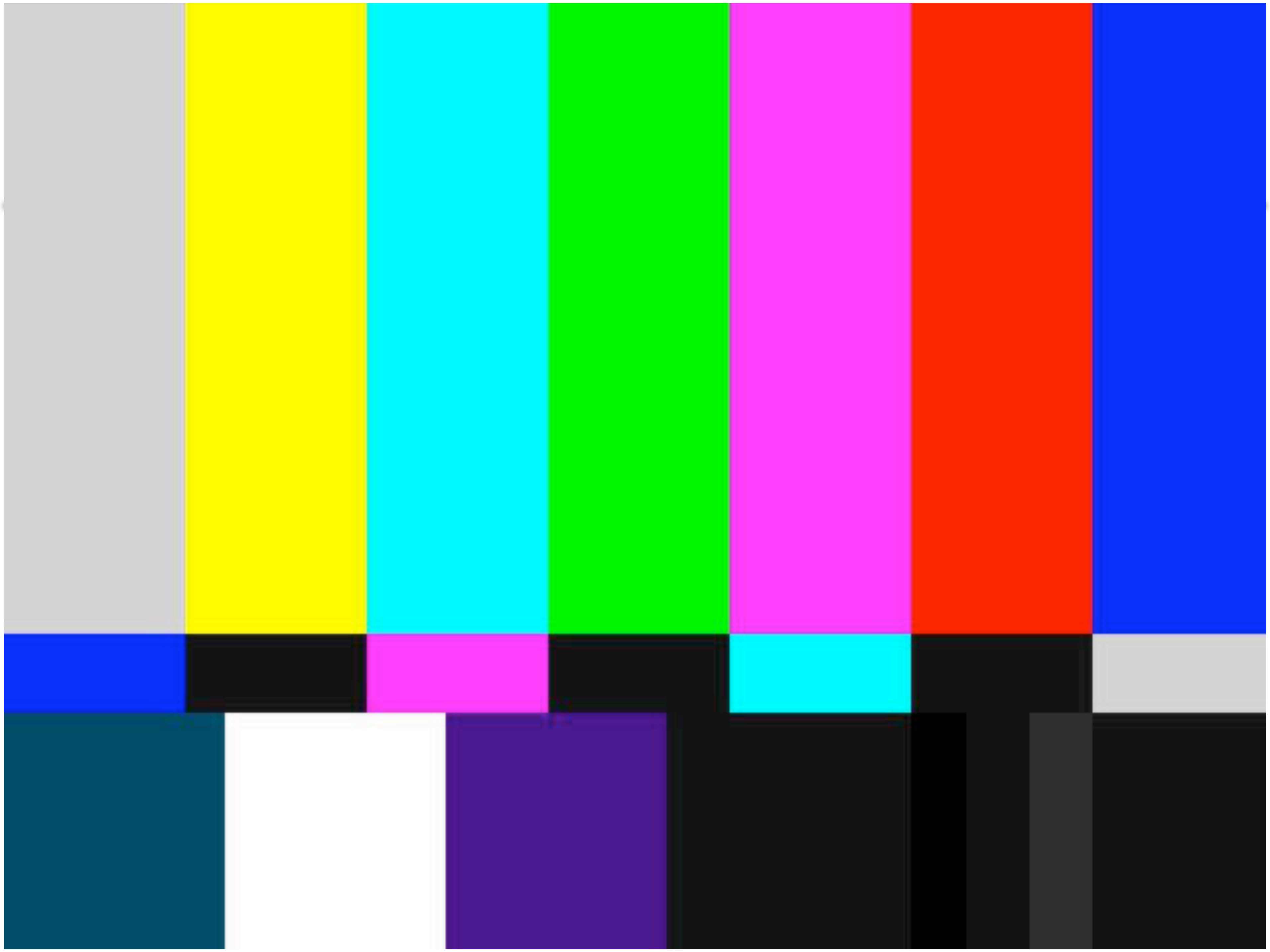
