“These are my Principals [sic]... if you don’t like them, I have others”

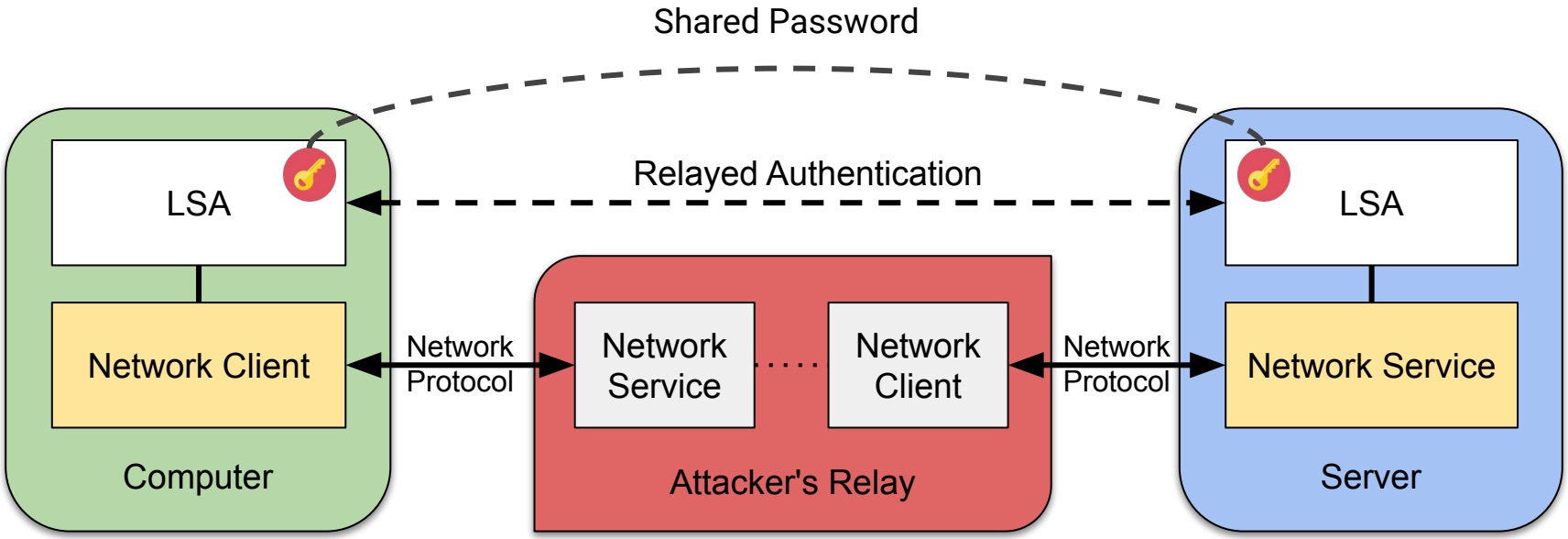
James Forshaw - Google Project Zero

OffensiveCon 2022

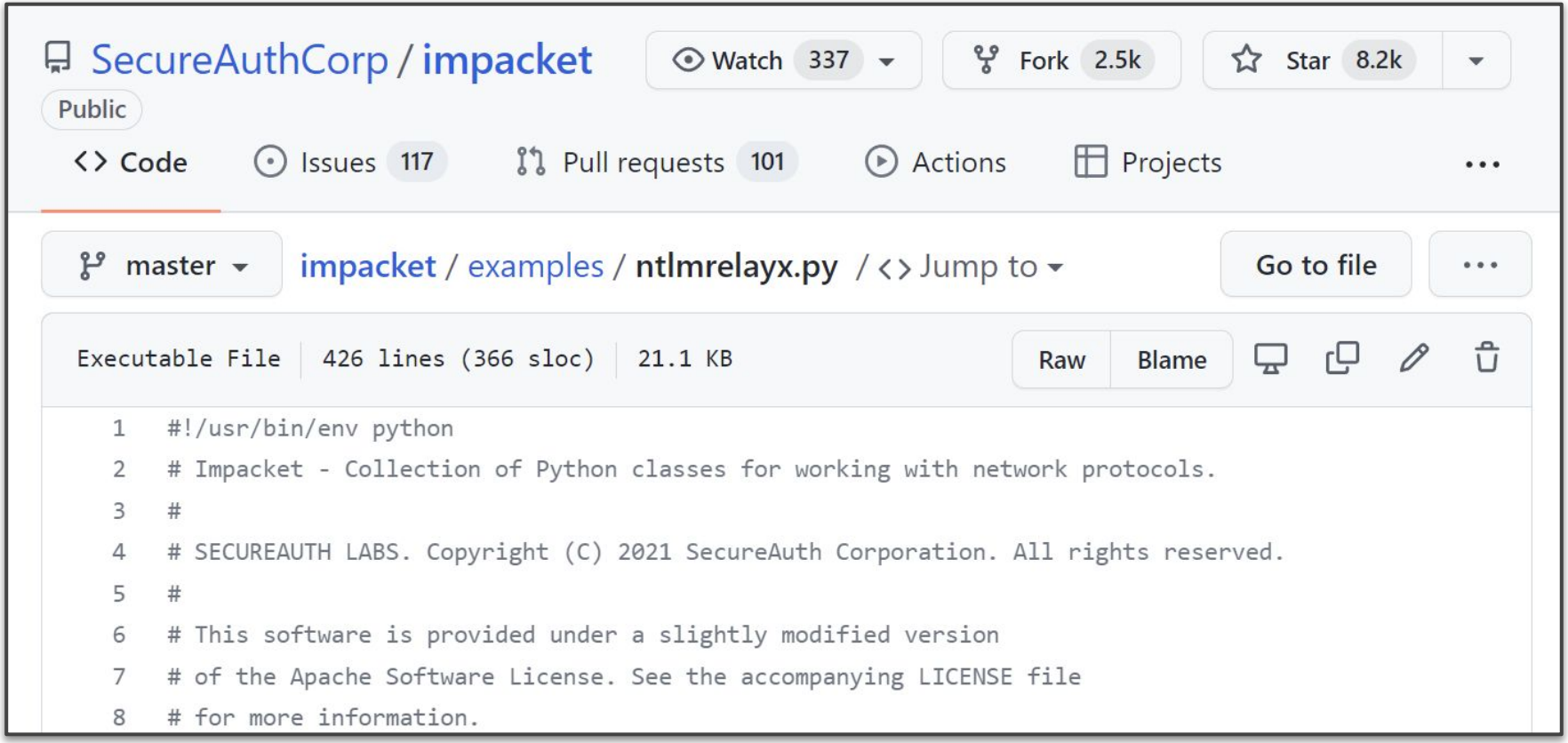
Agenda

- My research of relaying Kerberos authentication
- How protocols use Kerberos for authentication
- Demo exploit scenarios

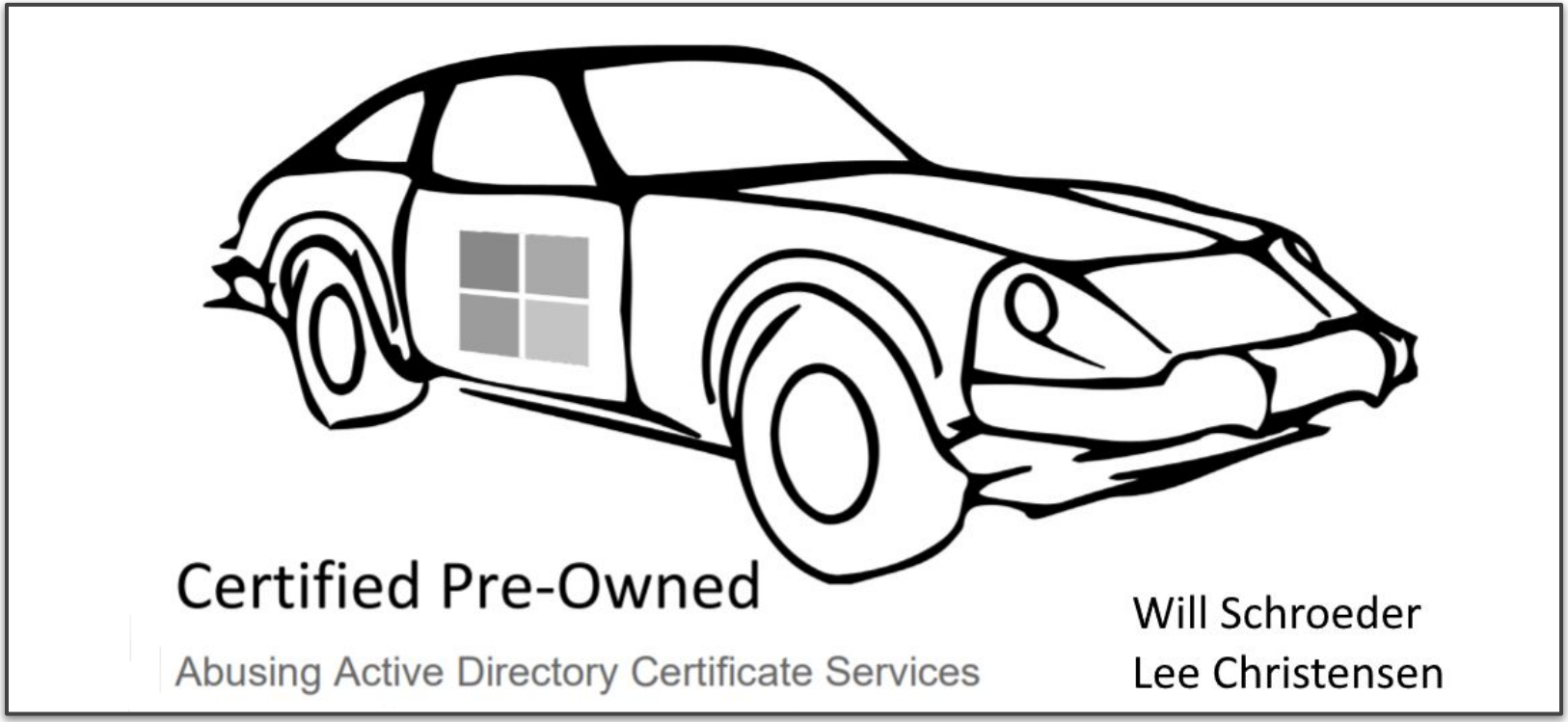
NTLM Authentication Relay



NTLMRelayX

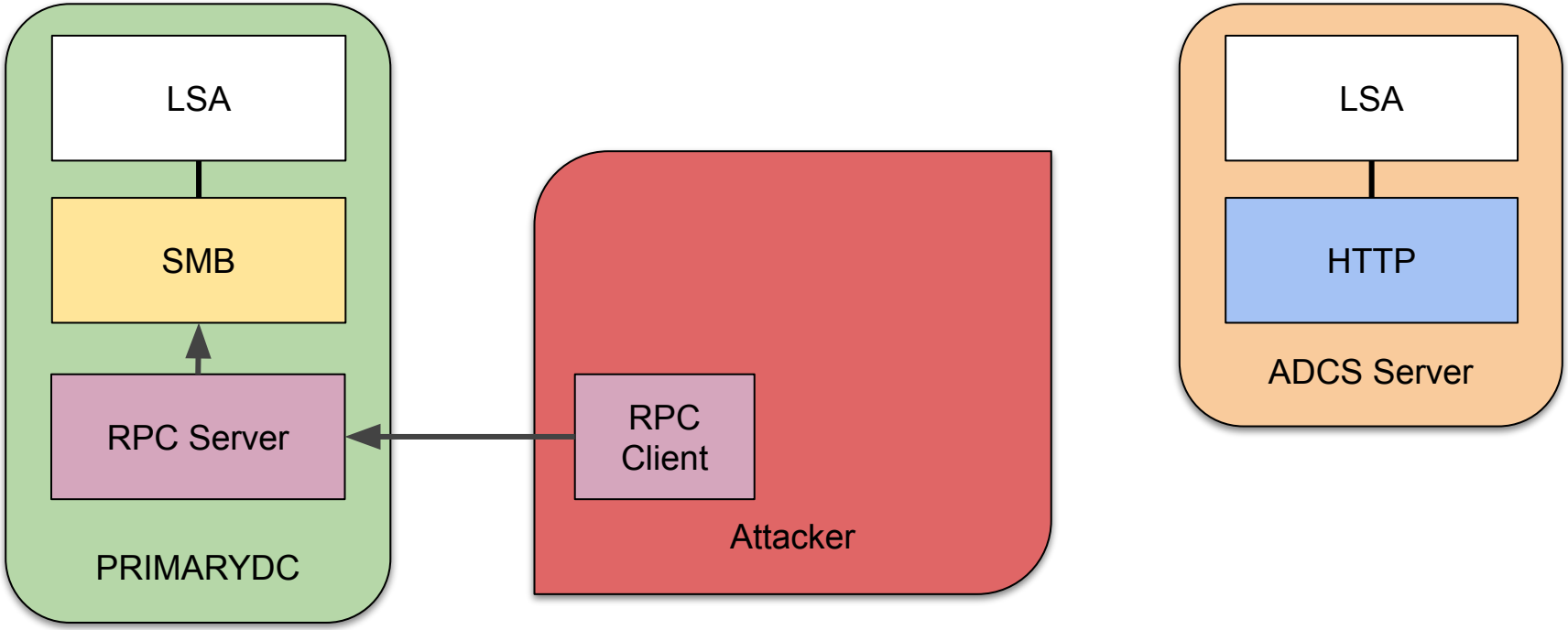

SecureAuthCorp / impacket
Watch 337
Fork 2.5k
Star 8.2k
Public
Code
Issues 117
Pull requests 101
Actions
Projects
master
impacket / examples / ntlmrelayx.py
Jump to
Go to file
Executable File
426 lines (366 sloc)
21.1 KB
Raw
Blame
1 #!/usr/bin/env python
2 # Impacket - Collection of Python classes for working with network protocols.
3 #
4 # SECUREAUTH LABS. Copyright (C) 2021 SecureAuth Corporation. All rights reserved.
5 #
6 # This software is provided under a slightly modified version
7 # of the Apache Software License. See the accompanying LICENSE file
8 # for more information.

AD Certificate Services

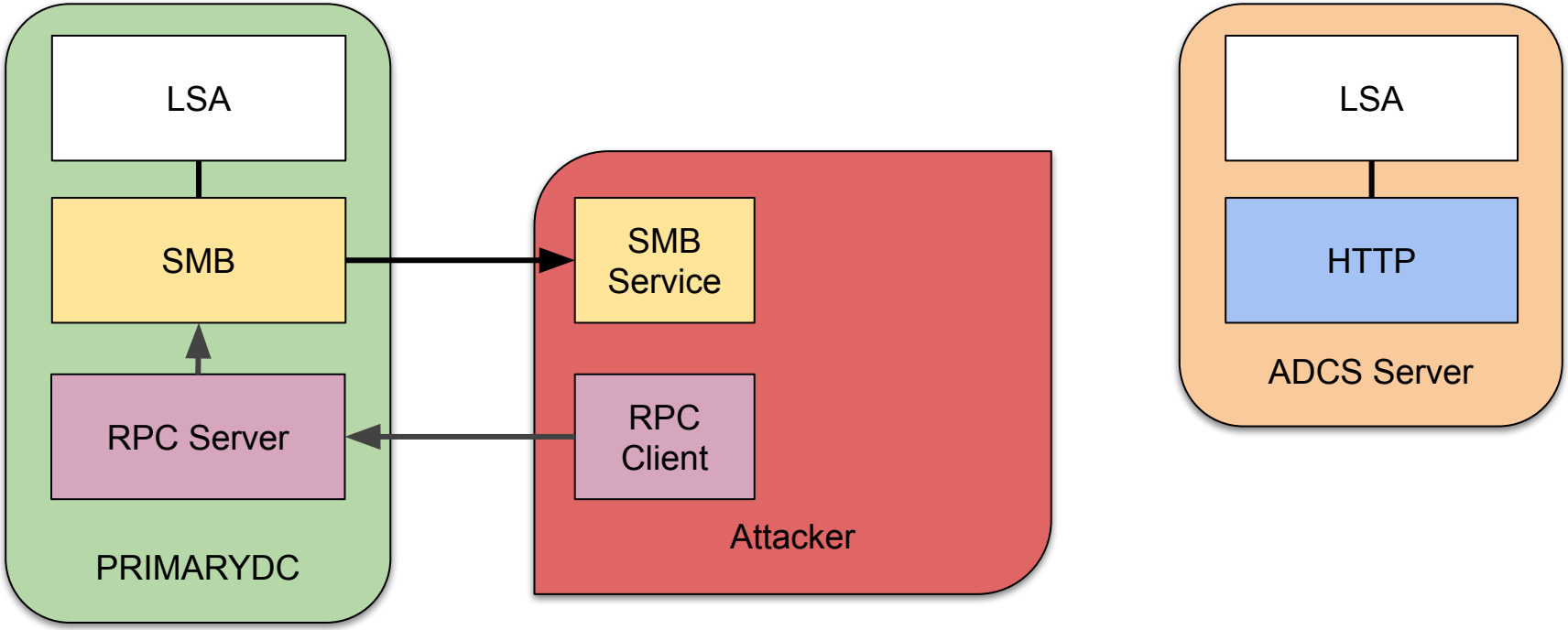


https://www.specterops.io/assets/resources/Certified_Pre-Owned.pdf

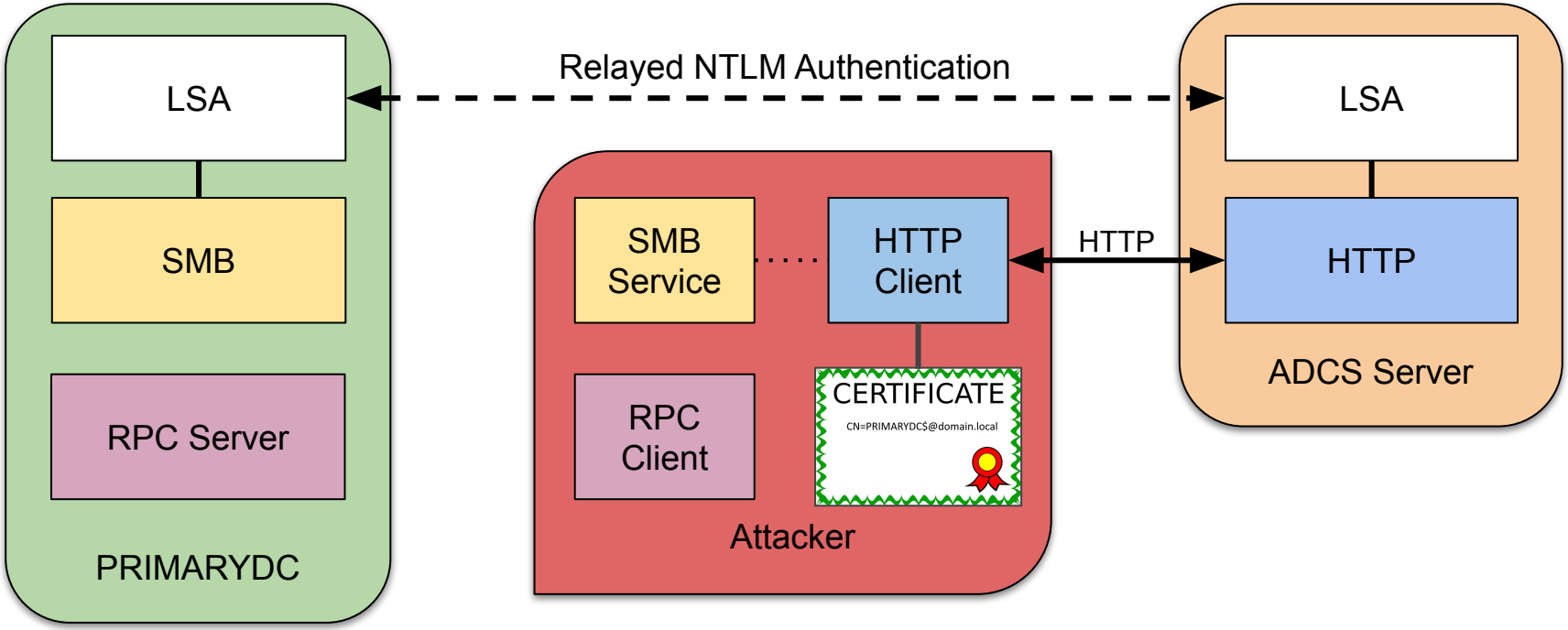
ADCS Authentication Relay (ESC8)



ADCS Authentication Relay (ESC8)



ADCS Authentication Relay (ESC8)

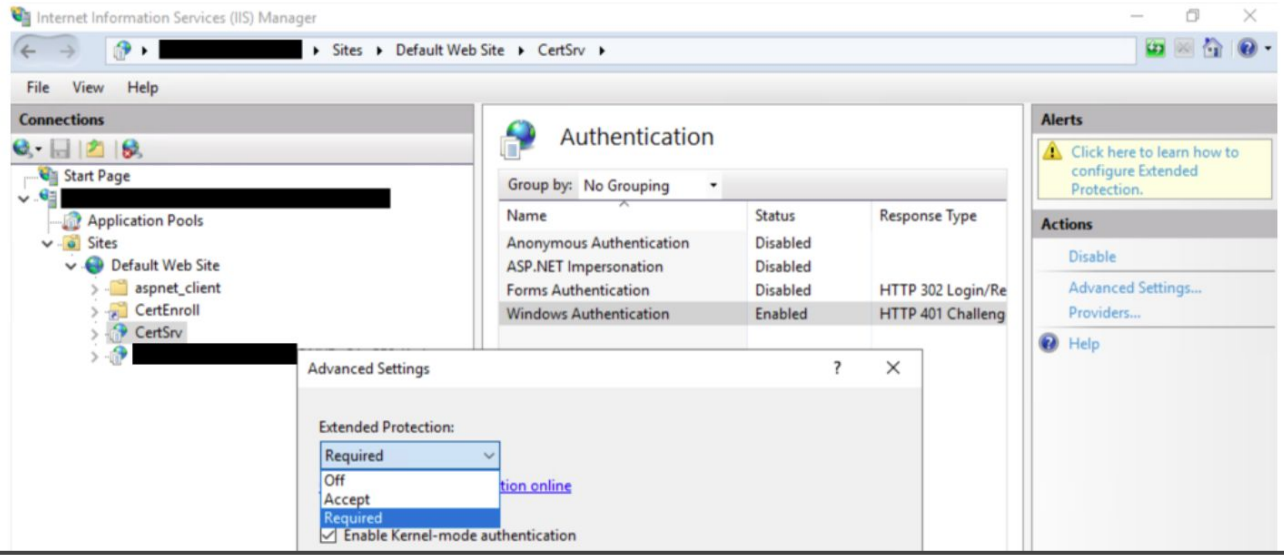


Extended Protection for Authentication (KB5005413)

Mitigation

We recommend enabling EPA and disabling HTTP on AD CS servers. Open the Internet Information Services (IIS) Manager and do the following:

1. Enable EPA for Certificate Authority Web Enrollment, **Required** being the more secure and recommended option:



https://support.microsoft.com/en-gb/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429

Disable NTLM!

