Tron: He Fights for the User

Alan Bradley

Setup+Testing: Kevin Flynn(s)

Why is no one at the Podium?

- This is a “proof of concept”
  - Anonymous public speech is now a technical possibility.. (Social difficulties?)
- To make it real, we will break USC 17-1201 (the DMCA) live and with simply an act of speech
- The hope is this format will be useful for those who research DRM-related cryptography
  - And of course other censored speech in US and elsewhere.

The DMCA Still Exists

2) No person shall manufacture...any technology.. ***component, or part thereof***, that...

(B) has only ***limited commercially significant purpose*** or use other than to circumvent a technological measure that effectively controls access to a work protected under this title; OR

(C) is ***marketed by that person*** ... for use in circumventing a technological measure that effectively controls access to a work protected under this title

C: First Amendment anyone?

We fought the law...

- Do free research tools have commercially significant purposes? Maybe.. Maybe not..
- What counts as “marketing”? Speech?
  - How about just telling the truth?
- “Tron CAN BE USED as a ***component*** to help circumvent some software copy protection scheme somewhere.”
  - Anything that relies on verifying its copy protection code will do. Can cloak crack, ship it with Tron.
  - Probably better ways to do this.. But a feature is a feature. Gotta ship product :)

A lot to cover

- Two Rootkits-as-Reversing-Tools
- One Rootkit support lib
  - Includes hooking API and binary patching API
- One IDA Plugin
- Talk Setup

Goals: Debugger Hiding and Covert Userland Code Modification and Instrumentation

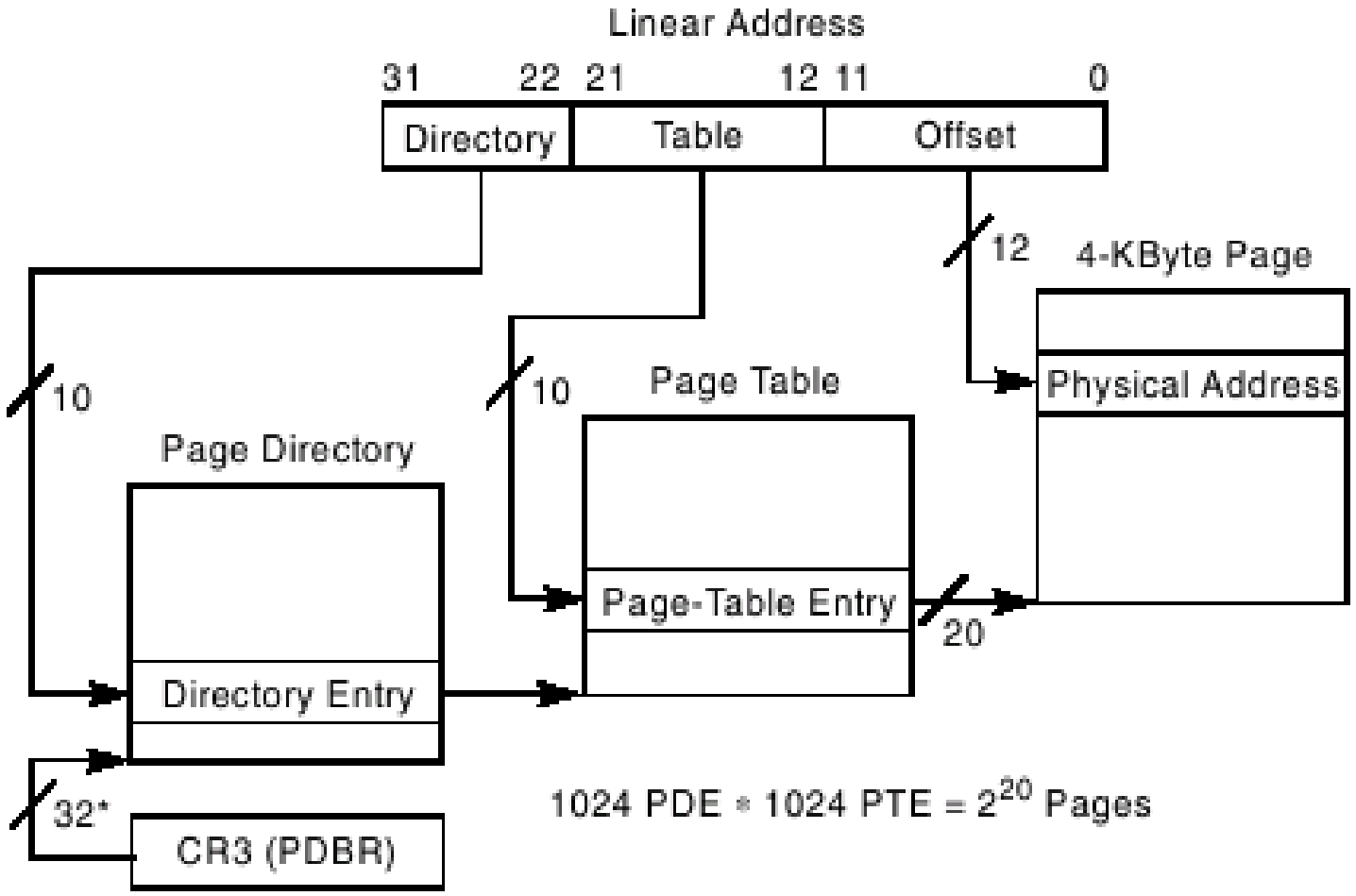
Applications: Malware research, reversing beasts like Skype, DRM research, online games

