

Web Technologies & Web Security

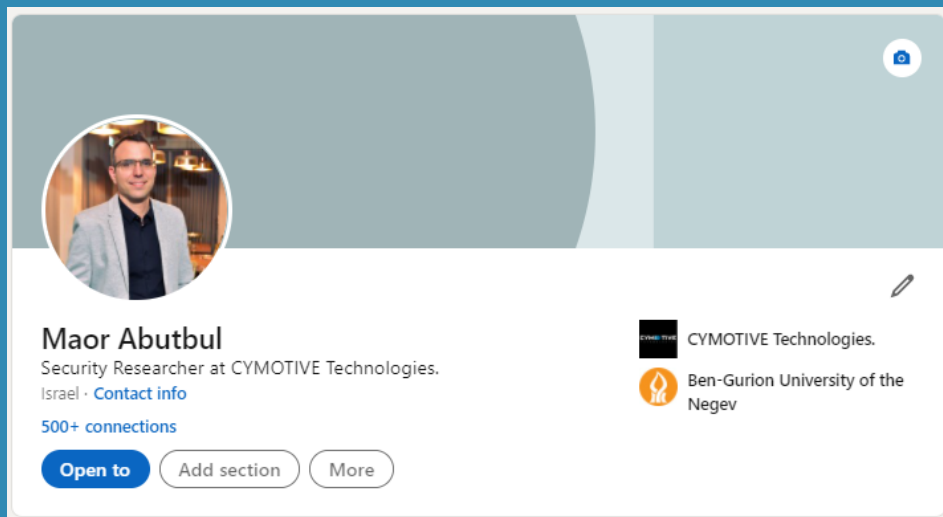
מאור אבוטבול
עבור חניכי מגשימים 2023

27.06.2023



#whoami

מי אני



- יבנה
- רקע מקצועי
- לימודים

- עד לאחרונה - חוקר אבטחה - סיימוטיב טכנולוגיות

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

תוכנית - Agenda

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

טכנולוגיות ווב (בפיתוח) – Web Technologies

Web Technologies - HTML CSS JavaScript

HTML - HyperText Markup Language

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

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

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

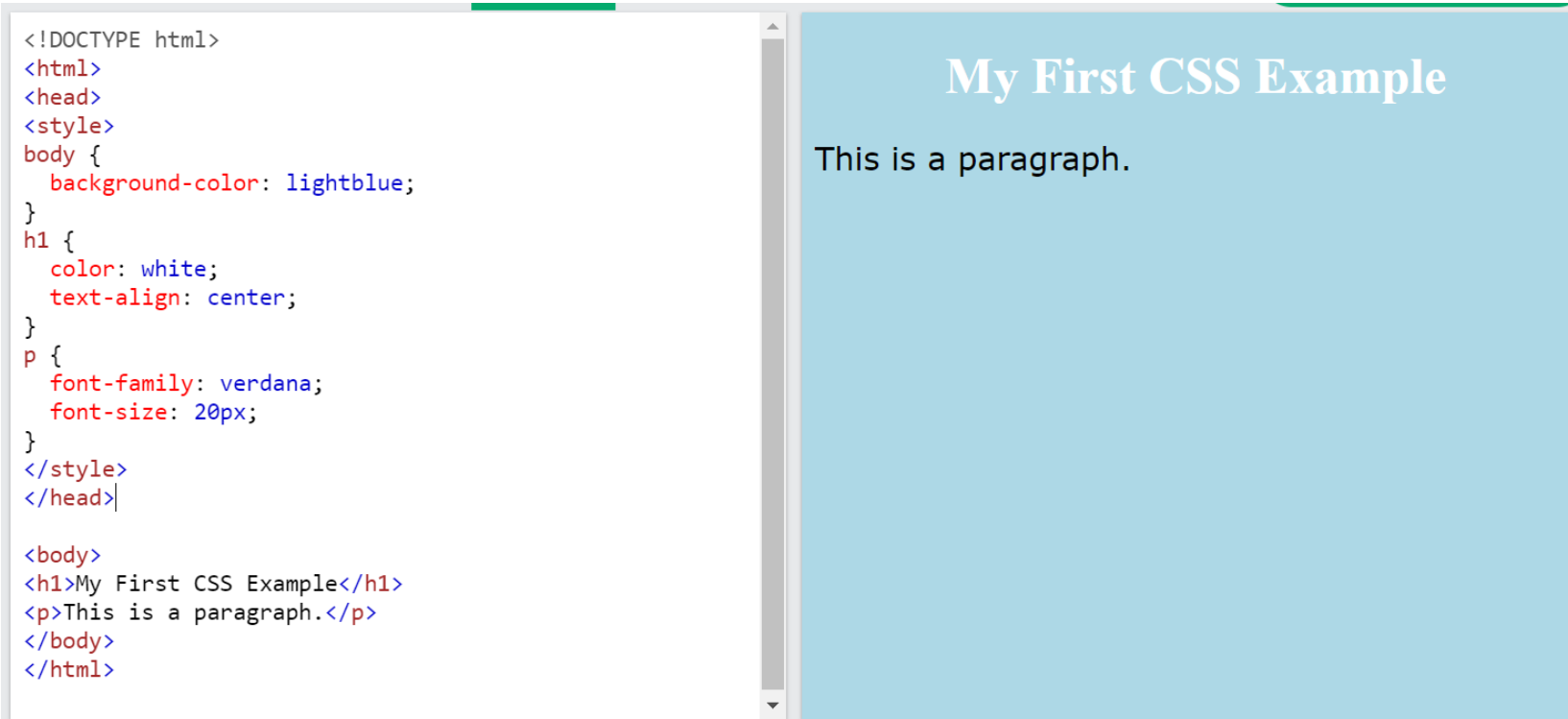
</body>
</html>
```

This is a Heading

This is a paragraph.

CSS - Cascading Style Sheets

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



JavaScript

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

