

Week 05

Windows Environments

Thomas & Chris

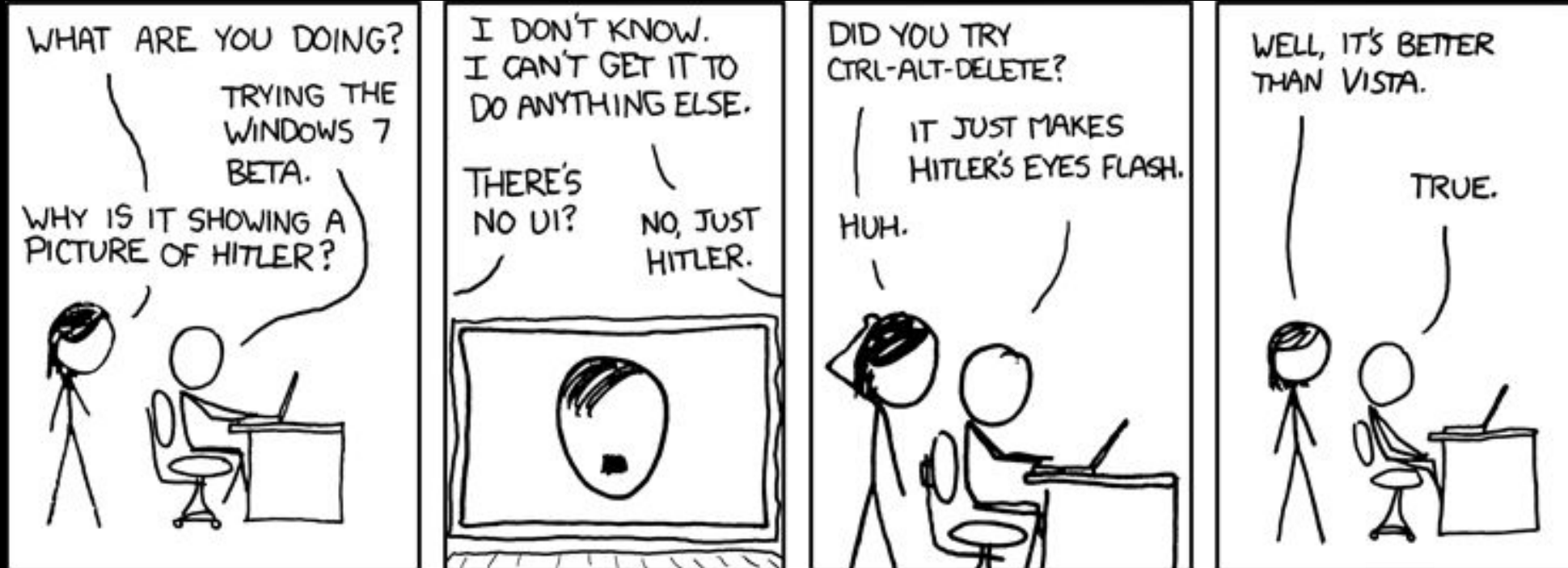


Announcements

- TracerFIRE, next weekend!
- Spray Paint Social: First week of April
- Cyphercon! (Late April, Tickets on sale now I think...)
 - We will get sponsorship money