Rootkit 1: Tron

Virtual Memory Recap

- Virtual Memory: MMU in CPU handles translations of virtual to physical addrs
- Mappings described in page tables
  - Each lookup requires 3 memory accesses
- Cached in two TLBs: Instruction and Data
  - Userland TLBs flushed at context switch
  - But not at thread switch

Page Tables


Linear Address
31
22
21
12
11
0
Directory
Table
Offset
12
4-KByte Page
Physical Address
10
Page Directory
10
Page Table
Page-Table Entry
20
Directory Entry
32*
CR3 (PDBR)
1024 PDE * 1024 PTE = 2^20 Pages
*32 bits aligned onto a 4-KByte boundary.

Shadow Walker Recap

- Mark hidden pages as invalid
- Page fault handler called if:

  1. TLB lookup fails AND THEN

  2a. Page table says not present OR

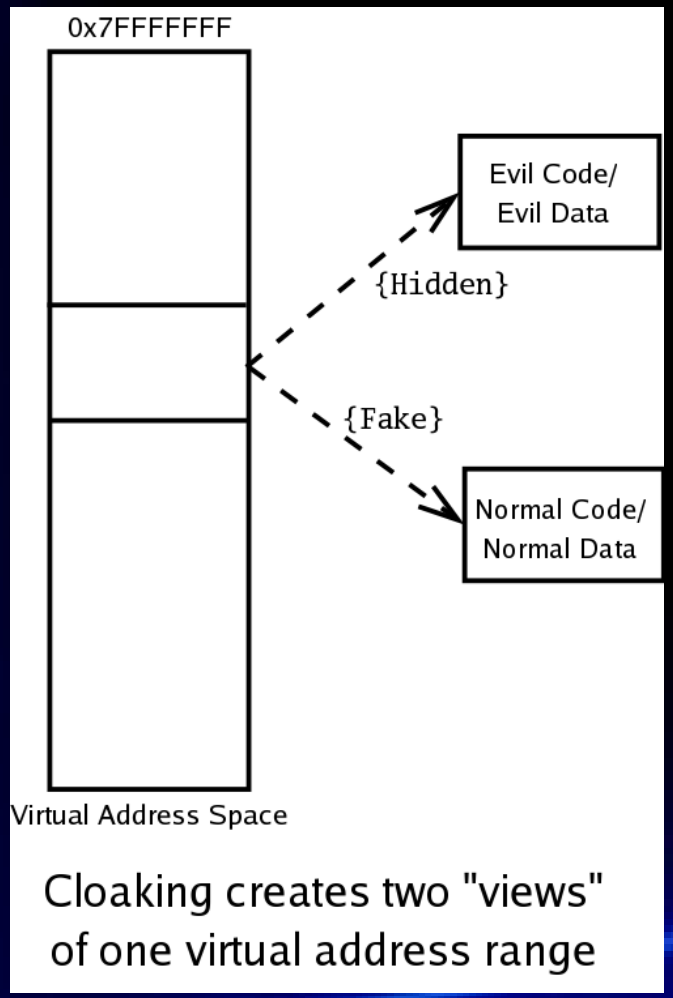
  2b. Page table says can't write OR

  2c. Page table says can't execute (newer x86s)
- Page fault handler puts fake or valid data in TLB depending on access type and permissions
  - EIP of fault = fault addr -> code access
  - 'call' if code, 'mov' if data