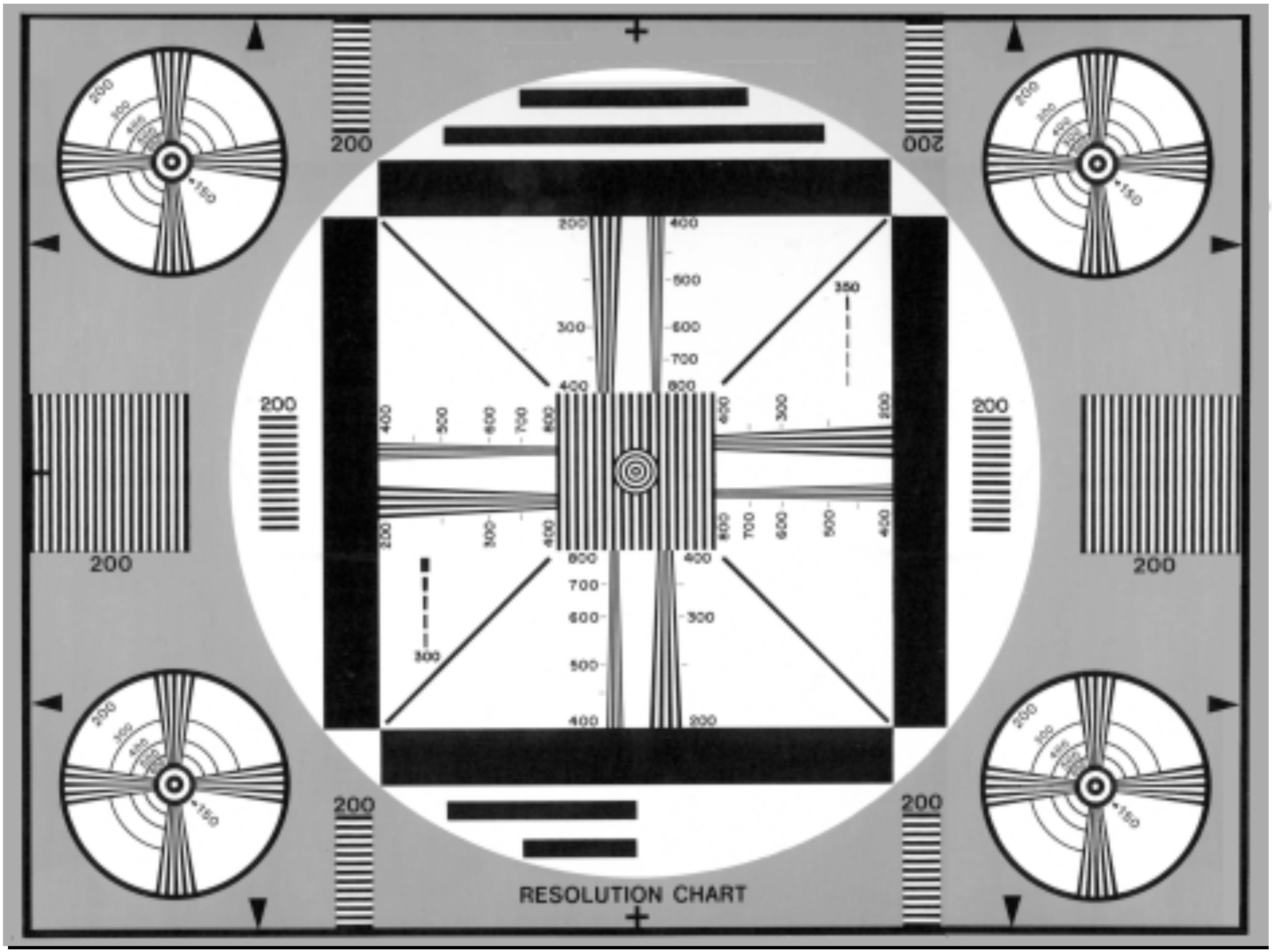



