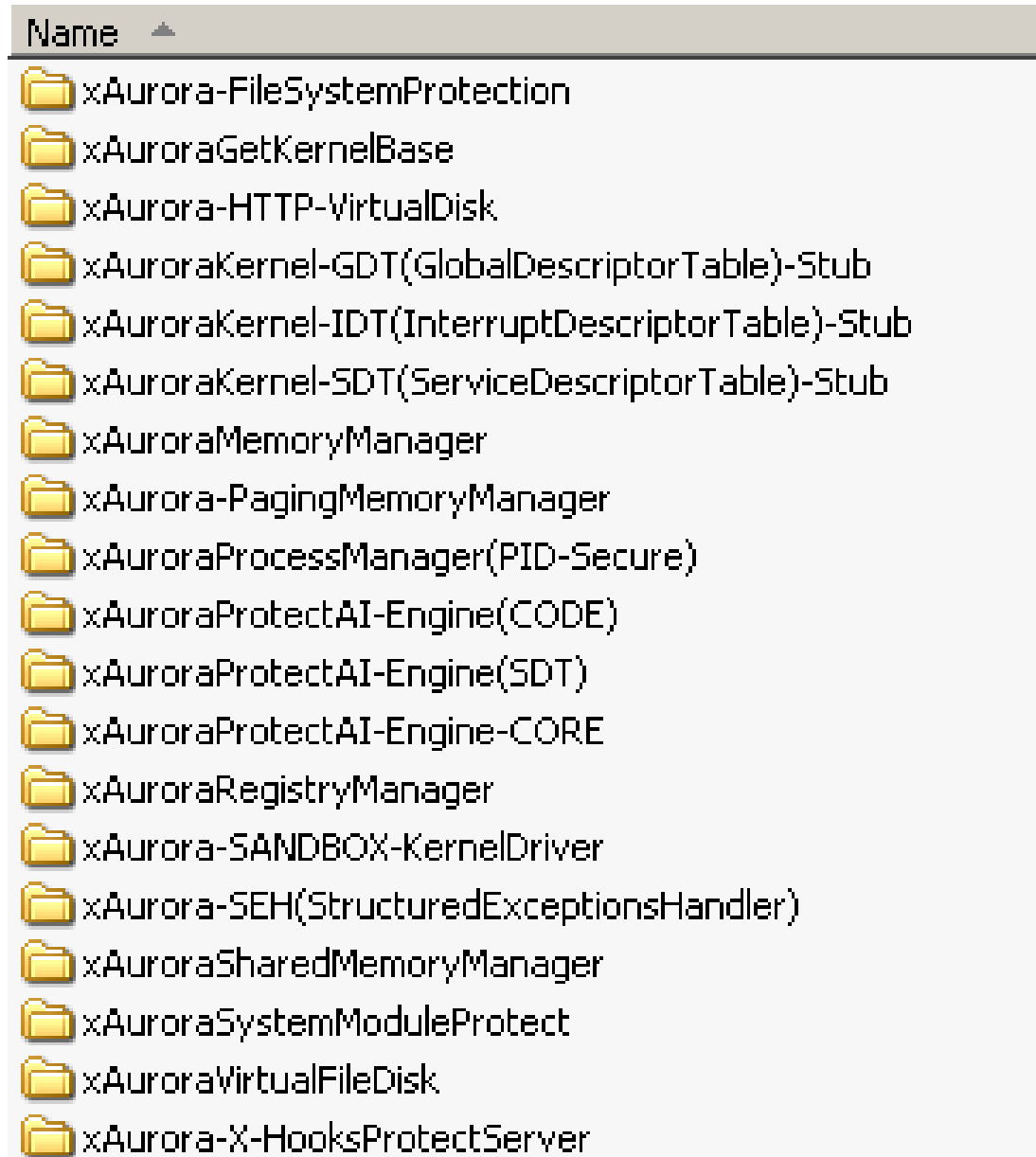


Reply to Mr. Gotaimbara

Demystifying the xAurora KMD(s) (Kernel Mode Drivers) Workflow

Response To: Mr. Anonymous Skywalker, Mr. Harshadeva Ariyasinghe and Mr. Kalinga Athulathmudali

Kernel Mode and User Mode Operations Visual Approach – Kernel Mode Drivers of xAurora



Entire Kernel Mode Driver List of xAurora

Detailed Overview of the Drivers

(1). xAurora-FileSystemProtection

- Providing Tight File System Security for NTFS/FAT
- Loads Anti-Disk/File Hooks Protection for Windows Kernel
- Stops ROOTKIT / SHELLCODE Hooks to Kernel
- Loads xAurora to Separate Virtual Spaces in Windows (xAurora Never loads to Standard Disk Space Allocated by Windows for better protection)

There are four methods,

- Virtual Memory
- Slack Space
- ADS - Alternate Data Stream (NTFS Only)
- Unallocated Area - Using (Dynamic Live Virtual FAT - DLV-FAT)
Any hard drive contains 8Mb Non Allocated Space which can not be used by any other program, xAurora creates 6 Mb DLV-FAT on the particular place for extreme Access Protection

(2). xAuroraGetKernelBase

- Initialize xAurora Web Kernel with Windows Kernel
- Initialize xAurora Web Kernel Hooks to USER MODE xAurora Stub
- Hook NTDLL.DLL Marshall Libraries Hooks to xAurora Stealth Mode Drivers
- Get all the IMPORT TABLE and EXPORT TABLE entries and initialize all
- Re-Initialize and Flush all the initialized hooks with Kernel and Marshall
- Close all unwanted hook and unbound the broken hooks with Kernel
- Make the link between xAurora Kernel mode drivers with Windows Kernel
- Unhooking and flushing all non standard Windows library hooks
- Hooking with HTTPAPI.DLL, WINHTTP.DLL, HTTPEXT.DLL, HTTPMB51.DLL, HTTPOD51.DLL and HTTPMLIB.DLL
- Bridging with Microsoft HTTP Stack Kernel Device Driver - HTTP.SYS

(3). xAurora-HTTP-VirtualDisk

- Web contents processing Virtual Container Driver
- Optimize Rendered Web contents and manipulate them initialize
- xAurora HTTP Web Cache and TCP/IP Stack processing cache initialize
- Virtual Storage for Artificial Intelligence analysis statistical data
- Creating Anti-Malware environment in xAurora Web Space
- Cache container for the xAurora Kernel Mode Device Drivers
- Repair and Reinitialize broken xAurora Device Drivers
- Maintain and Flush xAurora Device Driver Cache
- Bind and Unbind Kernel Mode/User Mode Hooking Cache Libraries
- Quarantine Detected Malicious contents and Flush them when exiting

(4). xAuroraKernel-GDT(GlobalDescriptorTable)-Stub

- Handle Environment Global Functionalities of the browser
- Handle Global Kernel and User hooks
- Maintain Kernel Descriptor Tables
- Manage User Mode Global Descriptors
- Marshalling User Mode to Kernel Mode
- Global Hooking and Unhooking
- Kernel Descriptor Table initialize
- Initializing Stealth Mode GDT(s) in Kernel Mode
- Bridging with IDT(s) and SDT(s)
- Analyze Kernel Mode Stealth GDT(s) with Win32 Native API(s)

(5). xAuroraKernel-IDT(InterruptDescriptorTable)-Stub

- Handle Environment Interrupts
- Initialize INT3 Anti-Debug Core
- Anti-Debug Interrupt Table Realignment
- Set Software and Hardware Breakpoints
- Initialize Breakpoint Buffers
- Bypass Windows Software Interrupts
- Kernel Mode Stealth IRQ Handling
- IDT(s) Unhooking with Windows Kernel
- Bridging with GDT(s) and SDT(s)
- Interrupt Vector Table Debugging with Smart Initialize Hooking Libraries

(6). xAuroraKernel-SDT(ServiceDescriptorTable)-Stub

- Handle Service Oriented Descriptors
- Windows Service Level Hooking
- Ring-0 and Ring-1 Mode Bridging
- Enforce to release the Software Breakpoints
- Reallocate Windows Service Level Buffering for xAurora
- Marshal NTDLL.DLL protection enforcing
- xAurora Local Descriptor Table handling
- Remove malfunctioned Core Service Level Threads and Hooks
- Bridging with IDT(s) and GDT(s)
- Anti-Service Level Debug process handling

(7). xAuroraMemoryManager

- Memory manipulation in xAurora environment
- Anti-Memory Debug Process hooking
- Allocating Kernel Mode processes to manage the memory
- Unhooking and Flushing Virtual Memory Spaces
- Link Virtual Disk Operations and Real Disk Operations - Vice Versa
- Manage environmental Memory for the browser
- Layering Memory for the xAurora components
- Kernel and Marshall Memory Core Initiator Mode handling for the browser
- Keep and stay live all the Memory based hooking methods
- Handling Core Memory based processes, stacks, layers and components

(8). xAurora-PagingMemoryManager

- Handling entire memory paging
- Handling Ghost operations in the browser environment
- Manage entire Global Vectors and Paging Buffers
- Handling Peak Memory Usage, Memory Delta and Page Faults
- Initialize Page Fault Delta and VM Paging
- Managing Paged Pool Data and Non Paged Pool Data
- Handle entire I/O Operations inside the browser
- I/O Debug Hooks handling
- Manipulate I/O Mode Page Operations
- Reinitialize and Flush all Paged and Non Paged operations

(9). xAuroraProcessManager(PID-Secure)

- Initialize Core Processes in browser
- Make the boundaries between Windows Processes and Browser Processes
- Manage Core PID related operations
- Analyze all Native Windows Processes for Anti-Hijacking
- Enforce Native Windows Processes to protect GDT(s)
- Bridging SDT(s) hooking with Native API Calls
- Flushing SDT Cache to remove unwanted buffer spaces
- Promoting and Demoting Ring0 Operations
- Inter-Process communication with Native Windows API(s)
- Obfuscation SDT(s) to protect from Buffer Underrun and Overrun Attacks

(10). xAuroraProtectAI-Engine(CODE)

- Managing AI Statistics
- Initialize the Fuzzy Stacks and Logic Controls
- Initialize the AI Protection Stubs
- Unbind and Rebind AI Security Descriptors
- Statistical Analysis of Malware and Malicious Contents
- Code Splicing and Code Smoothing for Web Rendering
- AI Engine Initialize for Buffer Underrun/Overrun Attacks
- Load Anti-Kernel Panic Code Buffer
- Initialize Anti-Crash Stub for Browser Operations
- Render Web contents through AI Engine

(11). xAuroraProtectAI-Engine(SDT)

- Managing AI Statistics for SDT Threads
- Broadcast AI functionalities in Web Interface
- Send Global Hooks directly to Address Table in browser
- API Redirection handling in Secured Hooks
- Kernel Mode and User Mode Inter-Processes Management in browser
- Advanced AI Threat Protection via SDT(s)
- Stop Global SDT Hooks when browser is in Ring-1
- Remove Unwanted SDT(s) in Kernel Mode
- Reinitialize and Realign entire SDT(s) in Device Drivers
- Enforce SDT operations for Anti-Freeze operations in browser environment

(12). xAuroraProtectAI-Engine-CORE

- Handle entire AI Threat filtering operations inside browser
- Run Real Mode Hooks to Kernel and link with them
- Drop Mature Kernel Hooks and replace them with New Kernel Hooks
- Initialize Drop Mode and React Mode operations in browser
- Initialize AI Based Hack-Back operations
- Handle Threat Filtering objects and components
- Kernel Mode DDOS and BOTNET protection handling
- Remove Anti-Rootkit and Anti-Shellcode Threads/Hooks
- Use Anti-Reverse Engineering Engine for better protection
- Remove all Native Windows API Calls to Unbind Rootkits

(13). xAuroraRegistryManager

- Handle all xAurora Registry operations
- Stop QUERY, READ API Calls to protect the system from Malware
- Stop all Untrusted Registry Calls
- Quarantine all Untrusted Inter-Process Registry operations
- Remove all WRITE API Calls for Untrusted Foreign Processes
- Manage and Restrict Standard API Registry Calls
- Stop Core Privilege objects in Windows environment
- Make Anti-Tamper Signature of Core xAurora Objects in registry
- Take Live Snapshots of System Registry
- Compare and Restore Tampered Registry settings by Malware

(14). xAurora-SANDBOX-KernelDriver

- Complete protection from Spyware and Malcodes for browser
- Creating Virtual Sanbox Environment for each tab
- Split Parent Sandbox to Multiple Child Sandboxes in the browser
- Create Virtual Loaders inside each Child Sandbox
- Flush and Manipulate Sandboxes regularly
- Remove and Unload unwanted used Sandbox Virtual Pools
- Select closet Sandbox according to the usage
- Protect and select Sandbox environment for better performance
- Create Virtual Sandbox Arrays
- Integrate with DEP on x86 and x64 environment for better security

(15). xAurora-SEH(StructuredExceptionsHandler)

- Handle entire Win32 Exceptions inside the browser environment
- Manage SEH in x86 Threads
- Run SEH Kernel Mode Threads in Web Browser
- Separate Native SEH with xAurora SEH sections
- Wipe all unwanted Exceptions
- Close all Untrusted SEH
- Replace Native API Calls Handling with xAurora API SEH Handlers
- Manage Minor and Temporary Exceptions
- Replicate and Reuse all the used Exceptions
- Manage Kernel and User Mode Exceptions
- Vector Exception Handles (VEH)

(16). xAuroraSharedMemoryManager

- Handle Shared Memory pool in Windows
- Acquire Page Information in Real Mode for the browser environment
- Create Virtual Memory for Sandboxes
- Link and Share the Core Kernel Objects for the Sandbox
- Secure wipe all shared memory Pages
- Handle I/O Core Manager for better performance
- Close all Non Paged handles
- Separate entire I/O Operations with Shared Memory Operations
- Enforce I/O Operations to be allocate the Shared Memory
- Manage Share Memory I/O for CPU Threading

(17). xAuroraSystemModuleProtect

Manage and Secure System modules

- Handle entire Module Based Operations in browser environment
- Protect System Modules from Unknown hooking
- Create protected Anti-Overflow environment
- Layering Systems Calls for Protected API(s)
- Handle all I/O Paging System Calls and Modules
- Unhook and Unbind sensitive Modules from the browser
- Link Shared objects with Anti-Hook Call Manager
- Reallocate Threads for System Module Protections
- Bridging System Modules with xAurora Modules to achieve maximum security

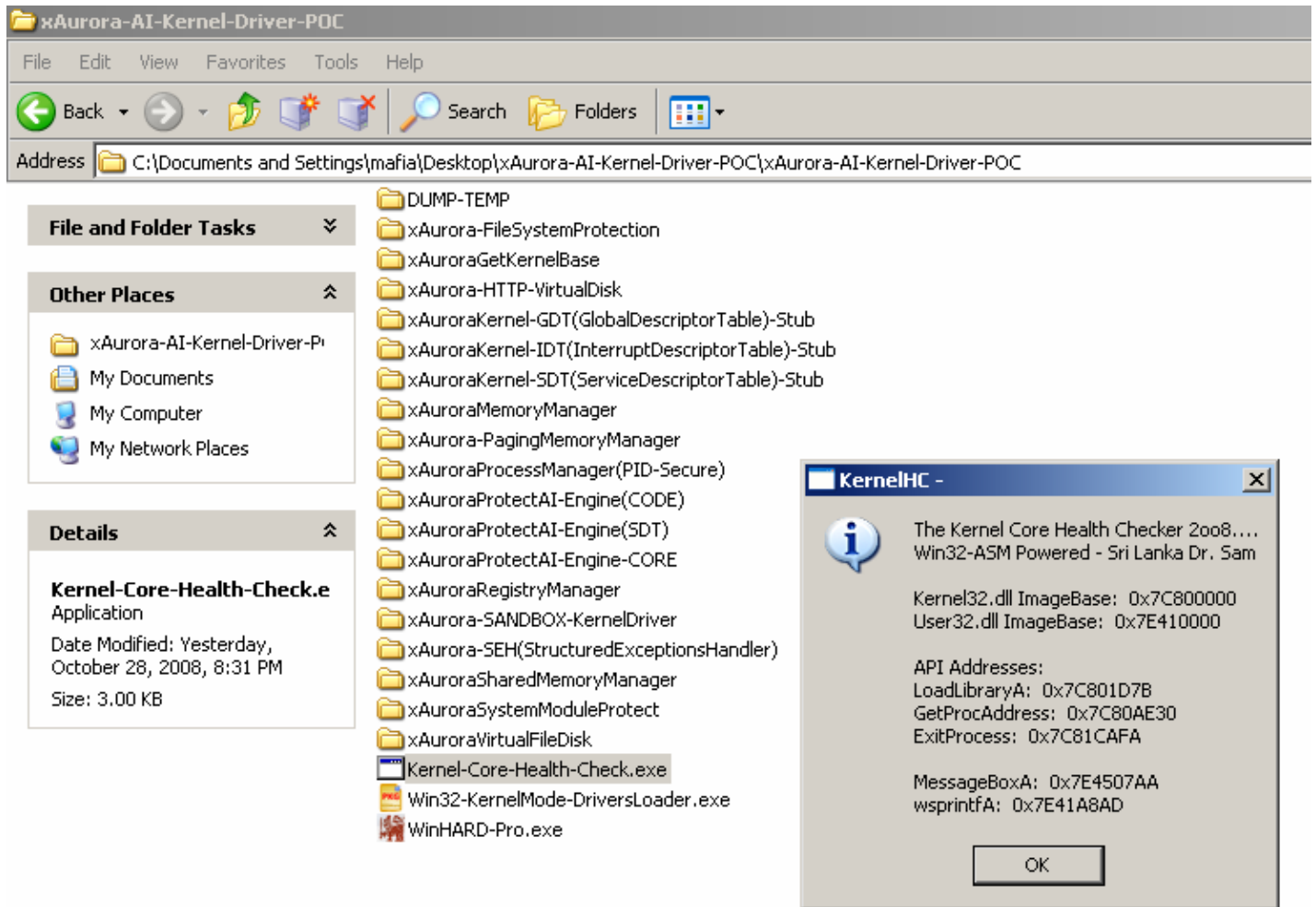
(18). xAuroraVirtualFileDisk

- Map Real File to Virtual Stealth Drive
- Splice unwanted Virtual File Hooks
- Control Real Time Threats
- Release File System Cache in browser environment
- Inter-Process operations with Microsoft Shadow Copy Service Provider
- Share Virtual resources with Microsoft Hypervisor
- Creating Hyper Virtual environment for xAurora
- Link xAurora directly to the Virtual Disk
- Load all stealth mode device drivers in Virtual Disk environment
- Run xAurora in a Separate Virtual Disk Stack with support of Shadow Copy

