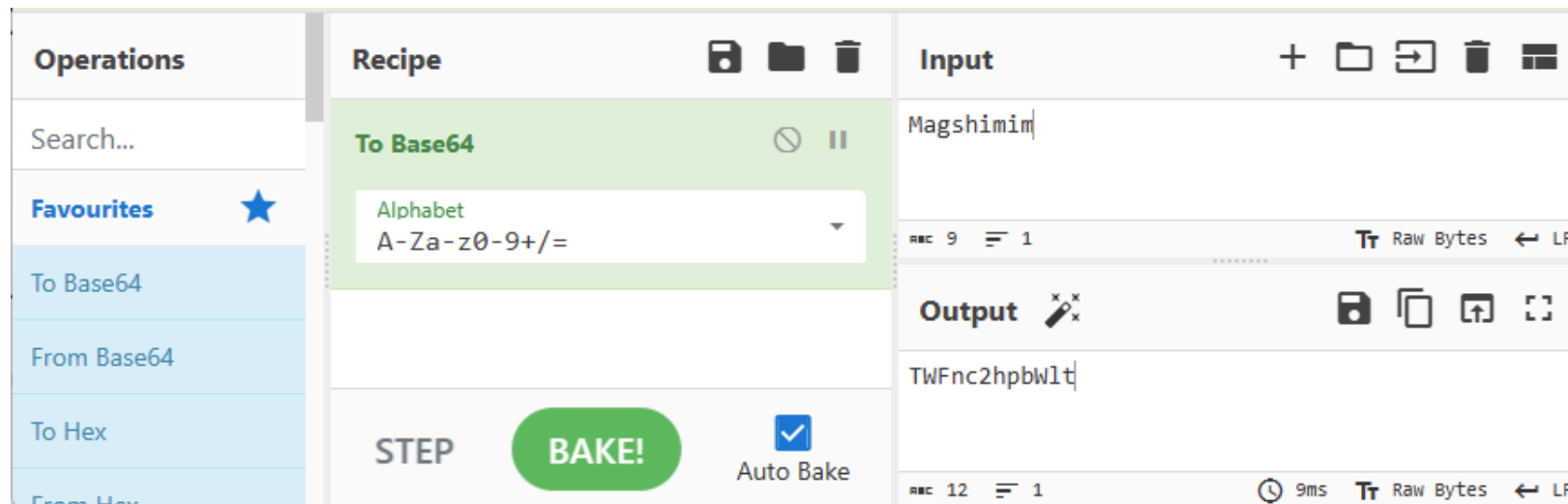
Tools Demos

כלי פיתוח ומחקר

קידוד - Base64 Encoding + CyberChef

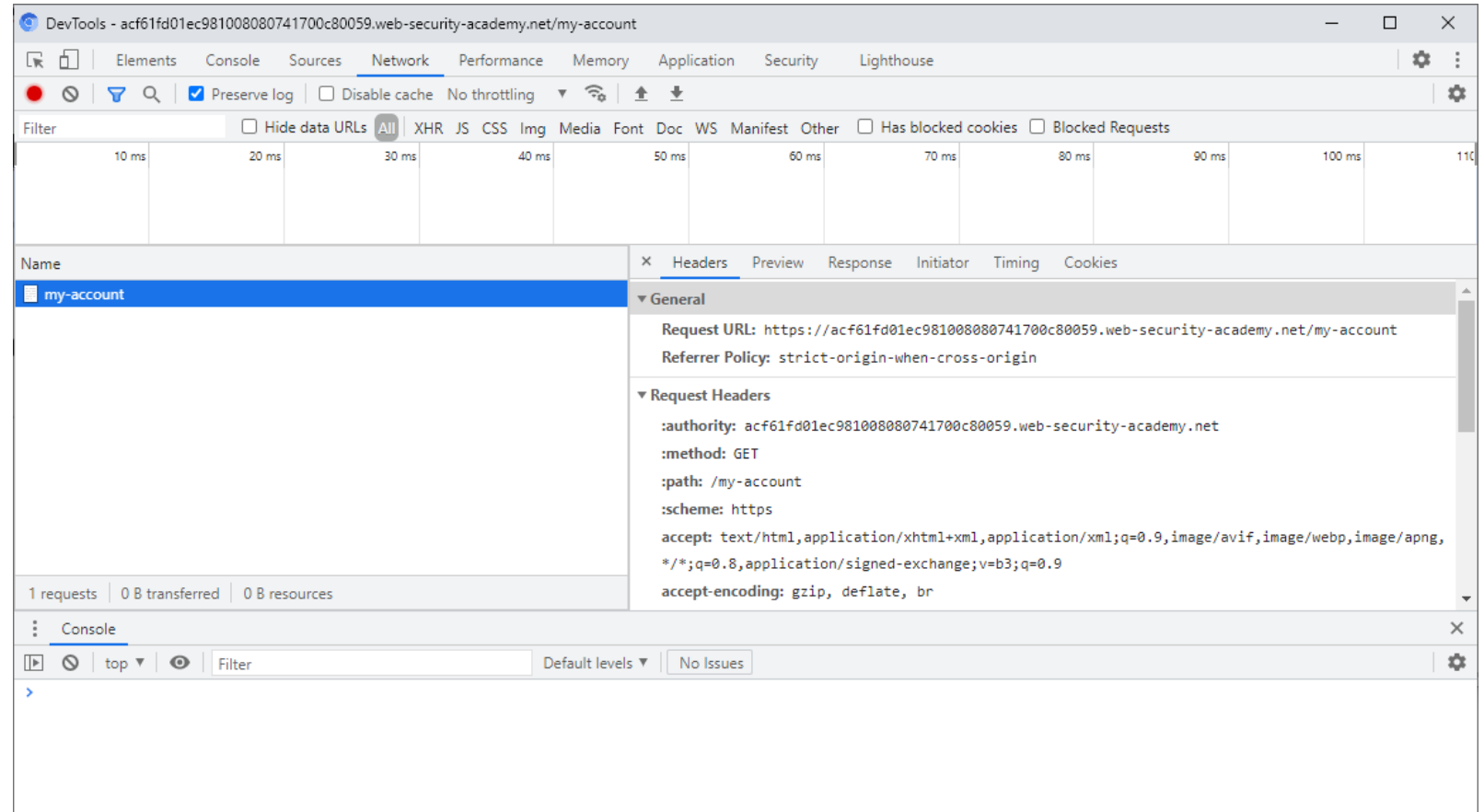
- קידוד - Base64 מיועד להעברת מידע בינארי על גבי ערוץ שתומך בטקסט בלבד (תווי אסקי)
- משמש לשליחת צרופות (קבצים) בשליחת מיילים (SMTP)