200
300
400
500
600
700
800
150
350
RESOLUTION CHART

(un)Smashing the Stack

(Overflows, countermeasures, and the real world.)

Shawn Moyer
agura digital security
blackhat@agurasec.com

BlackHat®
DC 2008

Hey, who is this guy?

- Attacker by nature, defender by trade

Hey, who is this guy?

- Attacker by nature, defender by trade
- IRDF, WebAppSec, “Architect” ← (LOLOLOL)

Hey, who is this guy?

- Attacker by nature, defender by trade
- IRDF, WebAppSec, “Architect” ← (LOLOLOL)
- Obsessive-compulsive quixotic insomniac with messianic tendencies

#include std_disclaimer.h

=

My humble attempt to understand a complex topic.

Whiskey Tango Foxtrot?

- The Exploitation Wayback Machine™
  - What did Lincoln say about history?

Whiskey Tango Foxtrot?

- The Exploitation Wayback Machine™
  - What did Lincoln say about history?
- Exploit Mitigation
  - Compile bits, lib bits, kernel bits
  - Memory integrity, canaries
  - Anti-heterogeneity (ASLR, PIC/PIE)

Whiskey Tango Foxtrot?

- The Exploitation Wayback Machine™
  - What did Lincoln say about history?
- Exploit Mitigation
  - Compile bits, lib bits, kernel bits
  - Memory integrity, canaries
  - Anti-heterogeneity (ASLR, PIC/PIE)
- Bonus defensive fu
  - MAC / MIC
  - Static analysis
  - Rubber meets the road

InfoSec is a fork bomb

- PatchThenScanThenPatchThenScanThenPatchThenScanThenPatchThenScanThen …

BlackHat DC 2008

InfoSec is a fork bomb

- PatchThenScanThenPatchThenScanThenPatchThenScanThenPatchThenScanThen …
- Retrofit of the 80's antivirus model
  - Patches (and exploits) on a subscription basis

BlackHat DC 2008

InfoSec is a fork bomb

- PatchThenScanThenPatchThenScanThenPatchThenScanThenPatchThenScanThen …
- Retrofit of the 80's antivirus model
  - Patches (and exploits) on a subscription basis
- Getting a bit old, innit?

Breaking the membrane

- Corruption of memory space == control of execution flow
  - Hilarity ensues.

Breaking the membrane

- Corruption of memory space == control of execution flow
  - Hilarity ensues.
- As far back as the 1960's…
  - Overrun screw, wild pointer, stack scribbling, fandango on core

@ pre-epoch

@ the epoch

In the age of the dinosaurs

- Spaff's Morris doc + RFC 1135, circa 1988
  - Stack-based BO in fingerd gets() call
  - Spaff: Avoid unsafe calls in C, mmm-kay?

In the age of the dinosaurs

- Spaff's Morris doc + RFC 1135, circa 1988
  - Stack-based BO in fingerd gets() call
  - Spaff: Avoid unsafe calls in C, mmm-kay?
- http://www.securityfocus.com/bid/2
- Happy 20th birthday, cluephone.

Things get interesting

- Lopatic, circa 1995
  - Stack-based BO in NCSA httpd
  - “Looks like Morris”… Hrmm.

Things get interesting

- Lopatic, circa 1995
  - Stack-based BO in NCSA httpd
  - “Looks like Morris”... Hrmm.
- Mudge, circa 1995
  - *“How to write buffer overflows”*
  - Shellcode w/o ASM, NOP sleds

BlackHat DC 2008

Curiouser and curiouser

- Aleph One, circa 1997
  - Snapshot of attack landscape in the 90’s
  - Memory segments, “eggs”, NOPs

