Layering In Defense

“Securing” Web Systems

Erik Berls

LayerOne

April 23, 2005

Layering in defense:

- Disparate software
- Operates on different layers
- Isolation zones
  - You mean DMZs?

Why?

Why?

Why?

Why?

Why?

Why?

Why?

Why?

Why Goals and Non-Goals

- What is the problem we are trying to solve?
- What are we NOT trying to solve?

Existing Art?

- Comparison to the smtpd model
  - Mail beginnings
  - Why smtpd came to pass
  - Mail evolved

What is going to do for me?

- Business Needs
- Technical Needs
- Security Needs

Business Needs

- Control release process
- Limit beta testing to specific groups
- Administrative domain controls
  - Different domains on the same server
  - Different domain for the control point

Technical needs

- Off server control of web presence
- Non-overrideable access by application owners
- Choke point control for applications
- Flexibility in taking applications offline quickly
  - Either partial, specific to client, very flexible

Security Needs

- Folds back on both Technical and Business
  - A tool to better manage risk

Common avenues of attack

- What does the firewall protect
  - Filtering inbound
  - Filtering outbound
  - Packet level
  - Protocol level
- What does the web server protect
- What does the application "protect”

Addressing these needs

- Architecture
- What we can deploy

Architecture Greenfield

- routers
- firewalls
- loadbalancers
- choke points
- firewalls
- web servers
- firewalls
- middleware servers
- firewalls
- database servers

Reality (budget) sets in

- firewall
- choke point
- web server
- backend servers

Choke point?

- Squid!
- Note:
  - squid is NOT an application firewall

Using squid as the choke point

- "Accelerator mode”
- What can it do?
- What can it not do?
  - What does squid NOT grant (at this time?)

Protecting the front end

- Another piece of software to exposure to the cold, cruel, Internet
- Risks
  - Vulnerability can occur in squid or backend
- Advantages
  - squid is lighter than apache or a servelet engine
  - Well defined set of operations

Building the system

- OS Specifics
- Software Specifics

Core OS : NetBSD

- chroot
- systrace
- veriexec
- Non-executable stack

Proxy Layer Configuration

ACLs

```
acl demo1_block dst 63.201.53.5
acl demo1_block dst proto HTTP
acl demo1_block url_regex "^http://demo1.worst.com/no.html$"

acl demo1 dst 63.201.53.5
acl demo1 dst proto HTTP
acl demo1 url_regex "^http://demo1.worst.com/"

acl all dst 0.0.0.0/0.0.0.0

http_access deny demo1_block
deny_info ERR_error_demo1 demo1_block
http_access allow demo1
http_access deny all
```

Squid Proxy

- Expert Knowledge is Vital!

Future directions with squid

- Realtime log analysis
- Dynamic rule creation
- Realtime URL processing
  - This is trickier than it sounds!

Cost / Benefit summary

Benifits

- extra layer to control access
- ability to control access to application independ of application owners
- tightly controllable front end
- cross domain logs processing

Caveats

- additional system to administer
- doesn't protect against application attacks
- requires someone on staff who understands squid

More Information

Squid: http://www.squid-cache.org

NetBSD: http://www.netbsd.org

chroot, ipfilter, systrace, & veriexec are available as man pages under NetBSD.

Questions?