Hands-On Network Security: Practical Tools & Methods

Security Training Course

Dr. Charles J. Antonelli
The University of Michigan
2012

Hands-On Network Security

Module 7

Intrusion Detection

Topics

- Fundamentals
- Network IDS
  - Snort
- Host-based IDS
  - Tripwire

Fundamentals

Intrusion Detection

- Location
  - Network-based (NIDS)
  - Host-based (HIDS)
- Action
  - Detection
    - Only alerts
  - Prevention
    - Performs some reactive action
    - IPS (NIDS + prevention)
    - HIPS (HIDS + prevention)

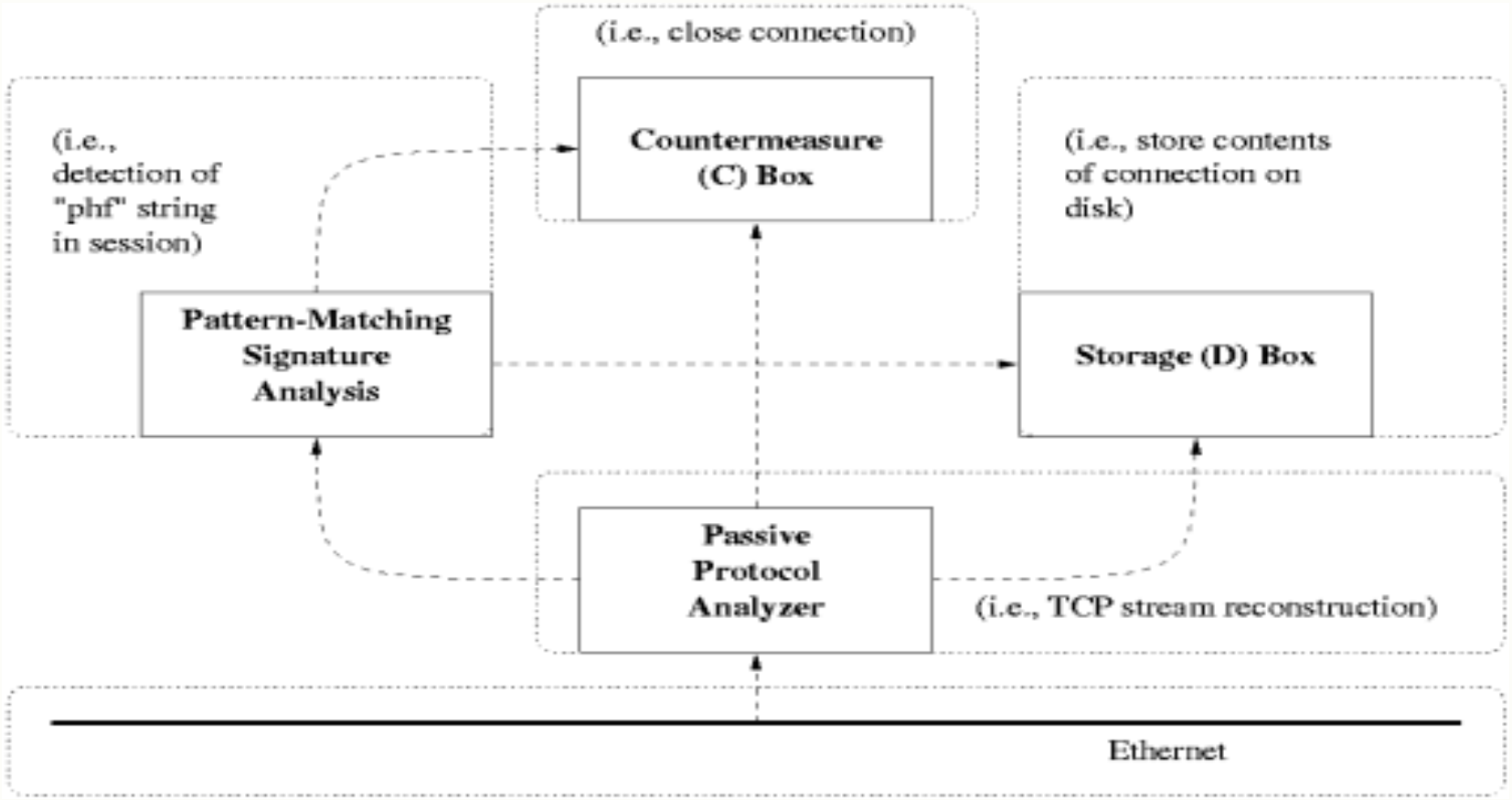
NIDS

- Change the way you think about NIDS
  - By itself, no direct impact on C.I.A.
    - ▼you do!
- IDS is a piece of network security monitoring
- *Intrusion* is a misnomer
  - Detects network traffic that has some property of an attack
  - IDS thinks in these units, so should you when thinking about IDS

Traditional CIDF model

- Event (E) box
  - Collect data
    - sniff packets from the wire
    - OS shim (HIDS)
- Analysis (A) box
  - analyze data from E box
- Countermeasure (C) box
  - prevention, blocking
- Data storage (D) box
  - alerting mechanism, log storage

