BLACK HAT BRIEFINGS

Alex Wheeler
Neel Mehta

Owning Anti-Virus: Weaknesses in a Critical Security Component

AV software is becoming extremely popular because of the its percieved protection. Even the average person is aware they want AV on their computer (see AOL, Netscape, Netzero, Earthlink, and other ISP television ads). What if: Instead of protecting ppl from hackers AV software was actually making it easier for hackers?

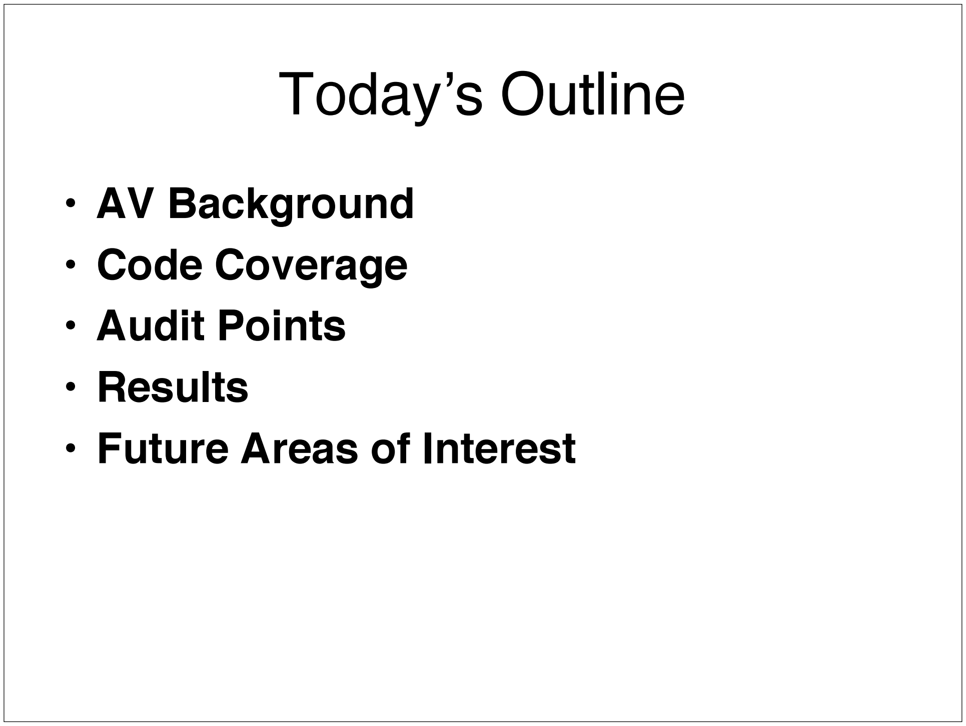
This talk will outline general binary auditing techniques using AV software as an example, and demonstrate examples of remote AV vulnerabilities discovered using those techniques..

Alex Wheeler *is a security researcher, who specializes in reversing engineering binaries for security vulnerabilities. His research experience was cultivated during his time with ISS X-Force, which he spent auditing critical network applications and technologies for security vulnerabilities. Alex's recent audit focus on AV products has lead to the discovery of serious systemic and point vulnerabilities in many major AV products.*

Neel Mehta *works as an application vulnerability researcher at ISS X-Force, and like many other security researchers comes from a reverse-engineering background. His reverse engineering experience was cultivated through extensive consulting work in the copy protection field, and has more recently been focused on application security. Neel has done extensive research into binary and source-code auditing, and has applied this knowledge to find many vulnerabilities in critical and widely deployed network applications.*

Øwning Antivirus

Alex Wheeler (alexbling@gmail.com)
Neel Mehta
(nmehta@iss.net)



Today's Outline

- **AV Background**
- **Code Coverage**
- **Audit Points**
- **Results**
- **Future Areas of Interest**

Why AV? High ROI.

Attractive Attack Surface

- Gateways, Servers, Clients, ISP’s, Third Party Vendor Products
- Heterogeneous and Layered Environments
- Øwning security products is fun

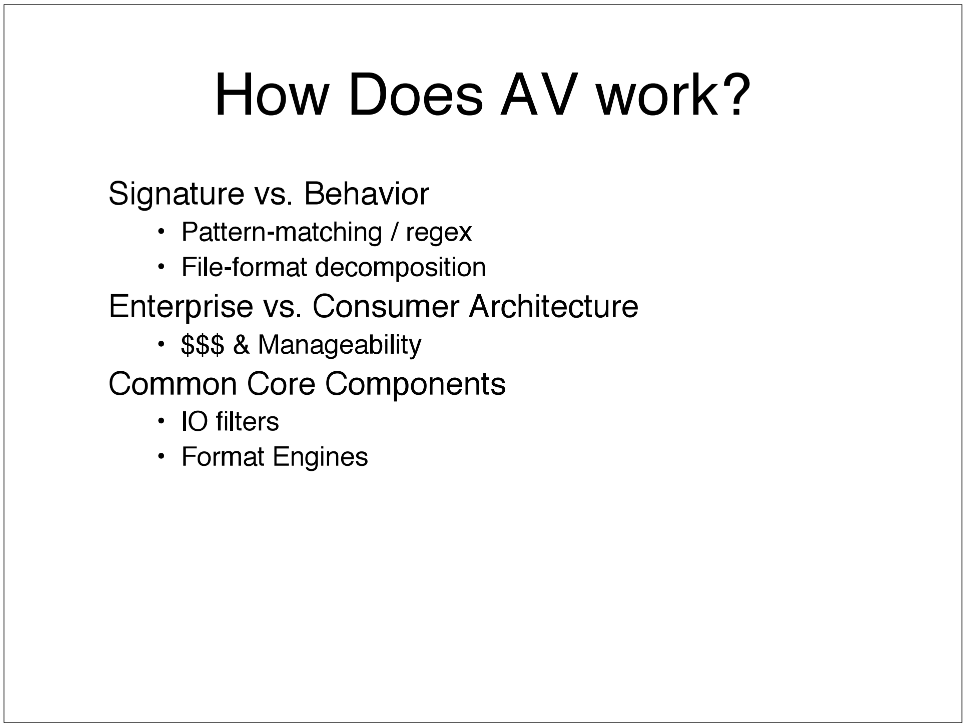
Why AV? High ROI.

Un-trusted Data Processing

- Most functionality is designed around this

Run on a variety of Platforms (targets)

- Windows, Linux, Solaris, Mac



How Does AV work?

Signature vs. Behavior

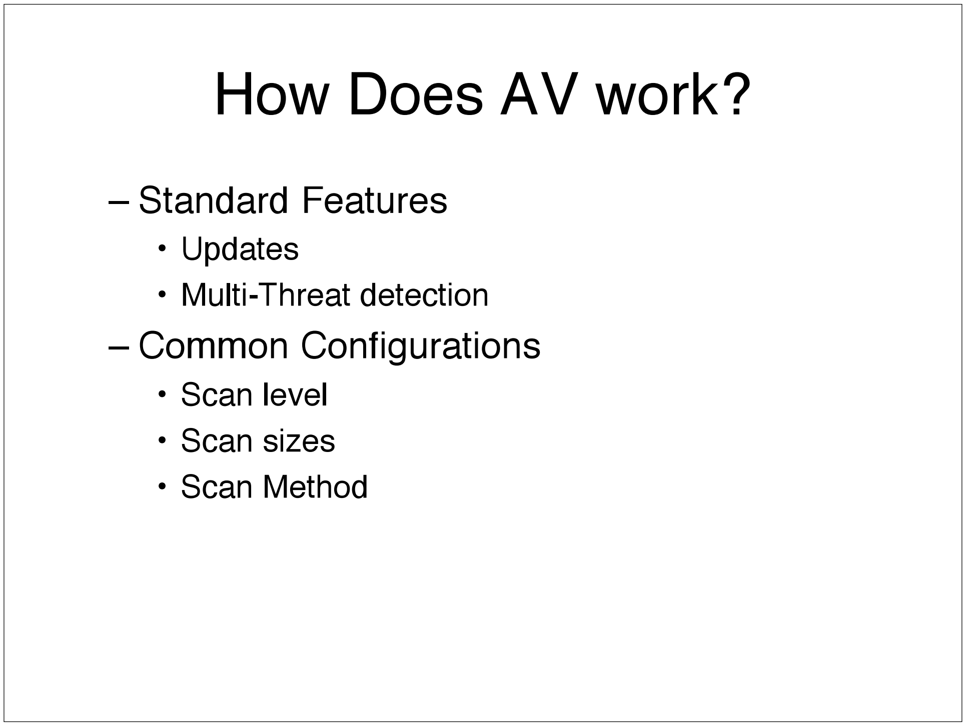
- Pattern-matching / regex
- File-format decomposition