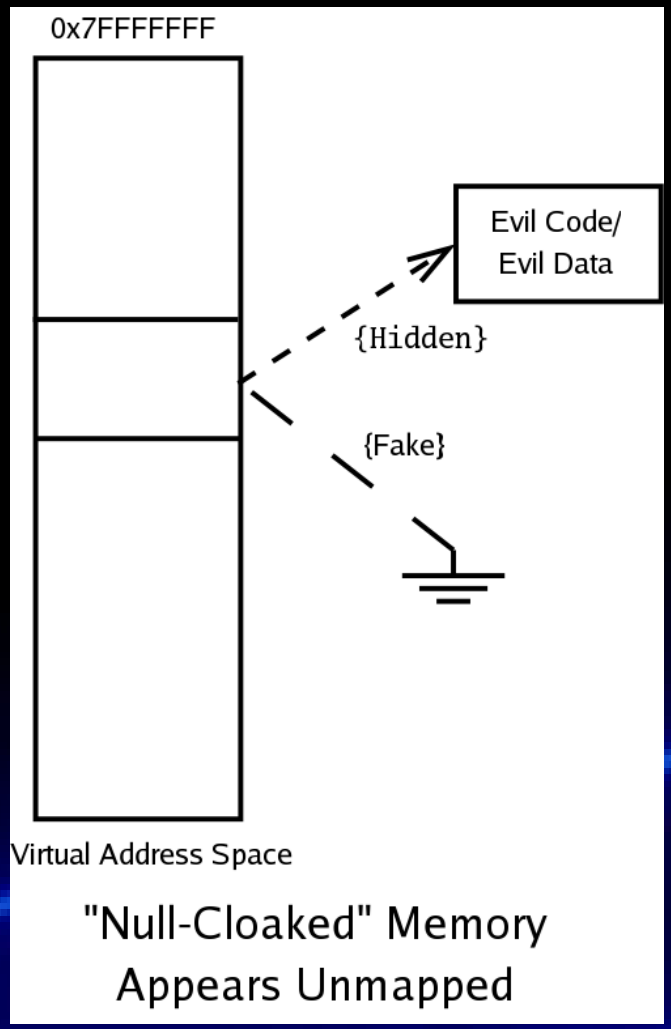
Bugs, Flaws, Limitations

- Does not handle writes (especially COW)
- Races in threaded environment
- No Userland support
- Myopic in usage of cache-priming technique
  - Conceptually, it's better to forget about ITLB vs DTLB and instead think about 'fake' vs 'hidden' and access permissions for this.
  - Code AND data cloaking is possible
- Really not so useful to hide rootkits..
  - Until PatchGuard shows up ;)

Visualizing Hidden Memory

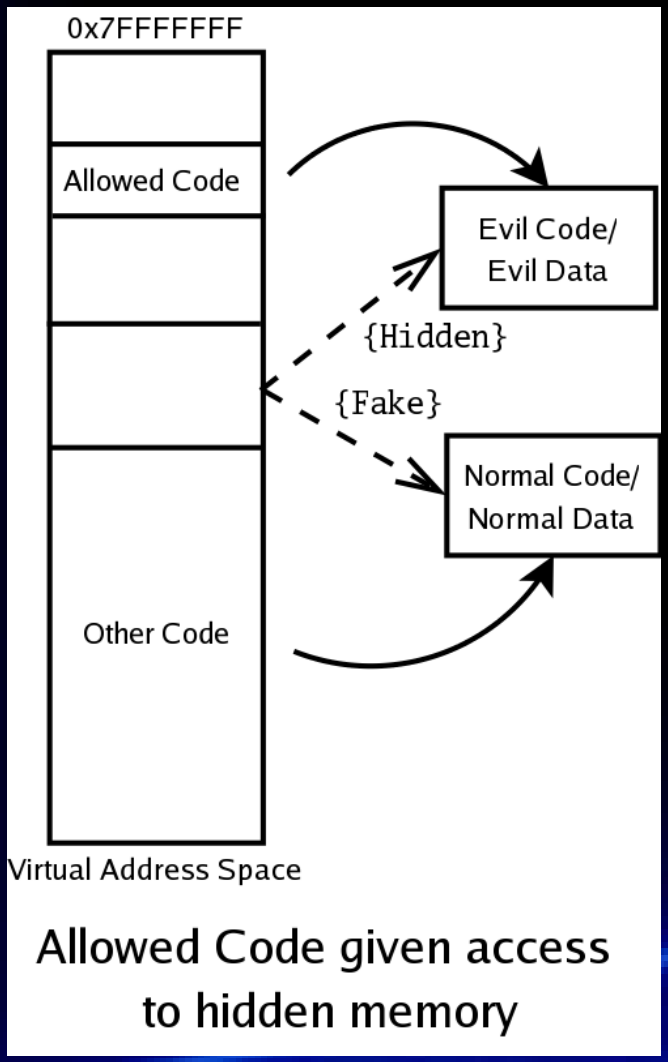


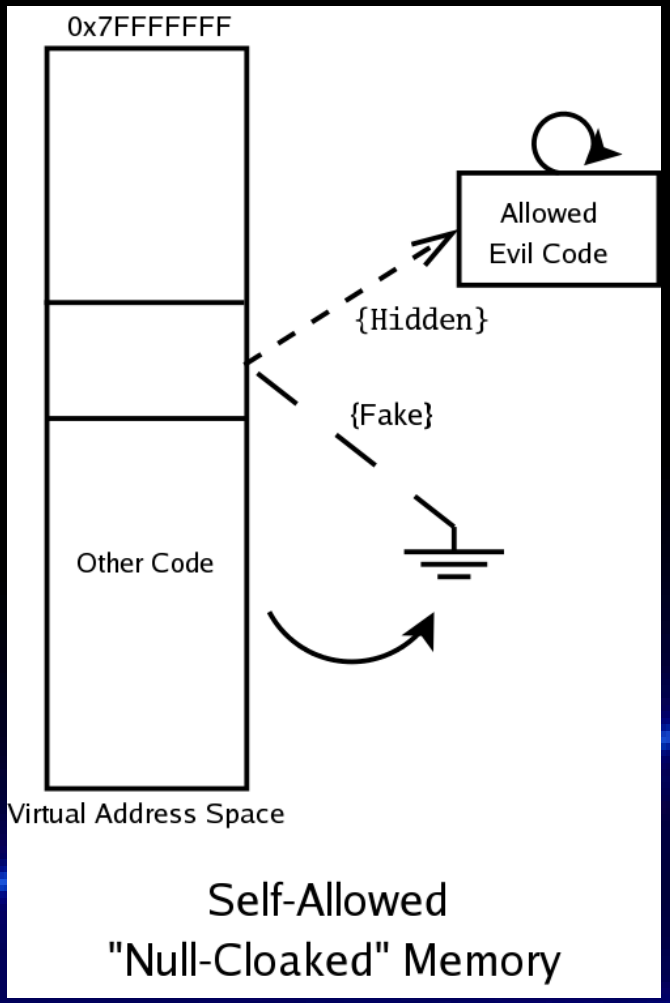
Cloaking creates two "views" of one virtual address range