Traditional CIDF model



from "***Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection***" **Ptacek & Newsham**

Sniffing

- E box
- Passive
  - Sniffs network packets
    - “smart” tcpdump (or Ethereal)
    - No cost to the network
- Sniffing modes
  - Span port
  - Tap

Sniffing

- Usually in conjunction with some operating system or hardware tweaks
  - fast BPF
    - no copies from kernel -> userland
- In 2010 ~1+ GB/s continues to be the limit
- Traffic mangling hacks
  - Aggregate
  - Separate 1GB+
    - Hash network input to a bank of IDS

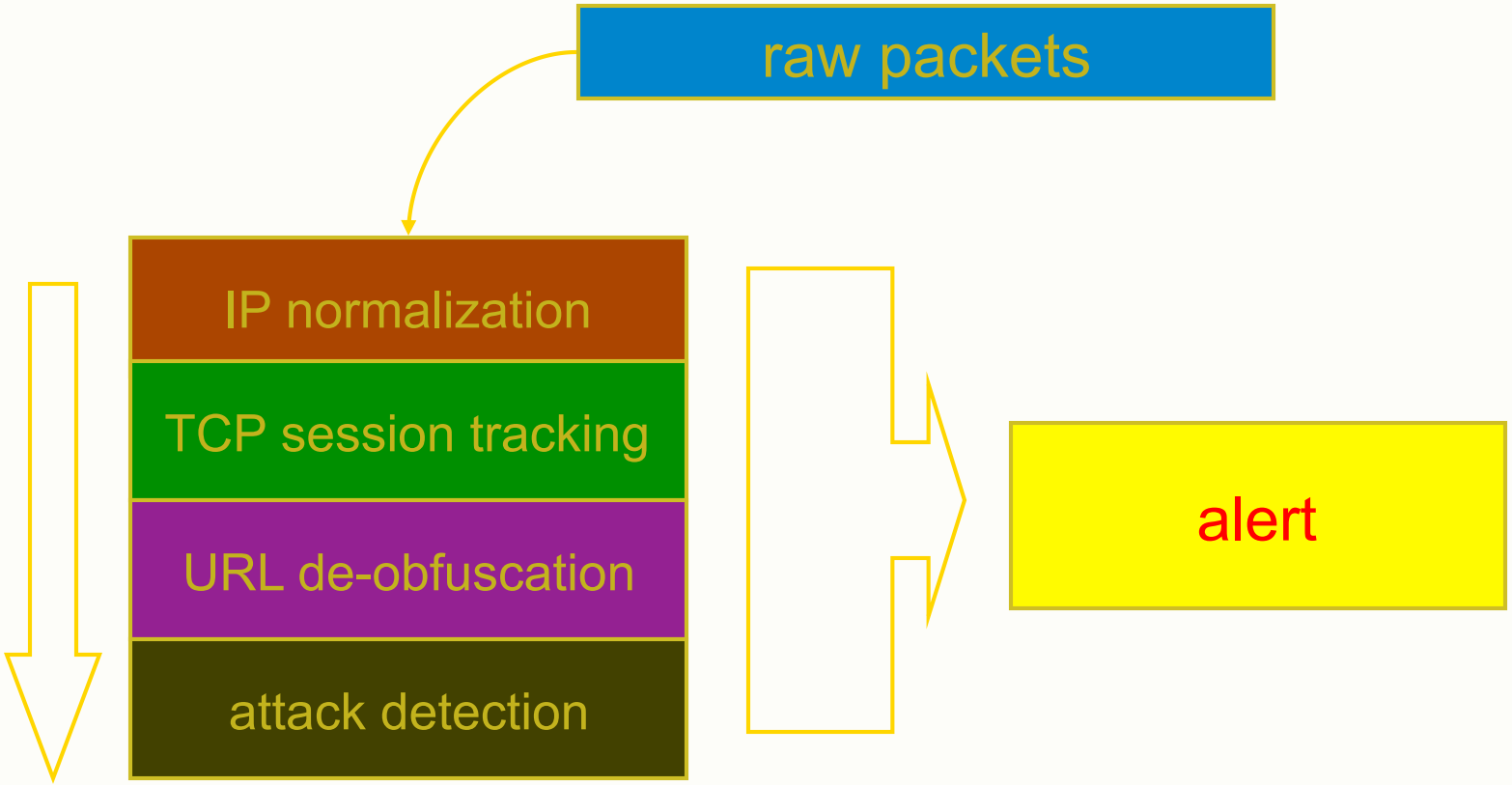
Sniffer Placement

- In front of a firewall
  - More information
    - Too much?
  - Do you care about the anklebiters?
- Behind a firewall
  - Less information
    - More useful?
- Both?
  - You got that kind of time?
- Rule of thumb:
  - Closest to asset you're trying to protect

Analysis

- *The goal of the NIDS is to surmise what the end host will process at each network layer and look for some indication of intrusion*
- A box
  - This is where the magic happens
- Session tracking at each network layer passed up the stack
  - MAC addresses
    - usually ignored
  - IP defragmention
  - TCP session reassembly
  - Application layer deobfuscation