Enterprise vs. Consumer Architecture

- $$$ & Manageability

Common Core Components

- IO filters
- Format Engines



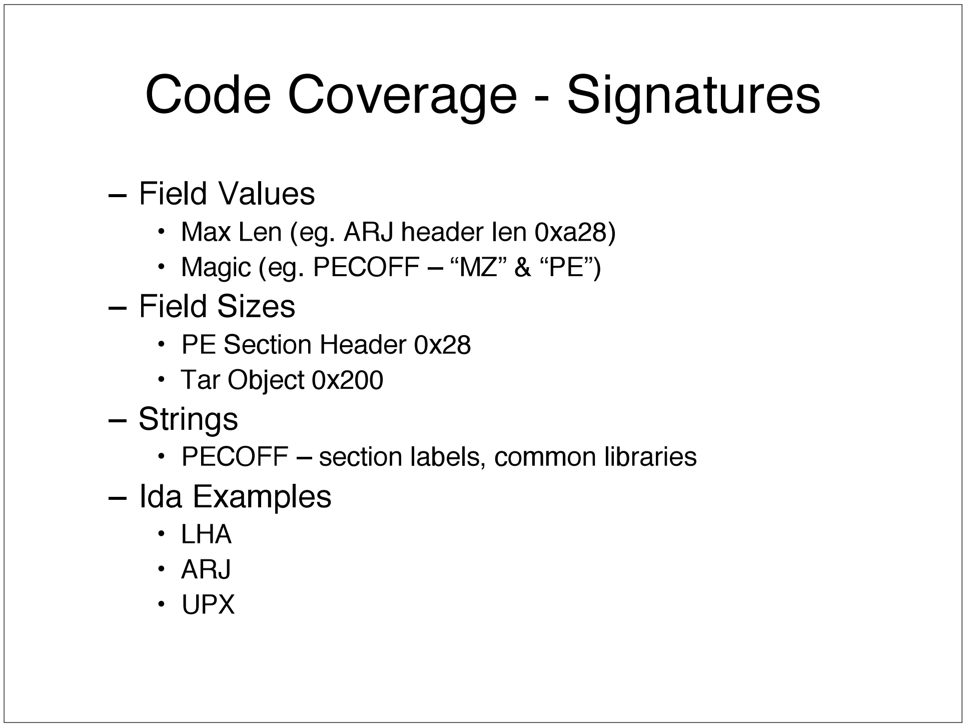
How Does AV work?

– Standard Features

- Updates
- Multi-Threat detection

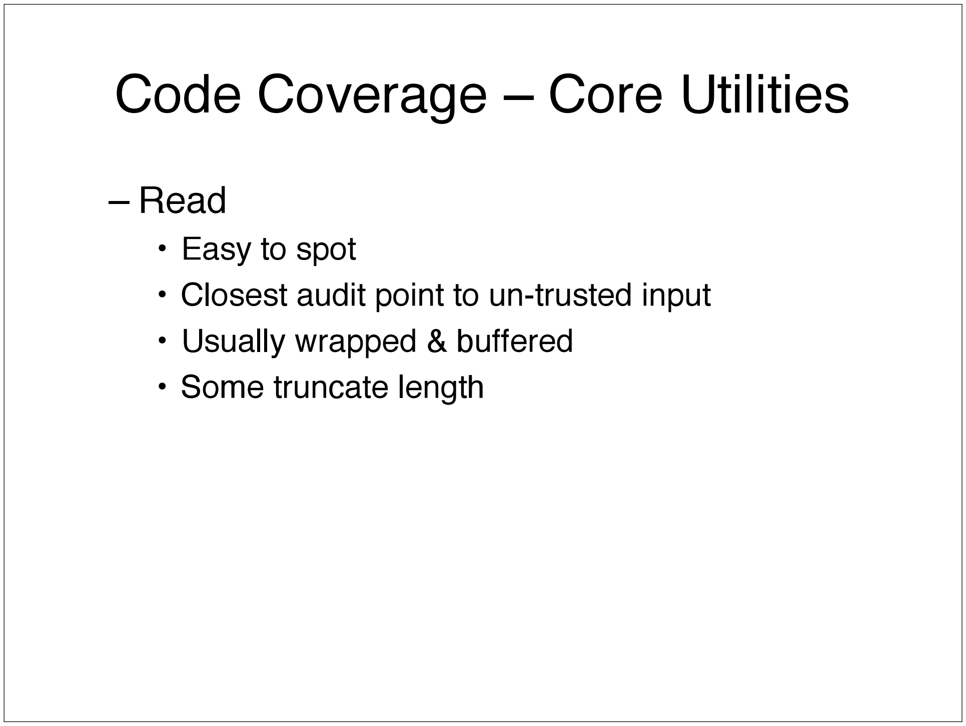
– Common Configurations

- Scan level
- Scan sizes
- Scan Method



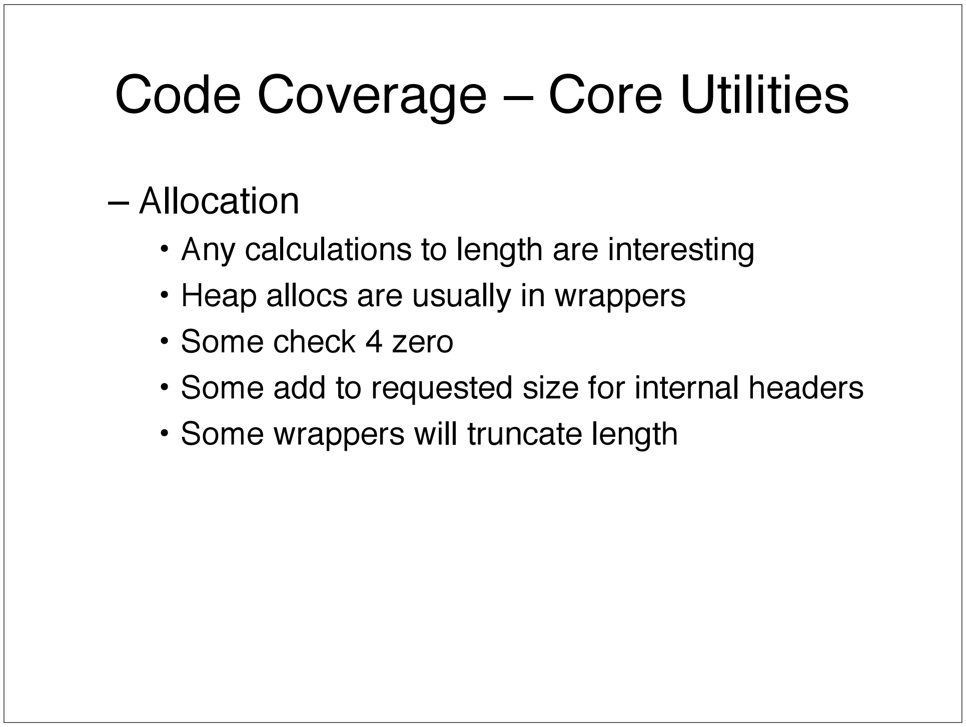
Code Coverage - Signatures

- Field Values
  - Max Len (eg. ARJ header len 0xa28)
  - Magic (eg. PECOFF – “MZ” & “PE”)
- Field Sizes
  - PE Section Header 0x28
  - Tar Object 0x200
- Strings
  - PECOFF – section labels, common libraries
- Ida Examples
  - LHA
  - ARJ
  - UPX



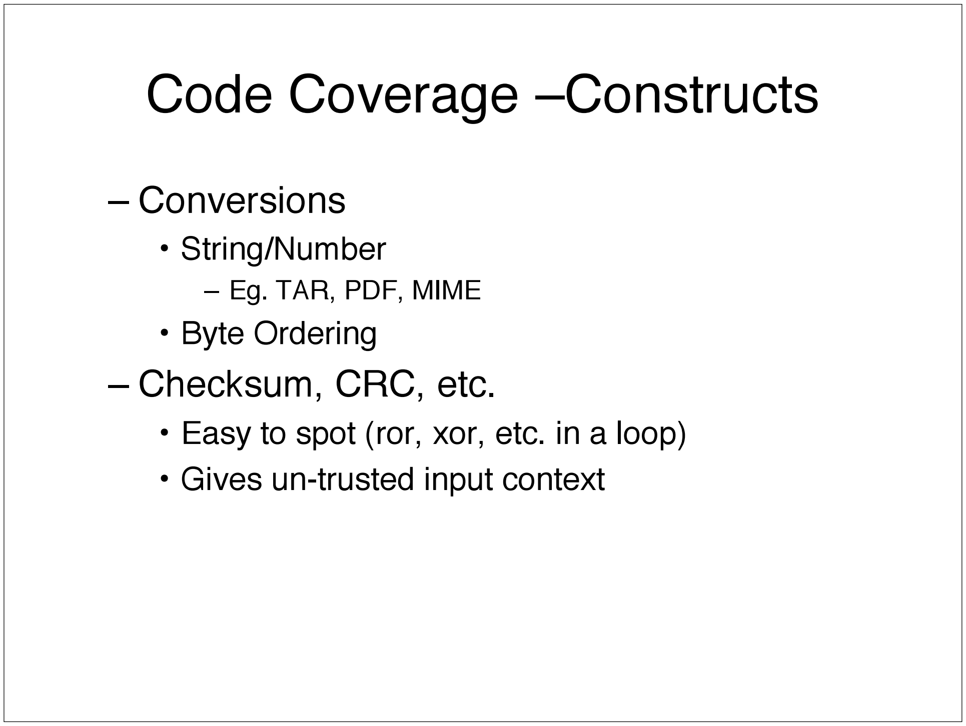
Code Coverage – Core Utilities

- Read
  - Easy to spot
  - Closest audit point to un-trusted input
  - Usually wrapped & buffered
  - Some truncate length