"Null-Cloaked" Memory Appears Unmapped

Visualizing Allowed Code





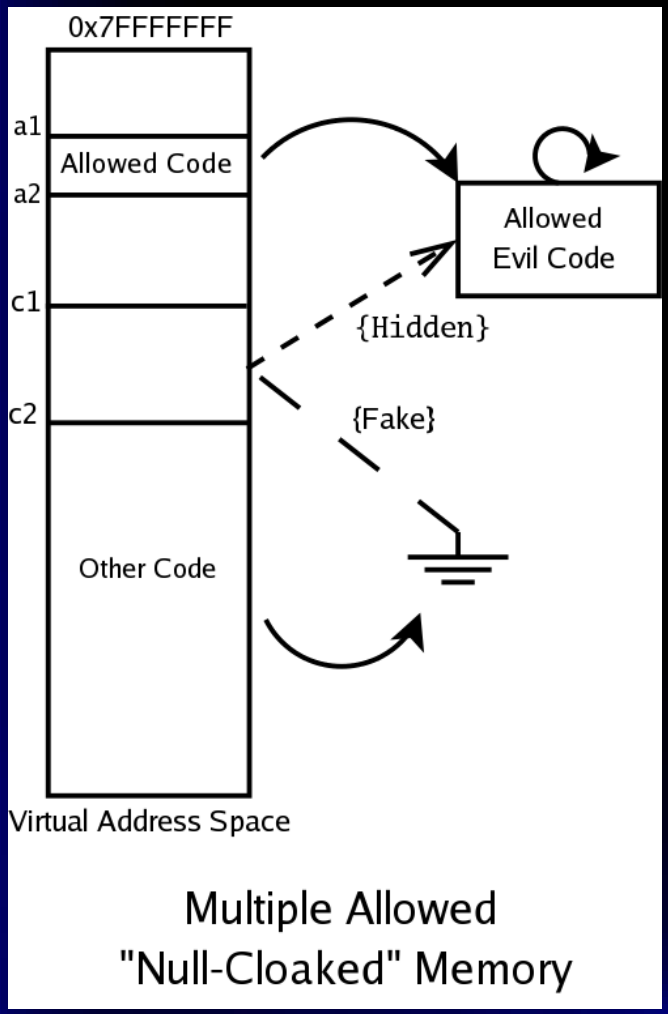
Core Tron API

All APIs implemented via nonce hijack of VirtualAlloc

- ADD_CLOAK(pid, start, end, fake_start, fake_end)
  - Fake is copied to kernel. Can use local stack
  - VirtualLock and COW probing done for you
- REMOVE_CLOAK(pid, cloak_start, cloak_end)

- ADD_ALLOWED(pid, code_start, code_end, cloak_start, cloak_end)
  - Allowed code regions need not be page aligned
- REMOVE_ALLOWED(pid, code_st, code_end)

Core Usage Example



Multiple Allowed
"Null-Cloaked" Memory

```
#include “tron_user.h”
ADD_CLOAK(c1, c2, 0, 0)
ADD_ALLOWED(c1, c2, c1, c2)
ADD_ALLOWED(a1, a2, c1, c2)
```

Tron Convenience APIs

Helper functions to make things easier for you!

- HIDE_DLL_BY_NAME/HANDLE(pid, wcname/handle, fake_st, fake_end)
  - Convenience wrappers for hiding memory and creating self-allowed code in one package.
  - Edits module list for you (Uses WinXP offsets)
- WRITE_HIDDEN(pid, dest, srcbuf, len)
- READ_HIDDEN(pid, dest, outbuf, len)

Tron Race-Proofing APIs

- CHANGE_TRUST(pid)
  - Flushes TLBs. Call before/after running allowed code to prevent intra-thread scanning
    - Ex: job-based scanning running on your thread
- PATCH_SCHEDULER(IDASwapContext)
  - Give it the address of ntoskrnl's SwapContext() from IDA to 'hotpatch' the Windows scheduler
  - Causes scheduler to flush TLB on thread switch
  - Addresses inter-thread access races
  - Useful for apps with dedicated scanner thread

Misc Tricks

- Allows you to “Null-cloak” memory, which makes it look unmapped
  - Hooks VirtualQuery and friends to lie
- Kernel is NOT trusted, so syscalls give fake views
- Normal memory permissions are enforced

Tron Weaknesses

- Hidden code does NOT run in VMWare
  - VMWare's iret seems to flush ITLB :(
    - Causes page fault loop on hidden code execution
  - Data hiding works just fine though
- Breakpoint verification
  - Scanner sets breakpoint, installs handler, jumps to breakpoint, verifies handler runs
  - Only works if hidden data actually presents itself as mapped. “Null-cloaked” memory is safe. Touching it causes access fault