Layered detection



Signature based rules

- Statically analyze network traffic for known intrusions
- For instance (look familiar?)

```
GET /awstats?configdir=|cmd
```

Signature based rules

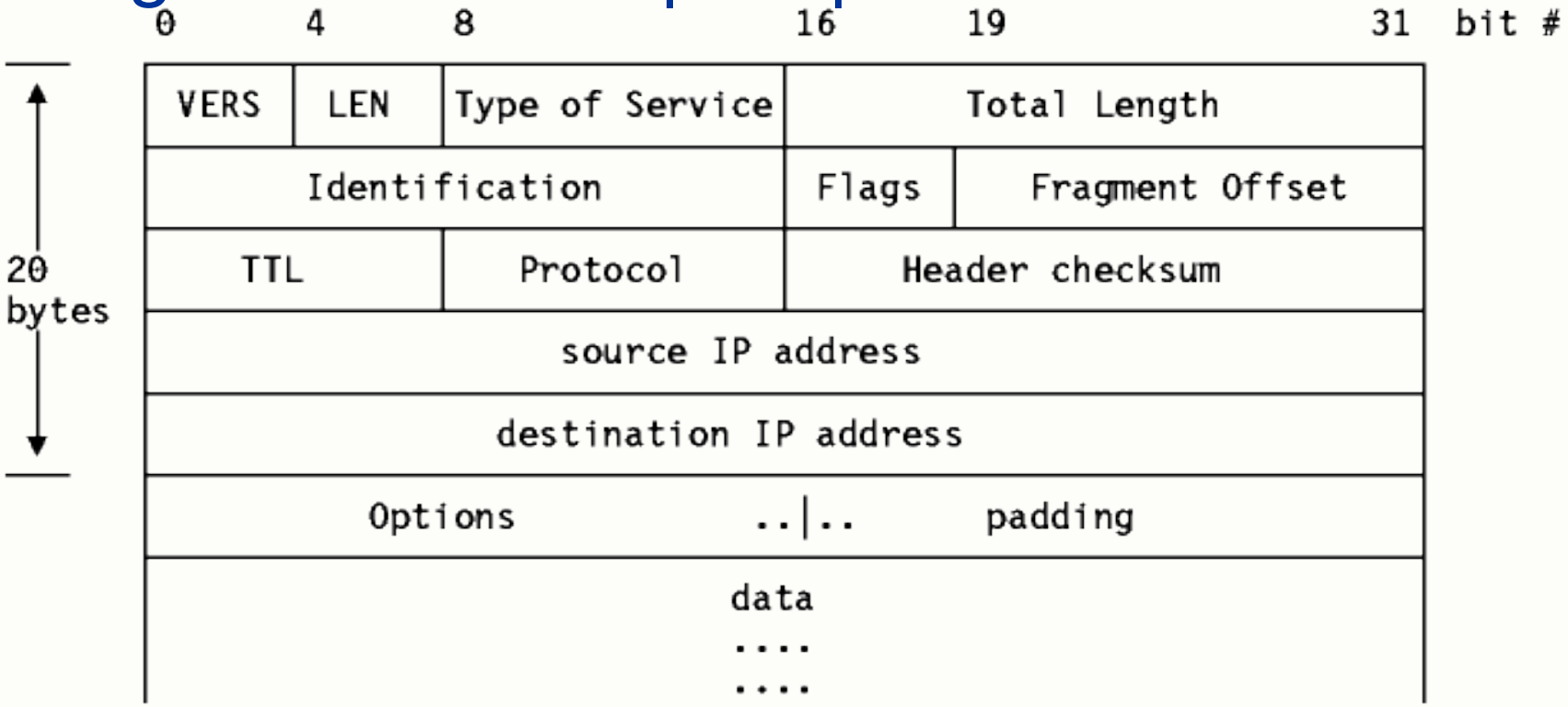
- A (pseudocode) rule for this might be:

```
if (url contains “awstats?configdir=|”)
   alert()
```

  - Doesn’t matter where the awstats binary is located on the web server
  - Don’t care what the command is, just that the first character of the value is a pipe
  - Looking for the vulnerability, not the exploit

