Google

Our Relationships Need Improvement

James Forshaw - Google Project Zero

https://pixabay.com/illustrations/achievement-success-mountain-clouds-5597527/

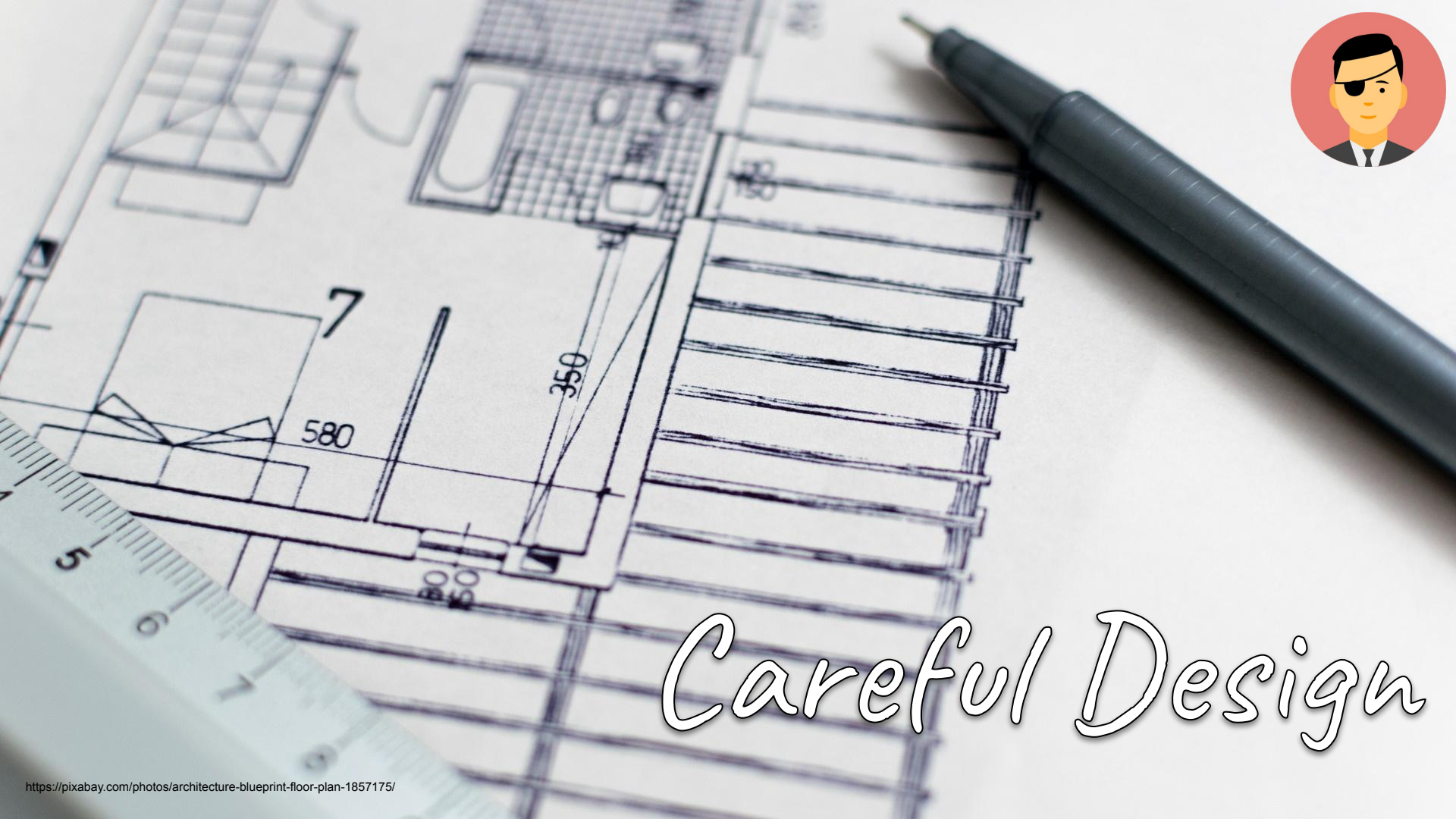
Who am I?

- Researcher in Project Zero
- Specializes in Windows
- Sometimes a vendor (Chrome)
- Opinionated!