(19). xAurora-X-HooksProtectServer

- Create real secure cluster inside the browser environment
- Server section handle entire Kernel Hooking Operations
- Server can Hide/Stealth Windows Processes
- Handle Protection of all running processes in Windows Environment
- Manage HWND and Window Class operation
- Hide Use Mode Windows from Windows Task Manager
- Provides Reboot Protection and Protect From Window Hooks
- Provides Disk Format Protection (Anti-Format for USB-Flash, Hard Disks)
- Provides Parent Process Enumeration for GDT(s)
- Anti-Anti Debug Protection for Runtime Application including Browser

Checking the Health of the Kernel to Load the KMD(s)



The screenshot shows a Windows Explorer window titled "xAurora-AI-Kernel-Driver-POC" with the address bar showing the path "C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Driver-POC\xAurora-AI-Kernel-Driver-POC". The file list includes:

- DUMP-TEMP
- xAurora-FileSystemProtection
- xAuroraGetKernelBase
- xAurora-HTTP-VirtualDisk
- xAuroraKernel-GDT(GlobalDescriptorTable)-Stub
- xAuroraKernel-IDT(InterruptDescriptorTable)-Stub
- xAuroraKernel-SDT(ServiceDescriptorTable)-Stub
- xAuroraMemoryManager
- xAurora-PagingMemoryManager
- xAuroraProcessManager(PID-Secure)
- xAuroraProtectAI-Engine(CODE)
- xAuroraProtectAI-Engine(SDT)
- xAuroraProtectAI-Engine-CORE
- xAuroraRegistryManager
- xAurora-SANDBOX-KernelDriver
- xAurora-SEH(StructuredExceptionsHandler)
- xAuroraSharedMemoryManager
- xAuroraSystemModuleProtect
- xAuroraVirtualFileDisk
- Kernel-Core-Health-Check.exe
- Win32-KernelMode-DriversLoader.exe
- WinHARD-Pro.exe

The "KernelHC -" dialog box displays the following information:

The Kernel Core Health Checker 2008....
Win32-ASM Powered - Sri Lanka Dr. Sam

Kernel32.dll ImageBase: 0x7C800000
User32.dll ImageBase: 0x7E410000

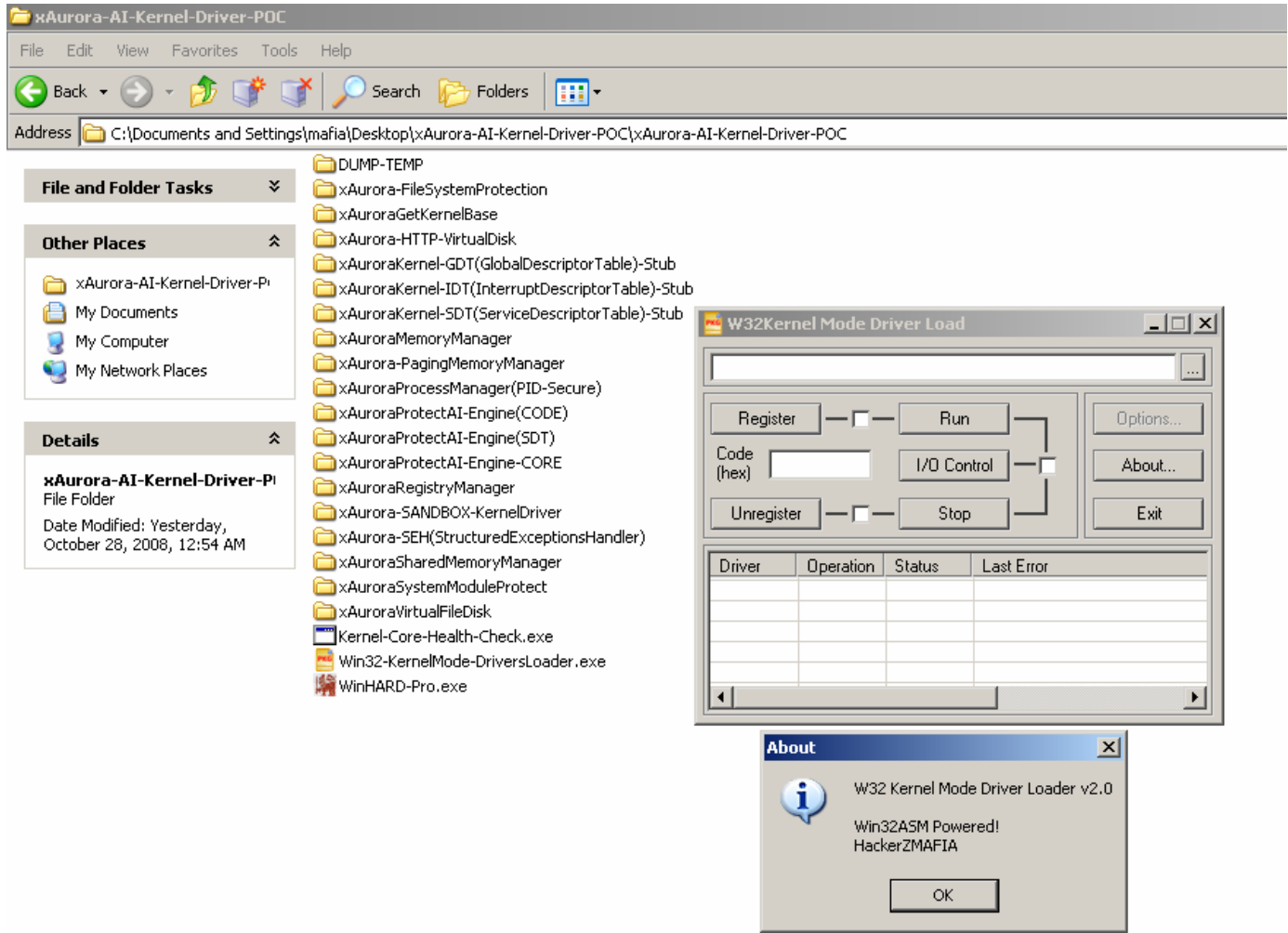
API Addresses:
LoadLibraryA: 0x7C801D7B
GetProcAddress: 0x7C80AE30
ExitProcess: 0x7C81CAFA

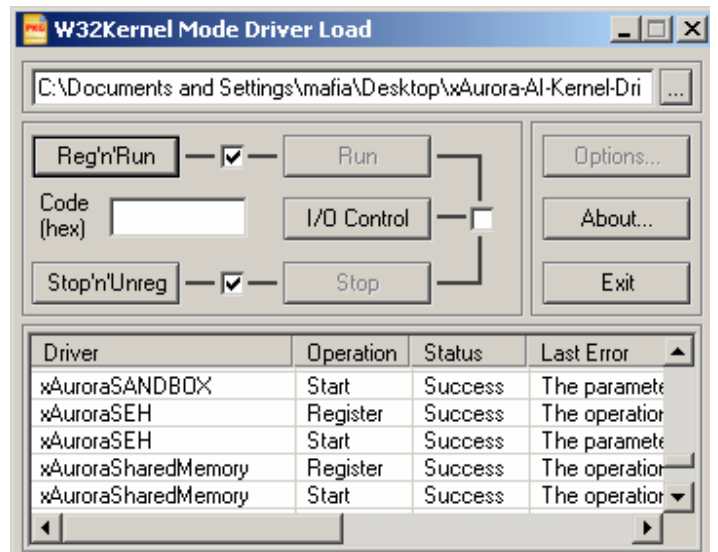
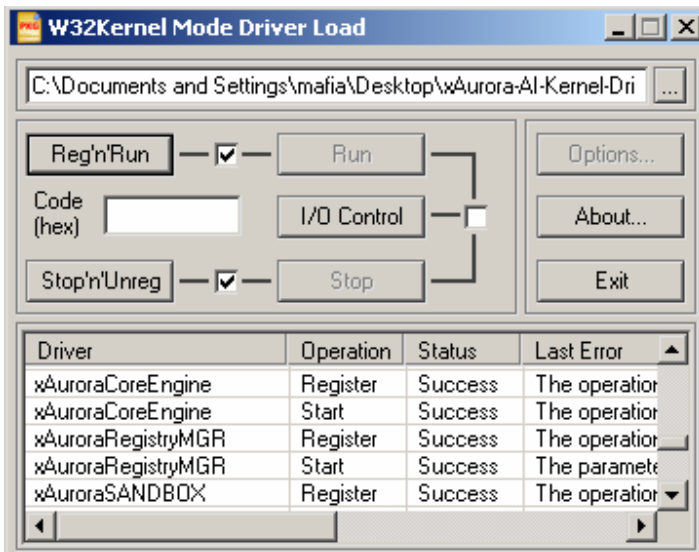
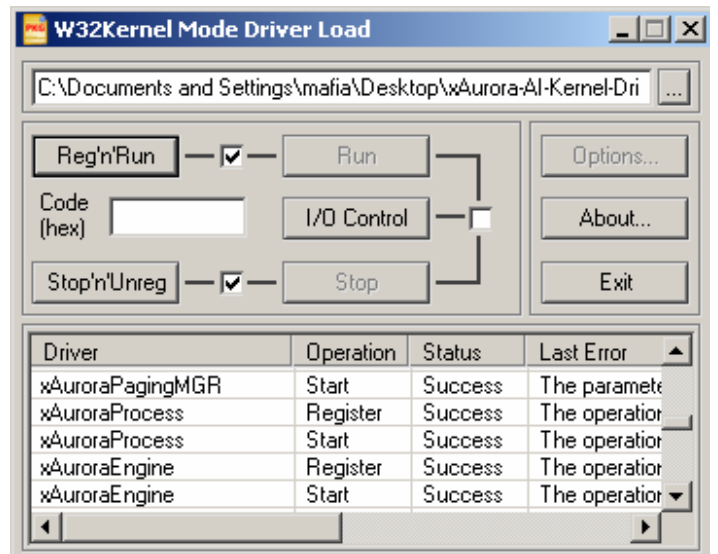
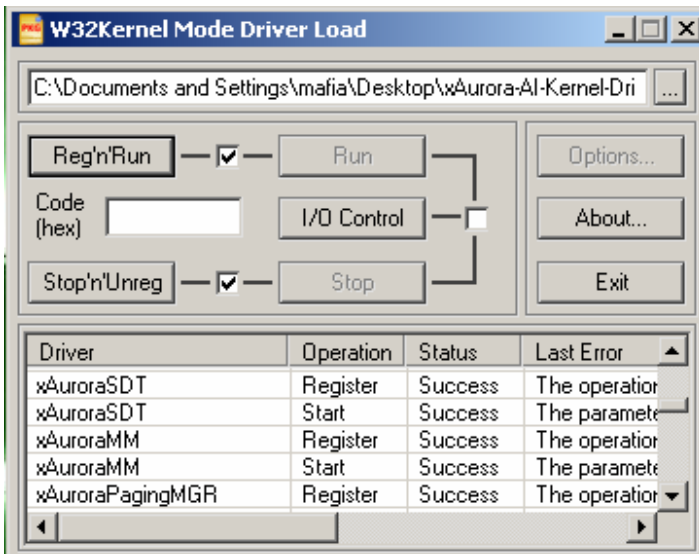
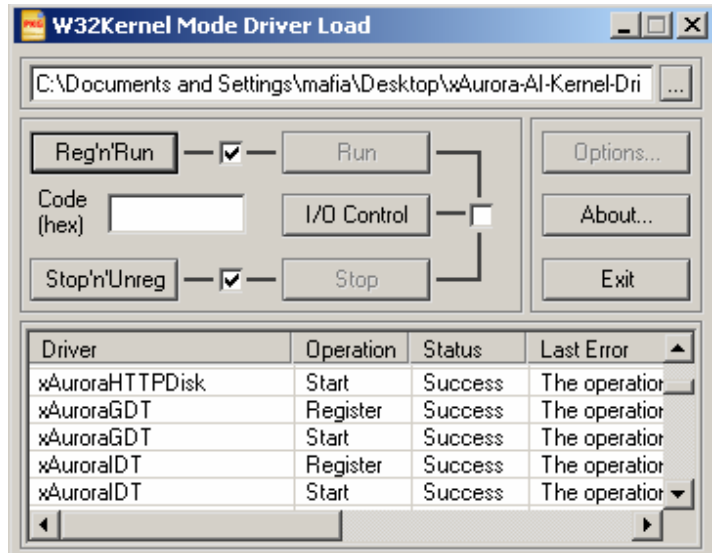
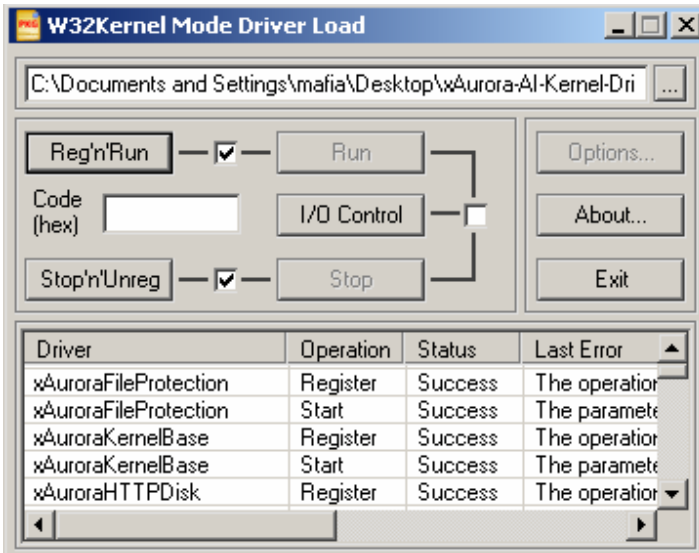
MessageBoxA: 0x7E4507AA
wsprintfA: 0x7E41A8AD

OK

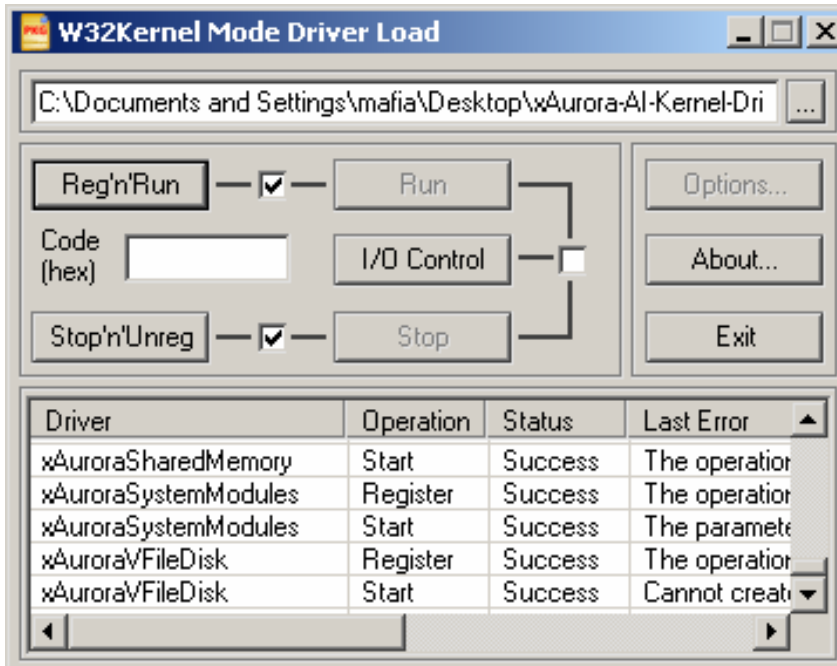
Load the standalone Kernel Mode Drivers of xAurora to the tested healthy Kernel