Additional mitigation

In addition to the primary mitigations, we recommend you disable NTLM authentication where possible. The following mitigations are listed in order from more secure to less secure:

- Disable NTLM Authentication on your Windows domain controller. This can be accomplished by following the documentation in Network security: Restrict NTLM: NTLM authentication in this domain.

Disable NTLM!

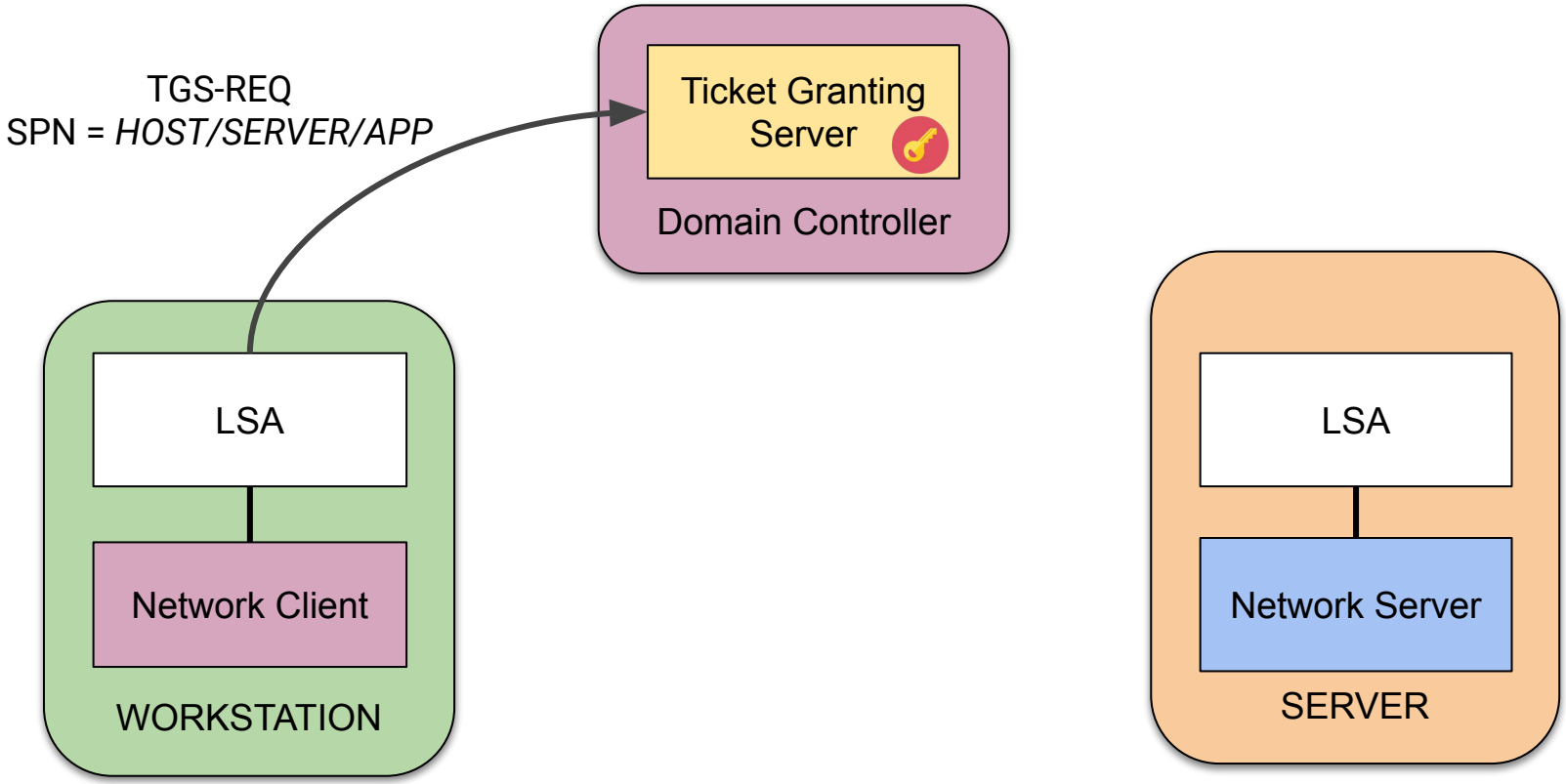
Security considerations

This section describes how an attacker might exploit a feature or its configuration, how to implement the countermeasure, and the possible negative consequences of countermeasure implementation.

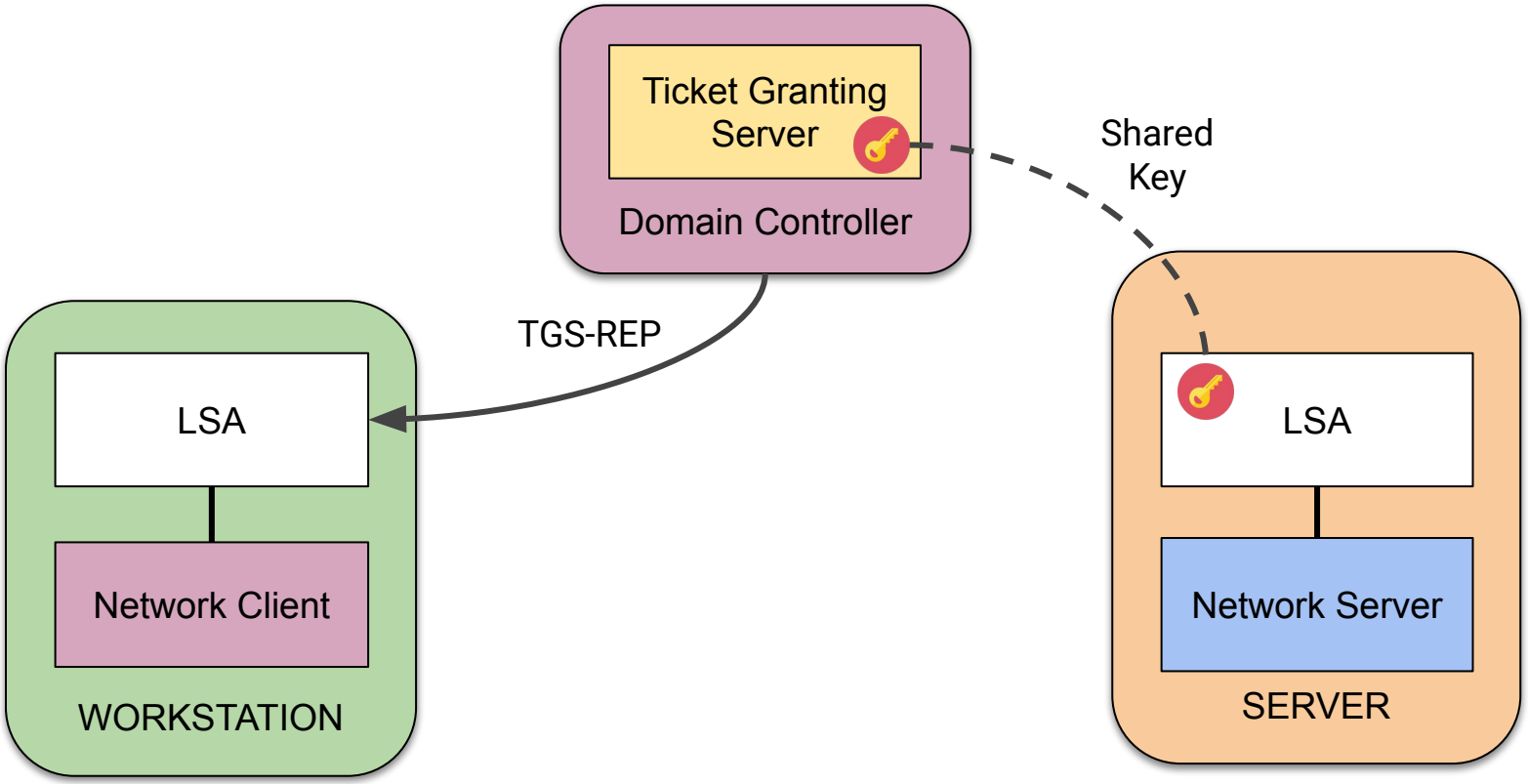
NTLM and NTLMv2 authentication is vulnerable to a variety of malicious attacks, including SMB replay, man-in-the-middle attacks, and brute force attacks. Reducing and eliminating NTLM authentication from your environment forces the Windows operating system to use more secure protocols, such as the Kerberos version 5 protocol, or different authentication mechanisms, such as smart cards.

What stops you relaying Kerberos authentication?