Researcher
X
Vendor
https://pixabay.com/photos/laptop-apple-macbook-computer-2561018/
https://pixabay.com/photos/suit-business-man-business-man-673697/

https://pixabay.com/illustrations/creation-genesis-transhumanism-7427286/

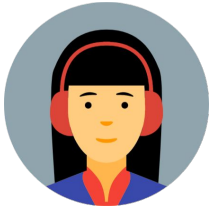
https://pixabay.com/photos/mexico-puebla-carriage-vehicle-1388435/



Careful Design

https://pixabay.com/photos/architecture-blueprint-floor-plan-1857175/

Test latest builds.



Microsoft | Windows Insider

How to preview Windows

Previewing Windows, also known as flighting is the process of running Windows Insider Preview Builds on your device. When you run these early versions of Windows and give us feedback, you can help us shape the future of Windows.

Preview Windows now

Microsoft MSRC | Researcher Portal

Report an issue to MSRC

What type of issue would you like to report to Microsoft?

Vulnerability report

Accept Reports

The Microsoft Security Response Center investigates all reports of security vulnerabilities affecting Microsoft products and services. If you are a security researcher and believe you have found a Microsoft security vulnerability that meets Microsoft's definition of a security vulnerability, we would like to work with you to investigate it.

Microsoft follows Coordinated Vulnerability Disclosure (CVD). We request that you follow these guidelines to help us protect customers and the ecosystem from harm.