Driver Loader Started – Now we will load the drivers





Load KMD(s) are in progress



Load KMD(s) are in progress

Check the LOAD status of the KMD(s)

WinHARD - Win32-ASM@LK

Root Help

Driver

Name	Type	Additional Information
Raspti	Driver	Direct Parallel
RDPCDD	Driver	
rdpdr	Driver	Terminal Server Device Redirector Driver
redbook	Driver	Digital CD Audio Playback Filter Driver
s24trans	Driver	WLAN Transport
SDBGMsg	Driver	
swenum	Driver	Software Bus Driver
SynTP	Driver	Synaptics TouchPad Driver
sysaudio	Driver	Microsoft Kernel System Audio Device
SyserBoot	Driver	
SyserLanguage	Driver	
Tcpip	Driver	TCP/IP Protocol Driver
TermDD	Driver	Terminal Device Driver
truecrypt	Driver	
TSKNF800.SYS	Driver	
Update	Driver	Microcode Update Driver
usbhci	Driver	Microsoft USB 2.0 Enhanced Host Cont...
usbhub	Driver	USB2 Enabled Hub
usbuhci	Driver	Microsoft USB Universal Host Controlle...
VgaSave	Driver	
VolSnap	Driver	
Wanarp	Driver	Remote Access IP ARP Driver
wdmaud	Driver	Microsoft WINMM WDM Audio Compati...
Win32k	Driver	
winachsf	Driver	
WinObjEx	Driver	
WMDrive	Driver	
WmiAcpi	Driver	Microsoft Windows Management Interf...
WMIxWDM	Driver	
xAuroraCoreEngine	Driver	
xAuroraEngine	Driver	
xAuroraGDT	Driver	
xAuroraHTTPEDisk	Driver	
xAuroraIDT	Driver	
xAuroraProcess	Driver	
xAuroraSharedMemory	Driver	
{95808DC4-FA4A-4c74-92FE-58863F82066B}	Driver	

Driver

Finding the Stealth Mode - KMD(s) in the Live System

The screenshot shows the WinHARD application interface. On the left is a tree view of system components. The main area displays a table of drivers. A 'Seek Object' dialog box is open, showing a search for 'aurora' in the driver directory. The search results are as follows:

Name	Type	Additional Information
Raspti	Driver	Direct Parallel
RDPCCDD	Driver	
rdpdr	Driver	Terminal Server Device Redirector Driver
redbook	Driver	Digital CD Audio Playback Filter Driver
s24t		
SDBr		
swet		
Syn		
sysa		
Syse		
Syse		
Syse		
Tcpip		
Tern		
true		
TSKf		
Upd.		
usbe		
usb1		
usb2		
Vga		
VolS		
War		
wdr		
Win		
wine		
Win		
WMI		
Wmi		
WMI		
xAuroraCoreEngine	Driver	
xAuroraEngine	Driver	
xAuroraGDT	Driver	
xAuroraHTTPIDisk	Driver	
xAuroraIDT	Driver	
xAuroraProcess	Driver	
xAuroraSharedMemory	Driver	
{95808DC4-FA4A-4c74-92FE-5B863F82066B}	Driver	

The 'Seek Object' dialog box shows the search criteria: 'Type: *' and 'Find' button. The search results are listed in a table below the dialog:

Object	Type
\Driver\xAuroraCoreEngine	Driver
\Driver\xAuroraEngine	Driver
\Driver\xAuroraGDT	Driver
\Driver\xAuroraHTTPIDisk	Driver
\Driver\xAuroraIDT	Driver
\Driver\xAuroraProcess	Driver
\Driver\xAuroraSharedMemory	Driver

Deep Inspection of xAurora Core Engine Stealth Mode - KMD(s)

The screenshot shows the WinHARD application interface. On the left is a tree view of system components. The main pane displays a list of drivers with columns for Name, Type, and Additional Information. The driver 'xAuroraCoreEngine' is selected and highlighted. A 'Driver Properties' dialog box is open over the driver list, showing details for 'xAuroraCoreEngine'.

Name	Type	Additional Information
Rasl2tp	Driver	WAN Miniport (L2TP)
RasPppoe	Driver	Remote Access PPPOE Driver
Raspti	Driver	Direct Parallel
RDPcDD	Driver	
rdpdr	Driver	Terminal Server Device Redirector Driver
redbook	Driver	Digital CD Audio Playback Filter Driver
s24trans	Driver	
SDbgMsg	Driver	
swenum	Driver	
SynTP	Driver	
sysaudio	Driver	
SyserBoot	Driver	
SyserLanguage	Driver	
Tcpip	Driver	
TermDD	Driver	
truecrypt	Driver	
TSKNF800.SYS	Driver	
Update	Driver	
usbehci	Driver	
usbhub	Driver	
usbuhci	Driver	
VgaSave	Driver	
VolSnap	Driver	
Wanarp	Driver	
wdmaud	Driver	
Win32k	Driver	
winachsf	Driver	
WinObjEx	Driver	
WMDrive	Driver	
WmiAcpi	Driver	
WMIxWDM	Driver	
xAuroraCoreEngine	Driver	
xAuroraEngine	Driver	
xAuroraGDT	Driver	
xAuroraHTTPEDisk	Driver	
xAuroraIDT	Driver	
xAuroraProcess	Driver	

Driver Properties

Basic | Object | Registry | Type

General Information

Name: xAuroraCoreEngine
Type: Driver

Attributes: Permanent
References: 3
Handles: 0
Header: 0x862895D0
Object: 0x862895E8

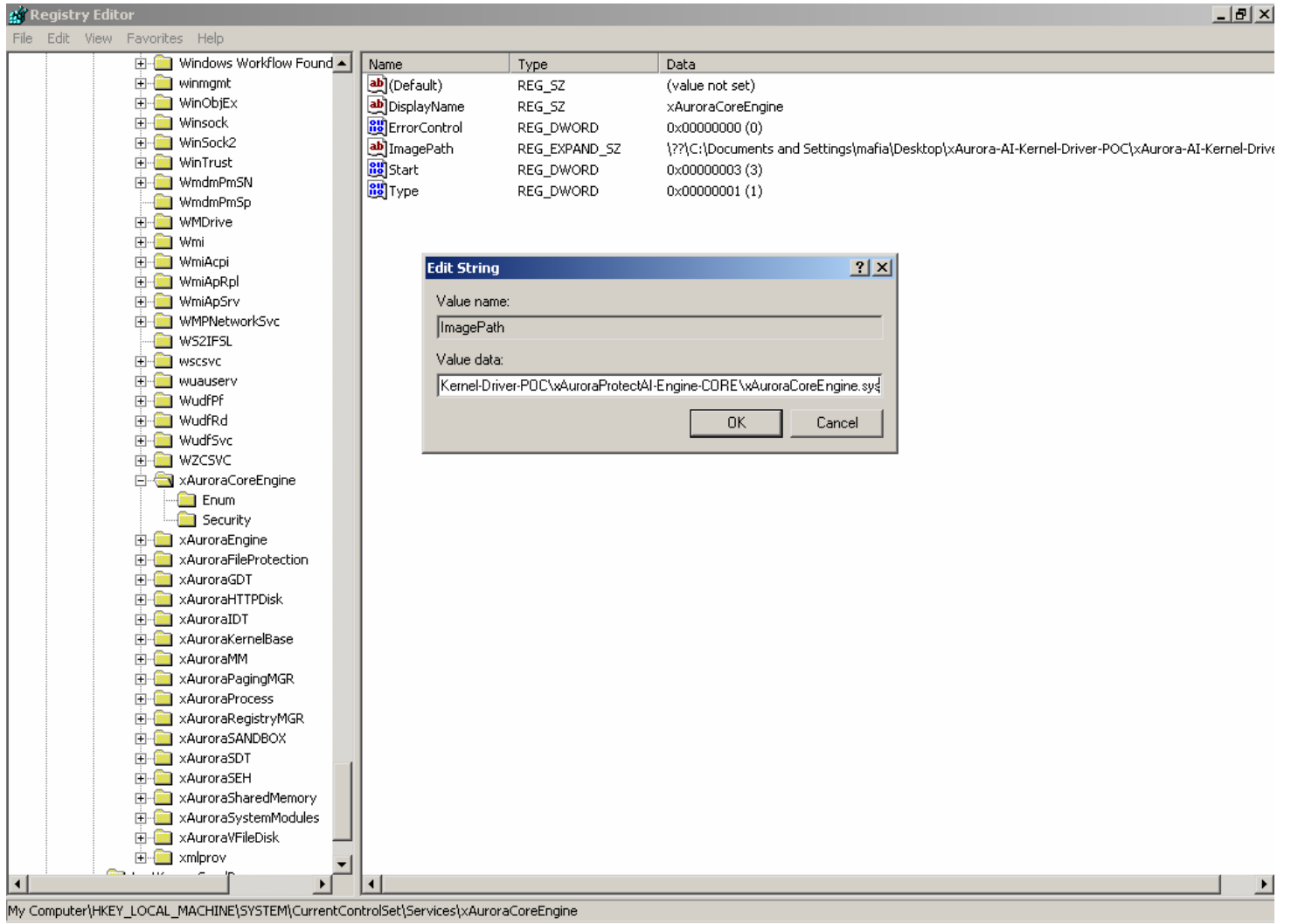
Paged Pool Charge: 0
Nonpaged Pool Charge: 0

Display Name

OK Cancel Apply

\\Driver\\xAuroraCoreEngine

xAurora Core Engine Stealth Mode Driver Path Info in Registry



xAurora Kernel Mode Drivers Bound/Used with (xAurora.EXE) File

The screenshot displays the Resource Browser interface. On the left, a tree view shows the 'KERNELMODEDRIVERS' folder expanded, listing various drivers such as XAURORA-FILESYSTEMPROTECTION, XAURORA-HTTP-VIRTUALDISK, XAURORA-PAGINGMEMORYMANAGER, XAURORA-SANDBOX-KERNELDRIVER, XAURORA-SEH(STRUCTUREXCEPTIONSHANDLER), XAURORA-X-HOOKSPROTECTSERVER, XAURORAGETKERNELBASE, XAURORAKERNEL-GDT(GLOBALDESCRIPTORABLE)-STUB, XAURORAKERNEL-IDT(INTERRUPTDESCRIPTORABLE)-STUB, XAURORAKERNEL-SDT(SERVICEDESCRIPTORABLE)-STUB, XAURORAMEMORYMANAGER, XAURORAPROCESSMANAGER(PID-SECURE), XAURORAPROTECTAI-ENGINE(CODE), XAURORAPROTECTAI-ENGINE(SDT), XAURORAPROTECTAI-ENGINE-CORE, XAURORAREGISTRYMANAGER, XAURORASHAREDMEMORYMANAGER, XAURORASYSTEMMODULEPROTECT, and XAURORAVIRTUALFILEDISK. The right pane shows a hex dump of resource data, including the text: 'This program cannot be run in DOS mode', 'TheThemida6d', and 'TheThemida 0'. The hex dump also contains strings like 'Company Name: Bolobrant', 'Product Name: TheThemida', and 'Legal Copyright: Bolobrant'.

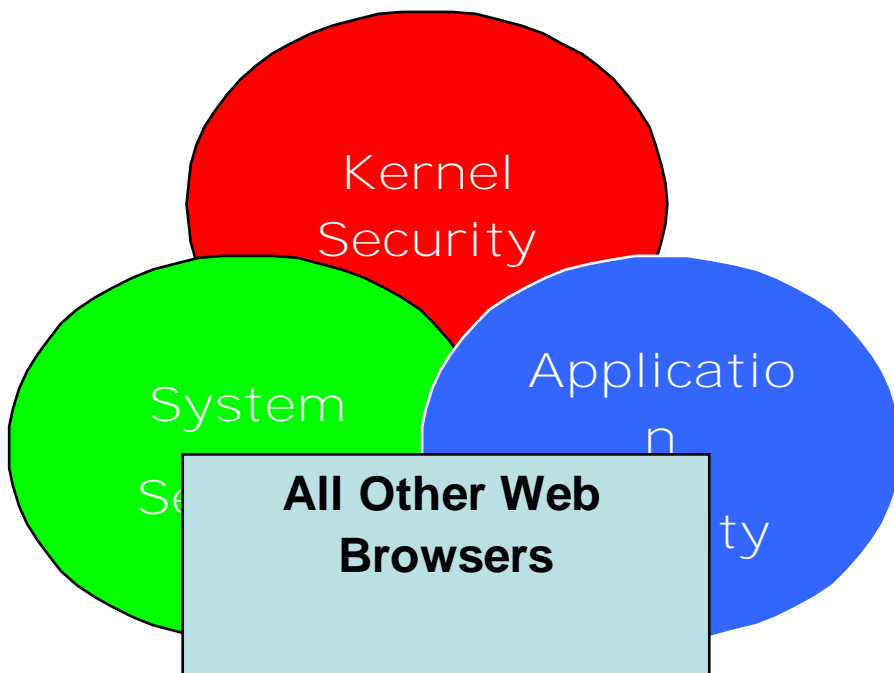
xAurora is the ONLY and FIRST browser in the world that running on CPU Ring0 (Kernel Mode), Artificial Intelligence (AI) and Offline Firewall (Pre HIPS Kind), Almost all the important browser workflow and browser core components are working in the Kernel Mode, not in the NTDLL/Marshalling Core (User Mode). All other browsers are working in USER MODE (Including Maxthon). But the shell and few other components are working in USER Mode. It has 2 rendering mechanisms,

1. Based on TRIDENT – Internet Explorer Core
2. Based on Artificial Intelligent own xAurora Core

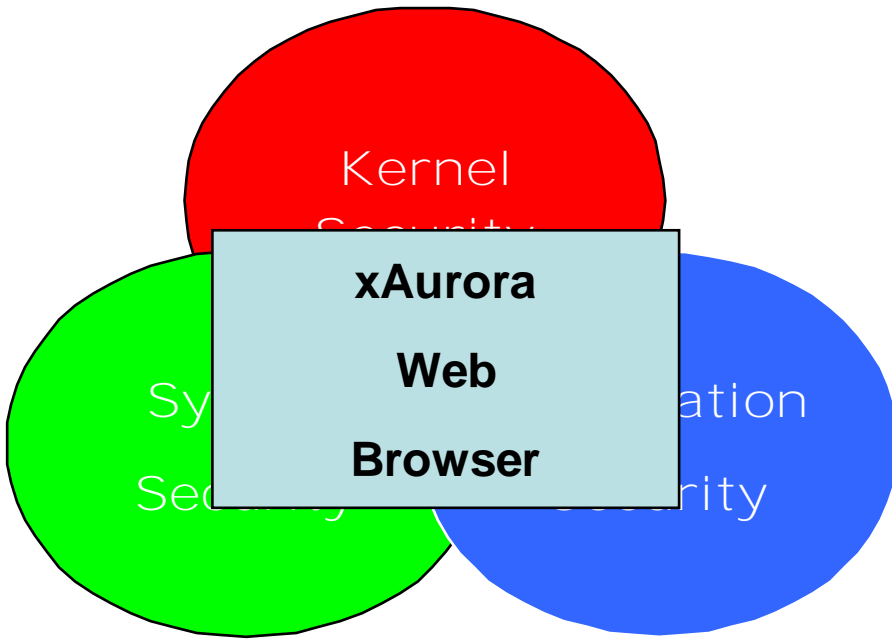
Currently xAurora doesn't support Gecko Rendering Engine. But, Maxthon supports Gecko Rendering Engine.

We tried to achieve Supreme Speed by working with Kernel Mode Core TCP/IP and RPC Layer, Extreme Safe with Sandboxing, Encrypted Kernel Hooks and many security tricks, Ultimate Stability with Kernel Based Anti-Crash Engine. This was totally written in Win32-Microsoft Macro Assembler v8.20 and LK-ASM (This Assembler Compiler was written by me).