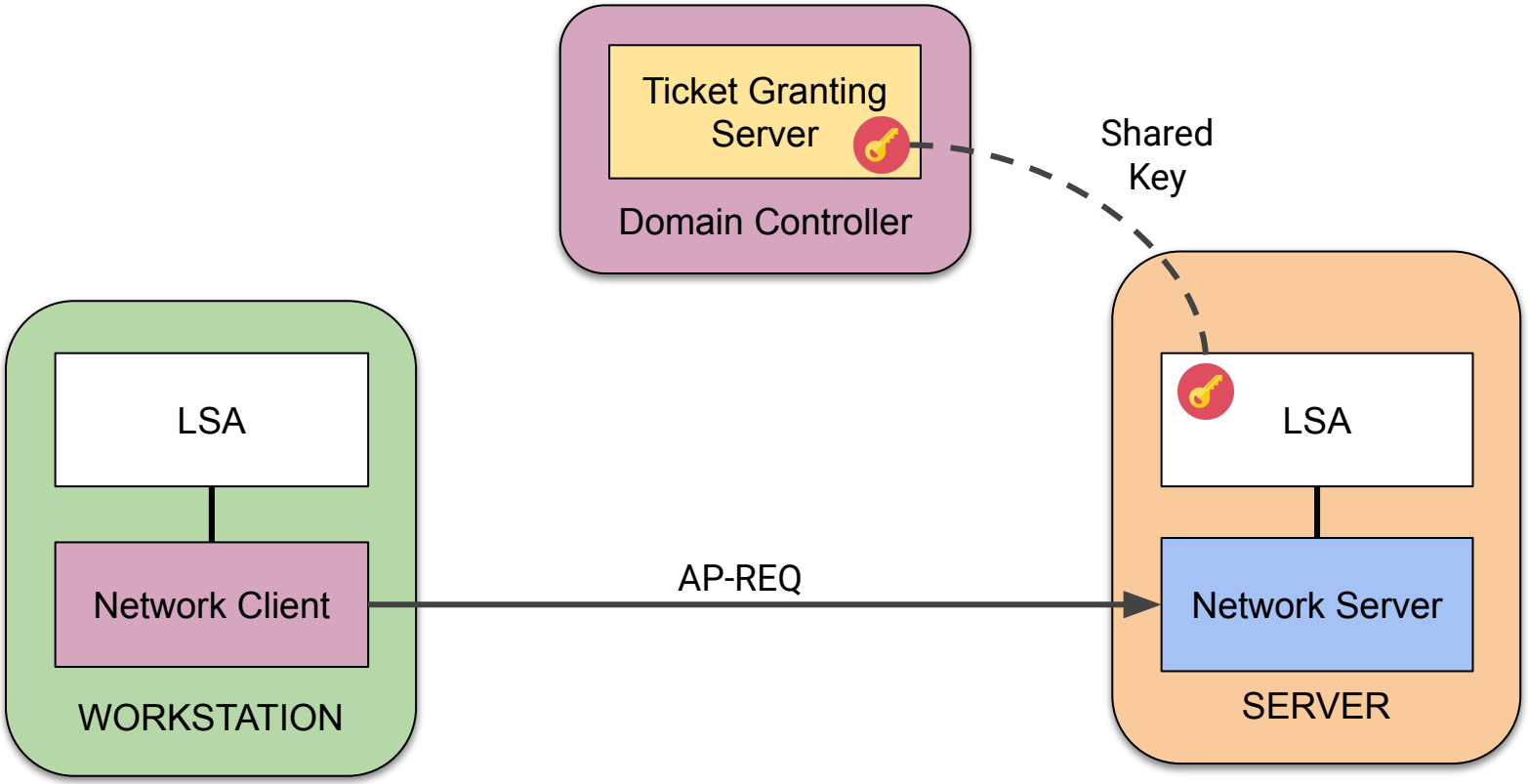
Kerberos Service Authentication



Kerberos Service Authentication



Kerberos Service Authentication



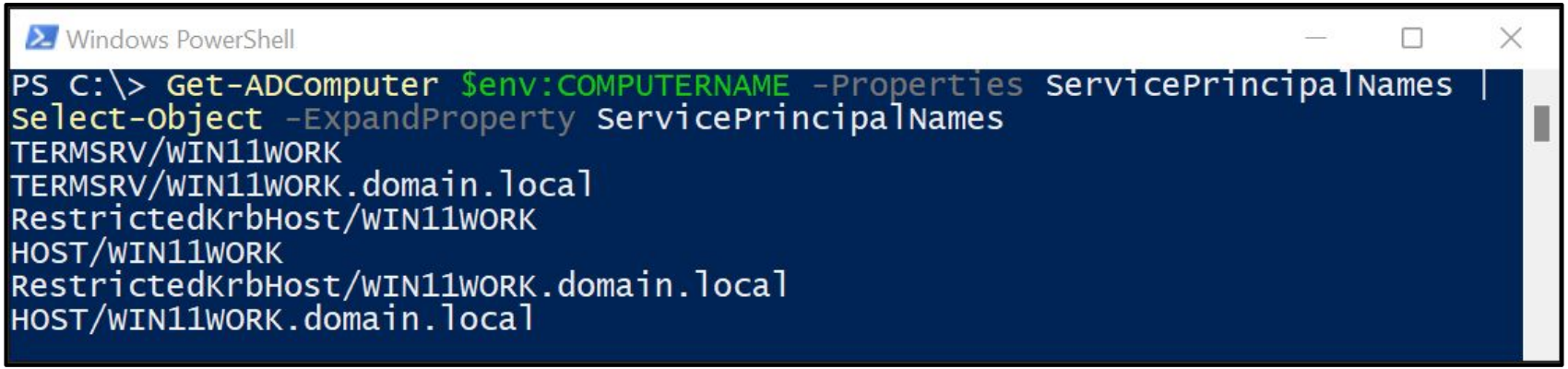
Service Principal Name



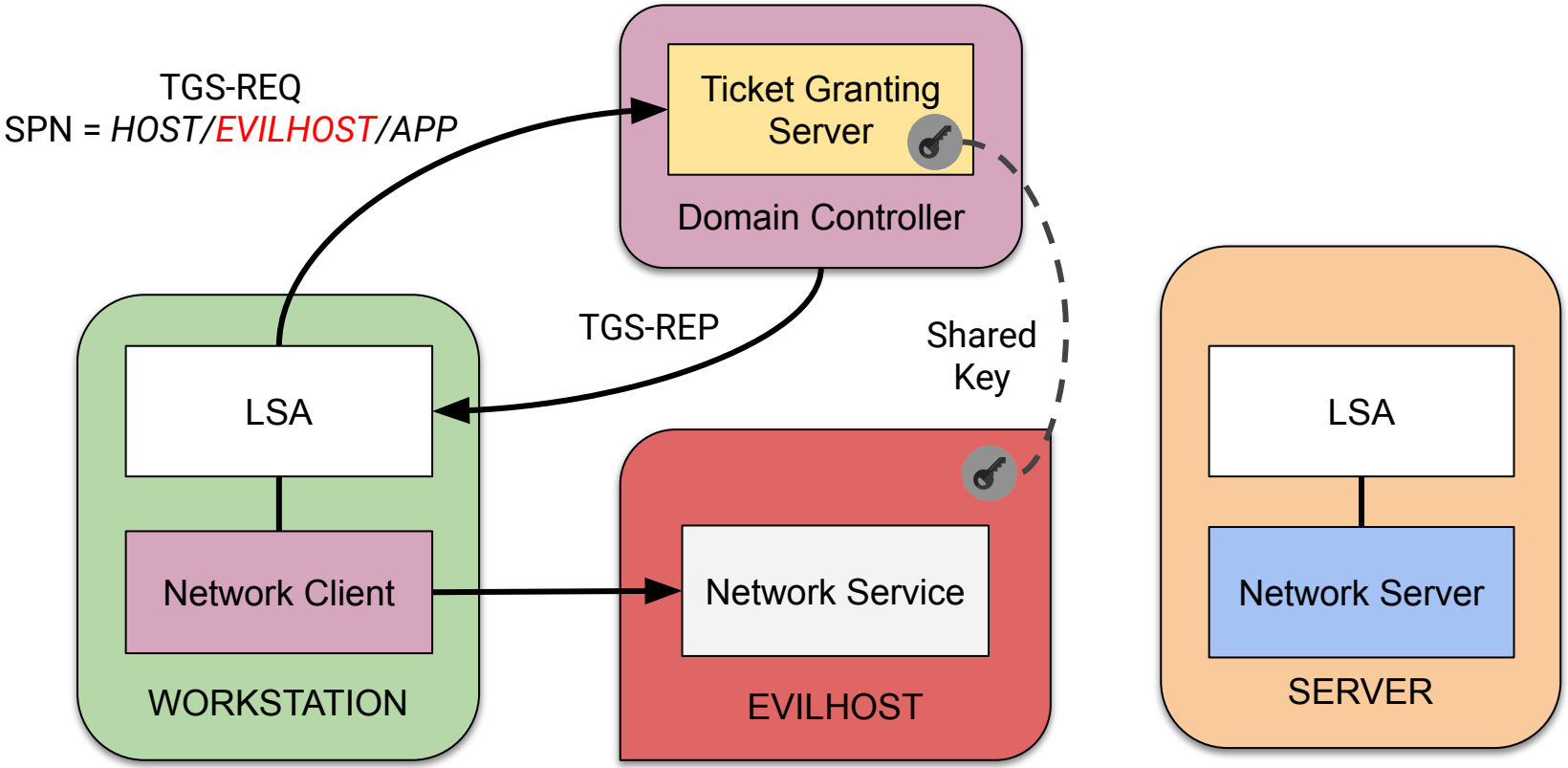
Service Class

Instance Name

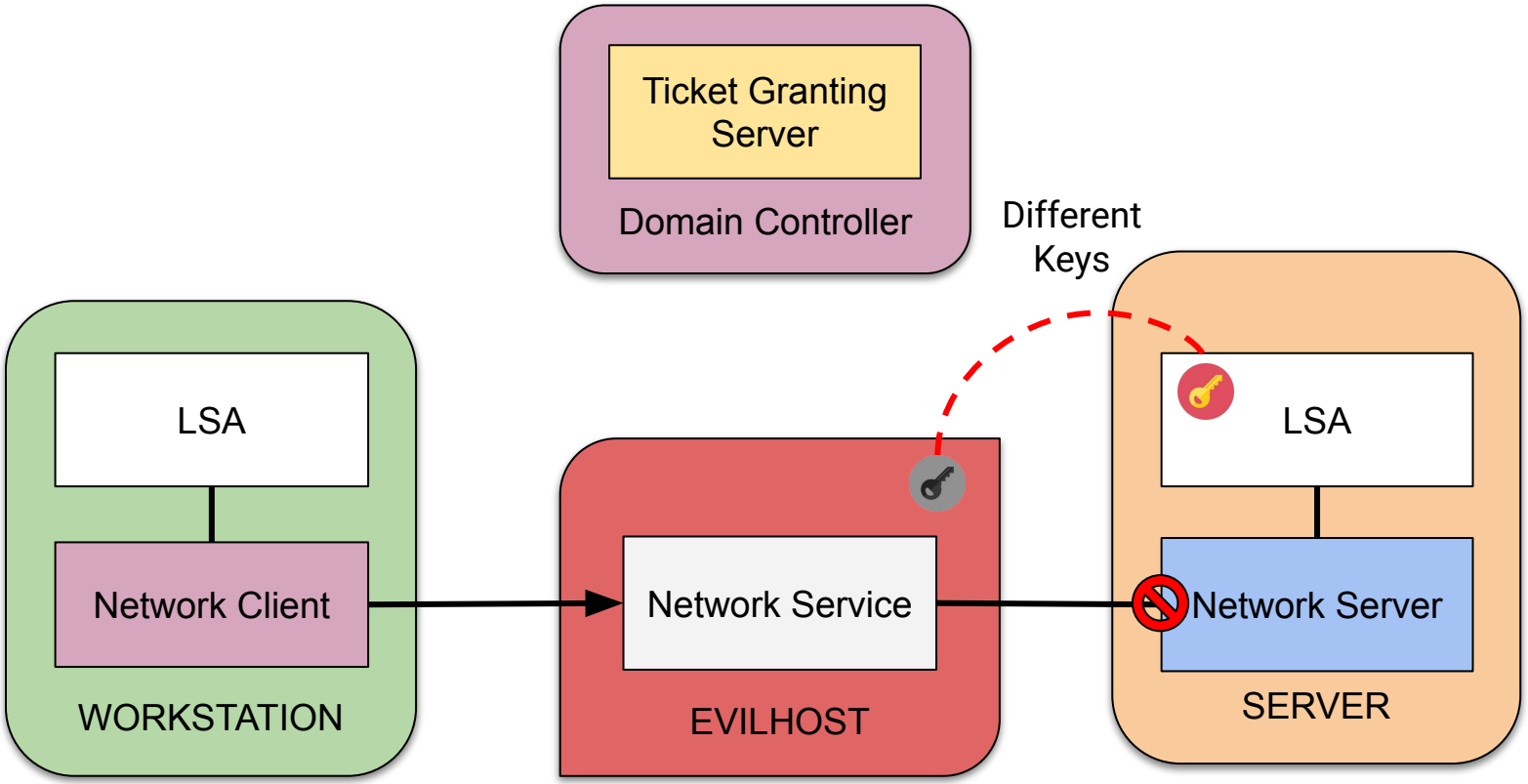
Service Name



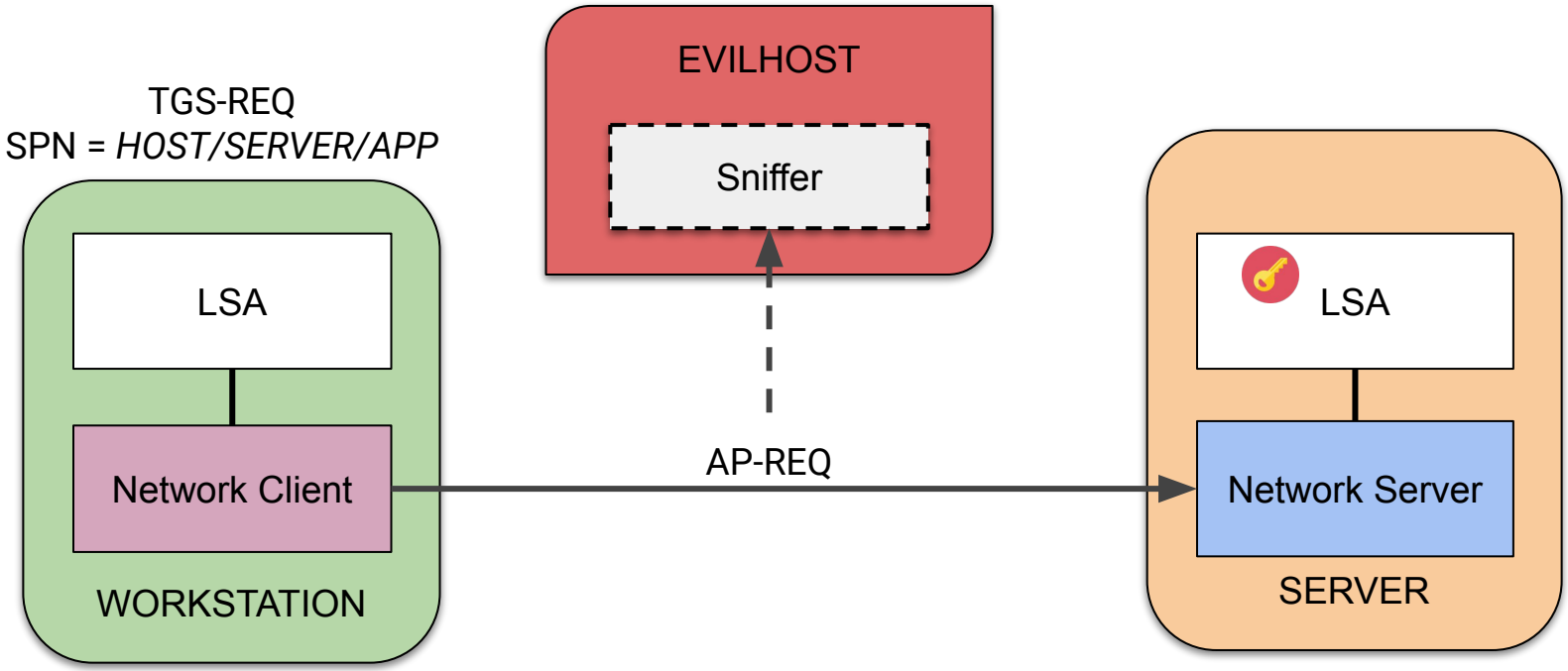
Interposing the Connection



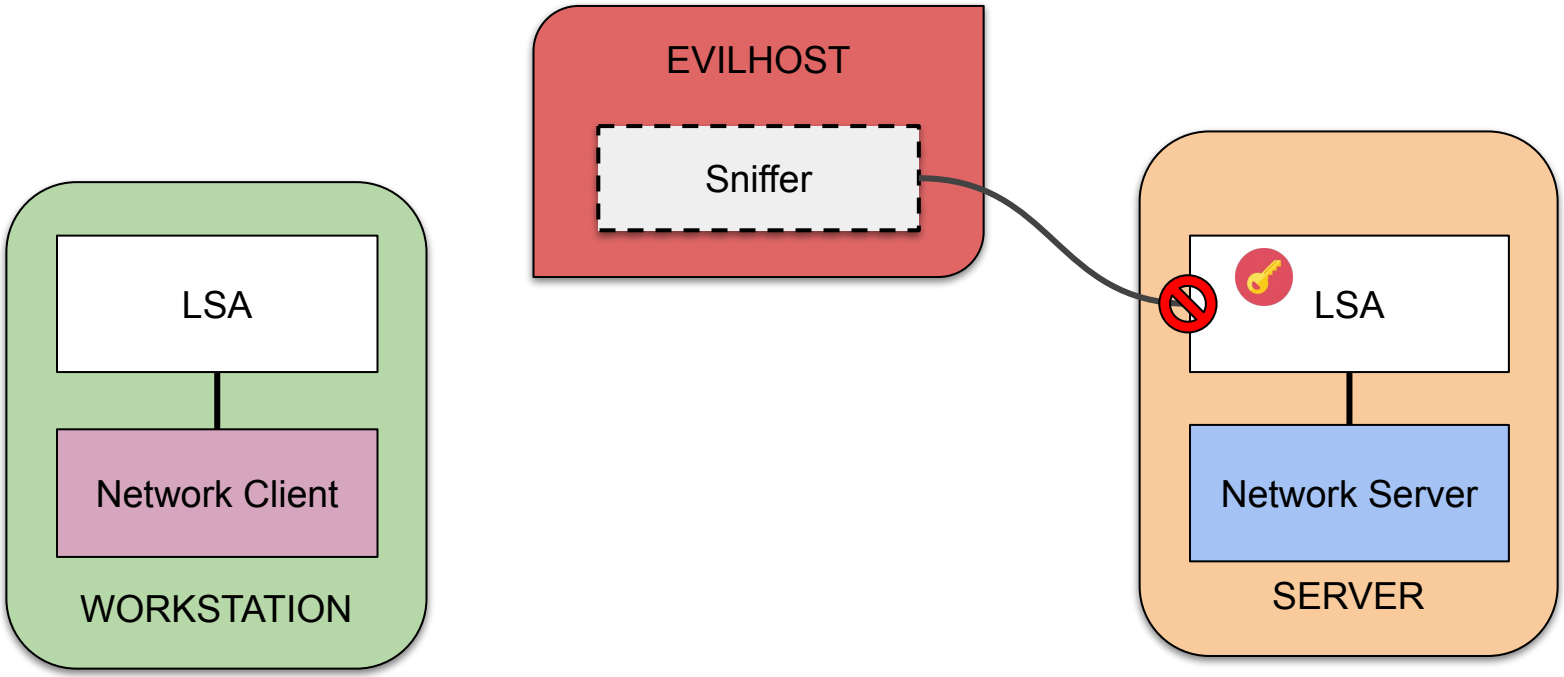
Interposing the Connection



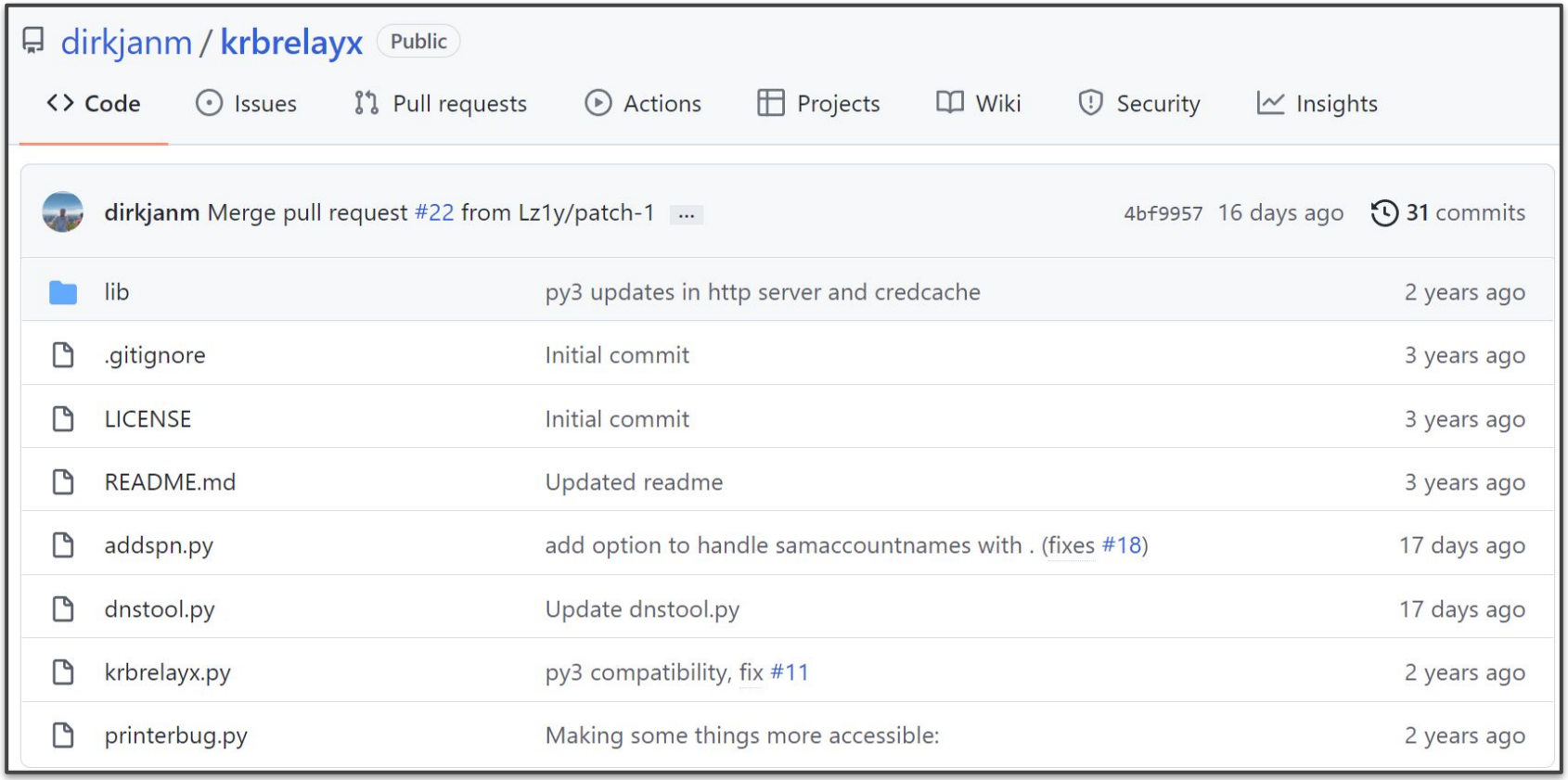
Sniffing for AP-REQs



Sniffing for AP-REQs



KrbrelayX


dirkjanm / krbrelayx
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
dirkjanm Merge pull request #22 from Lz1y/patch-1 ...
4bf9957 16 days ago
31 commits
lib py3 updates in http server and credcache 2 years ago
.gitignore Initial commit 3 years ago
LICENSE Initial commit 3 years ago
README.md Updated readme 3 years ago
addspn.py add option to handle samaccountnames with . (fixes #18) 17 days ago
dnstool.py Update dnstool.py 17 days ago
krbrelayx.py py3 compatibility, fix #11 2 years ago
printerbug.py Making some things more accessible: 2 years ago

Network Protocol Research

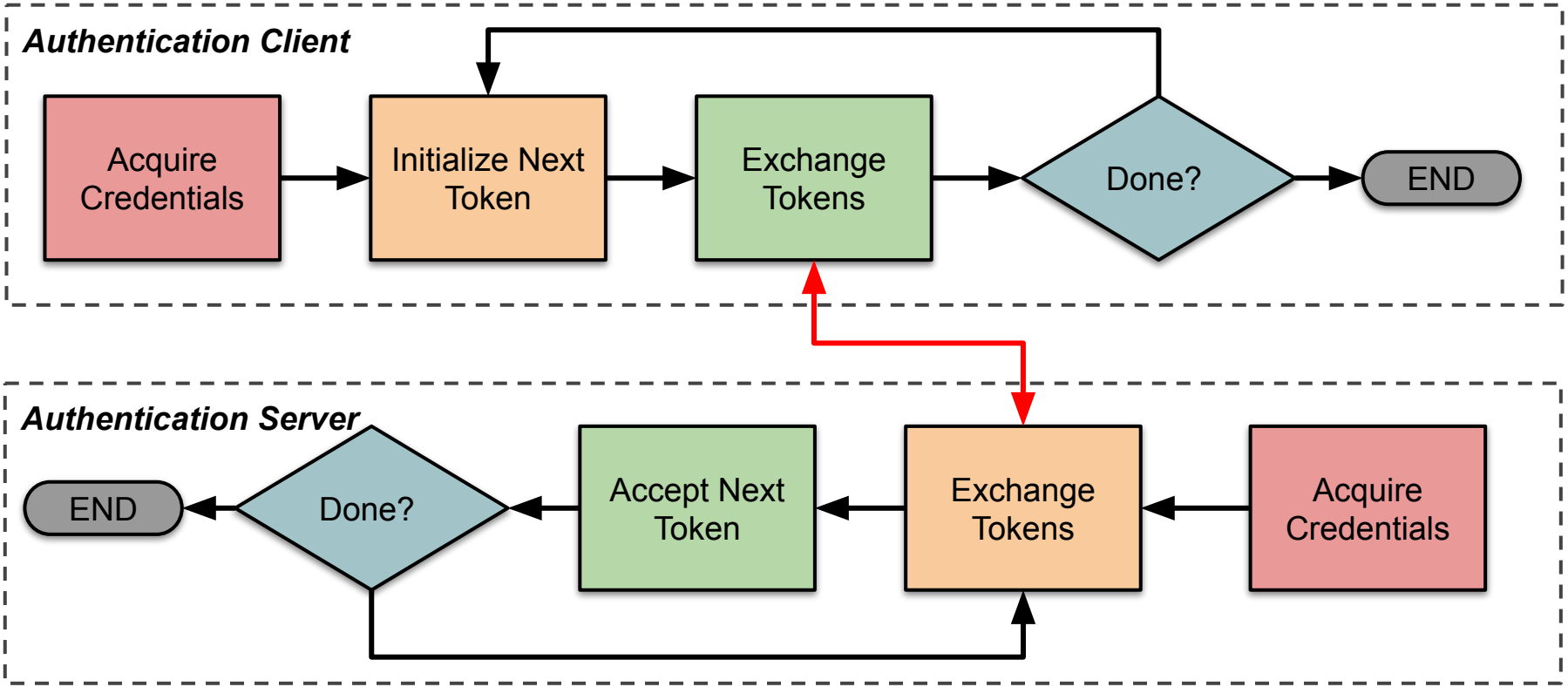
Research Rules

Look at network protocols for how the SPN is selected

Determine what security package is used

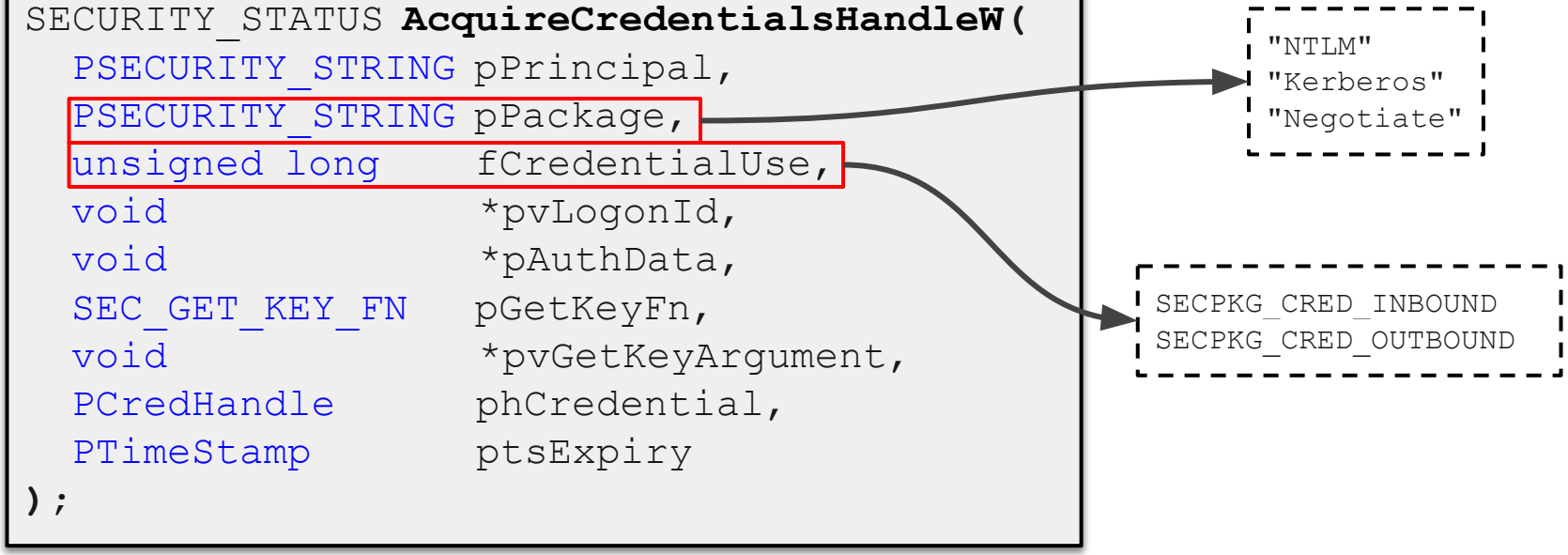
Determine what protections can be enabled

The Security Support Provider Interface (SSPI)



Acquiring Credentials

```
SECURITY_STATUS AcquireCredentialsHandleW(
  PSECURITY_STRING pPrincipal,
  PSECURITY_STRING pPackage,
  unsigned long    fCredentialUse,
  void             *pvLogonId,
  void             *pAuthData,
  SEC_GET_KEY_FN   pGetKeyFn,
  void             *pvGetKeyArgument,
  PCredHandle      phCredential,
  PTimeStamp       ptsExpiry
);
```