IP Fragmentation

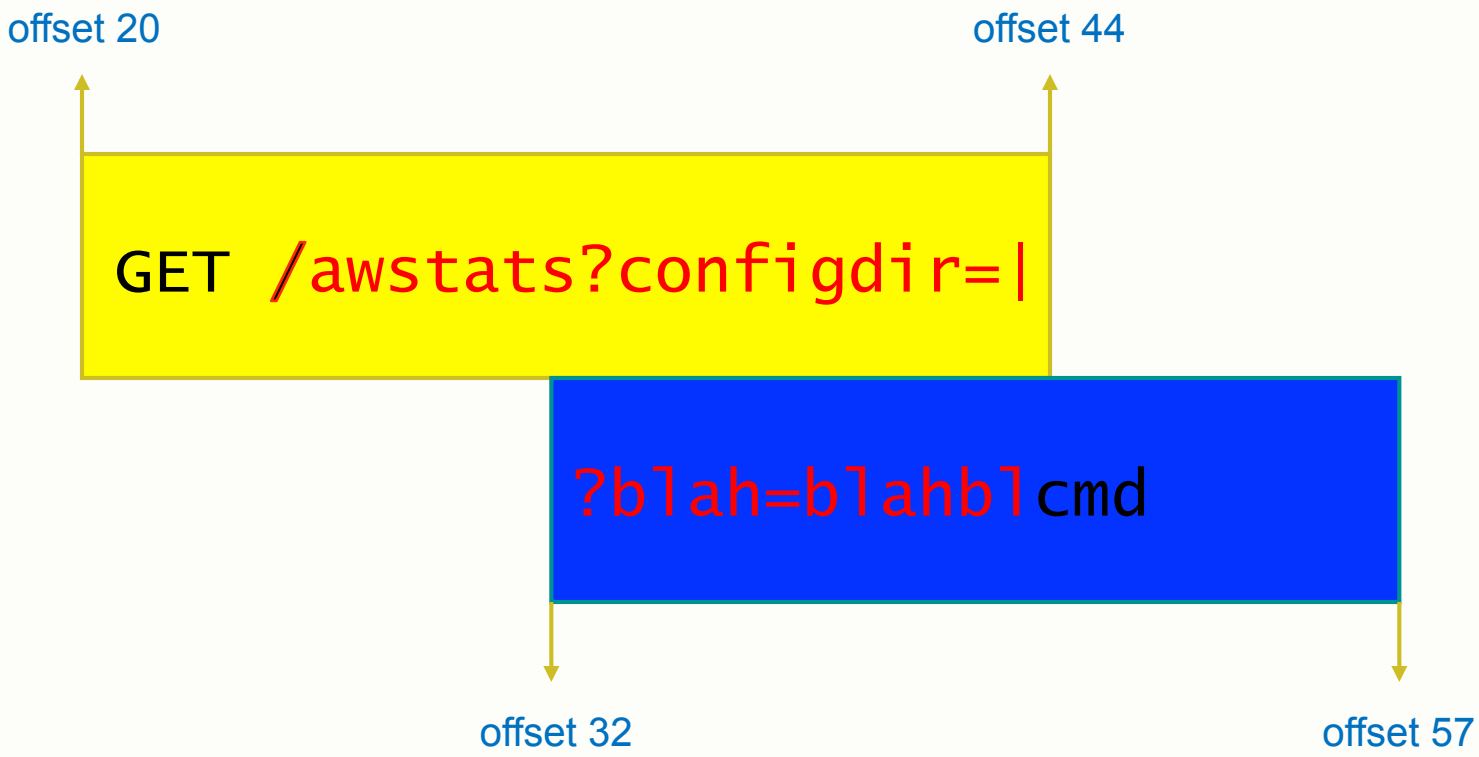
Flags = Reserved | DF | MF



IP fragmentation

- IP packet divided into chunks if some MTU over the traversed route is too small
- End host reassembles packets
  - NIDS must also reassemble packet
- IP protocol allows overlapping fragments
  - Different OSs reassemble fragments differently
  - How will the NIDS know whether to favor new or old data?

2 overlapping IP fragments



Does the NIDS use “?configdir=|” or “?blah=blahbl”?
(offset is 20 because of the prepended TCP header)

Overlapping IP fragments

- A few options
  - Alert on tiny fragments
    - Attacker can use bigger fragments
  - Reassemble both ways
    - Slow, can lead to DoS condition
  - Passively fingerprint the end-host
    - Can make an educated guess which way it will reassemble
  - Alert on overlapping fragments

Other network games

- Out-of-order packets
  - NIDS has to cache packets until reassembled
    - How long?
- Old packets
- Overlap TCP segments
  - Same concept as IP fragmentation
- Low TTL games
- See Ptacek & Newsham paper
  - Dugsong's fragroute for an implementation

Other network games

- Most network ambiguities are solved
  - Reasonably permissive TCP/IP stack
    - aggressive timeouts to avoid DoS
  - Do not accept data until ACKed
  - Alert on any obvious anomalies
  - UDP remains a problem
    - connection-less

Application layer

- Quoth the RFCs: be liberal in what you accept and strict with what you send
- Sometimes too liberal, especially in a hostile environment
  - URL obfuscation
  - Telnet escape codes (in FTP too)
  - (MS)RPC fragmentation
  - DNS compression
  - etc etc etc
- Ever read an RFC?
  - Vendors haven't

```
$ wc -l rfc*
 6267750
```

Detection

- Majority of vendors
  - Heuristics register interest in sessions (TCP) or types of packets (UDP, IP, et al) or application protocols (http, rpc, dns, et al)
  - Dispatcher iterates over ruleset or executes pseudocode
    - Application level parsing if applicable
  - Alert if evaluation for intrusion passes

Detection

- IDS typically alerts on
  - Attacks
    - well-formed intrusion attempts
    - DoS
  - Probes
    - portscans, hostscans
  - Anomalies
    - packet floods
    - bizarre protocol behavior (more later)
  - Policy violations
    - RFC 1918 addresses
    - p2p traffic