Submit vulnerability report

Microsoft | MSRC Report an issue

Microsoft Security Servicing Criteria for Windows

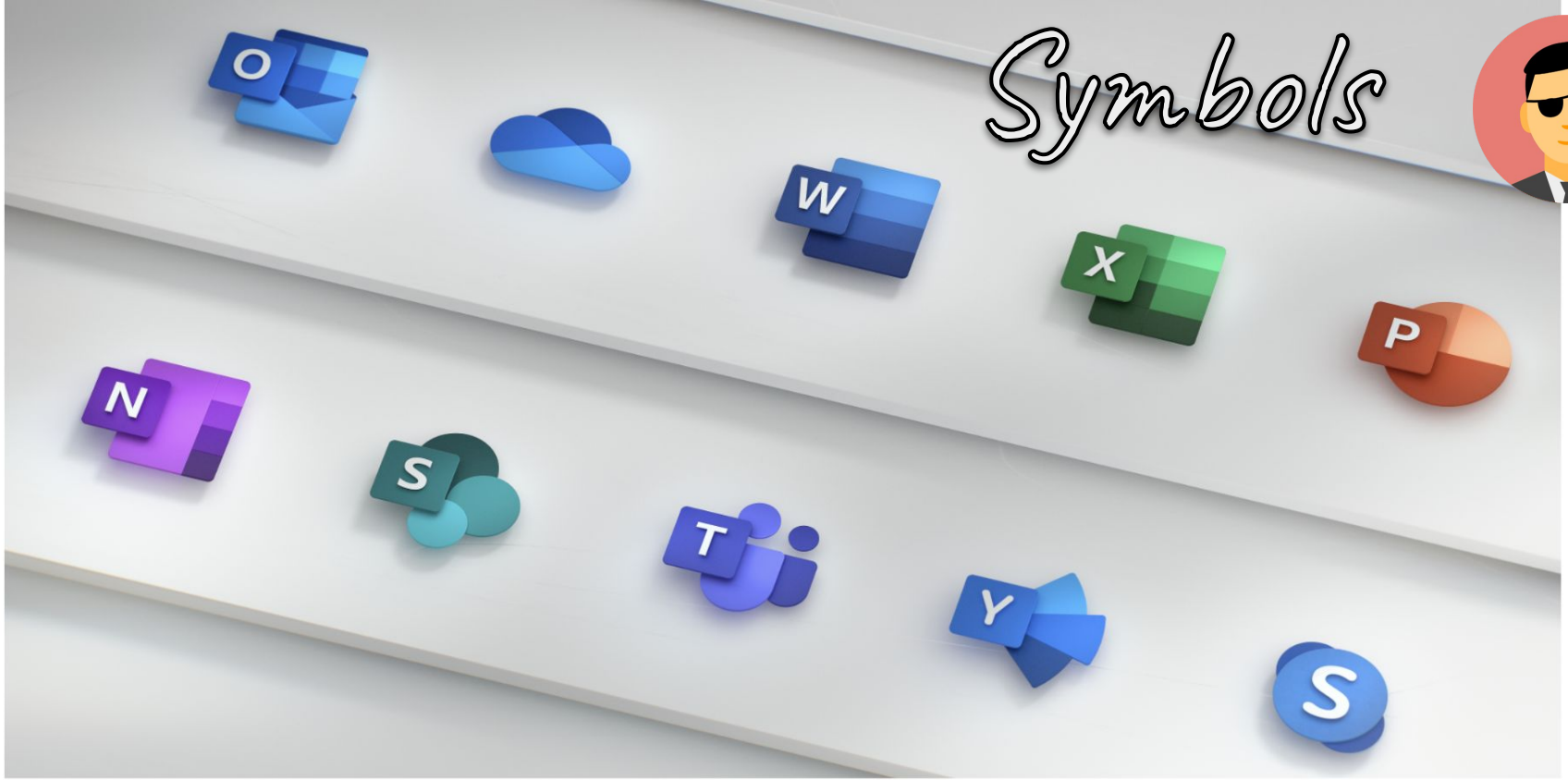
Define what's in Scope

https://pixabay.com/photos/money-cash-bills-currency-496229/

https://pixabay.com/photos/silhouette-head-bookshelf-knowledge-1632912/

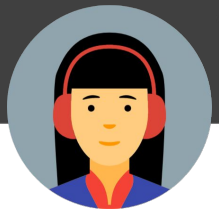
Documentation and Whitepapers

https://pixabay.com/photos/binding-contract-contract-secure-948429/



Microsoft Office to publish symbols starting August 2022

You have a Vulnerability Losers



Microsoft Security Response Center (microsoft.com)

You have a Vulnerability Losers

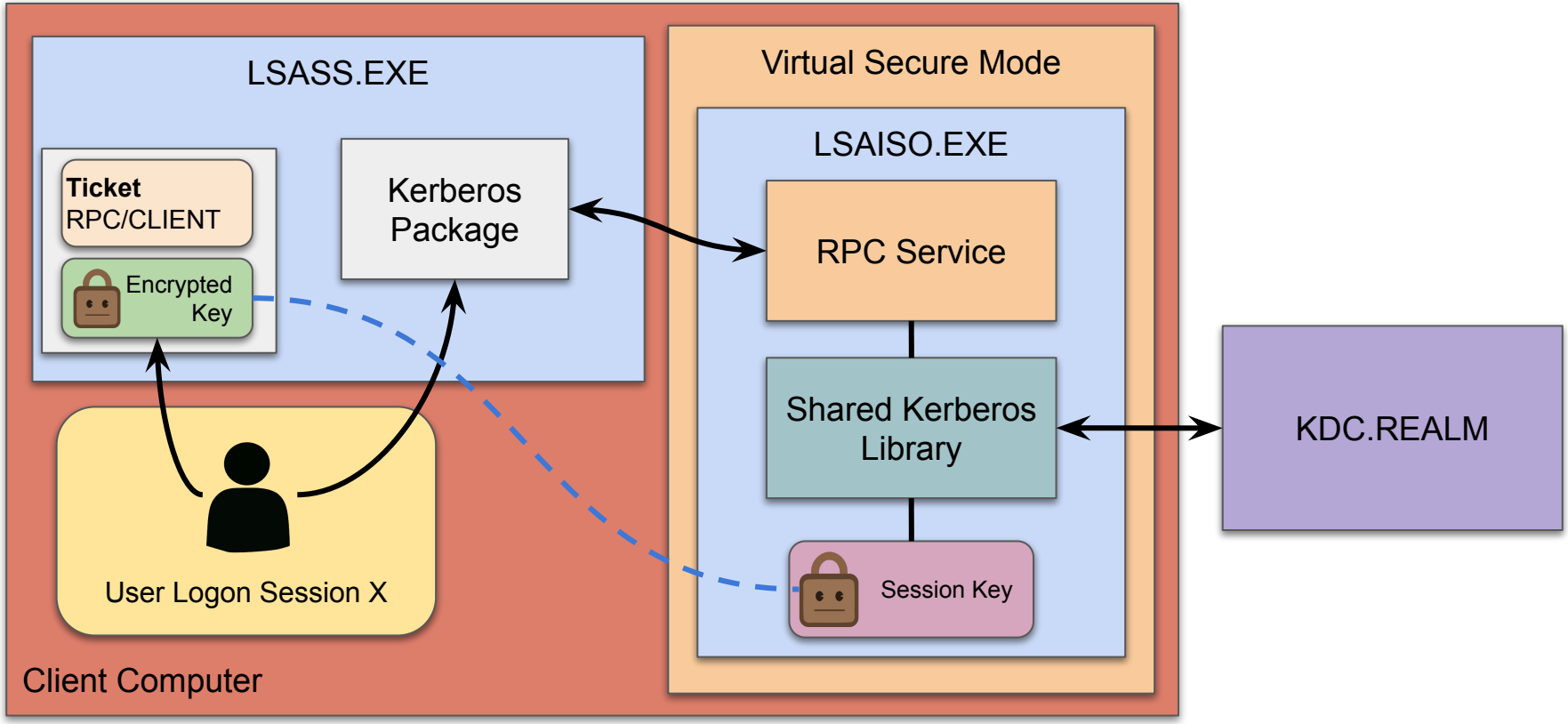
It's in the area blah blah blah! Here's a 200KB crash dump, you sort it out!

