

How to prevent session hijacking

Although session hijacking is basically a client security issue, most attacks can be detected on the server side. This article describes how airlock helps to protect web applications against session hijacking.

Session tracking on airlock

By default, airlock recognizes http requests of the same browser/user with a **session ID** in a cookie.

As an alternative mechanism, HTTPS sessions may be tied to the SSL session ID instead (option Track sessions based on SSL session ID on configuration page Session settings). An SSL session ID is much harder to reuse than a http cookie. A drawback of this solution is that some clients may have to re-login very frequently (due to a short SSL session timeout in their browser/operating system). Another problem is introduced by forward proxies that share SSL sessions among the users: This leads to inadvertent "session sharing" between those users; airlock can therefore no longer distinguish different users.

Cookie stealing and session hijacking

If an attacker is able to steal the session cookie, he can pretend to be the same user, or hijack the session during its lifetime. An attacker can therefore send requests (or issue transactions) in the name of the user until either the session times out or the user manually terminates the session by clicking some logout button. airlock mitigates this threat by using a separate session cookie for HTTPS requests, using SSL to transport the token, clearing the token when the session ends, and causing the token to expire after a period of client inactivity. Additionally, current browsers try to protect all cookies ("same-origin policy").

Client vs. server side security

Nowadays, the best chance for attackers to steal session cookies is by installing malicious software on the client (viruses, worms, trojan horses, spyware etc.) [1]. This "malware" can easily steal the cookie and either send http requests directly from the infected machine or forward the cookie to another system controlled by the attacker.

Please note that the infection itself can only be prevented by client security measures like a personal firewall, Spyware-, Virus- and Trojan-Scanners. Some remote access solutions with so-called "clientless security" features claim to enforce that such protection software is active and up-to-date. In practice however, they are a viable solution for controlled client environments only, but they are not suited for public websites, B2C or e-Business solutions.

In order to maximize the protection on the server side, **airlock offers several countermeasures to detect and prevent session hijacking:**

- Since version 3.5, the option **VerifyClientIP** ensures that the IP address does not change during a session. If it changes anyway, the session is terminated immediately. Unfortunately, this may lead to undesired session terminations if the client is behind a load-balancing cluster of forward proxies.

For enabling this option in airlock 4.x, edit the file `/zone/mgt/root/opt/slt/ses/gatekeeper/resource/Static.rsc` to change this line:

```
SecurityGateway * VerifyClientIP "TRUE"
```

Do not forget to restart the `security_gate` process after the change:

```
zlogin int; kill -HUP `cat /var/run/sg/security_gate.pid`
```

Session Hijacking

- With version 4.0-9.35 or later, airlock detects changes of the SSL session-id within the same (cookie-tracked) session. If a session is accessed intermittently using two distinct SSL sessions-ids, this is treated as a clear sign of session hijacking and the session is therefore terminated.

For backward-compatibility reasons, this feature has to be manually activated after installing update 4.0-9.35: Edit the file `/zone/mgt/root/opt/slt/ses/gatekeeper/resource/Static.rsc` and un-comment the following lines (remove the leading #):

```
#SecurityGateway * SessionHijackPenaltyThreshold "1000"  
#SecurityGateway * SessionHijackPenaltySslSessionIdChange "10"  
#SecurityGateway * SessionHijackPenaltySslSessionIdAlternation "1000"
```

- Session hijacking prevention has been extended with airlock 4.1-10.28:

An intelligent engine **fingerprints all requests** of a session by not only tracking the IP address and SSL session id, but also by **analyzing the http headers**. Each change (e.g. different http header order) adds penalty points to the session, which is terminated as soon as the sum exceeds a configurable limit. For more details, consult the article Prevent Session Hijacking with HTTP Client Fingerprints available in » [Myphion](#).

Conclusion

Secure session tracking should not rely on either cookies or ssl session-ids alone, but rather a combination of these two plus many more factors. **airlock detects and prevents session hijacking by continuously checking this fingerprint of a users requests.**

References

[1] MELANI, Swiss Federal Administration: » [Hacker attacks on the increase](#)