Session Hijacking: A New Method of Prevention
The Increasing Importance of Secure Web Application Delivery

Today, the enterprise runs on the Web. Whether it’s providing employees around the world with efficient access to information or offering customers products and services, more and more organizations are leveraging the power of online applications.

New access points to the Internet are opened every day. Laptops, tablets and smartphones enable users to log in from anywhere at any time.

But this flexibility presents a dilemma. Successful organizations must balance the need for convenient user access with appropriate security techniques to keep hackers from exploiting access points for e-commerce fraud, identity theft and other malicious activities.
Session Hijacking: Defined

Due to server-side convenience, HTTP is the predominant method for offering users access to web applications. And because HTTP is a stateless protocol, web applications primarily employ cookies to maintain a session state once the user has logged in.

Before

Maintaining a session state via cookies offers a simple experience for end users—they’ve proven their identities (and identities of their devices) via authentication and can move quickly to accessing desired information from an application.

After

But this user-friendly experience comes at a cost. Cookies can be stolen, intercepted or replayed.

While the need for a favorable user experience is a driving concern for organizations of all sizes, the reliance on the HTTP protocol and HTTP cookies creates vulnerabilities for hackers to steal authenticated sessions.
Session Hijacking: Understanding the Threat

Session hijacking is not a new phenomenon and has been considered a viable threat since HTTP 1.1. A persistent thorn in the side of IT security, session hijacking is now returning to prominence. In part, this is due to the increased use of stricter security protocols.

As two-factor login, risk-based authentication and other methods become more widespread in the IT security community, hackers have shifted their focus to weaker links in the chain, such as HTTP sessions themselves. For hackers, mimicking the identity of a user via stolen log-in credentials is becoming increasingly difficult. Instead, they are allowing users to create a session and then hijacking that session’s credentials to steal data.

The Open Web Application Security Project (OWASP) highlights session hijacking in its report, *OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks*. OWASP specifically calls out these areas:

- **Difficulty enabling mobility without increasing risk**
- **Cross-site Scripting (XSS)**
- **Cross-site Request Forgery (CSRF)**
Session Hijacking: Understanding the Threat

Session hijacking is not a new phenomenon and has been considered a viable threat since HTTP 1.1. A persistent thorn in the side of IT security, session hijacking is now returning to prominence. In part, this is due to the increased use of stricter security protocols.

As two-factor login, risk-based authentication and other methods become more widespread in the IT security community, hackers have shifted their focus to weaker links in the chain, such as HTTP sessions themselves. For hackers, mimicking the identity of a user via stolen log-in credentials is becoming increasingly difficult. Instead, they are allowing users to create a session and then hijacking that session’s credentials to steal data.

The Open Web Application Security Project (OWASP) highlights session hijacking in its report, *OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks*. OWASP specifically calls out these areas:

- **Difficulty enabling mobility without increasing risk**
  Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys or session tokens, or to exploit other implementation flaws to assume the identities of other users.

- **Cross-site Scripting (XSS)**

- **Cross-site Request Forgery (CSRF)**
Session Hijacking: Understanding the Threat

Session hijacking is not a new phenomenon and has been considered a viable threat since HTTP 1.1. A persistent thorn in the side of IT security, session hijacking is now returning to prominence. In part, this is due to the increased use of stricter security protocols.

As two-factor login, risk-based authentication and other methods become more widespread in the IT security community, hackers have shifted their focus to weaker links in the chain, such as HTTP sessions themselves. For hackers, mimicking the identity of a user via stolen log-in credentials is becoming increasingly difficult. Instead, they are allowing users to create a session and then hijacking that session’s credentials to steal data.

The Open Web Application Security Project (OWASP) highlights session hijacking in its report, OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks. OWASP specifically calls out these areas:

- **Difficulty enabling mobility without increasing risk**
- **Cross-site Scripting (XSS)**
- **Cross-site Request Forgery (CSRF)**

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim’s browser, which can steal user sessions, deface websites or redirect the user to malicious sites.
Session Hijacking: Understanding the Threat

Session hijacking is not a new phenomenon and has been considered a viable threat since HTTP 1.1. A persistent thorn in the side of IT security, session hijacking is now returning to prominence. In part, this is due to the increased use of stricter security protocols.

As two-factor login, risk-based authentication and other methods become more widespread in the IT security community, hackers have shifted their focus to weaker links in the chain, such as HTTP sessions themselves. For hackers, mimicking the identity of a user via stolen login credentials is becoming increasingly difficult. Instead, they are allowing users to create a session and then hijacking that session’s credentials to steal data.

The Open Web Application Security Project (OWASP) highlights session hijacking in its report, OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks. OWASP specifically calls out these areas:

- **Difficulty enabling mobility without increasing risk**
- **Cross-site Scripting (XSS)**
- **Cross-site Request Forgery (CSRF)**

A CSRF attack causes a logged-on victim’s browser to send a forged HTTP request to a vulnerable web application. This allows the attacker to force the victim’s browser to generate requests the vulnerable application thinks are legitimate.¹

¹ The Open Web Application Security Project (OWASP) https://www.owasp.org/index.php/Top_10_2013-Top_10. Licensed under the creative commons Attribution-ShareAlike 3.0 license (http://creativecommons.org/licenses/by-sa/3.0/). No changes were made to the content.
As risk-based authentication becomes a stronger hacking deterrent, session hijacking is gaining popularity. Strong authentication is emerging as the impenetrable front door, but there are still critical security weaknesses inherent in HTTP sessions. Some of the specific session hijacking techniques include:

**Session Hijacking Techniques**

- **Spoofing attack**
  Unauthorized session access based on falsifying data

- **Man-in-the-middle attack**
  Stealing an in-transit HTTP cookie without the user’s knowledge

- **Man-in-the-browser attack**
  Installation of code on the browser itself to forward data to a third party

Various CSS and XSS attacks are also common approaches used by hackers and fraudsters. And perhaps more concerning for enterprise security are the unknown types of attacks that are being continually developed and directed against web sessions with increasing frequency.
Single Sign-On (SSO) is rightly regarded as one of the most effective ways to provide a convenient experience—one log in and the user can access the data needed from multiple applications.