Lulz, you're so owned. Gimme $$$$.

Don't be a D**k

Sans Serif

Send



Windows Credential Guard!

New Windows 11 security features are designed for hybrid work

David Weston Vice President, Enterprise and OS Security

Marketing?

Enhanced identity protection and simplified password management

With Windows 11, you can protect your valuable data and enable secure hybrid work with the latest advanced security that small or medium-sized businesses say results in 2.8 times fewer instances of identity theft.[5] Here are a few enhancements that can help you stay secure now and in the future:

- **Windows Defender Credential Guard** is enabled by default with Windows 11 Enterprise. Credential Guard uses hardware-backed, virtualization security to help protect against credential theft techniques such as pass-the-hash or pass-the-ticket. In addition, this feature helps prevent malware from accessing system secrets even if the process is running with admin privileges.

Bug Bounty?

Program Name	Start Date	Eligible Entries	Bounty Range
Microsoft Hyper-V	2017-05-31	Critical remote code execution, information disclosure and denial of services vulnerabilities in Hyper-V	Up to $250,000 USD
Microsoft Windows Insider Preview	2017-07-26	Critical and important vulnerabilities in Windows Insider Preview	Up to $100,000 USD
Microsoft Applications and On-Premises Servers	2021-03-24	Critical and important vulnerabilities in Microsoft Applications and On-Premises Servers	Up to $30,000 USD
Windows Defender Application Guard	2017-07-26	Critical vulnerabilities in Windows Defender Application Guard	Up to $30,000 USD
Microsoft Edge (Chromium-based)	2019-08-20	Critical, important, and moderate vulnerabilities in Microsoft Edge (Chromium-based) Dev, Beta, and Stable channels	Up to $30,000 USD
Office Insider	2017-03-15	Vulnerabilities on Office Insider	Up to $15,000 USD
ElectionGuard	2019-10-18	Vulnerabilities in ElectionGuard	Up to $15,000 USD

[MS-RDPEAR]:

Documentation?

