

Reverse Engineering Windows AFD.sys

Steven Vittitoe

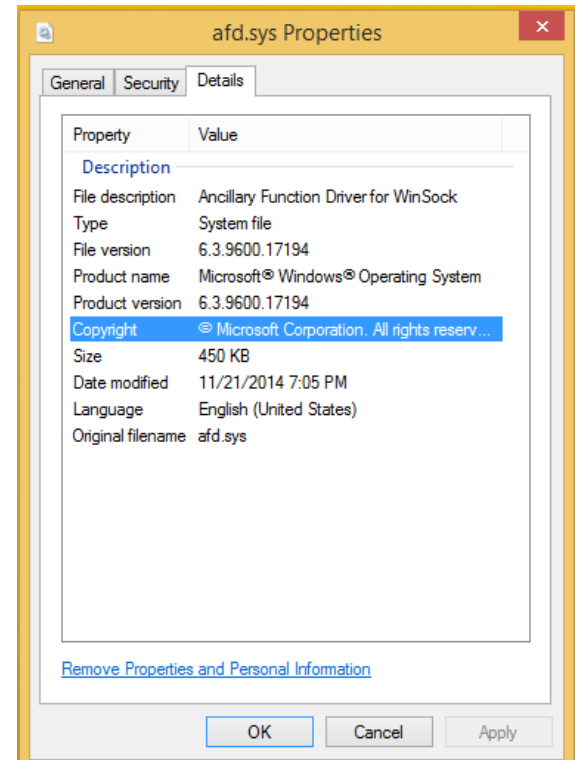

@bool101
bool@google.com

Outline

- Why AFD.sys
- Winsock overview
- Interesting findings
- Input to AFD.sys
- Analysis
- Fuzzing
- Future

What is AFD.sys?

- Default kernel module
- Ancillary Function Driver
- Ring 0 entrypoint for Winsock
- Required for socket() calls
- Not all network comms use it:
 - winhttp wininet
 - webdav mrxsmb



Why AFD.sys?

- Sandbox accessibility

- Chrome

YES

- Adobe Reader

YES

- IE EPM

YES

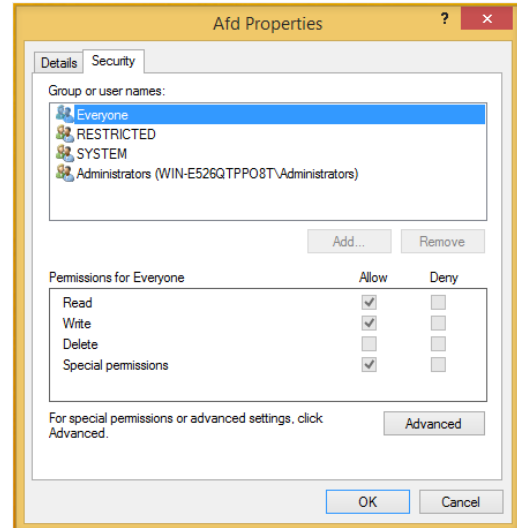
- History of bugs:

- CVE-2011-2005

CVE-2012-0148

- CVE-2013-3887

CVE-2014-1767



Goals

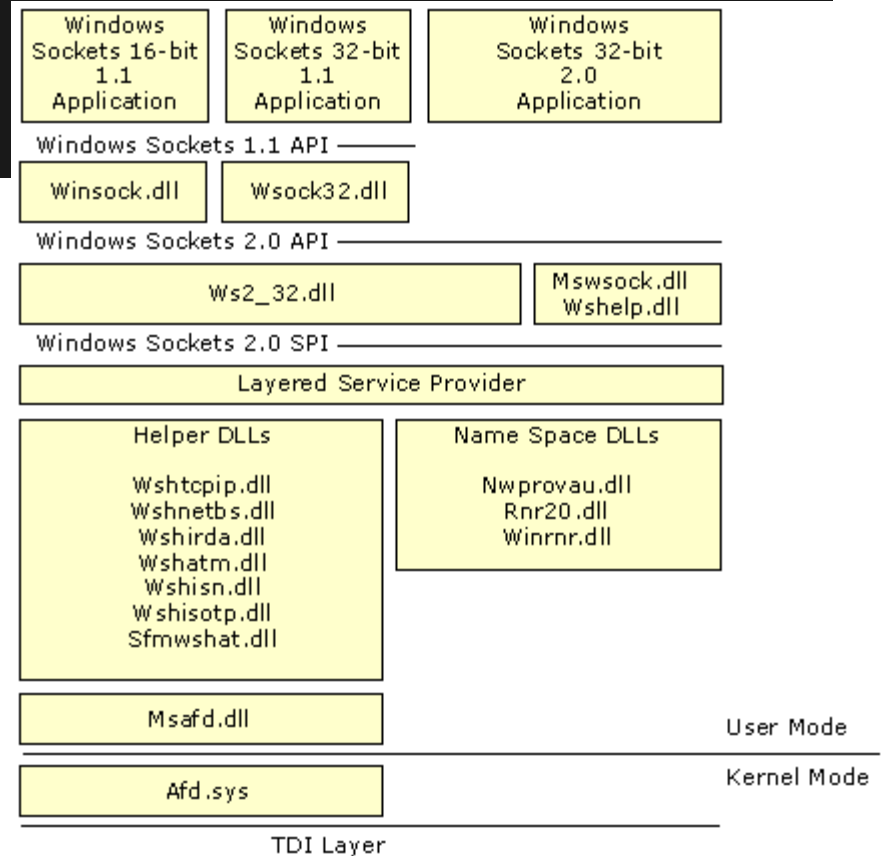
- Project Zero's goal:
 - “Make 0-days hard(er)”
- Strengthen Sandboxes
 - Widely adopted strategy
 - Increase attacker cost
 - Ways to escape:
 - Logic errors (broker process)
 - Bugs in syscalls / win32k.sys
 - Bugs in accessible devices!