Client Token Initialization

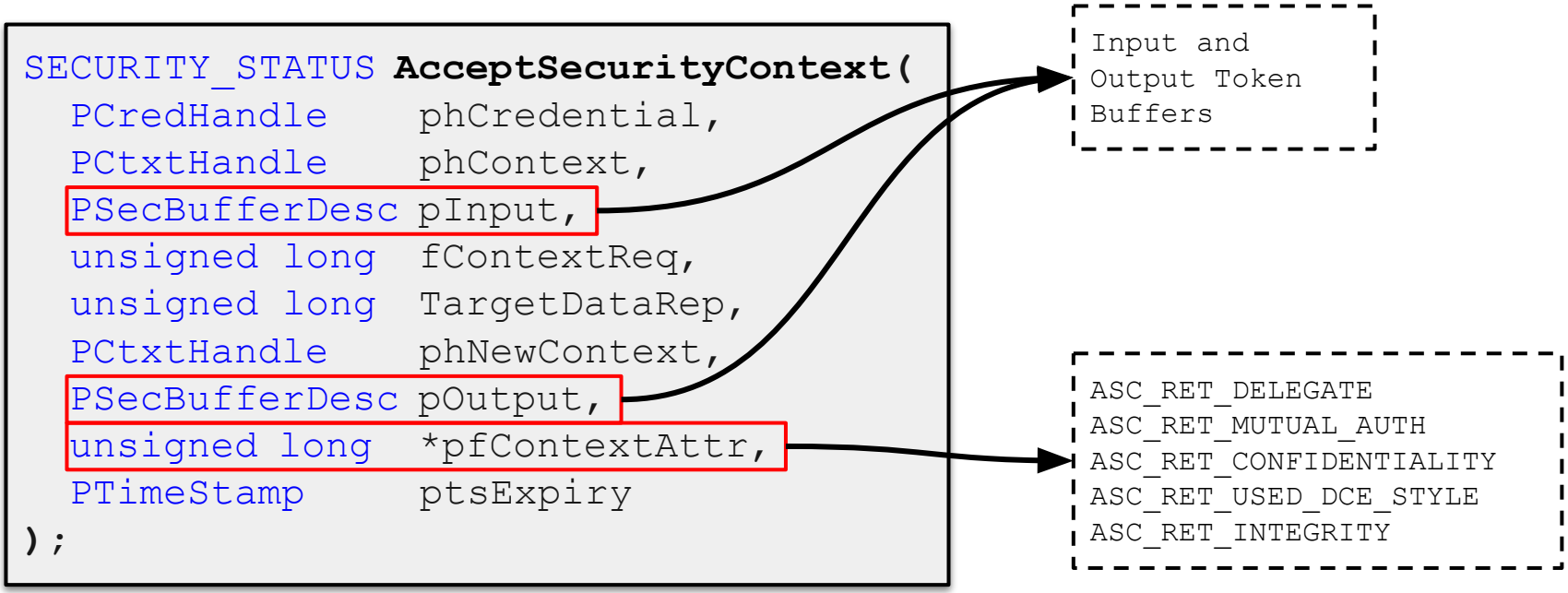
```
SECURITY_STATUS InitializeSecurityContextW(
  PCredHandle       phCredential,
  PCtxtHandle       phContext,
  PSECURITY_STRING  pTargetName,
  unsigned long     fContextReq,
  unsigned long     Reserved1,
  unsigned long     TargetDataRep,
  PSecBufferDesc    pInput,
  unsigned long     Reserved2,
  PCtxtHandle       phNewContext,
  PSecBufferDesc    pOutput,
  unsigned long     *pfContextAttr,
  PTimeStamp        ptsExpiry
);
```

SPN

ISC_REQ_DELEGATE
ISC_REQ_MUTUAL_AUTH
ISC_REQ_CONFIDENTIALITY
ISC_REQ_USE_DCE_STYLE
ISC_REQ_INTEGRITY
ISC_REQ_IDENTIFY
ISC_REQ_NO_INTEGRITY
ISC_REQ_UNVERIFIED_TARGET_NAME

Input and Output
Token Buffers

Server Token Acceptance



Signing and Sealing

```
SECURITY_STATUS MakeSignature(
  PCtxtHandle     phContext,
  unsigned long   fQOP,
  PSecBufferDesc  pMessage,
  unsigned long   MessageSeqNo
);
```

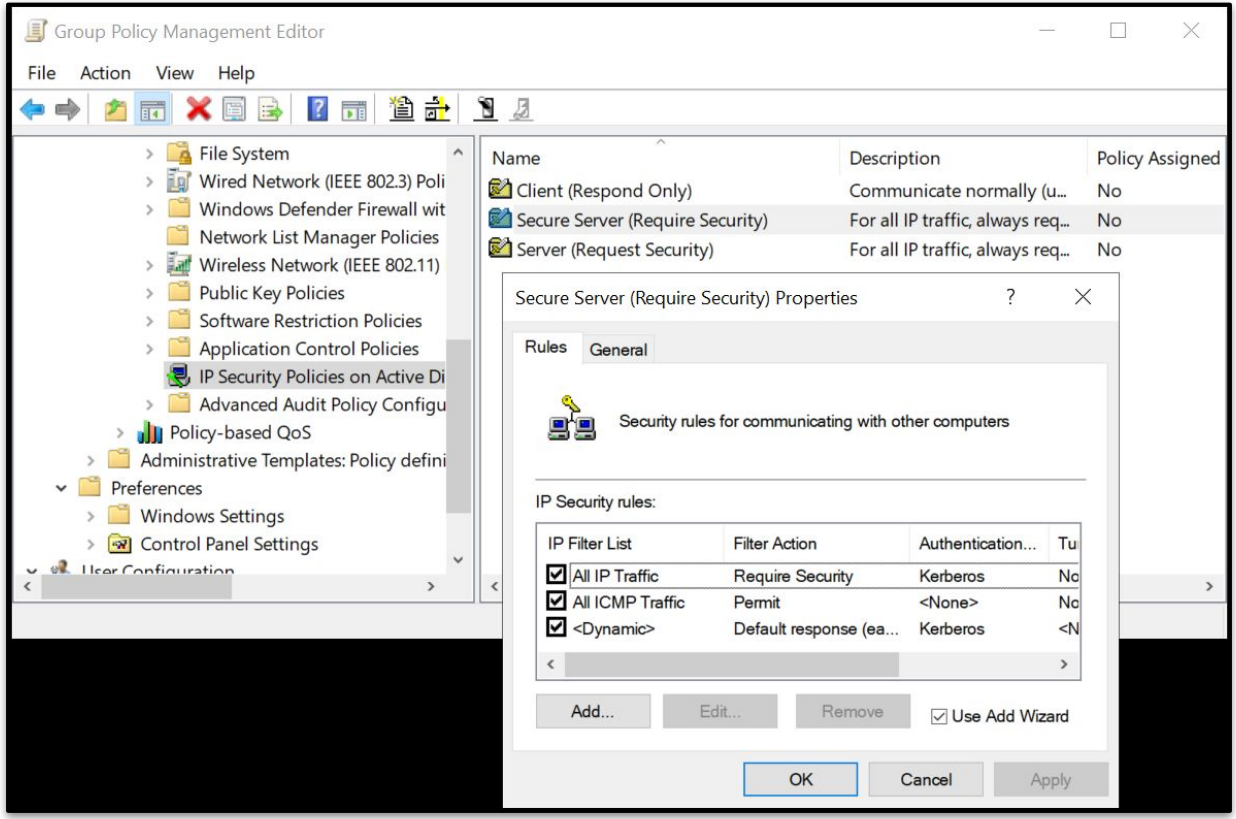
```
ISC_REQ_INTEGRITY or
ISC_REQ_CONFIDENTIALITY
```

```
SECURITY_STATUS EncryptMessage(
  PCtxtHandle     phContext,
  unsigned long   fQOP,
  PSecBufferDesc  pMessage,
  unsigned long   MessageSeqNo
);
```

```
ISC_REQ_CONFIDENTIALITY
```

DEMO

IP Security


Group Policy Management Editor
File Action View Help
File System
Wired Network (IEEE 802.3) Poli
Windows Defender Firewall wit
Network List Manager Policies
Wireless Network (IEEE 802.11)
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Active Di
Advanced Audit Policy Configu
Policy-based QoS
Administrative Templates: Policy defini
Preferences
Windows Settings
Control Panel Settings
Name
Description
Policy Assigned
Client (Respond Only)
Communicate normally (u...
No
Secure Server (Require Security)
For all IP traffic, always req...
No
Server (Request Security)
For all IP traffic, always req...
No
Secure Server (Require Security) Properties
Rules
General
Security rules for communicating with other computers
IP Security rules:
IP Filter List
Filter Action
Authentication...
Tu
All IP Traffic
Require Security
Kerberos
No
All ICMP Traffic
Permit
<None>
No
<Dynamic>
Default response (ea...
Kerberos
<N
Add...
Edit...
Remove
Use Add Wizard
OK
Cancel
Apply

AuthIP GSS-ID

2.2.3.3 GSS_ID 0x86 Payload Packet

In IKEv1, the identity information is passed between peers as an attribute of the security association (SA) in the form of an identity payload, as specified in [RFC2408] section 3.8. In the Authenticated Internet Protocol, a separate payload is used to convey identity information in the form of a security principal name.

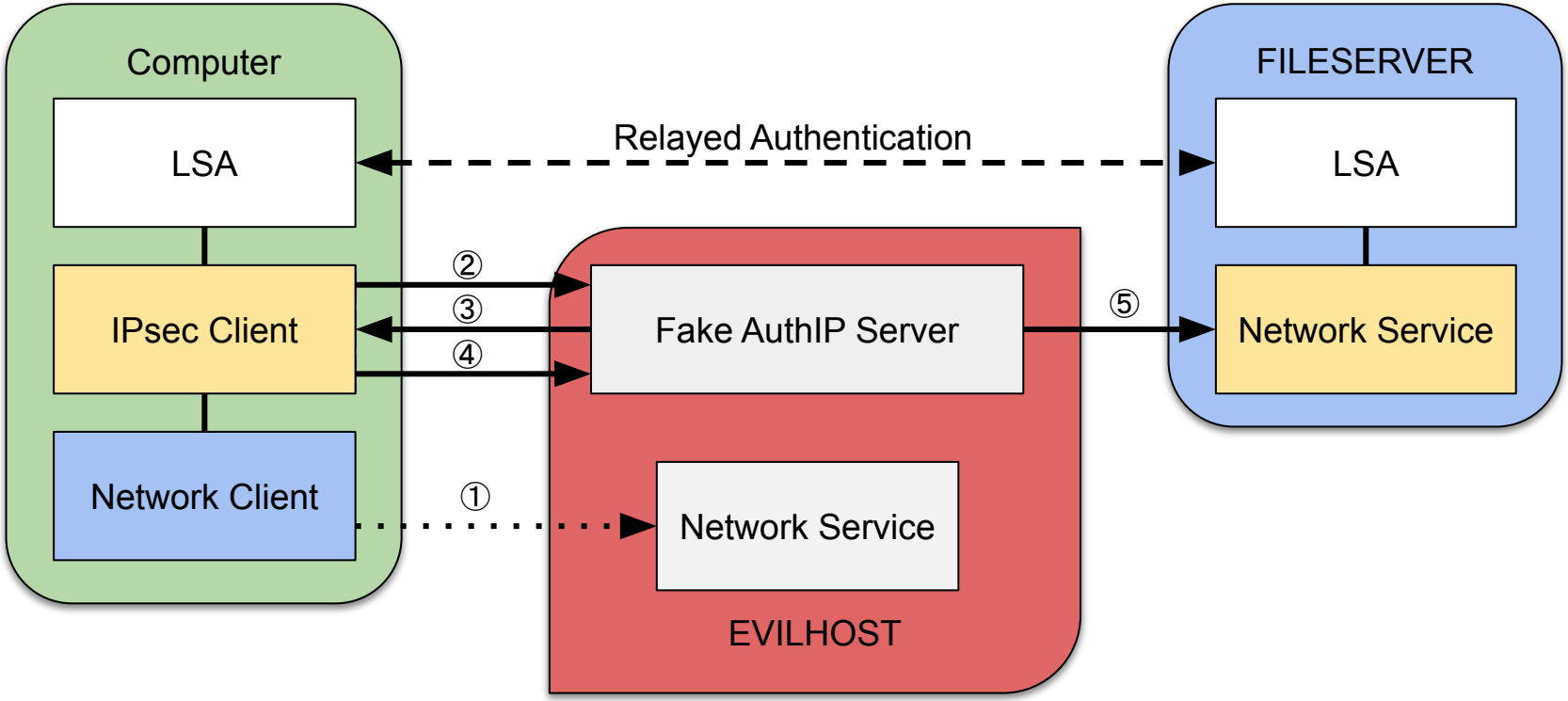
The following diagram shows the GSS_ID 0x86 payload packet structure.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Security Principle Name (variable)																															
...																															

Security Principle Name (variable): The GSS-API representation of a security principal name, as specified in [MS-KILE] section 3.1.5.11. The security principal name sent MUST be a Unicode string. The security principal name MUST NOT contain the NULL string terminator. All authentication mechanisms MUST use this representation of the security principal name.

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-aips/9ab1ccc0-d92e-4ca4-bae9-1c93871399ac

Relaying AuthIP Authentication



Would MS fix it?

Issue 2213: Windows: IKEEXT AuthIP Unvalidated GSS_ID EoP

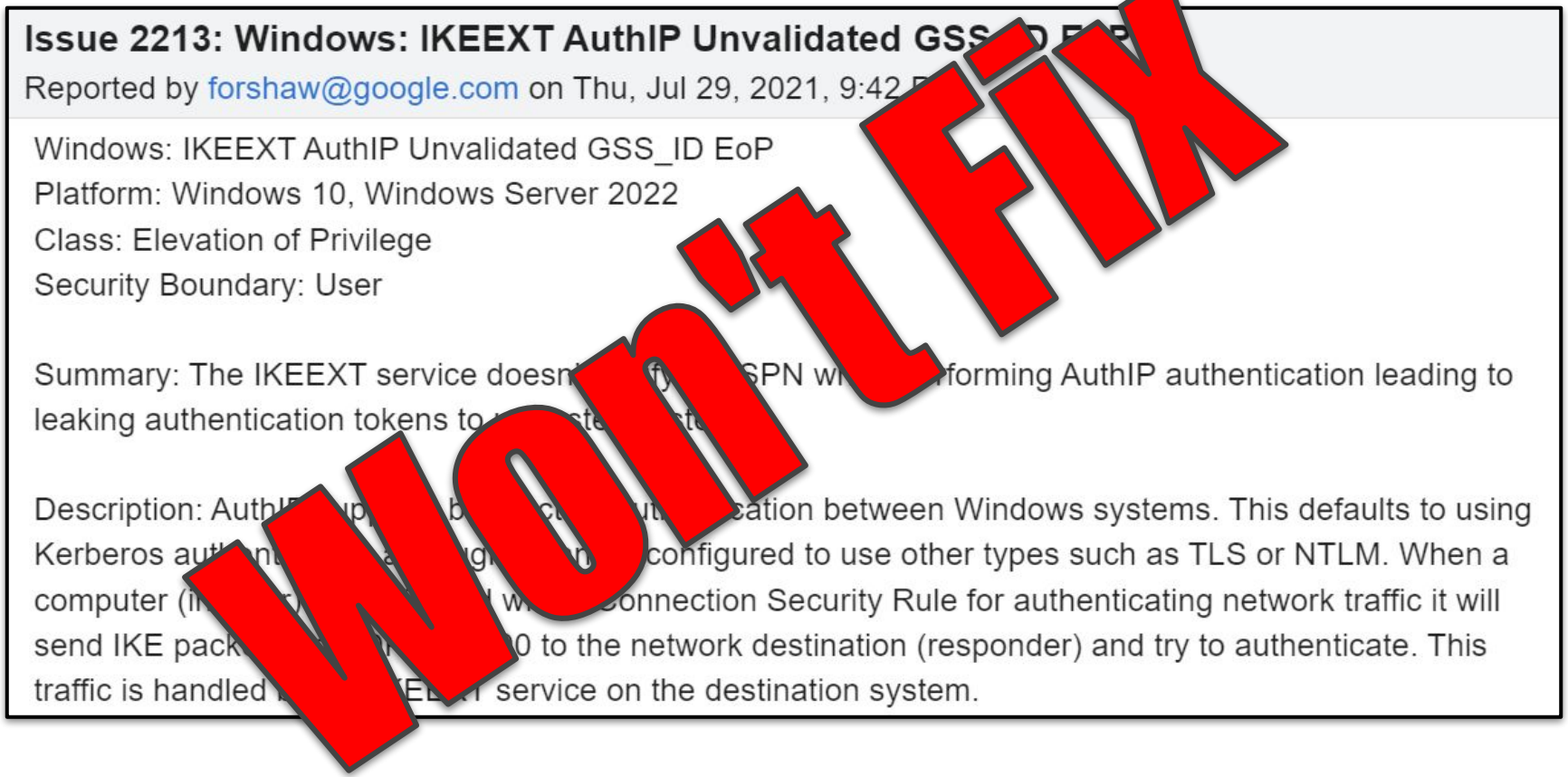
Reported by forshaw@google.com on Thu, Jul 29, 2021, 9:42 PM

Windows: IKEEXT AuthIP Unvalidated GSS_ID EoP
Platform: Windows 10, Windows Server 2022
Class: Elevation of Privilege
Security Boundary: User

Summary: The IKEEXT service doesn't verify the SPN when performing AuthIP authentication leading to leaking authentication tokens to untrusted systems.

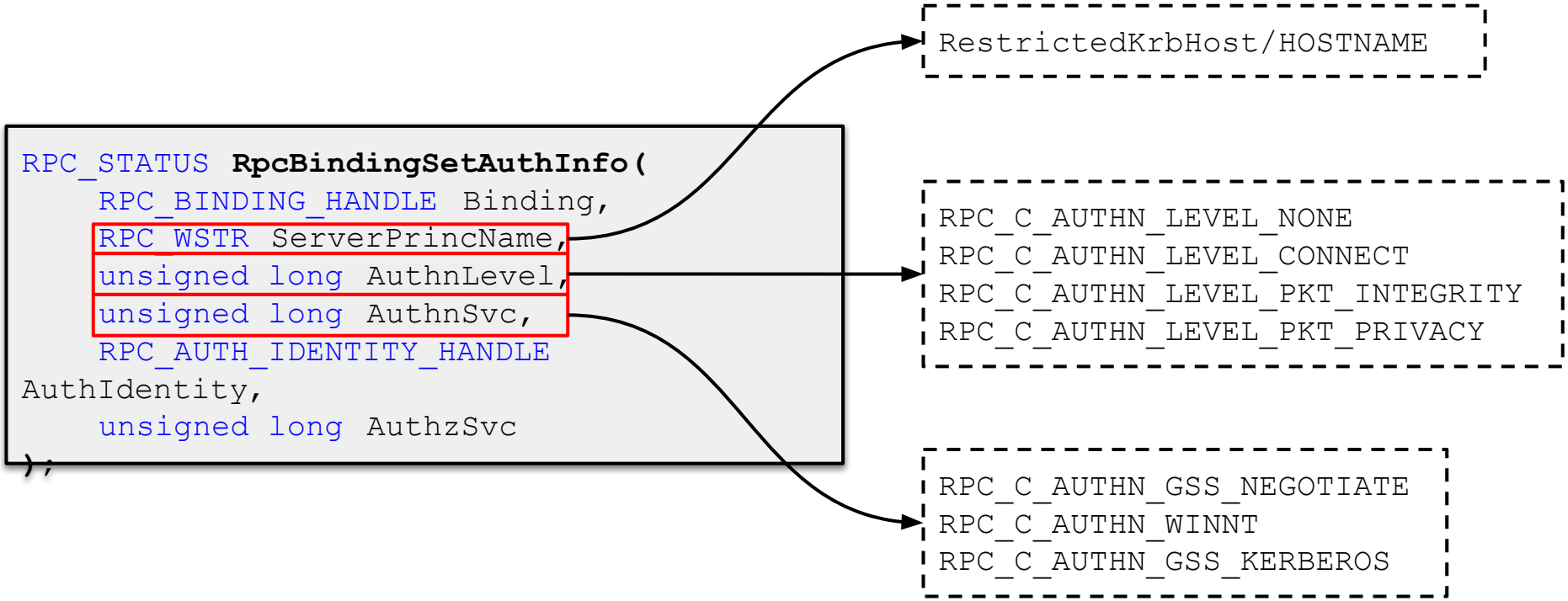
Description: AuthIP supports bi-direction authentication between Windows systems. This defaults to using Kerberos authentication although it can be configured to use other types such as TLS or NTLM. When a computer (initiator) is configured with a Connection Security Rule for authenticating network traffic it will send IKE packets on UDP port 500 to the network destination (responder) and try to authenticate. This traffic is handled by the IKEEXT service on the destination system.

Would MS fix it?