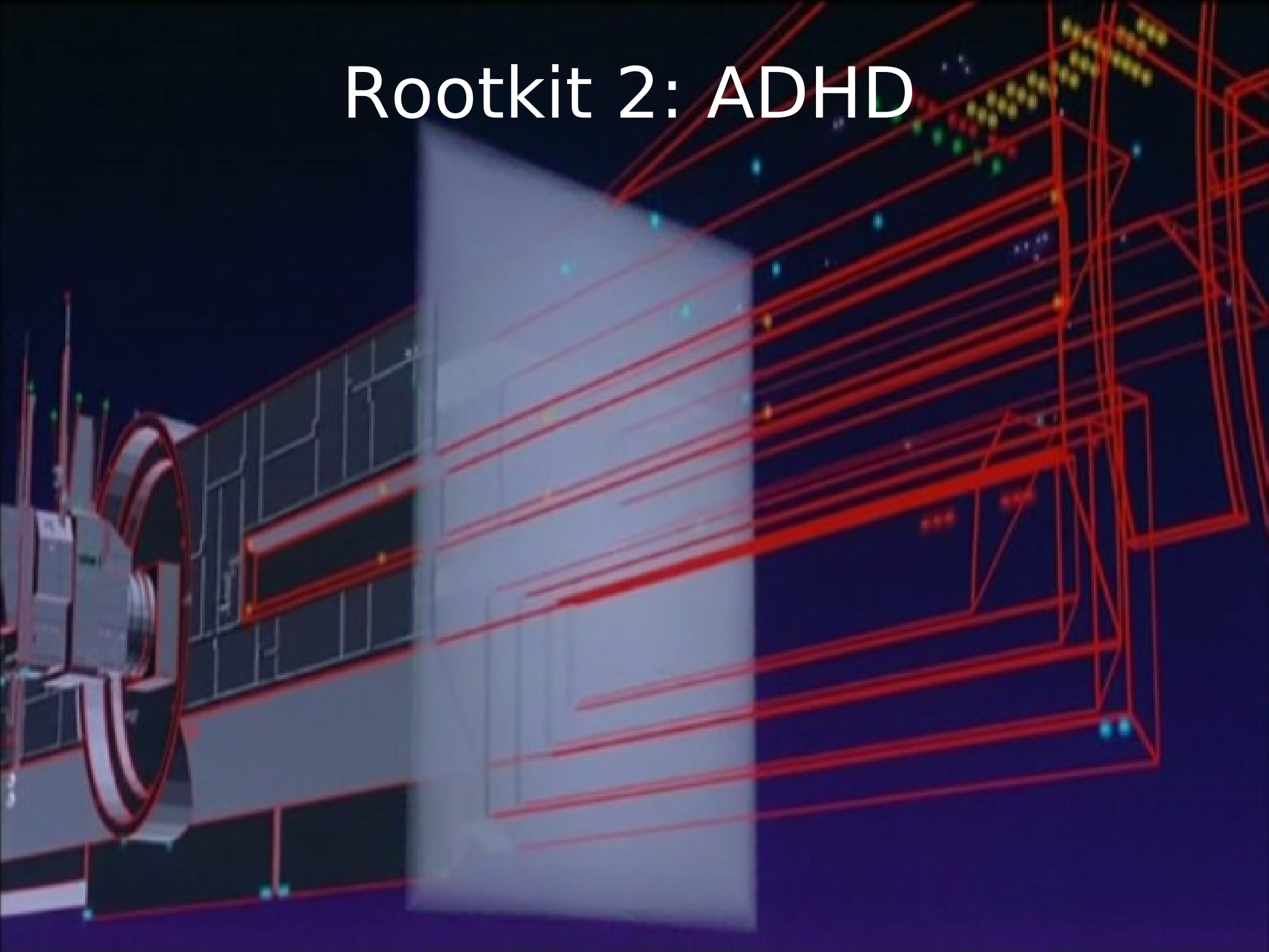
Room for Improvement

- NOT SMP/HT SAFE
- API non-reentrant
- Not 64BIT Clean
  - Fair amount of x86 asm
  - DWORD/PVOID mixing madness
  - 64Bit will also require PatchGuard break
- Optimizations
  - Borrow either ProtoPTE or Reserved PTE bit so hooks are not always searched
  - Make hook arrays into hash tables

Thoughts on PatchGuard

- Essentially locks out custom kernel modifications and 3rd party scanners
- Will likely get in the way of kernel, malware, botnet, and worm research; also misc reversing.
  - All these people will look for breaks
  - Yet legit AV is helplessly locked out
- Provides a possible deterministic source of page faults to help cloak a page fault handler
  - Mark div fault handler NX
  - If EIP == KiOp_Div, prime DTLB with fake page fault handler and we are GHOST

Rootkit 2: ADHD



Debugger Detection (1)

- PEB->BeingDebugged
  - IsDebuggerPresent()
  - CheckRemoteDebuggerPresent()
- NtQueryInformationProcess(DebugPort)
- Write code to DbgUiRemoteBreakin or DbgBreakPoint in ntdll
  - Kakeeware's AntiDebug
- Check Parent PID
- Call NtSetInfoProc(ThreadHideFromDebugger)

Debugger Detection (2)