But SSO can also be a hacker’s best friend—because hackers can access multiple applications with just one stolen cookie or session token.
Traditional Approaches to Session Hijacking Prevention

Session hijacking techniques and methods of prevention have developed side by side for nearly two decades. Some of the common forms of prevention employed by the enterprise include:

**Architectures**
Rearranging the network architecture is a typical IT security approach. By reorganizing their IT systems to a hub-and-spoke model, security experts attempt to dramatically limit the spread of cookies to other applications.

**Cookie management**
IT security teams may also restrict the proliferation of data by instituting rules and best practices around cookies themselves—for example, limiting usage to only secure cookies, such as those that are HTTP-only or host-only.

**Limited external checking**
Focusing on IP addresses is another common approach to verifying the validity of a given session. However, proxies, dynamic IP addresses and other factors can obscure the IP address of the actual device initiating the session.

**Timeouts**
Limiting the duration of a cookie’s validity can also be used to blunt the threat of session hijacking, but the damage may be done by the time a session timeout requires the user to re-authenticate the session.

Although a good starting point, these methods are all riddled with shortcomings that render them ineffective to enterprises that need to reduce the threat of session hijacking and maintain a superior end-user experience while supporting multiple ways to access applications.
By limiting the scope and damage of attacks, traditional approaches to session hijacking have achieved some acceptable level of effectiveness. However, there are new techniques emerging—methods that work in concert with strong authentication—to make it extremely difficult for hackers to steal sessions.
More-in-the-Middle: An Innovative Approach to Session Hijacking Prevention

By limiting the scope and damage of attacks, traditional approaches to session hijacking have achieved some acceptable level of effectiveness. However, there are new techniques emerging—methods that work in concert with strong authentication—to make it extremely difficult for hackers to steal sessions.

1. User logs into website
2. Before session token is created user redirected to DeviceDNA™ collection service
3. DeviceDNA collected
4. DeviceDNA stored and session token created
5. User taken to web server with secured session

More Control

By adding an interim step between the client and the server, the enterprise can better control the security of a given session.

The user receives a session token not from the server or website, but from the layer in the middle that ties the session token to the specific device that was used for authentication. This layer serves as an objective check against stolen cookies. Technology that employs this methodology typically re-checks the device’s identity on a periodic basis, preventing a fraudster from stealing the cookie and using it to log in to an application.

More Connection

An Active Approach to Session Security
More-in-the-Middle: An Innovative Approach to Session Hijacking Prevention

By limiting the scope and damage of attacks, traditional approaches to session hijacking have achieved some acceptable level of effectiveness. However, there are new techniques emerging—methods that work in concert with strong authentication—to make it extremely difficult for hackers to steal sessions.

1. User logs into website
2. Before session token is created user redirected to DeviceDNA™ collection service
3. DeviceDNA collected
4. DeviceDNA stored and session token created
5. User taken to server with secured session
More-in-the-Middle: An Innovative Approach to Session Hijacking Prevention

By limiting the scope and damage of attacks, traditional approaches to session hijacking have achieved some acceptable level of effectiveness. However, there are new techniques emerging—methods that work in concert with strong authentication—to make it extremely difficult for hackers to steal sessions.

CA Single Sign-On provides IT with a critical feature: Enhanced Session Assurance with DeviceDNA™. This component ‘remembers’ the device the user was initially authenticated on and then actively compares the settings and history of the user’s device against the initial device to further guarantee the identity of the user and legitimacy of the login attempt. The patent-pending technology from CA Risk Authentication that is incorporated provides one of the differentiators that sets this approach apart from traditional methods of session security.

1. User logs into website  
2. Before session token is created user redirected to DeviceDNA™ collection service  
3. DeviceDNA collected  
4. DeviceDNA stored and session token created  
5. User taken to web server with secured session
# Session Hijacking in Summary

## Threat
- ✓ As web applications dominate the business landscape, session hijacking is increasingly dangerous to the security of the enterprise.
- ✓ Improvements in authentication methods have shifted hackers’ focus away from web session login credentials and toward the actual session.
- ✓ Session hijacking comes in a variety of guises, including man-in-the-middle and man-in-the-browser attacks.

## Traditional Approaches
- ✓ Enterprise security has been locked in a war of innovation against session hijackers since HTTP 1.1.
- ✓ Alternate architectures, cookie management and IP checking are some of the common approaches that enterprise security has employed to guard against session hijacking.
- ✓ These methods provide limited effectiveness and only serve to restrict the inevitable damage.

## Technology of Today
- ✓ More modern approaches involve the insertion of another layer between the application and the server.
- ✓ This middle layer is controlled by enterprise IT security and serves up the session token to the user, making session hijacking extremely difficult.
- ✓ To increase the efficacy of this middle layer, the enterprise must link the SSO cookie, and any application cookies in use, to a specific device.
About the Solution from CA Technologies

CA Single Sign-On secure SSO & flexible access management can provide your organization enterprise-class secure single sign-on (SSO) and flexible identity access management so that your organization can authenticate users and control access to web applications and portals. Across internet, intranet and cloud applications, it helps enable the secure delivery of essential information and applications to your employees, partners, suppliers and customers via secure single sign-on. It also scales to help you meet your growing business needs with flexible administration tools that can support either centralized or distributed administration.

For more information about preventing session hijacking, visit ca.com/secure-sso.