Web Browser Security Dilemma Comparison

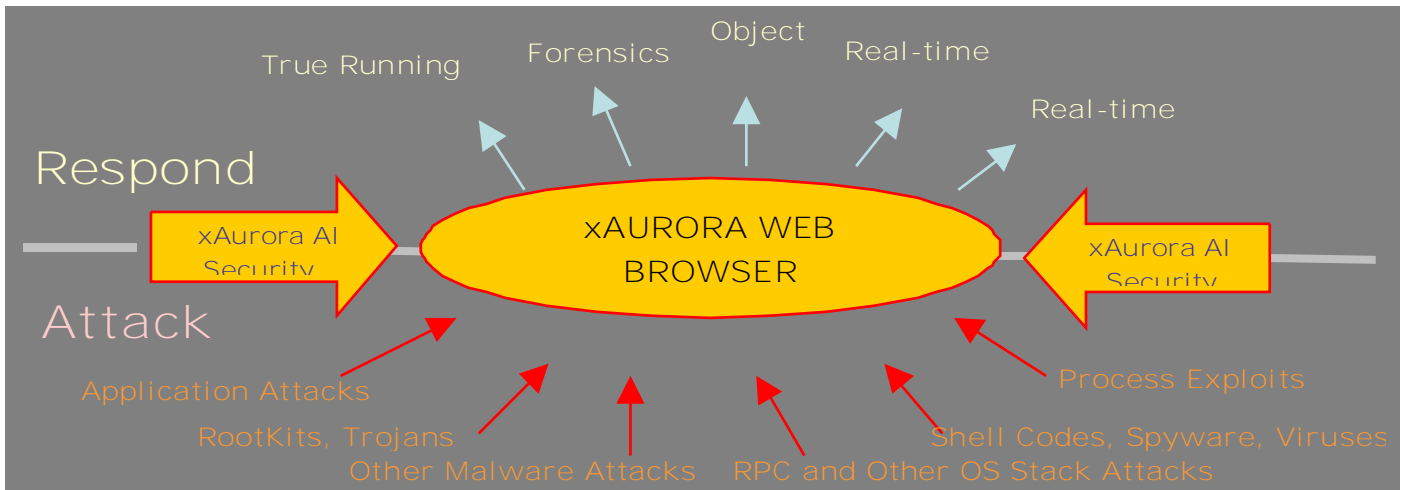


Existing Web Browser's Security Model

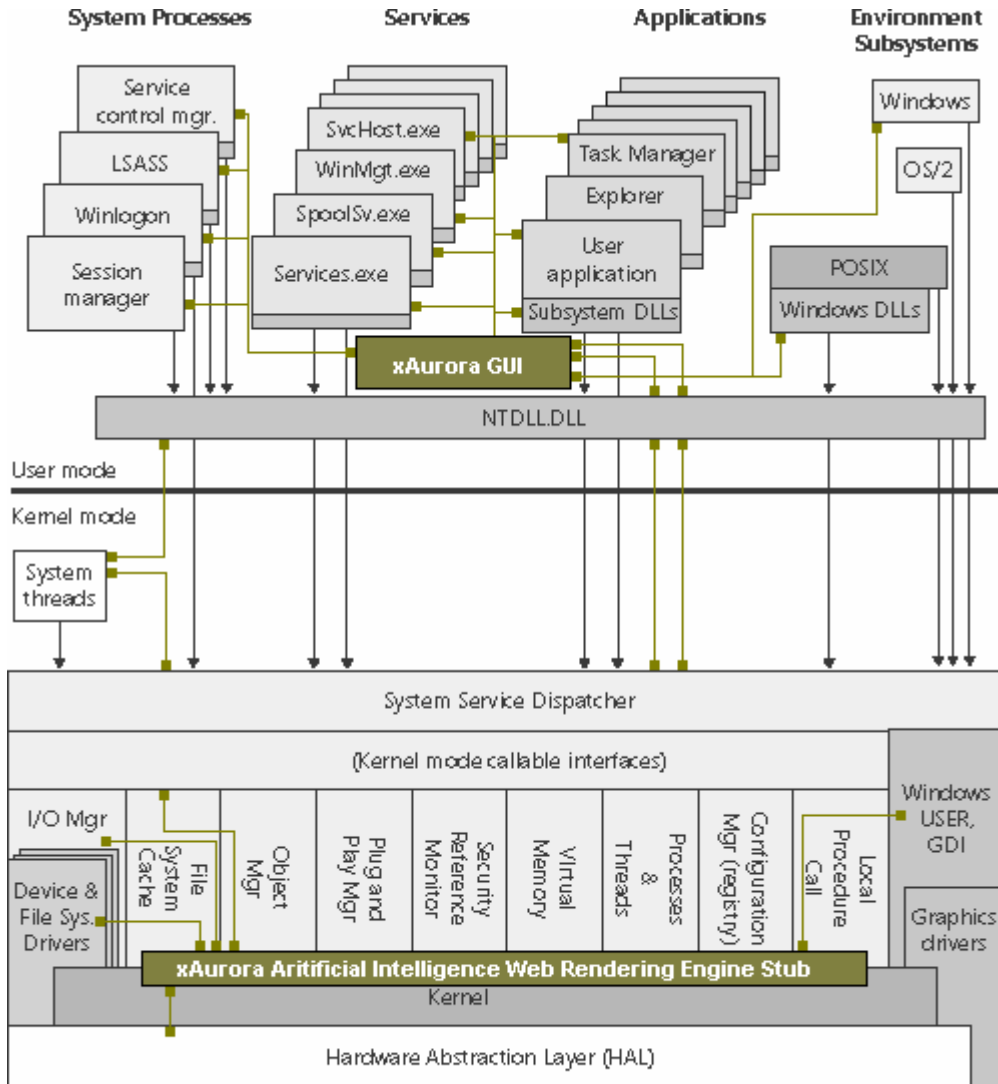


xAurora's Security Model

xAurora AI Core Technology



xAurora Placement and Function Flows in Windows Architecture



Hardware interfaces (buses, I/O devices, interrupts, interval timers, DMA, memory cache control, etc.)

xAurora User Mode (GUI) Engine

USER MODE

Favorites and Windows Tray Controller, Toolbar/Button/Interface and Zoom Page Controller, Redo-Undo, Fonts Manager	Auto-Save, Native Browser Functions Manager, Offline/Online Module, Browser Hide/Show, TAB Module, Lock/Unlock Password Manager
Artificial Intelligent Web Content Filter, AI Web AD-POPUP Filter, AI Web Banner-Toolbar Filter, System Garbage Cleaner and Collector, Threat Quarantine Module, Skin Manager, Browser Customization Manager and URL Filter Manager	Web Encode/Decode Manager, Download Controller, Alias/Groups/Key Manager, Automated Browser Module Manager, IE and Network Settings Manager, xAurora System Settings and Options Manager
Web Cache, Proxy Cache, DNS Cache, Web Blackhole Detection, Blacklist IP Management, HTTP and Protocol Stack Manager	RSS and ATOM Feed Management, Window Controller, External Utility, Plugins, Language Translation and Unicode Management Module
Credential Manager, Spool Manager, Service Control Module and Top Level Security Manager	WMI, LSASS, DLL Subsystem Module, SVCHOST and INETINFO Management Module
Task Management, CPU Thread Management and High Level Network Services Controller	
xAurora GUI with Other Core Components	

xAurora AI Kernel Mode Engine

KERNEL MODE

NTLM Authentication Interaction and Application Layer Gateway Module, Background Intelligent Transfer Network Call and Integration Module	Font Smoothing and Anti-Alias Engine, Panel and Sidebar Module, XP-SP2/2000-SP4/2003-R2-SP1 and VISTA Security Framework Manager
Seek 'N Destroy Memory Manager, Virtual and Real Memory Module, Context Menu Handlers, Active Live Tracking Module, BHO Manager, Bandwidth Manager, CPU Ring Push/Switch Manager, SHELL Integration Management Module	Interface Manager, Instance Manager, Web Script Manager, Vulnerability and Threat / Malware Detection/Prevention Module, Self Learning Module, Fuzzy Logic Engine, TTL/QoS Manager, CPU Process Module, Debug Code Engine Module
URL/Web Services and Search Engine Manager, Duplicated URL and ActiveX Low Level Filter, DDE Call and DSDM Call Manager	DotNET Framework and Browser Cache Usage Manager, JAVA Virtual Machine Module, Thread Module, MTU-TCP/IP Window/Frame Module
DEP, SEH, Terminal Server Module, Multi CPU structure handling Module, IPsec Manager	Anti-Crash and Crash Recovery Module, CPU Priority Manager, CPU 16/32/64 Bits Manager
HT/SSE/AMD-INTEL ITANIUM Hardware Support, Mouse/Keyboard Actions Manager, Hook Manager	
Kernel Mode Artificial Intelligence Web Rendering Engine @ CPU RING (0) and HAL	

Antihook Server-Client Architecture (Kernel Mode-User Mode)

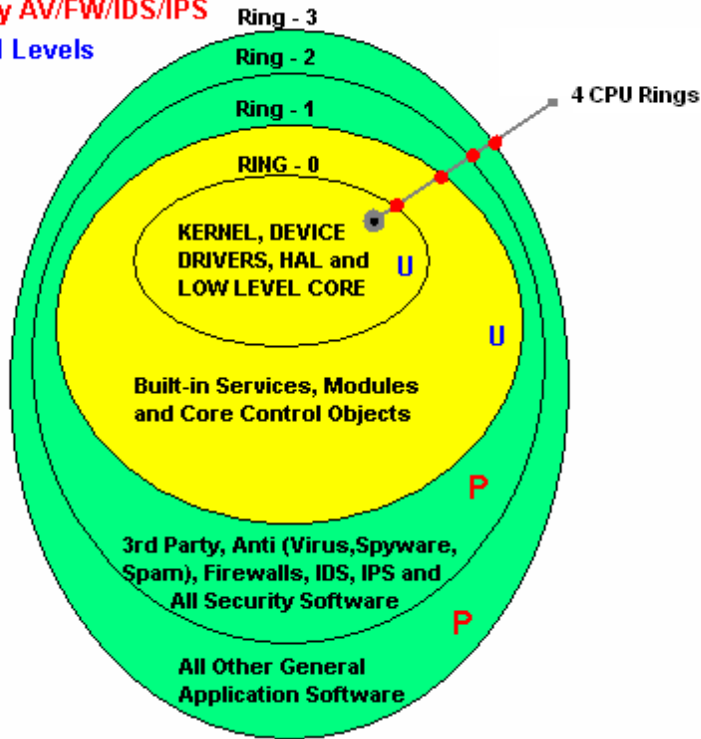
Antihook Server-Client Security Model specially designed to prevent the malicious code hooking/attaching into the Operating Systems Kernel. This is total Artificial Intelligence Module based on High Interaction HIPS, unlike Windows VISTA HIPS Model. Windows VISTA HIPS Client always depends on the USERS Interaction and USERS knowledge, especially for a non technical person can not understand the HIPS Interaction Messages. Rootkits, Shellcodes and modern malware (Ex. Virus like KIDO) runs on Kernel space to achieve the maximum performance and damage. Because the Kernel space is the most privilege area in any Operating System. Once Rootkit or any other Kernel driven malware infected, that is going to be the most terrifying state for any of the PC/Server. It is very difficult to remove, and some are they will never clean 100% from the system. We need special tools, special knowledge and special methods to remove them from the system and it is very time consuming job.

Malware like Rootkits will hook/attach their low level components (Ex. Rogue Device Drivers) to the Kernel by exploiting some vulnerability, using some misconfiguration or using wrong USER interaction. These types of attacks can not be prevented by installing Anti-Virus, because, Anti-Viruses most of the core components are running on CPU Ring2 or sometimes in Ring1, very rarely runs on CPU Ring0, but that core component(s) are not going to protect CPU Ring0 or Ring1 (Kernel Space), because, program like Anti-Virus has dynamic multi CPU threading process, Kernel space can be processed only very specific selected CPU threads and Kernel hooks, therefore if the Anti-Virus tries to open dynamic multi threading on CPU Ring0, Operating System will gets crash by occurring KERNEL PANIC.

Most Rootkit attacks are coming through the web; therefore, web browsers are the most vulnerable entry point to the Operating Systems for the Rootkits kind of malware attacks. Unfortunately, none of the web browsers can prevent these types of attacks.

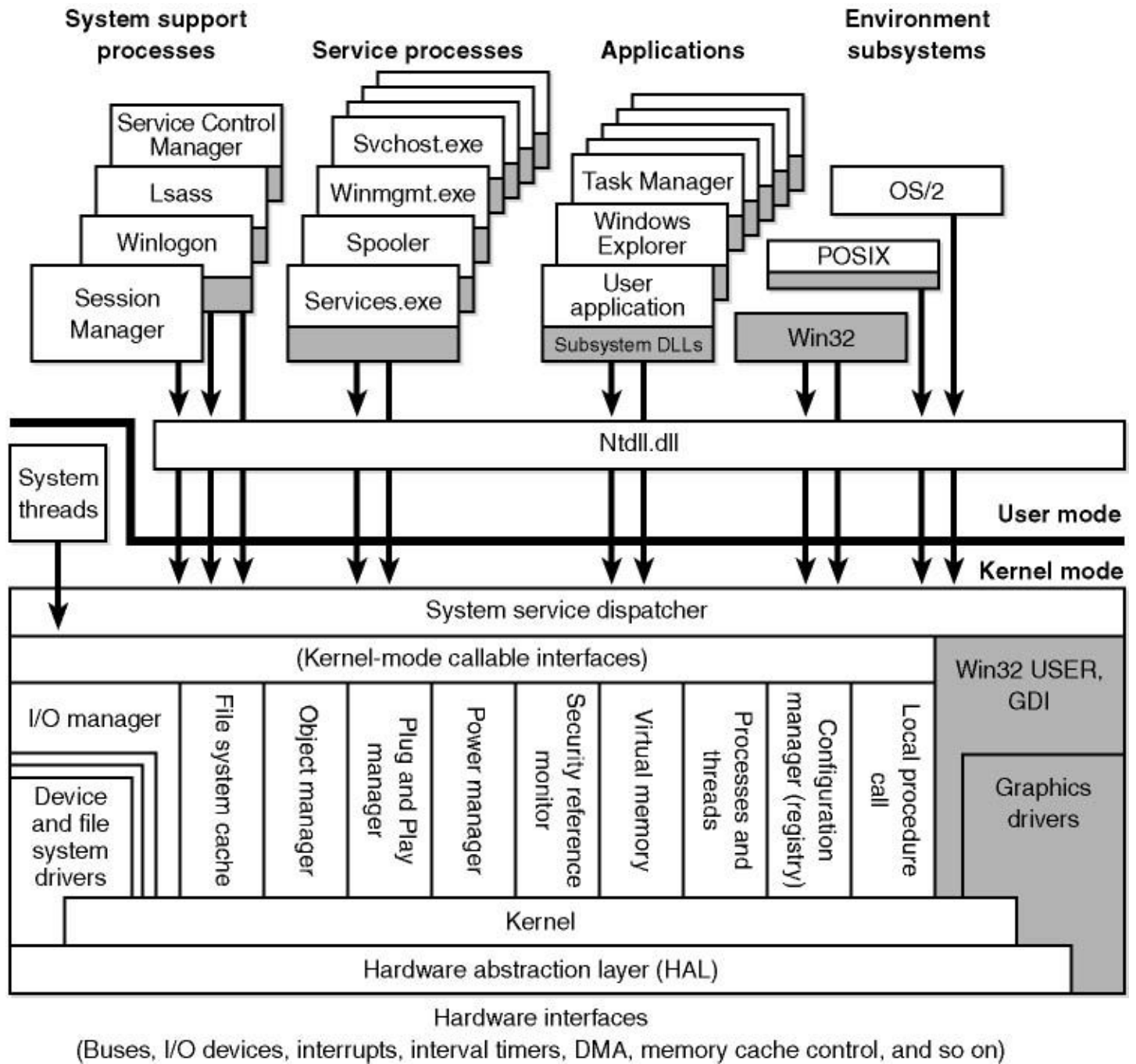
P = Protected By AV/FW/IDS/IPS

U = Unprotected Levels

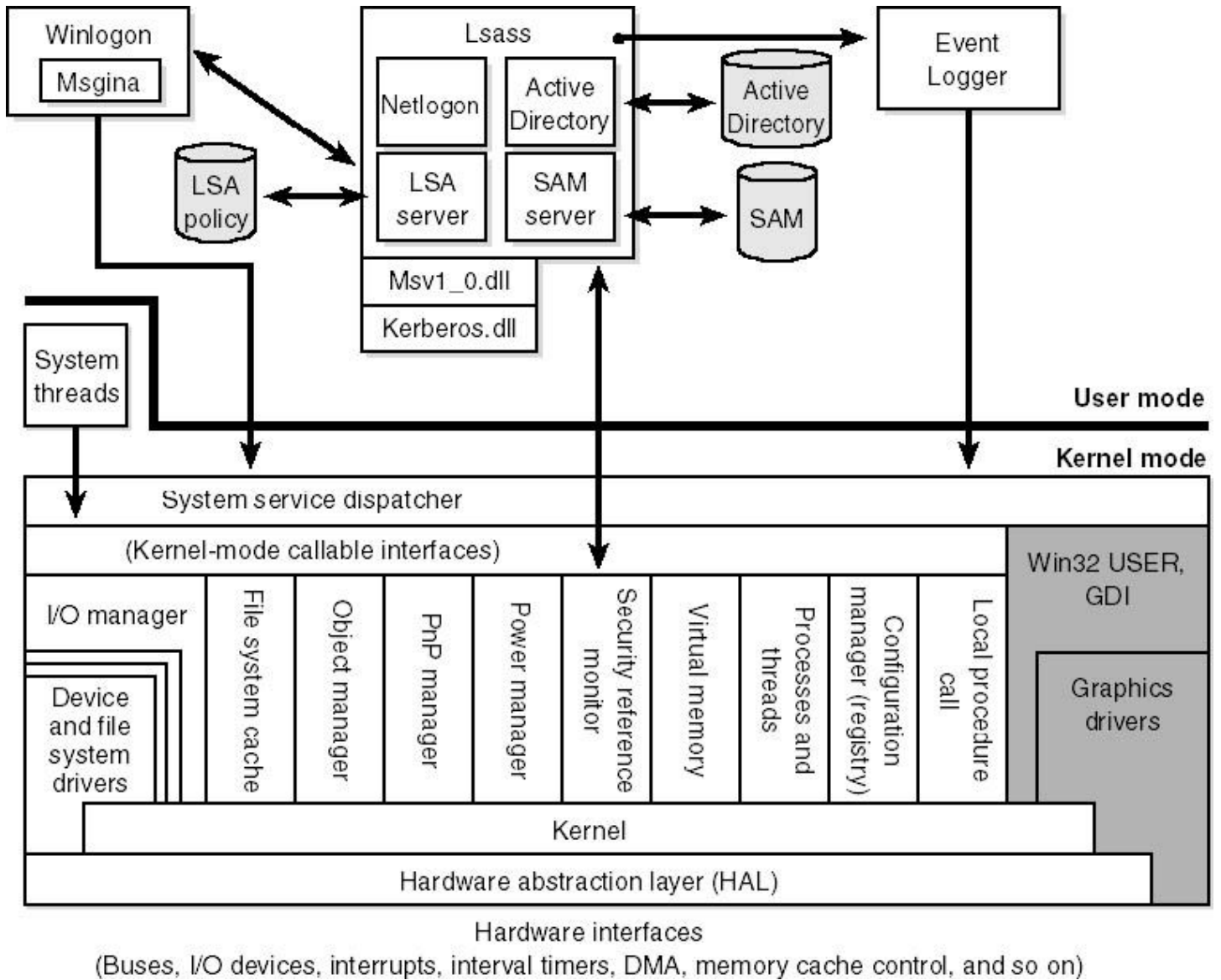


But, xAurora has the right and revolutionary technology/mechanism that can prevent Rootkit kind of malware attacks entirely. It has the World's first artificial intelligent based Brand New Anti-Hooking Client-Server Core for the revolutionary protection from Malware attacks to the Operating Systems Core. We achieved this task only with less than 100 Kb of Anti-Hook Core Client-Server components. That is the power of Assembler.