<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>
---	--

טכנולוגיות צד שרת

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

Server Side

- PHP
- ASP
- Node.js

- עיבוד הבקשה נעשה בצד השרת

טכנולוגיות Web (בפיתוח) – מקורות מומלצים

URL

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

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

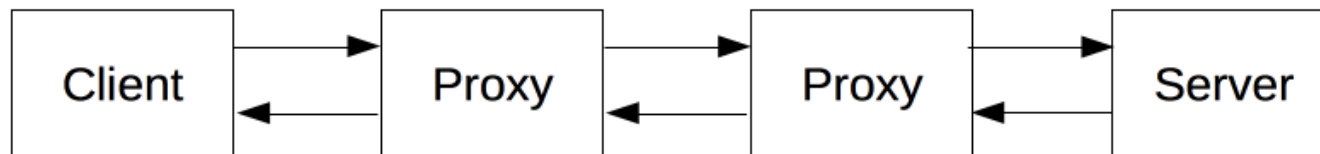
<https://developer.mozilla.org/en-US/docs/Web>

מבט מלמעלה - אבטחת מערכות תקשורת

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

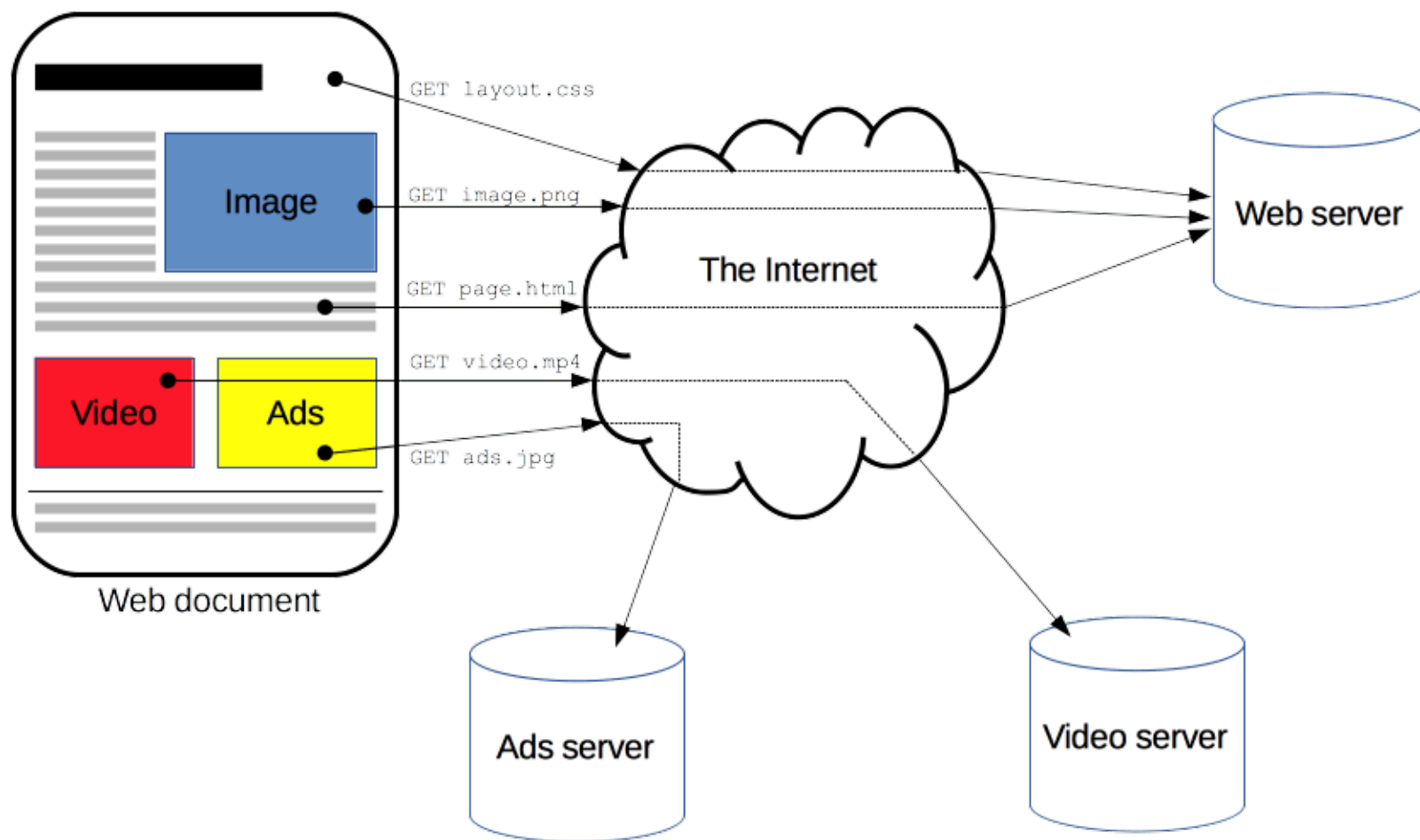
- רכיב פרוקסי (Proxy)
- Application Load Balancer
- Web Cache
- WAF – Web Application Firewall
- Web Server (שרת ווב)

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

פרוטוקול - HTTP

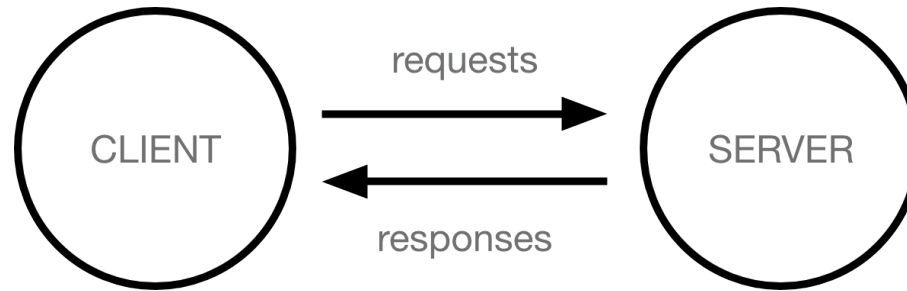
Hypertext Transfer Protocol (HTTP)

פרוטוקול HTTP



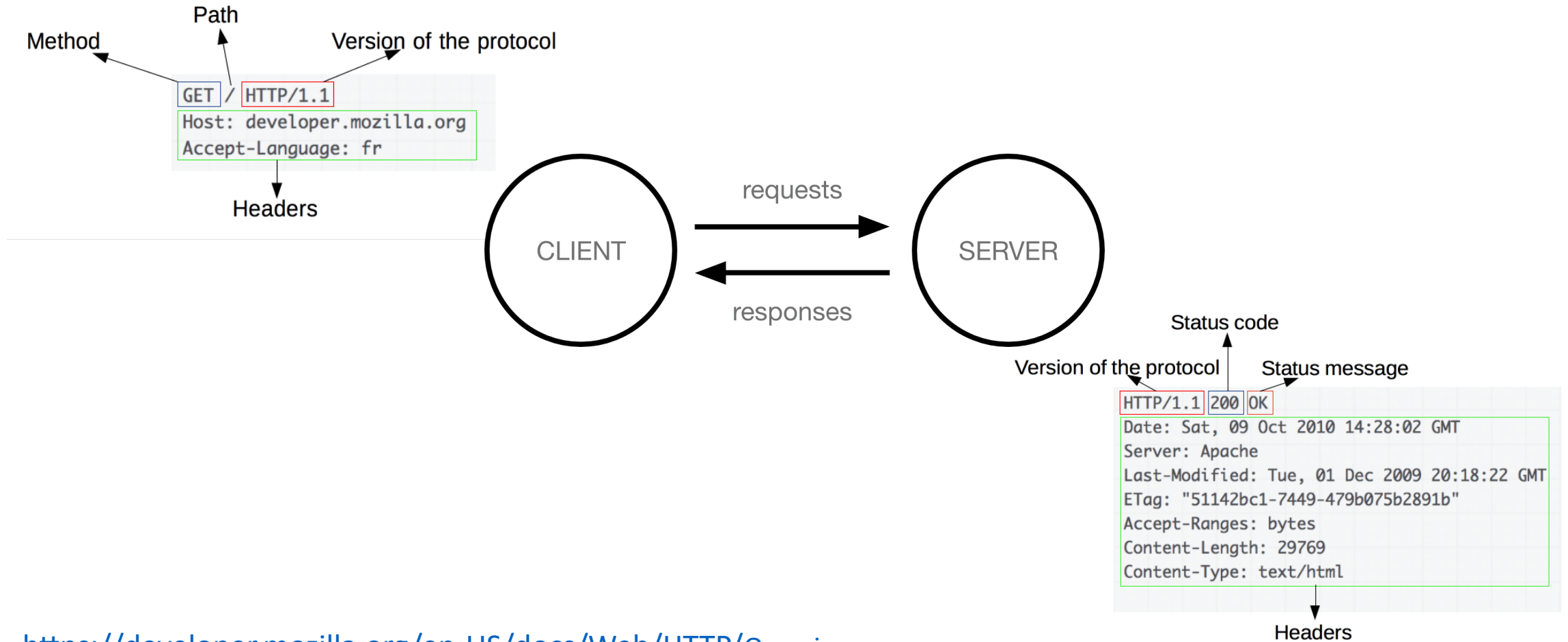
פרוטוקול HTTP - מודל שרת לקוח והודעות