Curiouser and curiouser

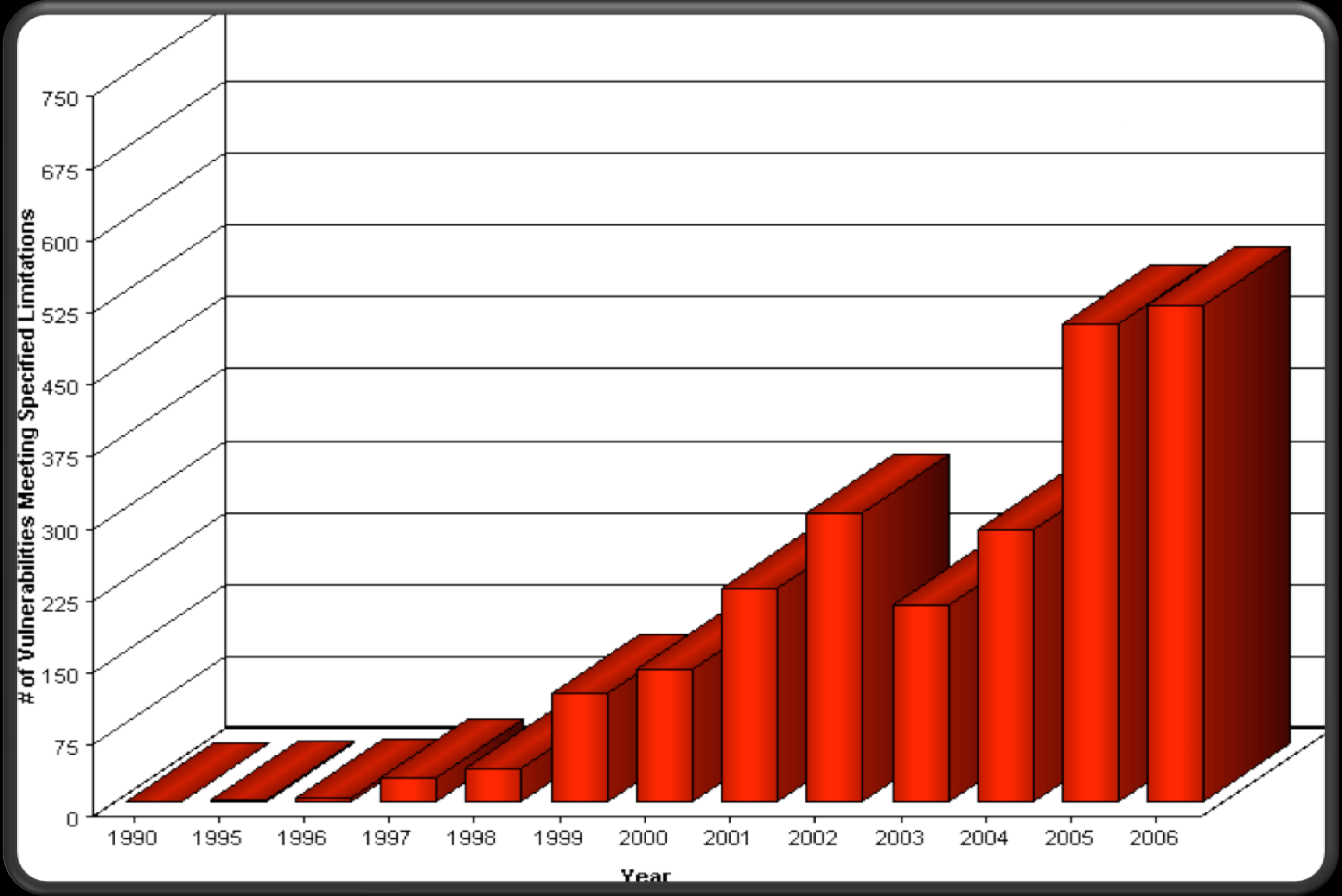
- Aleph One, circa 1997
  - Snapshot of attack landscape in the 90's
  - Memory segments, "eggs", NOPs
- Solar Designer, circa 1997
  - Ret2libc: call preloaded functions in payload
  - Works without stack execution

Curiouser and curiouser

- Aleph One, circa 1997
  - Snapshot of attack landscape in the 90's
  - Memory segments, "eggs", NOPs
- Solar Designer, circa 1997
  - Ret2libc: call preloaded functions in payload
  - Works without stack execution
- Conover / woowoo, circa 1999
  - *"woowoo on heap overflows"*
  - Writes to the heap, function ptr overwrites