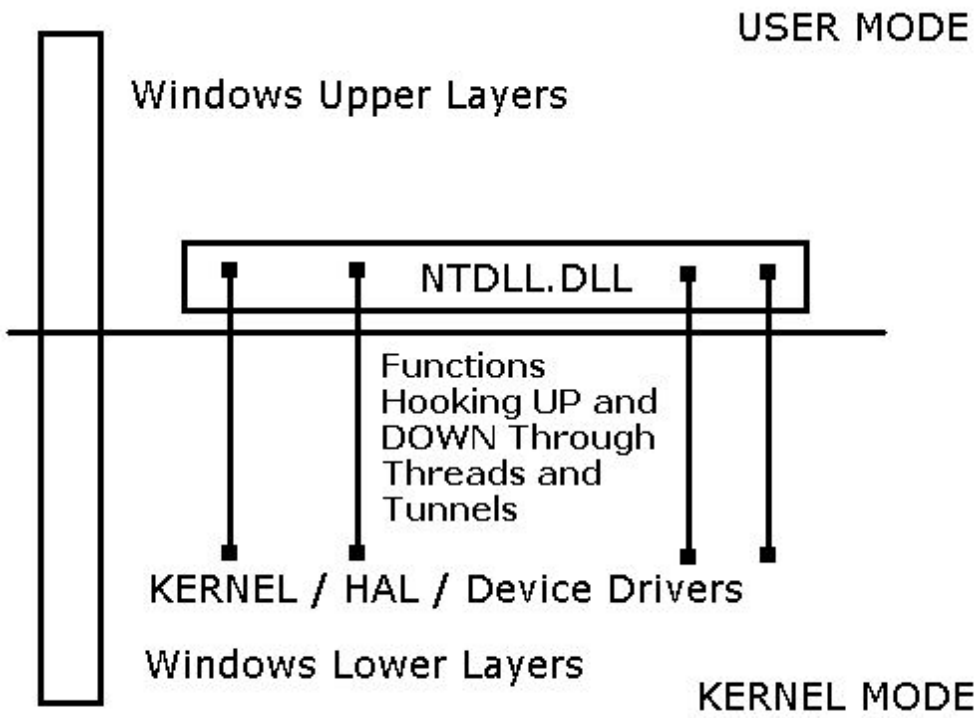
Native Windows NT Architecture



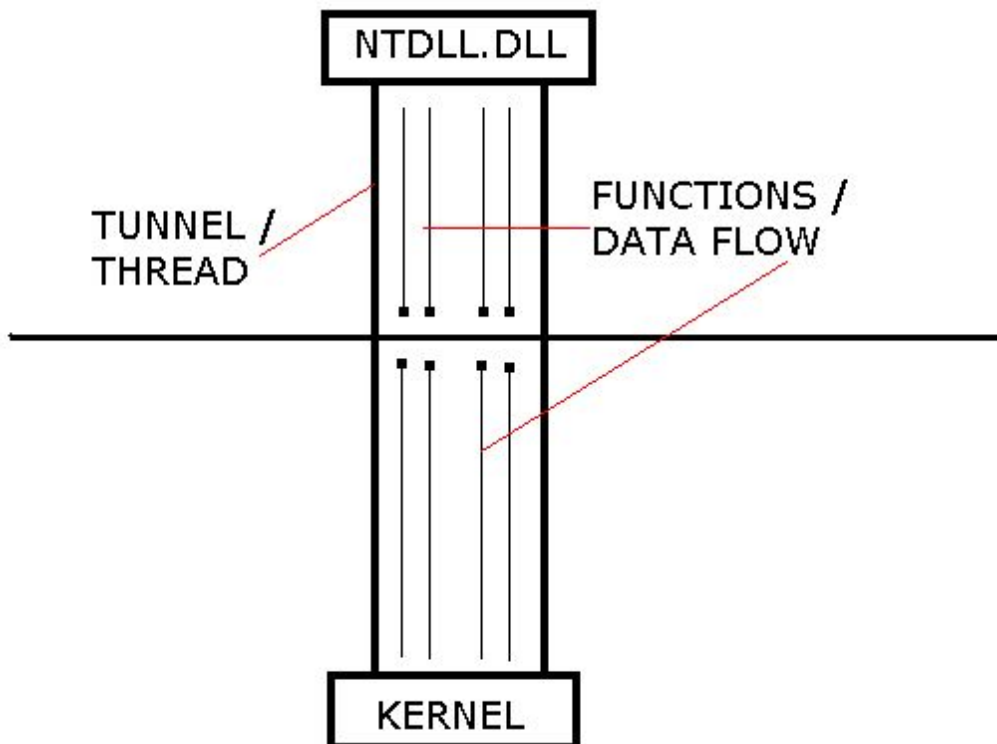
Native Windows NT Security Architecture



Basic Function Flows in Windows Core



Basic NTDLL (Marshalling Library) and Kernel Interaction in Windows



xAurora Extreme Protection Model (PoC)

Achieving Maximum Protection via xAurora Anti-Hook Client and Anti-Hook Server

Name	Size	Type	Date Modified
HookProtection.exe	98 KB	Application	11/8/2008 00:32
HookProtection.ini	1 KB	Configuration Settings	11/8/2008 00:34

Anti-Hook

Processes: 29

PID	Name
0	<System Idle>
4	<System>
404	smss.exe
452	csrss.exe
476	winlogon.exe
524	services.exe
536	lsass.exe
708	svchost.exe
772	svchost.exe
812	svchost.exe
864	svchost.exe
892	svchost.exe
920	spoolsv.exe
1024	inetinfo.exe
1064	nTuneService.exe
1156	nvsvc32.exe
1204	wdfmgr.exe
1276	UpdateCenterService.exe
1564	explorer.exe
1784	CTSysVol.exe
1796	rundll32.exe
1824	rundll32.exe
1832	UnlockerAssistant.exe
1840	USBSafelyRemove.exe
1184	wcescomm.exe

Paths: 0

Path

Options

Self Settings

- Hide Process
- Protect Process
- Hide Windows
- Protect From Windows Hooks

Global Settings

- Reboot Protection
- Format Protection
- Parent Process Emulation
- Anti-Anti Debug
- Auto Start

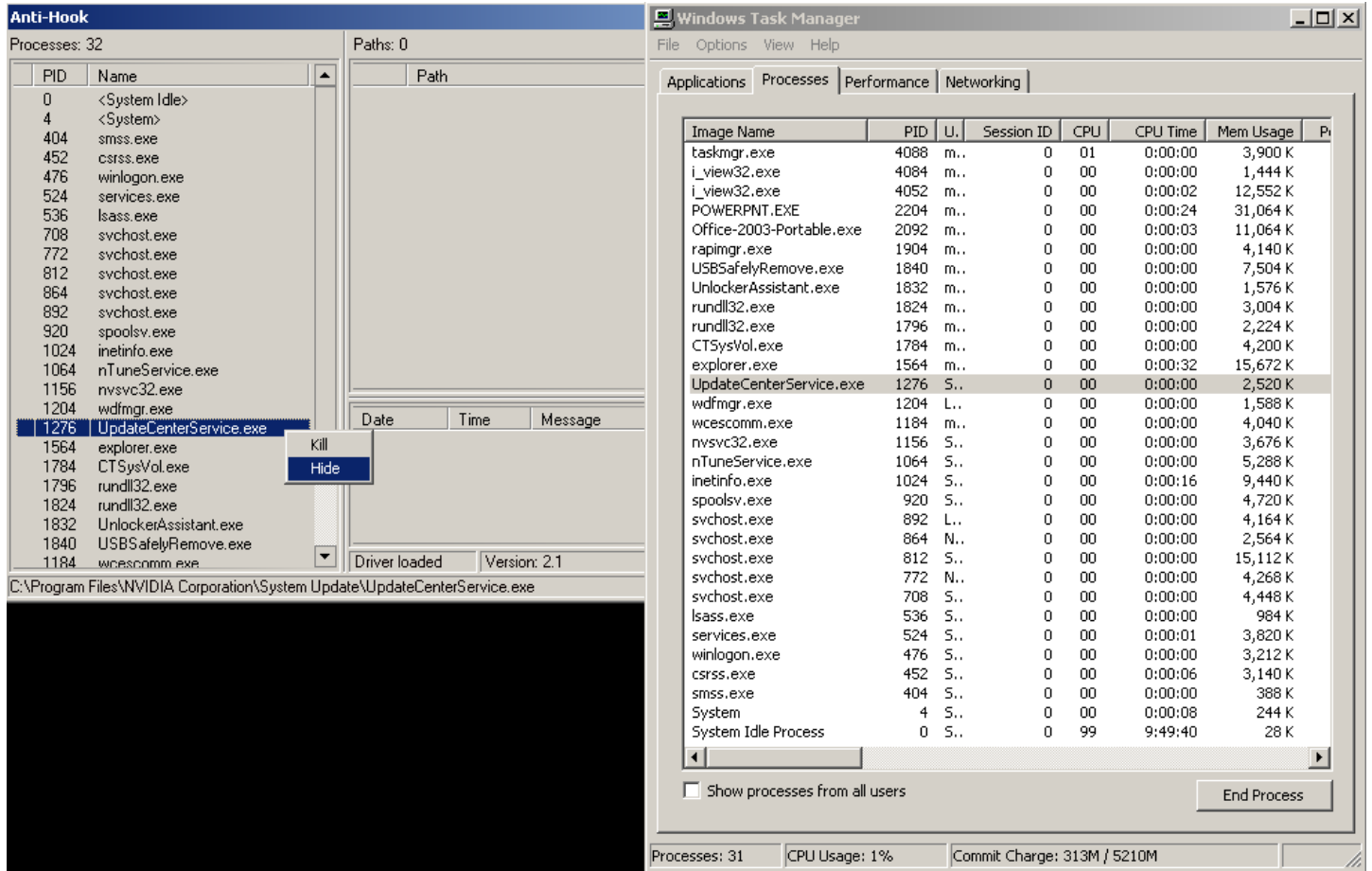
OK Cancel

Driver loaded | Version: 2.1

C:\Documents and Settings\mafia\Desktop\Final\Anti-HookProtectionClient\HookProtection.exe

Anti-Hook Client Side Settings

Hide Processes from Task Manager by Enforcing Windows Kernel – Hide



Show Hidden Processes from Task Manager by Enforcing Windows Kernel - Show

Anti-Hook

Processes: 32 Paths: 0

PID	Name
0	<System Idle>
4	<System>
404	smss.exe
452	csrss.exe
476	winlogon.exe
524	services.exe
536	lsass.exe
708	svchost.exe
772	svchost.exe
812	svchost.exe
864	svchost.exe
892	svchost.exe
920	spoolsv.exe
1024	inetinfo.exe
1064	nTuneService.exe
1156	nvsvc32.exe
1204	wdfmgr.exe
1276	UpdateCenterService.exe
1564	explorer.exe
1784	CTSysVol.exe
1796	rundll32.exe
1824	rundll32.exe
1832	UnlockerAssistant.exe
1840	USBSafelyRemove.exe
1184	wcescomm.exe

Driver loaded Version: 2.1

C:\Program Files\NVIDIA Corporation\System Update\UpdateCenterService.exe

Windows Task Manager

File Options View Help

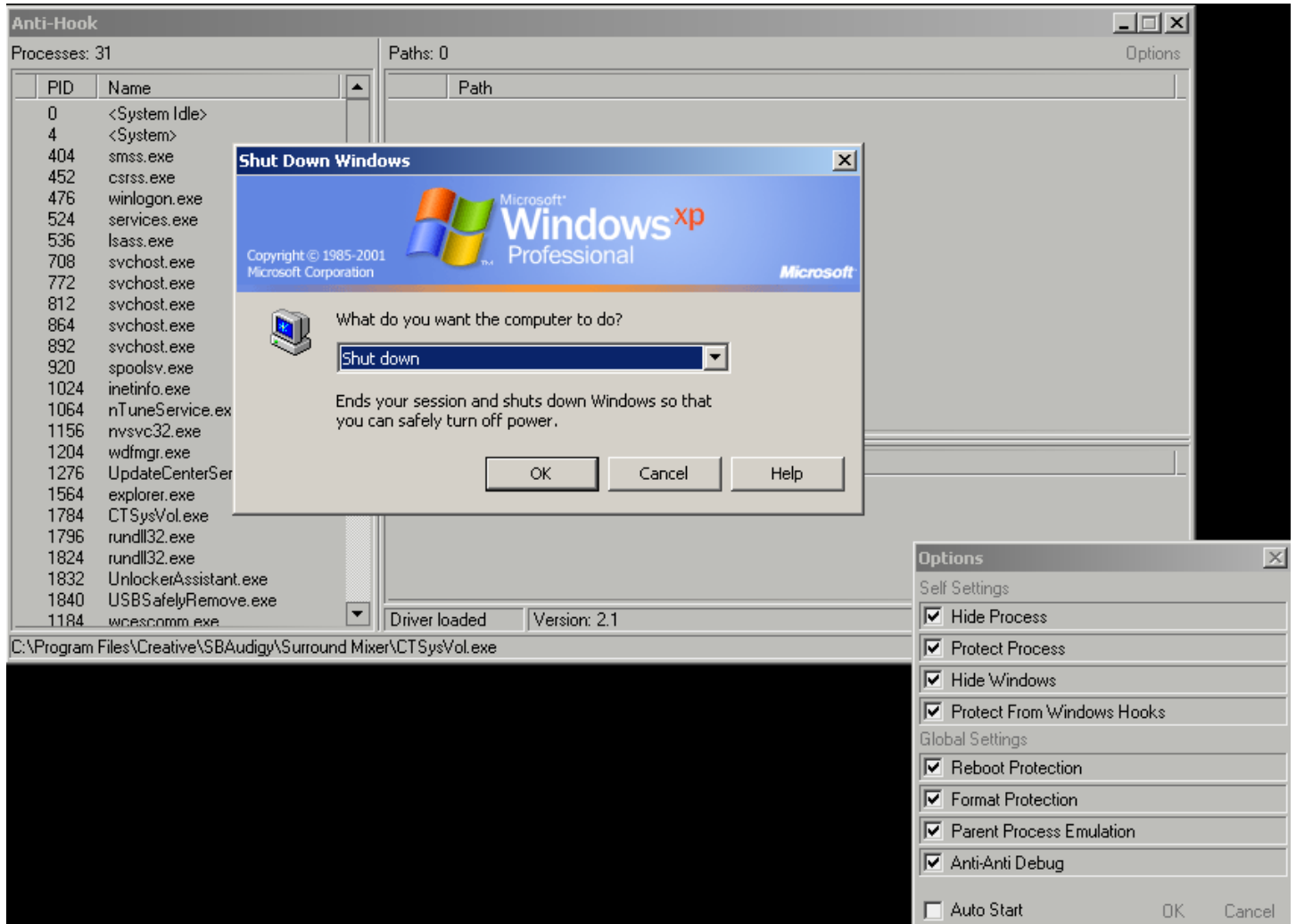
Applications Processes Performance Networking

