hack 2002

rain forest puppy / wiretrip / rfp.labs
rfp@wiretrip.net

RAIN FOREST PUPPY

HACK 2002

Security is a war as well as an art form: you need to be methodical and militant, but also creative and flexible.

- ancient rfp.labs proverb

Assessing the web:

When a firewall is not enough

Question:

What do all websites have in common?

Answer:

Unrestricted access
to port 80

Why target the web?

- Everyone is using it
- Safe bet the protocol will not become obsolete anytime soon
- New technology is being implemented/ retrofitted on top (e.g. SOAP, WebDAV)
- It’s everywhere! Mobile phones, cars, watches, toasters…
- Protocol fundamentally not suited to do a majority of what it’s doing today

Problems with HTTP

- Multitude of involved technologies
- The involved protocols are extremely simple; therefore, it is easy to (mis)code services
- Lack of experience coding public-service, multi-user applications
- Stateless nature makes transaction-based systems tricky

Sources of the problem

- Unskilled/robotic programmers (“code mills”)
- Lack of security-oriented programming training
- Tendancy to ‘code now, fix later’
- Current tools make it very easy to code insecurely
- Misconceptions about what ‘security’ really involves

Common web vulnerability areas

Buffer overflows

- Classic bug that's been exploited for quite a while
- Lack of bounds-checking is really poor programming practice
- Can exist in the web server, application server, database server, or the CGI programs
- Fortunately it's a well-advertised problem
- Many scripting languages (ASP, PHP, Perl, .NET, etc) are generally immune, as they have auto-expanding elements

Cross-site scripting

- Reprinting user data without filtering it for web-specific characters
- Potential to trick users into executing javascript in vulnerable site's context
- Partly a 'social engineering' technique
- More of a liability than a vulnerability—it's a way to hack the users, not the server

SQL tampering

- Web server already has/allows access to the database server
- Attacker can cause arbitrary SQL to be executed
- Results vary from data exposure to full system compromise
- Does not require direct database access!
- Many applications are vulnerable
- Stems from CGIs/scripts making assumptions about user input and not double-checking/filtering

File includes

- It’s common for a CGI to open and display/manipulate the contents of a file on the server
- If the filename is composed of user-supplied elements, an attacker may be able to trick the server into opening another file
- Can lead to info disclosure or script/command execution

Authentication weaknesses

- CGIs can fail to check credentials with every request
- Thus you bypass the login script and directly access the following scripts, without needing username/password
- Or, certain actions/functions may not check for the proper authentication

Weak session mechanism

- The session/state mechanism uses predictable token IDs
- Or, the ID keyspace is too small for the number of users
- Either way, an attacker can ‘guess’ a valid token and hijack the session

Other vulnerability areas

- Format strings, signed conversion, double-free, …
- All tend to be limited to low(er)-level languages

Bugs, bugs, everywhere bugs

Finding the bugs in your site

- Best place to start is with a vulnerability scanner: Nessus, ISS Scanner, etc.
- Or, you can use a web-centric scanner: Nitko, whisker
- Goal is to identify as many known problems as can be found
- Also be concerned with what configuration information these tools find: server banners, software versions, etc.
- However, none of these will help when it comes to your custom CGI applications…

Useful free tools

- NMAP

  http://www.insecure.org/nmap/nmap_download.html

- Nikto

  http://www.cirt.net/

- Whisker 2.0

  http://www.wiretrip.net/rfp/

- Nessus

  http://www.nessus.org/

Custom scripts

- Checking custom CGIs/applications is not as easy as running a scanner…
- Automated tools are unaware of how to interoperate with your site, applications, forms, etc.
- Therefore, you will typically need to involve a human in the review/analysis process
- This can be a drawback, as you will need someone who is skilled enough in web vulnerabilities to make sure they know what they are doing/looking for

