IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

# Black Ops 2007: Design Reviewing The Web

Dan Kaminsky
Director of Penetration Testing
IOActive Inc.

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

# Three Interesting Things

- Slirpie:  Come to my website, be my VPN
- P0wf: Automagically discovering the toolkits behind the web
- LudiVu: Pretty

# Intro to Slirpie: Dependence And Otherwise

- The fundamental design of the web is *late binding* – pieces are pulled together and assembled at runtime, independently from one another
  - As soon as independence was established, people wanted to be able to create dependencies
    - You read my page, I read your mail
    - Could be problematic ☺

# The Same Origin Policy

- Basic concept
  - Independent resources (images, self-contained iframes, etc) can load across security domains
  - Dependent resources (scripts, etc) can only be dependent on eachother when they're hosted from the same origin
    - A page can read from an iframe it gives you, but not an iframe Hotmail gives you

# The Obvious Bug

- Content does not come from names
  - Content comes from addresses
- DNS provides the name to address mapping
- The presumption was that this mapping would stay the same
  - Wrong

# DNS Pinning

- Swapping around the DNS address (“DNS Rebinding”) has been known for years
- DNS Pinning, implemented in browsers, has attempted to lock the browser to one particular address
- Old attack, old defense, nobody checked to see if it still worked…until recently
  - RSnake
  - Dan Boneh from Stanford

# New Era of DNS Rebinding Attacks

- Browsers only *try* to pin DNS – they fail open rather than closed
  - More reliable that way
- The real problem is plugins, which can make connections of their own
  - Plugins don’t share the pin cache with the browser – can load the applet from one address and deliver traffic to another

# Plug and Play

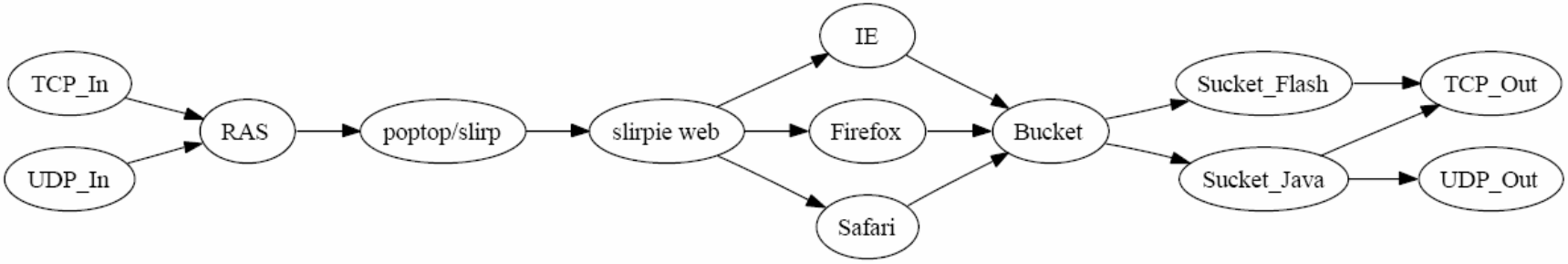
- What did RSnake and Dan Boneh's team find?
  - Browser itself will provide arbitrary HTTP
  - XMLHTTPRequest provides "crippled TCP"
  - Flash9 provides arbitrary TCP sockets
  - Java provides arbitrary TCP and UDP sockets
- Everything's supposed to be bound to the site that provided the applet
  - Doesn't work very well

# What can we do with this?

- Some people don't see the significance of this attack
  - Every once in a while, you really have to demonstrate the problem
  - This is going to be hideous to fix – lots of people need to work together – meaning this is the sort of thing that really needs a demo
  - OK this is a fun one to write

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

# Slirpie: The Browser VPN Concentrator




Design in a nutshell:

Applications generate streams of data, which are sent to **sockets**.

Sockets are consumed by RAS, and turned into a stream of **packets.**

Packets are consumed by poptop (a PPTP daemon), and given to SLIRP, which converts them back into **streams**.

Streams are consumed by slirpie (a web server), and sent to any one of the major browsers. Each builds a page with Ajax, called a **bucket**, that creates any number of socket providers, or **suckets**, which ultimately send the data along.

