# POSTER: Shepherd – Enabling large-scale post-login studies

No Author Given

No Institute Given

## Shepherd: automating the login process

Many websites offer logins. In 2010, the Firefox extension FireSheep trivialised session hijacking of users in the same wifi network. Many websites still used the insecure HTTP protocol, which transmits session identifiers and passwords in the clear. Firesheep listened to wifi traffic, recognising authentication cookies of specific, preconfigured websites (incl. Google and Facebook), and intercepted these. In response, the major websites switched to secure HTTPS communication. However, it remains unclear to what extent other websites adopted security measures. Though prior research (e.g., [MFK16, ZE14]) showed scale benefits over manual efforts, post-login studies are still rare. In this work we present our efforts to make such studies feasible. We created Shepherd, a framework that achieves automatic logging in on unknown websites, either by using domain-specific credentials or SSO accounts. For domain-specific logins, a set of websites and credentials is loaded into Shepherd, after which it performs a mulitphase programme to login and evaluate these sites (c.f. Figure 1). It further provides the possibility to hook in custom modules which will be hooked in after successfully logging in. This allows the conduction of arbitrary research of post-login areas.

### Contributions.

1. We designed and developed Shepherd, a tool to automate logging in for large-scale studies.
   As large-scale application tends to surface use cases not encountered in tests, Shepherd includes verification of supposedly successful logins and detection of invalid credentials.
2. We performed a large-scale evaluation of Shepherd.
   We acquired domain-specific credentials for 50K domains from a legitimate source. Unfortunately, many were no longer valid. Out of the 15,343 sites where Shepherd did not detect an "invalid credentials"-alike message, Shepherd successfully logged in 7,113 times. Figure 2 presents the detected failures in each step
3. We evaluated the security of authentication cookies on these sites.
   Our implementation allows automated detection of authentication cookies, building on the work by Mundada et al. [MFK16]. Then, we evaluated the adoption of cookie-affecting security measures. Specifically, we checked

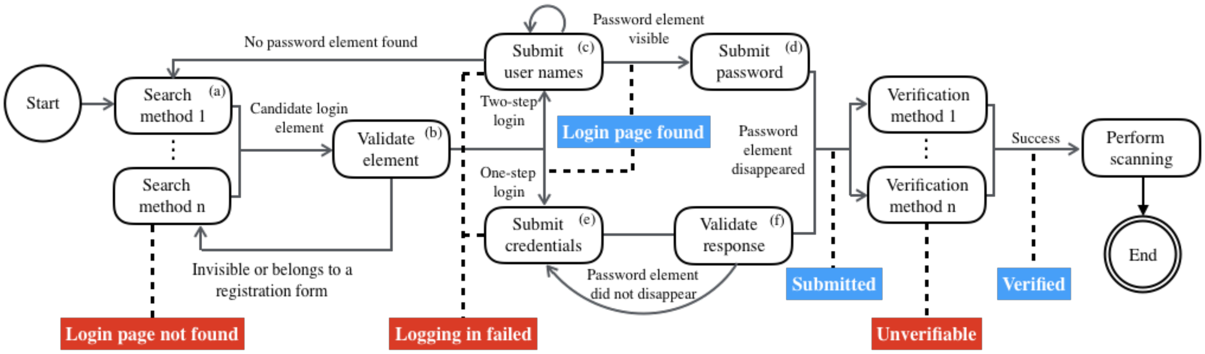

**Fig. 1.** Steps in automating logging in.

whether the site supported HSTS, and whether it set any of the following cookie flags: SameSite, HTTPOnly, Secure.

4. We found that 34% of tested websites were susceptible to straightforward session hijacking attacks.

## Results

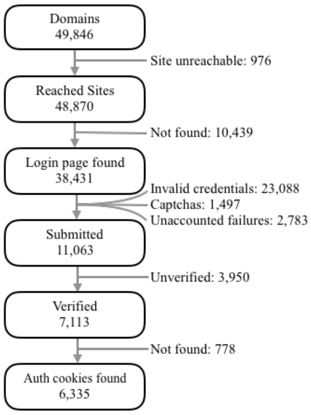

**Fig. 2.** Scanning results domain-specific scan.

| | #sites | % |
|---|---|---|
| | 7,113 | 100.0% |
| HSTS | 1,268 | 17.8% |
| HttpOnly | 4,417 | 62.1% |
| Secure | 4,543 | 63.9% |
| HSTS/insecure | 153 | 2.2% |
| SameSite | 0 | 0.0% |
| **Susceptible** | **2,417** | **34.0%** |

Security evaluation
(domain specific credentials).

## References

MFK16. Yogesh Mundada, Nick Feamster, and Balachander Krishnamurthy. Half-Baked Cookies: Hardening cookie-based authentication for the modern web. In *Proc. 11th Asia Conference on Computer and Communications Security (ASIACCS)*, pages 675–685, 2016.

ZE14. Yuchen Zhou and David Evans. Ssoscan: Automated testing of web applications for single sign-on vulnerabilities. In *Proc. 23rd USENIX Security Symposium*, pages 495–510. USENIX Association, 2014.