Notice!

Updated presentation materials are available online at:

http://www.wiretrip.net/rfp/blackhat-vegas/

Breaking the silence: new toys in the works

A look at the tools in development by rfp.labs

rain forest puppy
rfp@wiretrip.net

What the hell has rfp.labs been doing lately?

- Nothing! (or so it seems)
- Identifying needs/areas that lack adequate tools
- Defining new goals to shoot for
- Trying to catch up on all the vulnerabilities and technological advances (almost impossible!)
- Rumors of whisker 2.0, some funky library thing, and a proxy…

Tools on the drawing board

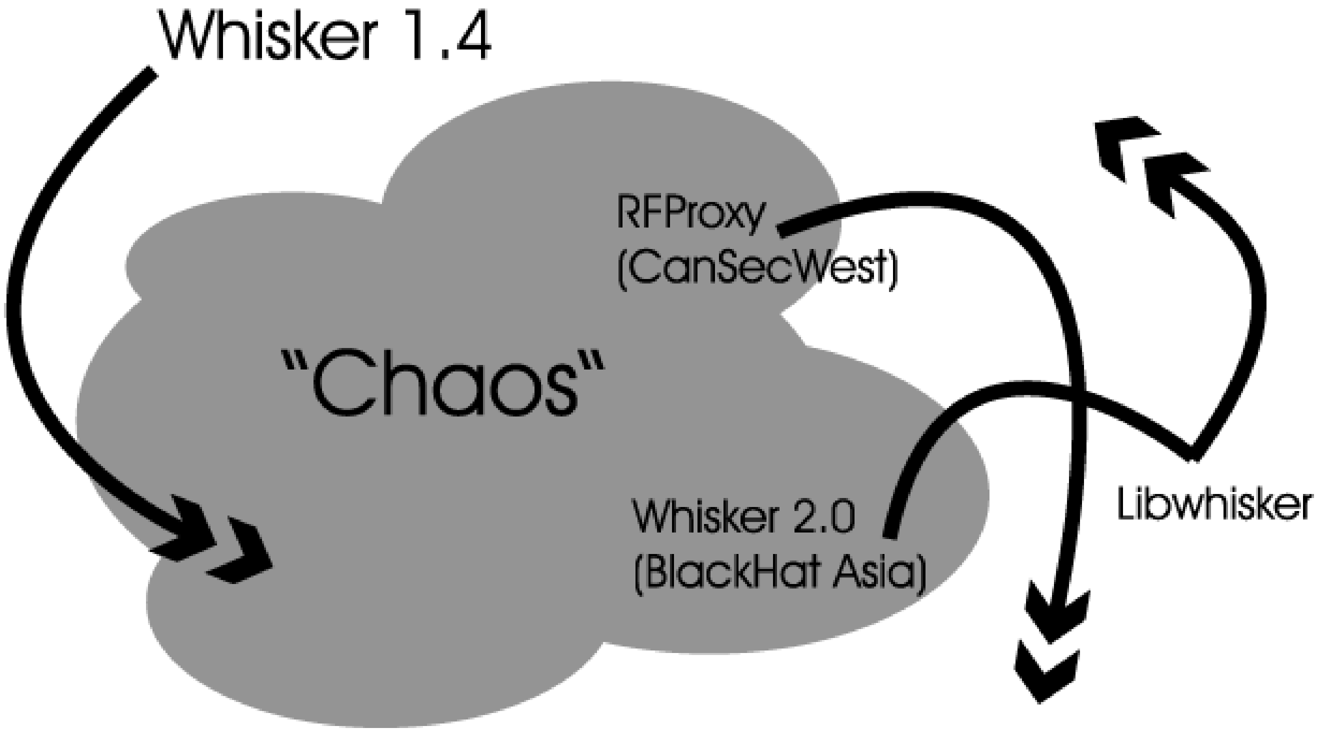
RFProxy

libwhisker

whisker 2.0

????

The confusing roadmap



RFProxy: where it’s at

- Preview Release 1 was released in March
- Basic framework and functionality implemented
  (although very sloppy)
- Given to the community on an evaluation basis
  (I.e. would this be useful?)
- Future was uncertain…

RFProxy: where it’s going

- It has a future!
- Will be ported to use libwhisker for most of it’s functionality
- Will be designed not only as a proxy, but an interface/platform to other rfp.labs standalone tools

libwhisker: where it’s at

- Preview Release 3 was made available in late June
- A well-documented and full-featured library of components useful to assessing web applications
- Core HTTP functionality (with SSL support) solid
- Seen a little (although not as much as hoped for) use by the community
- Much more functionality is currently in unreleased alpha stage
- Attempting to get some of the fundamental hurdles out of the way

libwhisker: where it’s going

- Go as far as I can take it!
- Hopefully will have more contribution as it has time to work itself into the community
- The functionality of whisker is (err, will be) in libwhisker
- Intend to move rapidly to an official 1.0 release

Peak into libwhisker: HTTP module

- The core of libwhisker, passes a request to a server and receives the response
- Capable of handling HTTP 0.9, 1.0, and 1.1 connections
- HTTP 1.1 keep-alive/connection reuse
- Receive chunked encoding
- Full integrated HTTP proxy and virtual host support
- Complete cross-platform timeout support
- Transparent data handling (PUT, POST, etc)
- Handles HTTP ‘100 Continue’ responses

HTTP module cont.