Remote Desktop Protocol Authentication Redirection Virtual Channel

---

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

IDA - Lsalso.i64 (Lsalso.exe) D:\re\Lsalso.i64

File Edit Jump Search View Debugger Options Windows Help

No debugger

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Functions window

Function name

- lumAlloc
- KerbFreeRpcKey
- KerbClientIumRpcInit(_LsaIsoSupportFunctions con
- KerbClientIumRpcCleanup(void)
- KerbIumGetClientContext
- KerbIumEncryptPassword
- KerbIumBuildPasswordList
- KerbIumBuildAsReqAuthenticator
- KerbIumVerifyServiceTicket
- KerbIumCreateApReqAuthenticator
- KerbIumDecryptApReply
- KerbIumUnpackKdcReplyBody
- KerbIumComputeTgsChecksum
- KerbIumBuildEncryptedAuthData
- KerbIumPackApReply
- KerbIumHashS4UPreauth
- KerbIumSignS4UPreauthData
- KerbIumVerifyChecksum
- KerbIumBuildTicketArmorKey
- KerbIumBuildExplicitArmorKey
- KerbIumVerifyFastArmoredTgsReply
- KerbIumVerifyEncryptedChallengePaData
- GetPkinitAuthenticator(KERB_KDC_REQUEST const
- ValidateAuthPackBuffer
- ValidatePkinitAuthenticator
- KerbIumBuildFastArmoredKdcRequest
- KerbIumDecryptFastArmoredKerbError
- KerbIumDecryptFastArmoredAsReply
- KerbIumUpdateSharedConfiguration
- EncryptSupplementalCredentials
- KerbIumDecryptPacCredentials
- FinishKeyAgreementCreation
- KerbIumCreateECDHKeyAgreement
- KerbIumCreateDHKeyAgreement

Line 184 of 663

Graph overview

IDA View-A | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums | Imports | Local Types | Exports

```
1  __int64 __fastcall KerbIumHashS4UPreauth(__int64 a1, const struct _KERB_ASN1_DATA *a2, __int64 a3, unsigned int a4, _DWORD *a5, _QWORD *a6)
2  {
3    unsigned int v6; // er12@1
4    __int64 v7; // rdi@1
5    const struct _KERB_ASN1_DATA *v8; // r15@1
6    __int64 v9; // rbx@1
7    signed int v10; // edx@3
8    __int32 v11; // ebx@6
9    _DWORD *v12; // rsi@10
10   _QWORD *v13; // r14@12
11   __int128 v15; // [rsp+30h] [rbp-40h]@5
12   __int64 v16; // [rsp+40h] [rbp-30h]@5
13   __int128 v17; // [rsp+48h] [rbp-28h]@1
14   __int128 v18; // [rsp+58h] [rbp-18h]@1
15   int v19; // [rsp+68h] [rbp-8h]@1
16   __int64 v20; // [rsp+A0h] [rbp+30h]@1
17
18   v20 = 0i64;
19   v19 = 0;
20   v6 = a4;
21   v7 = a3;
22   v8 = a2;
23   v9 = a1;
24   v17 = 0i64;
25   v18 = 0i64;
26   if ( a3 && *(_DWORD *)(a3 + 8) == 3 )
27   {
28     v17 = 0i64;
29     v18 = 0i64;
30     v10 = DecryptKey((const struct _KERB_ENCRYPTION_KEY *)a3, (struct _KERB_ENCRYPTION_KEY *)&v17);
31     v19 = v10;
32   }
33   else
34   {
35     v10 = 0xC000000D;
36     v19 = 0xC000000D;
37   }
38   v15 = 0i64;
```

0000C7C0 KerbIumHashS4UPreauth:23

Symbols?

Output window

```
1400397C8: using guessed type __int32 __stdcall KerbClientUnpackAsn1BufferVoid(const struct _KERB_ASN1_DATA *, unsigned __int32, void **);
140039F58: using guessed type __int64 __fastcall RtlAllocateHeap(_QWORD, _QWORD, _QWORD);
140045000: using guessed type void *WPP_GLOBAL_Control;
140045A00: using guessed type __int64 qword_140045A00;
```

IDC

AU: idle Down Disk: 324GB

project-zero project-zero

Success?