sigpwny{ctrl_alt_delete}



Core OS Basics



System Architecture

- NTOSKRNL - Core kernel
- HAL - Hardware abstraction layer

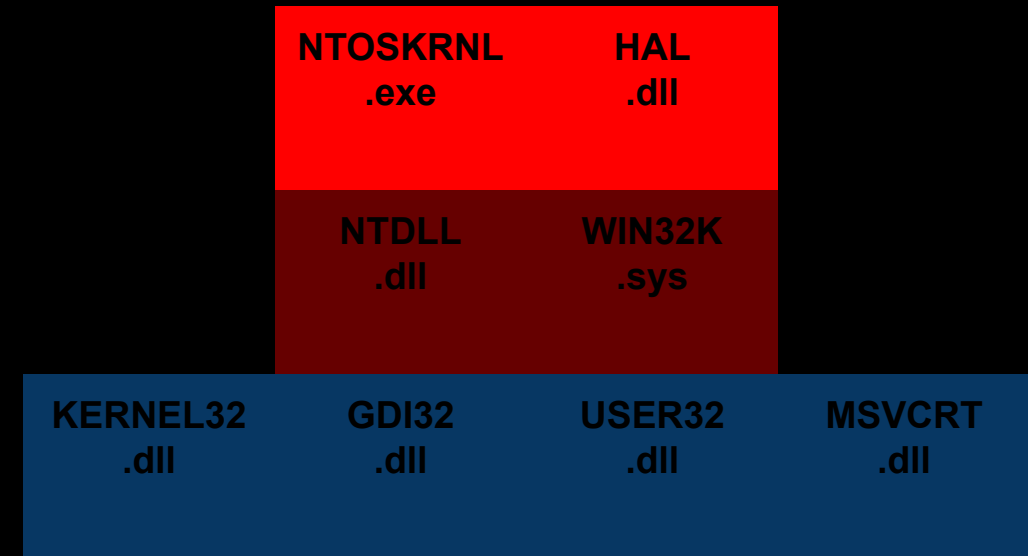
Layer 1

- NTDLL - Syscalls and runtime library
- WIN32K - Graphics & UI

Layer 2

- kernel32 - System API
- gdi32 - Drawing API
- user32 - UI API
- MSVCRT - C API (stdlib)

Layer 3



System Architecture

- NTOSKRNL - Core kernel
- HAL - Hardware abstraction layer

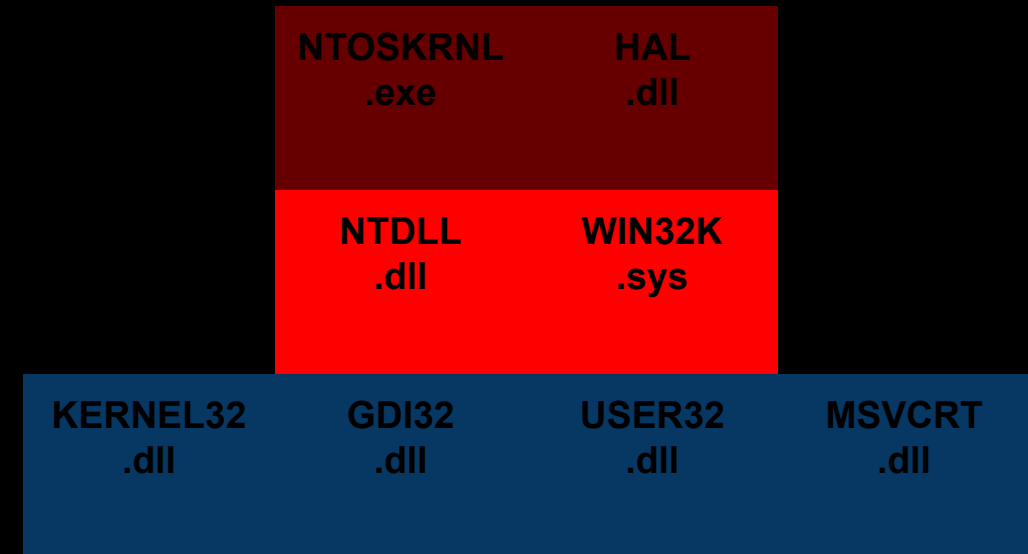
Layer 1

- NTDLL - Syscalls and runtime library
- WIN32K - Graphics & UI

Layer 2

- kernel32 - System API
- gdi32 - Drawing API
- user32 - UI API
- MSVCRT - C API (stdlib)