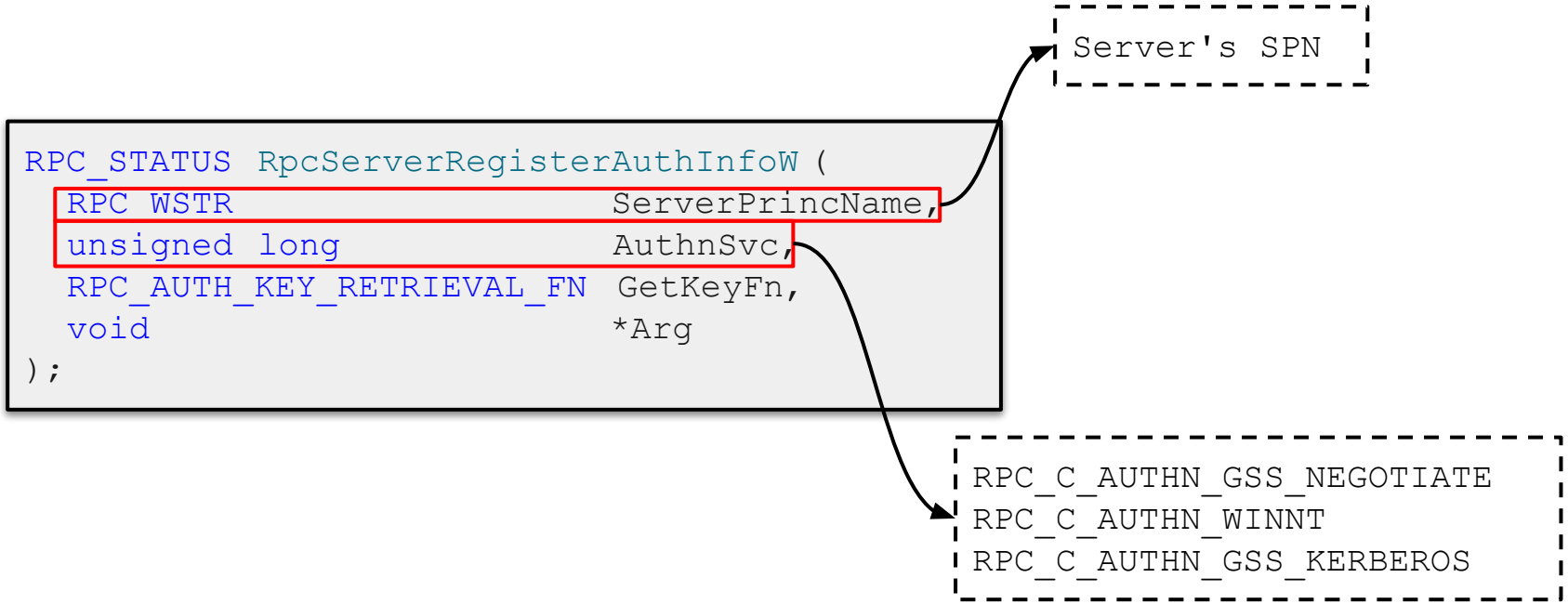


RPC Protocols

MSRPC Clients



MSRPC Server



Querying the Server's SPN

RpcMgmtInqServerPrincNameW function (rpcdce.h)

The **RpcMgmtInqServerPrincName** function returns a server's principal name.

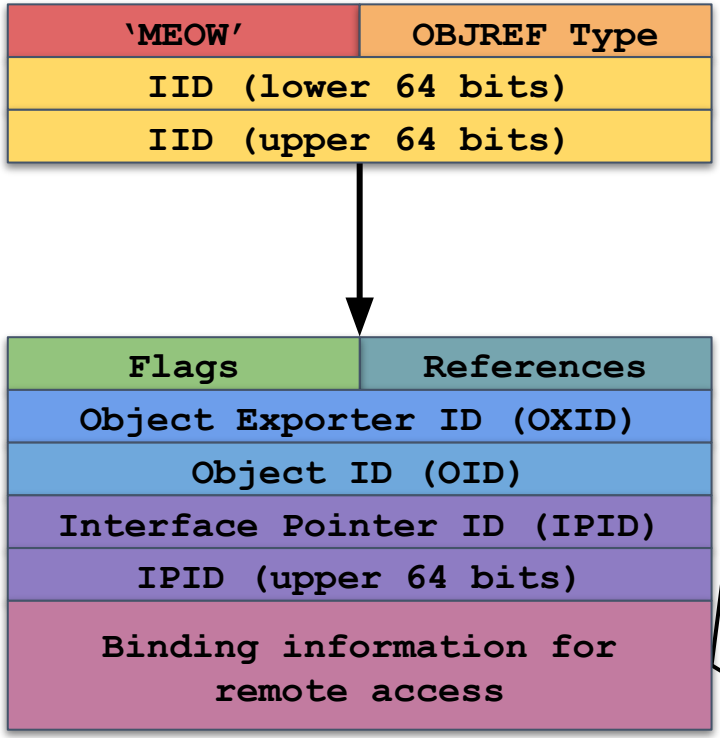
Syntax

C++ Copy

```
RPC_STATUS RpcMgmtInqServerPrincNameW(
  RPC_BINDING_HANDLE Binding,
  unsigned long      AuthnSvc,
  RPC_WSTR           *ServerPrincName
);
```

DEMO

OBJREF



Standard OBJREF

2.2.19.3 STRINGBINDING

The STRINGBINDING structure describes an RPC protocol, a network address, and, optionally, an RPC endpoint for the RPC protocol that a client can use to communicate with either an object resolver or an object exporter.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
wTowerId																aNetworkAddr (variable)															
...																															

2.2.19.4 SECURITYBINDING

The SECURITYBINDING structure describes an RPC security provider and a service principal name (SPN). A client uses these to communicate with either an object resolver or an object exporter.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
wAuthnSvc																Reserved (optional)															
aPrincName (variable)																															
...																															

Security Binding

2.2.19.4 SECURITYBINDING

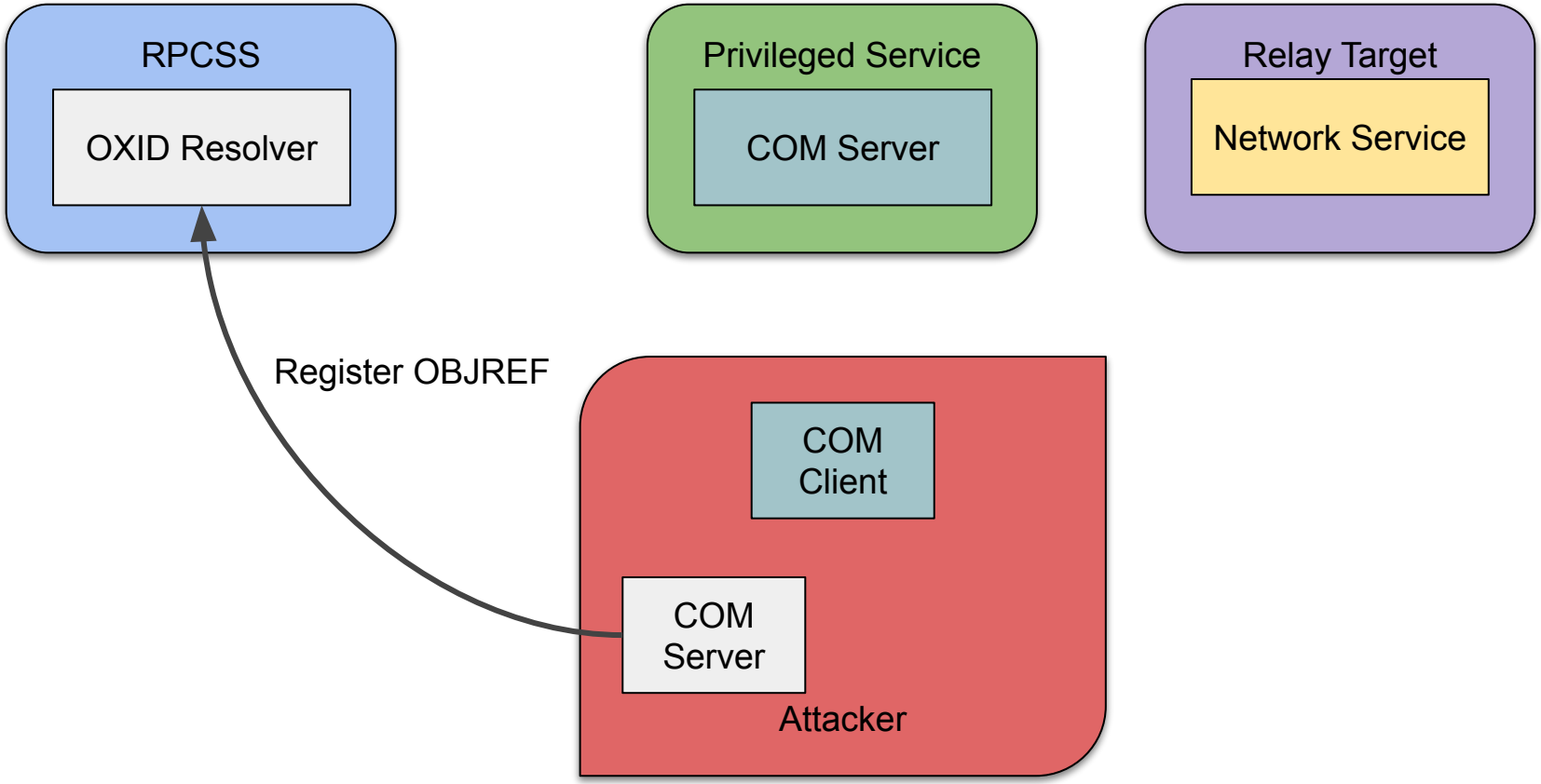
The SECURITYBINDING structure describes an RPC security provider and a service principal name (SPN). A client uses these to communicate with either an object resolver or an object exporter.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
wAuthnSvc																Reserved (optional)															
aPrincName (variable)																															
...																															

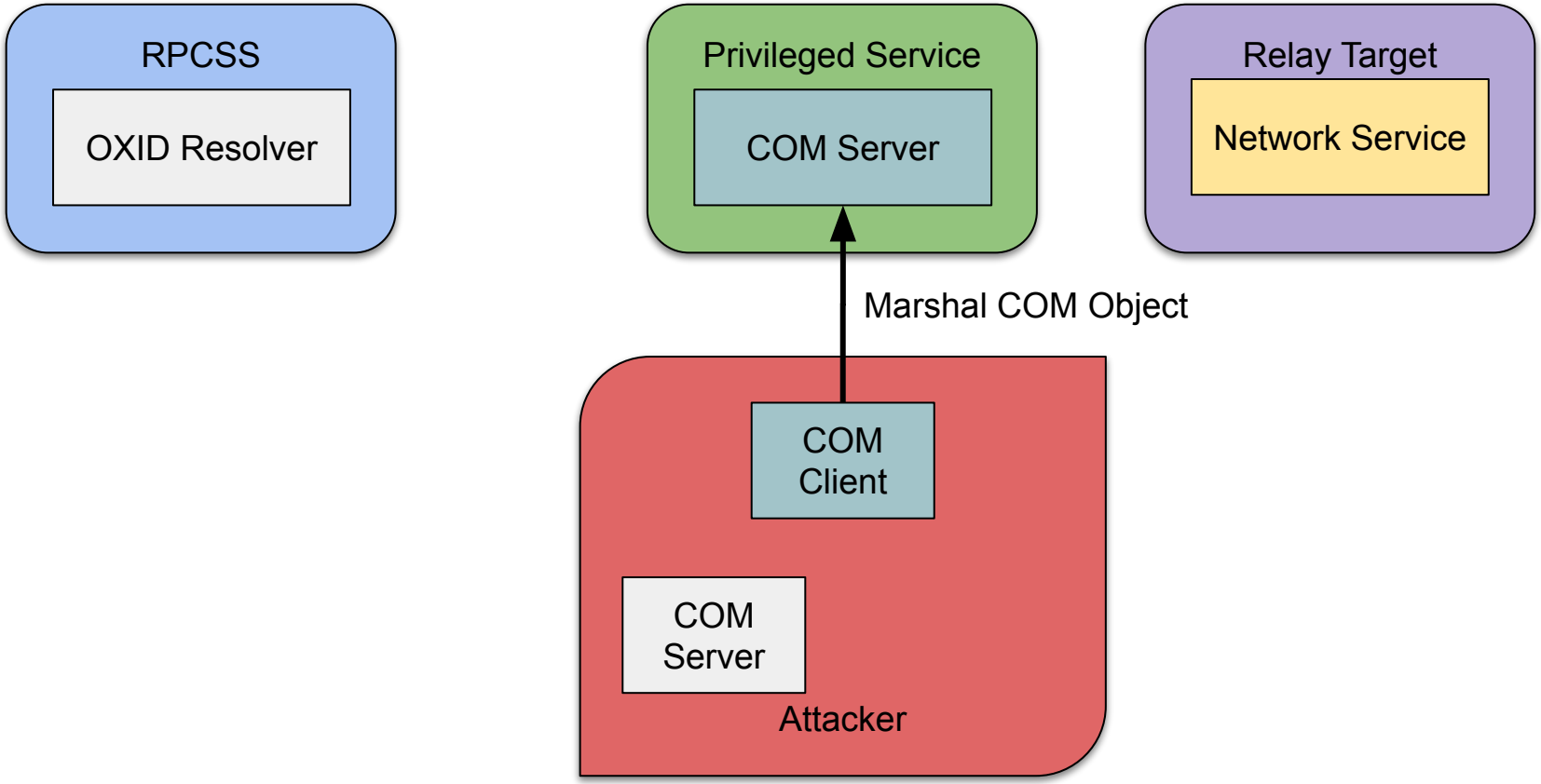
```
RPC_C_AUTHN_WINNT
RPC_C_AUTHN_GSS_NEGOTIATE
RPC_C_AUTHN_GSS_KERBEROS
```

```
HOST/myserver.domain.local
```

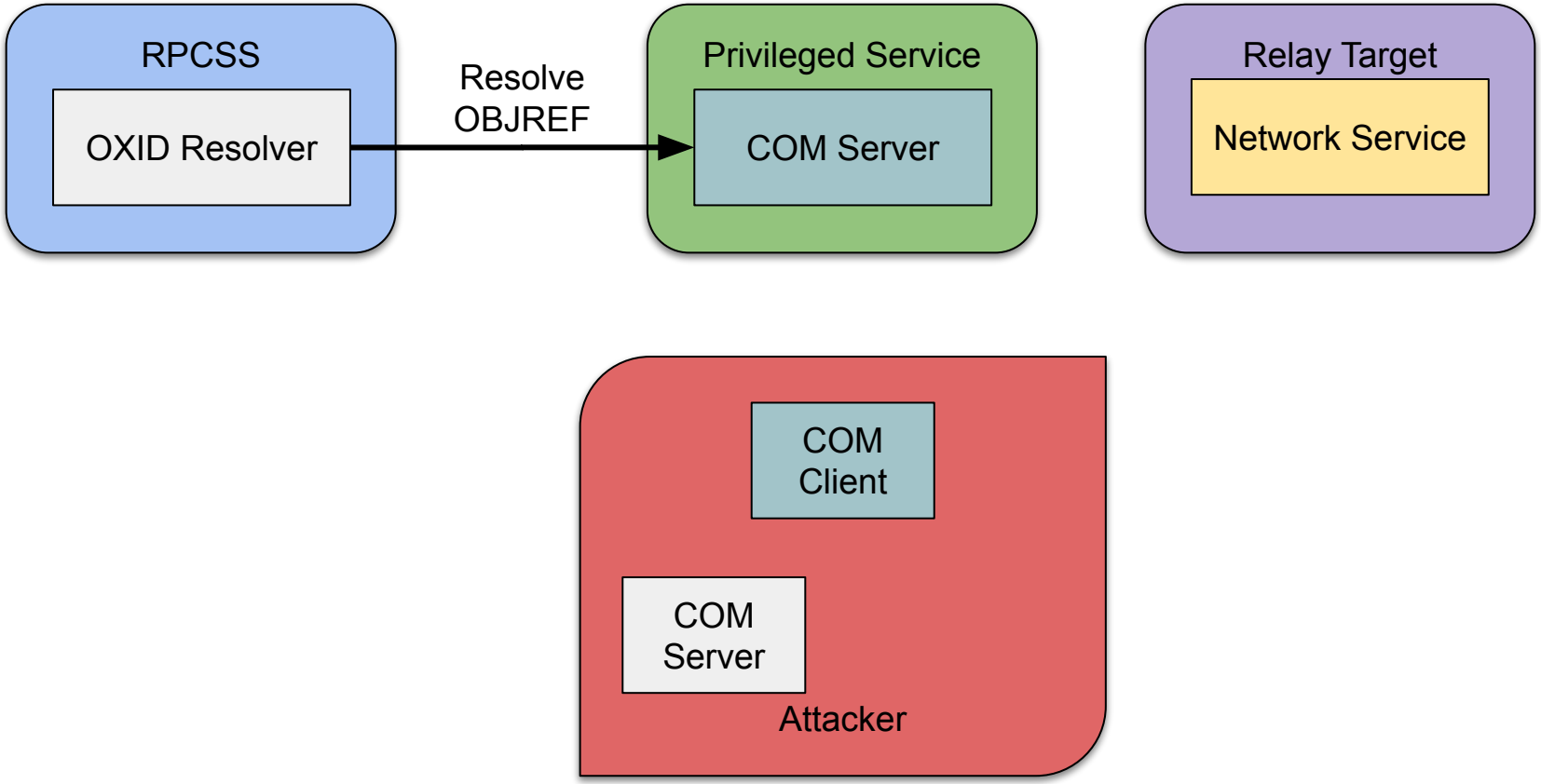
DCOM Relay


RPCSS
OXID Resolver
Privileged Service
COM Server
Relay Target
Network Service
Register OBJREF
COM
Client
COM
Server
Attacker

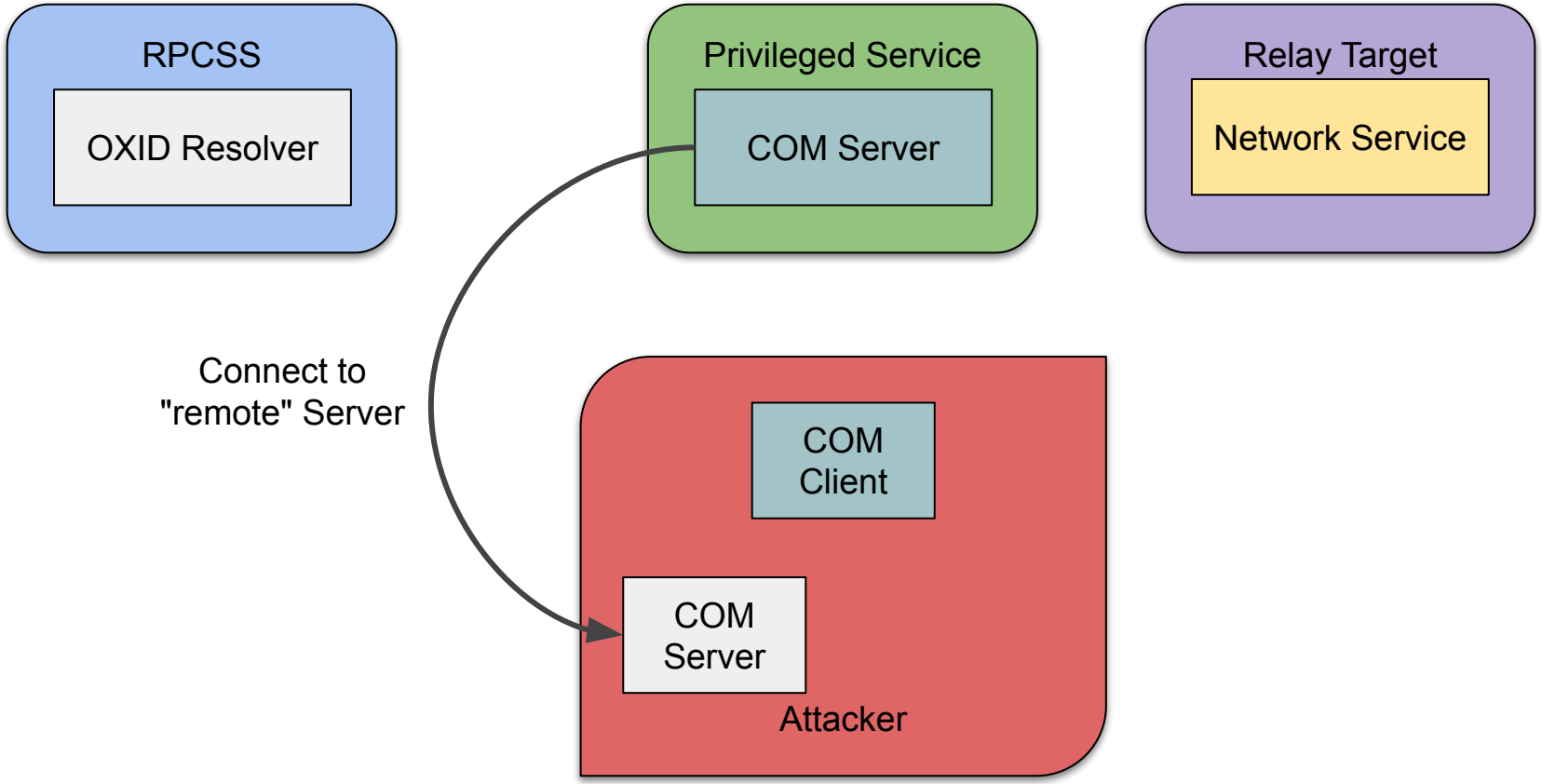
DCOM Relay


RPCSS
OXID Resolver
Privileged Service
COM Server
Relay Target
Network Service
Marshal COM Object
COM
Client
COM
Server
Attacker