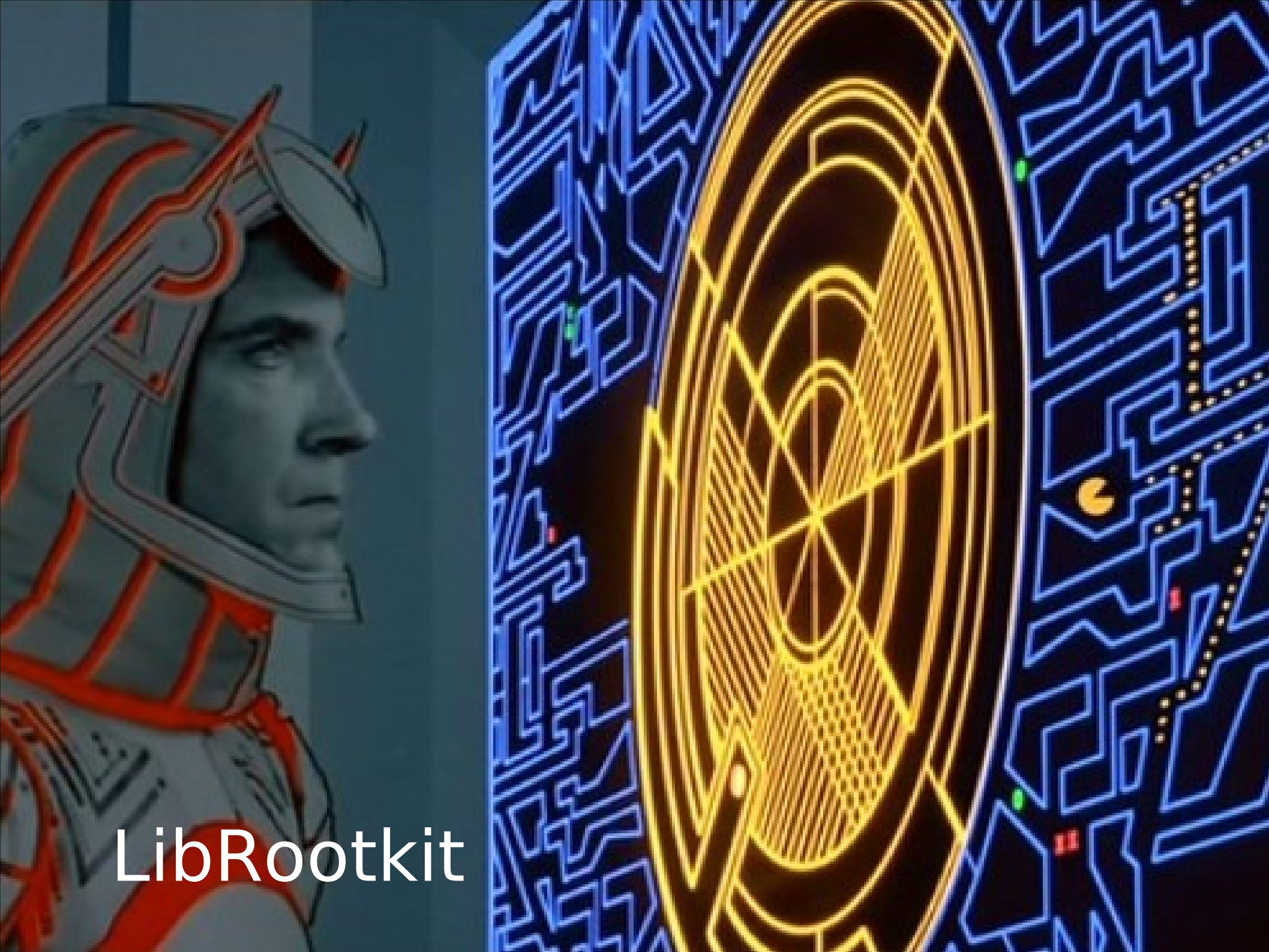
- Exception Raising/Checking
  - Write 0xcc, verify handler gets exception
- Wall Clock Time
- Check Window Titles
- Check running processes

Another Debugger Hiding Driver

- Resets PEB->BeingDebugged flag
- Hooks ZwQueryInformationProcess
  - Zeroes DebugPort
- Protects DbgUiRemoteBreakin and DbgBreakpoint from hooks
  - Defeats Kakeeware's Antidebug
- Resets parent PID to explorer.exe
- Blocks ZwSetInfoProc(ThreadHideFromDbgr)

ADHD Weaknesses

- Exception redelivery needs to be handled by userland debugger
  - IDA has options to pass through exceptions
- GetTickCount/NtSystemTime() not spoofed
  - Not hookable from kernel.. Shared memory
  - Script time-critical areas in debugger
- No process/title cloaking
  - CLU spoofs IDA's window title
  - Use FUTo to hide process