- 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>

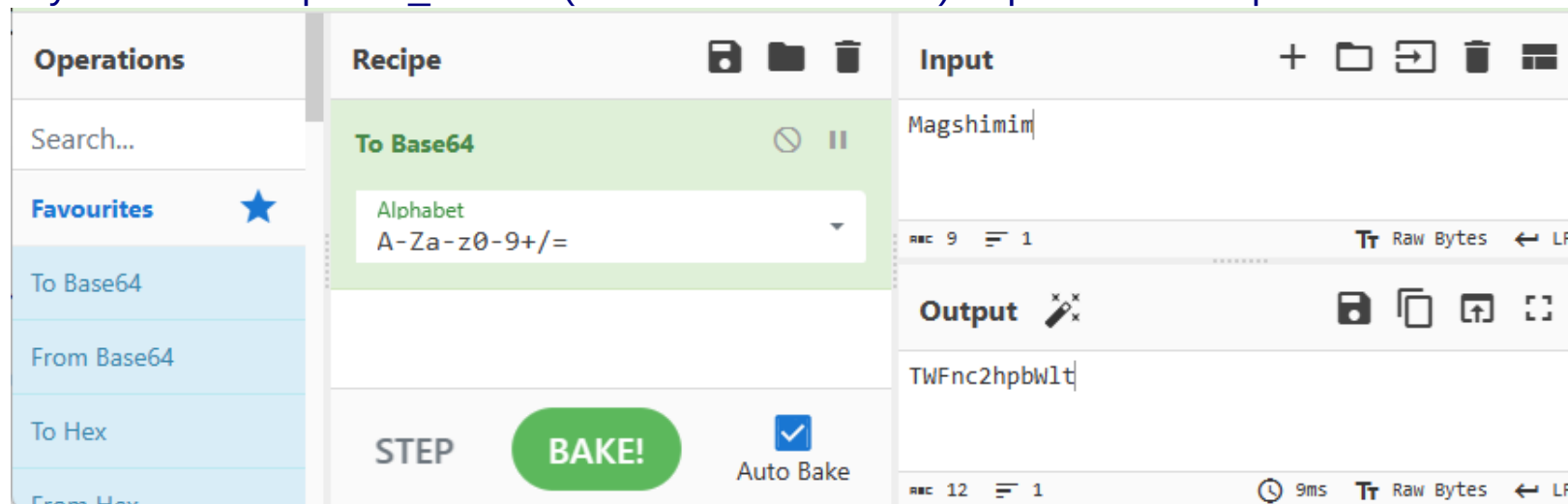
כלי פיתוח ומחקר – Tools

קידוד - Base64 Encoding

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

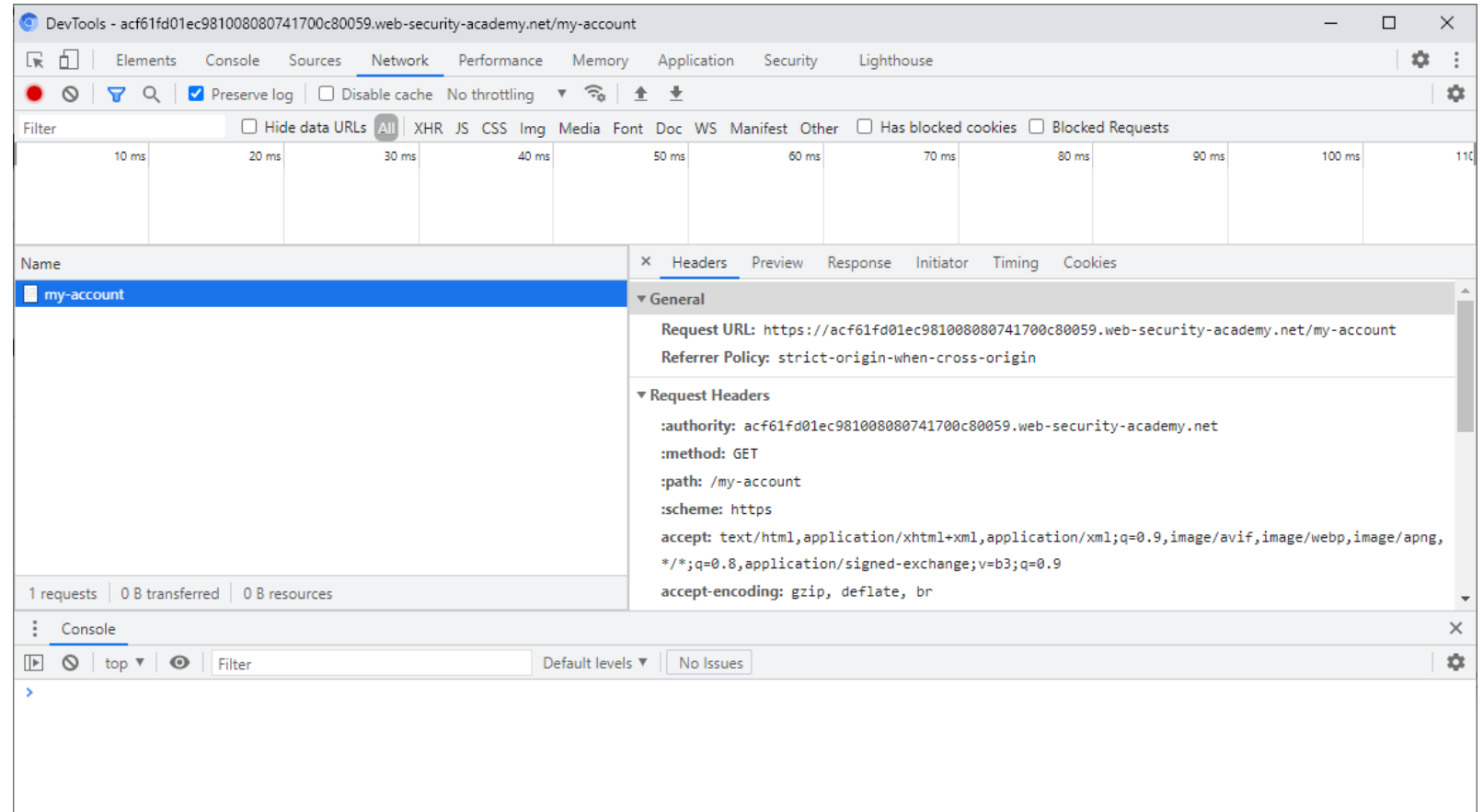
- Base64 is designed to carry data stored in binary formats across channels that only reliably support text content.
- Widely used for sending E-mail attachments. This is required because SMTP – in its original form – was designed to transport 7-bit ASCII characters only.

- <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)



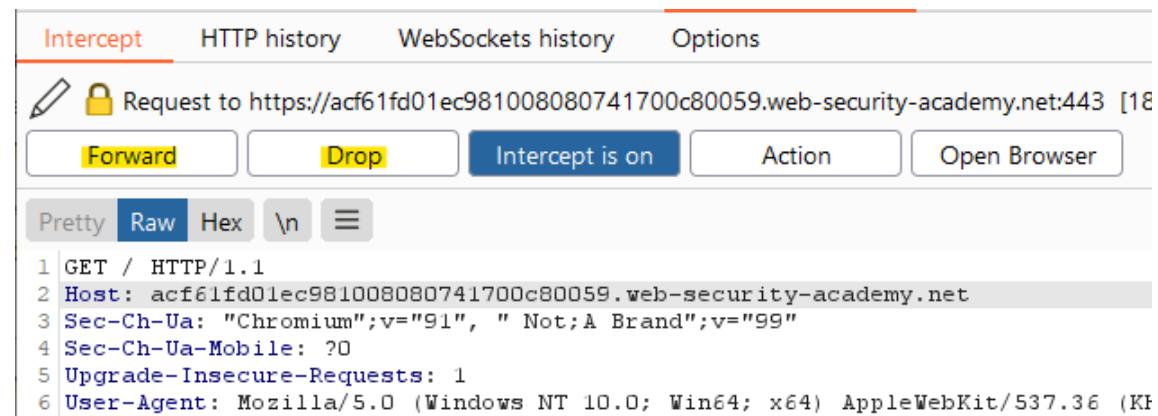
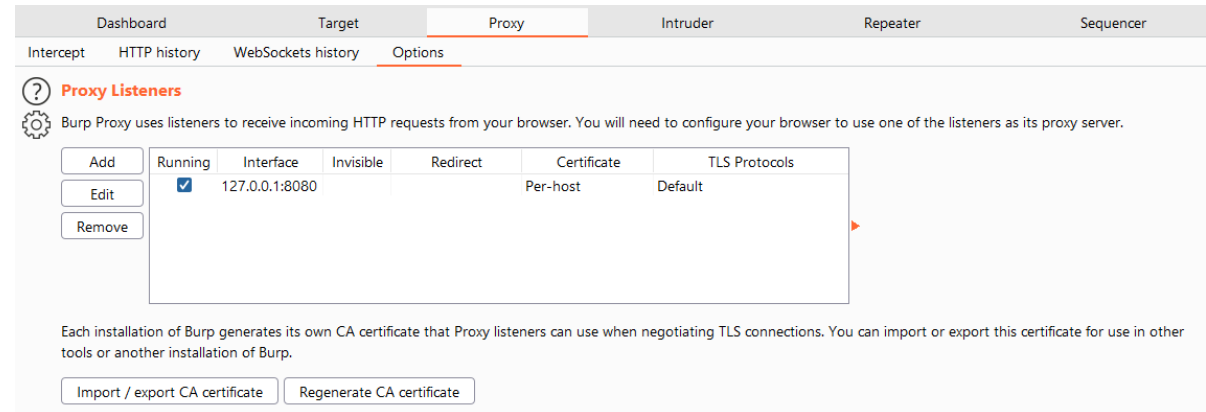
Browser DevTools

- 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



Burp Suite Cont'

- Repeater (Ctrl + R)

• עריכה ושליחה ידנית

- Intruder (Ctrl + I)

• הגדרת "משתנים" וערכים (Payloads) לשליחה אוטומטית

- Decoder

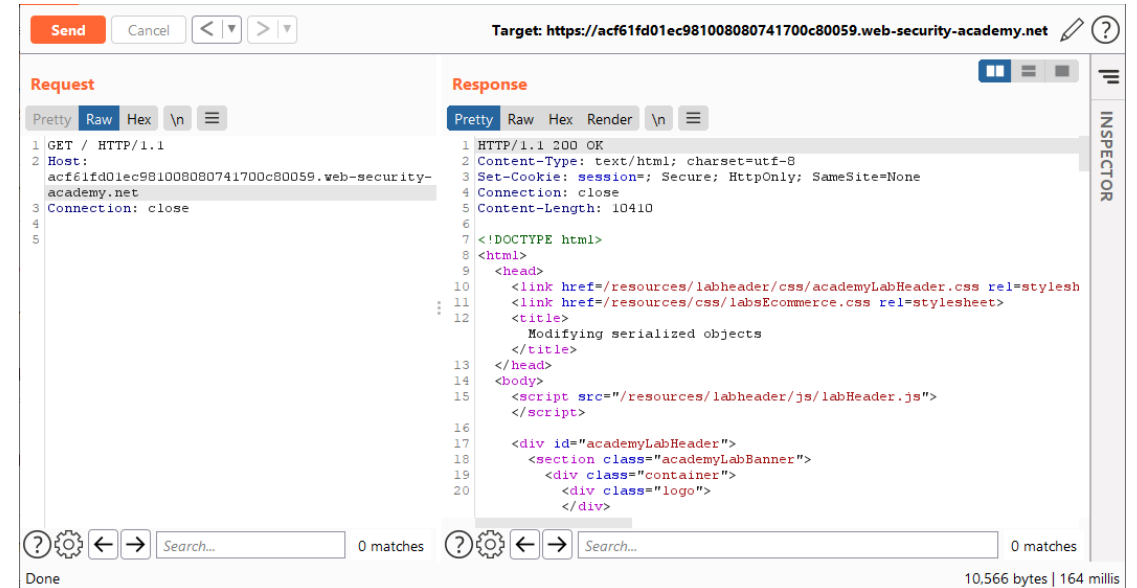