Detection

- Common methodologies
  - I saw *XYZ*
  - I saw *A XYZ*s in *B* seconds
  - I didn't see *XYZ* where I expected to

Signature vs anomaly

- Signature
  - Does this network traffic match a known, well-formed pattern of a particular attack?
    - GET /awstats?configdir=|cmd
    - Indicative of a particular attack method or the actual vulnerability?
  - Writing good signature rules is an art

Signature vs anomaly

- Anomaly detection
  - Does this seem “wrong”?
    - Suspect number of SYNs
    - Really long URL

```
GET /blah.ida?
   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA[Ax240]=x
   HTTP/1.1
```

    - “Protocol lint”
    - “Wrong” is “right”
      - Or at least acceptable

Alerts

- Alerts stored
  - Correlated
  - Throttled
- Analyst mines alert data
  - Response
- Easiest way to render a NIDS useless is to flood the alerting mechanism (D box)
  - Admin misses the real attack in the avalanche of alerts

Alerts

- CVE – Common Vulnerabilities Exposure
  - http://cve.mitre.org
  - Dictionary, not a database
  - Common names for vulnerabilities
    - Alerts linked with CVE entries
    - Solves the AV naming problem

```
CVE-2005-0116

Description

   AWStats 6.1, and other versions before 6.3, allows remote
   attackers to execute arbitrary commands via shell
   metacharacters in the configdir parameter.
```

Countermeasures

Countermeasures

- Manual
  - Block with firewall/router filter rules
- Automated
  - TCP RSTs / UDP port unreachable
    - Race condition
  - Inline blocking
    - More on this later
- All of these are temporary!
  - Buy time to do proper investigation

Snort

- Free
  - Wait 30 days for the free rules
- Excellent way to cut your teeth
- Rule based rather than a language
  - One line per rule
  - Syntax supported by most vendors
  - Official rules at http://www.snort.org/vrt/
  - User contributed rules
    - http://www.emergingthreats.net/
- Excellent documentation at http://www.snort.org/docs/

Snort

- Preprocessors
  - Handles things rules can't
    - TCP state machine
      - » portscan and hostscan
    - URL deobfuscation
    - Interfaces with rules
- Alerts
  - Output in a variety of formats
  - ASCII, syslog, database, OPSEC, etc
- Lots of open source add-ons
  - SGUIL console

Snort components

- http://www.snort.org/downloads
  - Sources, Binaries
- http://www.snort.org/docs
  - How to write Snort rules and run Snort
- http://www.snort.org/snort-rules
  - VRT Certified Rules
    - Offical Snort Ruleset
  - Subscription required for immediate access
  - Registration required for 30-day delayed access

Snort rules

- http://www.snort.org/login
  - Register
- http://www.snort.org/snort-rules
  - Get VRT rules for *registered* users
- Or install sample snort rules from class server

Lab: Get and install Snort

- Snort and rulesets are available on the course web server
- To install:
  1. `cd /usr/local/lab`
  2. `sudo wget http://www.umich.edu/~cja/HNS12/supp/snort.tgz`
  3. `sudo tar zxf snort.tgz`
  4. `cd snort`
  5. `./INSTALL.sh`
  6. `sudo vi /etc/sysconfig/snort`
     - » Set INTERFACE to your correct network interface

Lab: Run Snort

- Popular arguments

```
-v      output headers to console
-d      output packet data too
-e      output layer 2 header too
-l d    log packets to directory d
-h a    home network is a
-b      tcpdump log to single file
-c c    config file c (nids mode)
-r f    read packets from file f
-i i    read packets from interface i
```

Lab: Run Snort

1. As a sniffer

   ```
   snort -v [-d -e] [-i dev]
   ```

2. As a logger

   ```
   snort [-v] [-d -e] [-b|-h] -l ./log [-i dev]
   ```

3. As a NIDS

   ```
   snort [-v] [-d -e] [-i dev]
     -c /etc/snort/snort.conf -A console -k none
   ```

   Try a trigger: browse to

   ```
   http://www.umich.edu/~cja/HNS12/awstats.pl?
   ```

Lab: Run Snort

- Notes
  - –k none ignores packet checksum errors

Snort rules 101

- header (options)
- Header
  - `alert tcp any any -> any any`
    - action
    - protocol
    - source address/port
    - “->”
    - destination address/port

Snort rules 101

- Options
  - Where the actual processing is performed
  - msg: “this is an alert”; sid:1000; rev:1; flow: to_server,established;
    - msg – alert string
    - sid – Snort ID, unique per alert
    - rev – revision of the rule
    - flow - only matches data sent from the initiator of the established TCP session

Snort rules 101

- Payload options
  - content – matches a string in the packet
    - `content: "USER root";`
    - "|" delimits binary data
      - `content: "|00 00 00|";`
    - `nocase;` modifier
  - uricontent – just like content, but is deobfuscated
    - `uricontent: "evil";`
    - matches "GET /%65vil"