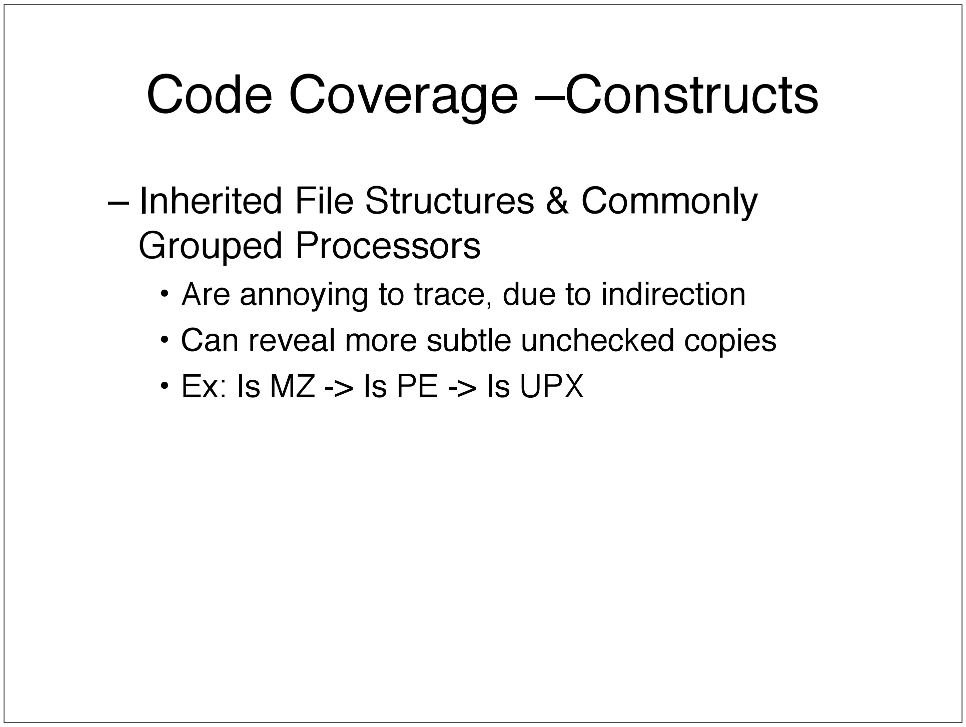
Code Coverage – Core Utilities

- Allocation
  - Any calculations to length are interesting
  - Heap allocs are usually in wrappers
  - Some check 4 zero
  - Some add to requested size for internal headers
  - Some wrappers will truncate length



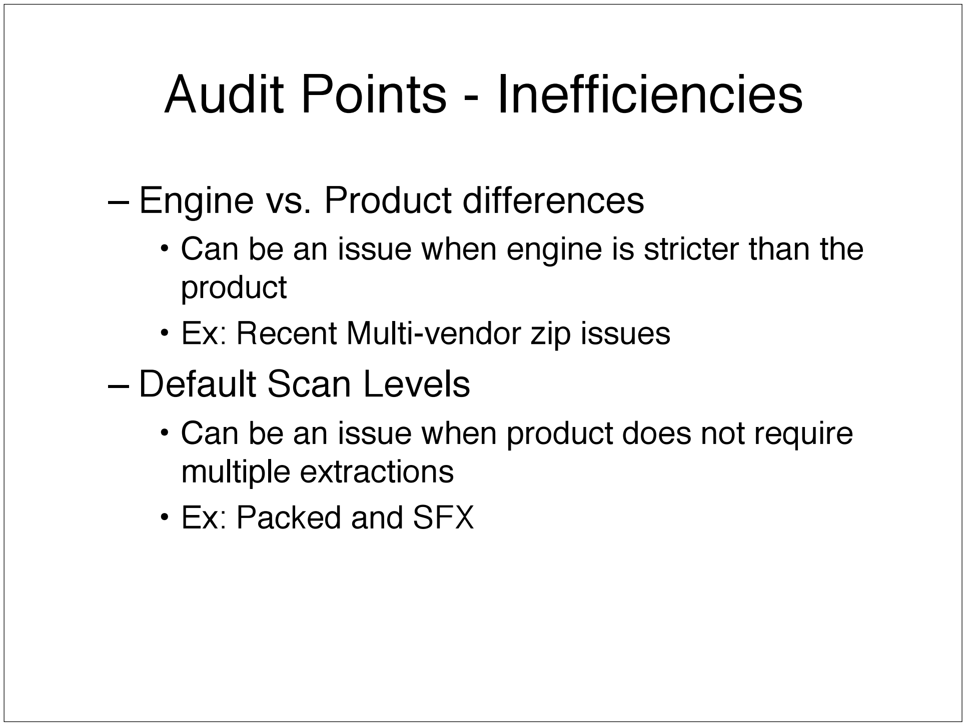
Code Coverage –Constructs

- Conversions
  - String/Number
    - Eg. TAR, PDF, MIME
  - Byte Ordering
- Checksum, CRC, etc.
  - Easy to spot (ror, xor, etc. in a loop)
  - Gives un-trusted input context



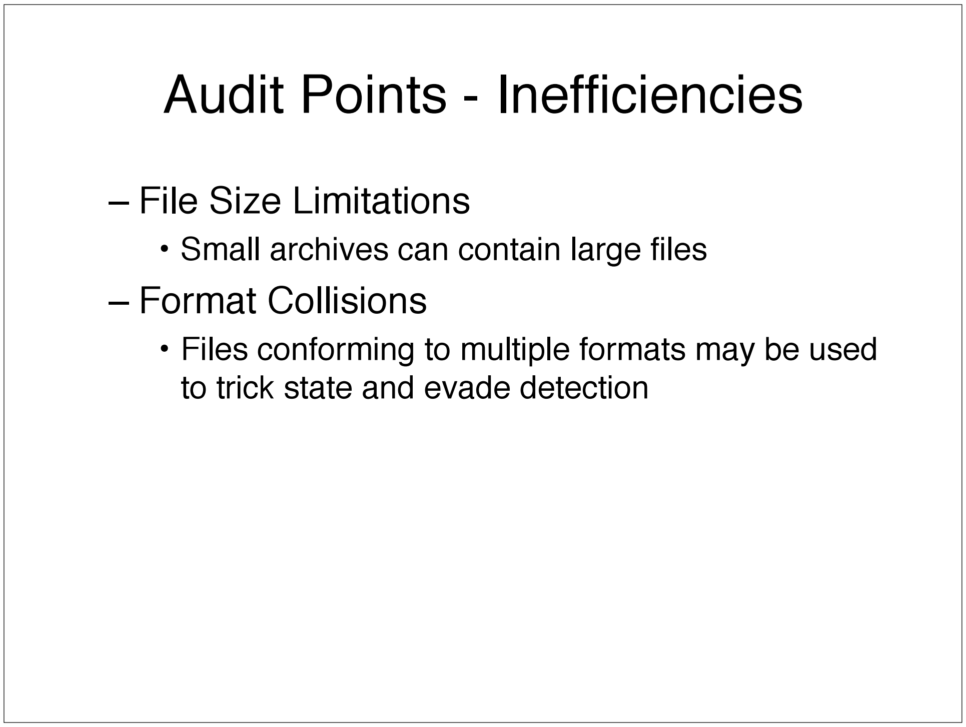
Code Coverage –Constructs

- – Inherited File Structures & Commonly Grouped Processors
  - Are annoying to trace, due to indirection
  - Can reveal more subtle unchecked copies
  - Ex: Is MZ -> Is PE -> Is UPX



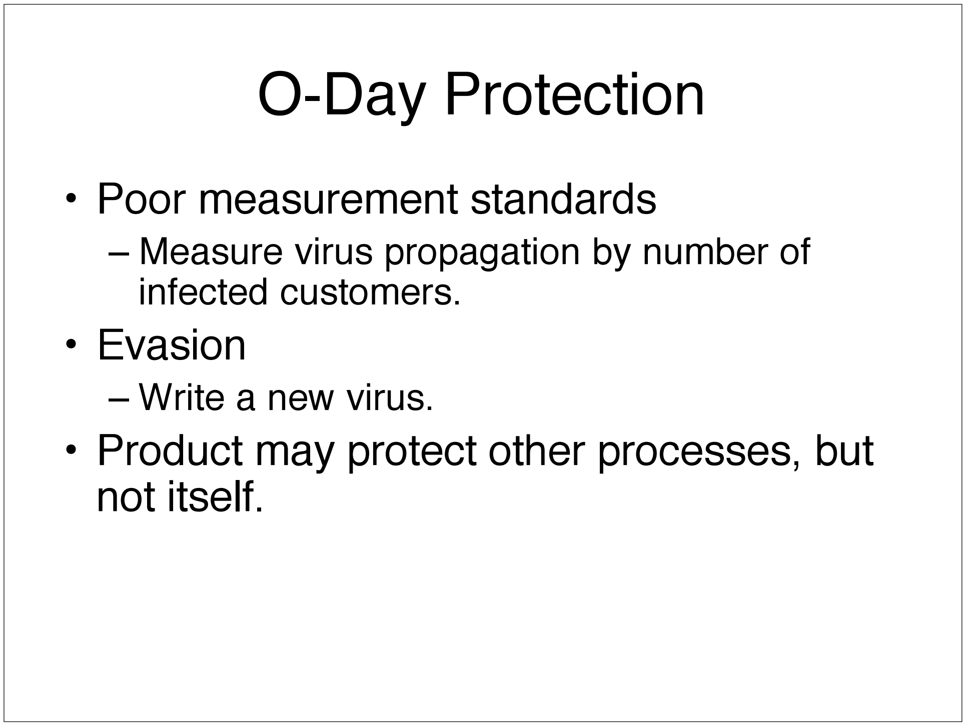
Audit Points - Inefficiencies

- – Engine vs. Product differences
  - Can be an issue when engine is stricter than the product
  - Ex: Recent Multi-vendor zip issues
- – Default Scan Levels
  - Can be an issue when product does not require multiple extractions
  - Ex: Packed and SFX



