
# LSA Whisperer

"You can get what you want
if you ask nicely."

# Introductions

- Systems developer
- On the “Internal and Community Products” team
- Special thanks to Elad, Lee, Will, Daniel, and Kai




# Acknowledgements


- Adam Chester (@_xpn_)
- Alberto Solino (@agsolino)
- Alex Ionescu (@aionescu)
- Alex Short (@alexsho71327477)
- Benjamin Delpy (@gentilkiwi)
- Charlie Clark (@exploitph)
- Dirk-jan Mollema (@_dirkjan)
- James Forshaw (@tiraniddo)
- Mor Rubin (@rubin_mor)
- Dr. Nestori Syynimaa (@DrAzureAD)
- Passcape Software (passcape.com)
- Steve Syfuhs (@SteveSyfuhs)

# Problem Statement

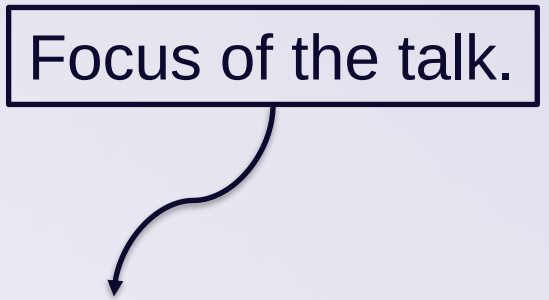
## Why do I care?

Accessing LSASS memory is a common goal to recover user credential material. Multiple features make gaining and abusing access more difficult:

- Credential Guard
- Remote Credential Guard
- Protected Processes Light (PPL)

# Solution

**What this talk is about.**




Request credentials from the LSA directly via Authentication Package Calls:

1. Used for years for Kerberos ticket recovery
2. Mitigations are irrelevant if LSA grants a requester credential access
3. Largely unexplored by non-Microsoft developers

# Part 1
# LSA Internals

# Part 2
# The APs

# LSA Internals

The relevant parts

# Security Support Providers (SSPs)

### Authentication Packages (APs)

1. Implement authentication logic
2. Maintains logon session information
3. Must implement at least one AP callback functions
   (ex. **LsaApLogonUser**)

### Security Packages (SPs)

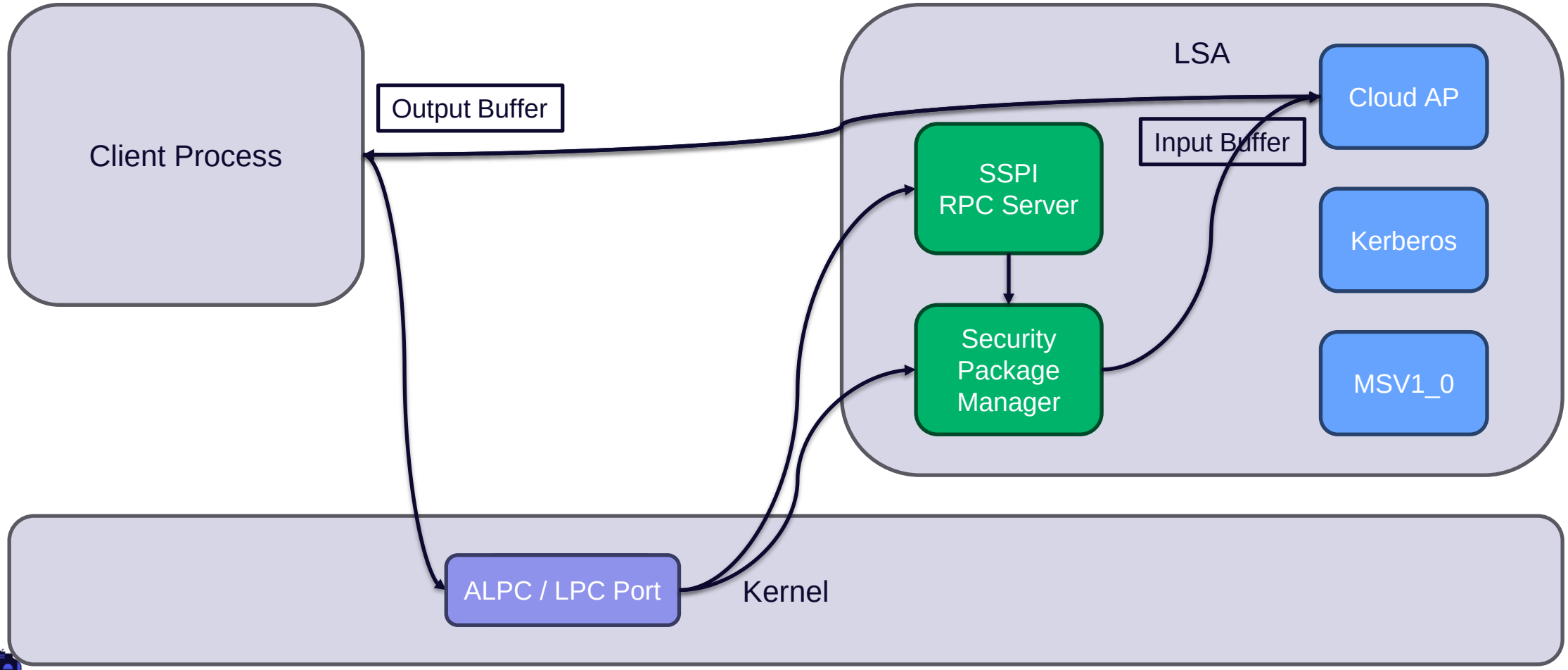
1. Implement a security protocol
2. Must implement at least one SP callback functions
   (ex. **SpAcceptCredentials**)