- <https://en.wikipedia.org/wiki/Base64>
- [https://gchq.github.io/CyberChef/#recipe=To_Base64\('A-Za-z0-9%2B/%3D'\)&input=TWFnY2hpbWlt](https://gchq.github.io/CyberChef/#recipe=To_Base64('A-Za-z0-9%2B/%3D')&input=TWFnY2hpbWlt)



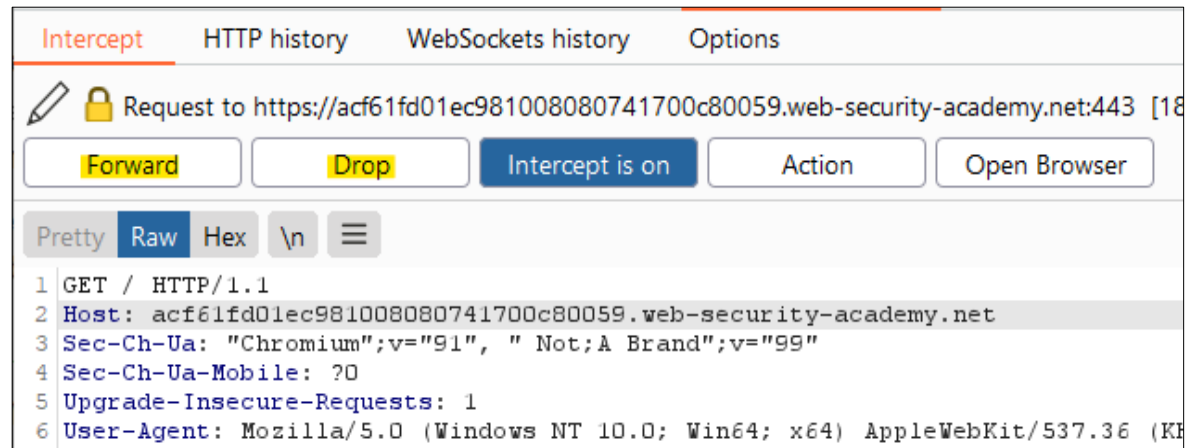
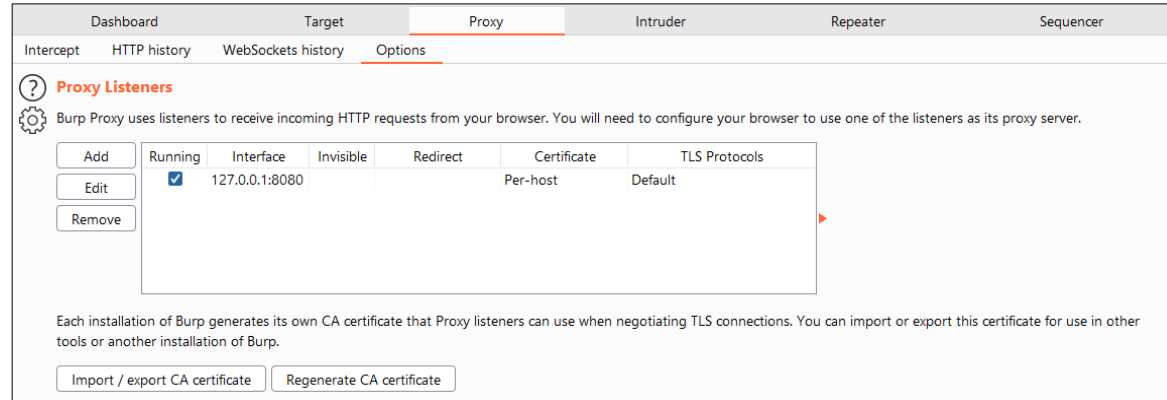
Browser DevTools + Demo