Image Name	PID	U.	Session ID	CPU	CPU Time	Mem Usage	Pri
taskmgr.exe	4088	m..	0	01	0:00:01	1,532 K	
i_view32.exe	4084	m..	0	00	0:00:00	3,436 K	
i_view32.exe	4052	m..	0	00	0:00:02	12,552 K	
POWERPNT.EXE	2204	m..	0	00	0:00:25	21,312 K	
Office-2003-Portable.exe	2092	m..	0	00	0:00:03	11,064 K	
rapimg.exe	1904	m..	0	00	0:00:00	4,140 K	
USBSafelyRemove.exe	1840	m..	0	00	0:00:00	7,504 K	
UnlockerAssistant.exe	1832	m..	0	00	0:00:00	1,576 K	
rundll32.exe	1824	m..	0	00	0:00:00	3,004 K	
rundll32.exe	1796	m..	0	00	0:00:00	2,224 K	
CTSysVol.exe	1784	m..	0	00	0:00:00	4,200 K	
explorer.exe	1564	m..	0	00	0:00:32	15,672 K	
wdfmgr.exe	1204	L..	0	00	0:00:00	1,588 K	
wcescomm.exe	1184	m..	0	00	0:00:00	4,040 K	
nvsvc32.exe	1156	S..	0	00	0:00:00	3,676 K	
nTuneService.exe	1064	S..	0	00	0:00:00	5,288 K	
inetinfo.exe	1024	S..	0	00	0:00:16	9,440 K	
spoolsv.exe	920	S..	0	00	0:00:00	4,720 K	
svchost.exe	892	L..	0	00	0:00:00	4,164 K	
svchost.exe	864	N..	0	00	0:00:00	2,564 K	
svchost.exe	812	S..	0	00	0:00:00	15,112 K	
svchost.exe	772	N..	0	00	0:00:00	4,268 K	
svchost.exe	708	S..	0	00	0:00:00	4,448 K	
lsass.exe	536	S..	0	00	0:00:00	984 K	
services.exe	524	S..	0	00	0:00:01	3,820 K	
winlogon.exe	476	S..	0	00	0:00:00	3,212 K	
csrss.exe	452	S..	0	00	0:00:06	3,140 K	
smss.exe	404	S..	0	00	0:00:00	388 K	
System	4	S..	0	00	0:00:08	244 K	
System Idle Process	0	S..	0	99	9:51:50	28 K	

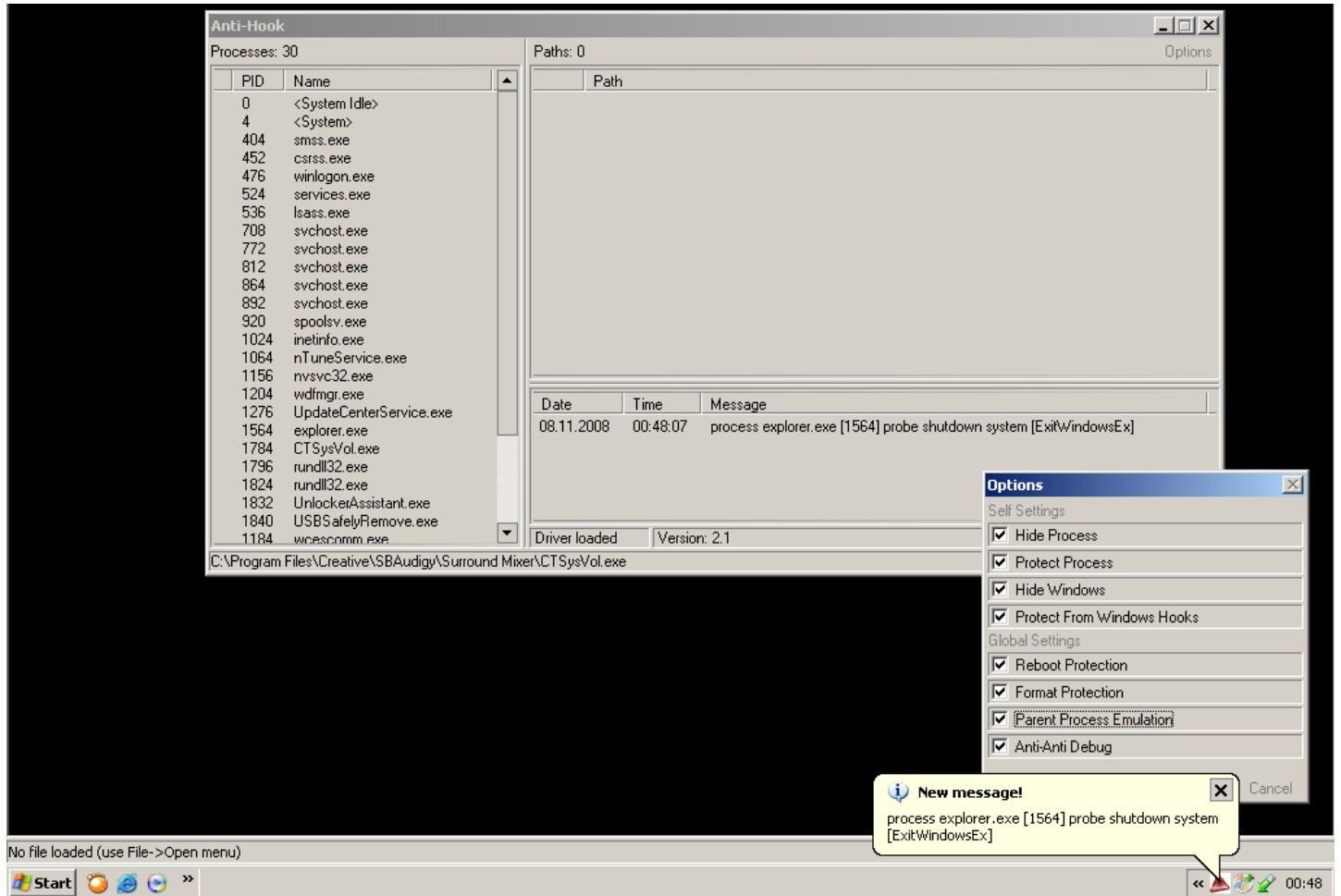
Show processes from all users End Process

Processes: 30 CPU Usage: 2% Commit Charge: 317M / 5210M

Windows Reboot / Shutdown Protection - Shutdown Initiated



Windows Reboot / Shutdown Aborted by Enforcing Windows Kernel API(s)



Format Operation Initiated

Name	Type	Total Size	Free Space	Comments
Hard Disk Drives				
Local Disk (C:)	Local Disk			
SAM-Backup-UP (D:)	Local Disk			
SAM-BACK3 (E:)	Local Disk			
SAM-BACK2 (G:)	Local Disk			
SAM-BACK4 (H:)	Local Disk			
Devices with Removable Storage				
Removable Disk (F:)	Removable Disk			
DVD-RAM Drive (I:)	CD Drive			
DVD Drive (J:)	CD Drive			
Other				
Control Panel	System Folder			
Mobile Device	System Folder			

Formatting Removable Disk (F:)

Capacity: 124 MB

File system: FAT32

Allocation unit size: Default allocation size

Volume label:

Format options:
 Quick Format
 Enable Compression
 Create an MS-DOS startup disk

Start Cancel

Formatting Removable Disk (F:)

Format Complete.

OK

Provides options for ...

Click to safely remove the device

Generic Flash Disk USB Device (F:)

Format Operation Aborted by Enforcing Windows Device Driver Hooks in Kernel

Name	Type	Total Size	Free Space	Comments
Hard Disk Drives				
Local Disk (C:)	Local Disk	19.5 GB	13.7 GB	
SAM-Backup-UP (D:)	Local Disk	74.5 GB	14.4 GB	
SAM-BACK3 (E:)	Local Disk	19.5 GB	5.57 GB	
SAM-BACK2 (G:)	Local Disk			
SAM-BACK4 (H:)	Local Disk			

Devices with Removable Storage	
Anti-Hook	Processes: 28 Paths: 0
Options	
Self Settings	
<input checked="" type="checkbox"/> Hide Process	
<input checked="" type="checkbox"/> Protect Process	
<input checked="" type="checkbox"/> Hide Windows	
<input checked="" type="checkbox"/> Protect From Windows Hooks	
Global Settings	
<input checked="" type="checkbox"/> Reboot Protection	
<input checked="" type="checkbox"/> Format Protection	
<input checked="" type="checkbox"/> Parent Process Emulation	
<input checked="" type="checkbox"/> Anti-Anti Debug	
<input type="checkbox"/> Auto Start	OK Cancel

Time	Process	Action
08.11.2008 01:08:47	process explorer.exe [1700]	write \Device\Harddisk2\DP(1)0-0+8

Driver loaded Version: 2.1

Formatting Removable Disk (F:)
Capacity: 124 MB
File system: FAT32
Allocation unit size: Default allocation size
Volume label:
Format options:
 Quick Format
 Enable Compression
 Create an MS-DOS startup disk

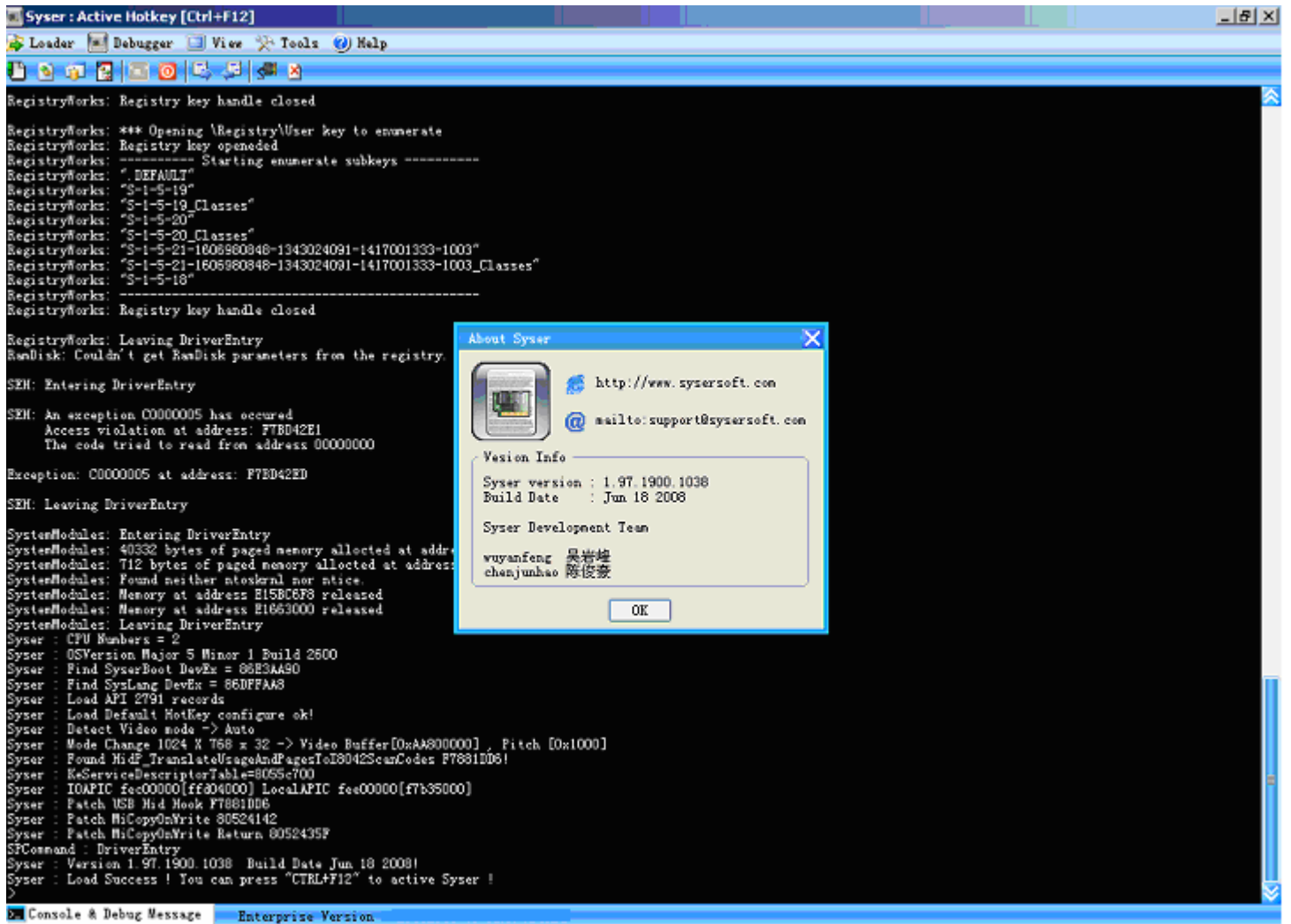
Formatting Removable Disk (F:)
Windows was unable to complete the format.
OK

New message!
process explorer.exe [1700] write \Device\Harddisk2\DP(1)0-0+8

Anti-Hook 01:08

Live Kernel Mode Debugging for xAurora Device Drivers (KMD(s)) (PoC)

Start Kernel Mode Debugger (Syser Debugger)



Load Export Symbol Modules to Debugger

The screenshot displays the Syser debugger interface. The main window is titled "Syser : Active Hotkey [Ctrl+F12]" and contains a console window with the following logs:

```
RegistryWorks: Registry key handle closed
RegistryWorks: *** Opening \Registry\User key to enumerate
RegistryWorks: Registry key opened
RegistryWorks: ----- Starting enumerate subkeys -----
RegistryWorks: "DEFAULT"
RegistryWorks: "S-1-5-19"
RegistryWorks: "S-1-5-19 Classes"
RegistryWorks: "S-1-5-20"
RegistryWorks: "S-1-5-20 Classes"
RegistryWorks: "S-1-5-21-1606980848-1"
RegistryWorks: "S-1-5-21-1606980848-1"
RegistryWorks: "S-1-5-18"
RegistryWorks: -----
RegistryWorks: Registry key handle cl

RegistryWorks: Leaving DriverEntry
RamDisk: Couldn't get RamDisk paramet

SEH: Entering DriverEntry

SEH: An exception C0000005 has occur
Access violation at address: F7B
The code tried to read from addr

Exception: C0000005 at address: F7BD4

SEH: Leaving DriverEntry

SystemModules: Entering DriverEntry
SystemModules: 40332 bytes of paged m
SystemModules: 712 bytes of paged mem
SystemModules: Found neither ntoskrnl
SystemModules: Memory at address E15B
SystemModules: Memory at address E166
SystemModules: Leaving DriverEntry
Syser : CPU Numbers = 2
Syser : OSVersion Major 5 Minor 1 Buil
Syser : Find SyserBoot DevEx = 86E3AA
Syser : Find SysLang DevEx = 86DFFAA8
Syser : Load API 2791 records
Syser : Load Default HotKey configura
Syser : Detect Video mode -> Auto
Syser : Mode Change 1024 X 768 x 32 -> Video Buffer[0xAA800000] , Pitch [0x1000]
Syser : Found HidF TranslateUsageAndPagesToI8042ScanCodes F7881DD6!
Syser : KeServiceDescriptorTable=8055c700
Syser : IOAPIC fec00000[fff04000] LocalAPIC fee00000[ffb35000]
Syser : Patch USB Hid Hook F7881DD6
Syser : Patch MiCopyOnWrite 80524142
Syser : Patch MiCopyOnWrite Return 8052435F
SFCommand : DriverEntry
Syser : Version 1.97.1900.1038 Build Date Jun 18 2008!
Syser : Load Success ! You can press "CTRL+F12" to active Syser !
>
```