https://learn.microsoft.com/en-us/windows/win32/secauthn/authentication-functions

| Dll | Common Name | SP | AP | RPC ID | RPC Authn |
|---|---|---|---|---|---|
| cloudap | Cloud AP | OAuth 2.0 | ✔ | 36 | `CLOUD_AP` |
| credssp | Credential Delegation SSP | TLS+SPNEGO | — | | |
| kerberos | Kerberos | Kerberos | ✔ | 16 | `GSS_KERBEROS` |
| livessp | Live SSP | ? | ✔ | 32 | `LIVE_SSP` |
| msapsspc | DPA Client | RPA | — | 17 | `DPA` |
| msnsspc | MSN Client | NTLM | — | 18 | `MSN` |
| msv1_0 | Microsoft Authentication Package v1.0 | NTLM | ✔ | 10 | `WINNT` |
| negoexts | Negotiate Extender | NEGOEX | ✔ | 30 | `NEGO_EXTENDER` |
| negotiate | Negotiate | SPNEGO | ✔ | 9 | `GSS_NEGOTIATE` |
| pku2u | Public Key User to User | PKU2U | ✔ | 31 | `NEGO_PKU2U` |
| schannel | Secure Channel | SSL/TLS | ✔ | 14 | `GSS_SCHANNEL` |
| tspkg | Terminal Services Package | | ✔ | 22 | ? |
| wdigest | Windows Digest | Digest Access | ✔ | 21 | `DIGEST` |

# Authentication Package Calls

## LsaCallAuthenticationPackage



# The Authentication Packages

What each is and what calls they allow

# Kerberos

## Domain Logins

Useful AP call functionality:

- Host enumeration
- Ticket recovery
- Ticket usage (PTT)
- Ticket purging
- Domain name binding and pinning

**Related Commands**

```
AddBindingCacheEntry
PinKdc / UnpinAllKdc
PrintCloudKerberosDebug
Query/Purge BindingCache
Query/Purge KdcProxyCache
Query/Purge TicketCache[Ex|Ex2|Ex3]
QueryDomainExtendedPolicies
QueryS4U2ProxyCache
Retrieve[Encoded]Ticket
SubmitTicket
...
```

# Cloud AP

## Azure, AD FS, and Microsoft Accounts Logins

Useful AP call functionality:

- Host enumeration
- SSO cookie recovery
- Refresh PRT
- *Possible* PRT recovery

**Related Commands**

```
GetAuthenticatingProvider
GetDpApiCredKeyDecryptStatus
GetPrtAuthority
GetPwdExpiryInfo
GetTokenBlob
GetUnlockKeyType
IsCloudToOnPremTgtPresentInCache
RefreshTokenBlob
...
```

# Cloud AP Plugins

## Azure, AD FS, and Microsoft Accounts Logins

### AzureAD / AD FS Commands

```
Create[Device/Enterprise]SSOCookie
CreateNonce
DeviceAuth
DeviceValidityCheck
GetPrtAuthority
RefreshP2P[CA]Cert[s]
ValidateRdpAssertionRequest
...
```

### Microsoft Account Commands

TBD 🙂

# Microsoft Authentication Package V1.0

## Local Machine Logins

Useful AP call functionality:

- User/session enumeration
- DPAPI key recovery
- NTLMv1 response generation
- *Possible* NTLMv2 response generation