- Elements
- Console
 - Page interaction
- Application
 - Cookies
- Sources
 - Debug
- Network
 - Copy As
 - (fetch + console)



Tools - Burp Suite (Community)

- Host Proxy
- Dashboard
- Target
- Proxy
 - Intercept
 - HTTP History
 - Options



- <https://portswigger.net/burp/communitydownload>

Burp Suite Cont'

- Repeater (Ctrl + R)

• עריכה ושליחה ידנית

- Intruder (Ctrl + I)

• הגדרת "משתנים" וערכי Payload

• לשליחה אוטומטית

- Decoder

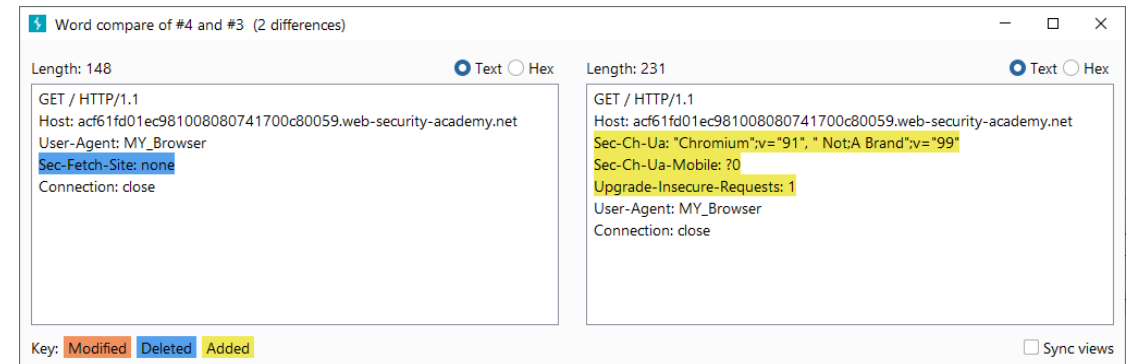
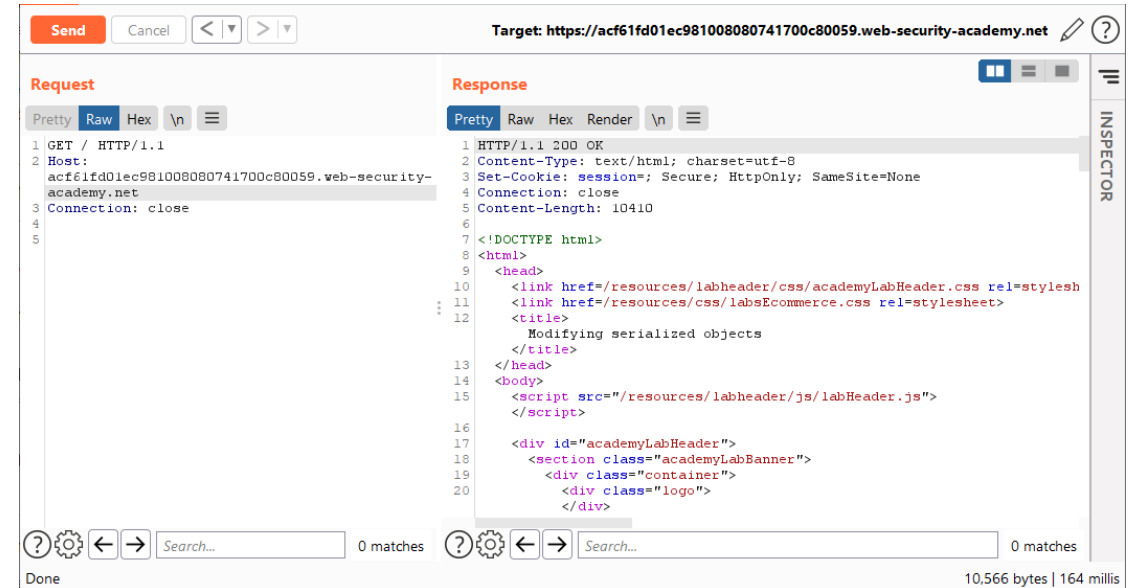
• פענוח קידודים