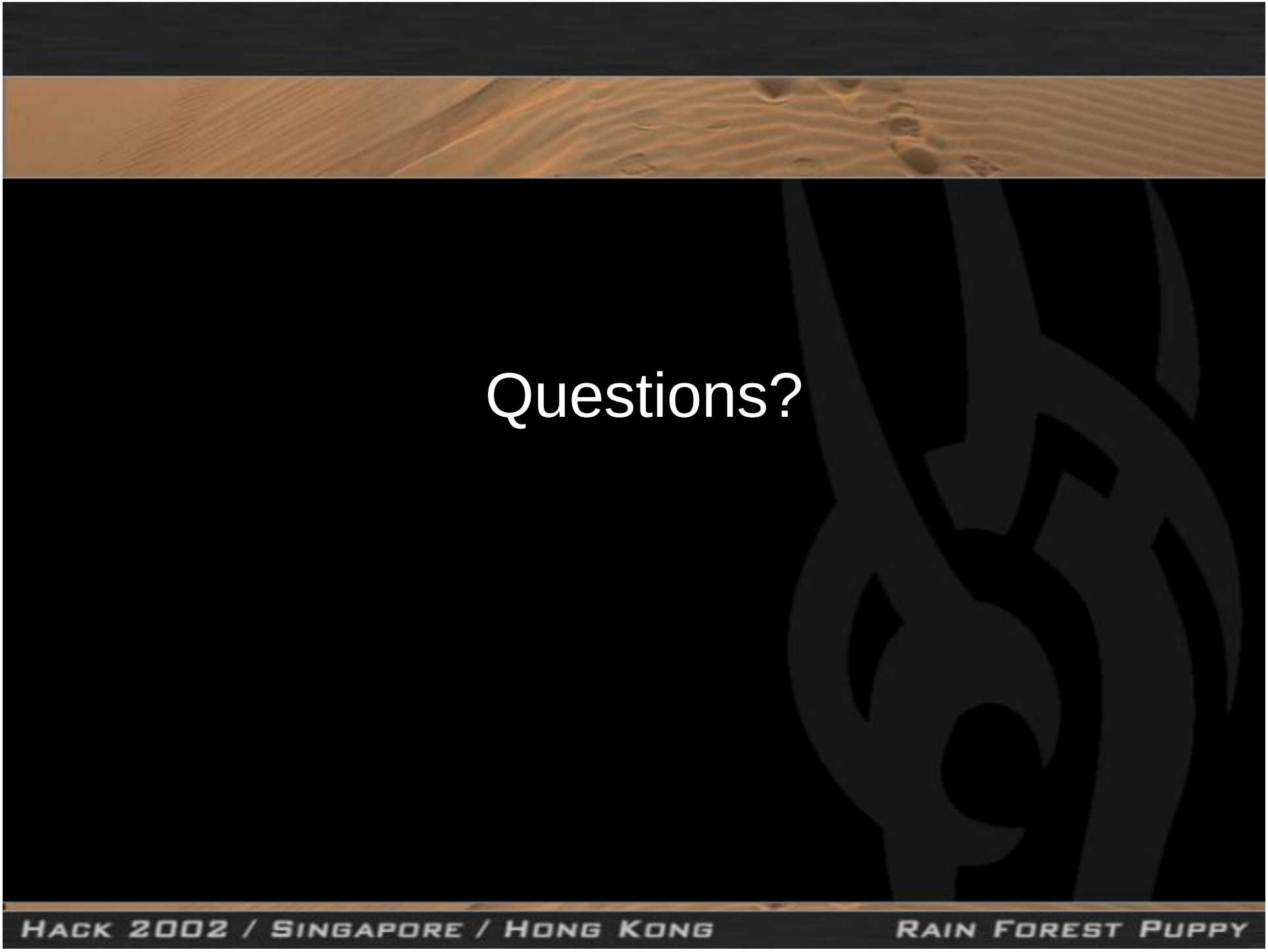
Custom analysis tools

- WHArsenal
- WebSleuth
- @Stake WebProxy
- AppScan

Looking by hand

- SQL tampering: insert ‘ into dynamic parameters
- CSS: insert <TEST> into form fields
- Buffer overflows: submit large amounts of data
- Try to directly access all CGIs (without authentication)
- Attempt to gather sequential session ID tokens and look for a pattern
- Look at the source code itself