**Related Commands**

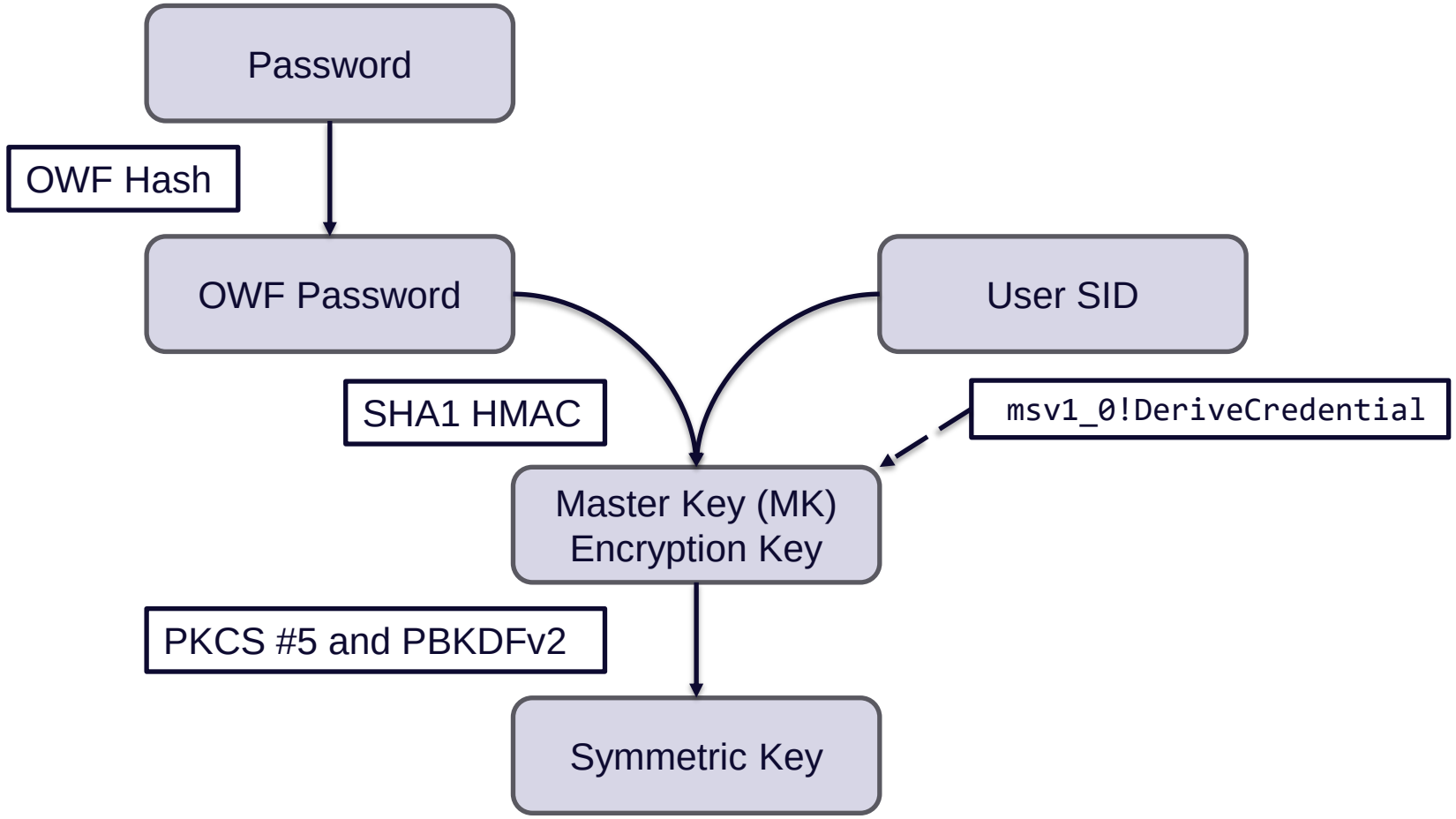
```
DeriveCredential
EnumerateUsers
Get[Strong]CredentialKey
GetUserInfo
Lm20GetChallengeResponse
...
```