• פענוח קידודים

- Comparer

• השוואה בין בקשות

- Extender

• הורדת וניהול "הרחבות"



כלים - Tools

- Cyber Chef
 - <https://gchq.github.io/CyberChef/>
- Browser Dev-Tools (developer)
- (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
- pdf.חוק המחשבים
- 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/>
 - <https://owasp.org/www-project-top-ten/>
- 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

- Portswigger - Web Security Academy
 - <https://portswigger.net/web-security/learning-path>

Server-side topics

1 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 →](#) 16 Labs

2 Authentication

[Go to topic →](#) 14 Labs

3 Directory traversal

[Go to topic →](#) 6 Labs

4 Command injection

[Go to topic →](#) 5 Labs

5 Business logic vulnerabilities

[Go to topic →](#) 11 Labs

6 Information disclosure

[Go to topic →](#) 5 Labs

7 Access control

[Go to topic →](#) 13 Labs

8 Server-side request forgery (SSRF)

[Go to topic →](#) 7 Labs

9 XXE injection

[Go to topic →](#) 9 Labs

Client-side topics

10 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

11 Cross-site request forgery (CSRF)

[Go to topic →](#) 8 Labs

12 Cross-origin resource sharing (CORS)

[Go to topic →](#) 4 Labs

13 Clickjacking

[Go to topic →](#) 5 Labs

14 DOM-based vulnerabilities

[Go to topic →](#) 7 Labs

15 WebSockets

[Go to topic →](#) 3 Labs

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.

16 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

17 Server-side template injection

[Go to topic →](#)

7 Labs

18 Web cache poisoning

[Go to topic →](#)

13 Labs

19 HTTP Host header attacks

[Go to topic →](#)

6 Labs

20 HTTP request smuggling

[Go to topic →](#)

12 Labs

21 OAuth authentication

New Topic

[Go to topic →](#)

6 Labs

- <https://portswigger.net/web-security/learning-path>