- Comparer

• השוואה בין בקשות

- Extender

• הורדת וניהול "הרחבות"



Tools - כלים

- Cyber Chef
 - <https://gchq.github.io/CyberChef/>
- Browser Developer Tools (Dev-Tools)
- (Host) Proxy Tools
 - Burp Suite - <https://portswigger.net/burp/communitydownload>
 - OWASP ZAP Proxy - <https://www.zaproxy.org/>
 - Fiddler - <https://www.telerik.com/fiddler/fiddler-classic>

Web Security



רגע לפני - אחריות – חוק המחשבים

- יש חוק בישראל
- חוק המחשבים, התשנ"ה–1995
- [חוק המחשבים - קישור](#)
- הסבר מתוך קורס של המרכז לחינוך סייבר
- [חוק המחשבים.pdf](#)
- ובכל מקרה
 - חובה להשיג אישור (בכתב!) של בעל הנכס.
 - לשים לב שלא חורגים מהנכסים שהוגדרו לנו.
 - לדוגמא:
 - שרתים משותפים.
 - שירותי אחסון.

Web Security

מקורות - Resources



Web Security – Resources - מקורות

- OWASP - Open Web Application Security Project
 - <https://owasp.org/>
 - OWASP Top Ten
 - <https://owasp.org/www-project-top-ten/>
 - OWASP Web Security Testing Guide
 - <https://github.com/OWASP/wstg/releases>
- Mozilla - Web security
 - <https://developer.mozilla.org/en-US/docs/Web/Security>
 - https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy
- Portswigger - Web Security Academy
 - <https://portswigger.net/web-security/learning-path>

Web Security – Resources - Portswigger - מקורות

Client-side topics

Client-side vulnerabilities introduce an additional layer of complexity, which can make them slightly more challenging. These materials and labs will help you build on the server-side skills you've already learned and teach you how to identify and exploit some gnarly client-side vectors as well.

Cross-site scripting (XSS)

Simply put, XSS is one of the most important vulnerabilities out there. It's both incredibly common and extremely powerful, especially when used as part of a wider exploit chain. This is a huge topic, with plenty of labs for complete beginners and seasoned pros alike.

[Go to topic →](#) 30 Labs

Cross-site request forgery (CSRF)

[Go to topic →](#) 12 Labs

Cross-origin resource sharing (CORS)

[Go to topic →](#) 4 Labs

Clickjacking

[Go to topic →](#) 5 Labs

DOM-based vulnerabilities

[Go to topic →](#) 7 Labs

WebSockets

[Go to topic →](#) 3 Labs

Server-side topics

For complete beginners, we recommend starting with our server-side topics. These vulnerabilities are typically easier to learn because you only need to understand what's happening on the server. Our materials and labs will help you develop some of the core knowledge and skills that you will rely on time after time.

SQL injection

SQL injection is an old-but-gold vulnerability responsible for many high-profile data breaches. Although relatively simple to learn, it can potentially be used for some high-severity exploits. This makes it an ideal first topic for beginners, and essential knowledge even for more experienced users.

[Go to topic →](#) 18 Labs

Authentication

[Go to topic →](#) 14 Labs

Path traversal

[Go to topic →](#) 6 Labs

Command injection

[Go to topic →](#) 5 Labs

Business logic vulnerabilities

[Go to topic →](#) 11 Labs

Information disclosure

[Go to topic →](#) 5 Labs

Access control

[Go to topic →](#) 13 Labs

File upload vulnerabilities

[Go to topic →](#) 7 Labs

Race conditions

[Go to topic →](#) 6 Labs

Server-side request forgery (SSRF)

[Go to topic →](#) 7 Labs

XXE injection

[Go to topic →](#) 9 Labs

NoSQL injection

[Go to topic →](#) 4 Labs

API testing

[Go to topic →](#) 5 Labs

- Portswigger - Web Security Academy
 - <https://portswigger.net/web-security/all-topics>

Web Security – Resources - Portswigger - מקורות

Advanced topics

These topics aren't necessarily more difficult to master but they generally require deeper understanding and a wider breadth of knowledge. We recommend getting to grips with the basics before tackling these labs, some of which are based on pioneering techniques discovered by our world-class research team.

Insecure deserialization

Deserialization has a reputation for being difficult to get your head around but it can be much easier to exploit than you might think. We'll guide you through the process step-by-step so you can pick off some high-severity bugs that even experienced testers may have missed altogether.