LibRootkit

Noteworthy LibRootkit APIs

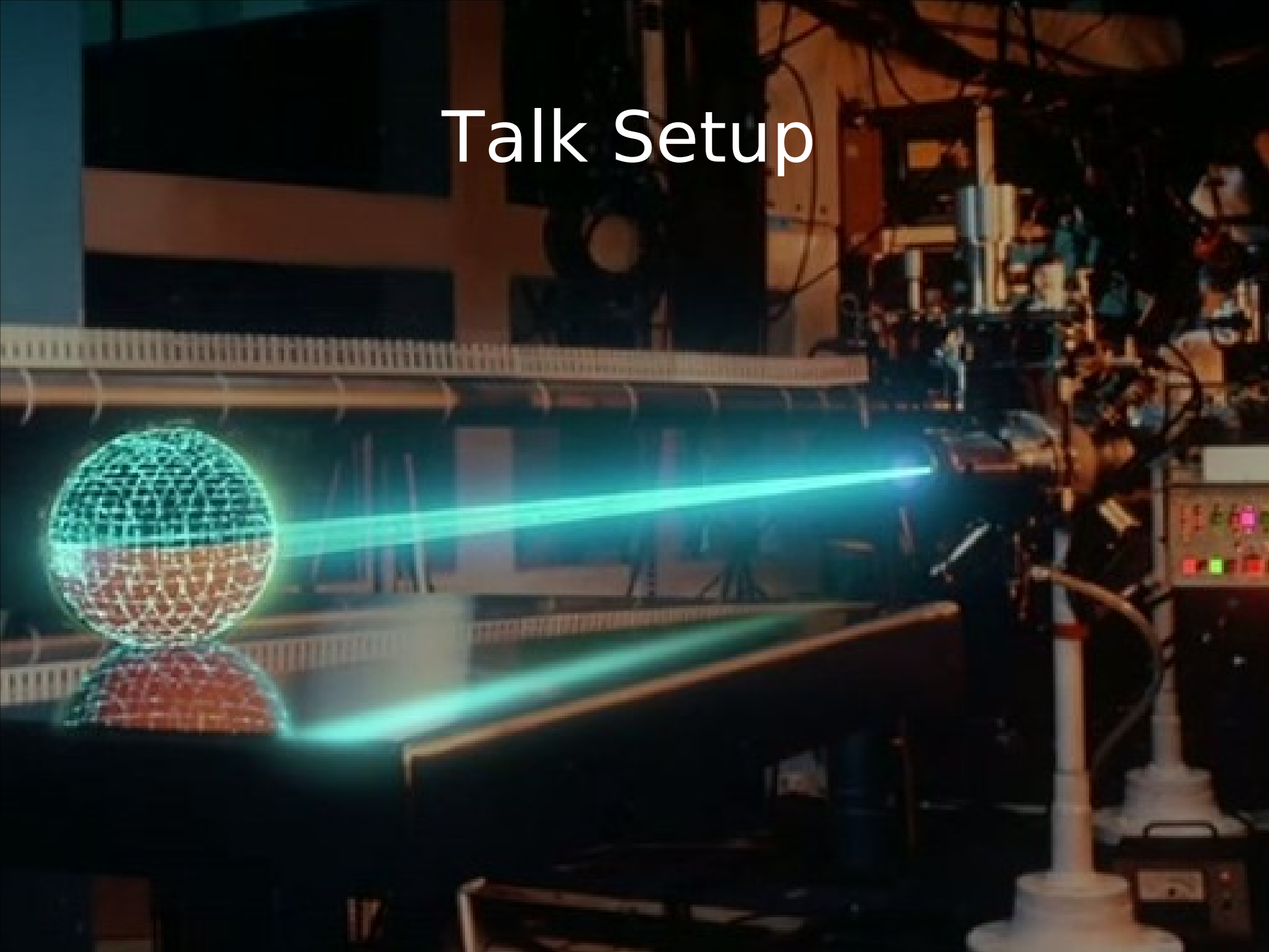
- HookSysCall(char *syscall, DWORD NewFcn)
  - Hooks a syscall by name, returns old pointer
  - Uses ntdll. Based on gareth's rk.com post
- SetupHotPatch(ToPatch, Tgt, Buf, BufLen)
  - Uses Nicolas Capens's x86 length decoder
  - Used by Tron to patch SwapContext
- ProbeForRead/WriteStatus, Get/SetIntVector, TEB “borrowing”, GetShadowTable, logging...


CLU

CLU

- Hooks IDA's debugger plugin functions to set hidden breakpoints with Tron
  - Software breakpoints only
- IDA debugger plugin API is broken/unsane
  - Never calls write_memory....
  - Even worse, randomly writes breakpoint ops in between add_bpt and del_bpt calls
  - Had to Detour WriteProcessMemory to hook
- Also changes IDAs window titles to random strings to avoid detection

Talk Setup



Components

1. Speaker laptop with
   a. Ventrilo Client
   b. Voice Changer
   c. VNC Client
   d. Tor
2. Linksys in client mode (firewall + antenna)
3. VM image to make it easier on h1kari (<3)
   a. VNC server
   b. Talk tools + slides + demos
   c. Ventrilo Server + Client
4. Plenty of paranoia
5. 