Insecure Deserialization

חולשה נפוצה - פתיחה לא מאובטחת של רצף סדרתי

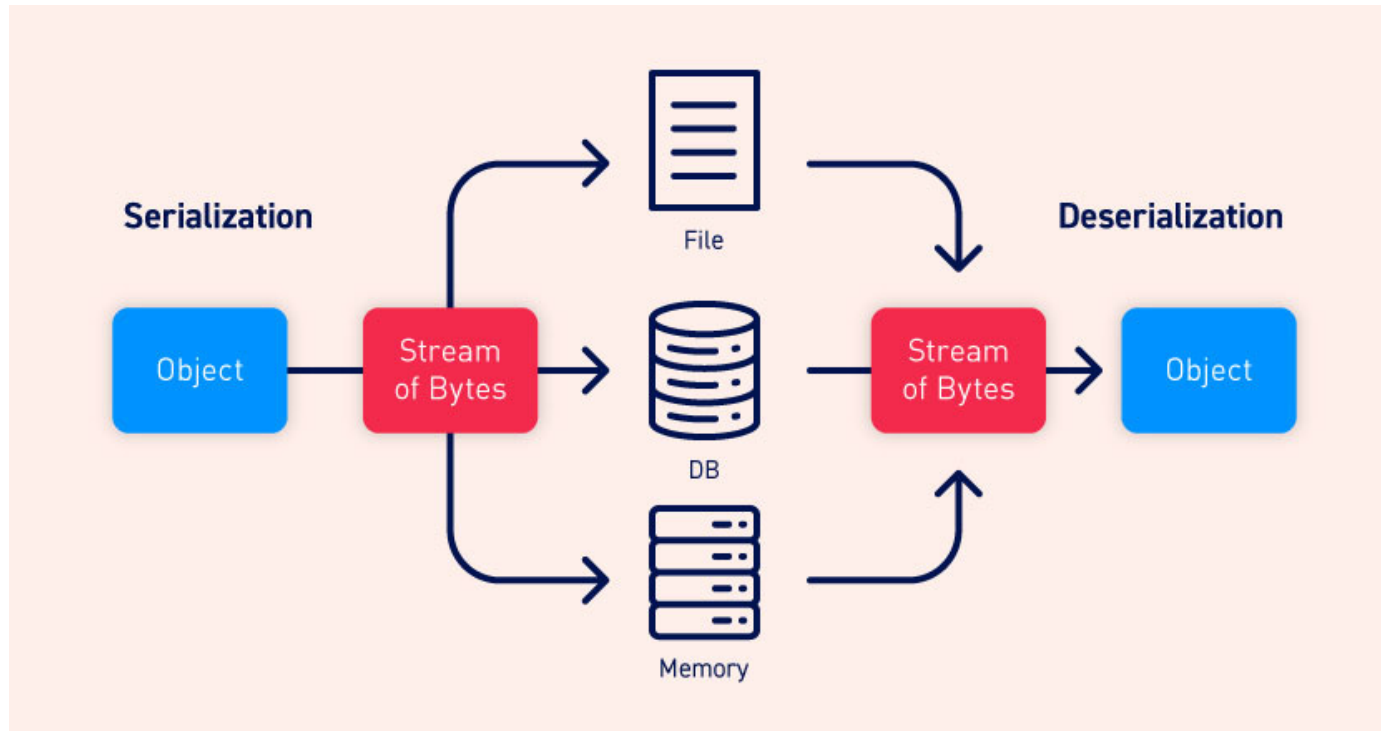
פתיחה לא מאובטחת של רצף סדרתי - Insecure Deserialization

- 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

• פורמט הסריאליזציה של PHP

• פורמט מבוסס טקסט

• אותיות מייצגות את סוג המשתנה (Data Type)

• מספרים מייצגים אורך של השדה המקודד

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

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

Jackson

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](#) : Vulnerability Statistics

[Products \(6\)](#) [Vulnerabilities \(69\)](#) [Search for products of Fasterxml](#) [CVSS Scores Report](#) [Possible matches for this vendor](#) [Related Metasploit Modules](#)

[Vulnerability Feeds & Widgets](#)

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2016	1														
2017	1														
2018	6		5							3					
2019	22		4												