Snort rules 101

- Payload options
  - pcre – match Perl regular expression
    - `pcre: "/joe[^\r\n]\*cool/Ui";`
      - » Pattern modifiers:
        - ~ U: ungreedy match
        - ~ i: ignore case
  - Multiple payload options in one rule
    - Implicit logical "and"

Snort rules 101

- Let's look at a few rules

Lab: Snort rules

- Write a rule to detect the awstats exploit
  - Attack is a string of the form

    http://localhost/cgi-bin/awstats.pl?configdir=|blah
  - Put your rule in `rules/local.rules`
  - Look at other rules for examples

Official Snort rule

- <>

Some problems with Snort

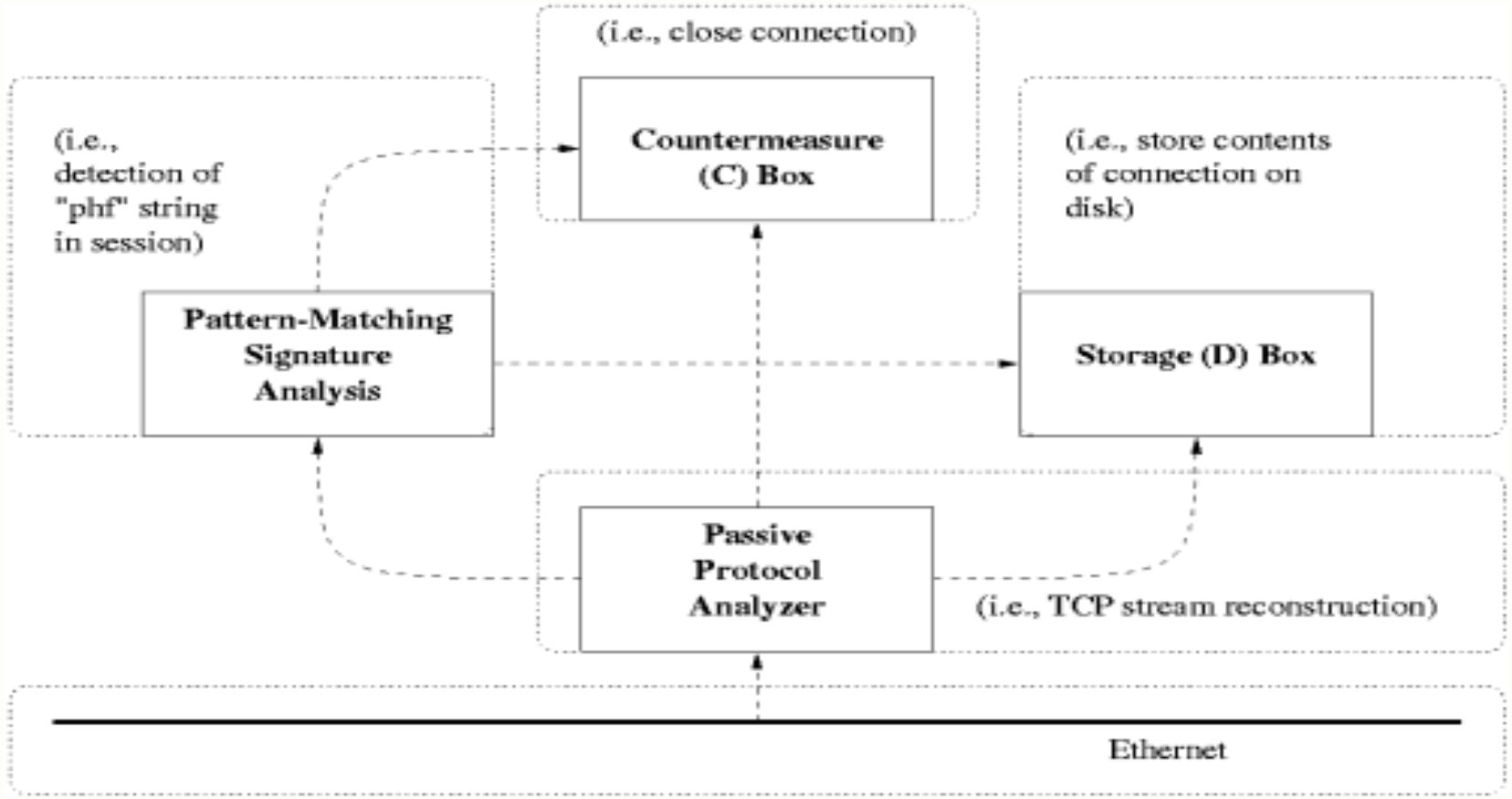
- Without a preprocessor
  - No way to say "I didn't see *XYZ*"
  - No way to correlate non-adjacent events
- Rule syntax is not Turing complete
  - You could argue this, but it still doesn't "feel" like a language
- 0day rules aren't free anymore

IDS Issues

Accepts arbitrary input

- Like any other complex system that accepts input, NIDS are subject to attack
  - Classic programming errors
    - BlackIce – March 2004
      - » Witty worm not so funny
    - Snort has had numerous remotely exploitable attacks