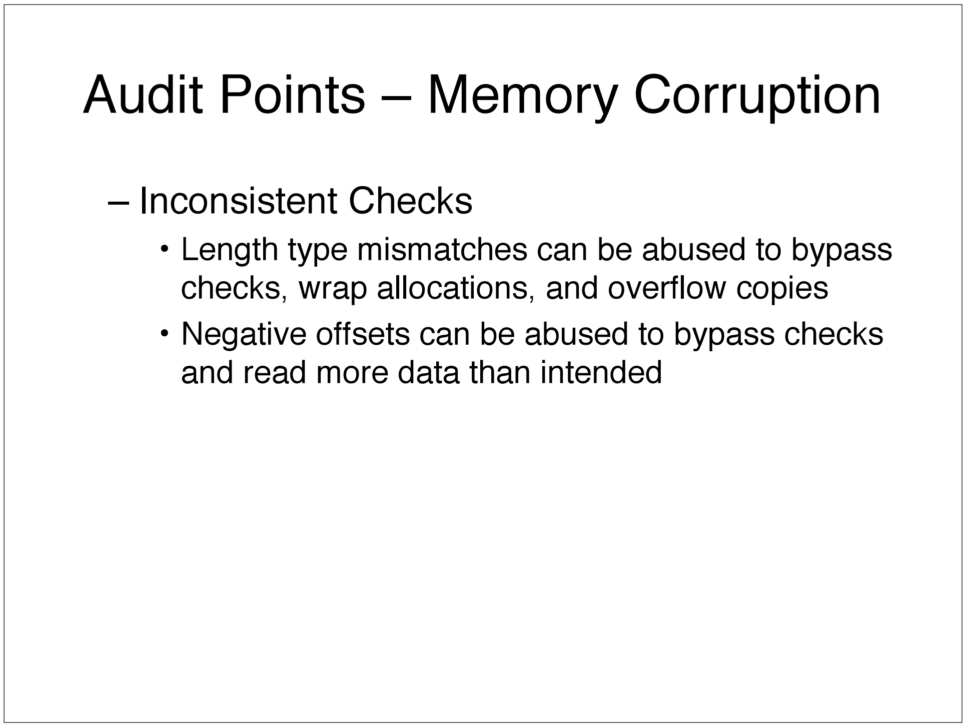
Audit Points - Inefficiencies

- – File Size Limitations
  - Small archives can contain large files
- – Format Collisions
  - Files conforming to multiple formats may be used to trick state and evade detection



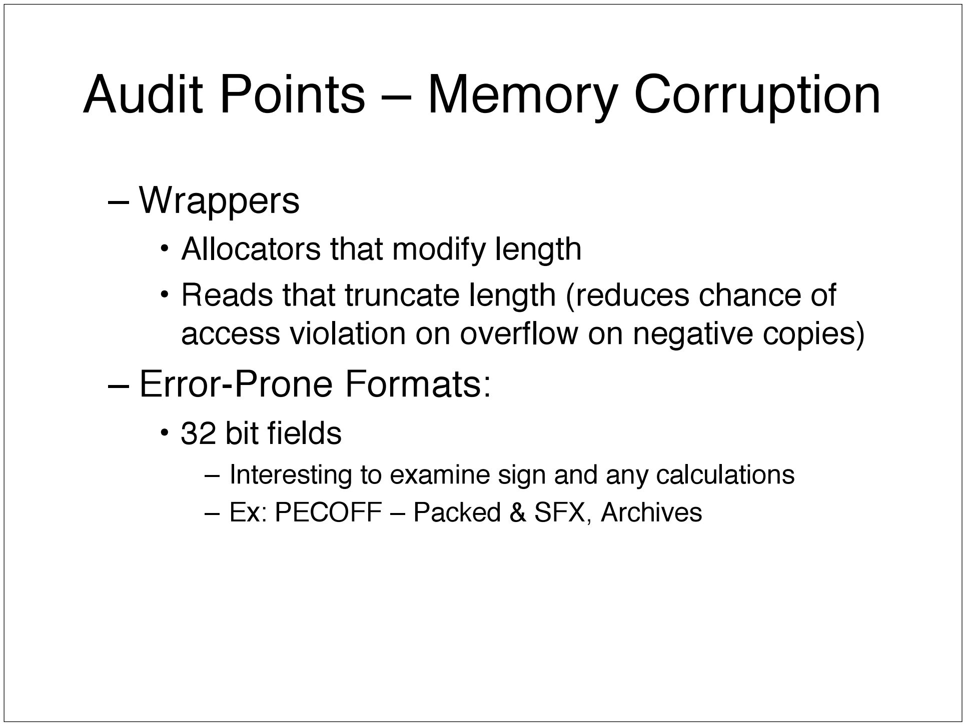
O-Day Protection

- Poor measurement standards
  - – Measure virus propagation by number of infected customers.
- Evasion
  - – Write a new virus.
- Product may protect other processes, but not itself.



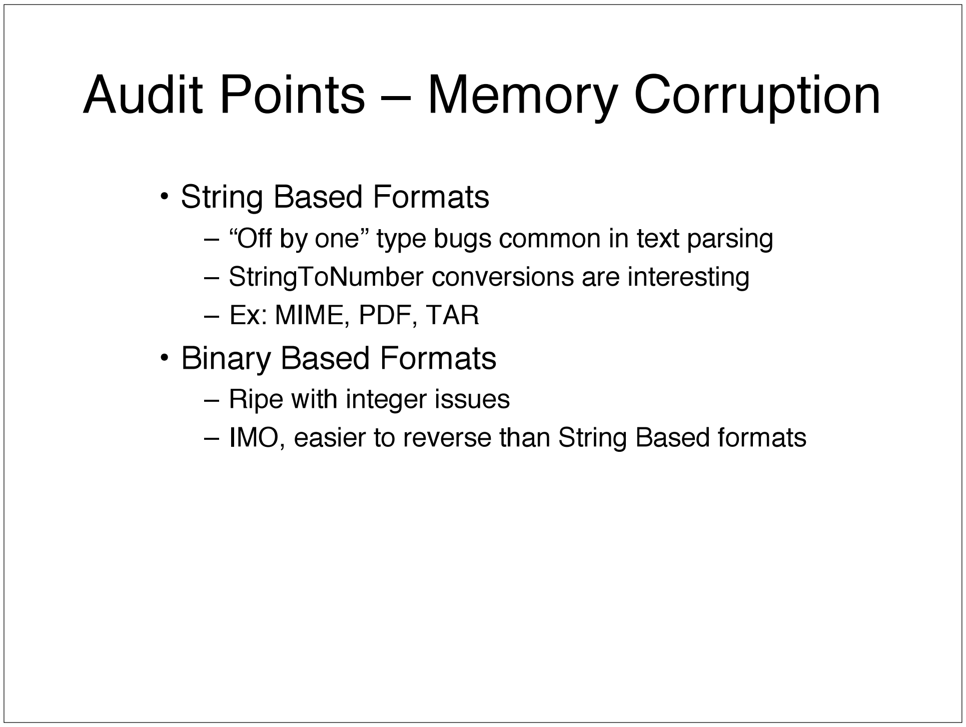
Audit Points – Memory Corruption

- Inconsistent Checks
  - Length type mismatches can be abused to bypass checks, wrap allocations, and overflow copies
  - Negative offsets can be abused to bypass checks and read more data than intended