[Go to topic →](#) 10 Labs

Web LLM attacks

New topic

[Go to topic →](#) 4 Labs

GraphQL API vulnerabilities

[Go to topic →](#) 5 Labs

Server-side template injection

[Go to topic →](#) 7 Labs

Web cache poisoning

[Go to topic →](#) 13 Labs

HTTP Host header attacks

[Go to topic →](#) 7 Labs

HTTP request smuggling

[Go to topic →](#) 22 Labs

OAuth authentication

[Go to topic →](#) 6 Labs

JWT attacks

[Go to topic →](#) 8 Labs

Prototype pollution

[Go to topic →](#) 10 Labs

Essential skills

[Go to topic →](#) 2 Labs

Insecure Deserialization

חולשה נפוצה - פתיחה לא מאובטחת של רצף סדרתי

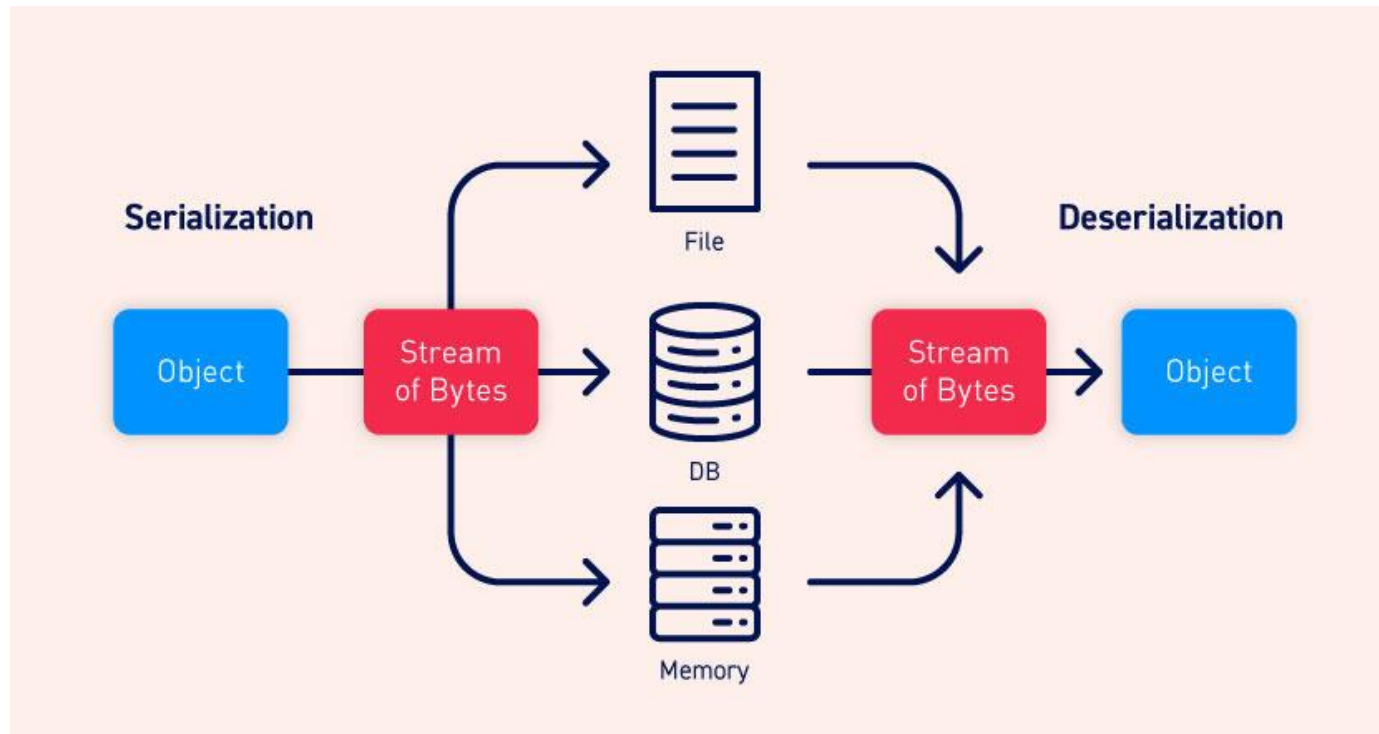
פתיחה לא מאובטחת של רצף סדרתי - Insecure Deserialization

- OWASP Top 10 Reference:
 - https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization
- Portswigger - Web Security Academy - Insecure deserialization
 - <https://portswigger.net/web-security/deserialization>

Insecure Deserialization

- What is serialization?
- Serialization vs deserialization

- מהי סריאליזציה ?
- דיסריאליזציה



Insecure Deserialization – PHP serialize function

serialize

(PHP 4, PHP 5, PHP 7, PHP 8)

serialize — Generates a storable representation of a value

Description

```
serialize(mixed $value): string
```

Generates a storable representation of a value.

This is useful for storing or passing PHP values around without losing their type and structure.

To make the serialized string into a PHP value again, use [unserialize\(\)](#).