- Native SSL support via Net::SSL (Crypt::SSLeay) or Net::SSLeay
- All aspects are completely customizable for ultimate control (“no rules” design concept)
- Controlled by a single structure (Perl hash)

Other libwhisker modules

- Auth: basic, MD5*, and NTLM* authentication support routines
- BruteURL: brute forcing support
- HTML: HTML tag parsing routines
- Encode: various encoding routines (hex, Unicode*, etc)
- Crawl: site/link crawling
- DAV: WebDAV request wrappers
- FrontPage*: MS FrontPage client emulation
- IDS*: anti-IDS routines (new methods; however effectiveness is still untested)

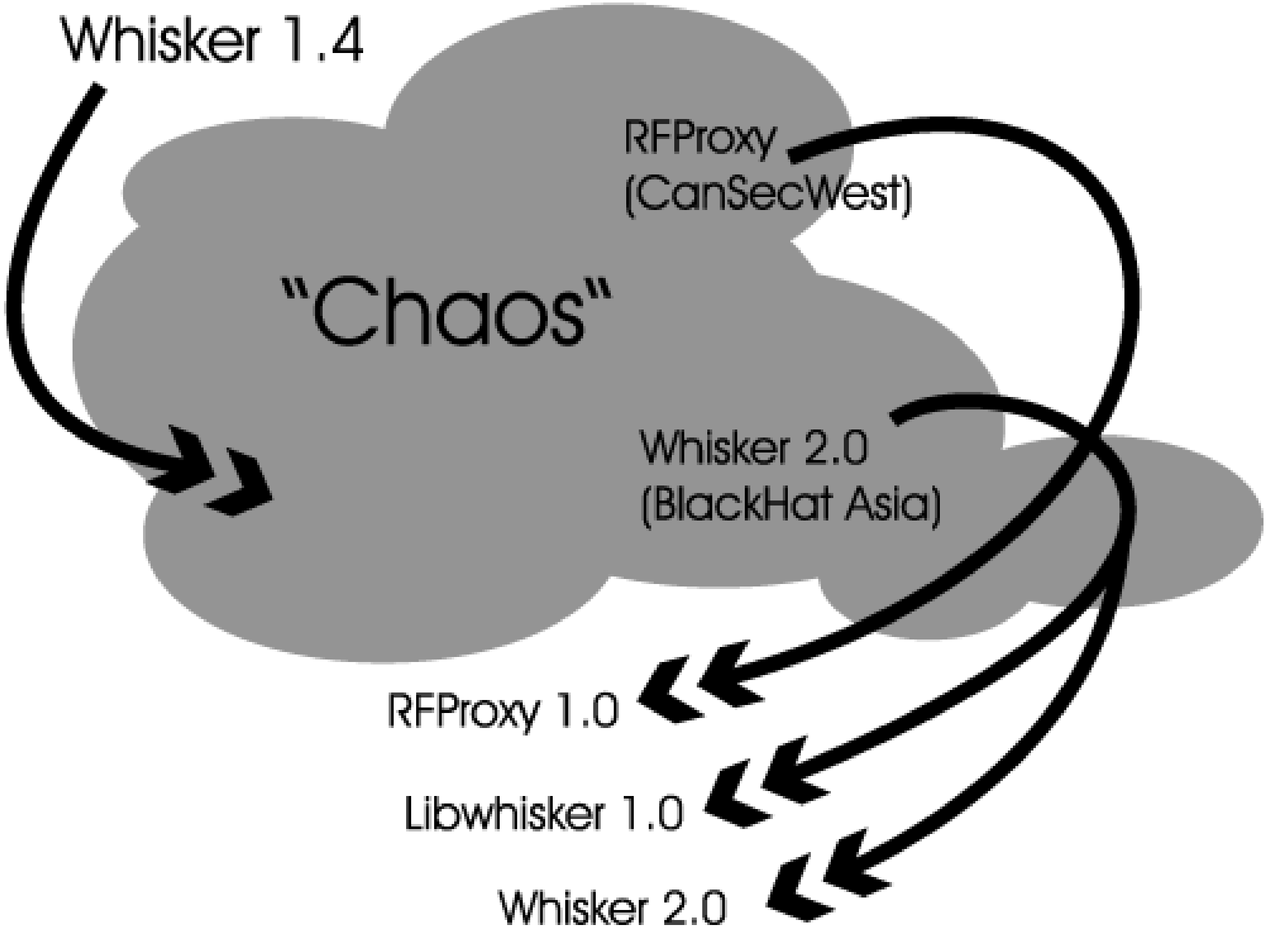
whisker 2.0: where it’s at

- More like “where is it?”
- Plan was to only have libwhisker, and have scan logic coded directly in Perl
- General purpose example tools would be included with libwhisker that would illustrate the API
- However many people have complained this limited overall usability

whisker 2.0: where it’s going

- Separated from libwhisker!
- A standalone tool, in whisker fashion, build on libwhisker
- However, libwhisker needs to get to a prominent point before whisker 2.0 can be built on top
- Coding has begun! But PR1 is still a ways off…

The new roadmap



When can I play?

Today! The current releases of RFProxy and libwhisker are always available on my website at:
http://www.wiretrip.net/rfp/

People interested in keeping track of or getting involved in libwhisker/whisker (and some RFProxy) development can hop onto the whisker-devel mailing list hosted by SourceForge. Sign up at:
http://sourceforge.net/projects/whisker/

Questions

Thanks

rain forest puppy
rfp@wiretrip.net