2020	26		2												
2021	13														
Total	69		11							3					
% Of All		0.0	15.9	0.0	0.0	0.0	0.0	0.0	0.0	4.3	0.0	0.0	0.0	0.0	

Warning : Vulnerabilities with publish dates before 1999 are not included in this table and chart. (Because there are not many of them and they make the page look bad; and they may not be actually published in those years.)

- <https://www.cvedetails.com/vendor/15866/Fasterxml.html>

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>


CVE-2023-28667 Detail

Description

The Lead Generated WordPress Plugin, version <= 1.23, was affected by an unauthenticated insecure deserialization issue. The tve_labels parameter of the tve_api_form_submit action is passed to the PHP unserialize() function without being sanitized or verified, and as a result could lead to PHP object injection, which when combined with certain class implementations / gadget chains could be leveraged to perform a variety of malicious actions granted a POP chain is also present.

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

 **NIST: NVD** **Base Score:** **9.8 CRITICAL** **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

<https://www.tenable.com/security/research/tra-2023-7>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-28667>

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 הקרוב – נדרש רישום בתשלום
 - <https://bsidestlv.com/ctf/>
 - התחיל אתמול 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>

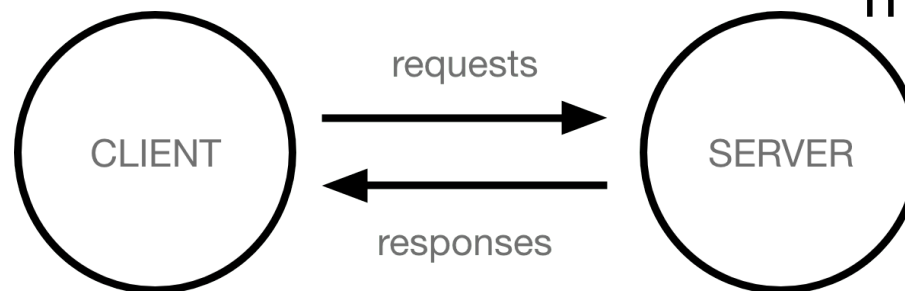
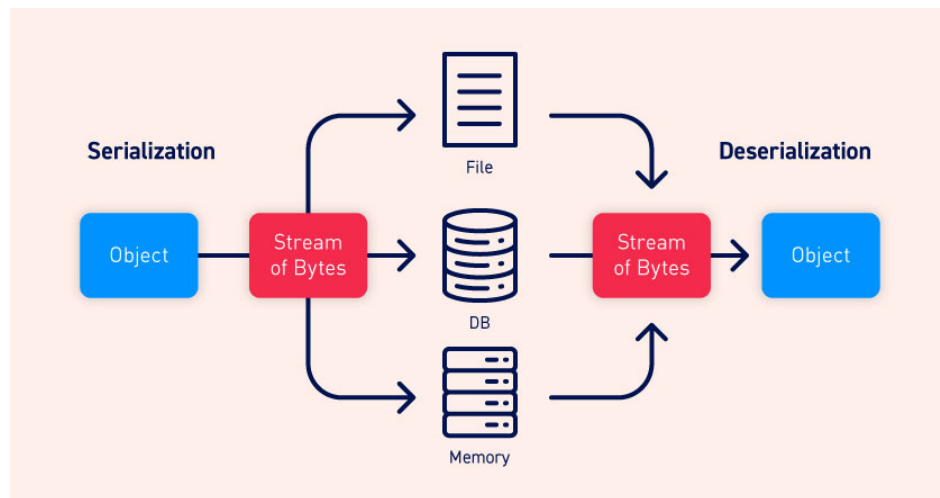
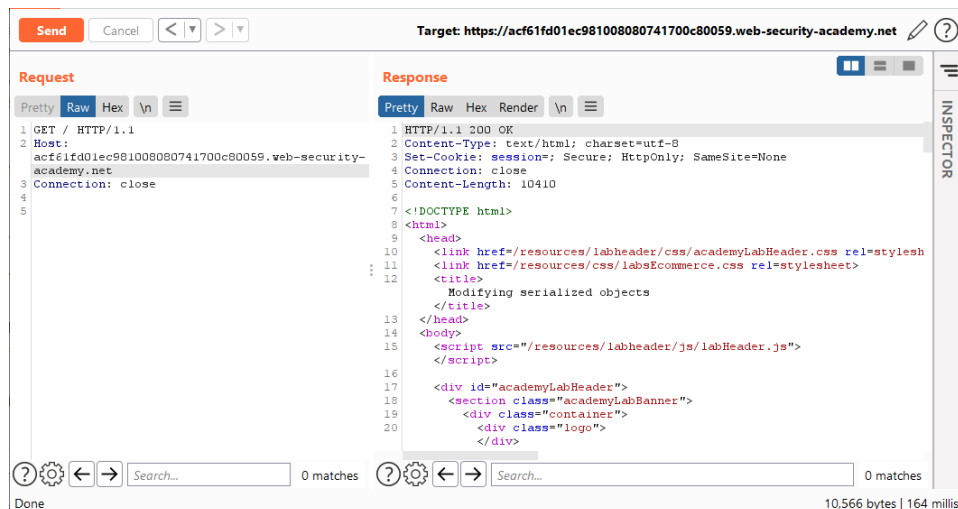
תחרויות ותוכניות חיפוש חולשות – CTF & Bug Bounty

- 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

• רשימה לא מלאה של כלים שכדאי להכיר

- 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)
- 7Zip
- VSCode
- Docker
- VirtualBox / VMware
- Git & Github