Why AFD.sys?

- Cannot be disabled until Windows 8
 - Even then not easy to disable
- Complexity and accessibility
 - AFD.sys size ~500KB
 - win32k.sys is 3.1MB
 - most kernel drivers < 100KB
 - 70 IOCTL's reachable from \\Device\Afd\Endpoint
 - Handles everything from TCP/IP to SAN

Winsock

- socket(AF_INET) call
 1. ws2_32 (2 fn)
 2. mswsock (4 fn)
 3. wshtcpip (1 fn)
 4. mswsock (IOCTL)



AFD is a translator

- AFD acts as a server to user mode Winsock
 - Abstracts multiple protocols
 - Ends up relaying to:
 - Transport Driver Interface (TDI)
 - Winsock Kernel (WSK)
- Serves kernel mode clients as a WSK provider (internal IOCTL)

First Glance

- DbgPrint
 - Normally removed in release builds?
 - 23 xrefs in Win7
 - 113 xrefs in Win8
- 74/279 import DbgPrint* (~25%)
 - Event Tracing for Windows (ETW) extensively used
 - Helpful in RE efforts



Registry

- Several configurations pulled from registry:
 - HKLM\System\CCS\Services\Afd
 - Buffer sizes
 - DisableRawSecurity - admin raw sockets
 - DefaultSendWindow
 - AfdReadRegistry() populates `_AfdConfigInfo`
- Few are “Volatile” configurations
 - Change notification registered

Inputs

```
nmemset32(DriverObject->MajorFunction, (int)AfdDispatch, 0x1Cu);
DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL] = (PDRIVER_DISPATCH)AfdDispatchDeviceControl;
DriverObject->MajorFunction[IRP_MJ_INTERNAL_DEVICE_CONTROL] = (PDRIVER_DISPATCH)AfdWskDispatchInternalDeviceControl;
DriverObject->MajorFunction[IRP_MJ_SYSTEM_CONTROL] = (PDRIVER_DISPATCH)AfdEtwDispatch;
DriverObject->FastIoDispatch = (PFAST_IO_DISPATCH)&AfdFastIoDispatch;
DriverObject->DriverUnload = (PDRIVER_UNLOAD)AfdUnload;
```