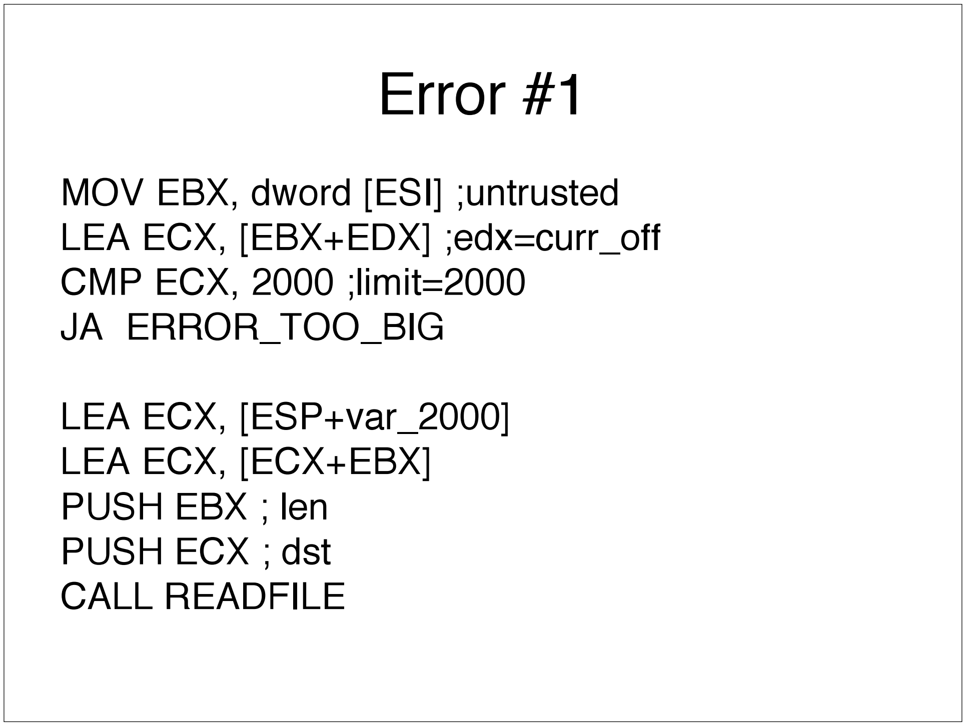
Audit Points – Memory Corruption

- Wrappers
  - Allocators that modify length
  - Reads that truncate length (reduces chance of access violation on overflow on negative copies)
- Error-Prone Formats:
  - 32 bit fields
    - Interesting to examine sign and any calculations
    - Ex: PECOFF – Packed & SFX, Archives



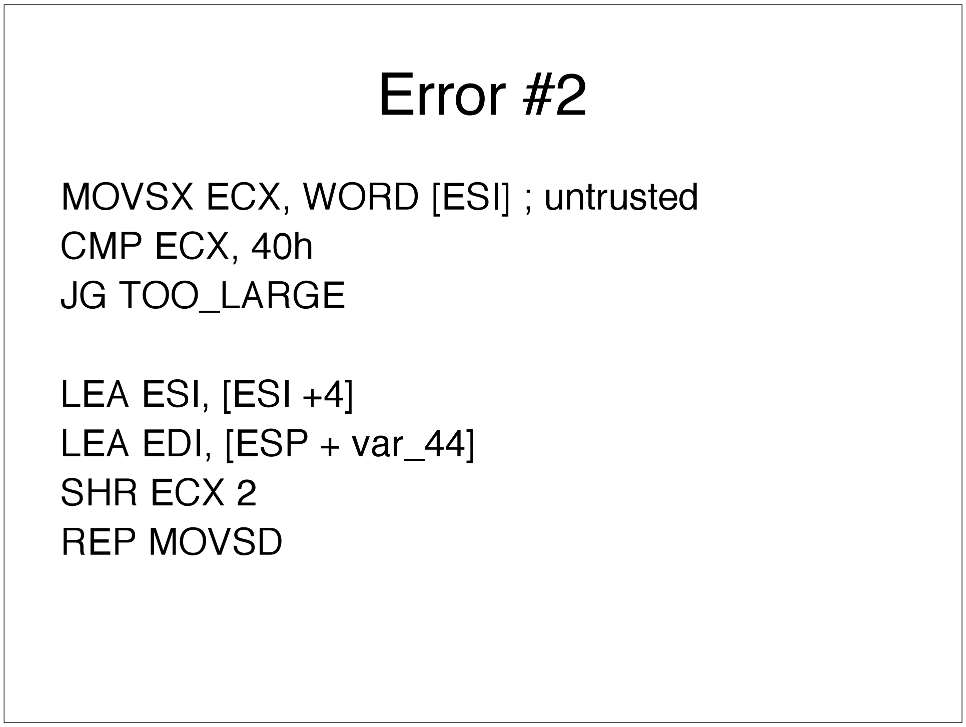
Audit Points – Memory Corruption

- String Based Formats
  - “Off by one” type bugs common in text parsing
  - StringToNumber conversions are interesting
  - Ex: MIME, PDF, TAR
- Binary Based Formats
  - Ripe with integer issues
  - IMO, easier to reverse than String Based formats



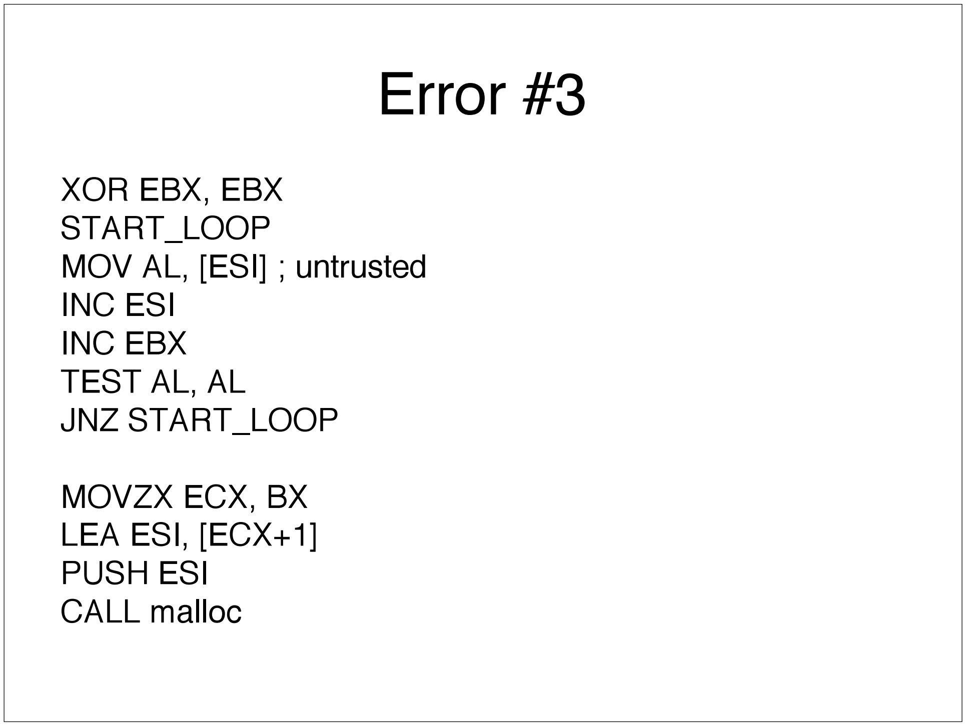
Error #1

```
MOV EBX, dword [ESI] ;untrusted
LEA ECX, [EBX+EDX] ;edx=curr_off
CMP ECX, 2000 ;limit=2000
JA  ERROR_TOO_BIG

LEA ECX, [ESP+var_2000]
LEA ECX, [ECX+EBX]
PUSH EBX ; len
PUSH ECX ; dst
CALL READFILE
```



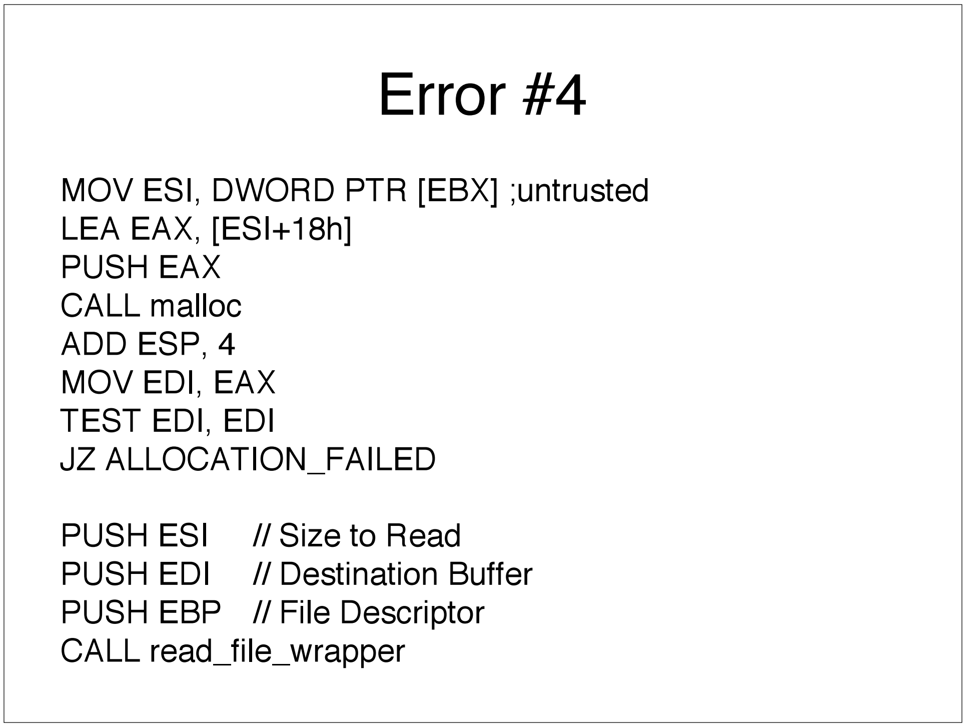
Error #2

```
MOVSX ECX, WORD [ESI] ; untrusted
CMP ECX, 40h
JG TOO_LARGE

LEA ESI, [ESI +4]
LEA EDI, [ESP + var_44]
SHR ECX 2
REP MOVSD
```