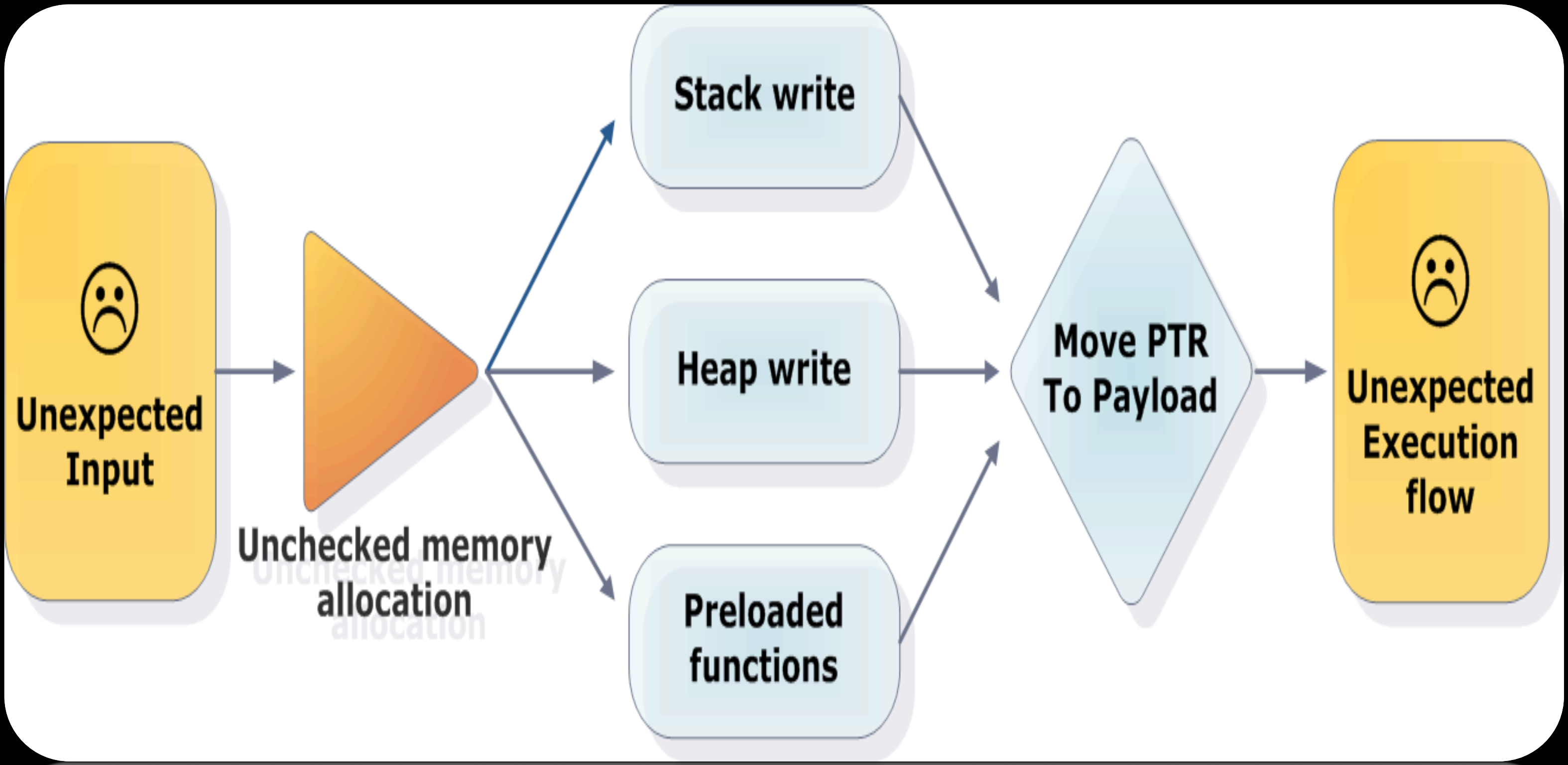
BlackHat DC 2008

NIST NVD remote BO's



BlackHat DC 2008

One more time, for the CISSPs



NX

- Nonexecutable stacks
  - Data is data, code is code, right?
  - Ne’er the twain shall meet