Local Setup

- Find coffeeshop/hotspot/friend's apartment
  - Change Linksys MAC address
  - Put Linksys (dd-wrt) into client mode
- Ventrilo has UDP traffic (and phones home)
  - Block all but Tor server at Linksys router:

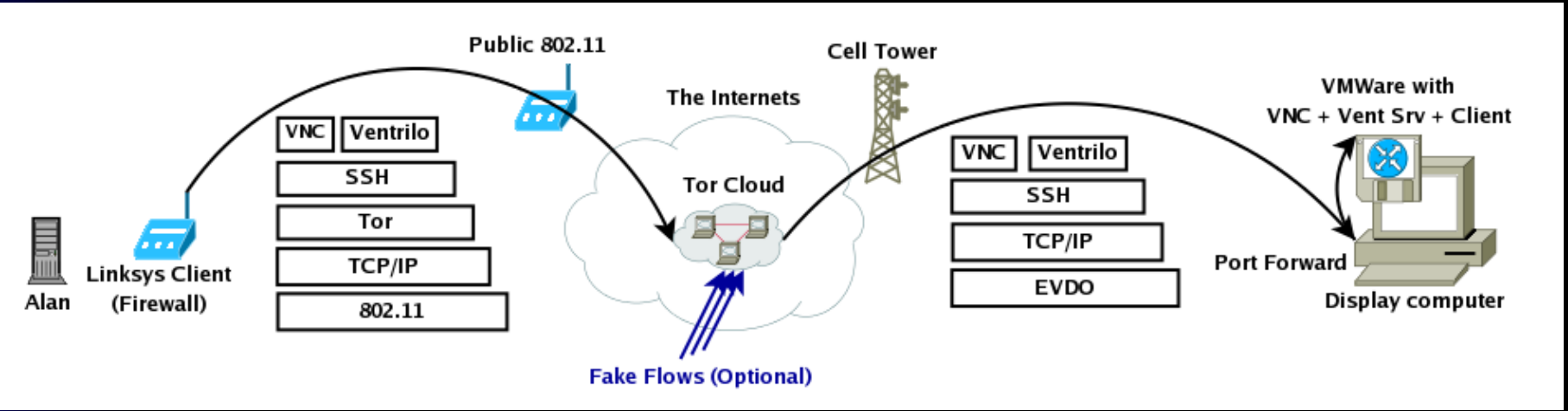
```
iptables -A OUTPUT -d tor-router-ip -j ACCEPT
iptables -A OUTPUT -j DROP
```

- Setup torrc:

```
__LeaveStreamsUnattached 1
ControlPort 9051
```

- Tunnel VNC + Ventrilo over SSH -L
- Echo script because Chaos is a Bitch

Setup Diagram



- Only one Tor circuit needed
  - Easy to manually build one
- Fake Tor flows if “Global Adversary” is present
  - EU data retention may or may not qualify..
- VM can be Linux once Ventrilo client is done
  - Eliminates serial number + update issues

Building a Custom Tor Circuit

```
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.

AUTHENTICATE
250 OK
SETEVENTS STREAM CIRC
250 OK
EXTENDCIRCUIT 0 ccc2,morphium,altron
250 EXTENDED 27
650 CIRC 27 EXTENDED ccc2
650 CIRC 27 EXTENDED ccc2,morphium
650 CIRC 27 EXTENDED ccc2,morphium,altron
650 CIRC 27 BUILT ccc2,morphium,altron

650 STREAM 81 NEW 0 66.102.7.147:80
ATTACHSTREAM 81 27
250 OK
650 STREAM 81 SENTCONNECT 27 66.102.7.147:80
650 STREAM 81 SUCCEEDED 27 66.102.7.147:80
```

Be sure to check the exit policy!

Documents + Tools

- All IDA copies tagged, but SDK is clean
- OpenOffice used for slides
  - Easier to verify no personal info in docs
  - Found username/path, even if “Apply User Data” is unchecked
  - M$ has metadata tool, but no thanks :)
- Windows serial + AutoPatcher updates for vm
- DDK is free but MS will only mail it...
- Set windbg symbol path to search locally first
- Tor used for Google, mail, etc

Thoughts on Anonymity

Easy to Screw Up

- Even if you are technically sound, friends will:
  - Post on forums or /. to defend your talk/idea
  - Repeatedly visit con website from work IP
  - Mention/brag/blog they know the speaker
- Subtle intersections add up:
  - Who purchased equipment you used?
  - Who has the expertise needed to do this?
  - Mistakes during talk setup testing
  - Slide style, coding style, speaking style..
- From here, legwork & interrogation begin

Numerous Downsides

- No resume padding
- No PR whoring
- Audience deprived of human element
  - Interaction still possible, but not optimal
- Speaker deprived of networking with others who share interest
- If you screw up, you look dumb :(
  - And may or may not be in some trouble with the law and/or angry thugs

Is it worth it?

- We're hackers. We'll try anything once ;)
  - It was a hell of a lot of fun to try
  - Helps raise awareness of privacy-enhancing technology
- If we fail, we have a reasonable test case against the DMCA, and/or people can learn from our mistakes
- For DRM research and whistle blower info, maybe some talk is better than no talk
  - Have a look at 'States Secrets Privilege' on Wikipedia some time...

Contact Info

Email: abradley@fastmail.fm

IRC (temporarily): irc.freenode.net #tron

Talk setup paper will be posted to Tor Wiki

http://wiki.noreply.org/noreply/TheOnionRouter

Look for Tron and pals on OpenRCE.org

(also ToorCon website)