Bulk edit | + Add to hotlist | Change columns | 1 - 11 of 11

Status	Summary + Labels
Fixed	Windows: Windows Defender Remote Credential Guard Authentication Relay EoP CCProjectZeroMembers
Fixed	Windows: Kerberos Redirected Logon Buffer EoP CCProjectZeroMembers
Fixed	Windows: Credential Guard Domain-joined Device Public Key EoP CCProjectZeroMembers
Fixed	Windows: Credential Guard ASN1 Decoder Type Confusion EoP CCProjectZeroMembers
Fixed	Windows: Credential Guard BCrypt Context Use-After-Free EoP CCProjectZeroMembers
Fixed	Windows: Credential Guard Insufficient Checks on Kerberos Encryption Type Use CCProjectZeroMembers
Fixed	Windows: Credential Guard Kerberos Change Password EoP CCProjectZeroMembers
Fixed	Windows: Credential Guard KerbIumCreateApReqAuthenticator Key Information Disclosure CCProjectZeroMembers
Fixed	Windows: Credential Guard KerbIumGetNtlmSupplementalCredential Information Disclosure CCProjectZeroMembers
Fixed	Windows: Credential Guard Non-Constant Time Comparison Information Disclosure CCProjectZeroMembers
Fixed	Windows: Credential Guard TGT Renewal Information Disclosure CCProjectZeroMembers

CVE-2022-34712

Issue 2306: Windows: Credential Guard KerblumGetNtlmSupplementalCredential Information Disclosure

Reported by forshaw@google.com on Fri, May 27, 2022, 8:34 PM GMT+2

Windows: Credential Guard KerbIumGetNtlmSupplementalCredential Information Disclosure
Platform: Windows 10+
Class: Information Disclosure
Security Boundary: Virtual Secure Mode

Summary: The KerbIumGetNtlmSupplementalCredential CG API doesn't check the encryption key type leading to information disclosure of key material.

Assumptions: The vulnerability described assumes that a user has authenticated to a machine but is currently not active. As in they're not typing in their password which could be captured by a malicious SSP or changes to WDigest. This is an attack on the design and/or implementation of Credential Guard.

I also assume that you have SYSTEM and/or Kernel privileges on the machine and that you can get into the LSASS process. You could do this with a user-mode PPL bypass (assuming such a thing exists) or a memory corruption vuln or just having SYSTEM if PPL is not enabled. As far as I can tell this is a defended boundary as you shouldn't be able attack Credential Guard even from the VTL0 kernel.

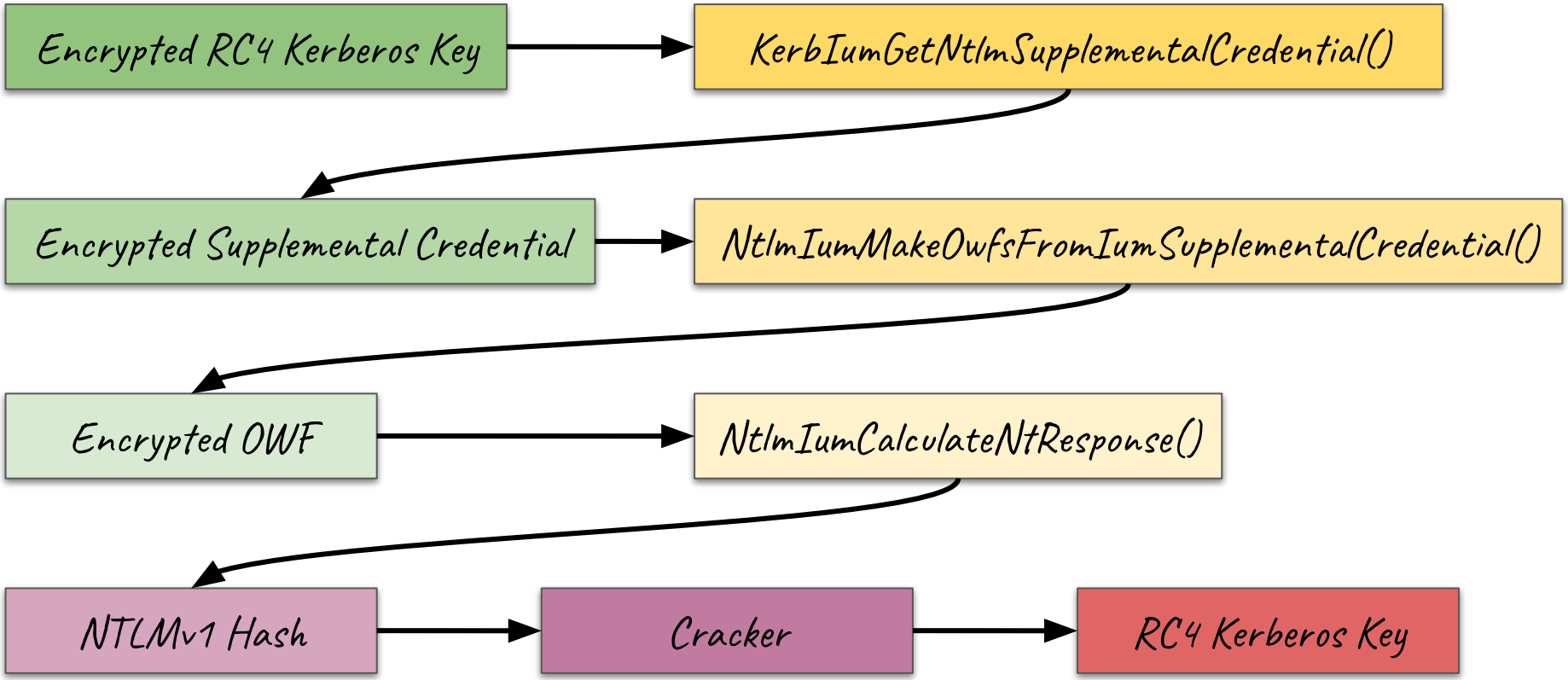
However, this research isn't going to verify that injecting into the process works. I can demonstrate that I can read out the encrypted key blobs for users from LSASS process memory so it'd only be a matter of injecting code into LSASS and calling the RPC API in LSAISO to invoke Credential Guard APIs.

