OpenID

Shivaram Lingamneni
Ben Newman
The Protocol

1. Initiator

2. Discovery URL

3. OpenID Provider

4. Relying Party

5. Optional

6. Connectors
Protocol Messages

- Initiator: User-supplied identifier (USI) (1)
- RP: discovery (2), secret sharing (3)

Indirect messages

- RP to OP: USI, RP, secret handle (4)
- OP to RP: USI, OP, RP, secret handle, nonce, signature (5)

Fields must match, the signature must verify, nonce must be unique

- RP issues an ID token (6)
Identity Asymmetries

- RP and OP identified by URIs
- Initiator identified by:
  - User-supplied identifier
  - Session cookies
  - IP address (implicitly)
Message-Level Vulnerabilities

- Protocol designers/implementors not concerned with conventional MITM attacks:
  - Attacker could substitute own OP endpoint URL during discovery
  - OP session cookie could be stolen by eavesdropper
Message-Level Vulnerabilities

• Entire protocol can be conducted over SSL

• HTTPS URLs make MITM attacks impossible for our purposes

• Far from universally implemented, but an easy excuse for ignoring MITM attacks

• Nonce to prevent replay attacks: the only network-level countermeasure
Message-Level Vulnerabilities

● Protocol designers more concerned with user agent-level manipulations

● Nonce needed since response messages may be passed through user agent

● Still not all such manipulations: phishing ignored as “out-of-scope”
A Less Trivial Attack

- Malicious JavaScript submits login form automatically
- User invisibly forced to login with mode “checkid_immediate”
- Puts RPs with XSRF vulnerabilities at particular risk, since users stay logged in with an OP for extended periods
Another Nontrivial Attack

- Session Swapping (Barth, et al.)
- Victim logged in with malicious party’s credentials
- Relies on RP willingness to set a cookie with any user agent that supplies a legitimate-seeming authorization response
Variation on Session Swapping

• Suppose the RP prevents cross-site login form submission

• Adversary initiates login in with victim’s USI

• XSRF the RP-OP authentication request

• Victim unwittingly logged in with own credentials
Limited Adversaries

- Full MITM power, but only over information passed through user agent?
- Malware?
- Denial of Service?
Problems

- Web-based protocol attacks are hard to model
- Messages sources a subtle issue: multiple kinds of identifiers (USI, cookie, IP)
- What privileges should the intruder possess?
- Much unspecified by OpenID protocol
One More Idea

• Fallacy: RP has nothing to gain from dishonesty

• Authentication status not strictly binary

• OpenID extensions allow arbitrary information to be transmitted back to the RP

• Falsifying the realm attribute