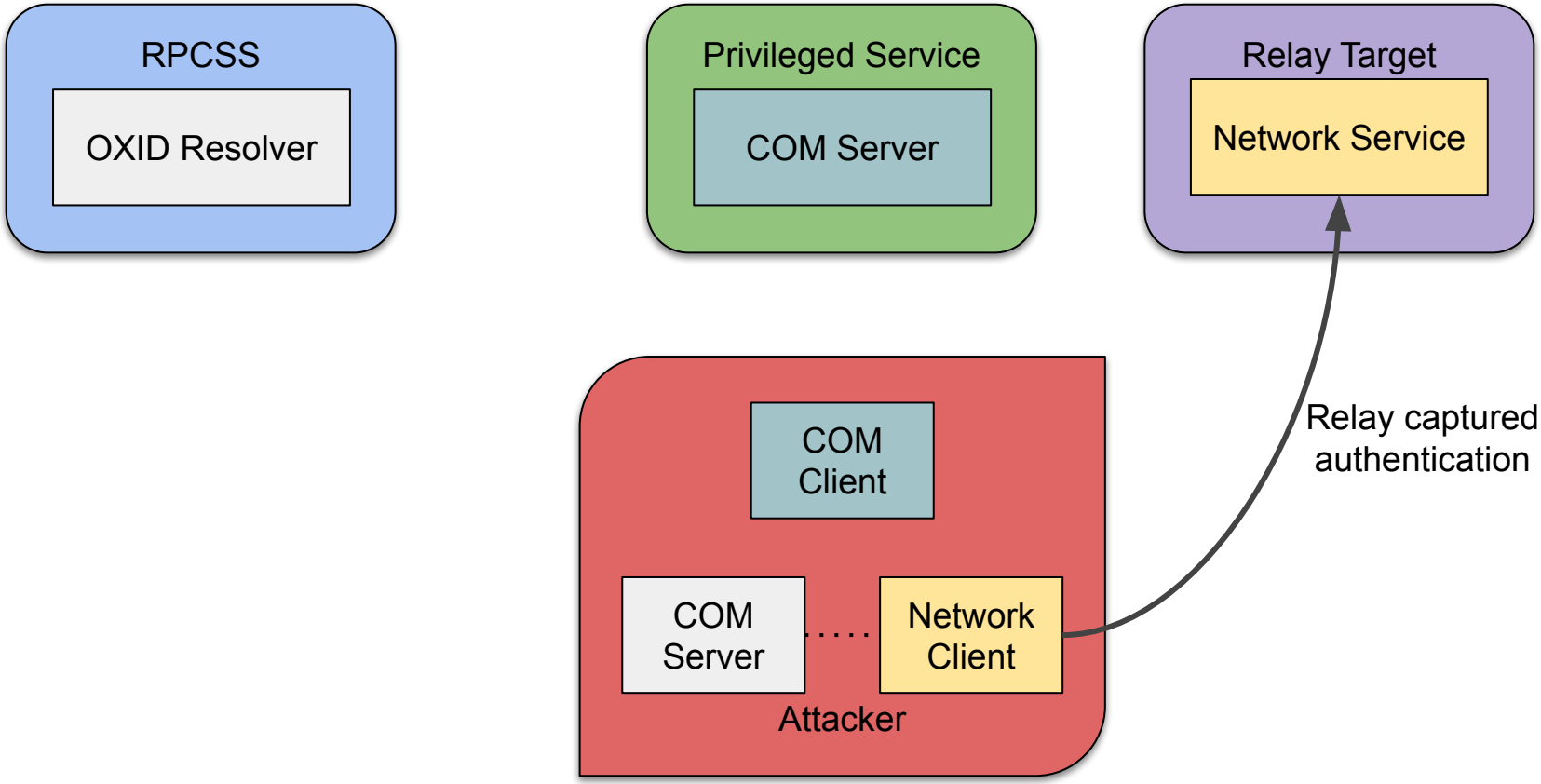
DCOM Relay


RPCSS
OXID Resolver
Resolve
OBJREF
Privileged Service
COM Server
Relay Target
Network Service
COM
Client
COM
Server
Attacker

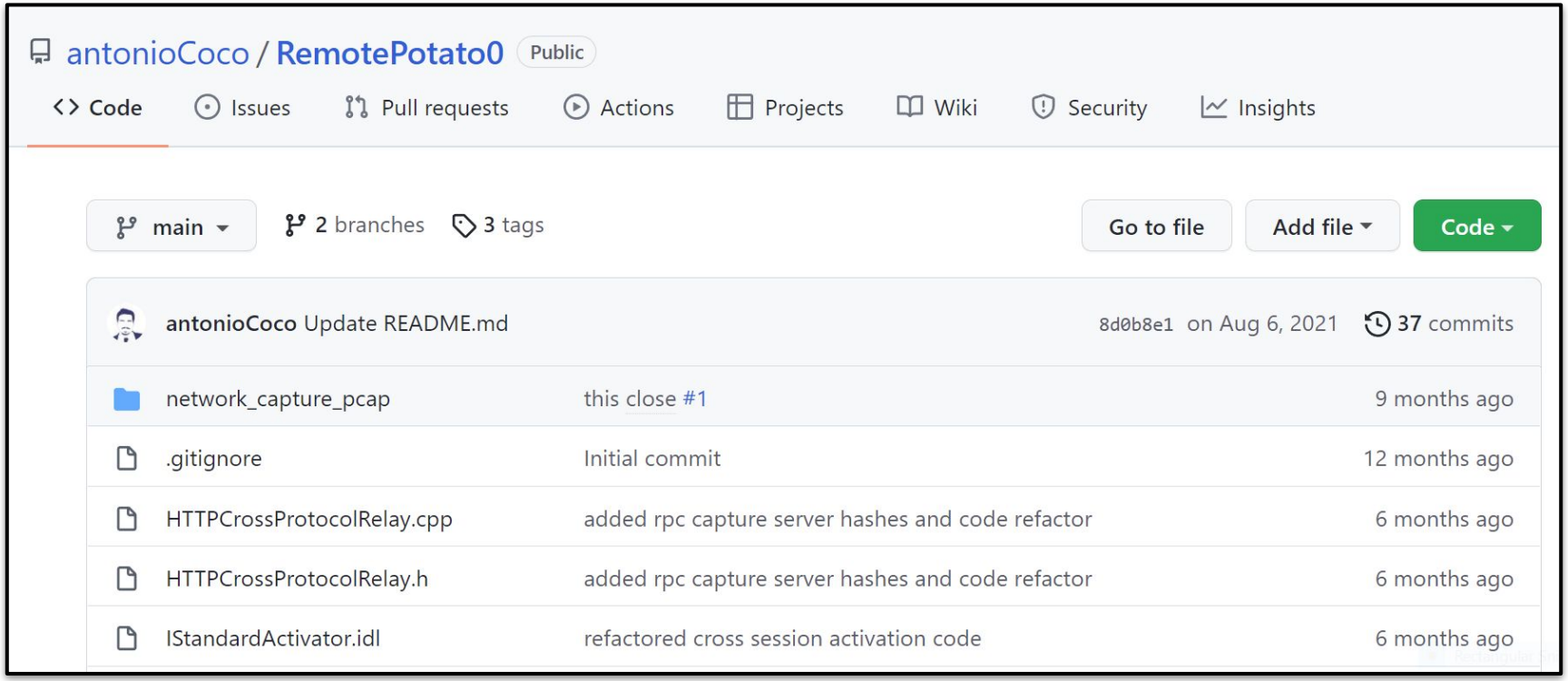
DCOM Relay


RPCSS
OXID Resolver
Privileged Service
COM Server
Relay Target
Network Service
Connect to
"remote" Server
COM
Client
COM
Server
Attacker

Complex Relay



DCOM RemotePotato



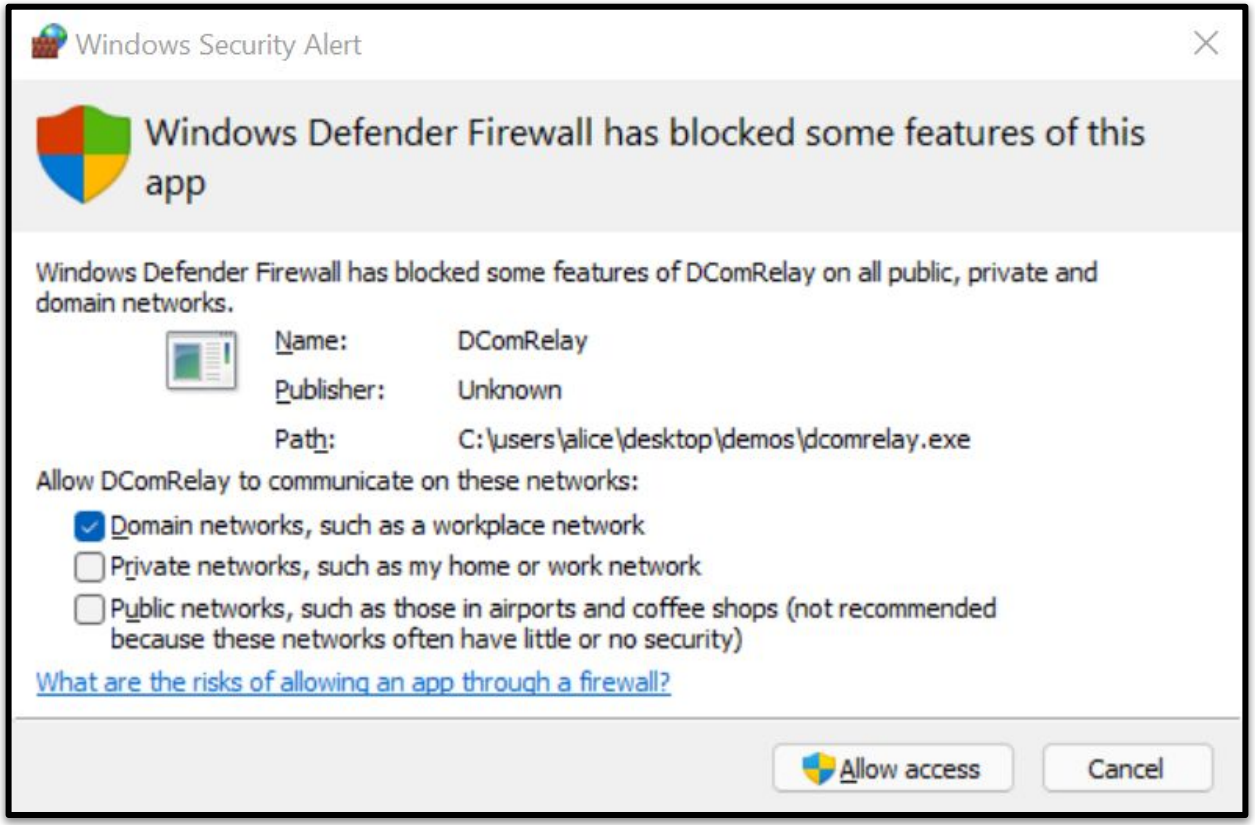
https://github.com/antonioCoco/RemotePotato0

Setting COM Server Security

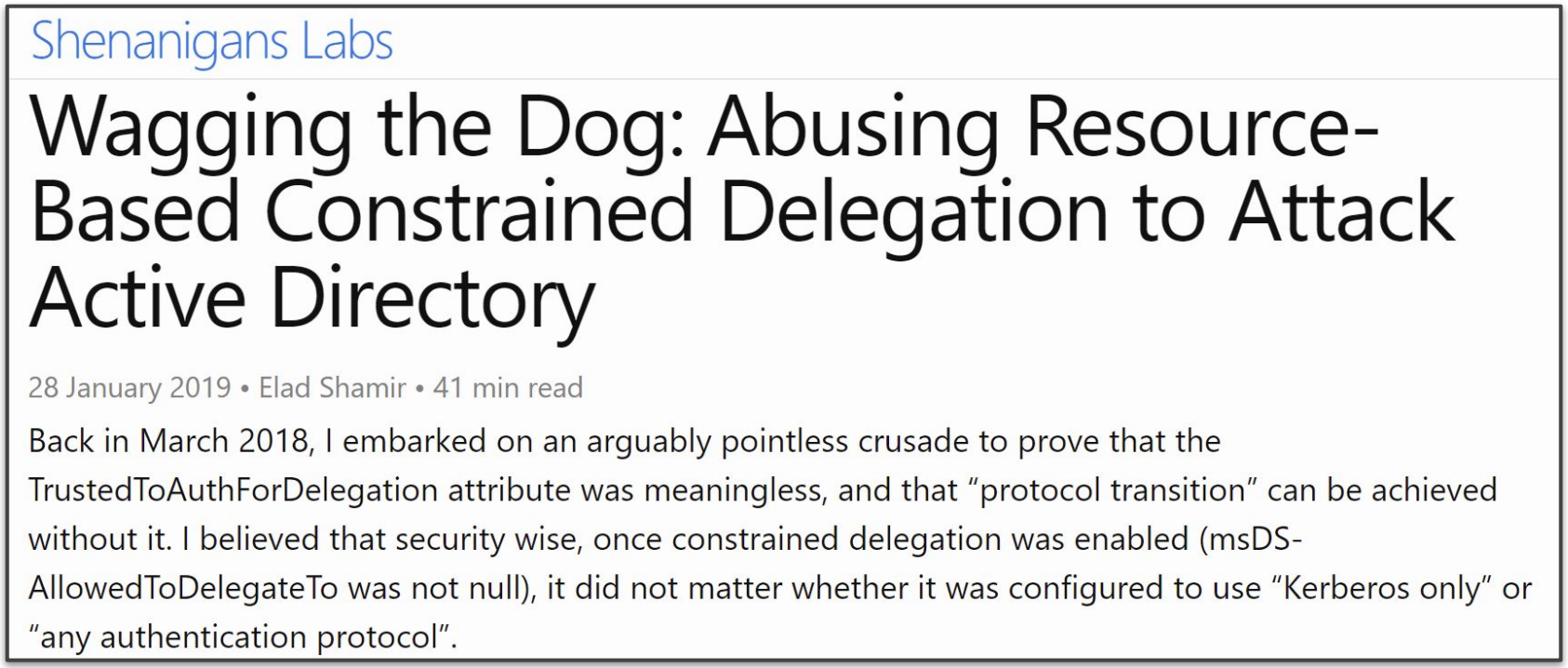
```
SOLE_AUTHENTICATION_SERVICE auth[] = {
    { RPC_C_AUTHN_GSS_KERBEROS,
      RPC_C_AUTHZ_NONE,
      L"HOST/PRIMARYDC" }
};

CoInitializeSecurity(nullptr, _countof(auth),
   auth, nullptr, RPC_C_AUTHN_LEVEL_CONNECT,
   RPC_C_IMP_LEVEL_IMPERSONATE, nullptr,
   EOAC_NONE, nullptr);
```

Bypass Firewall Restrictions



Resource Based Constrained Delegation



Shenanigans Labs

Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory

28 January 2019 • Elad Shamir • 41 min read

Back in March 2018, I embarked on an arguably pointless crusade to prove that the TrustedToAuthForDelegation attribute was meaningless, and that "protocol transition" can be achieved without it. I believed that security wise, once constrained delegation was enabled (msDS-AllowedToDelegateTo was not null), it did not matter whether it was configured to use "Kerberos only" or "any authentication protocol".

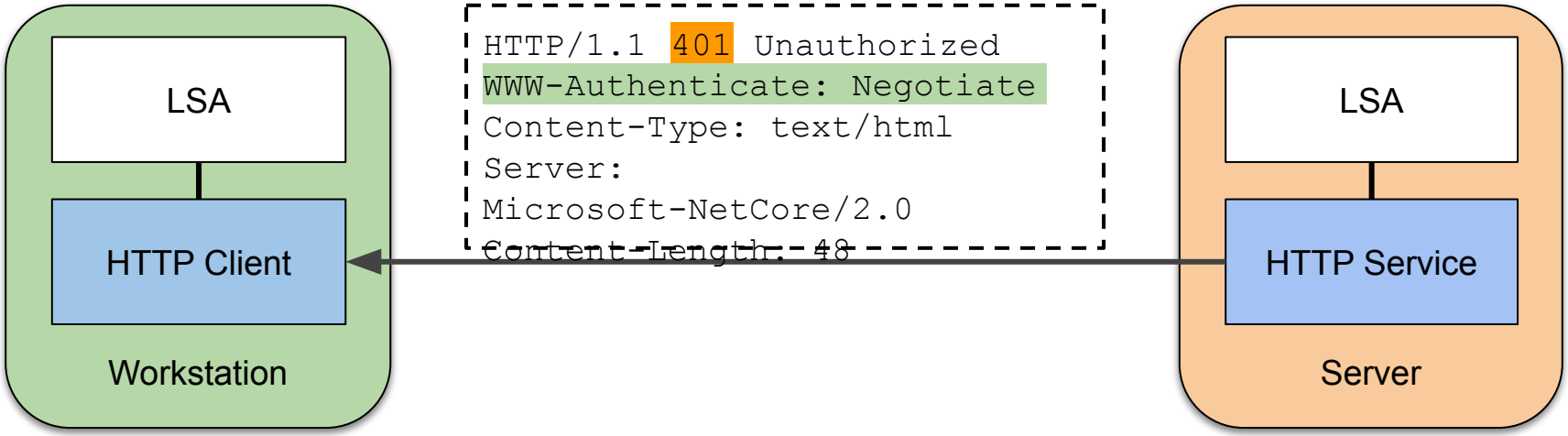
https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html#case-study-2-windows-1020162019-lpe

DEMO

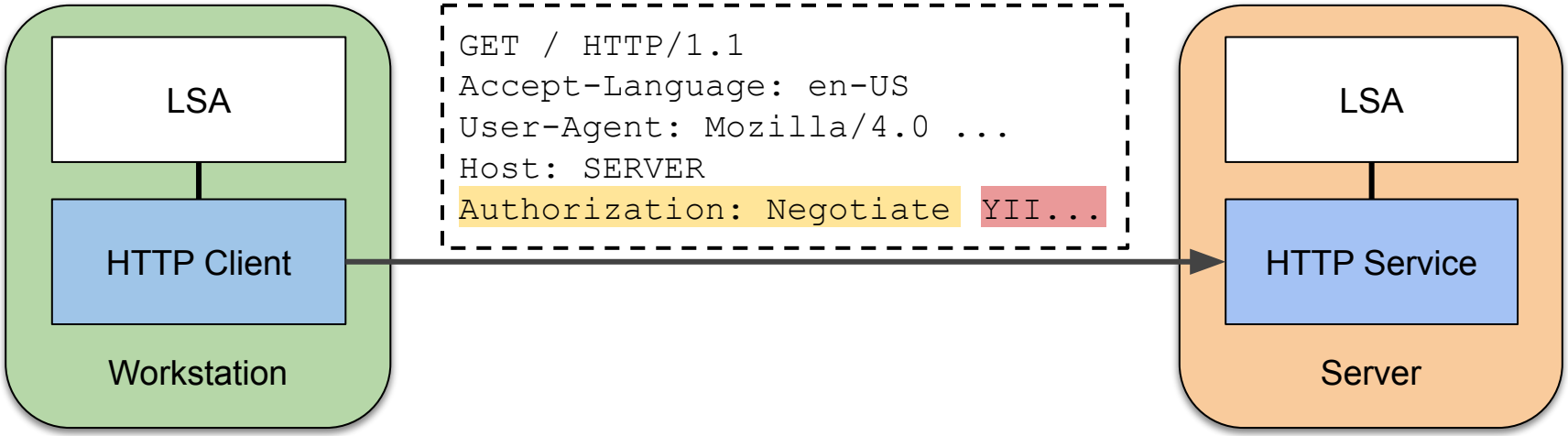
HTTP Authentication



HTTP Authentication



HTTP Authentication



HTTP Negotiate (RFC4559)

Network Working Group — K. Jaganathan
Request for Comments: 4559 — L. Zhu
Category: Informational — J. Brezak
Microsoft Corporation
June 2006

SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows

...

When the Kerberos Version 5 GSSAPI mechanism [RFC4121] is being used, the HTTP server will be using a principal name of the form of "HTTP/hostname".