- <https://www.php.net/manual/en/function.serialize.php>

Insecure Deserialization – PHP unserialize function

unserialize

(PHP 4, PHP 5, PHP 7, PHP 8)

unserialize — Creates a PHP value from a stored representation

Description

```
unserialize(string $data, array $options = []): mixed
```

unserialize() takes a single serialized variable and converts it back into a PHP value.

Warning Do not pass untrusted user input to **unserialize()** regardless of the **options** value of `allowed_classes`. Unserialization can result in code being loaded and executed due to object instantiation and autoloading, and a malicious user may be able to exploit this. Use a safe, standard data interchange format such as JSON (via `json_decode()` and `json_encode()`) if you need to pass serialized data to the user.

If you need to unserialize externally-stored serialized data, consider using `hash_hmac()` for data validation. Make sure data is not modified by anyone but you.

- <https://www.php.net/manual/en/function.serialize.php>

Exploiting Insecure Deserialization Vulnerabilities

PHP serialization format

```
$user->name = "carlos";  
$user->isLoggedIn = true;
```

When serialized, this object may look something like this:

```
O:4:"User":2:{s:4:"name":s:6:"carlos"; s:10:"isLoggedIn":b:1;}
```

This can be interpreted as follows:

- O:4:"User" - An object with the 4-character class name "User"
- 2 - the object has 2 attributes
- s:4:"name" - The key of the first attribute is the 4-character string "name"
- s:6:"carlos" - The value of the first attribute is the 6-character string "carlos"
- s:10:"isLoggedIn" - The key of the second attribute is the 10-character string "isLoggedIn"
- b:1 - The value of the second attribute is the boolean value true

<https://portswigger.net/web-security/deserialization/exploiting>

• פורמט הסריאליזציה של PHP

• פורמט מבוסס טקסט

• אותיות מייצגות את סוג המשתנה (Data Type)

• מספרים מייצגים אורך של השדה המקודד

Insecure deserialization

- What is insecure deserialization?

- מהי חולשת "פענוח לא בטוח" ?

- המידע לפיענוח נשלט ע"י המשתמש (או תוקף)

- אף ניתן לשחזר (לייצר) אובייקט מסוג אחר

Exploiting Insecure Deserialization Vulnerabilities

ניצול החולשה

Exploiting Insecure Deserialization Vulnerabilities

- How to identify insecure deserialization

- כיצד נזהה חולשות "פענוח לא בטוח"

- סטטי (סקר קוד)

- במידה ויש לנו את קוד המקור

- נחפש את שמות הפונקציות הרלוונטיות לאותה השפה.

- דינאמי - (מבדק חדירות)

- במידה ומכירים את "פורמט הסריאליזציה" של אותה השפה

- ניתן לזהות ע"י הסתכלות על התוכן שמתקבל מהשרת.

- כאשר נזהה מידע כזה – ננסה לראות האם אנחנו יכולים לשלוט עליו

- <https://portswigger.net/web-security/deserialization/exploiting>

Exploiting Insecure Deserialization - Lab

Lab Time - Insecure Deserialization

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-objects>

Insecure Deserialization

- How do insecure deserialization vulnerabilities arise?
 - כיצד נוצרות החולשות "פענוח לא בטוח" ?
 - בדרך כלל נובע מחוסר הבנה - בכמה מסוכן יכול להיות מידע שנשלט ע"י המשתמש
 - לפעמים חושבים שהקוד בטוח (כשיש בדיקת שגיאות)
 - מניחים שאובייקטים מסורלזים (מקודדים) הם בטוחים
 - במיוחד כאשר פורמט הסריאליזציה הוא בינארי

הגנות - Insecure Deserialization

- How to prevent insecure deserialization vulnerabilities ?
- כיצד נמנע מתקפות "פענוח לא בטוח" ?
- להימנע ככל הניתן מדיסריאליזציה של תוכן (שנשלט ע"י משתמש)
- יישום בדיקות אמינות (Integrity)
- אכיפת מגבלות סוג אובייקט (מחמירות) בתהליך פתיחה של רצף סדרתי
- בידוד והרצת קוד לפתיחת רצף סדרתי בסביבה בעלת הרשאות נמוכות ככל הניתן
- <https://owasp.org/www-pdf-archive/OWASP-Top-10-2017-he.pdf>

Insecure Deserialization – Real World Example

CVE's

Insecure Deserialization – Real World Example

- What is Jackson?
- Open-Source Library
- Jackson has been known as "the Java JSON library" or "the best JSON parser for Java". Or simply as "JSON for Java".
 - Source: <https://github.com/FasterXML/jackson>