Stack prophylactics

- Solaris / uSparc
  - noexec_user_stack = 1

Stack prophylactics

- Solaris / uSparc
  - noexec_user_stack = 1
- nX, XD, on IA64, AMD64, others
  - PAE bit 63 0/1
  - Opt-in: OS, libs, etc must flip this bit

Stack prophylactics

- Solaris / uSparc
  - noexec_user_stack = 1
- nX, XD, on IA64, AMD64, others
  - PAE bit 63 0/1
  - Opt-in: OS, libs, etc must flip this bit
- Software emulation
  - Less fine-grained (Segment-based)
  - Solar's StackPatch, PaX, MS DEP, RH ExecShield

Functionality breeds exposure

- Some breakage may occur in shipment
  - JIT compilers, Virtualization
  - Wha? I can’t run my CP/M z80 emulator?

Functionality breeds exposure

- Some breakage may occur in shipment
  - JIT compilers, Virtualization
  - Wha? I can't run my CP/M z80 emulator?
- User-configurable opt-outs
  - ProcessExecuteFlags
  - Mprotect(), VirtualProtect()
  - DEP exceptions list

Counter-countermeasures

- Ret2libc
  - Call preloaded functions
  - Call mprotect(), set new allocation rwx
  - Needs "known" useful address

BlackHat DC 2008

Counter-countermeasures

- Ret2libc
  - Call preloaded functions
  - Call mprotect(), set new allocation rwx
  - Needs “known” useful address
- Heap-based overflows
  - More interesting nowadays
  - Little protection on the heap at this point

Counter-countermeasures (deux)

- Piromposa / Embody
  - “Hannibal attack”
  - Fuction ptr overwrite, shellcode via argv

Counter-countermeasures (deux)

- Piromposa / Embody
  - "Hannibal attack"
  - Fuction ptr overwrite, shellcode via argv
- Skape / Skywing
  - Forcible opt-out in MS DEP via ret2libc
  - MEM_EXECUTE_OPTION(ENABLE|DISABLE)
  - "*/noexecute=AlwaysOn*" boot.ini flag

Counter-countermeasures (trois)

- Optional security, isn’t.
  - Compiler flags rarely on by default
  - Most optimization flags disable checks
  - Trampolines, workarounds, other ugliness

Canary in a coalmine

- “Tripwire for the stack”
  - Compiler extensions to detect corruption
  - Initially, canary value of RTA (StackGuard)
  - Halt execution if value changes (function_epilogue)

Canary in a coalmine

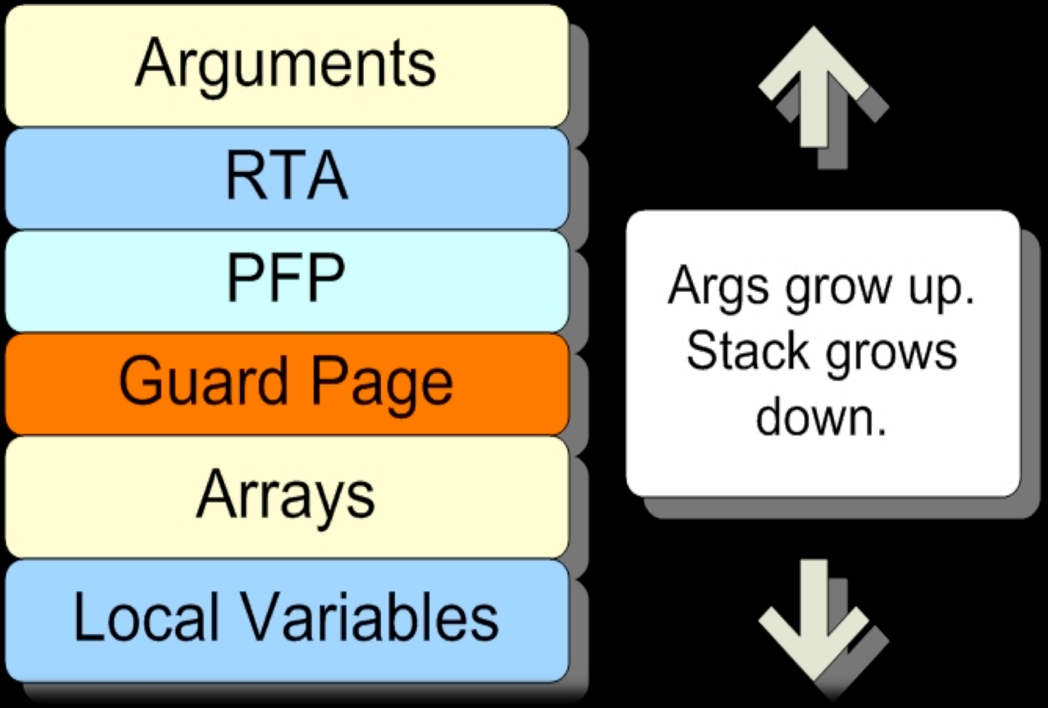
- "Tripwire for the stack"
  - Compiler extensions to detect corruption
  - Initially, canary value of RTA (StackGuard)
  - Halt execution if value changes (function_epilogue)