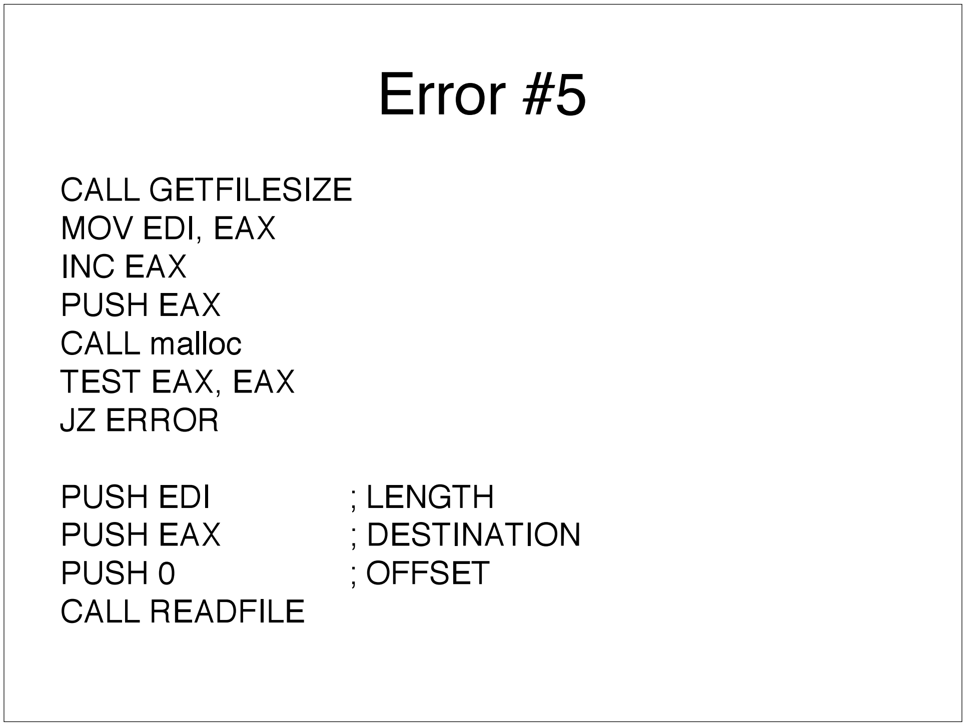
Error #3

```
XOR EBX, EBX
START_LOOP
MOV AL, [ESI] ; untrusted
INC ESI
INC EBX
TEST AL, AL
JNZ START_LOOP

MOVZX ECX, BX
LEA ESI, [ECX+1]
PUSH ESI
CALL malloc
```



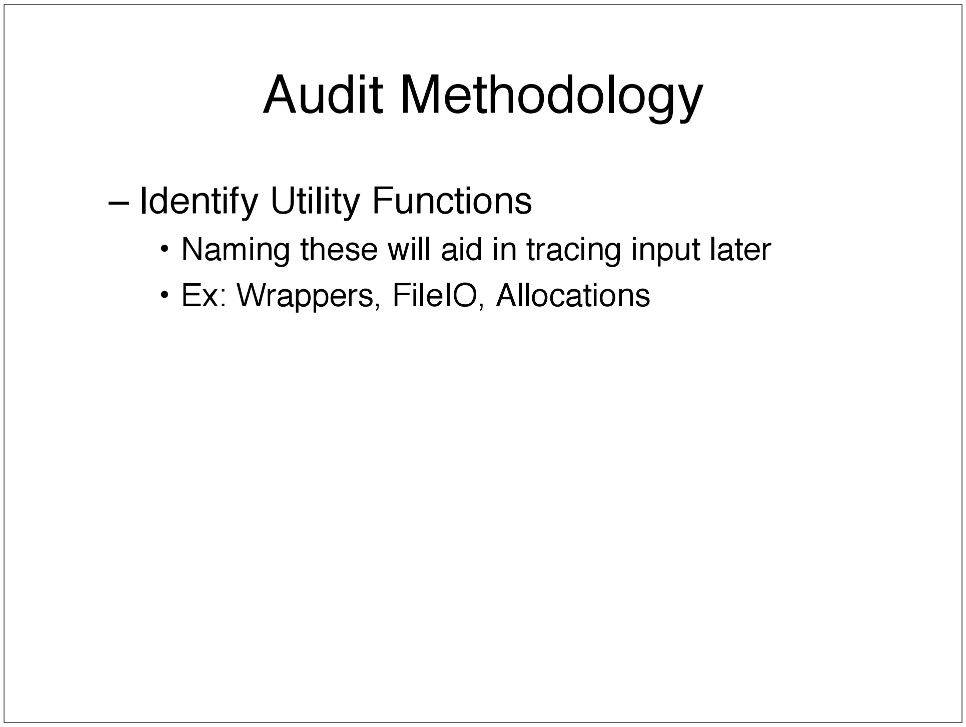
Error #4

```
MOV ESI, DWORD PTR [EBX] ;untrusted
LEA EAX, [ESI+18h]
PUSH EAX
CALL malloc
ADD ESP, 4
MOV EDI, EAX
TEST EDI, EDI
JZ ALLOCATION_FAILED

PUSH ESI    // Size to Read
PUSH EDI    // Destination Buffer
PUSH EBP    // File Descriptor
CALL read_file_wrapper
```



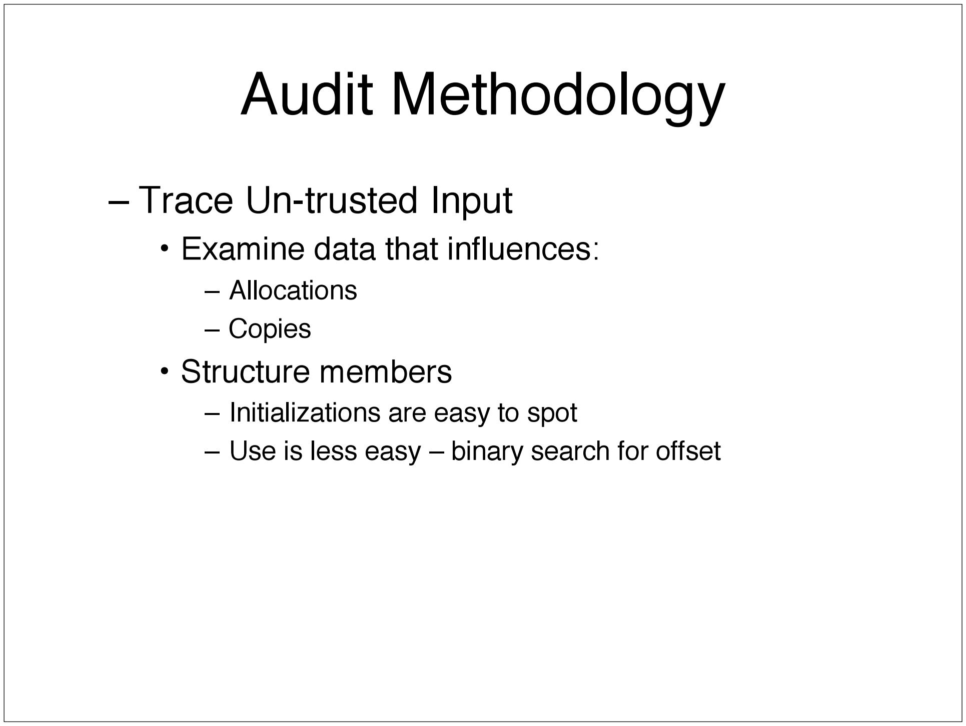
Error #5

```
CALL GETFILESIZE
MOV EDI, EAX
INC EAX
PUSH EAX
CALL malloc
TEST EAX, EAX
JZ ERROR

PUSH EDI        ; LENGTH
PUSH EAX        ; DESTINATION
PUSH 0          ; OFFSET
CALL READFILE
```



Audit Methodology

- Identify Utility Functions
  - Naming these will aid in tracing input later
  - Ex: Wrappers, FileIO, Allocations