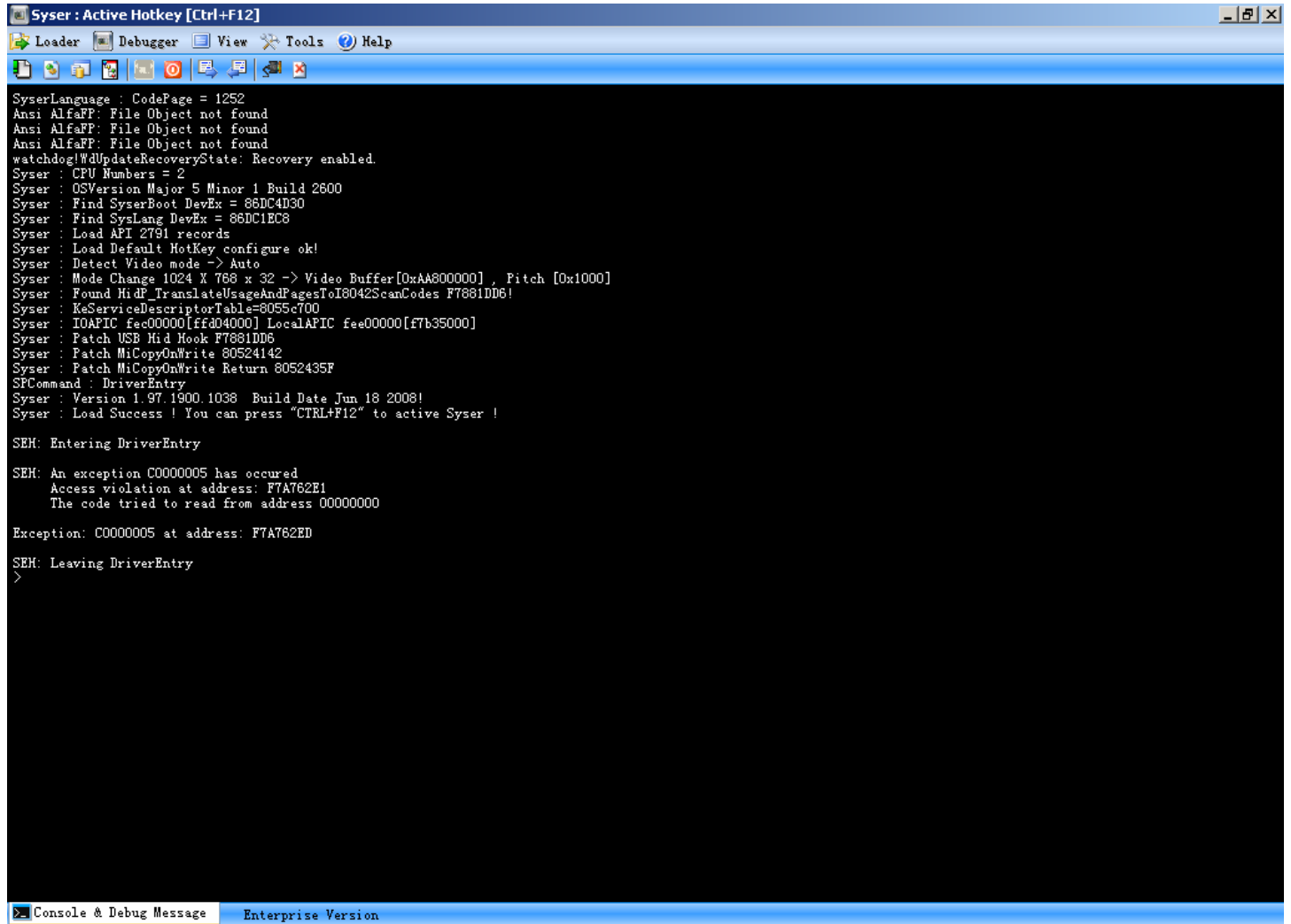
The "Syser Option" dialog box is open, showing a list of modules to be loaded after syser initialization:

- \SystemRoot\system32\Kernel132.dll
- \SystemRoot\system32\GDI32.dll
- \SystemRoot\system32\User32.dll
- \SystemRoot\system32\AdvApi32.dll
- \SystemRoot\system32\ws2_32.dll
- \SystemRoot\system32\msvrt.dll
- \SystemRoot\system32\comdlg32.dll
- \SystemRoot\system32\comctl32.dll
- \SystemRoot\system32\ImageHlp.dll
- \SystemRoot\system32\win32k.sys
- \SystemRoot\system32\ntoskrnl.exe
- \SystemRoot\system32\ntkrnlpa.exe
- \SystemRoot\system32\ntkrnlmp.exe
- \SystemRoot\system32\ntkrnpmp.exe
- \SystemRoot\system32\ntdll.dll
- \SystemRoot\system32\hal.dll

The dialog box includes buttons for "Add", "Remove", "Reload", "Default", and "Save".

At the bottom of the Syser window, there is a status bar with "Console & Debug Message" and "Enterprise Version".

xAurora Device Drivers Loading is in Progress



```
Syser : Active Hotkey [Ctrl+F12]
Loader  Debugger  View  Tools  Help
SyserLanguage : CodePage = 1252
Ansi AlfaFP: File Object not found
Ansi AlfaFP: File Object not found
Ansi AlfaFP: File Object not found
watchdog!WdUpdateRecoveryState: Recovery enabled.
Syser : CPU Numbers = 2
Syser : OSVersion Major 5 Minor 1 Build 2600
Syser : Find SyserBoot DevEx = 86DC4D30
Syser : Find SysLang DevEx = 86DC1EC8
Syser : Load API 2791 records
Syser : Load Default HotKey configure ok!
Syser : Detect Video mode -> Auto
Syser : Mode Change 1024 X 768 x 32 -> Video Buffer [0xAA800000] , Pitch [0x1000]
Syser : Found HidF_TranslateUsageAndPagesToI8042ScanCodes F7881DD6!
Syser : KeServiceDescriptorTable=8055c700
Syser : IOAPIC fec00000[fffd04000] LocalAPIC fee00000[f7b35000]
Syser : Patch USB Hid Hook F7881DD6
Syser : Patch MiCopyOnWrite 80524142
Syser : Patch MiCopyOnWrite Return 8052435F
SFCCommand : DriverEntry
Syser : Version 1.97.1900.1038 Build Date Jun 18 2008!
Syser : Load Success ! You can press "CTRL+F12" to active Syser !

SEH: Entering DriverEntry

SEH: An exception C0000005 has occurred
Access violation at address: F7A762E1
The code tried to read from address 00000000

Exception: C0000005 at address: F7A762ED

SEH: Leaving DriverEntry
>
```

Console & Debug Message Enterprise Version

xAurora Device Drivers Loaded and Driver is Activated

The screenshot displays the Syser debugger interface with a console window showing the following output:

```
Syser : Active Hotkey [Ctrl+F12]
Loader  Debugger  View  Tools  Help
SyserLanguage : CodePage = 1252
Ansi AlfaFP: File Object not found
Ansi AlfaFP: File Object not found
Ansi AlfaFP: File Object not found
watchdog!WdUpdateRecoveryState: Recovery enabled.
Syser : CPU Numbers = 2
Syser : OSVersion Major 5 Minor 1 Build 2600
Syser : Find SyserBoot DevEx = 86DC4D30
Syser : Find SysLang DevEx = 86DC1EC8
Syser : Load API 2791 records
Syser : Load Default HotKey configure ok!
Syser : Detect Video mode -> Auto
Syser : Mode Change 1024 X 768 x 32 -> Video Buffer [0xAA800000] , Pitch [0x1000]
Syser : Found HidF_TranslateUsageAndPagesToI8042ScanCodes F7881DD6!
Syser : KeServiceDescriptorTable=8055c700
Syser : IOAPIC fec00000[fffd04000] LocalAPIC fee00000 [f7b35000]
Syser : Patch USB Hid Hook F7881DD6
Syser : Patch MiCopyOnWrite 80524142
Syser : Patch MiCopyOnWrite Return 8052435F
SPCommand : DriverEntry
Syser : Version 1.97.1900.1038 Build Date J
Syser : Load Success ! You can press "CTRL+F12"

SEH: Entering DriverEntry
SEH: An exception C0000005 has occurred
Access violation at address: F7A782E1
The code tried to read from address 00000000
Exception: C0000005 at address: F7A782ED
SEH: Leaving DriverEntry
>scrshot
Syser : Save ScreenShot to \??C:\Program Files\Syser\
>scrshot
Syser : Save ScreenShot to \??C:\Program Files\Syser\Syser.bmp
SEH: Entering DriverEntry
SEH: An exception C0000005 has occurred
Access violation at address: F7A9B2E1
The code tried to read from address 00000000
Exception: C0000005 at address: F7A9B2ED
SEH: Leaving DriverEntry
Driver loaded
>|
```

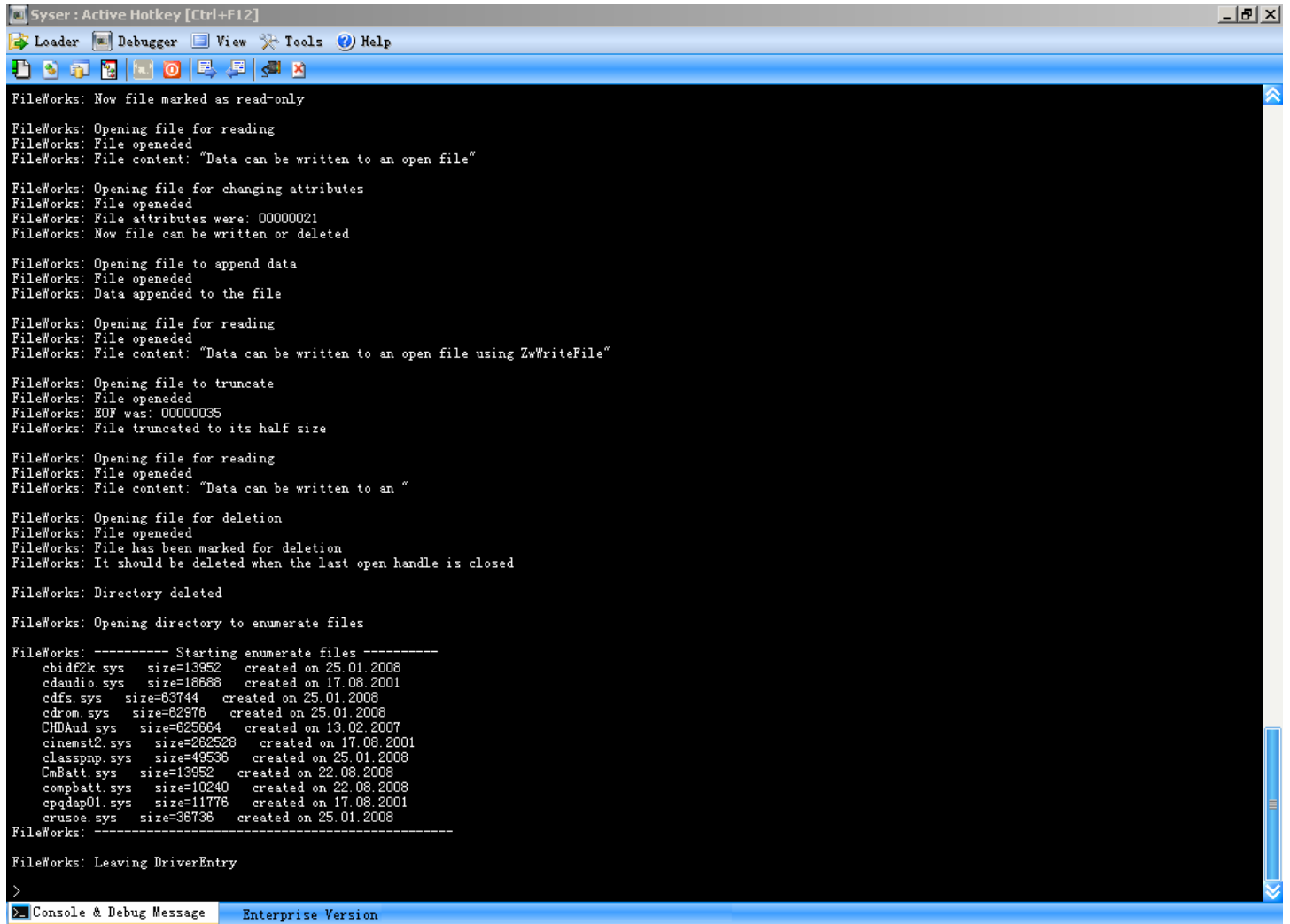
Overlaid on the console is a "Quick NT Driver Loader" dialog box with the following fields and buttons:

- Driver File Name: C:\Documents and Settings\ymafia\Deskt [Browse]
- Service Name: xAuroraEngine
- Buttons: Install, Uninstall, Start, Stop

Below the driver loader dialog is a "Drvloader" dialog box with a warning icon and the text "Driver is activated" and an "OK" button.

At the bottom of the Syser window, the status bar shows "Console & Debug Message" and "Enterprise Version".

xAurora Device Drivers Running Status in Kernel Mode



The screenshot shows a debugger window titled "Syser : Active Hotkey [Ctrl+F12]". The window has a menu bar with "Loader", "Debugger", "View", "Tools", and "Help". Below the menu bar is a toolbar with various icons. The main area is a black console displaying white text logs from the FileWorks driver. The logs show various file operations such as opening, reading, writing, truncating, and deleting files and directories. A section titled "Starting enumerate files" lists several system files with their sizes and creation dates. The window has a status bar at the bottom with "Console & Debug Message" and "Enterprise Version".

```
Syser : Active Hotkey [Ctrl+F12]
FileWorks: Now file marked as read-only
FileWorks: Opening file for reading
FileWorks: File opened
FileWorks: File content: "Data can be written to an open file"
FileWorks: Opening file for changing attributes
FileWorks: File opened
FileWorks: File attributes were: 00000021
FileWorks: Now file can be written or deleted
FileWorks: Opening file to append data
FileWorks: File opened
FileWorks: Data appended to the file
FileWorks: Opening file for reading
FileWorks: File opened
FileWorks: File content: "Data can be written to an open file using ZwWriteFile"
FileWorks: Opening file to truncate
FileWorks: File opened
FileWorks: EOF was: 00000035
FileWorks: File truncated to its half size
FileWorks: Opening file for reading
FileWorks: File opened
FileWorks: File content: "Data can be written to an "
FileWorks: Opening file for deletion
FileWorks: File opened
FileWorks: File has been marked for deletion
FileWorks: It should be deleted when the last open handle is closed
FileWorks: Directory deleted
FileWorks: Opening directory to enumerate files
FileWorks: ----- Starting enumerate files -----
cbidf2k.sys size=13952 created on 25.01.2008
cdaudio.sys size=18688 created on 17.08.2001
cdfx.sys size=63744 created on 25.01.2008
cdrom.sys size=62976 created on 25.01.2008
CHDAud.sys size=625664 created on 13.02.2007
cinemst2.sys size=262528 created on 17.08.2001
classpnp.sys size=49536 created on 25.01.2008
CmBatt.sys size=13952 created on 22.08.2008
compbatt.sys size=10240 created on 22.08.2008
cpqdap01.sys size=11776 created on 17.08.2001
crusee.sys size=38736 created on 25.01.2008
FileWorks: -----
FileWorks: Leaving DriverEntry
>
Console & Debug Message Enterprise Version
```

xAurora Hooks and Memory Page Debugging in Kernel Mode

```
0 1 2 3 4 5 6 7 - 8 9 A B C D E F
0x80042068 0400 0 0 0 1 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 0
0x8004206A 0018 1 0 0 0 0 0 0 0 0 - 0 0 0 1 1 0 0 0 0 1
0x8004206C 1800 2 0 0 0 1 1 0 0 0 - 0 0 0 0 0 0 0 0 0 2
0x8004206E 0000 3 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 3
0x80042070 0000 4 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 4
0x80042072 0000 5 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 5
0x80042074 0000 6 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 6
0x80042076 0000 7 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 7
0x80042078 0000 8 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 8
0x8004207A 0000 9 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 9
0x8004207C 0000 A 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 A
0x8004207E 0000 B 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 B
0x80042080 0000 C 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 C
0x80042082 0000 D 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 D
0x80042084 0000 E 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 E
0x80042086 0000 F 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 F
0 1 2 3 4 5 6 7 - 8 9 A B C D E F
>tss
TR=0028 BASE=80042000 LIMIT=20AB
LDT=0000 GS=3B41 FS=3990 DS=3308 SS=0000 CS=E281 ES=1114 CR3=003AE000
EAX=0310E8C1 EBX=F7104D39 ECX=C25D0845 EDX=B60F0C55 EIP=E1750855
ESI=8B1F76D0 EDI=B60F0C55 EBP=F7104D39 ESP=C9330845 EFL=08458B5E
SS0=0010:805514F0 SS1=4008:758B0855 SS2=00FF:FFB68110
I/O Map Base=20AC I/O Map Size=0
Interrupt redirection bit map. Base=0x80042068 Size=32
0 1 2 3 4 5 6 7 - 8 9 A B C D E F
0x80042068 0400 0 0 0 1 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 0
0x8004206A 0018 1 0 0 0 0 0 0 0 0 - 0 0 0 1 1 0 0 0 0 1
0x8004206C 1800 2 0 0 0 1 1 0 0 0 - 0 0 0 0 0 0 0 0 0 2
0x8004206E 0000 3 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 3
0x80042070 0000 4 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 4
0x80042072 0000 5 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 5
0x80042074 0000 6 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 6
0x80042076 0000 7 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 7
0x80042078 0000 8 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 8
0x8004207A 0000 9 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 9
0x8004207C 0000 A 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 A
0x8004207E 0000 B 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 B
0x80042080 0000 C 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 C
0x80042082 0000 D 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 D
0x80042084 0000 E 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 E
0x80042086 0000 F 0 0 0 0 0 0 0 0 - 0 0 0 0 0 0 0 0 0 F
0 1 2 3 4 5 6 7 - 8 9 A B C D E F
>
```

System Explorer Ctrl+1 Command Console Ctrl+2 Source Explorer Ctrl+3 System : ntkrnlpa!ntkrnlpa_KiDispatchInterrupt+0x3

xAurora Live Debugging is in Progress

The screenshot displays the xAurora Live Debugging interface with the following components:

- Menu Bar:** Debug, Monitor, View, Information, Tools, Help. Version: 1.97.1900.10.
- Register Window (Left):** Shows CPU 0 and CPU 1 registers. EFLAG is 00000202, CS is 0008, DS is 0023, ES is 0023, FS is 0030, GS is 0000, SS is 0010. GDTR, IDTR, LDTR, and TR are also visible.
- Memory Dump (Top Right):** A hex dump showing memory addresses from 00000000 to 00000050. The data is mostly unknown (??).
- Instruction List (Middle Right):**

Address	Hex	Disassembly
80545D40	90	NOP
80545D41	FA	CLI
80545D42	3B6D00	CMP EBP, [EBP]
80545D45	740D	JZ 80545D54
80545D47	B102	MOV CL, 02
80545D49	FF15B4804D80	CALL hal_HalClearSoftwareInterrupt
80545D4F	E8BA000000	CALL 80545E0E
80545D54	83BB2801000000	CMP [EBX+00000128], +00
80545D5B	74D7	JZ 80545D34
80545D5D	B91C000000	MOV ECX, 0000001C
80545D62	FF152C804D80	CALL hal_KfRaiseIrql
80545D68	FB	STI
80545D69	8D8B40050000	LEA ECX, [EBX+00000540]
80545D6F	E80CADFFFF	CALL 80540A80
80545D74	8BB328010000	MOV ESI, [EBX+00000128]
80545D7A	8BBB24010000	MOV EDI, [EBX+00000124]
80545D80	807E5000	CMP [ESI+50], 00
80545D84	753D	JNZ 80545DC3
80545D86	3BF7	CMP ESI, EDI
80545D88	745A	JZ 80545DE4
80545D8A	83C901	OR ECX, +01
80545D8D	89B324010000	MOV [EBX+00000124], ESI
- Command Console (Bottom Right):**

```

ws focus stack watch list (HotKey ALT+S)
ww focus watch list (HotKey ALT+S)
wx focus XMM register list (HotKey ALT+X)
x Return to host and continue running
zap Replace an embedded INT 1 or INT 3 with a NOP instruction.
>
    
```
- Taskbar (Bottom):** System Explorer Ctrl+1, Command Console Ctrl+2, Source Explorer Ctrl+3, System : ntkrnlpa!ntkrnlpa_KiDispatchInterrupt+0x3

Loading IDT (Import Data Table) in to xAurora Drivers

```
Wisp Syser Console
Syser : Find UHCI Ctrl 0 [Bus 00 Device 1D Function 00 USBBase 3020 FrameBase 0000E000]
Syser : Find EHCI Ctrl
Syser : Load module 937 export symbols \\??\C:\WINDOWS\system32\Kernel32.dll
Syser : Load module 596 export symbols \\??\C:\WINDOWS\system32\GDI32.dll
Syser : Load module 720 export symbols \\??\C:\WINDOWS\system32\User32.dll
Syser : Load module 648 export symbols \\??\C:\WINDOWS\system32\AdvApi32.dll
Syser : Load module 114 export symbols \\??\C:\WINDOWS\system32\ws2_32.dll
Syser : Load module 815 export symbols \\??\C:\WINDOWS\system32\msvcrt.dll
Syser : Load module 28 export symbols \\??\C:\WINDOWS\system32\comdlg32.dll
Syser : Load module 169 export symbols \\??\C:\WINDOWS\system32\comctl32.dll
Syser : Load module 109 export symbols \\??\C:\WINDOWS\system32\ImageHlp.dll
Syser : Load module 221 export symbols \\??\C:\WINDOWS\system32\win32k.sys
Syser : Load module 1464 export symbols \\??\C:\WINDOWS\system32\ntoskrnl.exe
Syser : Load module 1464 export symbols \\??\C:\WINDOWS\system32\ntkrnlpa.exe
Syser : Load module 1016 export symbols \\??\C:\WINDOWS\system32\ntdll.dll
Syser : Load module 92 export symbols \\??\C:\WINDOWS\system32\hal.dll
Syser : SystemRoot = \\??\C:\WINDOWS\
Syser : Win32 Service Table = 0x859D0004
Syser : SyserBoot Device Found ! DDraw hook is available !
Syser : SyserLanguage Device Found ! Safe Unicode Function is available !
Syser : Register Plugin Module SPCCommand (Syser Command Plugin Module)
Syser Debugger Enterprise Version : License to TEAM ArCADE
>scrshot
Error : unknown command !
>
```

System Explorer Ctrl+1 Command Console Ctrl+2 Source Explorer Ctrl+3 System : ntkrnlpa!ntkrnlpa_KiDispatchInterrupt+0x3

xAurora Core Engine Started in Kernel Mode (Ring0)

```
A9FE8000 00003000 8696E558 868856F0 A9FE9266 00000000 A9FE8FA6 s24trans
AA337000 00027C00 86B3B3B8 86AF2DB8 AA35AF85 00000000 AA359D6F NetBT
F7547000 0000F600 86DBF970 86DBEDD8 F75547F2 F74C8487 F74CE4B4 Cdrom
F7081000 00003C80 86C6B228 86CD77F0 F7083BE6 00000000 F7081528 mssmbios
F71CA000 000D4400 86E43920 86DC1FE0 F71CA5D2 00000000 F71CA300 SyserLanguage
F7A4F000 00000D00 86E43F98 86DFD200 F7A4F61E 00000000 F770B6DC PCIIde
AA638000 000A3000 86C61218 86C71DD8 AA6D6000 00000000 AA6D04F6 HdAudAddService
F7329000 0001D580 86E43EB8 86D68A10 F7343C92 00000000 F733EF42 Pcmcia
F7637000 00008700 86B8C550 86AE9B90 F763DFD6 00000000 F763DCC4 Wanarp
AA387000 00058480 86AF4BC0 86AF70D8 AA3D7D23 00000000 AA3B5A58 Tcpip
F79B1000 00001080 86AEF690 86AEC210 F79B1646 00000000 F6AB2DF0 mmmdd
F7ADD000 00000700 86804F08 8633BEC0 F7ADD442 00000000 F7ADD3F4 xAuroraEngine
AA4C6000 000F1680 86C3F2A0 86B8A3B0 AA5AE7B8 00000000 AA5ADC80 HSF_DPV
F74A7000 0000CC80 86E43AD8 86D61600 F74B0D3E 00000000 00000000 VolSnap
F7B9D000 00000D00 8623B4F8 8633BFB8 F7B9D437 00000000 F7B9D33C SPCCommand
F7517000 00008E00 86D65360 86DB9698 F751C985 00000000 F751A112 intelppm
F7877000 00004B60 868EC320 86881818 F787AB56 00000000 F78772E6 AegisP
F7537000 0000A480 86D76A00 86DBFFE0 F753FAB8 00000000 F753F0CC ImapI
F7BAD000 00000B80 86AF5218 86AF1270 F7BAD59A 00000000 F7BAD438 Null
F7757000 00007600 86D78160 86D6F3B0 F7765B05 00000000 F6A6A856 usbehci
AA3E0000 00012600 86B8F3D0 86AF5360 AA3F0A85 00000000 AA3E3CF2 IPSec
F74B7000 00008E00 86E438B8 86DB0FE0 F74BE8AD 00000000 F74CE4B4 Disk
F7347000 00010A80 86E56110 86E250D8 F7355004 00000000 F734FBAA PCI
F796B000 00002780 86C7A368 86C79450 F796CD58 00000000 F796B588 NdisTapi
F69ED000 00016580 86C79478 86DFEB10 F6A01223 00000000 F70D83BF NdisWan
F770F000 00004D00 86E43BB0 86DC44C8 F7712C3B 00000000 F770F830 PartMgr
F7597000 00008900 86DBD2B0 86DC04A0 F759EA85 00000000 F7599234 Gpc
F6A81000 00025000 86DBC098 86DFE9C8 F6AA1000 00000000 F6A8B190 HDAudBus
A9E50000 00003180 8696F590 867FFDB8 A9E52780 00000000 A9E504B4 mdmxdsk
F7358000 0002DD80 86E561E8 86DFF2A0 F7381059 00000000 F735E866 ACPI
F7999000 00001180 86DBE168 86CD5BD0 F7999A85 00000000 F799959A moh
00000000 00000000 00000000 86E06D80 8069B6CC 00000000 00000000 PnpManager
A9FF0000 00003900 8687DC08 8689CB90 A9FF2C56 00000000 A9FF272E Ndisuio
AA2EF000 00021D00 86AE6658 86984518 AA30CF40 00000000 AA2F64A0 AFD
F6A27000 00035DE0 86C82228 86CD33B0 F6A5A020 F6A32820 F6A32C20 SynIP
AA614000 00002880 86AF0C48 86AF5918 AA615F05 00000000 F769D548 HidUsb
F7527000 0000CD00 86DB3160 86C993B0 F7530285 F7527910 F752DEB6 i8042prt
F7943000 00003C00 86B8C300 86C754D0 F7945F05 00000000 00000000 fsh
F795B000 00003680 86DB0468 86D5E790 F795D966 00000000 F795CA86 CmBatt
F70F4000 0008C600 86E43628 86DB8800 F7179384 00000000 00000000 Ntfs
F7647000 00008780 86984280 86983AF8 F764E529 00000000 F764871A NetBIOS
F7198000 00011F00 86E43770 86D8C328 F71A7FD4 00000000 00000000 sr
AA28C000 0002AE80 8697F2B0 86AED450 AA2B3388 00000000 00000000 Rdbss
F7847000 00004A80 86AEE170 86B8C238 F784ABED 00000000 F784A1DE Msfs
AA21C000 0006F780 8697E2A8 86983360 AA284503 00000000 00000000 MRxSmb
```

System Explorer Ctrl+1 Command Console Ctrl+2 Source Explorer Ctrl+3 System : ntkrnlpa!ntkrnlpa_KiDispatchInterrupt+0x3

Entire xAurora Device Drivers Loaded Successfully

```
ACPIEC          F78A3000 00003000 00004419 3B7D8553 \\?\C:\WINDOWS\system32\drivers\ACPIEC.sys
SDBGMsg         F78A7000 00003000 000037EC 48591576 \\?\C:\WINDOWS\system32\drivers\SDBGMsg.sys
fsh             F7943000 00004000 0000A2B6 48B63D38 \\?\C:\WINDOWS\System32\Drivers\fsh.SYS
mhk            F7947000 00003000 000065C0 48A92E15 \\?\C:\WINDOWS\System32\Drivers\mhk.SYS
rasacd         F794F000 00003000 0000B2E7 3B7D84CB \\?\C:\WINDOWS\system32\DRIVERS\rasacd.sys
CmBatt         F795B000 00004000 00005C5C 479A2328 \\?\C:\WINDOWS\system32\DRIVERS\CmBatt.sys
wmiacpi        F7963000 00003000 00009136 479A2329 \\?\C:\WINDOWS\system32\DRIVERS\wmiacpi.sys
ndistapi       F796B000 00003000 00003160 479A274B \\?\C:\WINDOWS\system32\DRIVERS\ndistapi.sys
TSKNF800       F7973000 00003000 0000DA17 48B10422 \\?\C:\WINDOWS\system32\Drivers\TSKNF800.SYS
KDCOM          F7987000 00002000 00008311 3B7D8346 \\?\C:\WINDOWS\system32\KDCOM.DLL
WMILIB         F7989000 00002000 0000D600 3B7D878B \\?\C:\WINDOWS\system32\DRIVERS\WMILIB.SYS
dmload         F798B000 00002000 0000DC8A 3B7D8567 \\?\C:\WINDOWS\system32\drivers\dmload.sys
USBd           F7997000 00002000 000040AF 3B7D8682 \\?\C:\WINDOWS\system32\DRIVERS\USBd.SYS
moh            F7999000 00002000 0000A9BB 487F0B3C \\?\C:\WINDOWS\System32\Drivers\moh.SYS
swenum        F799F000 00002000 000095A5 479A2357 \\?\C:\WINDOWS\system32\DRIVERS\swenum.sys
Fs_Rec         F79A9000 00002000 000079A7 3B7D8361 \\?\C:\WINDOWS\System32\Drivers\Fs_Rec.SYS
Beep           F79AD000 00002000 0000C82C 3B7D82E5 \\?\C:\WINDOWS\System32\Drivers\Beep.SYS
mmrdd          F79B1000 00002000 0000F3F7 3B7D8538 \\?\C:\WINDOWS\System32\Drivers\mmrdd.SYS
RDPCCD         F79B5000 00002000 0000E2B7 3B7D82C0 \\?\C:\WINDOWS\System32\DRIVERS\RDPCCD.sys
dump_WMILIB    F79BD000 00002000 0000D600 3B7D878B \\?\C:\WINDOWS\System32\Drivers\dump_WMILIB.SYS
xAuroraSANDBOX F79C3000 00002000 00011202 4107E93F \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraRegistryMGR F79C7000 00002000 000073EF 4107E981 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraFileProtection F79DD000 00002000 00003474 4107E94A \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
000            F7A13000 00002000 0000E6DA 4459653C \\?\C:\Program Files\CyberLink\Power2Go\000.fcl
pciide         F7A4F000 00001000 0000213E 3B7D83B5 \\?\C:\WINDOWS\system32\drivers\pciide.sys
OPRGHDLR       F7A50000 00001000 0000B001 3B7D8553 \\?\C:\WINDOWS\system32\DRIVERS\OPRGHDLR.SYS
xAuroraSEH     F7A6D000 00001000 00001A2E 4107E961 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraSEH     F7A76000 00001000 00001A2E 4107E961 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraSEH     F7A9B000 00001000 00001A2E 4107E961 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
mchInjDrv      F7AB7000 00001000 000017A5 43C4AEC8 \\?\C:\WINDOWS\system32\Drivers\mchInjDrv.sys
xAuroraEngine  F7ADD000 00001000 00004865 42467444 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraProcess F7AF9000 00001000 000089F6 4107E969 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
dxgthk         F7B02000 00001000 000035E7 3B7D8438 \\?\C:\WINDOWS\System32\drivers\dxgthk.sys
xAuroraMM      F7B23000 00001000 000036BF 4107E95E \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraSDT     F7B3D000 00001000 00005E8F 4107E936 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
audstub        F7B4E000 00001000 000105B1 3B7D85BC \\?\C:\WINDOWS\system32\DRIVERS\audstub.sys
xAuroraIDT     F7B5D000 00001000 000087F5 4107E9EE \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraGDT     F7B75000 00001000 00005FC8 4107E9E5 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraKernelBase F7B85000 00001000 00001E02 4107E9AA \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
SPCommand      F7B9D000 00001000 000055BC 485915AE \\?\C:\WINDOWS\system32\drivers\Plugin\i386\SPCommand.sys
Null           F7BAD000 00001000 00008483 3B7D82EB \\?\C:\WINDOWS\System32\Drivers\Null.SYS
xAuroraSystemModules F7BB2000 00001000 00008F2A 4107E97D \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
xAuroraSharedMemory F7BD1000 00001000 00006AE1 4107E975 \\?\C:\Documents and Settings\mafia\Desktop\xAurora-AI-Kernel-Dr
\
```


END NOTE

Invisible KERNEL hooks will hide all the possible Kernel and Marshall (NTDLL) Hooks from xAurora and vice versa. That's why you may never been able to see any KMD(s) running in the browser, even when you loaded browser in to any debugger. Also, I have included many Anti-Debugging techniques, Anti-Debug Attacking Methods and many other ultra secured Anti-Reverse Engineering methodologies to achieve the maximum protection for the xAurora. And that information never gave out to any personal till now. All the software anti-reverse engineering tricks coded using Microsoft Macro Assembler 8.2x. And also I have used most of custom made API(s) and NT Undocumented API(s) for better protection of the browser. Above all the examples has been successfully redesigned, recoded (rewritten) and included into xAurora.

SPECIAL NOTE

I am sending the unwrapped (Protection Removed) xAurora Main Executable file to Mr. Gotaimbara to do the Live Demo and PoC of KMD Availability in the browser.

CONCLUSION

Mr. Anonymous Skywalker, Mr. Harshadeva Ariyasinghe and Mr. Kalinga Athulathmudali's reverse engineering methodologies are very lamer and they have very basic NT Kernel/Low Level knowledge. Also they may never work with UNDOCUMENTED NT API(s) in NT Subsystem. xAurora is based on Windows Kernel Mode Drivers. This guideline document will help you to understand why Mr. Anonymous Skywalker, Mr. Harshadeva Ariyasinghe and Mr. Kalinga Athulathmudali xAurora Kernel Mode PoC failed and they couldn't prove it. I know my own code better than all of you. Because, I am the founder of xAurora's concepts and I am the programmer of the xAurora Web Browser. No one can admit the wrong conclusion without proving it in the real world and to the community. Because, xAurora is a COPYRIGHTED, TRADEMARKED and PATENTED SOFTWARE.

xAurora is a great Sri Lankan product that entirely coded in Win32 Assembly Language. Hope you may be able to understand it. In future I will show you many stories behind the xAurora case. Hope Mr. Gotaimbara will help me to sort out the matters soon. Thank you very much for the great support and your support in the future is greatly appreciated.

Kind Regards

Dr. Sameera de Alwis

Founder – Team xAurora 2009