All of this is easier said than done…


Questions?
HACK 2002 / SINGAPORE / HONG KONG
RAIN FOREST PUPPY

Protecting your webservers

Available protections

- Historical record---what problems have others had?
- Code reviews
- Third-party pen tests and assessments

Keep in mind pen-tests may not find all bugs!

Commercial protections

- Sanctum AppShield
- KaVaDo Interdo
- eEye SecureIIS

However, you can do a lot on your own…

Tip #1:

Want to get
rid of IIS?

ASP HTTP Servers

- You’re not stuck with IIS for your ASP needs
- iASP

  http://www.halcyonsoft.com/products/iasp.asp
- Sun Chili!ASP

  http://www.chilisoft.com/chiliasp/

Tip #2

Securely configure the HTTP server

IIS

- Remove unused virtual dirs
- Clean out your /scripts/ dir
- Lower your connection timeout
- Remove unused app mappings
- Disable parent paths
- Don’t send detailed error messages
- Disable NTLM auth if allowing anonymous access
- Put content on a different drive

Apache

- Remove unused modules

  Minimal: access, actions, alias, autoindex, cgi, dir, env, log_config, mime, setenvif, so

- Double check Alias and ScriptAlias
- Be wary of MultiViews
- Double check /cgi-bin/ for samples
- Remove /cgi-bin/ if not used

Netscape

- Double check all ‘pfx2dir’ and ‘assign-name’ NameTrans in obj.conf
- Netware folk: put content on different volume

PHP

- Tweak php.ini values

  register_globals = Off
  safe_mode = On
  expose_php = Off
  display_errors = Off
  file_uploads = Off
  allow_url_fopen = Off

- No register_globals!!

Anti-CSS

- You can prevent cross-site scripting on a server level!
- Apache: mod_rewrite configuration in httpd.conf:

```
RewriteEngine on
RewriteCond %{QUERY_STRING} [<>]
RewriteRule ^/.* - [F]
```

- Netscape: use a custom filter
- IIS: Eeye SecureIIS will allow you to define a filter; Microsoft URLScan does not appear to be capable

Tip #3

Hide your head(ers) in the sand

Apache header

- Search for “Server” in src/main/http_protocol.c
- Remove following line:

```
ap_send_header_field(r, “Server”,
        ap_get_server_version());
```

- Optionally, change it to something else:

```
ap_send_header_field(r, “Server”,
        “Microsoft-IIS/4.0”);
```

PHP header

- Search for _VERSION_HEADER in main/main.c
- Remove following lines:

```
if (PG(expose_PHP)) {
 sapi_add_header(
            SAPI_PHP_VERSION_HEADER,
            sizeof(SAPI_PHP_VERSION_HEADER)
            -1,1);
}
```

- Or just set expose_php=Off

IIS header

- Use Microsoft's free UrlScan.exe ISAPI fitler
- In urlscan.ini set:

  RemoveServerHeader=0
- Optionally, change it by:

  AlternateServerName=Apache/1.3.23

Tip #4

Take advantage of kernel/system enhancements

Stackguard/Immunix

- Compiler that prevents buffer overflows and format string vulns
- Immunix is RedHat Linux compiled with Stackguard
- Attackers usually wind up crashing your app, rather than exploiting it to gain root
- Currently limited to x86 ELF arch
- Free from:

  http://www.immunix.org/

Solaris stack protection

- Solaris 2.6 and later come with built in buffer overflow protection—you just have to enable it
- Add the following to /etc/system:

```
set noexec_user_stack =1
set noexec_user_stack_log =1
```