WinINET

```
void AUTHCTX::GenerateFQDNAndSPN(const char *url) {
  LPWSTR fqdn;
  GetFqdn(url, &fqdn);
  LPWSTR cname;
  if (GlobalUseCNameForSPN && fqdn) {
    cname = fqdn;
  } else {
    GetCanonicalName(url, &cname);
  }
  wsprintf(this->SPN, L"HTTP/%s", cname);
}
```

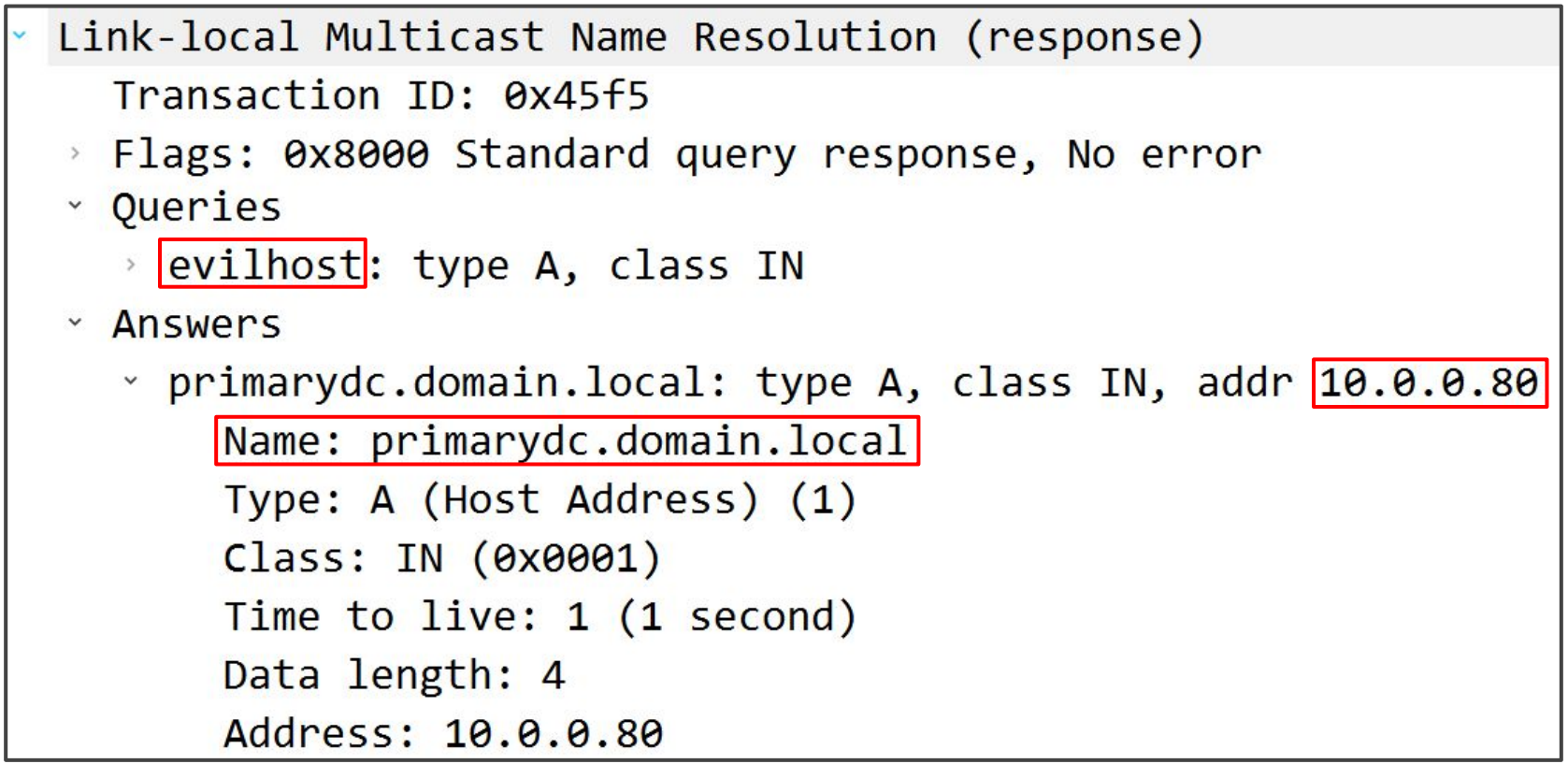
Resolving Names

```
void CAddressList::ResolveHost(LPCWSTR hostname) {
  ADDRINFOEXW hints = {};
  hints.ai_flags = AI_FQDN | AI_CANONNAME;
  PADDRINFOEX2W addr = nullptr;
  int i = 0;
  if (GetAddrInfoExW(hostname, NULL, NS_DNS, NULL, &hints,
       &addr, NULL, NULL, NULL, NULL) == NO_ERROR)
  {
      this->FQDN = addr->ai_fqdn;
      this->CName = addr->ai_canonname;
      this->Address = addr->ai_addr;
  }
}
```

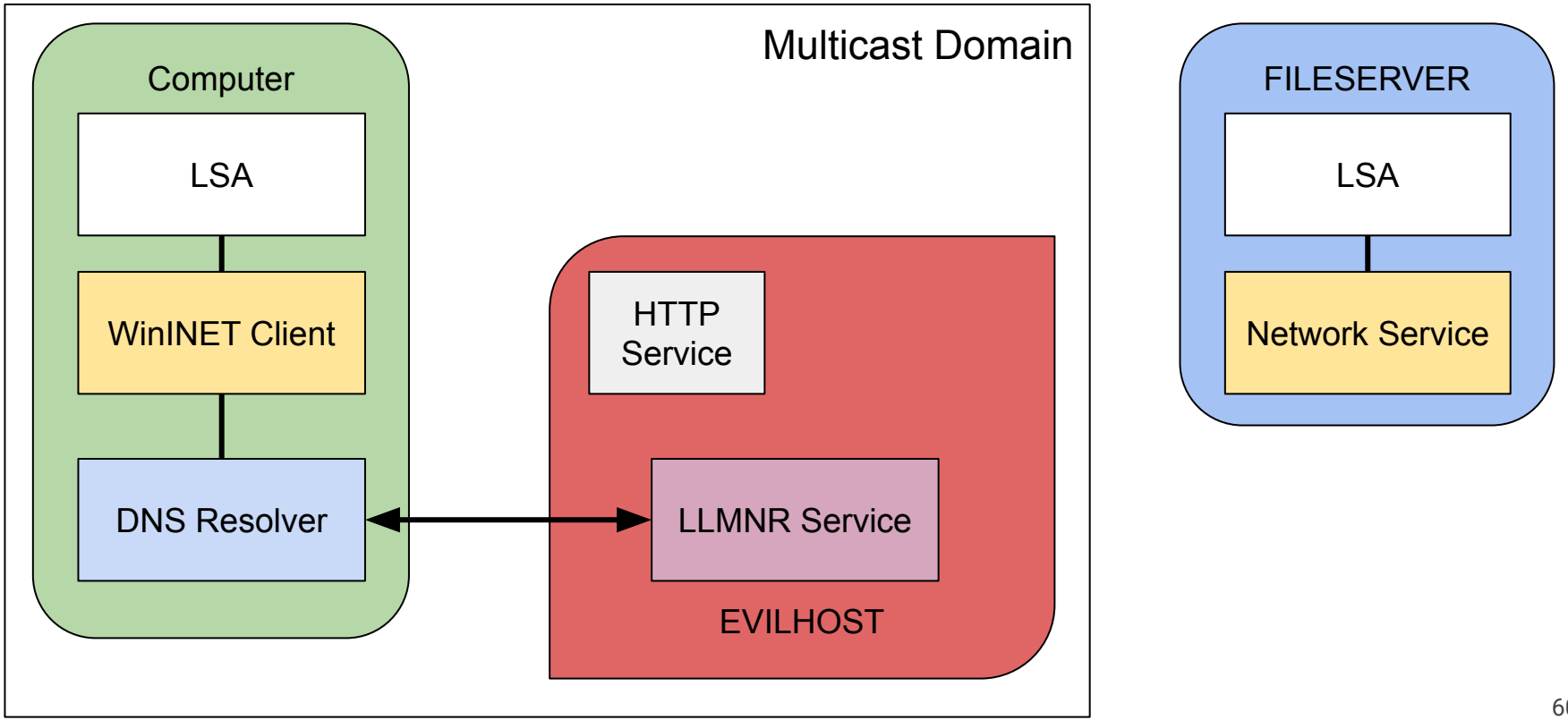
Canonical Name?

```
▾ Queries
  › testme.domain.local: type A, class IN
▾ Answers
  › testme.domain.local: type CNAME, class IN, cname primarydc.domain.local
  ▾ primarydc.domain.local: type A, class IN, addr 10.0.0.10
      Name: primarydc.domain.local
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 3600 (1 hour)
      Data length: 4
      Address: 10.0.0.10
```

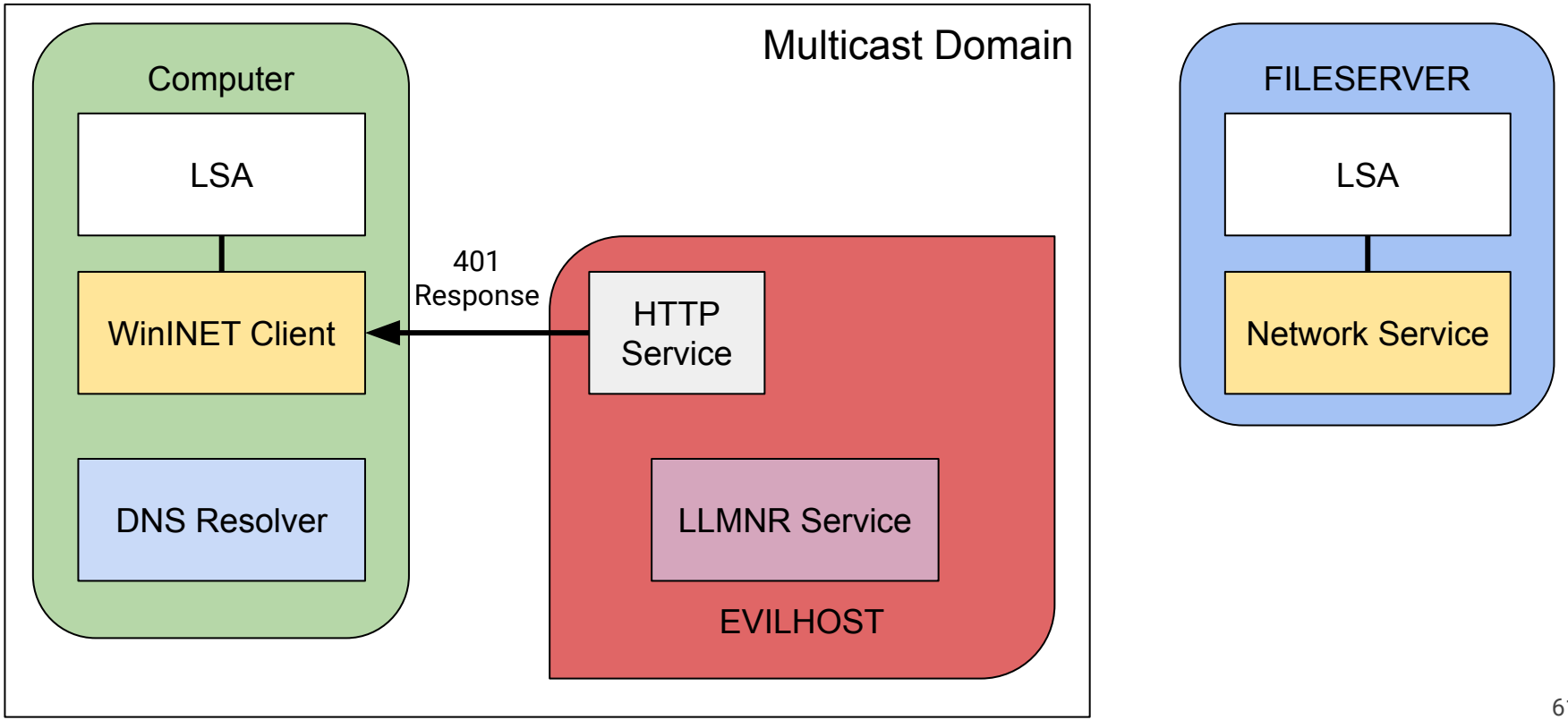
LLMNR To the Rescue



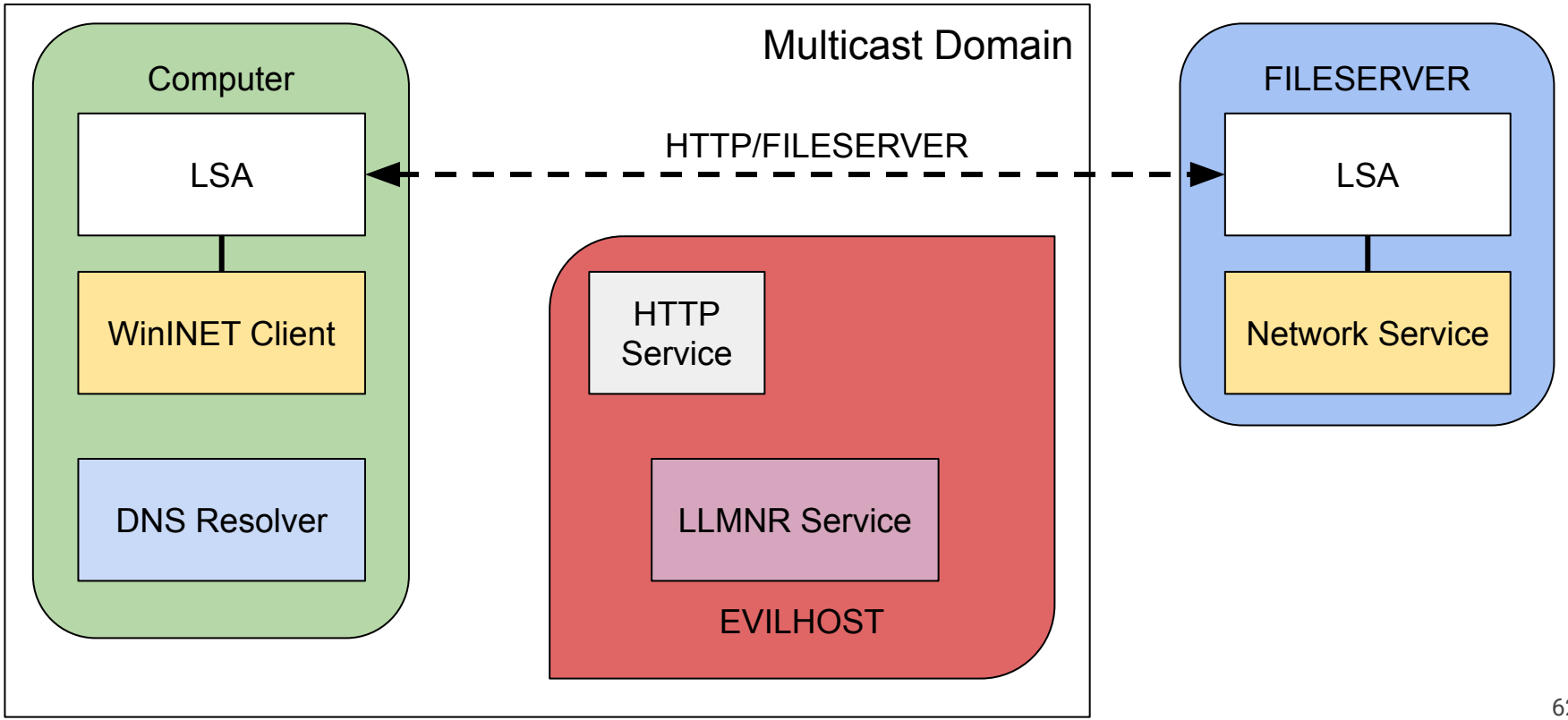
LLMNR Spoofing



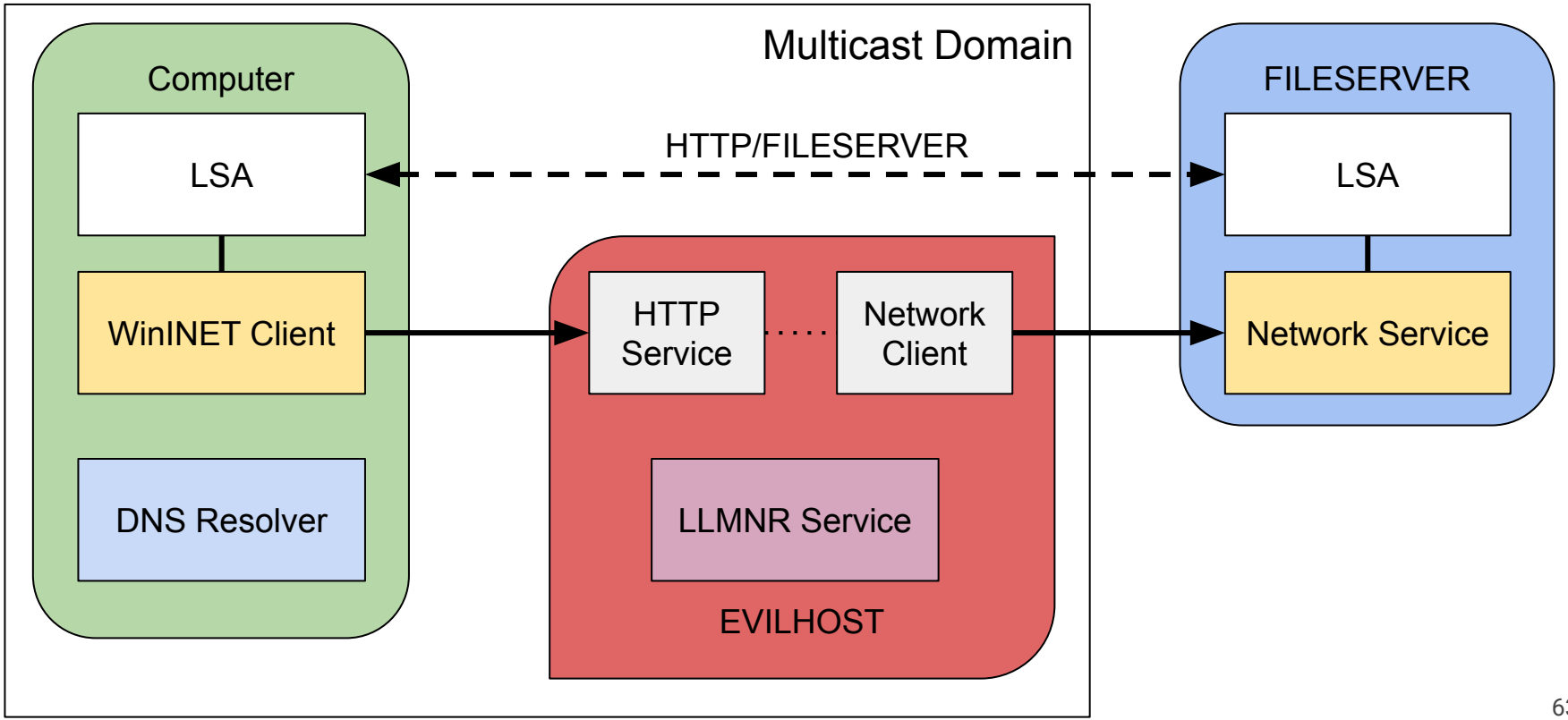
LLMNR Spoofing



LLMNR Spoofing



LLMNR Spoofing



DEMO

Chromium Browsers

```
int HttpAuthNegotiate::DoResolveCanonicalName() {
  if (http_auth_prefs_->NegotiateDisableCnameLookup())
    return OK;

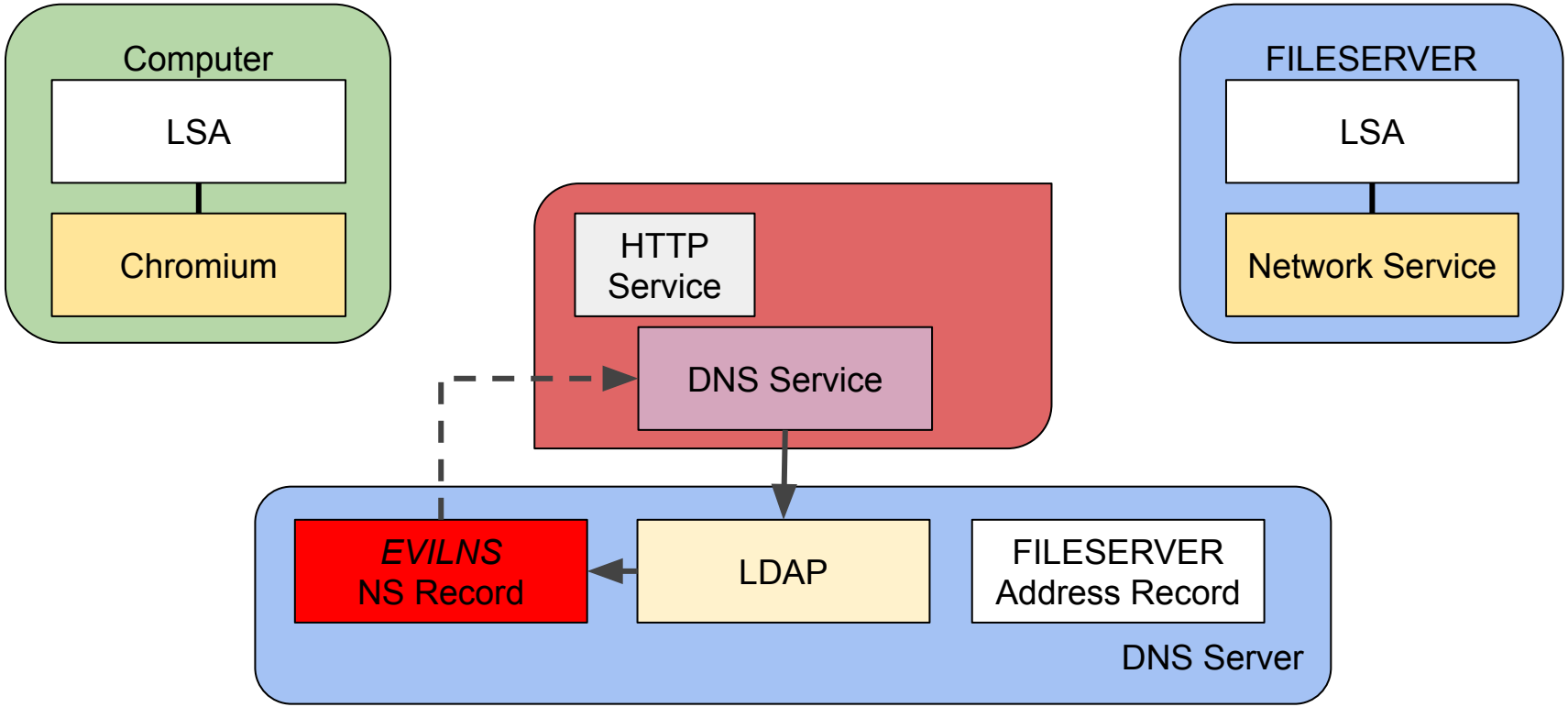
  HostResolver::ResolveHostParameters parameters;
  parameters.include_canonical_name = true;
  host_request_ = resolver_->CreateRequest(
      scheme_host_port_, parameters);
  return host_request_->Start();
}
```