Layer 3



System Architecture

- NTOSKRNL - Core kernel
- HAL - Hardware abstraction layer

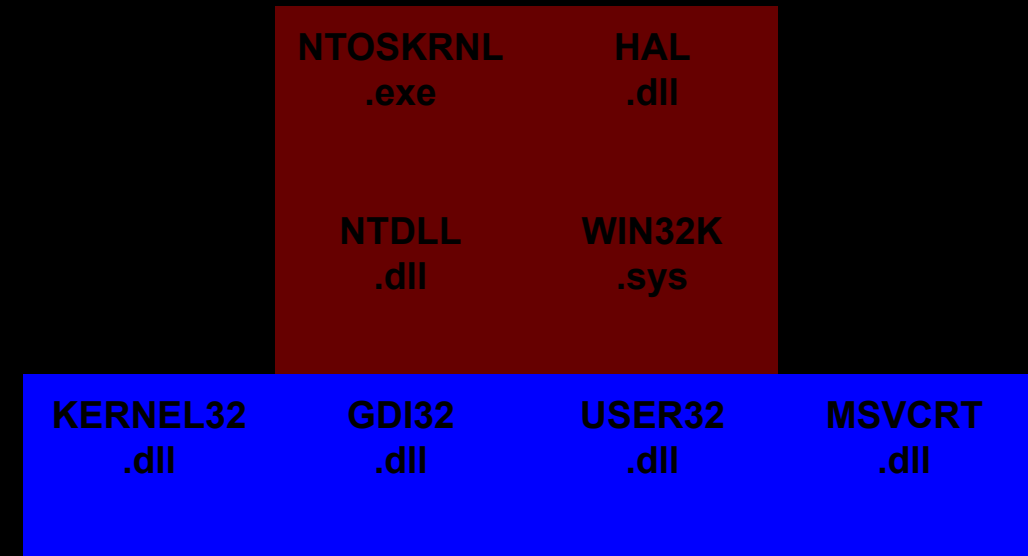
Layer 1

- NTDLL - Syscalls and runtime library
- WIN32K - Graphics & UI

Layer 2

- kernel32 - System API
- gdi32 - Drawing API
- user32 - UI API
- MSVCRT - C API (stdlib)

Layer 3



WINAPI

Core way that any program interacts with Windows

Includes interfaces for files, devices, windows, threads, etc.

Monitoring a program's WINAPI calls is extremely **powerful**!

[Here's A Cool Tool To Do It](#)



API Monitor v2 (Alpha-r6) 32-bit

File Edit View Tools Help

API Capture Filter: All Modules

Hooked Processes: C:\Program Files (x86)\Internet Explorer\... c:\Network Utilities\SecureCRT\SecureCR... C:\Programming\Microsoft Visual Studio... c:\windows\SysWOW64\cmd.exe (Termin... e:\Utilities\procexp.exe (Terminated) Thread 13772 (cmd.exe+0x829A) Thread 8816 (procexp.exe+0x4EB02)