Insecure Deserialization – Real World Example

Fasterxml : Security Vulnerabilities														
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9														
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending														
Total number of vulnerabilities : 69 Page : 1 (This Page) 2														
Copy Results Download Results														
#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2021-20190	502			2021-01-19	2021-04-24	8.3	None	Remote	Medium	Not required	Partial	Partial	Complete
A flaw was found in jackson-databind before 2.9.10.7. FasterXML mishandles the interaction between serialization gadgets and typing. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.														
2	CVE-2020-36189	502			2021-01-06	2021-06-14	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
FasterXML jackson-databind 2.x before 2.9.10.8 mishandles the interaction between serialization gadgets and typing, related to com.newrelic.agent.deps.ch.qos.logback.core.db.DriverManagerConnectionSource.														
3	CVE-2020-36188	502			2021-01-06	2021-06-14	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
FasterXML jackson-databind 2.x before 2.9.10.8 mishandles the interaction between serialization gadgets and typing, related to com.newrelic.agent.deps.ch.qos.logback.core.db.JNDIConnectionSource.														
4	CVE-2020-36187	502			2021-01-06	2021-06-14	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
FasterXML jackson-databind 2.x before 2.9.10.8 mishandles the interaction between serialization gadgets and typing, related to org.apache.tomcat.dbcp.dbcp.datasources.SharedPoolDataSource.														
5	CVE-2020-36186	502			2021-01-06	2021-06-14	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
FasterXML jackson-databind 2.x before 2.9.10.8 mishandles the interaction between serialization gadgets and typing, related to org.apache.tomcat.dbcp.dbcp.datasources.PerUserPoolDataSource.														
6	CVE-2020-36185	502			2021-01-06	2021-06-14	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
FasterXML jackson-databind 2.x before 2.9.10.8 mishandles the interaction between serialization gadgets and typing, related to org.apache.tomcat.dbcp.dbcp2.datasources.SharedPoolDataSource.														

- https://www.cvedetails.com/vulnerability-list/vendor_id-15866/Fasterxml.html

Insecure Deserialization – Real World – More CVE's

- <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=deserialization>
- <https://www.cvedetails.com/google-search-results.php?q=deserialization>
- <https://www.cvedetails.com/vulnerability-list/cwe-502/vulnerabilities.html>
- Python PyYAML (cve-2020-14343) Writeup
 - <https://hackmd.io/@harrier/uiuctf20>

Resources & Next Steps

Practice, Socialize, to the big boy's league

Web Security – Practice - תרגול

- **Pico-CTF
 - <https://www.picoctf.org/>
- Overthewire – Bandit + **Natas
 - <https://overthewire.org/wargames/>
- **Portswigger - Web Security Academy - Labs
 - <https://portswigger.net/web-security/all-labs>
 - <https://portswigger.net/web-security/learning-path>
- OWASP – Practice Systems
 - <https://github.com/digininja/DVWA>
 - <https://github.com/bkimminich/juice-shop>
 - <https://github.com/WebGoat/WebGoat>
- HackTheBox (more system related, also web challenges)
 - <https://www.hackthebox.eu/>

אירועים \ מפגשים – Socialize - Events / Meetups

- OWASP - Meetup
 - <https://www.meetup.com/OWASP-Israel/>
- BSidesTLV
 - <https://bsidestlv.com/>
 - עתיד להתקיים בחמישי – רישום בתשלום
 - <https://bsidestlv.com/ctf/>
 - <https://ctf24.bsidestlv.com/>
 - התחיל היום בבוקר פתוח ל-48 שעות
- Defcon
 - <https://defcon.org/index.html>
 - <https://www.youtube.com/user/DEFCONConference>
- BlackHat
 - <https://www.blackhat.com/>
 - <https://www.youtube.com/user/BlackHatOfficialYT>