https://eladshamir.com/2018/03/19/Internal-Monologue.html

# Side Quest

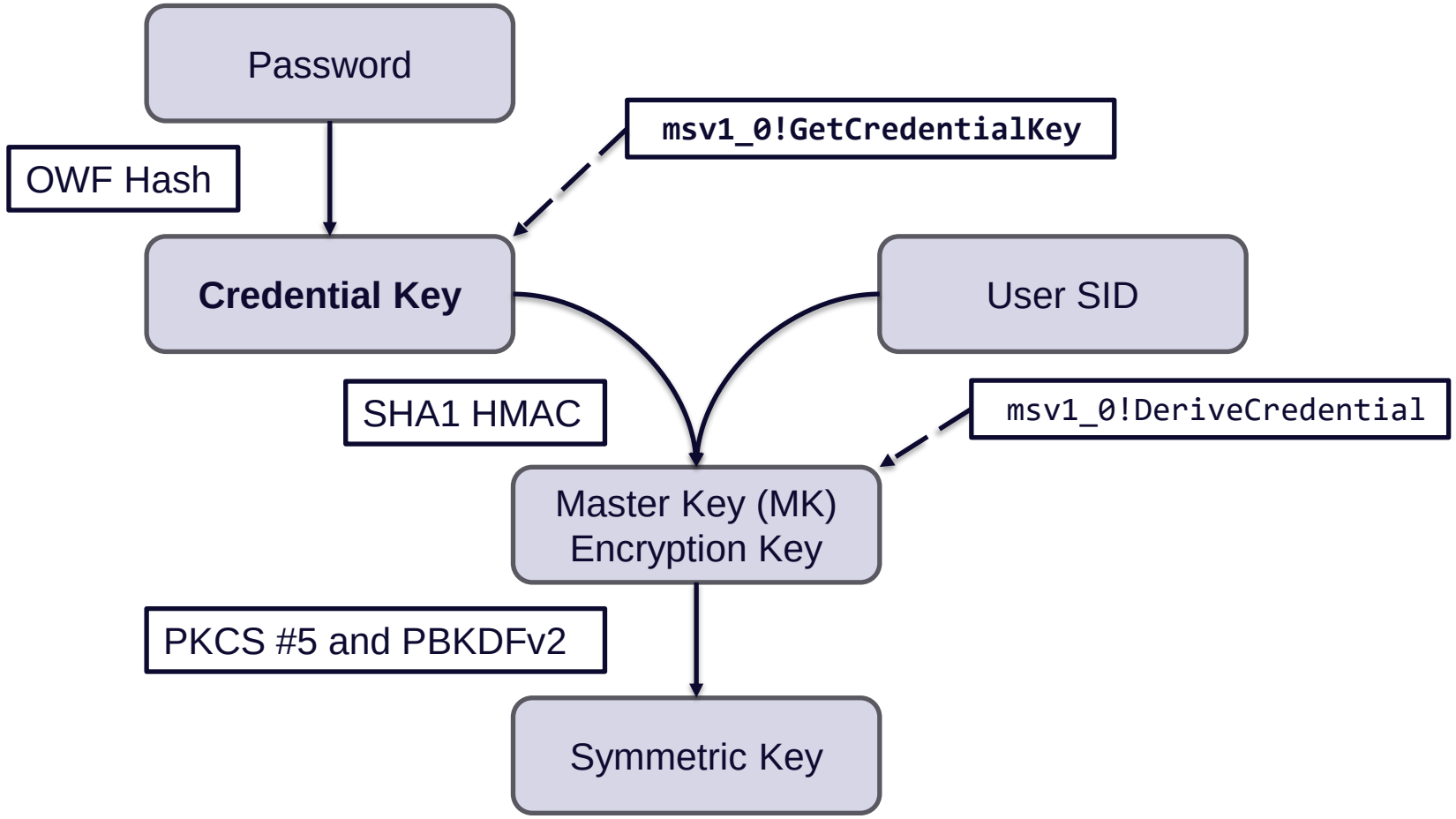
DPAPI Updates from NT 5.0 to NT 10 2004