Summary: 27096 Calls

#	TID	Module	API	Return	Error
26831	8816	KERNELBASE.dll	RtlInitializeCriticalSectionAndSpinCount (0x007012a0, 4000)	STATUS_SUCCESS	
26832	8816	KERNELBASE.dll	RtlInitializeCriticalSectionAndSpinCount (0x00700abc, 4000)	STATUS_SUCCESS	
26833	8816	procexp.exe	CreateFileA ("e:\Utilities\procexp64.exe", GENERIC_WRITE, FILE_SHARE_READ FILE_SHARE_WRITE, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL)	INVALID_HANDLE_VALUE	5 = Access is denied.
26834	8816	kernel32.dll	RtlInitAnsiStringEx (0x0030f23c, "e:\Utilities\procexp64.exe")	STATUS_SUCCESS	
26835	8816	KERNELBASE.dll	RtlAnsiStringToUnicodeString (0x0030f254, 0x0030f23c, 0)	STATUS_SUCCESS	
26836	8816	kernel32.dll	RtlInitUnicodeStringEx (0x0030f224, "e:\Utilities\procexp64.exe")	STATUS_SUCCESS	
26837	8816	KERNELBASE.dll	RtlInitUnicodeStringEx (0x0030f1d8, "e:\Utilities\procexp64.exe")	STATUS_SUCCESS	
26838	8816	KERNELBASE.dll	NtCreateFile (0x0030f1f8, GENERIC_WRITE SYNCHRONIZE, 0, 0x0030f360, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL)	STATUS_ACCESS_DENIED	0xc0000022 = {Access Denied}...
26839	8816	KERNELBASE.dll	RtlFreeHeap (0x00710000, 0, 0x00775228)	STATUS_SUCCESS	
26840	8816	KERNELBASE.dll	RtlFreeHeap (0x00710000, 0, NULL)	STATUS_SUCCESS	
26841	8816	KERNELBASE.dll	RtlNtStatusToDosError (STATUS_ACCESS_DENIED)	STATUS_SUCCESS	
26842	8816	kernel32.dll	RtlFreeUnicodeString (0x0030f254)	STATUS_SUCCESS	
26843	8816	procexp.exe	GetFileAttributesA ("e:\Utilities\procexp64.exe")	FILE_ATTRIBUTE_ARCHIVE FILE_ATTRIBUTE_HIDDEN	
26844	8816	KERNELBASE.dll	RtlInitAnsiStringEx (0x0030f3d0, "e:\Utilities\procexp64.exe")	STATUS_SUCCESS	
26845	8816	KERNELBASE.dll	RtlAnsiStringToUnicodeString (0x0030f3e8, 0x0030f3d0, TRUE)	STATUS_SUCCESS	
26846	8816	KERNELBASE.dll	NtQueryAttributesFile (0x0030f3b8, 0x0030f390)	STATUS_SUCCESS	
26847	8816	KERNELBASE.dll	RtlFreeHeap (0x00710000, 0, 0x00775228)	TRUE	

Parameters: NtCreateFile (Ntdll.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	PHANDLE	FileHandle	0x0030f1f8 = NULL	0x0030f1f8 = NULL
2	ACCESS_MASK	DesiredAccess	GENERIC_WRITE SYNCHRONIZE 128	GENERIC_WRITE SYNCHRONIZE 128
3	POBJECT_ATTRIBUTES	ObjectAttributes	0x0030f19c	0x0030f19c
	OBJECT_ATTRIBUTES		{ Length = 24, RootDirectory = NULL, ObjectName ...	{ Length = 24, RootDirectory = NULL, ObjectNam...
	ULONG	Length	24	24
	HANDLE	RootDirectory	NULL	NULL
	PUNICODE_STRING	ObjectName	0x0030f1d8	0x0030f1d8
	UNICODE_STRING		{ Length = 60, MaximumLength = 538, Buffer = 0x...	{ Length = 60, MaximumLength = 538, Buffer = 0...
	USHORT	Length	60	60

Call Stack: NtCreateFile (Ntdll.dll)

#	Module	Address	Offset	Location
1	KERNELBASE.dll	0x74f3b634	0x1b634	CreateFileW + 0x35e
2	kernel32.dll	0x763b3fa6	0x13fa6	CreateFileW + 0x4a
3	kernel32.dll	0x763b53fc	0x153fc	CreateFileA + 0x36

Output

procexp.exe: Hooked Module 0x75DC0000 -> C:\Windows\syswow64\USP10.dll.
procexp.exe: Hooked Module 0x760C0000 -> C:\Windows\syswow64\LPK.dll.
procexp.exe: Hooked Module 0x766D0000 -> C:\Windows\syswow64\SHLWAPI.dll.
procexp.exe: Hooked Module 0x76860000 -> C:\Windows\syswow64\COMDLG32.dll.
procexp.exe: Hooked Module 0x75170000 -> C:\Windows\syswow64\SHELL32.dll.
procexp.exe: Hooked Module 0x73010000 -> C:\Windows\system32\VERSION.dll.
procexp.exe: Hooked Module 0x75EC0000 -> C:\Windows\syswow64\MSCTF.dll.