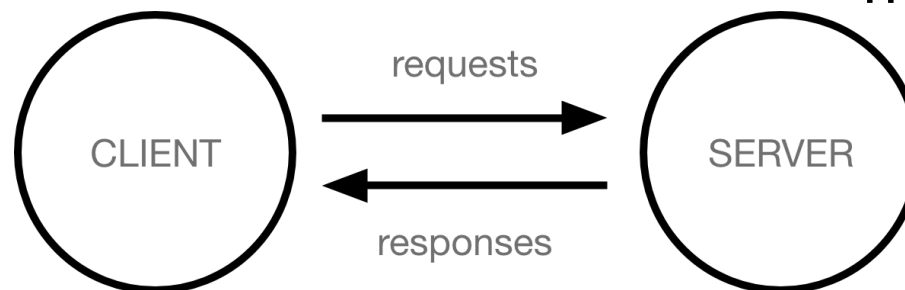
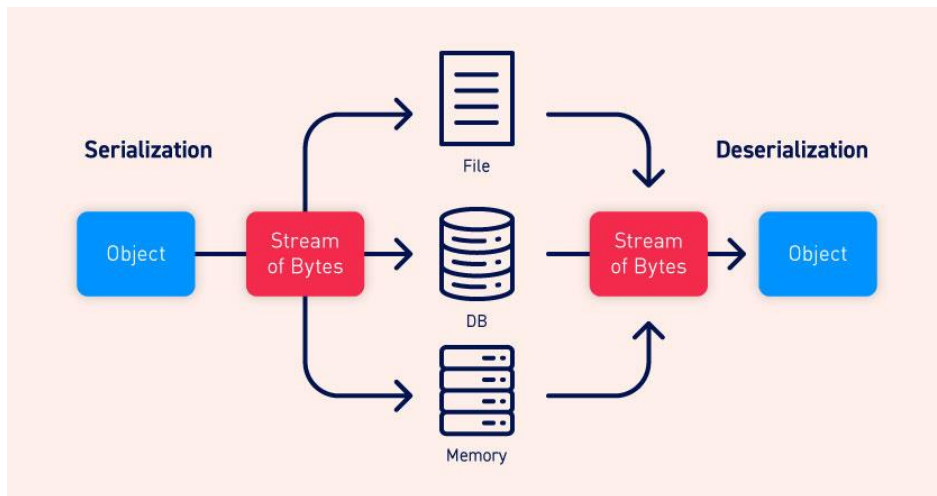
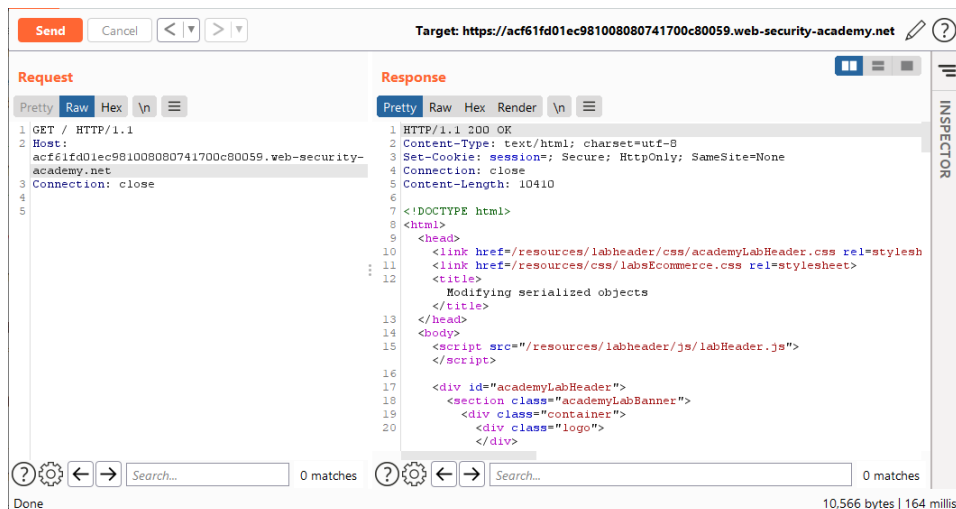
תחרויות ותוכניות חיפוש חולשות – CTF & Bug Bounty

- CTF (My two cents)
 - Try Alone
 - Write down what you tried
 - Then (after the event) Read Writeups !
- PicoCTF
 - https://picoctf.org/get_started.html
- CTF Time
 - <https://ctftime.org/event/list/upcoming>
 - לוח שנה עם תחרויות קרובות
- Bug Bounty - Platforms
 - <https://hackerone.com/bug-bounty-programs>
 - <https://www.bugcrowd.com/bug-bounty-list/>
 - <https://www.intigriti.com/>
- Bug Bounty – Programs for specific companies
 - <https://www.guru99.com/bug-bounty-programs.html>

Summary



סיכום



- רקע תיאורטי
- פרוטוקול – HTTP
- כלים שימושיים
- מנגנון סריאליזציה ודיסריאליזציה
- מימוש המנגנון בשפת PHP
- חולשת "פענוח לא בטוח"
- זיהוי וניצול החולשה
- מניעה והתגוננות
- מקורות להעשרה

Questions ?

Thank You!

Contact me at:

<https://il.linkedin.com/in/maor-abutbul>

Extra Tools

- Notepad++
- Wireshark + Tshark
- CLI Tools (ping, netcat, trace)
- Linux tools (Cat, grep, sort, wc, man, ls, jq, ...)
 - Wget
 - Curl
- OpenSSL
- Putty
- Nmap
- Python
 - Modules (Requests, BS4, Selenium)
- <https://docs.projectdiscovery.io/getstarted-example>
- 7Zip
- VSCode
- Docker
- VirtualBox / VMware
- Git & Github

- רשימה לא מלאה של כלים שכדאי להכיר

טכנולוגיות Web בפיתוח – (מקורות מומלצים)

URL

<https://www.w3schools.com/>

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web

<https://developer.mozilla.org/en-US/docs/Web>