# Key Derivation (NT 5.0)



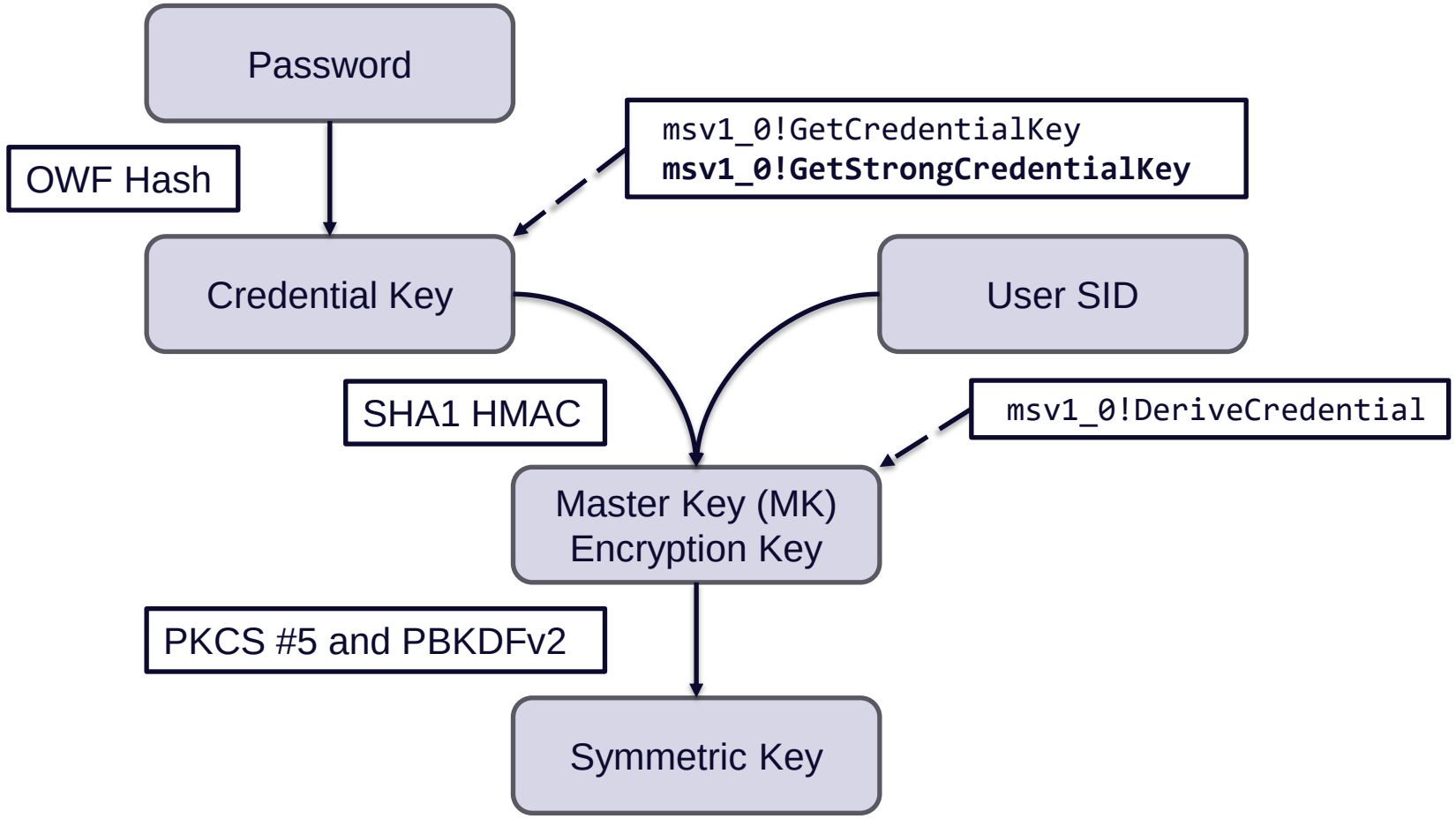
# Key Derivation (NT 6.2)

## Adds: GetCredentialKey



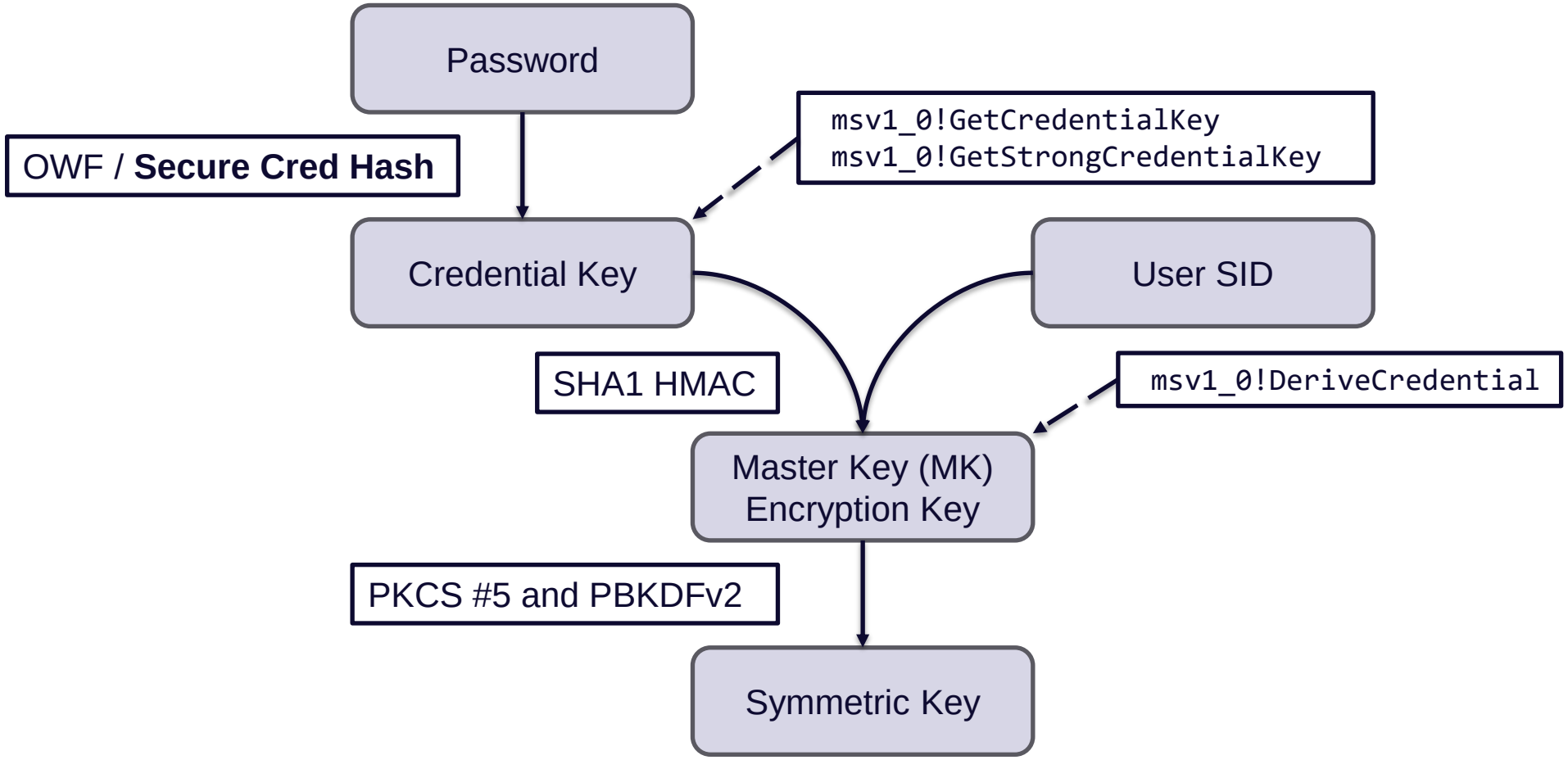
# Key Derivation (NT 6.4)