API Loader Hooks Output

Ready 24.22 MB Mode: Standard



Handles & Data Types

- Windows has weird names for types
- Uses Hungarian notation
 - Variable prefixed with abbreviation of its data type
 - ex. LPVOID = long pointer void = void*
- Kernel maintains a list of objects it is responsible for
 - Each is given unique address in this global list called a **HANDLE**
 - **HINSTANCE** - Handle to a process
 - **HMODULE** - Handle to a module (DLL)



Dynamic Link Libraries (DLLs)

Code fragments to be compiled into a standalone library

Linked statically & dynamically against multiple programs

Dynamically loading DLLs common practice in malware & game hacking 🙄

```
BOOL APIENTRY DllMain( HMODULE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved) {
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
```

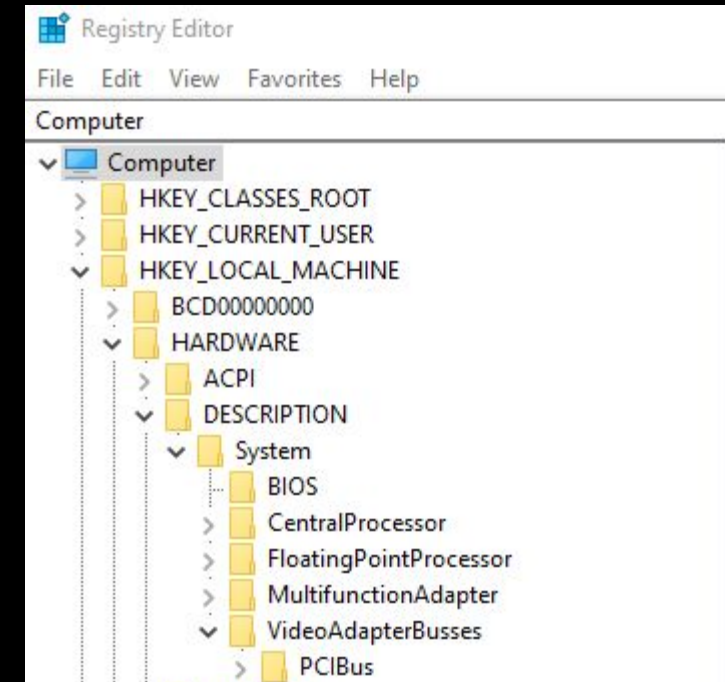


Registry

Hierarchical central database

Stores keys in tree format

Keys can be saved and queried with
WINAPI



Enterprise Windows

(We will run longer mtg on this, here are basics)