Handling Name Lookup

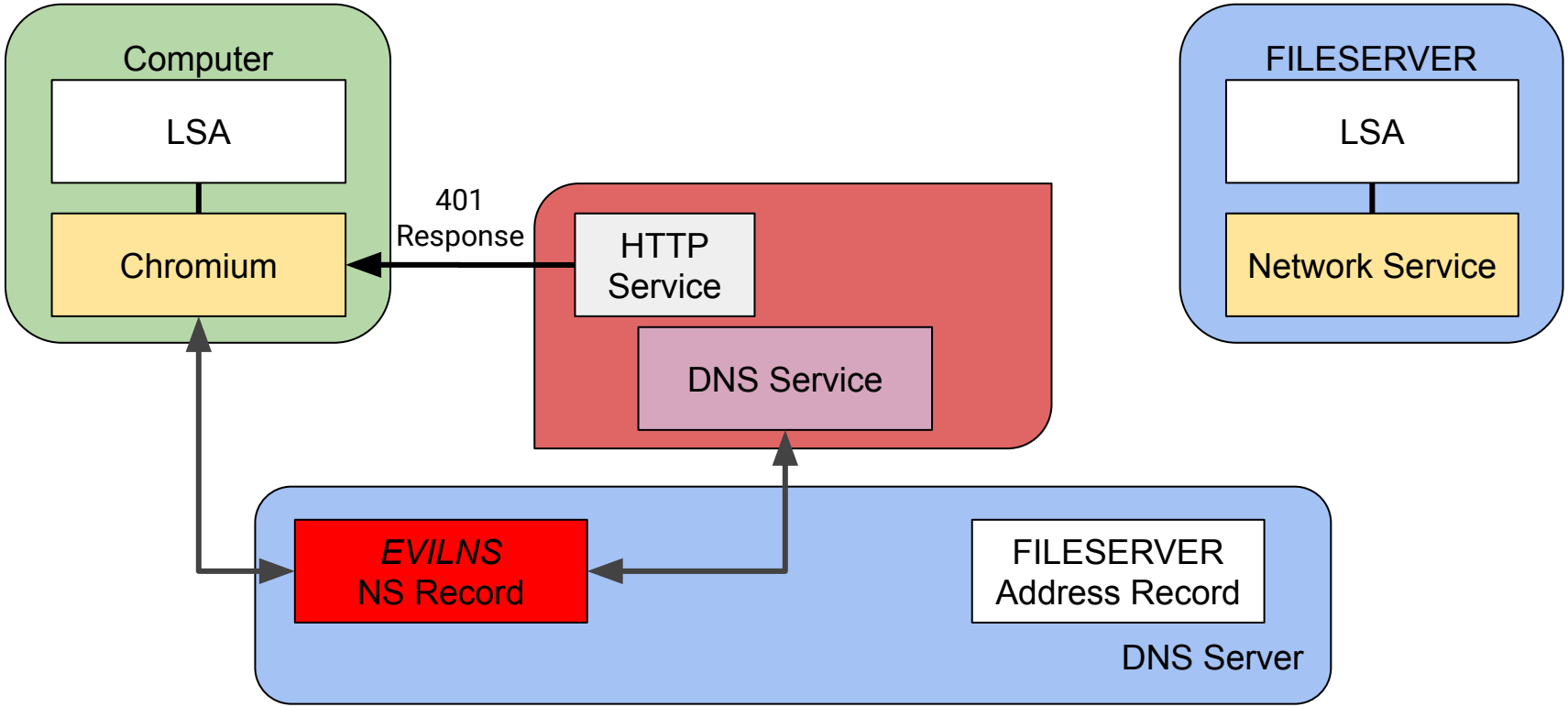
```
int HttpAuthNegotiate::DoResolveCanonicalNameComplete() {
  std::string server = scheme_host_port_.host();
  if (host_request_) {
    if (!host_request_->GetDnsAliasResults()->empty()) {
      server = *host_request_->GetDnsAliasResults()->begin();
    }
  }

  spn_ = base::StringPrintf("HTTP/%s", server.c_str());
  return rv;
}
```

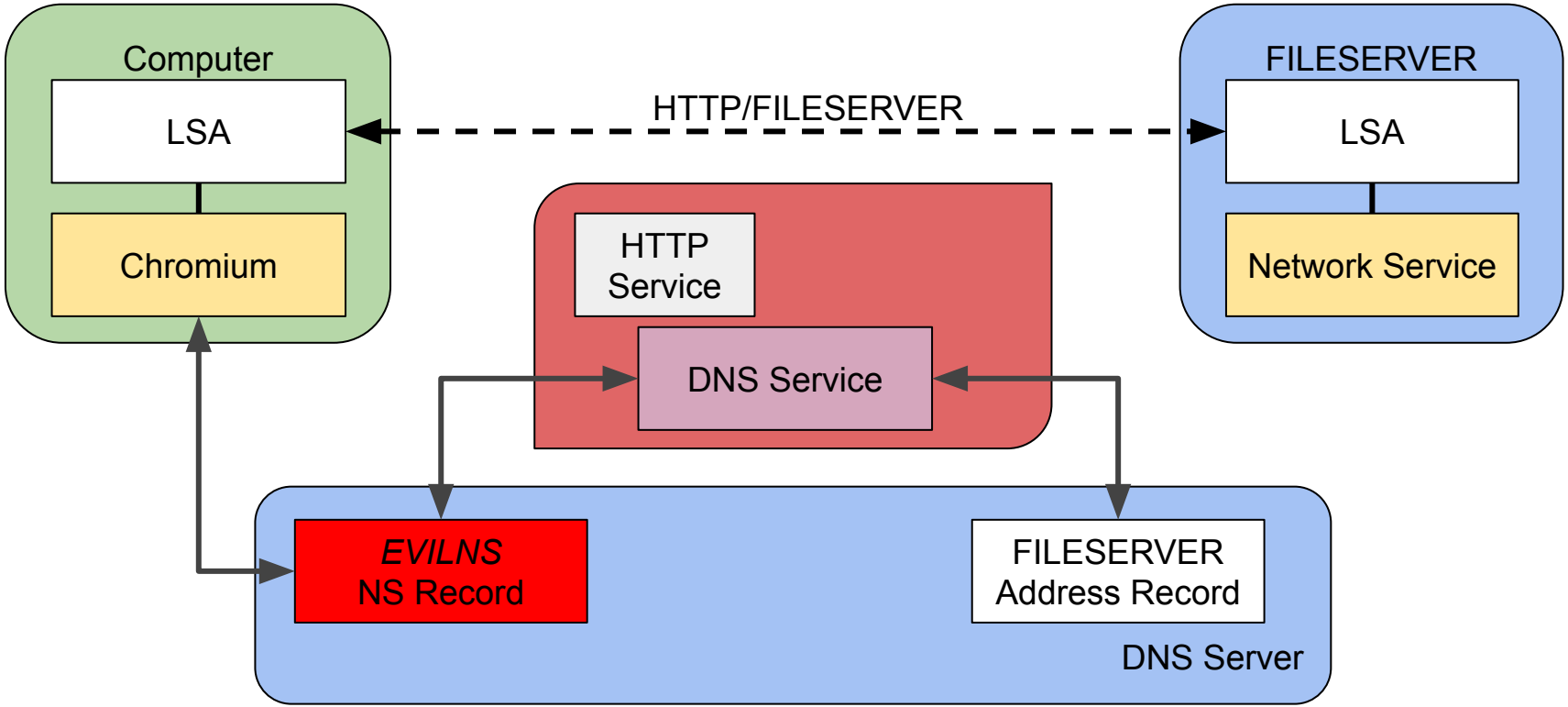
DNS CNAME Spoofing



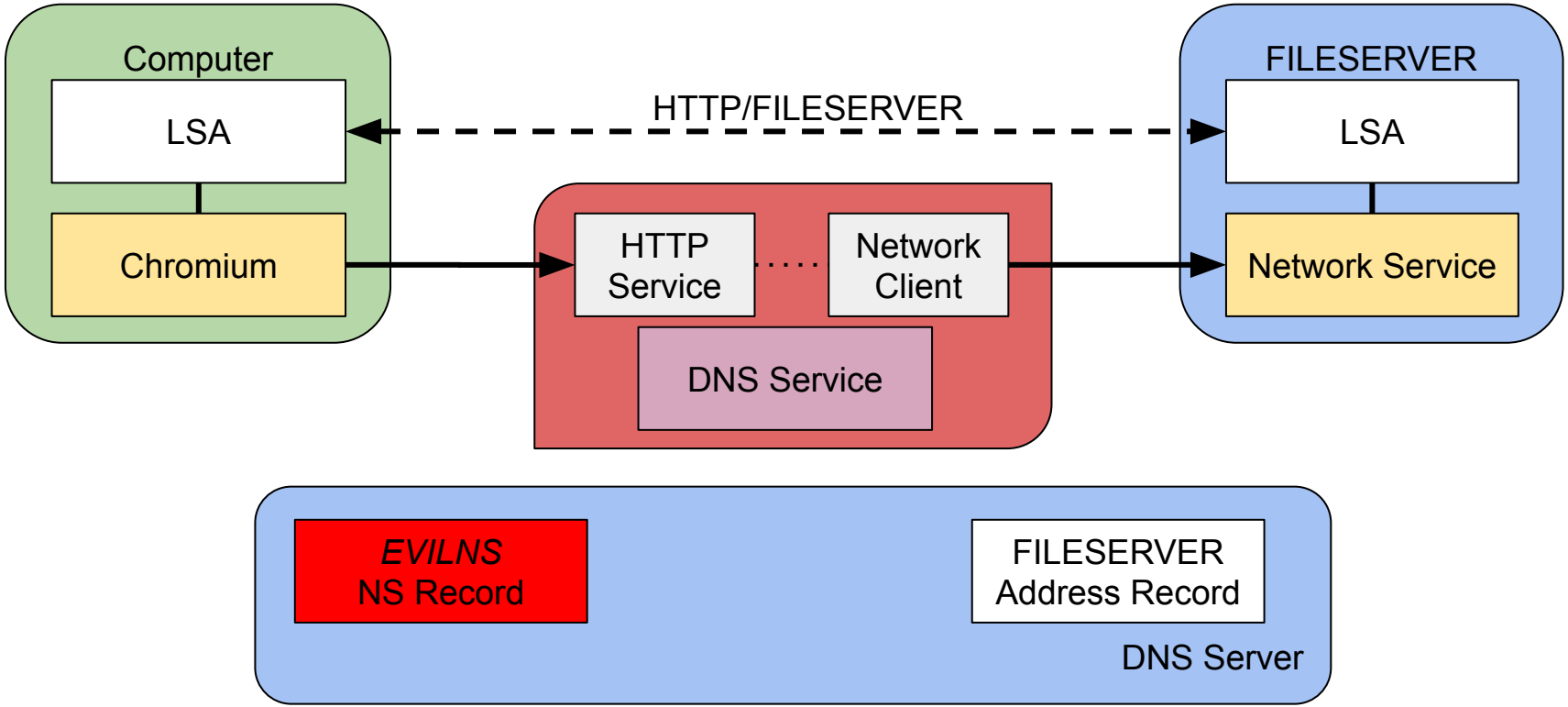
DNS CNAME Spoofing



DNS CNAME Spoofing



DNS CNAME Spoofing



Disabling Integrity

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Kerberos
Content-Type: text/html
Server: Microsoft-NetCore/2.0
Content-Length: 48
```

Using
Kerberos SSP

Specifying
Initial Token

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Negotiate AAFA
Content-Type: text/html
Server: Microsoft-NetCore/2.0
Content-Length: 48
```

HTTP Results

User Agent	*LLMNR*	*DNS CNAME*	*Delegation*	*Disable Integrity*
Internet Explorer 11 (WinINET)	Yes	No	Yes	No
WebDAV (WinHTTP)	Yes	No	Yes	No
Chromium (M93)	Yes	Yes	No	No
Firefox 91	Yes	Yes	No	No
.NET Framework 4.8	Yes	No	Yes	Yes
.NET 5.0	Yes	Yes	No	Yes
.NET 6.0	Yes	Yes	No	No

DEMO

Weird SMB Client Behavior

```
NTSTATUS BuildSpn(PSMB_CONTEXT Context,
                  PUNICODE_STRING *Spn) {
  ULONG TotalSize;

  return SecMakeSPNEx2(&CifsServiceName,
                       &Context->Hostname,
                       NULL, 0, NULL, NULL,
                       Spn, &TotalSize,
                       TRUE, FALSE);
}
```

Weird SMB Client Behavior

```
NTSTATUS BuildSpn(PSMB_CONTEXT Context,
                  PUNICODE_STRING *Spn) {
  ULONG TotalSize;

                                  &Context->HostName,
                                  NULL, 0, NULL, NULL,
                                  Spn, &TotalSize,
                                  TRUE, FALSE);
}
```

```
cifs/fileserver1UWhRCAAAAAAAAAAUAAAAAAAAAAAAAAAAAAAAfileserversBAAAA
```

LSA Service

```
NTSTATUS SspiExProcessSecurityContext(...) {
  // ...
  if (TargetName) {
    ULONG ActualSize;
    if (NT_SUCCESS(CredUnmarshalTargetInfo(
                        TargetName->Buffer,
                        TargetName->Length, NULL,
                        &ActualSize))) {
      TargetName->Length -= ActualSize;
    }
  }
  // ...
}
```

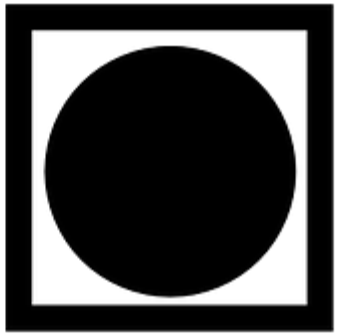
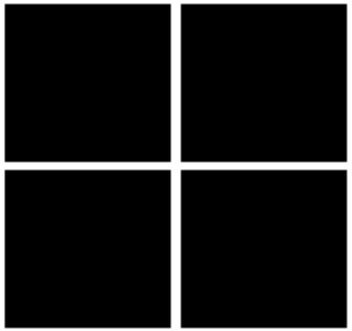
LSA Service

```
NTSTATUS SspiExProcessSecurityContext(...) {
  // ...
  if (TargetName) {
    ULONG ActualSize;
    cifs/fileserver1UWhRCAAAAAAAAAAUAAAAAAAAAAAAAAAAAAAAAfileserversBAAAA
                                  TargetName->Length, NULL,
                                  ) {
      TargetName->  cifs/fileserver  ize;
    }
  }
  // ...
}
```

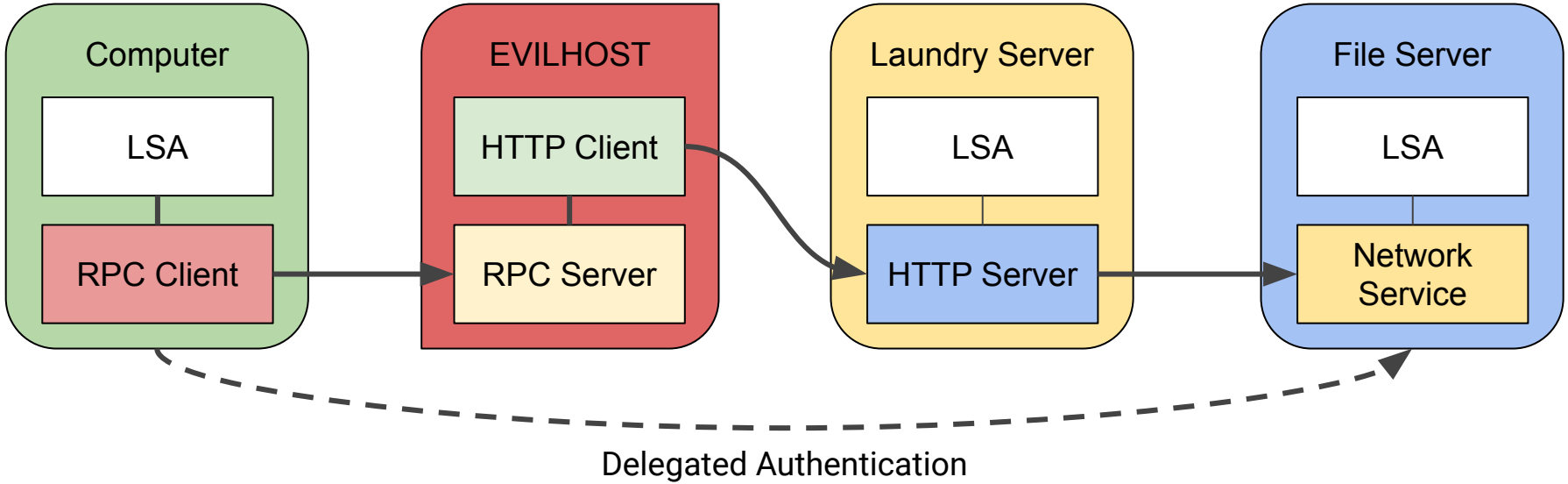
DEMO

Kerberos Ticket Laundry



Kerberos Delegation



EFSRPC (CVE-2021-43893)

Issue 2228: Windows: EFSRPC Arbitrary File Upload EoP

Reported by forshaw@google.com on Fri, Sep 17, 2021, 6:40 PM PDT

```
Windows: EFSRPC Arbitrary File Upload EoP
Platform: Windows Server 2019, 2022
Class: Elevation of Privilege
Security Boundary: User

Summary: The EFSRPC service doesn't prevent a caller specifying a local device path
allowing any authenticated user to upload arbitrary files to a server.

Description:
Note this is based on the "authenticated" EFSRPC service in efssvc.dll, not the down-level
service. This issue doesn't work with the down-level service after the fix for PetitPotam.
Also this bug is IMO already in the public domain as it was reported as part of MSRC case
51270 (https://bugs.chromium.org/p/project-zero/issues/detail?id=1817) but it seems you didn't fix
this.
```

https://bugs.chromium.org/p/project-zero/issues/detail?id=2228

DEMO

Protocols I didn't look at

- NFS (some versions support Kerberos authentication)
- RDP (well other than the demo)
- WinRM
- NegotiateStream
- Custom third party stuff.
- Interactions with Kerberos on non-Windows systems

Administrator Protections

- Disable Windows authentication HTTP endpoints.
  - Enable HTTPS with EPA
- Disable CNAME DNS lookup for Chromium browsers
- Keep disabling NTLM

Thanks

- Will Schroeder + Lee Christensen (ADCS + Rubeus)
- Antonio Cocomazzi + Andrea Pierini
- Dirk-jan Mollema
- Elad Shamir
- OffensiveCon organizers.