## Adds: GetStrongCredentialKey



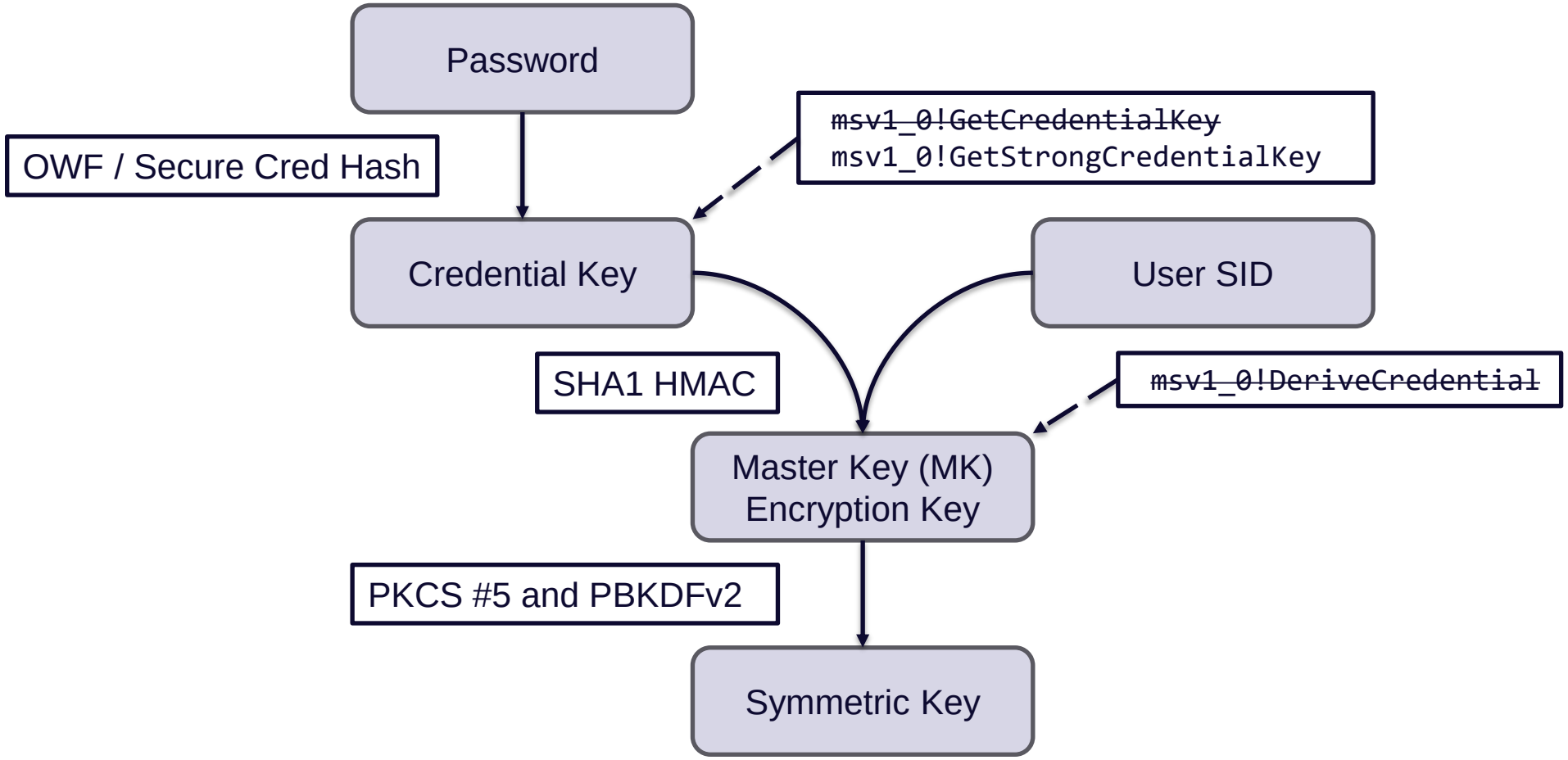
# Key Derivation (NT 10 1607)