Audit Methodology

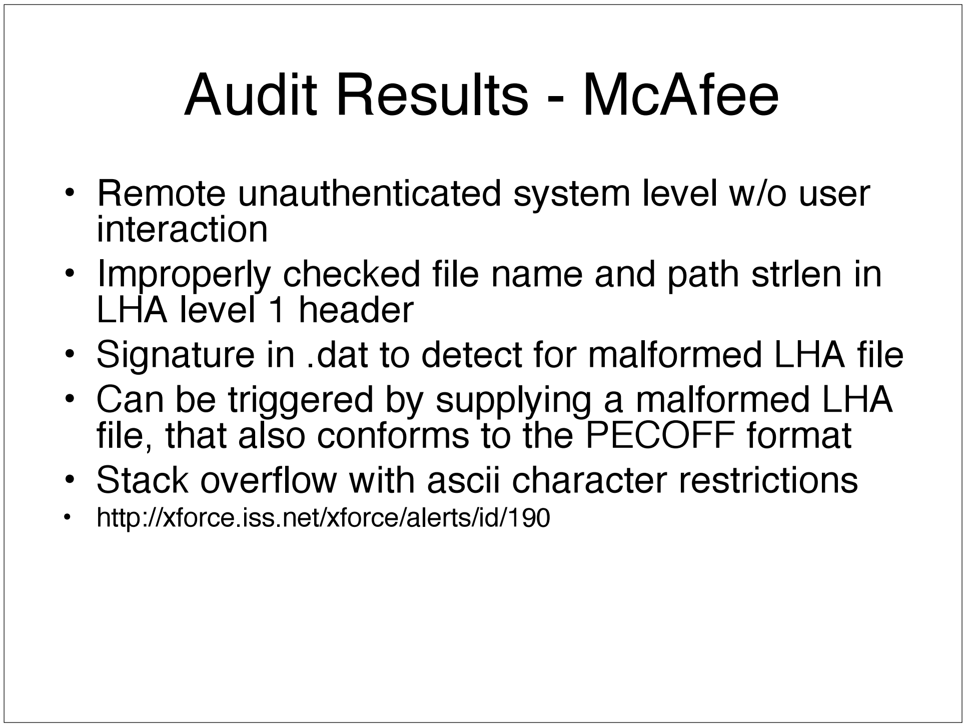
- Trace Un-trusted Input
  - Examine data that influences:
    - Allocations
    - Copies
  - Structure members
    - Initializations are easy to spot
    - Use is less easy – binary search for offset

Audit Methodology

- Reverse File Format Processors
  - Track class member offsets and sizes
  - Will reveal more subtle bugs

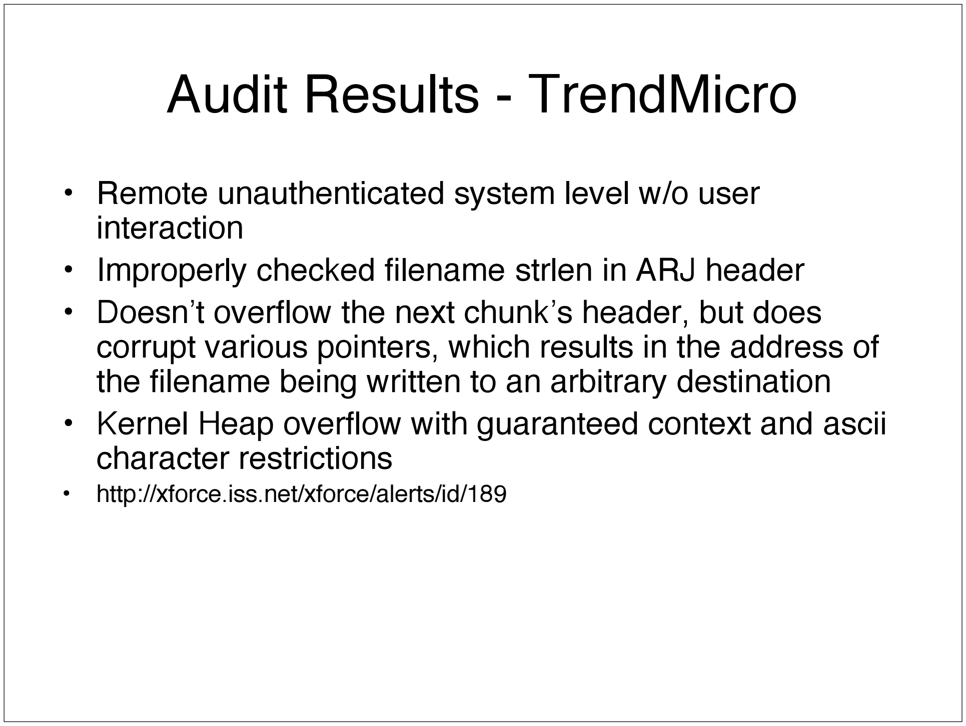
Audit Results - Symantec

- Remote unauthenticated system level w/o user interaction
- Unchecked offset reconstructing UPX PE header
- Can be triggered by providing a negative offset to prior heap chunk containing MZ header with crafted PE header
- Heap overflow with no character restrictions
- http://xforce.iss.net/xforce/alerts/id/187



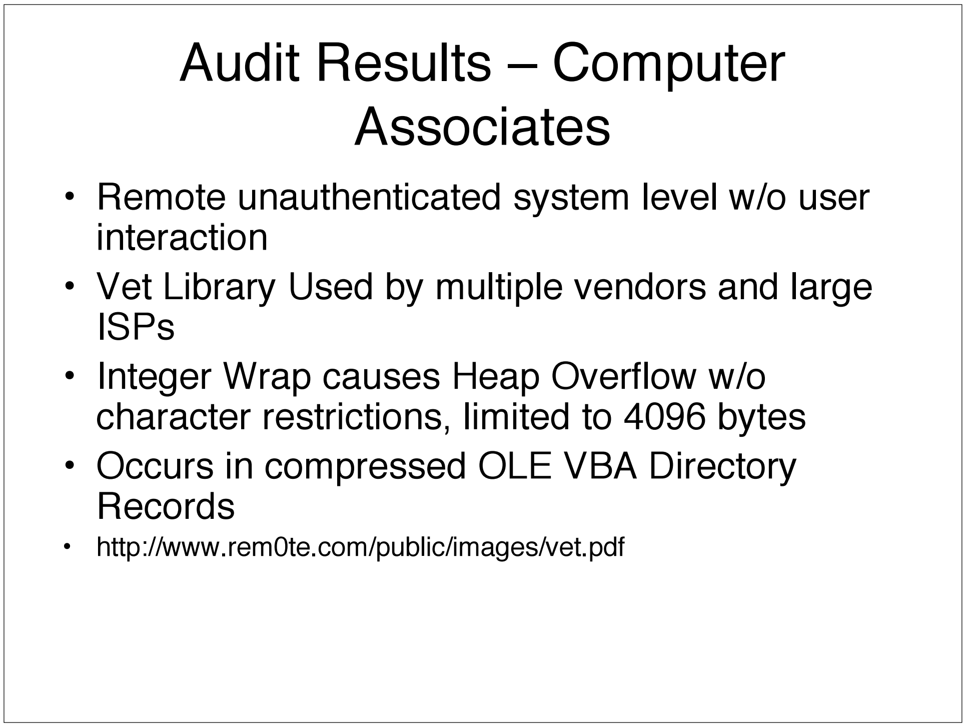
Audit Results - McAfee

- Remote unauthenticated system level w/o user interaction
- Improperly checked file name and path strlen in LHA level 1 header
- Signature in .dat to detect for malformed LHA file
- Can be triggered by supplying a malformed LHA file, that also conforms to the PECOFF format
- Stack overflow with ascii character restrictions
- http://xforce.iss.net/xforce/alerts/id/190



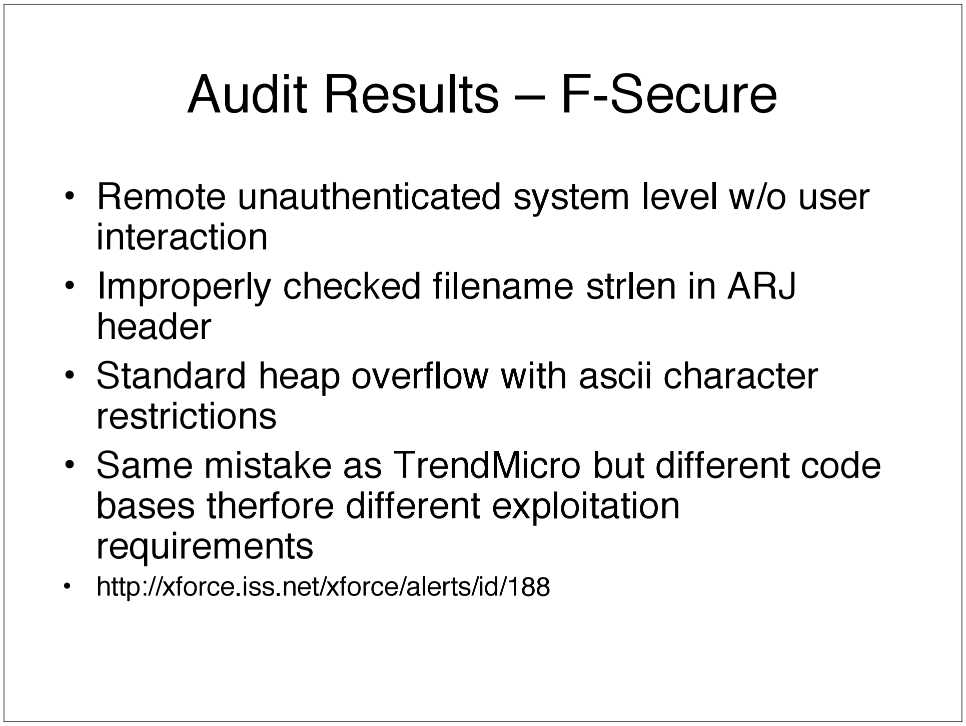
Audit Results - TrendMicro

- Remote unauthenticated system level w/o user interaction
- Improperly checked filename strlen in ARJ header
- Doesn’t overflow the next chunk’s header, but does corrupt various pointers, which results in the address of the filename being written to an arbitrary destination
- Kernel Heap overflow with guaranteed context and ascii character restrictions
- http://xforce.iss.net/xforce/alerts/id/189