- IOCTLs
- Plug-n-Play Events
- TDI address changes and filtering
- RPC

IOCTLs

- Easy to find tables
 - AfdIrpCallDispatch - functions
 - AfdIoctlTable - numbers
- Another level of indirection
 - AfdImmediateCallDispatch
 - For routines that always
IoofCompleteRequest

```

rdata:00026730 ; _AfdIrpCallDispatch dd offset 0afBind08
rdata:00026730 ;          ; DATA XREF: AfdDispatchDeviceControl(x,x)+324r
rdata:00026730 ;          ; AfdBind(x,x)
rdata:00026734 ; dd offset 0afDConnect08 ; AfdConnect(x,x)
rdata:00026730 ; dd offset 0afDStarListen08 ; AfdStarListen(x,x)
rdata:0002673C ; dd offset 0afDWaitForListen08 ; AfdWaitForListen(x,x)
rdata:00026740 ; dd offset 0afDAccept08 ; AfdAccept(x,x)
rdata:00026744 ; dd offset 0afDReceive08 ; AfdReceive(x,x)
rdata:00026740 ; dd offset 0afDReceiveDatagram08 ; AfdReceiveDatagram(x,x)
rdata:0002674C ; dd offset 0afDSend08 ; AfdSend(x,x)
rdata:00026750 ; dd offset 0afDSendDatagram08 ; AfdSendDatagram(x,x)
rdata:00026754 ; dd offset 0afDPoll08 ; AfdPoll(x,x)
rdata:00026758 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002675C ; dd offset 0afDAddress08 ; AfdAddress(x,x)
rdata:00026760 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026764 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026768 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026770 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026774 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026778 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002677C ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026780 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026784 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026788 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026790 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026794 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026798 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002679C ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267A0 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267A4 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267A8 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267AC ; dd offset 0afDTransmitFile08 ; AfdTransmitFile(x,x)
rdata:000267B0 ; dd offset 0afDSuperAccept08 ; AfdSuperAccept(x,x)
rdata:000267B4 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267B8 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267BC ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267C0 ; dd offset 0afDWaitForListen08 ; AfdWaitForListen(x,x)
rdata:000267C4 ; dd offset 0afDSetQos08 ; AfdSetQos(x,x)
rdata:000267C8 ; dd offset 0afDGetQos08 ; AfdGetQos(x,x)
rdata:000267CC ; dd offset 0afDNoOperation08 ; AfdNoOperation(x,x)
rdata:000267D0 ; dd offset 0afDDataListChange08 ; AfdDataListChange(x,x)
rdata:000267D4 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267D8 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267DC ; dd offset 0afDRoutingInterfaceChange08 ; AfdRoutingInterfaceChange(x,x)
rdata:000267E0 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:000267E4 ; dd offset 0afDAddressListChange08 ; AfdAddressListChange(x,x)
rdata:000267E8 ; dd offset 0afDConnect08 ; AfdConnect(x,x)
rdata:000267EC ; dd offset 0afDListenControl08 ; AfdListenControl(x,x)
rdata:000267F0 ; dd offset 0afDTransmitPackets08 ; AfdTransmitPackets(x,x)
rdata:000267F4 ; dd offset 0afDSuperConnect08 ; AfdSuperConnect(x,x)
rdata:000267F8 ; dd offset 0afDSuperDisconnect08 ; AfdSuperDisconnect(x,x)
rdata:000267FC ; dd offset 0afDReceiveDatagram08 ; AfdReceiveDatagram(x,x)
rdata:00026800 ; dd offset 0afDSendMessageDispatch08 ; AfdSendMessageDispatch(x,x)
rdata:00026804 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026808 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026810 ; dd offset 0afDSanConnectHandler08 ; AfdSanConnectHandler(x,x)
rdata:00026814 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026818 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002681C ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026820 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026824 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026828 ; dd offset 0afDSanAcquireContext08 ; AfdSanAcquireContext(x,x)
rdata:0002682C ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026830 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026834 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026838 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002683C ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026840 ; dd offset 0afDSanAddrListChange08 ; AfdSanAddrListChange(x,x)
rdata:00026844 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:00026848 ; dd offset 0afDDispatchImmediateIrp08 ; AfdDispatchImmediateIrp(x,x)
rdata:0002684C ; dd offset 0afDRIo08 ; AfdRiIo(x,x)
rdata:00026850 ; align 10h
rdata:00026850 ; int AfdIoctlTable[]

```

Immediate Call Dispatch

```
_AfdIrpcCallDispatch`dd offset @AfdBind@8  
; DATA XREF: Afd  
; AfdBind(x, x)  
dd offset @AfdConnect@8 ; AfdConnect(x, x)  
dd offset @AfdStartListen@8 ; AfdStartLi  
dd offset @AfdWaitForListen@8 ; AfdWaitF  
dd offset @AfdAccept@8 ; AfdAccept(x, x)  
dd offset @AfdReceive@8 ; AfdReceive(x, x)  
dd offset @AfdReceiveDatagram@8 ; AfdRec  
dd offset @AfdSend@8 ; AfdSend(x, x)  
dd offset @AfdSendDatagram@8 ; AfdSendDa  
dd offset @AfdPoll@8 ; AfdPoll(x, x)  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdGetAddress@8 ; AfdGetAddre  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f  
dd offset @AfdDispatchImmediateIrp@8 ; f
```

=>

```
dd 0  
dd 0  
dd 0  
dd offset _AfdPartialDisconnect@32 ; AfdP  
dd 0  
dd offset _AfdQueryReceiveInformation@32  
dd offset _AfdQueryHandles@32 ; AfdQueryH  
dd offset _AfdSetInformation@32 ; AfdSetI  
dd offset _AfdGetRemoteAddress@32 ; AfdGe  
dd offset _AfdGetContext@32 ; AfdGetConte  
dd offset _AfdSetContext@32 ; AfdSetConte  
.. ..
```

Static Bug Hunting

- Windows 7 x86
- Basic bottom up static analysis
 - memcpy, memmove, ExAllocatePool*, etc
 - functions with `__security_check_cookie` xrefs
 - functions with large stack buffers
 - object reference counts
- Script to find unchecked return values
 - ExAllocatePool* (Note: TagPriority raises exception)

Static Bug Hunting

- Manual review of all reachable IOCTLs
 - Not WSK or SAN related IOCTLs
 - Data alignment
 - Proper size restrictions
 - TOCTOU on METHOD_NEITHER IOCTLs
 - Integer under/overflow issues
 - Signed integer issues

Fuzzing

- Preference for static / dynamic analysis
 - Better understanding of target
 - Leads to better fuzzers
- Two weeks fuzz time
 - Single core
 - Simple fuzzer
 - Hit all IOCTLs
 - Usermode buffer mutated in another thread
 - Basic awareness of what was expected data

Future Work

- “Native” AFD library
 - Skip user mode winsock entirely
 - Compile into shellcode for use in a sandbox
 - Feedback into a more intelligent fuzzer
- More fuzzing
 - At scale
 - More expected data structures defined
- Manual review of WSK and SAN functions

Thanks

- Google
- Project Zero
- James Forshaw

Questions

