1) Install Squid (obviously), and Nmap. (This setup requires Ncat provided by Nmap)

(Make sure that Squid has been configured before proceeding. The below setup assumes that the checkbox for "Allow users on this interface" under Squid General Settings has been checked)

2) In the General section of the squid settings, under Custom Commands, put this is the "Custom ACLS (Before_Auth)" section.

(TTL is the time in seconds until a user must reauthenticate. Grace is the percentage of the TTL that Squid
will start trying to reauth, so as to get it out of the way before the timer runs out. Adjust as needed)

```
Code:
external_acl_type mauth children-startup=2 ttl=300 grace=50 %SRC /root/mauth
acl mauth external mauth
http_access allow mauth localnet
```

3) Under Diagnostics, go to Edit File. type the string "/root/mauth" under the file path, paste the below code
into it, change the $server and $port variables to something of your choosing, and then hit save.

(Server is the Windows computer that will run the Agent. ***Change this IP to the computer in which you plan to run the Agent.*** *Port is the TCP port to use. StripDomain dictates whether you'd like the domain passed back to Squid or stripped off)*

```
#!/usr/local/bin/php -q
<?php
  $server = "192.168.111.10";
  $port = 1234;
  $stripDomain = true;
  if (!defined(STDIN)) {
        define("STDIN", fopen("php://stdin", "r"));
  }
  while(!feof(STDIN)) {
    $user = "";
    $address = trim(fgets(STDIN));
    if(strlen($address) > 1) {
      exec("echo " . $address . " | /usr/pbi/bin/ncat " . $server . " " . $port,
$user);
      if(!$user || $user[0] == "E") {
        $user[0] = "ErrNoUserF";
      }
      if($stripDomain) {
        if(strstr($user[0], '\\', true)) {
          $user[0] = substr($user[0], strpos($user[0], '\\')+1, strle
n($user[0]));
        }
      }
      echo "OK user=" . $user[0] . "\n";
    }
  }
?>
```

4) Go to Diagnostics, then Command Prompt. Under Command, execute the command "chmod 755 /root/mauth"

5) On your Windows server, paste the below code into a text file and name it mauth.ps1.

(Port is the TCP port to listen on. ***Make sure this is the same port that the Helper has specified****. Timeout is how long before WMI gives up the attempt at querying for a username. Change as necessary)*

```
$port = 1234
$timeout = 1000
$endpoint = new-object System.Net.IPEndPoint ([system.net.ipaddress]::any,
$port)
$listener = new-object System.Net.Sockets.TcpListener $endpoint
$listener.start()
do {

  write-host __________ Listening on port $port __________  -backgroundcolor
"blue"
  $client = $listener.AcceptTcpClient()
  $stream = $client.GetStream();

  $reader = New-Object System.IO.StreamReader $stream
  $writer = New-Object System.IO.StreamWriter $stream
  $address = $reader.ReadLine()
  write-host Request from $address
  if($address) {
    try {
      $user = & wmic /Failfast:$timeout /Node:$address ComputerSystem Get
UserName
    } catch {
      $user[2] = "ErrNoUserW"
    }
  }
  if(([string]$user[2].length) -eq 0) {
    $user[2] = "ErrNoUserW"
  }

  write-host User returned is $user[2]
  $writer.Write($user[2] + "`n")
  $writer.flush()
  $client.close()
} while(1)
```

No Windows Server, executar no PowerShell:
`Set-ExecutionPolicy` **`Unrestricted`**

Setting as Unrestricted is the method I've been testing with. For the sake of simplicity, I also like to set it so that .ps1 files automatically open in the powershell exe, which is in "C:\Windows\System32\Windows Powershell".

That should be it. For the sake of testing, I found it helpful to SSH into the firewall, then run /root/mauth. From there, simply type an IP address and press enter. If everything works, you should get back an "OK user=JoeBob" (or whatever the user is). If you get back the user "ErrNoUserF", then the helper couldn't communicate with the Windows Agent. Check the firewall settings on the Agent computer and make sure the selected IP address is correct. If you get back the user "ErrNoUserW", then the Windows Agent failed to pull the currently logged in user via WMI. This could be caused by firewall rules on the workstation, no user being logged in, a request that took too long (timeout is set to 1000ms, adjust as needed), or a non-windows computer.

<Security Concern>
If you use this in a production network, it would be a really good idea to set the Windows Firewall (on the computer that runs the Agent) to only allow incoming connections on the TCP port that you choose from the pfSense firewall(s). This code does not validate the input string, and as such code injection is probably fairly easy.
</Security Concern>