- ProPolice / SSP
  - GCC > 4.1 integration, backports
  - MS adopted as /GS extensions
  - "Guard value", stored off-stack
  - Beyond canaries: Well-ordered stack

BlackHat DC 2008

Propolis

Safe Stack


Arguments
RTA
PFP
Guard Page
Arrays
Local Variables
Args grow up. Stack grows down.

(un)Smashing the Heap!

- Heap canary implementations!
  - Guard values around malloc()
- OpenBSD “G” option to malloc.conf
- Contrapolice
  - **http://synflood.at/contrapolice.html**
- wkr’s dlmalloc extensions
  - http://www.cs.ucsb.edu/~wkr/projects/

Killing the canary

- Gerardo of CoreSec:
  - GOT and PLT writes, SFP overwrites

Killing the canary

- Gerardo of CoreSec:
  - GOT and PLT writes, SFP overwrites

- HERT, Phrack 56
  - RTA-only problems in StackGuard
  - Overwrites to RTA without harming canary

Killing the canary (deux)

- Canary as target
  - Arbitrary memory reads, format string bugs
  - /proc/mem, other info leakage

Anti-heterogeneity

- PaX
  - The originator of this concept.
  - Userland, kstack, mmap()
  - Tunable knobs (paxctl / sysctl)

Anti-heterogeneity

- PaX
  - The originator of this concept.
  - Userland, kstack, mmap()
  - Tunable knobs (paxctl / sysctl)
- OBSD 3.3+
  - Randomized malloc(), mmap(), gaps / fencing

Anti-heterogeneity

- PaX
  - The originator of this concept.
  - Userland, kstack, mmap()
  - Tunable knobs (paxctl / sysctl)
- OBSD 3.3+
  - Randomized malloc(), mmap(), gaps / fencing
- ExecShield
  - Stack, base randomization, also noexec

Anti-heterogeneity (deux)

- Vista
  - Random .exe and .dll loader
  - /dynamicbase flag, opt-in model
  - Weaker on the heap (see Whitehouse/BHDC07)

Anti-heterogeneity (deux)

- Vista
  - Random .exe and .dll loader
  - /dynamicbase flag, opt-in model
  - Weaker on the heap (see Whitehouse/BHDC07)
- Leopard
  - Randomized libs, not heap or stack
  - Mach arch limitations – some fixed addresses

Anti-heterogeneity (trois)

- PIC or PIE
  - Execute sanely, regardless of location
  - Find the GOT and get random

Anti-heterogeneity (trois)

- PIC or PIE
  - Execute sanely, regardless of location
  - Find the GOT and get random
- Key to full ASLR
  - Without, only defended against ret2libc
  - 1 in 2^(STACK_RAND + MMAP_RAND)

Elegant solution, meet brute force.

- Hovav Shacham
  - Derandomization attack
  - Brute-force system() on forking service
  - What about client-side? Browser?

BlackHat DC 2008

Elegant solution, meet brute force.