## Adds: Secure Credential Hash



# Key Derivation (NT 10 2004)

## Adds: Credential Guard



# APs Continued

What each is and what calls they allow

# Public Key User to User

## Peer-to-Peer Logins

Useful AP call functionality:

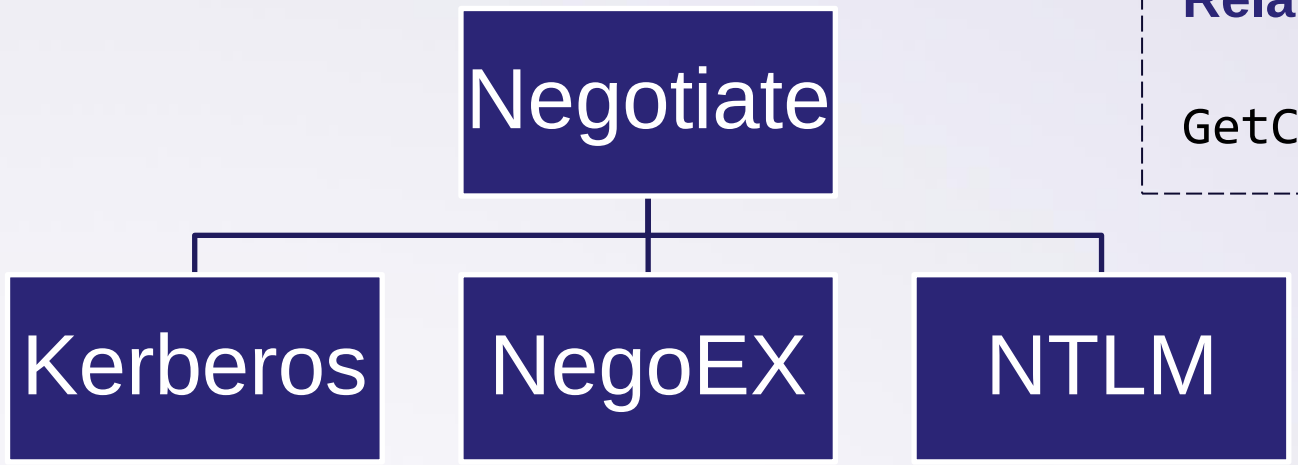
- Host enumeration
- Ticket purging

**Related Commands**

```
Query/Purge TicketCache
```

# Negotiate

## Logins Using a Negotiated Security Protocol

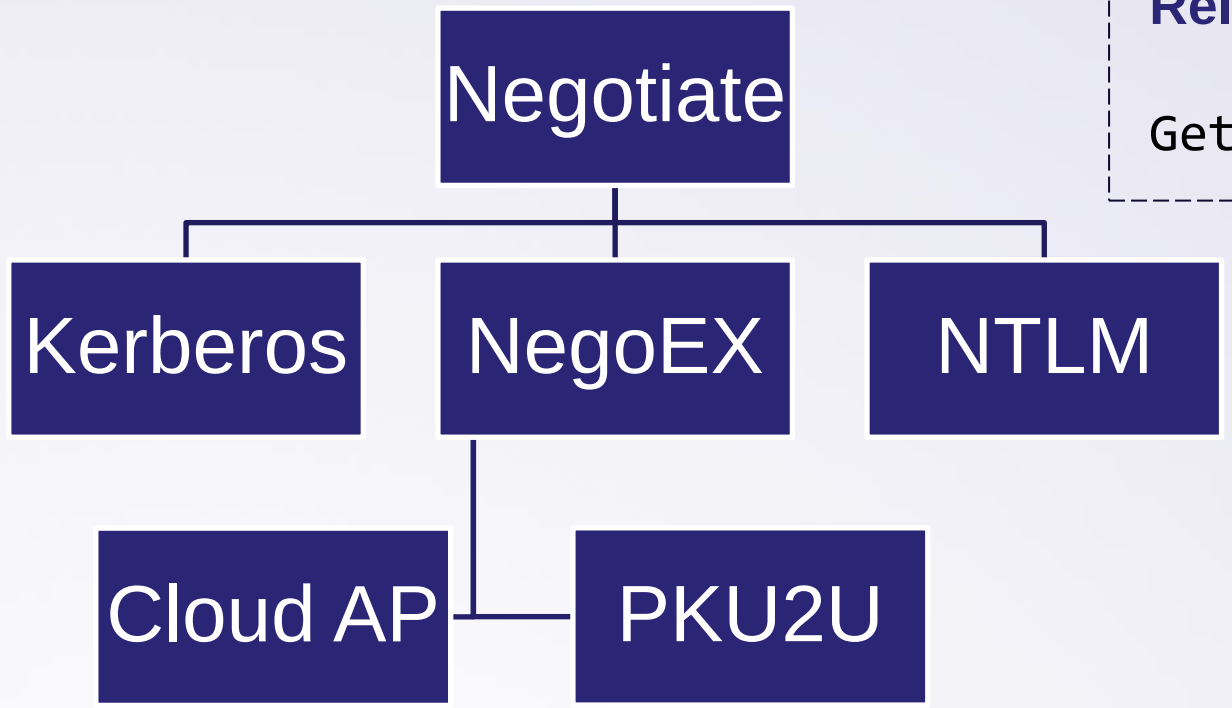



**Related Commands**

`GetCallerName`

# Negotiate Extender

## Logins Using a Negotiated Security Protocol




**Related Commands**

`GetCredUIContext`

# Secure Channel

## Session Layer Logins

Not too useful

# Terminal Service Package

## Remote Desktop Logins

Usefulness TBD 🙂

# WDigest

## Digest Logins

Does not support AP calls 😕

# Epilogue

# Mitigations

Monitoring

✅ ETW (needs more research)

Prevention

✅ Credential Guard (blocks some DPAPI credential recovery)

❌ RPC Filters

❌ RPC Firewall

# EDR Recommendations

Monitoring

- Monitor calls to `LsaCallAuthenticationPackage`
- Monitor for interaction with ALPC port `lsasspirpc`
- “RPC Call Interception” (Ionescu / Crowdstrike – WO 2015/084577 A1)

Prevention

- Block calls to LSASS RPC endpoints *other than* `lsasspirpc` for interface `4f32adc8-6052-4a04-8701-293ccf2096f0` (SSPI)

# Dead Ends

1. Dump credentials remotely with the SSPI RPC server
2. “LSA only” calls