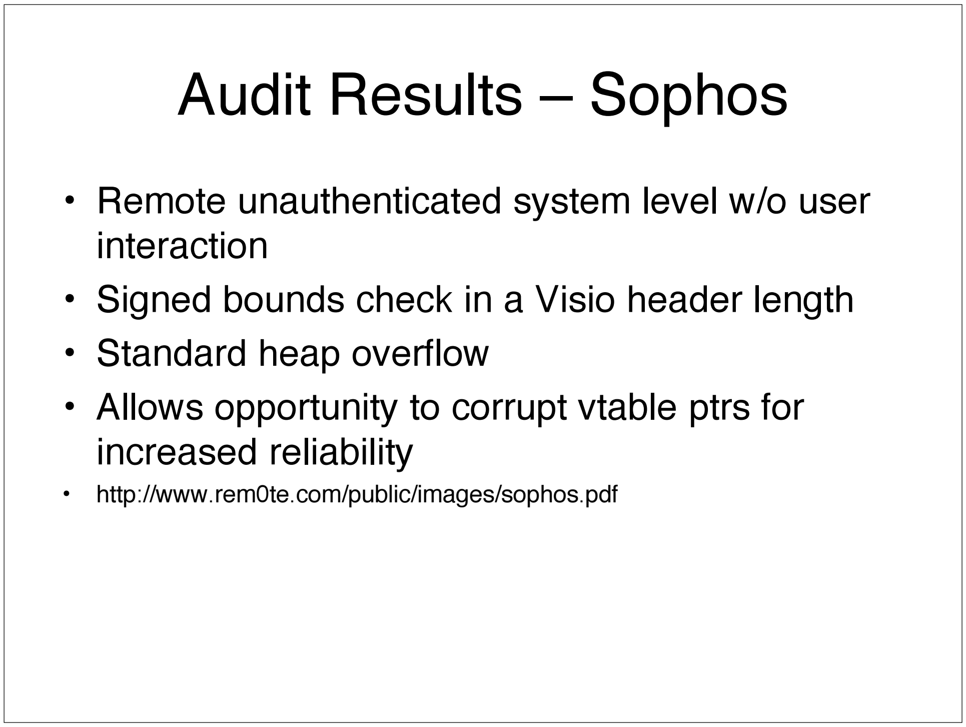
Audit Results – Computer Associates

- Remote unauthenticated system level w/o user interaction
- Vet Library Used by multiple vendors and large ISPs
- Integer Wrap causes Heap Overflow w/o character restrictions, limited to 4096 bytes
- Occurs in compressed OLE VBA Directory Records
- http://www.rem0te.com/public/images/vet.pdf



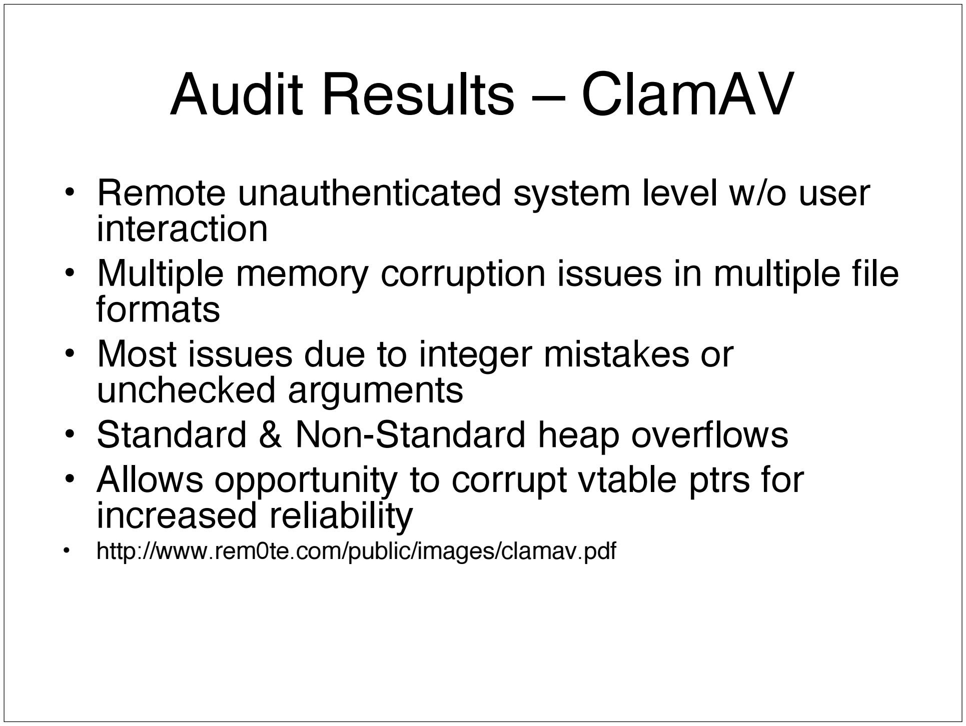
Audit Results – F-Secure

- Remote unauthenticated system level w/o user interaction
- Improperly checked filename strlen in ARJ header
- Standard heap overflow with ascii character restrictions
- Same mistake as TrendMicro but different code bases therfore different exploitation requirements
- http://xforce.iss.net/xforce/alerts/id/188



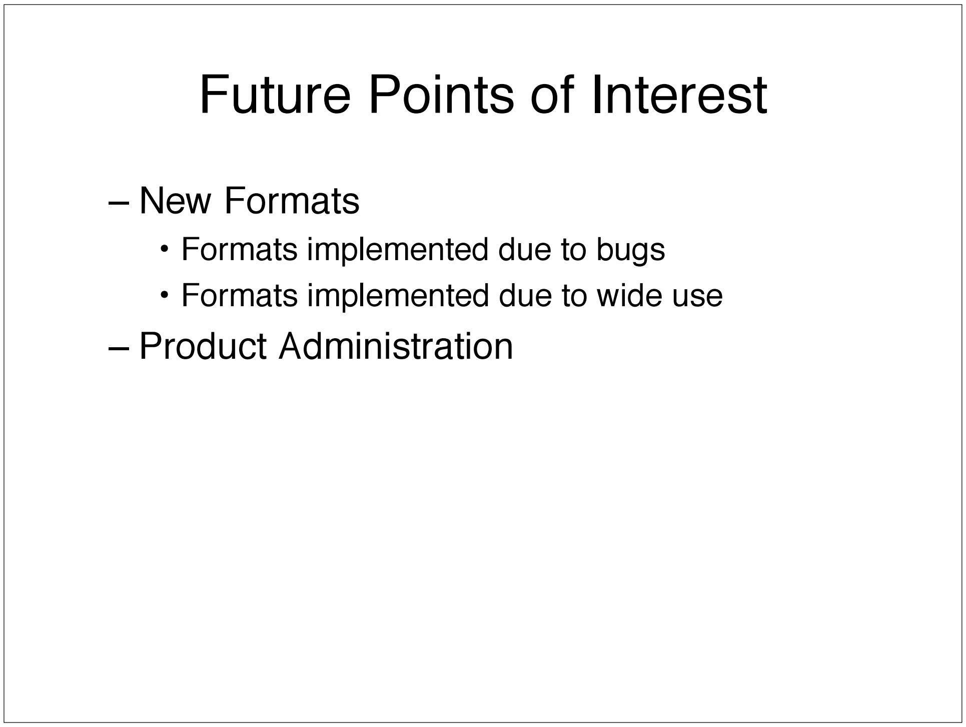
Audit Results – Sophos

- Remote unauthenticated system level w/o user interaction
- Signed bounds check in a Visio header length
- Standard heap overflow
- Allows opportunity to corrupt vtable ptrs for increased reliability
- http://www.rem0te.com/public/images/sophos.pdf



Audit Results – ClamAV

- Remote unauthenticated system level w/o user interaction
- Multiple memory corruption issues in multiple file formats
- Most issues due to integer mistakes or unchecked arguments
- Standard & Non-Standard heap overflows
- Allows opportunity to corrupt vtable ptrs for increased reliability
- http://www.rem0te.com/public/images/clamav.pdf



Future Points of Interest

- – New Formats
  - Formats implemented due to bugs
  - Formats implemented due to wide use
- – Product Administration

Questions?

Alex Wheeler
(alexbling@gmail.com)

Neel Mehta
(nmehta@iss.net)