- Hovav Shacham
  - Derandomization attack
  - Brute-force system() on forking service
  - What about client-side? Browser?
- Bonus unrelated cool Hovav stuff
  - ret2libc without function calls
  - Sequence chaining, “gadgets”

Brute force (deux)

- Ben Hawkes
  - Code-access brute-forcing
  - Unsuccessful reads to get ret2libc

Brute force (deux)

- Ben Hawkes
  - Code-access brute-forcing
  - Unsuccessful reads to get ret2libc
- Whitehouse / BHDC07
  - Varying degrees of randomization in Vista
  - Especially on heap

Exploit Mitigation: Cliff's Notes

- Noexec / NX
  - If runtime configurable it's pointless

BlackHat DC 2008

Exploit Mitigation: Cliff's Notes

- Noexec / NX
  - If runtime configurable it's pointless
- Canaries
  - Bad crypto != panacea
  - All memory space requires protection

Exploit Mitigation: Cliff's Notes

- Noexec / NX
  - If runtime configurable it's pointless
- Canaries
  - Bad crypto != panacea
  - All memory space requires protection
- ASLR
  - Bad crypto != panacea
  - Memory leaks, inconsistencies

Other ways to skin a cat

- Fix the @#$% code?
  - RATS, Flawfinder, FORTIFY_SOURCE
  - Lots of commercial stuff, obviously
  - DHS / Coverity joint project

Other ways to skin a cat

- Fix the @#$% code?
  - RATS, Flawfinder, FORTIFY_SOURCE
  - Lots of commercial stuff, obviously
  - DHS / Coverity joint project
- Rice's Theorem, Rumsfeld's Corollary
  - Automated analysis goes only so far
  - Unknown unknowns

Cat skinning redux

- Access control models
  - Post-exploit containment
  - File, device, inode

Cat skinning redux

- Access control models
  - Post-exploit containment
  - File, device, inode
- Another way to contain exposure
  - Varying degrees of complexity
  - Linuces, Vista, BSDs, now Leopard

Rubber, meet road.

- PaX
  - "The guaranteed end of arbitrary code execution"
  - SEGMEXEC, PAGEEXEC, sigtramp emulation
  - ASLR in userland, kstack
  - Configurable bits for misbehaving binaries
- Integration
  - http://kernelsec.cro.org
  - Hardened Gentoo, Ubuntu-Hardened

Rubber, meet road (deux)

- OpenBSD
  - First to integrate ProPolice / SSP
  - Heap canaries, W^X, ASLR
  - Mprotect () works, no rand or noexec for kstack
- FreeBSD
  - Very basic NX, other projects to add SSP
- NetBSD
  - Adding SSP, PaX-inspired bits to 4.0

Rubber, meet road (trois)

- Vista
  - ASLR, PIC/PIE, MIC, DEP / NX
  - Consistency is an issue
  - What is Crispin doing?
- 2003 / XP
  - DEP/NX, canaries
  - Wehnus! http://www.wehnus.com

Rubber, meet road (quatre)

- OSX Leopard
  - First toe in the water
  - Simple NX, heap remains executable
  - Seatbelt.kext / sandboxing based on policies
  - ASLR limitations due to Mach arch

Do you feel safer yet?

- Time to fire Von Neumann?
  - The computing model needs to change.
  - Compartmentalized Operating Systems?
  - Academia, where are you?

Do you feel safer yet?

- Time to fire Von Neumann?
  - The computing model needs to change.
  - Compartmentalized Operating Systems?
  - Academia, where are you?
- The devil is in the details
  - Legacy support, compatibility
  - Opt-in models for consumer OS's

Do you feel safer yet?

- Time to fire Von Neumann?
  - The computing model needs to change.
  - Compartmentalized Operating Systems?
  - Academia, where are you?
- The devil is in the details
  - Legacy support, compatibility
  - Opt-in models for consumer OS's
- Defenders need to understand this
  - Jon Erickson is teh awesome+++
  - What's in *your* stack?

Thanks for listening.

- http://pax.grsecurity.net
- http://www.wehnus.com
- Thanks to DT, Ping, Dom, BH goons
- Much love to everyone working on this!