- Not foolproof, but better than nothing

Access control systems

- LIDS – Linux patch

  http://www.lids.org/
- Engarde – RedHat with LIDS

  http://www.engardelinux.org/
- Argus Pitbull for Linux, AIX, Solaris

  http://www.argus-systems.com/
- Trusted Solaris, HP VirtualVault, NSA's SELinux

Bastille

- Perl lockdown scripts for various linux distros (RedHat and Mandrake) and HPUX
- Does everything: filesystem perms, daemon shutoff, service configs, etc.
- Interactive; undo option
- Free from:

  http://www.bastille-linux.org

Solaris ASET

- Solaris 8 includes basic lockdown scripts
- Allows three security levels
- Checks system file perms, user/group data, eeprom, and system environment
- Basically checks system against default installation/factory settings

Titan

- Various shell scripts to secure Solaris (beta does Linux and FreeBSD too)
- Does all kinds of system lockdown (over 60 various areas!)
- Free from:

  http://www.fish.com/titan/

Windows SCM

- Security Configuration Manager
- Optional on NT; comes with 2000 (Local Security Policy in Control Panel)
- Default policies provided for various network roles (DC, workstation, etc)

Tripwire

- Commercial and free versions
- Takes fingerprints of all your files, and does comparisons to detect changes
- Unix and Windows
- Free unix version from:

  http://sourceforge.net/projects/tripwire

AIDE

- Opensource Tripwire alternative
- Free from:

  http://www.cs.tut.fi/~rammer/aide.html

Tip #5

Use the source, Luke

ITS4

- Free for personal use (can't compete with author's code review services)
- C/C++ code review

- Free from:

  http://www.cigital.com/its4/

Flawfinder

- Free opensource Python script
- C/C++ code review
- Handles internationalized code

- Free from:

  http://www.dwheeler.com/flawfinder

RATS

- Opensource C program
- Understands C/C++, Python, Perl, and PHP

- Free from:

  http://www.securesw.com/rats/

FrontEnd Plus

- Disassembler for Java classes
- Basically a GUI around JAD

- Free from:

  http://kpdus.tripod.com/jad.html

Tip #6

Don’t depend on just your firewall

Windows firewalling

- Comes native with Windows NT/2000
- Look under ‘Options’ in the Advanced TCP/IP properties menu
- Let’s you only allow incoming traffic to certain ports
- Slightly klunky, but better than nothing
- Windows 2000 allows more options with IP Security Policies

Linux ipchains/iptables

- Comes native with Linux 2.2/2.4
- Highly flexible in letting you block or allow traffic based on various properties
- How to deny all incoming connections except to SSH and HTTP server:

```
ipchains -A input -p tcp --destination-port 80 \
        -j ACCEPT       # allow HTTP
ipchains -A input -p tcp --destination-port 22 \
        -j ACCEPT       # allow SSH
# Now we deny all other incoming SYNs
ipchains -A input -p tcp --syn -j DENY
```

Ipfilter

- Works with *BSD, Solaris, HPUX, IRIX
- Extremely flexible; has tons of features
- Deny traffic except to HTTP and SSH:

```
block in proto tcp from any to any flags S/S
pass in proto tcp from any to any port = 22 flags S
pass in proto tcp from any to any port = 80 flags S
```

Tip #7

Stay informed

Security mailing lists

- Bugtraq: vulnerability patch announcements and discussion

  http://www.securityfocus.com/

- VulnWatch: only the major stuff

  http://www.vulnwatch.org/

- Security Alert Consensus: pick your poison

  http://www.sans.org/

- Vendor mailing lists!

Other info

- SANS lockdown documents

  http://www.sans.org/

- Vendor guidelines (especially Microsoft)
- General internet (viva la Google)

Questions

rfp@wiretrip.net
http://www.wiretrip.net/rfp/