Calculate NTLMv1 Response

```
NTSTATUS NtlmIumCalculateNtResponse(
      _In_  LPBYTE Challenge,
      _In_  PMSV1_0_SECRETS_WRAPPER Credential,
      _Out_ LPBYTE NtResponse) {
  IumpUnprotectCredential(Credential);
  return RtlCalculateNtResponse(Challenge,
                                Credential->Data,
                                NtResponse);
}
```

Exported as SystemFunction009

Exploit Chain



Protected User Group

Accounts that are members of the Protected Users group that authenticate to a Windows Server 2012 R2 domain are unable to:

- Authenticate with NTLM authentication.
- Use DES or RC4 encryption types in Kerberos pre-authentication.
- Be delegated with unconstrained or constrained delegation.
- Renew the Kerberos TGTs beyond the initial four-hour lifetime.

https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group

Trying Something Different

```
NTSTATUS KerbIumGetNtlmSupplementalCredential(
  _In_  const PKERB_ENCRYPTION_KEY Key,
  _Out_ PMSV1_0_IUM_SUPPLEMENTAL_CREDENTIAL *Cred) {

  KERB_ENCRYPTION_KEY DecryptKey;
  DecryptKey(Key, &DecryptKey);
  if (DecryptKey.KeySize == 16) {
    LsaIsoEncryptNtlmSupplementalCred(DecryptKey.Key, Cred);
    return STATUS_SUCCESS;
  } else {
    return STATUS_INVALID_PARAMETER;
  }
}
```

Only checks size, not key type.

Can use an AES128 key

Your NETNTLM DES Cracking Job Results

C

crack.sh

To: You

Sat 11/06/2022 06:08

Crack.sh has successfully completed its attack against your NETNTLM handshake. The NT hash for the handshake is included below, and can be plugged back into the 'chapcrack' tool to decrypt a packet capture, or to authenticate to the server:

Token: $NETNTLM$1122334455667788$C1D0A554FE8B6749480B5BB14F698BDEFB65C89E775721D2
Key: 2419d566fb174678a126125985437b97

Recovered the AES128 key

This run took 45 seconds. Thank you for using crack.sh, this concludes your job.

Practical cracking time

Reply Forward

Microsoft | MSRC

Microsoft Windows Insider Preview Bounty Program

General Awards

Security Impact	Maximum Award
Remote Code Execution	$5,000
Elevation of Privilege	$2,000
Security Feature Bypass	$1,000
Information Disclosure	$1,000
Spoofing	$1,000
Tampering	$1,000
Denial of Service	$500

~$20,000?

https://pixabay.com/photos/love-dog-cat-together-animal-fur-2706241/