Attacking the CIDF model



from "***Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection***" **Ptacek & Newsham**

False alarms

- False positives
  - Fallacy rate
    - Test is 99.9% accurate
    - 10,000 evaluations means 10 false positives
- False negatives
  - Just plain misses stuff

Tuning

- We are constantly being attacked
- An untuned IDS is not worth running
- ***An untuned IDS is not worth running***
  - Only you know your environment
  - Background noise
    - Old worms
    - Wide-scale probes
  - Try running some old SNORT rules to see what I mean

Dealing with false positives

- Turn on your IDS for the first time, and you'll be inundated with alerts
- Find attacks you just don't care about
  - Old attacks, you're patched
  - Scans
  - Turn them off
- Turn everything off
  - Turn on things you're interested in
  - Like a firewall

Encryption

- After key exchange, IDS misses
  - SSH
  - IPSEC
  - SSL
    - https://victim/awstats/?configdir=|cmd
    - SSL accelerators
    - Give your IDS the keys (!)

Packet loss

- Drops packets
  - Misses part of an attack
  - Session desynchronized for a period
- Quoth Paxson: 30% of Internet connections are asymmetric
  - Not too much of a problem on leaf networks

Signature development

- How soon after a vulnerability goes public does the vendor (or community) release rules?
  - They dont have special information you don't
  - Reverse engineering patches
    - in the case of "specially crafted" or "malformed"
  - Timeframe is usually measured in hours
    - too late?

IDS rocks

- Is IDS dead?
  - No, market research companies are dead
    - Just kidding
- Is it hopeless?
  - No, as long as you understand the limits
- Policy
  - Inverse firewall rules
  - Better than tcpdump

Evaluating an IDS

- Very, very competitive market
- Read reviews
  - Pounds of salt
  - How fast can your NIDS ignore packets?
- Test them yourself
  - Run exploits in your own production environment
  - Tcpreplay
  - Vulnerability scans?
- Number of alerts is meaningless
  - Anomaly detection doesn't map well
  - Neither does CVE
- Use what you understand and like!

IPS

- Intrusion prevention
  - Inline NIDS
    - “Bump on the wire”
  - Alerts cause traffic to be blocked
    - Drop this packet only
    - Drop packets from this host for some time
  - Has a direct effect on availability

IPS

- You must carefully consider the implications of IPS
  - Attacker spoofs malicious UDP packets from `*.root-servers.net`
    - Game over

Future of IDS

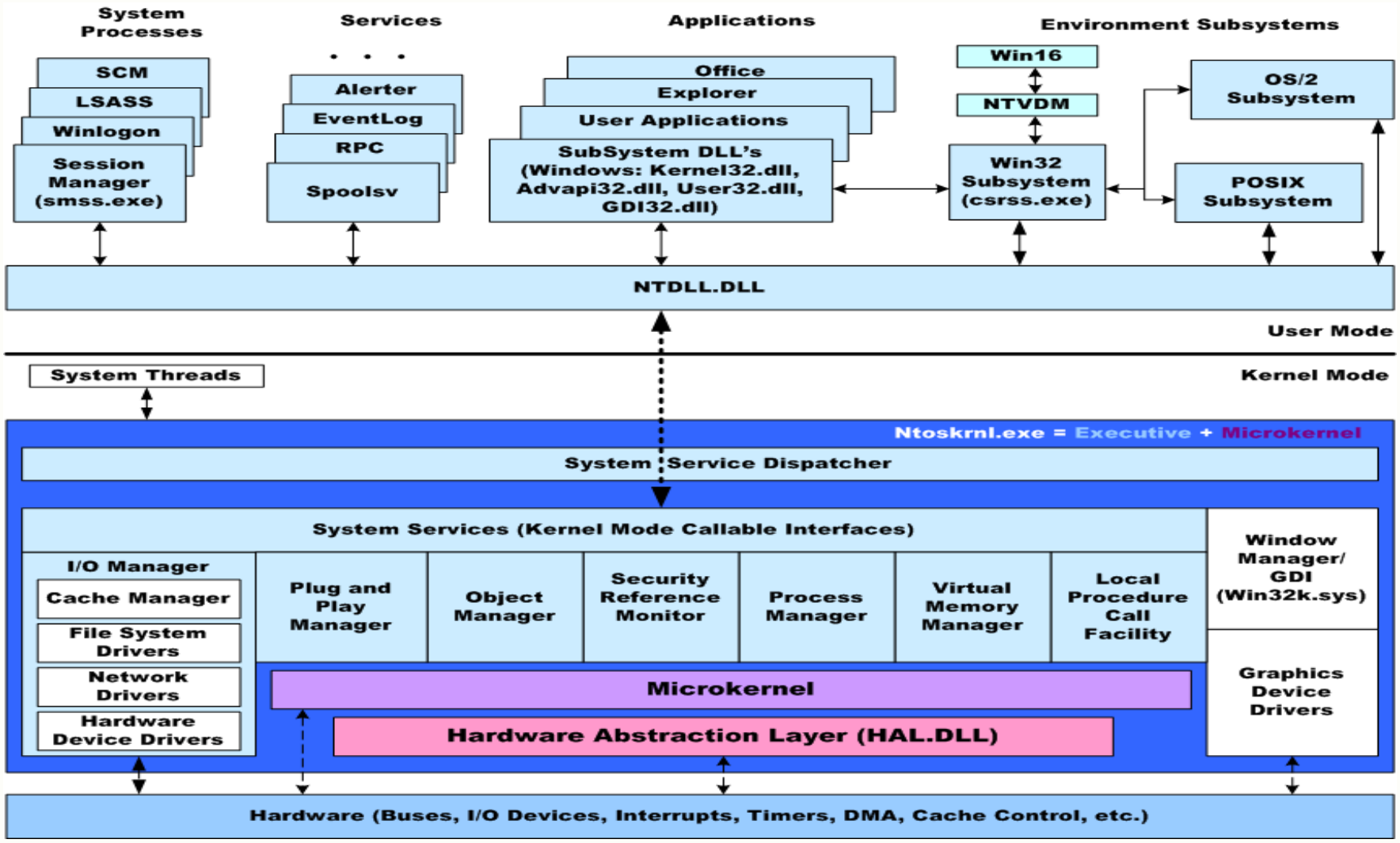
- Smarter correlation
  - Attack properties -> actual attack
- Storing earlier packets
  - “Packet TIVO”
- Flow-based
  - Networks are getting too fast
  - Better at anomaly detection
  - Arbor Networks, etc
- IPv6