   - Calls that check if the caller is the LSASS process (ex. `msv1_0!CacheLogon`)
   - Calls that require valid LSASS memory pointers (ex. `msv1_0!DecryptDpapiMasterKey`)
3. Using passthrough calls to bypass “is caller the LSASS process” checks
4. Transferring credentials

# Future Work

- Transferring Cloud AP credentials (`cloudap!TransferCreds`)
- AzureAD device authentication (`CreateDeviceSSOCookie → DeviceAuth`)
- AzureAD logon (`CreateNonce → ValidateRdpAssertion → LsaLogonUser`)
- Microsoft Account SSO cookie recovery
- MK encryption key recovery (`DeriveCredential → SharpDPAPI`)
- NTLMv1 / NTLMv2 relay (`Lm20GetChallengeResponse + impacket`)

# Future Work (Continued)

- Negotiate extender credential recovery (`GetCredUIContext`)
- Terminal Service Package interaction (24 total calls)
- Security Package Manager API interaction (23 total calls)
- IAKerb interaction, if Microsoft implements it as an AP
- The other half of AP functionality, the LogonUser API! 😄

# Wrap-Up

Multiple mitigations make it difficult to recover credentials by accessing LSASS memory.

LSA allows clients to directly request many of the credentials it manages.

- Requests may be done directly over RPC instead of the Win32 APIs
- Fills in host enumeration gaps
- Supports new tradecraft for credential abuse

Release of the tool and project wiki is coming soon! (~1 month)

SPECTEROPS

# Questions?

Evan McBroom | emcbroom@specterops.io


# Thank you!

Evan McBroom | emcbroom@specterops.io