There are no chals for this, start on chals if you want



Domains

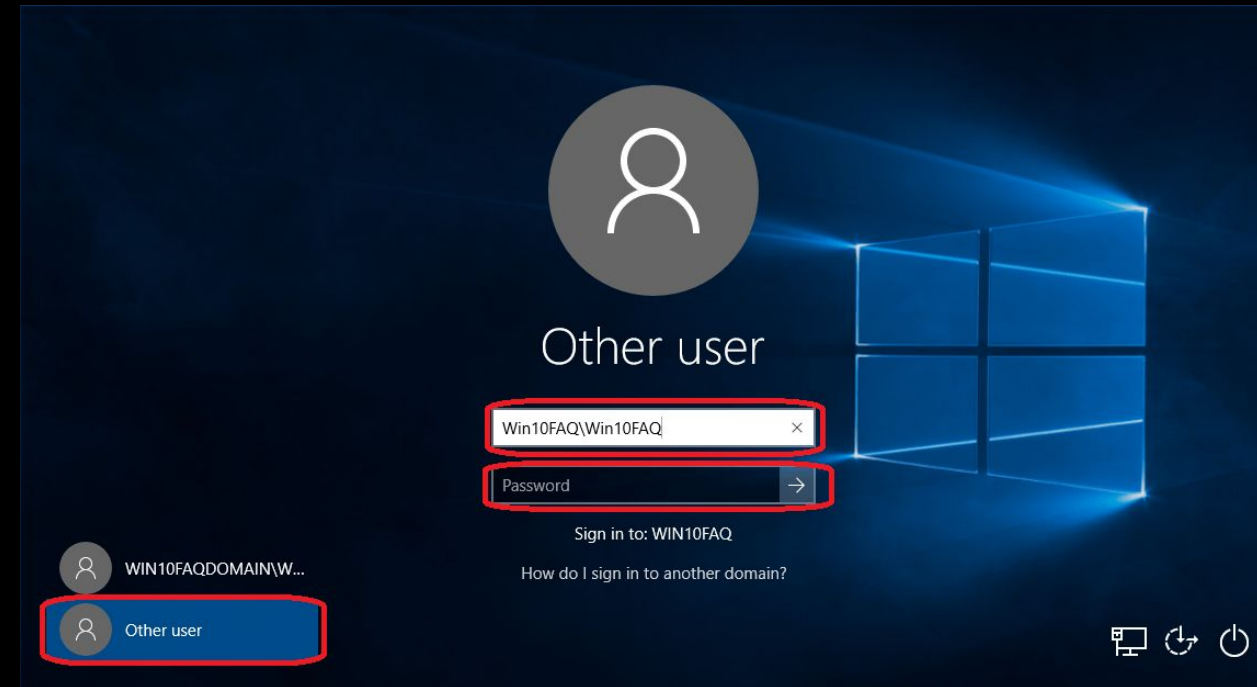
Domains are a network of **user accounts**, computers, devices (printers), and other **security protocols**.

- Users are often of the form `user@domain.tld` (or `domain\user`)
(`tquig2@illinois.edu`, `illinois.edu\tquig2`)

Users have different levels of access split into **Groups**

- Domain Admins

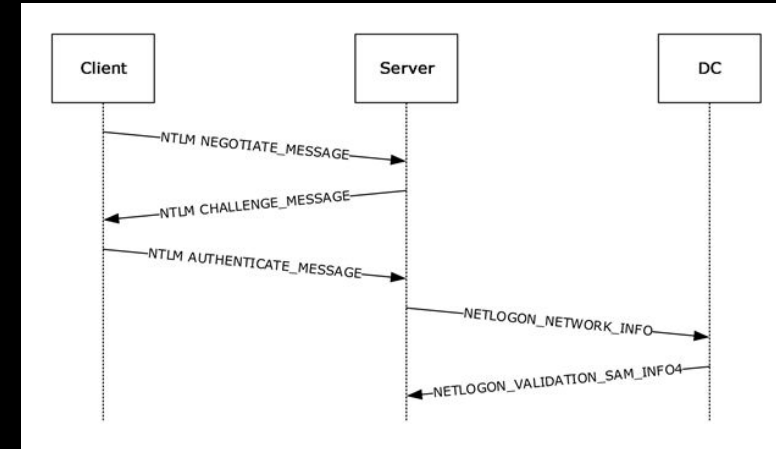
Controlled by a **domain controller**



Authentication Methods

NTLM

- Windows New Technology LAN Manager
- Challenge / Response method
- Allows for security Selection
- NTLMv1 is vulnerable, NTLMv2 is a bit better



Kerberos

- Strong ticket-based authentication method
- Vulnerable to “Kerberoasting”



Next Meetings

Sunday Seminar: Crypto Math w Nebu

- Explain the underlying math of crypto (RSA, AES, earlier crypto)
- Stop Crypto Paralysis!

Next Thursday: (??? Pentesting Fundamentals ???)

- We aren't sure yet what it will be
- Probably pentesting fundamentals
- might be something else :)