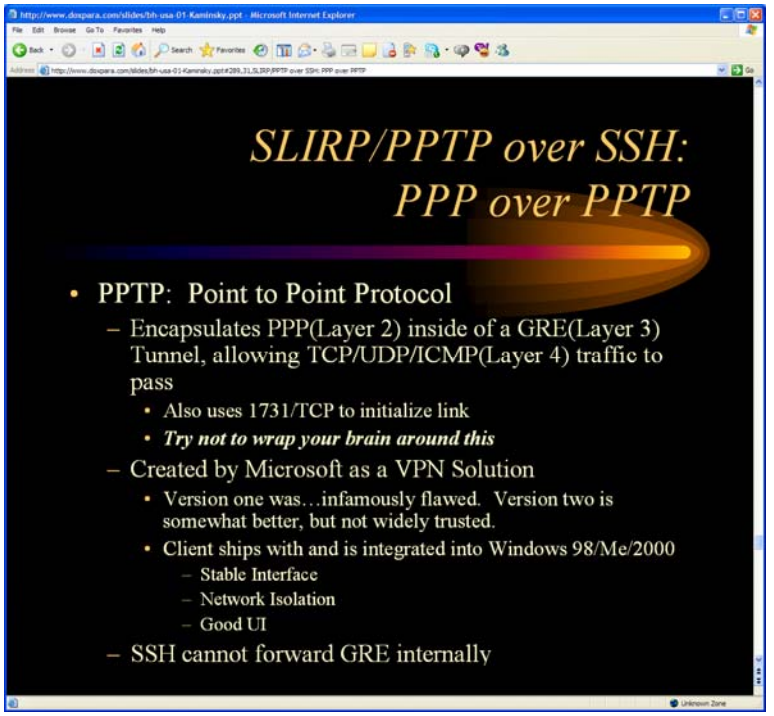
# History

- 1996: Slirp becomes popular
  - Converted **shell** accounts to **PPP** accounts
  - "Userspace NAT" – turned **packets** into **streams** for **sockets**
  - Less resource intensive for ISPs to support than to run apps locally

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

# History [1]

- 2001: PPTP over SSH
  - PoPToP is the Linux PPTP server
  - Uses external PPP provider
  - Slirp could be that provider
  - Slirp *over SSH* could be that provider



# Six Years Later...

- Slirp turned packets into streams, then streams into sockets
  - We take the streams…and hand them to something else entirely.

# Slirpie Design

- Slirpie keeps a list of streams waiting to be completed in a remote browser
  - Given by slirp
- Browser arrives and receives an AJAX page (“Bucket”)
  - Requests list of all unique IP addresses that packets need to be delivered to

# The DNS Two-Step

- Remember, we can't spawn traffic directly to these IP addresses – they have to always be coming to our name
  - But we can use many subdomains
  - We can encode the desired address in the name
  - We only need to provide our own address once
    - We need to provide our proxy applet (the "sucket")
- What to do?
  - For each IP address, register intent to create sucket. Then create iframes to a.b.c.d.notmallory.com, with a.b.c.d representing the IPv4 quad.
  - The registration will cause DNS for notmallory.com to still return the real address for notmallory.com. This will only happen once, though.

# Duke Suckets

- In each IFrame, an applet lives
- When it spawns, it requests *via the Javascript bridge* a list of ports and protocols to create connections to
  - This lets it use the browsers pin cache…when it wants to ☺
  - For each successful connection, it starts proxying traffic between the connection and slirpie, using standard HTTP tunnel mechanics
- Unique sucket per IP, not per port
  - One sucket can service many sockets.
- Should destroy suckets when no longer needed – have to watch efficiency

# Other Tricks

- P0wf: Passive OS Web Fingerprinting
  - Based on p0f – Passive OS Fingerprinter by Zalewski
- Most websites are made through template engines
  - Template engines provide more uniquely recognizable bits than we ever had in the stack
    - TCP/IP far more standardized than HTML
    - Template based websites are parsing far more weirdness than TCP/IP ever did

# Fingerprintable Elements in HTML

- Obvious choices
  - Filenames (especially included scripts)
  - Cookie formats
  - URL formats
  - RPC formats (for AJAX)
  - Function names
- Less obvious choices
  - Script and HTML formatting
  - Comment content
  - Validation failures
  - Prominent errors
  - Page Graph

# Page Graph?

- The DOM represents a Directed Graph
- Graph branches can be m-to-n compared reasonably effectively
- Depth and nature of template engines forms a fingerprint