Protecting Privacy with Alternating IP Addresses

Curtis Taylor

Outline

- Introduction
- Background
- Attempted Solutions
- Actual Solution (and tools used)
- Demo
- Implications of Approach
- Future Work
- Conclusion

Introduction 1

- Eckersley discussed the ability to distinguish machines behind a single IP address, even if those machines block cookies entirely.
- Yen *et al.* wanted to determine how much information was revealed or could be discovered about a user through identifiers such as browser information (user-agent string - UA), IP address, cookies, and user login IDs
  - found that 60% to 70% of HTTP user-agent strings alone can identify hosts, but if combined with the IP address it can be improved to 80%.
- “My computer always has the same IP address” – Professor Wills

Introduction 2

- Gruteser and Grunwald suggested a method of alternating MAC addresses to help protect location privacy

- Casado and M. Freedman found that although DHCP is deployed by many ISPs, 75% of users retained the same IP over a 2 week study.
  - Eckersley conducted a more recent study over a period of 3 weeks that found 95% retained the same IP address

Background

- Linux routes via “route” command
  - Default routes
  - Longest prefix matching
- IP Aliases – eth0:0->eth0:1->…->eth0:x
  - interfaces file
- Perl scripts!

Attempted Solutions

- DHCP
  - – Failed due to routes and subnet mismatches
- DHCP and NAT
  - – Attempt to overcome routing issues
  - – Which is consulted first? Unsure

Actual Solution 1

- Change Linux IP address aliases
  - By manually editing: /etc/network/interfaces via Perl script
  - Addresses are not “random”; out of the scope
- Default route changes after *x* (15s) seconds
- Active connections are given direct route
- IP addresses don’t change until all aliases have been used
  - Unfair to connections from the last alias

Actual Solution 2

- Firefox Plugin
  - Uses events to determine if tab changed or new window created then writes the address to file
  - File is cleared each time the default route is changed
  - If an address doesn't appear in the browser after *y* seconds (60s), direct route is removed

Actual Solution 3

Example interfaces file:

```
auto eth0:0
iface eth0:0 inet static
 address 10.16.16.4
 netmask 255.255.0.0
 network 10.16.0.0
 broadcast 10.16.16.255
 gateway 10.16.1.1
…
auto eth0:121
 iface eth0:0 inet static
 address 10.16.16.125
 netmask 255.255.0.0
 network 10.16.0.0
 broadcast 10.16.16.255
 gateway 10.16.1.1
```

Example Perl script to update route

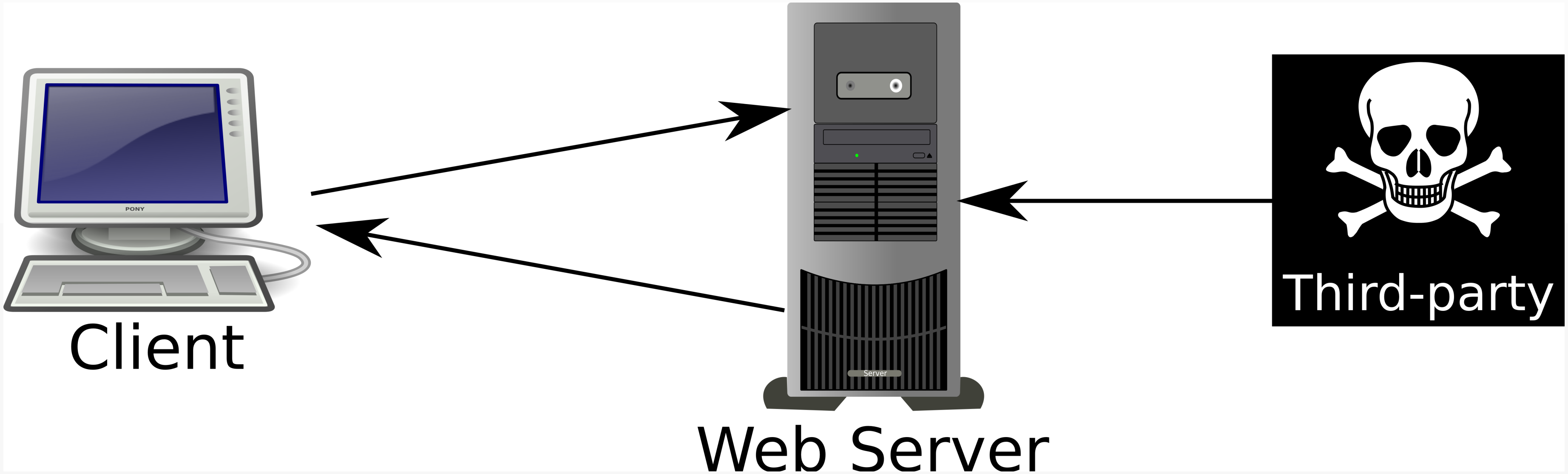
```
`route add -net 10.16.0.0 netmask 255.255.0.0 eth0:$eth`;
`route add default gw 10.16.1.1 metric 0 eth0:$eth`;
`route del -net 10.16.0.0 netmask 255.255.0.0 eth0:$eth`;
`route del default gw 10.16.1.1 metric 0 eth0:$ethToDel`;
            $eth+=1;
```

New routing table

```
$bash: route -n
Kernel IP routing table
Destination  Gateway   Genmask   Flags Metric Ref    Use Iface
   0.0.0.0     10.16.1.1   0.0.0.0     UG      0     0       0   eth0
```

Here, Iface displays the current interface for the default route. ***Notice** there is not alias number associated with it i.e., if the default route is using interface eth0:5, you cannot determine this from the routing table. This was a point of concern during development as to which alias was the default.

Demo!



Implications of Approach

- Not tested with other protocols
  - Assumed to fail
  - Sys admins would hate it
    - Logging
    - Possible new vulnerabilities are unknown
- If transfer isn’t complete at moment of IP address change, connection is broken;
  this is solvable via netstat

Future Work

- netstat for maintaining active connections
  - Attack vector via third-party?
- Tests need to be conducted in a real environment
  - WPI, please give me a block of public addresses
- Tie default route change into DNS
  - E.g., use Snort to catch DNS requests
- Actually use DHCP

Conclusion

- Allows user to appear to be coming from different IP addresses
- Allows you to maintain connection to first party for downloading large files
- Simple application
  - Uses all built-in, enterprise quality programs
- You may have cleverly noticed that my approach has a flaw that completely undermines my goal

Referenced Work

- B. Krishnamurthy, K. Naryshkin, and C. E. Wills, “Privacy leakage vs. Protection measures: the growing disconnect,” Web 2.0 Security and Privacy Workshop, May 2011.
- M. Gruteser and D. Grunwald, “Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis,” *Mob. Netw. Appl.*, vol. 10, no. 3, pp. 315–325, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1007/s11036- 005- 6425- 1
- T.-F.Yen,Y.Xie,F.Yu,R.P.Yu,andM.Abadi,“Hostfingerprintingand tracking on the web:privacy and security implications,” The 19th Annual Network and Distributed System Security Symposium (NDSS), February 2012.
- P. Eckersley, “How unique is your web browser?” in *Proceedings of the 10th international conference on Privacy enhancing technologies*, ser. PETS’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 1–18. [Online]. Available: http://dl.acm.org/citation.cfm?id=1881151.1881152
- M. Casado and M. Freedman. Peering through the shroud: The effect of edge opacity on IP-based client identification. In NSDI, April 2007.