Host-based IDS

Host Based IDS (HIDS)

- Network layer
  - Not vulnerable to obfuscation games
  - HIDS sees exactly what the application layer sees
- Library proxy
  - Entercept/Cisco CSA/etc
  - Did you really get every one?
    - Multiple kernel entry points in Windows
    - Hundreds of ways to execute a program

Windows NT Architecture



HIDS

- Behavioral
  - Behavior is learned
  - Has this program ever executed cmd.exe before?
  - Has it ever generated network traffic to this host?
- Explicitly specified policies
  - Provos' "syscall shim"
    - `systrace`
  - Mandatory access control policy implementation
    - `SELinux`

HIDS

- Binary checksums
  - Tripwire
- Log aggregators can take HIDS & NIDS input
  - Correlate your own events
- HIDS are “push”, NIDS are “pull”
  - Have to manually deploy HIDS
  - NIDS see everything

Tripwire

- HIDS tool
- Initially creates hashes of all stored files in a database
- Subsequently compares stored hashes to files and reports any changes
- Configuration
  - twcfg.txt - general configuration
  - twpol.txt - policy: what files to monitor, what file attributes to monitor for change, what to do if changes are detected
    - As shipped, monitors a large set of standard files
    - You will want to modify this file for your site
- Security
  - Site passphrase - encrypts and signs Tripwire files
  - Local passphrase - needed to run Tripwire

Tripwire lab

- Install
  - `cd /usr/local/lab/tripwire`
  - `sudo ./INSTALL.sh`
  - File `/tmp/victim` created for Tripwire to trip over later
- Configure
  - `sudo tripwire-setup-keyfiles`
    - Create passphrases when prompted
  - Signs and encrypts the Tripwire configuration and policy files
    - Enter passphrases when prompted
- Initialize database
  - `sudo tripwire --init`
    - Enter passphrase when prompted
  - Creates the encrypted database

Tripwire lab

- Check integrity -- should show no changes
  - `sudo tripwire --check`
    - Report sent to standard output and saved as *.twr in report directory
- Change something and re-check integrity -- should show change
  - Change something about /tmp/victim
  - `sudo tripwire --check`
    - Shows changes
  - `sudo ./Report`
    - `Shows all generated reports`
      - File name reflects date and time file was generated
  - `sudo ./Report` *`report.twr`*
    - Shows report *report.twr*

References

- Stefano Zanero, “Benchmarking IDS,” BlackHat 2006. http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Zanero.pdf (accessed April 2010)
- NSS IDS/IPS reviews http://www.nss.co.uk/
- Thomas Ptacek and Timothy Newsham, “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” Secure Networks, 1998. http://insecure.org/stf/secnet_ids/secnet_ids.pdf (Accessed April 2010)
- fragroute http://www.monkey.org/~dugsong/fragroute/ (accessed April 2010)
- Snort http://www.snort.org/ (accessed April 2010)
- Martin Roesch, “Snort - Lightweight Intrusion Detection for Networks,” 13th LISA Conference, pp. 229-238, 1999. http://www.usenix.org/event/lisa99/full_papers/roesch/roesch_html/ (accessed April 2010)
- NFR http://www.nfr.net/
- ISS http://www.iss.net/
- Niels Provos, “Systrace - Interactive Policy Generation for System Calls.” http://www.citi.umich.edu/u/provos/systrace/ (accessed April 2010)
- Loscocco, P. and S. Smalley, “Integrating Flexible Support for Security Policies into the Linux Operating System,” Proceedings of the FREENIX Track, Usenix Technical Conference, June 2001.