365-Day: HTTPS Cookie Stealing

Mike Perry
Riverbed Technology
DEFCON 2008

Who am I?

- Volunteer Tor developer
  - Work on Torbutton, TorFlow
- Privacy advocate, censorship opponent
- Forward+Reverse Engineer at Riverbed
- Flexitarian
- Random Hacker
  - Wrote a page-based malloc debugger
  - Wrote an IRC bot that got quoted as a human in a major magazine

Why am I doing this?

Exploit is not new or complicated... However:

- Vector is not narrow or wifi-only
  - Sophisticated attackers can drain bank accounts with custom cable/DSL modems
  - It also harms safe Tor usage, and that pisses me off
- Many sites are vulnerable, and don't seem to care.
- Response: Release a tool showing how bad this is
  - Basic “Proof of concept” mechanisms did not work
  - Encourage (correct and secure) SSL adoption
- It's a ONE BIT FIX PEOPLE!

Cookie Basics

- Variables set by websites in your browser
  - Used for authentication, tracking, storage
- Several properties that govern when transmitted
  - Domain
  - Path
  - Expiration
  - SSL bit (seldom used, this is where the fun begins)

The 'SideJacking' Attack

- Glorified sniffer
  - Sniffs cookies transmitted via plaintext http
- Janky proxy based approach to do control+saving
- Completely passive: User must visit target site
- Able to save domain and path info
  - Path info may be too specific
  - Can lead to issues
- Admirable PR machine for such a simple hack
  - Waay exceeds my PR abilities. Little help? :)

Active HTTP Cookie Hijacking

- Like CSRF, but we want the data transmitted, not any particular result
  - In fact, the server can reject the request
- Scenario:
  - Yesterday: User logs in to mail.yahoo.com. Checks "Remember me."
  - Today: User visits www.cnn.com via open wifi
  - Today: We inject <img src="http://mail.yahoo.com">
  - Today: Browser transmits yahoo cookies for image
  - Today: We sniff cookies, write them to cookies.txt
  - Tomorrow: Use cookies.txt to read their mail

Active HTTPS Cookie Hijacking

- New Scenario:
  - Yesterday: User logs in to httpS://mail.google.com
  - Today: User visits www.cnn.com via open wifi
  - Today: We inject <img src=”http://mail.google.com/mail">
  - Today: Browser transmits unprotected gmail GX cookie for http image fetch
  - Today: We sniff cookies, write them to cookies.txt
  - Tomorrow: Use cookies.txt to read their mail
- User never even checks gmail on hostile network!

Vectors

- Not just open wifi
- ARP poisoning
- DHCP spoofing
- Dan Kaminsky's DNS Hijacking Attack
- DSL+Cable modem networks?
  - Possible to sniff+inject on cable networks?
    - Sometimes DOCSIS encryption, but many modes are weak
  - May require two modems
    - One custom with TX/RX frequencies switched
    - Or custom software modem! (Guy Martin's talk)

'Manual' Attack

- Aka: How people were owned for the past 365 days.
- Fire up wireshark
- Fire up airpwn/netsed with custom rule
- Copy cookies out of wireshark.
- Lame.

Introducing CookieMonster

Fully automated pylorcon tool for cookie gathering

- Caches DNS responses
- Listens for 443 connections
  - Uses cache to map IP to domain name
- Stores IP+host into injection queue
- Next time IP connects to ANY http website:
  - Inject <img src=”http://dnsname”>
- Gathers any resulting cookies and writes cookies.txt file for use in Firefox 2

Ok, so there is some configuration..

- Need cookie path for injection for some sites
  - No worries. List of paths for popular sites provided!
- Might want to steal other non-ssl sites too
  - No worries. Additional target list can be provided!

Feed the Monster Some COokies!1!

Much Better

Bonus: (>?)40% of Internet's Gmail!

1. Search for 'CAU metasploit DNS hijack'
2. Scan for vulnerable DNS servers (>40% of net)
3. Hijack *.google.com to point to your transproxy
4. Inject http://mail.google.com imgs into www.google.com welcome page
5. Modify CookieMonster to only passively collect cookies at your IP (2 line change)
6. ???
7. PROFIT!

How to Protect Yourself Now

- Use ForceHTTPS Firefox addon (complicated)
- Use Gmail HTTPS pref (if available)
- Log out when done
- Clear cookies regularly

Thanks

- Damon McCoy for additional cards+headers
- Colin Jackson for ForceHTTPS and other work
- Nick Weaver for suggestions and correspondence
- LORCON, pylorcon, dpkt teams/authors